

Management for Professionals

Dominik Maximini

The Scrum Culture

Introducing Agile Methods in
Organizations

Second Edition

 Springer

Management for Professionals

More information about this series at <http://www.springer.com/series/10101>

Dominik Maximini

The Scrum Culture

Introducing Agile Methods
in Organizations

Second Edition

 Springer

Dominik Maximini
Novatec Consulting GmbH
Leinfelden-Echterdingen, Germany

Copyediting provided by Kevin Lee Potter

ISSN 2192-8096 ISSN 2192-810X (electronic)
Management for Professionals
ISBN 978-3-319-73841-3 ISBN 978-3-319-73842-0 (eBook)
<https://doi.org/10.1007/978-3-319-73842-0>

Library of Congress Control Number: 2017964607

© Springer International Publishing AG 2015, 2018

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Why a Second Edition?

My friend Kate Terlecka, one of the leading people in the Polish agile community, recently told me: “This book is excellent and so advanced, it’s hard to read it cover to cover. It’s a great reference and textbook though.” This is the same message I got from other professionals as they worked their way into this book.

This situation is also reflected by the numbers: as of August 2017, more than 84,000 people had read this work. Almost all of them accessed it via their universities. Compared to the total numbers, only very few people from the professional world were reading this book. However, these professionals are equally important for me as the academic readers. So my first product increment failed to appeal to both groups. This second increment is aiming at remedying that. I hope you will find this version much more easily digestible than the last one but still without lacking the desired depth of information.

Enjoy the read!

Leinfelden-Echterdingen, Germany
August 2017

Dominik Maximini

Foreword by Gunther Verheyen

One of the aspects of Scrum that is often perceived as very difficult is the simplicity of Scrum. Despite the lightness of Scrum, each of the elements in the Scrum framework does tie back to existing problems. It is a challenge to adopt Scrum from this understanding and implement it without additional phases, roles, and bureaucracy and still solve many problems in your product development.

There is, however, an even deeper challenge related to that. Scrum is much more about people, behavior, and culture than it is about “process.” The mind-set and principles underlying the Scrum framework hold at least an implicit indication of the environment in which Scrum will thrive. But, despite the growth and popularity of Scrum, it has remained implicit. The relationship of Scrum and culture remains unexplored.

Dominik acknowledged this gap and this need for exploration and has done so greatly. His book “The Scrum Culture” offers us a highly structured insight into his journey, findings, and discoveries, the path he followed, his conclusions, his definition of the Scrum culture, and the consequences when adopting Scrum. In the absence of existing resources on the relationship of Scrum and culture, Dominik created one.

Take your time to read his book. Digest the insights you will get from it. Absorb it. And use it to look beyond the mechanics of Scrum. Use this book, “The Scrum Culture,” to reflect on your organization’s culture and be prepared for some real improvement. Increase the benefits of your Scrum adoption by understanding the impact on your organization’s culture.

Antwerp
May 2014

Gunther Verheyen
Directing the Professional series at Scrum.org

Foreword by Christiaan Verwijs

The official Scrum Guide by Ken Schwaber and Jeff Sutherland counts only 16 pages. If size is any indication of difficulty, it might leave the reader with the impression that implementing the Scrum framework is easy. It's simply a matter of setting up a backlog and a Scrum Team and working through said backlog in short iterations that produce something of value to the organization. What can be hard about that?

But reality is—as it always is—quite different. The first challenge lies in achieving a thorough and deep understanding of why Scrum (through its empirical process control) works better for most projects than a planned approach. Without this understanding, it's easy to revert back to old behavior and familiar-but-ineffective approaches when the going gets tough. The second challenge lies in making Scrum work within the culture of an organization. Although Scrum is only a framework in which people can address complex problems, it espouses a great number of implicit values and norms on how a team should behave, communicate, work together, and perceive their work. Take, for example, the emphasis that Scrum places on not having a project manager that distributes tasks, makes decisions, and organizes work. The entire Scrum Team is responsible, and should autonomously self-organize toward a mode that works best for them. There are many important implicit values and norms to be found here, such as an emphasis on democratic decision-making, a need for open communication and continuous learning, a strong reliance on (personal) autonomy, and a desire to be involved in the process. If these implicit values and norms don't align with the (organizational) culture in which Scrum is implemented, and no attention is paid to this alignment, the change program will fail or not be as successful as it could've been. Although there are no statistics available for the success rate of Scrum change programs, the success rate of organizational change programs in general is low at best (<30 %, Fine et al. 2008). A common theme in organizational development is that a misalignment with organizational culture is the root cause of this low success rate (i.e., Schein 1992). And this is no surprise, as changing organizational culture is perhaps the most difficult part of any organizational change. Most of the implicit values and norms that make up organizational culture act as a powerful inertial force against change.

So, as Scrum practitioners we have to pay more attention to the alignment of organizational culture with the espoused values and norms of the “Scrum culture.”

But what does this “Scrum culture” look like? What values and norms are espoused, exactly? What methods and interventions can help to achieve better alignment? This wonderful book by Dominik Maximini is the first solid, well-researched publication to provide answers in this important area. In this book, you’ll find a wealth of information on Scrum and organizational culture, the results of an expansive quantitative study on “the perfect Scrum culture,” and a great number of methods and interventions that can be of help.

Building on the research and practices presented in this book, the community of Scrum consultants, trainers, change agents, coaches, and teams can certainly help more organizations implement Scrum successfully. And hopefully, the research in this book will also act as a stepping stone for more scientific research into this interesting area of organizational development.

Utrecht
May 2014

Christiaan Verwijs
Organizational Psychologist, Scrum Master and Trainer

References

- Fine, D., Hansen, M. A., & Roggenhofer, S. (2008, November). *From lean to lasting: Making operational improvements stick*. The McKinsey Quarterly.
- Schein, E. M. (2004). *Organizational culture and leadership* (3rd ed.). Jossey-Bass.

Acknowledgments and Disclaimer

I would like to take this opportunity to thank all those who helped me with this book. Without the help of all these people, you, dear reader, would not hold this book in your hands.

First I want to thank the great authors on whose writings this book is based. It's always easy for a dwarf to look big when he is standing on the shoulders of a giant. Dr. John Kotter's excellent work—especially his book “Leading Change”—provides a great framework for changing organizations. Kim Cameron especially helped me, not only by writing what he did, but also by allowing me to use his work and publish the results. Of course, without the pioneering work of Ken Schwaber, this book would not even have been thinkable. His uncomplicated willingness to help has supported me more than once.

Secondly, the people who guided me most on the scientific side: Brigitte Ott-Göbel, who always found time to answer my questions, no matter how trivial they might have been; and Christiaan Verwijs, who enlightened me on the matter of statistics and helped me both in reformulating my questions to make them analyzable and in actually running the analysis with the proper statistical methods.

Thirdly, the people who supported me in other regards: Jean-Pierre Berchez greatly helped in making it possible to distribute the paper version of the survey on both the Scrum Day Germany and the AgileTour; Elke Maximini, whose sharp mind helped me more than once to circumvent literary cliffs; Dieter Maximini whose faith in me never wavered and always gave me new strength when I needed it most; and Tina Harder, who is the person who suffered most due to my engagement. She raised my spirits whenever they were down and endured my isolation when I had to get some writing done. Of course I want to thank my customers, without whom I would never have been able to collect the experiences that led to this book. Chances are that the help of Glenn Lamming and Marc Armbrust ensured that you, dear reader, do not have to suffer too much since they greatly improved the quality of my wording, grammar, and punctuation.

This all would not have been possible without the participation of hundreds of Scrum practitioners around the globe. Every single one of them invested in some cases more than an hour to participate in the survey—that is, 13,740 min or 29 working days. Thank you for that!

If you think I should have mentioned you as well but you didn't find your name up there, be assured I did not leave you out on purpose. I most likely remembered you the minute this book went to print—sorry about that!

My final thanks go directly to you: Thank you for reading this and investing your precious time.

You will find both male and female use of pronouns in this book. However, the use is not necessarily balanced. This is not an attempt to put anybody at a disadvantage. Instead, it is intended as a simplification. Thank you for your understanding.

Introduction

Agile methods, especially Scrum, have become enormously popular. A Google search returns 30,000,000 hits today; Amazon shows 1,398 results worldwide when looking for the subject. Presumably, the number will have risen again by the time you are holding this study in your hands. This flood of information will help you get a deeper understanding of agile practices. It will not, however, help you in attaining a deep understanding of Scrum's culture and how to transition to it. Unfortunately, that is exactly what managers need to know in order to be prepared for what they will need to navigate through. Managers, developers, and Scrum Masters worldwide struggle every day to introduce Scrum. They fail over and over again, mostly for the same reasons. Those reasons originate from the Scrum Culture, which is often at odds with the existing corporate culture. The first step to tackle that is to know what you are getting into. Therefore this book is not about technical tools or Scrum's practices—you can find plenty of information about that elsewhere. This book is about the Scrum Culture and viewing it from an organizational change point of view.

When interacting with customers as a Scrum coach, I see Scrum implementations too often where all parties believe it has been a step backwards rather than a leap forward. At the same time, Scrum itself is very simple. The official "playbook" is no more than 17 pages long (cf. Schwaber and Sutherland 2016), including the cover sheet, table of contents, and acknowledgments. The issue is that management of the resulting consequences is a highly complex endeavor, which is easily underestimated. Personal sensitivities, dependencies on other processes and departments, fears, and reliance on other relationships are the traps one can easily fall into.

In this book, you will discover how to avoid such failure. You will not learn much about the Scrum mechanics.¹ Only the most important facts are stressed in order to avoid misunderstandings.

The book is split into six big parts. Part I is made up of the synthesis of the evaluation of different organizational culture models, the information gathered via

¹ Read the Scrum Guide, <http://scrumguides.org/> (Schwaber and Sutherland 2016).

primary and secondary research, and the conclusions drawn from it. It is the definition of the Scrum Culture. Part II will provide you with the theoretical knowledge about organizational change, garnished with some real examples from my daily practice, while the practical application is described even more vividly in Part III. To deepen this knowledge as well as enabling you to immediately picture it mentally in practice, a detailed case study is provided in Part IV. Although it never actually happened exactly as described in this book, individual elements are or were real.

Part V provides you with the detailed scientific study, without bothering you with statistical analyses.

In Part VI, the appendices, you will find more information and tips on what to pay special attention to when considering Scrum, or elements of it. This information is limited to the fundamentals and provides you with the most relevant information only, in just a few sentences. In addition, some deeper scientific material is provided.

This book deliberately does not provide you with checklists for use in a Scrum transition. The reason is context, the fact that every organization and project has special individual circumstances, rules, and problems. What works for one project can prove fatal in the next. What you do get though are summaries of all chapters in Part II. This will help you to recap what you have read and to look it up quickly whenever you need it.

Before we begin, please heed a word of warning: By introducing Scrum you enter the field of organizational change. This is not child's play. Great opportunities come head-to-head with great risks. I strongly warn you not to use the contents of this book blindly. Question whether what you have read is applicable to your context. At the end of the day, you—and nobody else—are responsible for your deeds and their consequences.

Throughout this book, you will encounter parts that are formatted like this paragraph. Those are about personal experiences I want to share with you. I tell you the “inside story,” so to say. The purpose of these passages is to share the learning I have been able to make with you. Judge yourself whether the case studies described here could also be applied to your context, and whether you could benefit from them.

Contents

Part I The Scrum Culture

1 Scrum Culture Definition	3
1.1 Why a Scrum Culture is Important	3
1.2 A Brief Definition of Organizational Culture	4
1.3 Scrum Culture Elements	5
1.4 Consequences for Enterprises	9
References	16

Part II The Theory of Introducing Scrum

2 Different Shapes of Scrum in the Enterprise	19
2.1 Scrum PRN	19
2.2 Virtual Scrum Software Studio	20
2.3 Scrum Software Studio	21
2.4 Façade Scrum Organization	23
2.5 Profound Scrum Organization	23
2.6 Sustainable Profound Scrum Organization	24
2.7 Which Scrum Shape for Which Goal?	25
References	27
3 Different Starting Points	29
3.1 Top–Down Introductions	29
3.2 Bottom–Up Introductions	30
3.3 Submarines	31
3.4 Choosing the Right Starting Point	32
Reference	32
4 Considerations for Scrum Introductions	33
4.1 Reasons for a Scrum Introduction	33
4.2 Stakeholder Management	36
4.3 Different Situations and Product Types	37
References	38

5 Scrum Introduction Overview 41
 5.1 Introducing Scrum with Scrum 42
 5.2 Multi-Change Initiatives/Change Programs 43
 References 44

Part III The Practical Application of Kotter’s Principles

6 Creating a Sense of Urgency 47
 6.1 The Impediment-In-Depth-Analysis 49
 6.2 Velocity Extrapolation 50
 6.3 General Advice for Creating a Sense of Urgency 51
 6.4 Things You Should Remember 52
 Reference 53

7 The Guiding Coalition 55
 7.1 Composition 55
 7.2 Organization 57
 7.3 Tasks 60
 7.4 Things You Should Remember 61
 Reference 61

8 Vision and Strategy 63
 8.1 Vision and Strategy in the Context of Leadership and Management 63
 8.2 How to Create Vision and Strategy 64
 8.3 Things You Should Remember 66
 References 66

9 Communicating the Change Vision 67
 9.1 What You Always Wanted to Know About Communication 67
 9.2 Things You Should Remember 69
 References 69

10 Empower Your Employees on a Broad Basis 71
 10.1 Transform Affected Parties into Involved Ones 71
 10.2 Typical Inhibitory Factors Regarding Scrum 73
 10.3 Things You Should Remember 81
 References 81

11 Generate Quick Wins 83
 11.1 Why Dreamers Need an Alarm Clock 83
 11.2 Characteristics of Quick Wins 84
 11.3 Pilots 85
 11.3.1 Identification 86
 11.3.2 Setting Up a Pilot Project 88
 11.3.3 Implementation 92
 11.3.4 Sprint 1 96
 11.3.5 Sprint n 98

11.4	Common Problems	98
11.5	Measurement of Results and Reporting	107
11.6	Things You Should Remember	109
	References	109
12	Consolidate Gains and Initiate Further Change	111
12.1	Promotions and Other Felonies	111
12.2	This Is Only the Beginning	112
12.3	Things You Should Remember	114
	Reference	114
13	Anchor New Approaches into the Corporate Culture	115
13.1	Origins of Culture	115
13.2	Anchoring	117
13.3	Things You Should Remember	118
	References	119
14	Introducing Scrum into Large Teams	121
14.1	Special Circumstances	121
14.2	Basic Rules for Scaling	123
14.3	Direct Comparison of Small and Large Introductions	124
14.4	Coordination	126
14.5	The Right Time	128
14.6	Things You Should Remember	128
	Reference	129

Part IV Case Study

15	Introducing Scrum	133
15.1	The Hardest Part: Starting Out	133
15.2	Urgency	137
15.3	The Guiding Coalition	141
15.4	Vision and Strategy	148
15.5	Communication	153
15.6	Empowering Broad-Based Action	155
15.7	Generate Short-Term Wins	162
15.8	Consolidating Achievements and Initiating Further Change	166
15.9	Embedding New Concepts in the Corporate Culture	172
15.10	Beyond the Case Study	174
15.11	The Actors in Brief	175
	Reference	176
16	Get Started!	177

Part V The Scientific Part of the Scrum Culture

- 17 Starting With Science** 181
 - 17.1 Problem Definition 181
 - 17.2 Goals of the Research Project 182
 - 17.3 Scientific Approach 183
 - 17.4 Expected Results 184
 - References 184
- 18 Organizational Culture Models** 187
 - 18.1 Different Approaches 187
 - 18.2 Model Selection 189
 - 18.3 A Broader View on Cultural Dimensions 199
 - References 203
- 19 Cultural Characteristics of Scrum** 205
 - 19.1 The Origins of Scrum 205
 - 19.2 General Cultural Characteristics of Scrum 207
 - 19.3 Specific Cultural Aspects of Scrum 216
 - References 219
- 20 Primary Research: The Nature of Scrum Survey** 221
 - 20.1 Study Setup 221
 - 20.2 Findings from the Organizational Culture Assessment Instrument[©] 223
 - 20.3 Findings from the Open Questions 226
 - References 231
- 21 Conclusions** 233
 - 21.1 Limitations of This Study 233
 - 21.2 Opportunities for Further Research 235
 - References 237

Part VI Additional Information

- 22 Appendix 1: Why John Kotter’s Model Was Chosen** 241
 - 22.1 How Mike Cohn ADAPTs to Scrum 241
 - 22.2 Schwaber’s Playbook from ‘Software in 30 Days’ 243
 - 22.3 Leading Change Like Kotter 245
 - 22.4 How Cameron and Quinn Use the OCAI for Change 249
 - 22.5 Why Kotter? 252
 - References 253
- 23 Appendix 2: Research Details** 255
 - 23.1 The Nature of Scrum Survey Questions 255
 - 23.2 Statistical Analysis: Descriptives 267
 - 23.3 Statistical Analysis: Data Preparation 278
 - 23.3.1 The OCAI Analyses Follow from Here 278
 - 23.3.2 Now the Analyses of the Open Questions Follow 282

23.4	Summary of Scrum’s Cultural Characteristics According to Literature	285
23.4.1	Summary of Survey Findings	289
23.5	Findings Comparison	293
	References	300
24	Appendix 3: A Brief Scrum Overview	301
24.1	Scrum Roles	301
24.1.1	Product Owner	302
24.1.2	Scrum Master	304
24.1.3	Development Team	306
24.1.4	Management	306
24.1.5	You as Change Manager	307
24.1.6	The Scrum Consultant	308
24.1.7	Approach to Fill the Roles	309
24.2	Artifacts	310
24.2.1	Product Increment	310
24.2.2	Product Backlog	310
24.2.3	Sprint Backlog	311
24.2.4	Definition of Done	311
24.3	Events	312
24.3.1	The Sprint Itself	313
24.3.2	Sprint Planning	314
24.3.3	Daily Scrum	314
24.3.4	Sprint Review	315
24.3.5	Sprint Retrospective	315
	References	316
25	Appendix 4: Methods	317
25.1	Planning Poker	317
25.2	Planning Poker for Absolute Numbers	318
25.3	Estimation Meeting	318
25.4	Timebox	319
25.5	Velocity	319
	Index	321

List of Figures

Fig. 1.1	Scrum culture values	9
Fig. 1.2	Individual Scrum culture cell	13
Fig. 1.3	Scrum culture body	14
Fig. 2.1	Scrum PRN in the enterprise (<i>Source NovaTec Consulting GmbH</i>)	20
Fig. 2.2	Virtual Scrum Software Studio in the enterprise (<i>Source NovaTec Consulting GmbH</i>)	21
Fig. 2.3	Scrum Software Studio and the enterprise (<i>Source NovaTec Consulting GmbH</i>)	22
Fig. 2.4	Profound Scrum in the enterprise (<i>Source NovaTec Consulting GmbH</i>)	24
Fig. 2.5	Sustainable Profound Scrum in the enterprise (<i>Source NovaTec Consulting GmbH</i>)	25
Fig. 4.1	Complexity in projects, according to Stacey (Based on Stacey 1996, p. 47; Schwaber and Beedle 2002, p. 93)	35
Fig. 5.1	Finding the right way	42
Fig. 5.2	Many small changes on the way to great change	44
Fig. 6.1	Realistic time extrapolation for complete delivery of all features	51
Fig. 18.1	Harrison’s culture model (Based on Harrison 1972, 1987)	189
Fig. 18.2	Deal and Kennedy’s culture model (Based on Deal and Kennedy 2000a, b)	191
Fig. 18.3	Schneider’s culture model (Based on Schneider 1999)	193
Fig. 18.4	The competing values framework (Based on Quinn and Rohrbaugh 1983)	195
Fig. 18.5	Cameron and Quinn culture model (Based on Cameron and Quinn 2011, p. 39)	195
Fig. 18.6	Culture change over time (Based on Cameron and Quinn 2011, pp. 64–65)	197
Fig. 19.1	Agile organizational structure (Based on Gloger and Häusling 2011, p. 31)	219

Fig. 20.1	Cumulated responses over time	222
Fig. 20.2	Scrum in Cameron’s and Quinn’s culture model (Based on Cameron and Quinn 2011, p. 39)	224
Fig. 24.1	Scrum viewed as a framework	302

List of Tables

Table 1.1	Organizational impact categories	14
Table 2.1	Scrum shape overview	26
Table 4.1	Suitability of Scrum	38
Table 18.1	How three theoretical perspectives complement each other	188
Table 18.2	Decision matrix for model selection	198
Table 18.3	Questions to deepen cultural insights	202
Table 19.1	Contrasting traditional and agile organizations	216
Table 20.1	Means of all OCAI dimensions	224
Table 20.2	All OCAI means	224
Table 20.3	OCAI results, experience comparison	225
Table 20.4	Jokes in a Scrum environment	228
Table 22.1	The ADAPT essence	242
Table 22.2	The CXO playbook	244
Table 22.3	Kotter’s approach	246
Table 22.4	Cameron’s and Quinn’s nine step approach	250

Part I

The Scrum Culture



1.1 Why a Scrum Culture is Important

Even though not every project is conducted in an agile way, the numbers are increasing considerably. “In 2002, agile projects made up less than 2 % of overall projects and less than 5 % of new application development projects. Today, agile projects account for almost 9 % of all projects and 29 % of new application development projects [. . .]” (Standish 2011, p. 1). The most popular member of the agile family is Scrum. According to a Forrester survey (2012, p. 15), 81.5 % of the respondents are using Scrum. VersionOne (2013, p. 5) is backing this tendency by stating that 72 % of their respondents are using Scrum at least partially. Other agile methods play a minor role (cf. VersionOne 2011, 2013; Standish 2011; Forrester 2012).

Scrum is defined as “a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value” (Schwaber and Sutherland 2016, p. 3). This does not explain the nature of Scrum or agile product development in depth. However, the ability to manage changing priorities, increased productivity, improved project visibility, improved team morale (cf. VersionOne 2013, p. 10), higher effectiveness, quality, and business stakeholder satisfaction (cf. Ambler 2008) with this approach speak for themselves. The Standish Chaos Manifesto even goes so far as to state: “The increase in project success rates can directly tie back to projects resolved through the agile process” (Standish 2011, p. 1).

Looking at the tremendous success numbers presented by all these sources, the question arises why more projects are not conducted using the Scrum approach. One reason is, that “becoming agile is hard” (Cohn 2009, p. 17). Cohn also stated, “It is harder than most other organizational change efforts I’ve witnessed or been part of”. Forrester supports this: “The future of Agile is bright, but only if you deal with the change management required to introduce it in your organization. Initial adoption is one thing, scaling it through the enterprise another” (Forrester 2012, p. 26). Analyzing failed agile projects highlights the same issue. The leading cause

for these failures is the “company philosophy or culture at odds with agile values” (VersionOne 2013, p. 6). If other culture related reasons are included, they even account for more than half of all failed agile projects.

In order to manage these issues and raise the project success rates even higher, Scrum has to be understood from a “corporate culture” point of view. Only if this dimension is understood, the compatibility and impact on existing corporate cultures can be anticipated and potentially resolved.

1.2 A Brief Definition of Organizational Culture

There are a multitude of authors who talk about organizational culture. Personally, I prefer Schneider’s definition: “Organizational culture is the way we do things in order to succeed” (1999, p. 128). Ed Schein went into more detail and specified three levels at which culture manifests (cf. Schein 2009, pp. 39–40):

1. External Survival Issues
2. Internal Integration Issues
3. Deeper Underlying Assumptions

The external survival issues are everything an external visitor can observe in the enterprise: Company mission and vision, strategy, goals, and the means to implement them. This involves organizational structure, the systems and processes used, including error-detection and correction.

Internal integration issues cover the parts of a culture only an employee of the company can perceive after some time in the organization. External visitors will most likely not be able to identify these aspects. They include the usage of a common language and common concepts as well as the answer to the question who is an “insider” and who is an “outsider”. Essentially, group boundaries and identity are defined on this level. The internal integration issues also describe how status and rewards are allocated, how authority is distributed, and how work relationships are established between people. For example, authority could be distributed based on technical knowledge (the one most knowledgeable on the issue at hand is followed) or based on position (the one with the most stripes on the shoulder has the last say).

The deeper underlying assumptions are difficult to name, even for people who have been with the company for many years. They contain basic assumptions about what makes the world go round, including questions like:

- What is the relationship of humans to nature like?
- What is reality?
- How is truth defined?
- What comprises the human nature?
- What defines human relationships?
- What is the nature of time and space?
- What do we believe about the unknowable and uncontrollable?

These questions are rather more philosophical than economic. In a business context, you only encounter them on rare occasions. However, they do have a huge impact on our everyday actions. Imagine two people coming from different cultures: One person believes it to be absolutely true that time is absolute and static. Once gone, it's over and lost. The other person is absolutely sure that time is elastic, moving relatively compared to the situation. A sentence like: "The project must be finished by date X", will be interpreted in a significantly different way by these two people. . .

The following chapter applies the findings of a scientific study to this specific concept of organizational culture. If you want to learn more about organizational culture in general, take a look at Part V of this book.

1.3 Scrum Culture Elements

Having reviewed literature and analyzed the findings from the Scrum Culture Survey, a pattern becomes apparent. All findings point in the same direction, in that there is no fundamental gap between literature and the expectations of individuals (cf. Sect. 23.5). While it was not clear at the beginning of this study if something such as a "Scrum culture" existed, it is now obvious that indeed people expect Scrum to work and succeed in conjunction with certain circumstances, values, and rules. Since people tend to project their expectations onto organizations and shape them accordingly, an impact on organizations has to be expected. This impact will be similar across a multitude of enterprises, therefore it can be said that Scrum has inherent cultural characteristics. A Scrum culture does exist.

This Scrum culture can be described with existing cultural models such as the Competing Values Framework, measured by the OCAI (see Parts V and VI of this book for further information). Such analyses are ideal to gain a first impression of what one is dealing with. However, if a more in-depth analysis is sought for, such models are not sufficient. As of today, I do not know of any cultural model that would thoroughly describe all aspects of Scrum. Therefore, a new model has to be created from the findings. I call this model "Scrum Culture".

The Scrum Culture's jargon revolves around teams, empirical process control, products, planning, business, IT, leadership, and Scrum. In addition, some predictive process modeling jargon is used, but usually not in an appreciative way. "Waterfall" is the adversary of Scrum, because Scrum practitioners believe that complex problems cannot be solved with predictive methods; but in the past, they were forced to do so and punished for the resulting failure. This leads in a strong rejection of everything that has to do with waterfall thinking: Authoritarian leadership, top-down management, micromangement, predictive planning, power struggles, information hiding, and so on. People seemingly showing attributes of that kind of thinking—no matter if they really do think in this way—will be turned down and confronted with sarcasm in the form of pointed jokes. This is also true for people "not getting Scrum" or violating its rules. They even run the risk of being considered an "outsider" and consequently being excluded from interaction, teamwork, and information flow. This is noteworthy since, in general, nobody is excluded from the Scrum Culture. It does not matter which department somebody

is working in, where the person comes from, what the person's background is, and so on. The only thing that matters is that this person fully participates as a member of the team and contributes something valuable to the overarching goals.

When one enters an organization living the Scrum Culture, a couple of things become visible. Firstly, people do not wear any sort of uniform but rather dress very casually. This ranges from T-shirts to sportswear or leather jackets to shirts and blazers, everything is possible and usually all occur at the same time. The only things one will not see are suits and ties. Those are implicitly forbidden unless important visitors (e.g. customers or sponsors) are expected. This is true for everybody in the Scrum Culture, not just the developers.

The next thing that is visible is the nature of the working space. It does not exhibit uniformity but is very individual. Sizes, shapes, and arrangements will vary widely throughout the company. Some desks might be adjustable to allow for working while standing or some people may sit on special chairs or inflatable balls. All desks are arranged in team rooms, bringing together everybody who has a role to play in the task the team is currently working on. This should usually not exceed 20 people. These team rooms are very communicative and spacious areas, supplemented with meeting rooms, quiet areas for individual work, and lounge areas with comfortable sofas, a coffee machine, and maybe even games in the form of table soccer, video games, or something similar. Friendly, comfortable, motivating, spacious, bright, creativity promoting, and inspiring are the words best used to describe these team rooms. The walls are plastered with information. Some is directly related to the work at hand, some connects the day-to-day business to the larger goals. Some information does not have anything in the slightest to do with the job at hand but is meaningful to the team working there. Whiteboards, flipcharts, and hundreds of colored sticky notes can be found everywhere. The whole workspace is constantly changing since the team owns and adjusts it according to their needs whenever it seems appropriate. This could even happen several times a day. While this might appear chaotic to an outsider, it is not for the team. Everything is always clean and structured in a way to support the work at hand. If some team members are not located in the same room—which is rarely the case in the Scrum Culture since face-to-face communication is highly valued—huge video-conferencing monitors will be visible in the team room with cameras and headphones available to ensure open communications at all times. Most team members use this equipment every single day. Due to the work being performed, one encounters a constant buzz in the team room. As in a beehive, there is a steady humming in the air, but never too loud to inhibit others from continuing to work. Heated discussions pop up from time to time and are quickly moved to a place where others are not disturbed, such as a meeting room.

People address each other on a first name or nickname basis and value each other equally in discussions. This is valid for everybody, not just the team members. In meetings, everybody is collaborating to find the best solution. These discussions are facilitated (often by a Scrum Master) and thus stay very goal oriented and focused. The communication is very open, no power struggles or politics exist. People treat each other with respect, even when discussions get heated, which often happens

during creative collaboration. Everyone participates, everybody is engaged. All opinions have equal weight even though on specific topics such as the product requirements or the Scrum process, the Product Owner or the Scrum Master respectively has the final say. Everybody has the opportunity to be heard and the person with the greatest knowledge or the most experience is valued most highly. Ranks and titles do not exist. In the end, the best thought out idea wins and is chosen via consensus or vote, not by top-down decision.

If somebody disagrees with anything, it is immediately and openly voiced. Since this is done in a respectful manner, even the manager can be criticized in front of a wider audience. People in the Scrum Culture prefer knowing the painful truth immediately rather than living comfortably but unsuccessfully, oblivious to reality. Since they constantly learn about their own shortcomings and those of their processes, they constantly develop themselves and adapt all processes to support their needs. The Scrum Culture is a place of constant change.

People are rewarded for their openness, alongside teamwork, supportive behavior, individual improvement, and active, engaging participation. Punishments are rare but do exist for behavior that degrades performance or productivity of the team, for example uncooperative, competitive, and antagonistic behavior. Lone wolves are hunted down, as is everybody who violates the rules or values of Scrum. This is a natural cultural defense mechanism since such antagonistic behavior would directly attack the Scrum Culture and therefore must be contained. Every deed worthy of reward or punishment is immediately communicated to the responsible person. Usually the people affected by the deed make sure the originator knows. On some occasions, it is the Scrum Master or Product Owner who talks to the person; but this is usually done in addition to and after the direct team feedback. Management is not the primary source of feedback to the individual employee.

If an employee is rewarded, this is usually not done in the form of a promotion. In fact, there is no ladder to climb in a Scrum culture. One can change roles, receive a salary increase, or be assigned to more important projects. It is important to realize that promotion means personal development rather than getting the corner office, a bigger company car, a better parking lot, or an important sounding title. If an employee gets a new title, it is usually one invented by the team to call out a novel aspect of their role like “chief motivation artist” or “senior build breaker”. It is not connected to privileges or power. Such team “gaming” is quite common in the Scrum Culture. Hierarchies are extremely flat since the teams organize themselves and everybody is actively collaborating with each other. There is no need for formal information hubs or centralized decision-making. Therefore, few management positions exist, let alone aligned in a pyramidal way.

Accordingly, what is expected from management is not the same as in traditional enterprises. In a Scrum Culture, management believes that people want to work and do not have to be closely monitored. So management refrains from closely controlling and instead focuses on forging the strategy and removing impediments to the teams’ success. They see themselves as serving their teams, not as being their superiors. Decisions are made at the position where the best experts for the issue are located. This is usually decentralized and closest to the work at hand. Generally,

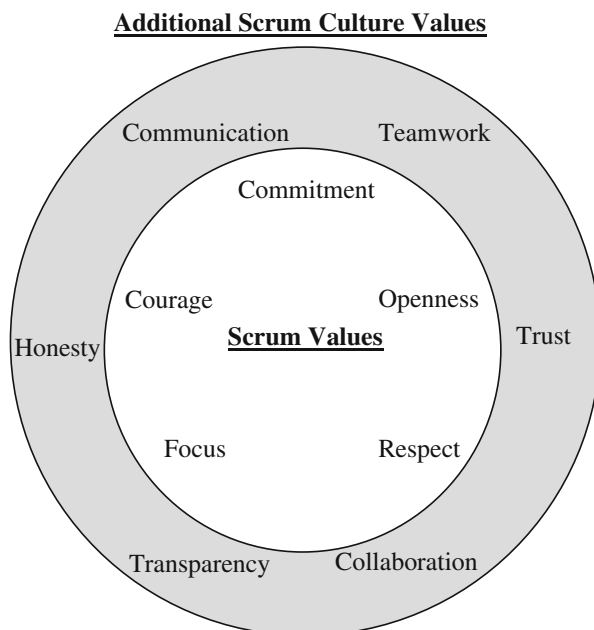
decisions are not made alone in a closed room, but the “wisdom of the crowds” (cf. Surowiecki 2005) is facilitated, resulting in democratic decisions with strong and mutual commitment. All decisions are aligned with the corporate and product visions, which of course have to be well prepared and communicated by management. Leaders in a Scrum Culture see themselves as growing and nurturing their teams so that they can be or become self-managing and mature. To achieve this, managers take care of their individual employees in frequent mutual feedback loops, for example on a weekly basis. Annual appraisals might still happen, but are just useful to wrap up all the individual talks that happened throughout the year. Thus, managers have a very close connection to their employees and can coach or mentor them in improving their personal and professional skills. The development of the team is managed by the Scrum Master, who is also a servant leader who mentors the team to grow and mature. Of course, the teams do not work completely unmonitored. In fact, they closely monitor themselves on a daily basis. Every single team member is able to explain to anybody within minutes what the exact status of the work is and how the team is planning to reach the goal at the end of the iteration. While these explanations can be very technical, the Product Owners can easily explain the essentials to outsiders at any time in their own language. However, the Sprint Review meeting is an event specifically designed to allow for outside monitoring and close collaboration with stakeholders. So full transparency into the real status reduces the urge of management to control adherence to a predictive plan.

Both managers and team members despise working overtime since it diminishes the value of the work created (quality decreases once the performance peak is reached) and reduces the creative potential of the people involved. Scrum Teams create increments of releasable product every iteration and thus replace the traditional notion of “being late” with iterative-incremental progress. So instead of working overtime, a sustainable pace of work is aspired. Teams decide themselves if overtime makes sense, provide it at their own discretion, and then balance out their total working hours as quickly as possible. This is sometimes difficult because team members look forward to coming to work and feel very good about being there. They sometimes tend to overwork themselves and must be supported by their Scrum Master who sends them home before they are completely exhausted.

Whenever a team encounters something unknowable or uncontrollable, it quickly faces and deals with it. The Scrum Culture sees the whole world as a complex place that is, to a large extent, unknowable and uncontrollable. Meeting such uncertainty is normal and is accepted if not embraced. Working in small timeboxes reduces complexity, experiments are run, and the outcomes are used as a basis for learning. A Scrum Culture breaks down large chunks of uncertainty into smaller portions and deals with them empirically.

In everything they do, people working in a Scrum Culture focus on continuous improvement, both for themselves and for their products. Customer delight is reached via teamwork and happy employees are paramount since only happy developers can generate happy customers. Success is measured in value provided to the customer rather than presence at the desk.

Fig. 1.1 Scrum culture values



The espoused values of the Scrum Culture support this focus. Openness, trust, teamwork, respect, transparency, honesty, courage, commitment, communication, goal orientation/focus, and collaboration govern every single action of each employee. This is noteworthy since this extends the values espoused as “Scrum values” by the creators of Scrum (cf. Fig. 1.1).

The roles, artifacts, meetings, and rules connected to Scrum support these values. They are the soul of the Scrum Culture.

This Scrum Culture, introduced into a company with another culture that is at odds with the Scrum Culture, can have both a tremendous positive or negative impact on the organization.

1.4 Consequences for Enterprises

Talking in terms of the Competing Values Framework, the more pronounced the hierarchy of an existing organizational culture, the bigger the problems when introducing Scrum. A hierarchy culture is trying to control every single process, make it stable, and adhere to it. This is not compatible with the worldview of the Scrum Culture, which sees the whole world as complex and changing all the times. In their eyes, no process can be perfectly planned, so process adherence does not make sense either. Instead, common sense and empirical methods (trying things out, then learning, and drawing conclusions) rule—which are fundamentally at

odds with a hierarchy culture. Different perceptions of the world are hitting each other head on and lead to a cultural clash of epic dimensions.

The same is true for an intense market culture. Here, similarly huge conflicts arise when Scrum is introduced. A market culture believes that the external environment is hostile and the primary goal is winning in the marketplace. Leaders are driving their employees quite hard, which often leads to strong competition between individuals. While a Scrum Culture also has a goal to win in the marketplace, it does not perceive the world as hostile and does not allow people to compete with each other at the cost of company success. Working in partnership with customers at a sustainable pace, nurturing employees, and fostering innovation are the chosen focus.

The first question for every enterprise considering Scrum is whether Scrum is appropriate for the task at hand. The Scrum Culture fits perfectly into a complex environment such as research or development. It does not fit as well into a production environment. The Scrum Culture can either be introduced throughout the whole enterprise or within a specific department or business unit. Accordingly, either the department or the whole organization is affected by the changes. These changes are largely the same, but if the core culture of the company differs too much, constant cultural battles have to be expected.

Derived from the findings described above, there are eight organizational areas in which the Scrum Culture has special demands that might influence, contradict or change the corporate culture into which it is merged:

- (1) Management style and leadership
- (2) Decision making
- (3) Cadence and speed
- (4) Planning
- (5) Focus on productivity
- (6) Soft factors
- (7) Hierarchy
- (8) Organizational structure

The Scrum Culture demands a very high degree of involvement and hence a democratic management style. Management is expected to provide leadership by providing a clear vision and strategy, being a role model in living the values, and by acting as a partner of the employees. Authoritarian styles are despised and lead to conflict. Managers are change champions of the Scrum Culture and have to shape the organization in a way that allows the culture to unfold and blossom. They live and protect the spirit of the Scrum Culture.

Decisions are not made by a central authority but rather by the people who are affected by the issue at hand. This usually means that it is not management but rather regular employees who make the decisions. This is not done by individual heroes but rather by whole teams that are empowered by the organization. What they decide stands. Of course, if new aspects arise and a better decision can be

made, the team will re-assess their earlier arbitration. The decision process is a democratic one, but not always based on consensus.

The cadence in the Scrum Culture is speedy and cyclic. Work is performed in short iterations; the same is expected from all processes in the affected part of the company (or the whole company). This means that within a standard iteration (1–4 weeks; usually it is 2 or 3 weeks), all issues have to be solved. Somebody needs a new laptop? A decision has to be made? A call for proposals has to go out? A new employee is sought and a job advertisement has to go live? It is all expected to happen within a single iteration. In each iteration all aspects of change can happen in order to improve the overall results. This change is welcomed. Of course, interaction with one's manager also happens in short cadences, on a weekly, bi-weekly or monthly basis. This is the heartbeat of the Scrum Culture and every part of the organism that is supplied with blood will need to follow this rhythm.

This cadence also applies to the planning process, which might have to fundamentally change. The Scrum Culture expects different kinds of problems, for which individual tools are needed to solve them. If the issue is complex, meaning that a lot of uncertainty is involved, the solution can only be found empirically by trial and error. For such problems, for example software development, no long-term predictive plans can be adhered to. Due to the highly unstable nature of the issue, the exact solution, required effort, and so on will evolve and cannot be perfectly predicted. Certainty is impossible for complex situations—otherwise they would not be complex. This means that fix-price contracts are difficult (if not impossible), especially if that means that scope and time are also fixed. It also means that budgeting processes might have to be rethought—the teams might discover new aspects, which could lead to an expense that has not been planned for at the beginning of the year. As long as the opportunity is sound, organizational processes may still support this. So budgeting and planning processes also have to follow the cadence of the Scrum Culture.

The next fundamental change is about the company's focus on productivity. In many of today's enterprises, the focus clearly is on efficiency.¹ This leads management to make sure every employee is busy with a 100 % workload. Experts are developed to make sure every task is performed with the highest possible skill. The Scrum Culture has a different approach since it is not individual tasks but rather the overall system that has to be optimized. So people plan themselves for a load of 70–80 % and have the rest of their time to decide on the spot where their help is needed most. They are also expected to be generalists rather than experts and should be able to lend a hand wherever needed. This leads to people not performing as seasoned experts for every task, but still pushing the whole solution forward in a way that optimizes effectiveness rather than efficiency. By doing that, risk of sickness, quitting, change of roles, holidays, etc. is also minimized. This does not mean however, that efficiency is neglected. It just takes a back seat to effectiveness.

¹ Simply speaking, “effectiveness” is about doing the right things while “efficiency” is about doing things right.

All these changes come together with a strong focus on people. In the Scrum Culture, people are never described as “resources” because they are not seen as easily interchangeable. Instead, the organization realizes that team performance very much depends on soft factors. So it is made certain that people sit in the same room, are empowered to shape their environment, teams are kept as constant as possible, and managers look out for the emotional well-being of their employees. Changes are discussed before they are introduced, coffee corners are communication hubs, and facilitators (or even mediators) are always available to solve conflicts before they erupt into explosions. People are key. They are like erythrocytes: they transport oxygen (e.g. work results) to all parts of the organizational body and take carbon dioxide (e.g. organizational problems) back out. People make the organization breathe.

Since every single employee is that important and teams are empowered to solve their own problems, hierarchies become largely obsolete. People talk to each other and think for themselves. They do not need a “boss” who does that for them, like Taylor described in 1911. Of course, there are still line managers who care for their individual employees. However, there are not several levels of hierarchy competing for power. “Career” does not mean rising in the ranks but rather evolving one’s personality and trying something new or being part of a more important project. On the team level (e.g. a software development team), there is no “tester” who is less important than a “programmer” who is less important than an “architect”. They are all “developers”, striving to achieve the same goal, hand in hand, as equals.

Since hierarchies change, so does the organizational structure. There is no pyramid of people and pillars of functions. Rather, there are bubbles, or cells, of products including all functions and people necessary to be successful. The organization consists of many cells, each nurturing their own product. These cells are largely independent. While central support functions might exist, they are just responsible for supporting and not for dictating. All cells are constantly communicating and collaborating with each other in order to fulfill the overarching company vision and strategy. If one cell fails, the people are distributed across the other parts of the organizational body. If new opportunities arise, people throughout different parts of the organization form a new cell. Since everybody should know each other on a first name basis, each cell will usually not exceed 150 people² and often be much smaller. A Scrum cell’s purpose is not purely product development, but rather wraps around everything relevant for the product or product line. This involves sales, operations, production, development, finance, and anything else that is necessary in operating the business.

Such a Scrum Culture cell structure consists of an enveloping “meta-organization” containing the managers and “benches” of all departments or “job families” (cf. Sect. 19.3). The members of the meta-organization are responsible for crafting the company strategy and managing the corporate culture. The managers are also disciplinarily responsible for their employees, spread across the Development

²This is called “Dunbar’s number”, described for example in Sutcliffe et al. (2012).

Teams. The meta-organization is like a cell wall, enveloping the nucleus, protecting it from harmful influence but letting everything that is healthy transmit through. The cross-functional Development Teams form this cell nucleus. All necessary members from all relevant functional areas are assembled in these teams. Figure 1.2 shows what such a cell could look like.

The interaction of all Scrum Culture cells is not restricted in any way. Instead, it is facilitated and encouraged. After all, every cell belongs to the same body—if the body fails at large, so does every cell. Therefore, management makes sure that vision and strategy are known and understood at all times. In addition, constant communication across cells guarantees that individual deviations stay within necessary and acceptable limits. Management does not perform this communication on behalf of the teams. Instead every individual team communicates as needed. An illustration of a Scrum Culture body, consisting of many cells as shown above, can be found below. Arrows indicate communication lines (Fig. 1.3).

The changes in the eight dimensions mentioned are fundamental and disruptive. Of course, not every organization that implements Scrum will change all aspects.

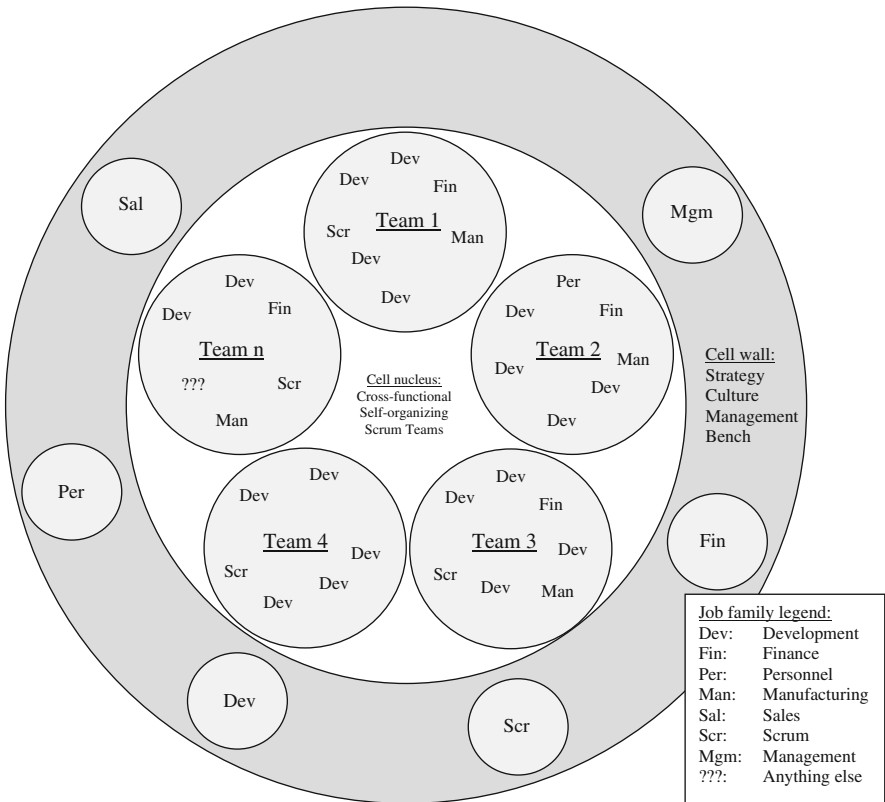


Fig. 1.2 Individual Scrum culture cell

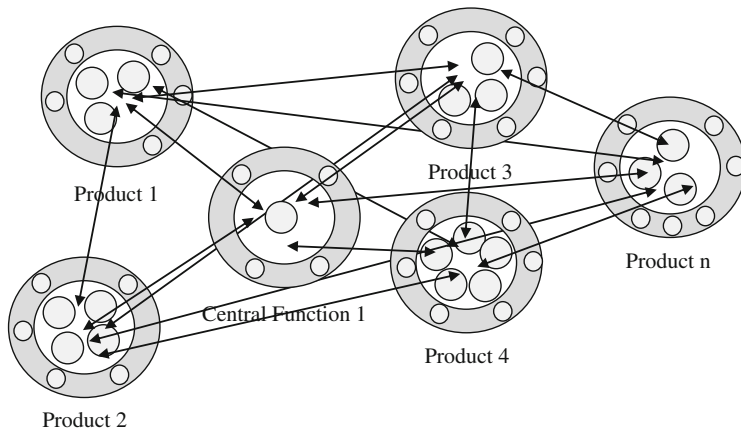


Fig. 1.3 Scrum culture body

Table 1.1 Organizational impact categories

Impact category	Required change
Management style and leadership	<ul style="list-style-type: none"> - Involving and democratic - Management provides clear vision and strategy - They are a role model in living the values - Management acts as partner of the employees - They champion change necessary for the Scrum Culture - Leaders live and protect the spirit of the Scrum Culture
Decision making	<ul style="list-style-type: none"> - People who are affected make decisions - Broad empowerment - Teams decide, not individuals - What they decide stands - Decisions can be re-assessed with new information - Democratic process - Consensus is not always needed
Cadence and speed	<ul style="list-style-type: none"> - Speedy and circular - Constant change is welcomed - Short iterations in all processes - One standard iteration has to be enough - Interaction with one's boss every iteration - This is the heartbeat of the Scrum Culture
Planning	<ul style="list-style-type: none"> - Only solutions to simple problems can be planned - Solutions to complex problems cannot be planned - Solutions evolve empirically - No long-term predictive plans can be adhered to - Certainty is impossible - Fix-price projects should be avoided - Budgeting for iterations, not years

(continued)

Table 1.1 (continued)

Impact category	Required change
Focus on productivity	<ul style="list-style-type: none"> – Effectiveness is more important than efficiency – Productivity of the whole system is more important than individual productivity – It is never planned for a 100 % workload. 70–80 % is enough – Generalists with special expertise are preferred to one-topic experts – Minimize risk of absence
Soft factors	<ul style="list-style-type: none"> – Strong focus on people – People and resources are two different things – Collocation – Teams are empowered to shape their environment – Teams are kept constant – The emotional well-being of employees is cared for – Changes are discussed prior to introduction – Facilitators are always available – People make the organization breathe
Hierarchy	<ul style="list-style-type: none"> – Hierarchies are largely obsolete – People talk and think for themselves – Line management cares for their individual employees – Career means personal development, not rising through the ranks – Career might also mean participating in a more important project – Everybody on a team is equal
Organizational structure	<ul style="list-style-type: none"> – No pyramid of people – No pillars of functions – Small (<150 people) cells of product-focused cross-functional teams work closely together – Management protects the teams like a cell wall – Communication and collaboration happens directly between the teams of different cells – All cells pursue the same organizational vision – Central functions may exist but do not dictate decisions

However, each aspect that is not changed will potentially lead to costs in terms of conflicts, lower productivity, and reduced employee satisfaction. That is, the Scrum Culture and non-Scrum culture will struggle with each other for survival. If only one part of the organization, e.g. the development department, is changed the fight will continue at the borders. In this example, central functions such as controlling, sales, purchasing, quality management, and finance might be in constant struggle with the Scrum Culture. Either they develop a mutual understanding and a way of peaceful coexistence, or one will have to be eliminated sooner or later. There cannot be two competing corporate core cultures at the same time.

For quick reference, Table 1.1 shows the organizational impact categories.

References

- Ambler, S. (2008). *Agile adoption rate survey results: February 2008*. <http://www.ambysoft.com>. 31.12.2012. <http://www.ambysoft.com/surveys/agileFebruary2008.html>
- Cohn, M. (2009). *Succeeding with Agile. Software development using scrum*. Amsterdam: Addison-Wesley.
- Forrester. (2012, April). *November 2011 global agile software application development online survey*. Forrester Research.
- Schein, E. H. (2009). *The corporate culture survival guide* (New and Revised ed.). San Francisco: Jossey-Bass.
- Schneider, W. E. (1999). *The reengineering alternative. A plan for making your current culture work* (Special reprint ed.). New York: McGraw Hill.
- Schwaber, K. & Sutherland, J. (July 2016). *The scrum guide*. Scrum.org. 28.08.2017. <http://scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf#zoom=100>
- Surowiecki, J. (2005). *The wisdom of crowds* (New ed.). Trade Paperback. New York: Anchor Books.
- Sutcliffe, A., Dunbar, R., & Bunder, J. (2012, May). Relationships and the social brain: Integrating psychological and evolutionary perspectives. *British Journal of Psychology*, 103(2), 149–168.
- Standish. (2011). *Chaos Manifesto. The laws of CHAOS and the CHAOS 100 best PM practices*. Boston: The Standish Group International.
- VersionOne. (2011). *State of Agile survey: Sixth annual survey*. Version one. 31.12.2012. http://www.versionone.com/pdf/2011_State_of_Agile_Development_Survey_Results.pdf
- VersionOne (2013). *7th annual state of Agile development survey*. Version one. 04.08.2013. <http://www.versionone.com/pdf/7th-Annual-State-of-Agile-Development-Survey.pdf>

Part II

The Theory of Introducing Scrum



There are different forms of Scrum, which are found again and again in different companies. Before we can look at how to reach a *desired* target shape, we must first be aware of the *possible* target shapes. Those are presented below. With the concepts I have stayed close to Ken Schwaber and Jeff Sutherland, who ensured a common understanding in their book “Software in 30 Days” (2012).

2.1 Scrum PRN

PRN comes from Latin “pro re nata” and means “take as much as needed” (Fig. 2.1). Scrum PRN thus denotes organizations where Scrum is used as needed—usually without a formal support organization and often driven by the people down the line. Most of the time, Scrum PRN implementations are created unconsciously or even virally by a single successful Development Team switching to Scrum. Other teams see this and then also want to use Scrum. In some cases, management is aware of Scrum’s possibilities and orders its use in critical situations. Organizational changes are confined to the team level. With Scrum PRN it is difficult though, to hold on to gained knowledge. Once a PRN project is over, hard-won knowledge is not retained and people are usually distributed across other projects. At the end of the day, the company has gained no added value on the process level. It is also problematic that every Scrum implementation within the company differs from the rest and insights are not bundled. Every team makes their own mistakes, creates their own good practices, and is independently successful—or not. Companies using Scrum PRN are giving away considerable potential, but can certainly be successful.

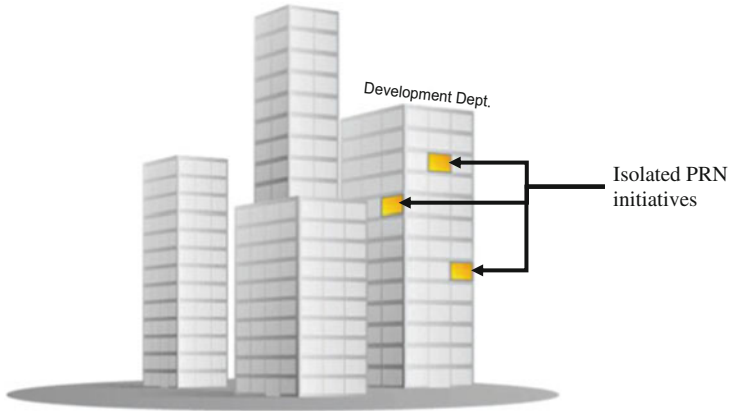


Fig. 2.1 Scrum PRN in the enterprise (*Source* NovaTec Consulting GmbH)

2.2 Virtual Scrum Software Studio

There are two forms of a Scrum Software Studio (Fig. 2.2). The first resides within the organization and pretty much plays along with the rules. This one is called the Virtual Scrum Software Studio. The second is a dedicated legal entity and sets its own rules. It depends on the situation at hand, which one is preferred. With a Virtual Scrum Software Studio, the complete organizational structure of the parent company stays intact. The Scrum function usually is added to an existing staff department, such as a project management office (PMO). In addition, the decision makers have the choice between Scrum and traditional methods for every project. Standard processes are established to start new Scrum projects, alongside helpful tools like checklists, a knowledge database, and training. The whole initiative for a Virtual Scrum Studio originates most often from two or three people who have experienced the value of Scrum for themselves but were not able to transport this value to the whole organization.

The biggest advantage of the Virtual Scrum Studio is its low barrier to entry. The organization does not have to change much; there is hardly any “pain” for the involved parties. This is paid for though with a lot of problems: The studio rules potentially cannot be pushed through, which leads to low productivity and lower employee satisfaction. Working spaces and other resources are often deficient. For example, it is difficult to seat the whole team in one room. Visibility within the organization is also not increased since projects run “normally”, just using a different process description, which might not even be read by some project managers. In addition, to solve organizational issues, the regular escalation paths have to be used; there is no shortcut through the Scrum Studio manager. Change endeavors meet heavy resistance this way and are—if at all feasible—very tedious. If you want to introduce such a Virtual Scrum Studio, you should plan the transition to a real Scrum Studio from the very beginning. You should have enough data after some successful projects to convince your management of the associated benefits.

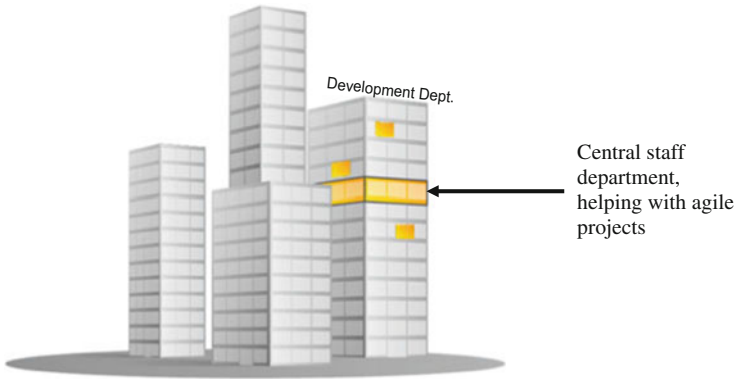


Fig. 2.2 Virtual Scrum Software Studio in the enterprise (Source NovaTec Consulting GmbH)

Be aware that the risk of falling back into old behaviors and losing what small progress was achieved is enormous with a Virtual Scrum Studio.

A large company decided to introduce Scrum into its entire development organization, mid term. While it was not clear to everybody why the company wanted to do this, a small group of brave change agents saw advantages for their teams. So they started the endeavor. Due to their lack of knowledge of the various shapes of Scrum as well as their advantages and disadvantages, this company found itself with a Virtual Scrum Studio one year later. Scrum made the problems of this organization transparent and the developers liked the new approach. Unfortunately, no additional benefits could be realized. As in other projects, the company spent a lot of energy on predictive planning, project members worked on several projects at the same time, and the only room that—after a long time—could be used as a team room was an unpopular basement office. This basement office, however, was not viewed as an issue by the project staff. Everybody only spent a small portion of their time there, since they were going to be deployed to a different project in a couple of months anyway and in the meanwhile had to work on several other projects from their standard office. Productivity stalled and some people became very frustrated.

2.3 Scrum Software Studio

A Scrum Software Studio is a learning institution, belonging to the parent company, which supports Scrum projects (Fig. 2.3). Innovation and experimentation rule here. Usually it is a separate legal entity.¹ Method knowledge is bundled here, as are work materials, metrics, and even workplaces for complete teams. For each individual project, the company decides whether it should be conducted with traditional processes or Scrum. If the decision is to apply Scrum, the project will be implemented physically within the Scrum Software Studio. The studio is an entity existing alongside established structures that does not have the right to

¹ In Germany, the “Deutsche Post E-Post Development GmbH” is such a Scrum Software Studio, for example.

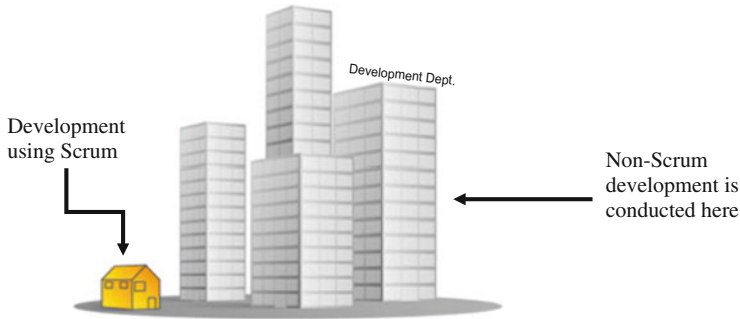


Fig. 2.3 Scrum Software Studio and the enterprise (*Source* NovaTec Consulting GmbH)

change the existing company globally. This avoids resistance since most people are not directly affected. It’s “them” over there in the Studio, not “us” in the parent company. For projects that shall be implemented within the studio, certain rules apply (cf. Schwaber and Sutherland 2012, p. 78):

1. Every project will adhere to Scrum processes and its principles of empiricism, bottom-up intelligence, and self-organization.
2. Every project will have a Scrum Development Team with a Product Owner, Scrum Master, and no more than nine developers.
3. The Scrum Master must be experienced in managing Scrum projects. To the extent that he or she is not, the person will accept guidance from Studio Scrum coaches.
4. The Product Owner will actively work with the team to formulate requirements, inspect work, inspect increments, and empirically adapt in order to optimize project value and achieve its vision or purpose. This is a hands-on role.
5. The Development Team will consist of software developers with all the skills needed to create an increment of potentially usable functionality, based on the Product Owner’s requirements.
6. Throughout the project, previous reporting relationships will be held in abeyance.
7. Each increment will conform to the Scrum definitions of “transparent” and “complete.”
8. The Scrum Team will use modern engineering practices and tools provided by the Studio and will receive training in how to use them, if necessary.
9. The project will conform to the standards of the organization and to the policies, procedures, and standards of the Studio.
10. To the greatest extent possible, the Scrum Team will be collocated within the Studio. The members will work full time on the project.
11. The Scrum Team will take advantage of the Studio’s metrics to assist it in managing its work.
12. The Scrum Team members will participate in adding to the Studio’s body of knowledge based on their experiences in working on the project.

Of course, the rules may be specific for to a given organization.

Only those who accept and follow all these rules are allowed to use Scrum for their projects. This way, the Scrum Software Studio can build significant knowledge and data over time, from which new projects benefit. Achievements also become provable. Changes in culture, mindset and approach are restricted to projects within the Studio, the organization as a whole does not have to change. Since simultaneously significant improvements are realized, this Scrum shape often gets widely supported throughout the company. The disadvantage is that projects that could greatly benefit from Scrum are not necessarily carried out using Scrum, since participation in the Studio is voluntary per project. Departments such as technical support, personnel, or controlling usually do not participate at all, which leaves significant potential untapped. Finally you should notice that the way of thinking within the company does not change, therefore conflicts and constant attempts to change the processes of the Scrum Software Studio are inevitable. Many large companies are founding such virtual studios right now. Usually, they go by names like “digital unit” or “innovation speed boat”. However, in all cases I know, the rules of a Scrum Software Studio are not applied consequently. This leads to mediocre results. A Scrum Software Studio only really works when it actually uses Scrum!

2.4 Façade Scrum Organization

Façade Scrum is the first shape that applies to the company as a whole instead of just a part of it. Usually the act of establishing this kind of Scrum originates from management. Unfortunately, it stays far from successful. As the name suggests, the company keeps all of its processes, but gives them new “agile” names. It creates a façade of Scrum terms that are not supported by the corresponding core practices and values. Frequently, one finds programmers working in iterations, calling these Sprints, but neither having a “Definition of Done” nor testers, analysts, or the like. As a result, these teams do not deliver “done” Product Increments, but only serve the next phase of the established waterfall-like development process in an iterative manner. These teams rarely get any advantage from the new approach—neither in terms of enjoying their own work, nor in terms of improved quality or productivity. Such implementations are doomed from the start. Be aware that with a Virtual Scrum Software Studio, you are constantly running the risk of creating Façade Scrum instead of being agile. Be on the watch!

2.5 Profound Scrum Organization

“Profound Scrum” refers to a Scrum implementation including the entire company in all of its areas (Fig. 2.4). Scrum has been fully understood here, every employee is trained and knows the concepts. The values and practices are well known. The mindset of the entire organization is focused, customer-centered, results-oriented, empirical, iterative, and focuses on people. Here it has been proven that Scrum

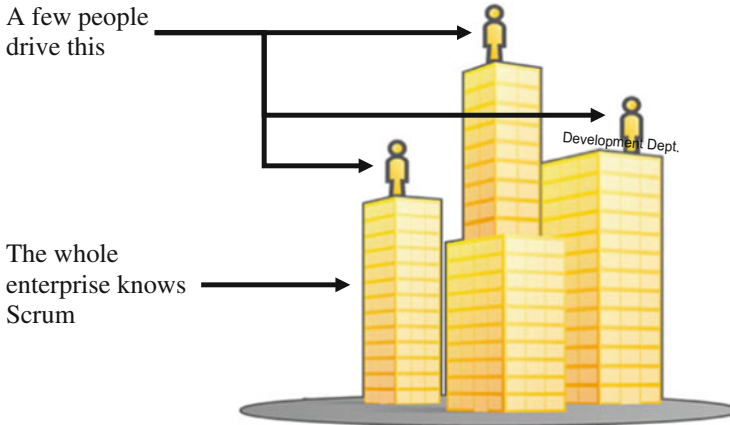


Fig. 2.4 Profound Scrum in the enterprise (Source NovaTec Consulting GmbH)

works well and employee turnover is low. To get to such a state, the company usually needs to invest 5–10 years (cf. Kotter 2012) into the change process. Often, one particular leader drives this. As long as Profound Scrum has not become Sustainable Profound Scrum, the whole organizational change effort is primarily driven by this executive champion. If she leaves the company or is promoted, the organization falls back into its old state. Although this is usually better than the original state, it is not as productive as Profound Scrum. In addition, there is a risk that the organization loses its flexibility with respect to outside changes. Thus, today's well fitting state can be tomorrow's problem. You easily get to this unsustainable Profound Scrum if you only correctly implement the first of the steps proposed in this book. Especially not properly anchoring the change leads to sustainability remaining a distant vision.

2.6 Sustainable Profound Scrum Organization

As with Profound Scrum, Scrum's success has been proven and staff turnover is very low (Fig. 2.5). The difference is that instead of a single lighthouse figure driving the change process alone, a broad base of people is rallied behind the cause. If the leader leaves the company, the ideas and drive stay. The organization continues to change and remains open to new ideas. The changes are deeply anchored into everyone's mind.

A company having established Sustainable Profound Scrum can confidently meet any challenge and develop appropriate solutions. All employees are highly motivated. The whole thinking and acting is focused on inspection and adaptation. Through transparency and focus both the early identification of problems and a productive way of working are ensured. Although you will have to adapt your processes over time to new circumstances, the values will remain intact and give you a significant competitive advantage. The Scrum Culture thrives here. Values,

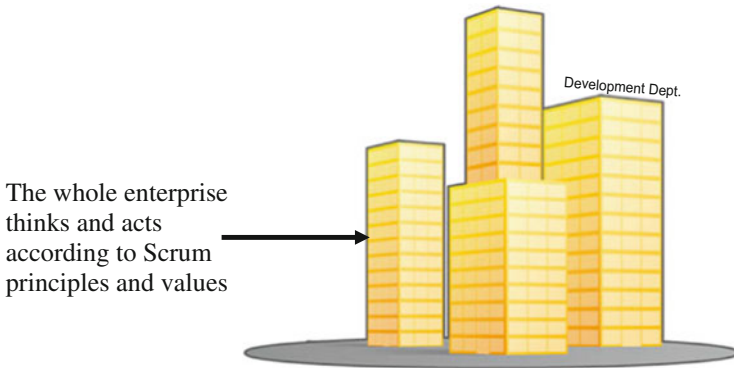


Fig. 2.5 Sustainable Profound Scrum in the enterprise (*Source* NovaTec Consulting GmbH)

processes, and structures are changed. For example, the organization will no longer have a multi-layered management pyramid, but instead wraps itself around cells of independent but interconnected units, sharing the same vision.

In this book we are mainly concerned with this very Sustainable Persistent Scrum. The reason is that this is not only the biggest challenge to introduce, but also potentially bears the greatest benefits for the enterprise.

The other shapes can be achieved with the same eight steps, but the actual implementation of these steps diverges to some extent.

2.7 Which Scrum Shape for Which Goal?

Before you start implementing Scrum, you need to make a decision for one of the described Scrum shapes. Depending on your position within the company, the awareness of your management team and the overall objective you are following, the result can vary significantly. Ask yourself the following questions:

1. What do I want to achieve in my current situation with the introduction of Scrum?
2. What does the company want to achieve with the introduction of Scrum?
3. Who is the “driver” of the Scrum implementation and what position does he have in the company?
4. What supporters do you have for your endeavor?
5. How many teams and individuals are directly affected?
6. Over what time period should the result of the Scrum introduction be visible? Only for the duration of the project or beyond?
7. What budget do you have?
8. Which experts can you count on?

If your only goal is to “save” a very critical project quickly, then Scrum PRN could be right for you. Although you turn down long-term gains across the company, you can quickly realize success for your current project. Even if you do not have the necessary power to introduce any of the other Scrum shapes, you can still start with Scrum PRN.

The Virtual Scrum Software Studio is useful if you already have gathered some experience with Scrum in the organization, but top management does not want a culture change or otherwise denies their full support. The virtual studio allows you to make success highly transparent, to gather experience in a structured way, and to productively develop products. However, you will most likely be constantly under attack from other parts of the company, as the Studio contradicts the corporation’s core culture. These conflicts can be dampened by creating a true Scrum Software Studio, outside the legal entity of the parent company. Here, you can set the rules and are not bound to standard processes.

If you have already gathered some positive experience and also have top management support, you can dare to try Profound Scrum. Your entire company can then benefit from the new way of thinking, improve productivity and experience delighting work. To achieve Profound Scrum, however, you must be patient. Five to ten years are common. If you want the whole thing to be sustainable, you must have found yourself that diamond project: difficult to find, extremely hard, but crystal clear and beautiful with a value that everyone understands and appreciates (Table 2.1).

Table 2.1 Scrum shape overview

Shape	Short description	Advantages	Disadvantages
Scrum PRN	Scrum can be used as needed, without fixed introduction processes	Quick; Can be done without management support; initially no consequences for existing structures	Knowledge is not maintained; Project participants are distributed to other (non Scrum) projects afterwards; Only team problems can be solved; Just small productivity gains, if any; No improvement for the organization as such
Virtual Scrum Software Studio	An internal organizational structure (e.g. staff department) helps setting up Scrum projects and keeps knowledge and specialists centralized	Knowledge is preserved; Standardized procedures and good practices are available; Small impact on existing structures; Can be the starting point for Profound Scrum	Necessary to follow the standard company procedures in many cases; This leads to only small productivity gains; Prone to conflicts and power play; Centralized functions are not in line with the Scrum Culture

(continued)

Table 2.1 (continued)

Shape	Short description	Advantages	Disadvantages
Scrum Software Studio	An organizational unit (own legal entity) conducts all Scrum projects of the group	Clear separation of traditional and agile world possible; High productivity gains possible; Clear rules and responsibilities for agile projects; The group can stick to old processes	The organization has to be willing to form a new entity and staff it with experts; Advantages for the whole group are just slowly and rarely implemented. A strong urge by the mother organization to control the Studio can quickly destroy the advantages
Façade Scrum	The old processes are kept but are labeled with new agile sounding names	Nothing changes for the organization	Nothing changes for the organization; Employees usually see through the plot and become unhappy
Profound Scrum	All parts of the enterprise know about Scrum and agility. They try to live and support Scrum	The whole organization is agile and can quickly react to a changing environment; All parts are focused on customer delight, work hand in hand, and live continuous improvement; extremely high productivity	This means a change of the organizational culture in many cases; Not every employee will feel comfortable here; Weaknesses of the organization become painfully transparent; It takes a long time to establish this shape; If one of the change champions leaves the company, the whole organization can fall back into old patterns
Sustainable Profound Scrum	As Profound Scrum, but when a change champion leaves, Scrum is still lived	Every single employee lives by the Scrum/ Agile values; The Scrum Culture has been established; All processes are culturally aligned	Old career models, hiring criteria, and maybe even the organizational structure have to be changed; People who fight the Scrum Culture may have to leave the company

References

- Kotter, J. (2012). *Leading change* (New ed.). Boston: Harvard Business Review.
- Schwaber, K., & Sutherland, J. (2012). *Software in 30 days*. Hoboken: Wiley.



There are several ways to start a Scrum introduction project. This depends considerably on the desired Scrum shape. In some companies, management decides—often after a relaxing and insightful golf duel—that Scrum is to be used throughout the whole company. Sometimes such a decision is the result of “lessons learned” of past projects. In both cases, management is backing the Scrum introduction and provides budget as well as other support.

However, more often it happens that an individual team or a single development department discovers Scrum for themselves and starts the introduction “bottom-up”. Quite often, upper management is unaware of the experiments, and therefore cannot support them.

“Submarines” are those changes or projects that consciously “dive under the radar”. The people responsible do not want upper management to see the project.

Below you will learn about the specific advantages and disadvantages of these approaches. However, the goal must always be the same: make the business more productive, move people into the center of your focus and work away with fun and pride. This only works if you get all parties into the same boat.

3.1 Top-Down Introductions

Ideally, you already have management support from the very beginning. This allows you to operate openly and show successes publicly. Do not let yourself be fooled: In the long run, you can only be successful if you have the support of your management. Small successes are obtainable without their help, but achieving impact within the organization is only possible if you can turn the big wheels—and these are in the hands of your management team. That’s a good thing, because power and responsibility should not lie too far apart. If you are successful (or striving for it), then you will end up having management “on board”.

The benefits of top-down implementations are obvious: a clear mandate, budget, personnel, active support and a clear goal will push your Scrum project ahead. Most

often you will also have access to training, external consultants, and other useful tools. In return, you have to generate results that justify the investment. You probably do this anyway; hardly anybody starts with a risky endeavor if he does not intend to emerge successfully and profit from it. Of course, the pressure generated can also be perceived negatively. You also still need to convince your Development Team and possibly other colleagues that a Scrum introduction is worthwhile for them as well. Another disadvantage is that in top-down implementations the belief often dominates that there is such a thing as a “silver bullet” against the “werewolves” of failing projects (cf. Brooks 1995). This leads to unrealistic expectations about speed and effectiveness of a Scrum introduction. Management can hardly be blamed for this: Most of them just fell victim to unrealistic expectations of their colleagues. There are even consultants who promise productivity increases of a factor of 10 in order to acquire new customers. This is nonsense of course. A Scrum introduction as such gives you true transparency, but no productivity gains; you need to remove barriers to productivity of your organization (they are called “impediments” in Scrum) in order to realize these gains.

If you are considering a top-down introduction, you can establish all shapes of Scrum in the enterprise. You have plenty of options at your disposal—assuming you can convince your management.

3.2 Bottom-Up Introductions

Most often, the introduction of Scrum is initiated “from below”. It is usually an experienced developer or a development manager who shows initiative and sees Scrum as the solution to her problems. An insular “trial balloon” is started and tolerated by senior management (if senior management would actively support it, it would be a top-down introduction). The advantage here is the high motivation of the team and the can-do attitude towards change, opposed to endless discussions common with other approaches. Also, the initial phase of the introduction is often very successful—until it reaches its natural limits. Once interfaces or processes beyond the reach of the team need to be changed, the organization slows down people’s enthusiasm and prevents further changes. From the company’s perspective this behavior is justified; after all, the processes have served them well in the past and led the enterprise to the point where it is today. The organization has problems, of course, but maybe the risk to damage well functioning processes in the course of a Scrum introduction is deemed higher than the potential benefit. Power struggles between rival managers can also kill the delicate bloom of agility quickly.

Such tainted implementations often lead initially to “hybrid processes”, that is agile sub-projects that are integrated into the context of traditional phase models. Typical characteristics are the breakdown of teams and work into technical topics instead of products and features, or the outsourcing of services to other teams (for example to a test or integration team). Such structures, also called “Water-Scrum” or “Water-Scrum-Fall”, only lead to small productivity gains compared to

traditionally organized teams. However, such approaches do not deserve to be called “Scrum”. In fact, you are looking at Façade Scrum if the fundamental elements of Scrum are not met, which is the case in the examples described above.

So with bottom-up introductions you can basically choose between Scrum PRN and Façade Scrum (if you actually want that). Depending on your specific situation, you can possibly even implement a Virtual Scrum Studio. Profound Scrum, however, is beyond your reach.

3.3 Submarines

Everyone knows them, everyone loves them: submarines. Finally the team can do what it has always held to be correct. As long as nobody notices, people simply do “what is necessary”. It does not matter what management says—we are the team. Scrum says: “The team has the power!” does it not?

Those statements are sarcastic and heavily flawed, of course, even though you sometimes meet them in the field. Submarines have to be viewed very critically. Sooner or later, Scrum fundamentally changes a company. You cannot keep that secret. Although a submarine can operate almost anywhere, its chances of survival are equal to zero once it is discovered. Since discovery is certain in the long run and the chances of survival are low, you should stay away from this approach.

Imagine yourself in the position of top management: You are trying to steer this battleship that we call business. You are responsible for the well being of your employees. You must comply with statutory regulations, otherwise you would have to close down your business. To make all this work, you have installed an elaborate system of processes. Suddenly you notice that a part of this system operates outside those processes thus putting the entire company at risk. Intolerable! You do not really know what they’re doing (no one took the time to explain Scrum and its benefit to you, after all), but one thing is obvious: It’s a secret and deviates from the established processes.

I do not know about you, but I would torpedo such a threat to my employees right away. In addition, my pride might have been hurt, because my authority was undermined. This is not a good basis for productive and factual discussion.

You should only use the submarine as a tool for introduction if you are in dire need and no other options are available to you. Also, let it rise to the surface as soon as possible. Take fresh water and top management on board. That might be more difficult, but certainly more successful in the long run. A further motivation to avoid this approach could be that the only shape you can achieve with a submarine tactic is Scrum PRN.

3.4 Choosing the Right Starting Point

However you begin, you have to strive to win both management and the team for your endeavor. After all, you are all working for the same goal: to make your company more successful and thus safeguard your workplace. Usually, the teams are quickly persuaded because they can feel the improvements within a couple of Sprints. Therefore you should start with a top-down implementation, if you have the chance. This will save you a lot of time and increases the likelihood of success dramatically. To put it more clearly: Most of the activities described in this book cannot be implemented without top management support. These measures, however, increase the likelihood of success for your Scrum introduction. If you cannot count on the support of management, you should start with a bottom-up introduction. Do not use the submarine approach if you can help it. Generate success and prove to management that the process changes are good for everyone. But the most important thing is: Get started!

Reference

Brooks, F. P. (1995). *The mythical man-month. Essays on software engineering* (Anniversary ed.). Reading, MA: Addison-Wesley.



So let us now get to the point: The introduction of Scrum into an enterprise. Before we get to the “how” of the implementation, we should first look at the “why.” The reasons listed here could possibly help you in elaborating the topic with your management. “Stakeholder management” is also briefly discussed in this chapter. Finally you will learn for which products you should consider Scrum—and for which ones you should not.

4.1 Reasons for a Scrum Introduction

You have decided to introduce Scrum. At least you are thinking about it. This is good! Whether Scrum is adequate or not has to be individually determined for every company and every project. I cannot take this work away from you. However, I can give you some food for thought along the way.

Scrum was designed for software development. In the 1990s, the industry had the same problems as it has today¹:

- Each release lasts longer than the previous one
- Release schedules slip
- Stabilization phases at the end of a release take longer and longer
- Planning seems to take a long time
- During the current release, it is almost impossible to start another one
- It is difficult to introduce changes mid-release
- The quality of software is constantly decreasing
- There is an increasing number of artifacts, documentations and other process control instruments that must be created and adhered to by the project team

¹ My thanks go to Scrum.org, who helped me here.

- “Death Marches”² are hurting the morale of the Development Teams
- Unmet customer needs are constantly increasing
- 60 % of the implemented features are rarely or never used³
- Hardly any of the project participants have transparency about what to expect from the project
- Project plans are outdated as quickly as they are created, there is no predictability
- The team members of many projects are unmotivated or bored
- The cost of projects is far too high
- The benefits of projects is not sufficiently evaluated (if at all)
- The occurrence of a risk⁴ is often first noticed at the end of the project when countermeasures are no longer possible

There are many more problems of course. So what do yours look like? What bothers you and your customers most? Take a moment now to be honest with yourself on the challenges you are facing.

In order to determine whether Scrum is generally suitable for your project or product, we need to consider the limiting factors. Those are the expected outcome, the technology used, the project environment, and of course the personalities involved. You should be able to determine the level of uncertainty in those areas without discussing “complexity” in detail. Just ask yourself the question if more is known than unknown, or the other way around. This leads to the following classification (Fig. 4.1):

Simple projects are those in which the conditions are almost completely known.

Complicated projects are those in which more is known than unknown.

Complex projects are those in which more is unknown than known.

Chaotic projects are those in which almost nothing is known.

When talking about complexity in a Scrum context, usually it is uncertainty that is discussed. Scrum was designed for complex problems. Software development is almost always complex: Customers rarely know in advance what they really want in the end, developers rarely know exactly how they are going to solve the technical challenges they will encounter, and groups of people are almost always inherently

² A “Death March” in this context means a project doomed to failure due to insufficient personnel, a fundamentally flawed schedule, inadequate funding, and/or unrealistic requirements. As a consequence, these projects are usually managed with massive pressure and driven into a wall. Edward Yourdon wrote a book on the topic (Yourdon 2003).

³ You do not believe this? Then you should check out your spreadsheet program. When clicking through the menu bar, try to remember which of these functions you use regularly and which you have pretty much neglected so far.

⁴ You can learn more about risk management from “Waltzing With Bears” by DeMarco and Lister (2003).

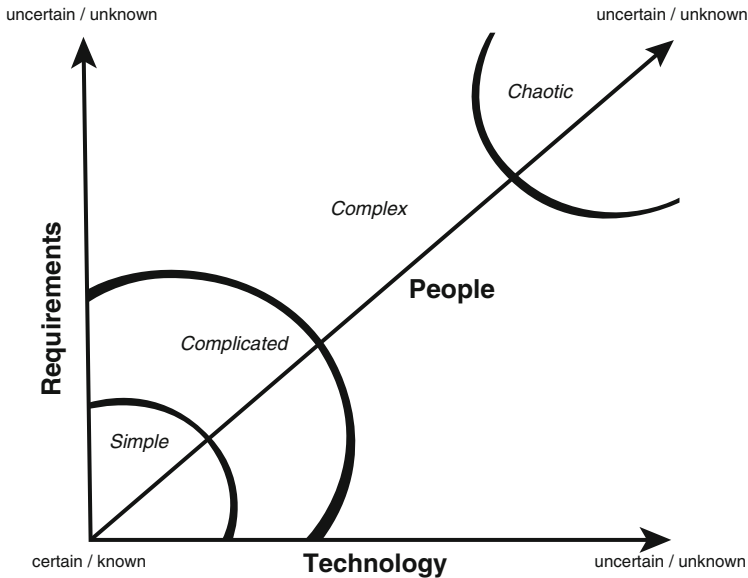


Fig. 4.1 Complexity in projects, according to Stacey (Based on Stacey 1996, p. 47; Schwaber and Beedle 2002, p. 93)

complex. Team dynamics are never simple to understand. If your environment is “complex” in this sense you can assume that Scrum is generally suitable for you.

You can use Scrum in complicated projects, but other approaches probably work just as well. For example, software maintenance often fits here. Quite often, people use Kanban approaches in the complicated realm. However, you will suffer no disadvantages through the use of Scrum.

In simple environments⁵ you should not use Scrum. If everything is already clear, you do not need an iterative and incremental approach. Instead, a predictive approach such as waterfall or V-model can be put to good use. You just have to make sure that you do not overly formalize them, for example by demanding unnecessary documentation.

In chaotic environments, it does not matter which approach you use. You have no chance for success. To put it simply: If you do not know what you want or how to achieve it, no model in the world can ensure your success. Scrum can help here only slightly by introducing clear roles, ensuring delivery of something usable every iteration, and the continuous inspection of achievements. This forces you to at least think about what you actually want. Once you have done that, there is a chance that you will reduce uncertainty, come out of the chaos realm, and move into the complex field. Another method that can help you here is “Lean Startup”

⁵ For example, this could be the integration of a new, already integration-tested machine or the rollout of standard software after the pilot testing is complete.

(cf. Ries 2011). Once you have reduced your complexity, Scrum might be your method of choice again.

Once you know the problems in your organization and the complexity of the project in question, you should be able to judge whether Scrum is right for you. Be very careful not to underestimate the complexity of your situation. I often encounter situations where complexity is assessed incorrectly to avoid applying agile methods even though they would be more appropriate. This produces inferior value and should be avoided.

4.2 Stakeholder Management

You will not get far without your stakeholders. At the end of the day, the customer pays your salary, management approves your projects, and your colleagues from other departments support your work or benefit from your results. If you do not manage to convince all stakeholders that your endeavor is meaningful for them, the risk of failure increases significantly. It is therefore essential that you constantly involve all stakeholders, provide them with information, and consider their needs. Make objectives, policies and benefits for each individual transparent. Use every opportunity to talk about your project. For that, all channels of information are valuable: letters, e-mails, phone calls, meetings, and personal conversations all have their specific advantages and disadvantages. If all the communication is starting to really freak you out, you probably have achieved around 30–40 % of the necessary communication level. You will find more about communication in the following chapters.

One thing you can do right now: Perform a stakeholder analysis. Take a whiteboard or flip chart and begin your deliberations by writing own name in the center. Then place all the groups and individuals you are directly or indirectly dealing with in the context of your Scrum introduction around you. The closer the stakeholders are to your own name, the more you have to do with them. Draw lines between all individuals and groups who communicate with each other regarding your project. The thicker the line, the more intense the interaction. Then, make it clear who is positive, neutral, or hostile towards the Scrum introduction. One good way to do that is by color (green, blue, red). Then note down (next to the lines) the personal interest motivators or de-motivators of each individual towards the others. These are usually both goals of the company and personal goals of the people involved. It can help to ponder the question of what needs, concerns, or fears direct the actions of these people. Finally, you should realize how important each person is for your success. For example you can visualize this by noting a percentage behind each name, where 100 % means that a project's success is not possible without this person. Aim your communication efforts to keep your supporters happy and to transform enemies into supporters. Judge for yourself how much, and through which channels you have to communicate with whom in order to achieve these goals.

4.3 Different Situations and Product Types

This chapter I deliberately named “product types” and not “project types.” With Scrum you develop products, not projects. From a Scrum point of view, every Sprint is a complete project. This definition is at odds with the definition common in most traditional enterprises. At the opposite end of the scale, it can happen in very advanced companies, that no more projects are conducted. Instead, only (agile) product development occurs, the continuous development of products with a focus on business value, without having to set up individual projects anymore. The benefit of Scrum is not that it supports different types of projects particularly well or badly, but that it supports the development of different types of products. The following Table 4.1⁶ provides a brief overview of different situations and shows you whether Scrum is generally suitable or not. Do not see it as a hard prescription though—use your own sound judgment to figure out what works best for you.

Software development is a clear case since Scrum was tailored to fit it. So if you want to develop software, you can opt for Scrum with a clear conscience.

In hardware development, you often have conditions that make adherence to fixed iterations and providing a potentially releasable Product Increment difficult. Your suppliers sometimes have delivery times that by far exceed 4 weeks. Also supplying a Product Increment usually means the completion of the corresponding hardware. This can result in considerable costs in time and material (you have to bolt the thing together, after all). Depending on what kind of product you are developing, you will have to choose an approach that might differ from Scrum. For example, you might only be able to supply a finished CAD model at the end of a Sprint—but this is not a customer usable Product Increment. This means that you did not meet a fundamental requirement of Scrum and thus cannot name your approach Scrum. Otherwise you run the risk of transmitting the wrong signals to the detriment of the future use of the Scrum approach. You are most successful with agile methods in hardware development if you choose development suppliers who can rapidly deliver prototypes. Remember: You are primarily trying to reduce uncertainty and gather feedback. Once you succeeded there you still can order parts that are suitable for production.

In a series production, the desired result as well as the process and all framing conditions are generally known. Empirical models are therefore totally unsuitable. Stay with the principles of Lean instead and use a predictive approach (e.g. waterfall, V-model, phase models, etc.).

If your situation involves product maintenance—resolving errors or adjusting certain details with known technology—you can use Scrum. However, you are

⁶This list is not conclusive. It is quite possible that there are other processes and methods that work well in certain contexts. However, my personal experience taught me that the approaches listed work very well in the situations mentioned.

Table 4.1 Suitability of Scrum

Situation	Classification	Recommendation
Software development	Complex	Scrum
Hardware development	Complex	Scrum combined with other agile methods
Serial production	Simple	Predictive approach
Unit production	Complicated	Adjusted Kanban
Product maintenance	Complicated	Adjusted Kanban
Research	Chaotic	Lean Startup and Scrum

probably better off with a customized Kanban⁷ approach. Simplified, Kanban means that you are visualizing your work with a “Kanban Board” (put up on the wall), you reflect your process on the board, you define a “work-in-process limit” for each process step, and you measure the time each item needs to move through every process step. “Adjusted” means in this context that you are well advised to adopt some elements also common in Scrum—especially the Daily Scrum, retrospectives, roles, and of course the values. Make sure that a self-organizing team owns the board. The results will reward you well.

In research environments, you often know next to nothing. Presumably this is the reason why it’s called research and not development. Lean Startup works quite well here and can be combined with Scrum. Scrum isn’t easy to implement here since the basic requirement to deliver a high-quality, done increment every Sprint is sometimes not possible to fulfill in research environments. What helps you though is the regular inspection of where you are and the appropriate adaptation to get back on track. This prevents your research team from hiding for months and not delivering in the end because they are bogged down in detail. Even under optimal conditions, the risk to achieve nothing at all is quite high in research work. Do not further increase that risk by using poor leadership or the wrong processes. Living the agile values and the Scrum culture will help you in chaotic situations. Using the principle of inspection and adaptation, you can also improve your processes continuously and thus reduce risk. Lean Startup concepts can help you reduce uncertainty even more. In conjunction with Scrum you have an excellent starting point for research scenarios.

References

- DeMarco, T., & Lister, T. (2003). *Waltzing with bears: Managing risk on software projects* (Paperback ed.). New York: Dorset House.
- Kniberg, H. (2010). *Kanban and Scrum – making the most of both*. Paperback. Lulu.com
- Ries, E. (2011). *The lean startup*. New York: Crown Business.

⁷One place where you can learn more about Kanban is Henrik Kniberg’s book “Kanban and Scrum—making the most of both”. The book is free to download at <http://www.infoq.com/minibooks/kanban-scrum-minibook> (Kniberg 2010).

-
- Schwaber, K., & Beedle, M. (2002). *Agile software development with scrum* (Pearson Internationalth ed.). Upper Saddle River: Prentice Hall.
- Stacey, R. D. (1996). *Strategic management & organisational dynamics: The challenge of complexity* (2nd ed.). London: Financial Times Management.
- Yourdon, E. (2003). *Death March* (2nd ed.). Upper Saddle River: Prentice Hall.



Depending on the target state (see Chap. 2) and approach (see Chap. 3), you need to plan and perform a different course of action. The basic steps are the same everywhere though (Fig. 5.1). There is just one exception from this rule: If you implement Scrum PRN through a submarine approach, you do not need the steps outlined below—at least not if this PRN submarine will be sufficient for you into the future as well. In the following section, I refer to the book “Leading Change” by John Kotter (refer to the appendix if you want to know why I did not refer to other authors). In my daily work, I use the procedure described by him to introduce Scrum into organizations and to solve the systemic problems uncovered. It is these experiences I would like to share with you. The phases described by John Kotter (2012) are:

1. Establish a sense of urgency
2. Create a guiding coalition
3. Develop a vision and strategy
4. Communicate the change vision
5. Empower employees for broad-based action
6. Generate short-term wins
7. Consolidate gains and produce more change
8. Anchor new approaches in the culture

A recurring point of criticism with Kotter’s model is that we are looking at “phases” here that must be worked on one after the other. This has little to do with agility. While I generally agree to this criticism, I am not so critical about the wording myself. How you apply these eight “phases” in your change endeavor is up to you. My personal experience with embedding them into Scrum and working on them in Sprints is excellent.

You must go through each of these phases multiple times to have sustainable success. Usually you have to go through these phases linearly over and over again,

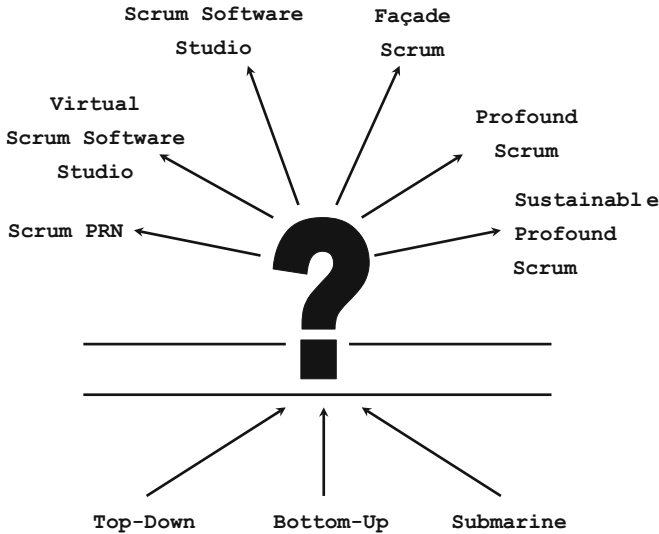


Fig. 5.1 Finding the right way

i.e. you cannot complete the eighth phase prior to the first one (cf. Kotter 2012, p. 85).¹ However, usually you will be active in several phases at the same time. Depending on the desired Scrum shape and starting point, your vision will differ considerably. Similarly, you will also tackle the transition stages differently and over different time spans. Be aware though, that the change of an organization can easily take 3–10 years. It is not something you can do overnight. For example, if your goal is to introduce Sustainable Profound Scrum in order to make all major problems in the organization transparent, and to increase productivity by a factor of five, then this is a realistic, albeit lofty goal. It will take 10 years rather than three in a large company group environment to reach it, while in a very small business you might be successful after just 2 years.²

5.1 Introducing Scrum with Scrum

No matter what shape you seek or which way (except submarines) you choose to reach it: Use Scrum for yourself! That means in detail:

¹ Please note that “linearly” applies to completing the phases, not to starting them. Usually, you run these phases concurrently, but obviously you cannot produce quick successes without a guiding coalition. . . .

² “A decade is a short period of time in which to expect to institutionalize cultural change within a large organization”—Denison (1990).

1. Appoint the person who has the greatest interest in the Scrum introduction, namely the Product Owner.
2. Find someone with Scrum experience to fill the Scrum Master role.
3. Do not mix these roles!
4. Assemble a team of three to nine people, which helps you to introduce Scrum.
5. Maintain and organize a Product Backlog in which all work packages to achieve the objectives are listed.
6. Work in iterations.
7. Deliver something “done” at the end of every iteration. In the context of a Scrum introduction, this may include for example documents, processes, approvals, or tools. A working integration environment could be such a Product Backlog Item.
8. Present your results at the end of each iteration and get feedback from your stakeholders at that point. Show your accomplishments.
9. Inspect your approach in a retrospective at the end of each iteration and improve your processes.
10. Meet with your team every day and plan the next steps to reach the Sprint goal.

By doing this, you very quickly demonstrate the feasibility and flexibility of your endeavor. You also will learn more about Scrum, because its essence can not be conveyed by books nor training alone. Another advantage is that it will make you focus and not allow yourself to be diverted from the goal. Scrum will not only help you to achieve success fast, but also to make it transparent.

5.2 Multi-Change Initiatives/Change Programs

It is very likely that you find yourself in an environment in which it is not about the implementation of a single change, but about a variety of smaller changes (cf. Kotter 2012, p. 26) (Fig. 5.2). You are in good company then, as it is like this in most cases. Each of these changes has to go through all the phases described here—otherwise, each of these small steps could cause you to stumble or even fall. In general there is an overarching larger change (e.g., the Scrum introduction). To achieve this, however, many smaller changes have to be implemented (e.g., continuous integration, test-driven development, significant transparency, etc.). The exact specification of the phases described in the following chapters can be quite different from case to case. For example, if the trigger for the introduction of Scrum is the current lack of transparency, you do not have to create a new urgency for the “sub-goal” of transparency. You can build on the preliminary work already done. This is different with sub-goals like “test-driven development”: While it supports transparency, you cannot see this at first glance. In addition, most developers will have to fundamentally rethink and learn new skills. Therefore you should take the time to carefully work out the urgency for the sub-goal of test-driven development in a way that supports all levels of your company. This will also save you a lot of employee turnover.

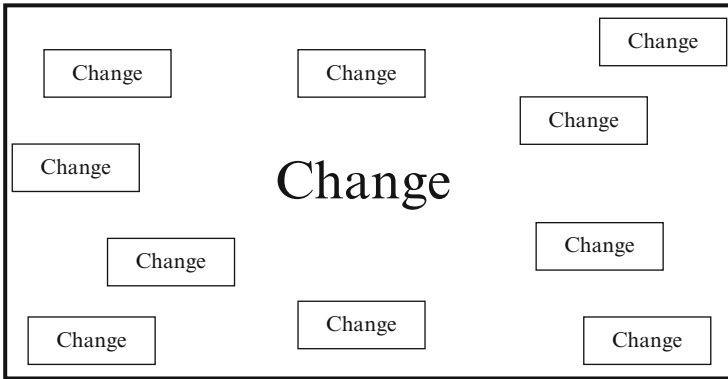


Fig. 5.2 Many small changes on the way to great change

In summary: Every change must go through every phase. Each phase can be finished at different speeds. Check in every single case, if you can build on existing results. Do not skip any phase or leave it unfinished, because such failure will catch up with you again sooner or later.

A medium-sized company started with the introduction of Scrum. To achieve this, many individual changes were necessary—some tiny, some of considerable size. At the beginning, the company did a very good job. A guiding coalition was formed, a sense of urgency was created, and successes generated. Unfortunately, this happened only at the level of the “larger” goal of introducing Scrum. For many smaller changes the employees were passed over, there was no one who felt responsible, and management was of the opinion that proclaiming a change would be sufficient to get the employees to change their thinking and acting. That mistake significantly retaliated two years later: Some of the best employees left the company and the remaining active development processes could only be described at best as chaotic. The benefits of Scrum could not be leveraged. Much time and money was wasted.

References

- Denison, D. R. (1990). *Corporate culture and organizational effectiveness*. New York: Wiley.
 Kotter, J. (2012). *Leading change* (New ed.). Boston: Harvard Business Review.

Part III

The Practical Application of Kotter's Principles



Creating a Sense of Urgency

6

If a company operates successfully in the market, it builds up a pool of processes, structures, and regulations over the years. These are usually very useful and helpful to the organization at the time of their introduction (cf. e.g. Kotter 2012, p. 149). However, often the utility diminishes with time because the world—especially in today’s globalized and fast-paced environment—changes, but the processes remain rigid. If you ask your employees about the perceived benefits of the existing processes, structures, and rules, you will probably find that they are being experienced as a burden rather than an aid. You can prove this—especially for non-production processes—easily with value stream analyses.¹ On the other hand, however, these structures provide footing and safety. Every employee knows exactly where her place is in the overall structure, her power and influence. If you want to implement far-reaching changes, you directly attack people’s safety needs. Maslow clearly demonstrated in his “hierarchy of needs” that “safety” is almost as important for every individual as “physiological needs” and thus a foundational need far from the higher levels of “esteem” and “self-actualization”. This literally means you are shaking your employees at their very foundation. Accordingly, you will encounter resistance, which can stop your change process, even before you have really started. Therefore, you must transport a sense of urgency that is recognized by everybody. It must illustrate that your change endeavor is not only justified but also necessary for survival. Brief: Make the crisis transparent, on which your desire for change is based on (cf. Kotter 2012, pp. 46). You would not undertake a major change effort, if you did not have good reasons. Share them. With smaller changes, it may be that it is not about survival—they are nevertheless usually about long-term profitability and thus competitiveness. There

¹ In a value stream mapping, you first write down all process steps and add the process inputs and outputs. Then you split them into value creating, supporting, and wasteful process steps and measure their throughput times. The goal is to minimize the supporting processes and to eliminate the wasteful ones.

is a reason that you want to change something. Make this reason visible to everyone.

Once I encountered a company that wanted to improve their situation: The developed software had no recognizable quality, it sometimes did not even start when the customers tried to run it. The development environment was not suitable for the size of the development organization (e.g. code integration was not possible) and the staff were not qualified for their tasks (both in their domain know-how and technical skills). While they did want to solve the problems, the company initially saw no need to change. This was due to a general “we were successful in the past, so why should we change” mentality, which was supported by a culture in which no mistakes were admitted and bad news was never openly communicated to the top. The management presentations always showed that the software development department achieved good results. Everything was reported “green”, not mentioning any problems. Consequently, no issues were highlighted in these board meetings. It was only when I started an impediment in-depth analysis together with one of the Product Owners and several Scrum Masters to gather concrete data on how much time was wasted pointlessly (it was more than 80 % at that time), and openly showed that we could not make any statement as to whether the functionality would work at all for the customers, when everyone began to change their behaviors. We had created a sense of urgency and could now work on the issues slowing down the organization.

Do not search for abstract problems. Reality is usually full of good, suitable, and serious nuisances. Ask as many people as possible to share their opinions about what constitutes a risk or is already going wrong. Make sure that all information is treated confidentially and cannot fall back negatively on individuals. Avoid euphemisms. Allow yourself to change into the bird’s eye view and to expand your current mindset. Be brutally honest with yourself and everybody involved. Make the existing crisis clearly visible. Be sure to select a simple, specific, and figurative language, avoid complicated or technical jargon (cf. Kotter 2012, pp. 92). Compare these two statements:

- “Due to the low quality of our software, we have an innovation rate of 30 %. The rest of the time we spend on bug fixing.”
- “Our software is a heap of garbage. We create so many errors that we need to spend 3 million Euros in repairs every month.”

Judge for yourself: Which of these statements is more likely to draw the attention of the decision makers?

Sometimes it is not easy to present concrete numbers. Unfortunately, this is usually necessary to create a sense of urgency. No matter if your case is about market share, time, money, or something else: It always helps to provide numbers. This is important in the context of a Scrum introduction as well. Often you encounter a process that is called “Scrum” by the organization, but is not in reality. You can still use the mechanisms and metrics of Scrum to your advantage. Sometimes, the velocity is enough to demonstrate the productivity of a team.

In the following you find two tools that can be used with existing Scrum teams (e.g. pilot teams or during a Scrum transition that already is on its way).

6.1 The Impediment-In-Depth-Analysis

This approach emerged when in a challenging project a Product Owner came to me and wanted to know why, in his perception, his team was so unproductive. He had not felt like this with other teams and was curious about the reasons. He wanted exact numbers of how much time was spent on which activity. But neither of us wanted to introduce time recording or to further burden the team with intricate methods. My Product Owner suggested to estimate the time based on specific categories. This developed into the Impediment²-In-Depth-Analysis:

1. Assemble your team for a retrospective.
2. Provide a few main categories (e.g. bug fixing, meetings, assistance to colleagues in other teams, manual testing, etc.) and make it clear that these are only suggestions, which can be changed and supplemented by the team at any time. It makes sense to prepare these categories based on the observations of the Scrum Master or yourself.
3. Make it clear to the participants that this is not a regular retrospective, but a deep analysis of the impediments. This means that those problems that everyone has already “accepted” and are considered “institutionalized”, have to be put back onto the table again. One example might be the slow response time of servers. You need to make it clear that you are interested in digging up everything in depth in terms of the team members’ thoughts and feelings for these and other issues.
4. Give them 5 min to write down impediments that caused costs in the past Sprint. The amount of wasted time is not yet estimated. Conversations are not allowed here. Everybody writes his own ideas by himself.
5. Gather the team to explain and cluster the results.
6. Create, new categories, if necessary.
7. Conduct a brainstorming session in the team to reveal additional impediments.
8. Ask the team to estimate the time required for every impediment. I usually use Planning Poker^{®3} for this step to make sure quieter colleagues are also heard.
9. Sum up all estimates at the specified category levels.
10. The sum of all impediments and work performed should total close to 100 % of the available team time of the last Sprint. If that number is above 110 %, verify if some impediments were estimated twice (e.g. one impediment “release candidate phase” and another one stating “manual testing”). If the sum totals less than 85 %, you probably missed something important. Ask your team to look for it again.

²“Impediment” is a Scrum term. It means “problem, that inhibits project progress.”

³Planning Poker is a method to estimate work. Usually it is used for relative estimates (e.g. Story Points), but it is possible to use the method for absolute numbers (e.g. hours) as well. This method has the advantage that no one dominating person can influence the estimates and cause a bias. In addition, the method nicely facilitates discussion. You can learn more about this method in the appendix. Planning Poker[®] is a registered trademark of Mountain Goat Software, LLC.

11. Grab the main category, which contributes most to the time wasted. Here you can define improvements with your team. At the same time, you can now prove with specific figures that there is a problem.

This approach works especially well if your team is already well rehearsed⁴ and trusts you. If this trust is lacking there is a risk that the developers do not dare to highlight problems openly. Be especially careful that all things the team has already become accustomed to are brought into the open. These often constitute far greater problems than even the most striking singular event of an individual Sprint. The method gains its credibility by cross-checking the results with the work really done. This requires that your Sprint Backlog is well maintained. It is enough to refer to the estimates of the tasks (you do not have to track the real spent time) inaccuracies usually equal out over the sum of the tasks again.

6.2 Velocity Extrapolation

Velocity extrapolation is something that every Product Owner should already do routinely. However, in organizations that have implemented Façade Scrum, this important metric is often missing. Velocity extrapolation is nothing but an honest release planning (Fig. 6.1).

1. The Product Owner decides when a release will be delivered and what should be included based on the current velocity of the team.
2. The Product Owner asks the Development Team to estimate all relevant requirements (Product Backlog Items).
3. The costs (time) for the release can now be precisely compared with the expected benefits (features).

Just this small increase in transparency often leads to a significant surprise for the people involved. The project plan had said something entirely different, after all! Theory and practice do not fit together anymore and lead to considerations whether the costs of product development are still justified under these circumstances.

In a company I coached, the release plans were usually created with little empirical data and a lot of faith. While the anger about features that had not been delivered as expected was always great at the end of each release, this did not lead to a change in behavior. Only when we got all the people responsible for planning around a table and confronted them with the reality of their teams' speeds, a rethinking process started. Still, it took another three months until this group of people switched into "crisis mode" and informed top management about the fact that the planned targets were not going to be achieved. Only

⁴If your team is not yet well rehearsed working together, usually team-internal issues are so important that your team will not uncover organizational ones. In addition, a lack of familiarity with the organization or the project can make it difficult to uncover deeper problems.

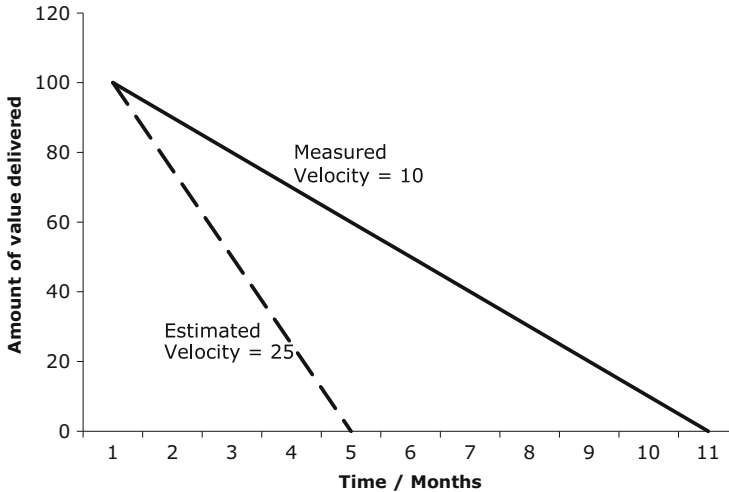


Fig. 6.1 Realistic time extrapolation for complete delivery of all features

from this point in time did a sense of urgency exist. The company could now openly consider what measures it could take to avert the crisis and how such situations could be avoided in the future.

6.3 General Advice for Creating a Sense of Urgency

No matter whether you already adopted a shape of Scrum or not, you need numbers that transport the urgency. Let a crisis evolve (cf. Kotter 2012, p. 46) in the minds of those involved⁵ or make an ignored crisis transparent. Adhere to Kotter’s advice and stay credible. This may also mean that privileges of certain groups of people need to be abolished, if they do not fit to the proper crisis behavior. If there is a crisis, everybody has to act accordingly! In an existence-threatening crisis, for example, it is not appropriate to fly with the corporate jet; an economy class train ticket might be better. To break out of crises, ordinary targets do not suffice—if this were the case, the objectives of the past would have led the company into a better present. Set the goals in a way that they cannot be achieved with normal work efforts, but that they require extraordinary commitment and creative solutions. Also make sure that the whole company no longer uses any personal objectives, but just objectives and targets that are focused on the overall performance of the organization. This way, everybody is sitting in the same boat.

Make sure that the employees directly deal with unhappy customers, shareholders and suppliers, so that corresponding issues cannot be ignored

⁵ This does not mean that you should create an artificial crisis, but that you simply agree to let an existing crisis come to the surface and into people’s consciousness.

(cf. Kotter 2012, pp. 44–46). Make the urgency visible across all available communication channels as often as possible. Do not forget to highlight the opportunities that are offered by the change. Scrum is an excellent tool to transport a sense of urgency: You can relate to the crisis in each and every retrospective and planning meeting. In addition, impediments revealed by the team can quickly be made transparent. A decrease in velocities⁶ may contribute to a sense of urgency as well.

Depending on what exactly you want to achieve, the severity of your urgency varies of course. If your Scrum Team needs to improve the quality of code, it does not make sense to abolish benefits like paid travel for holidays. Although desirable, it does not help if the marketing department exchanges their individual department objectives for general business goals. Important here is merely that the team recognizes the urgency and management (or the Product Owner) provides the necessary time to keep it in focus.

The situation is different if the entire company shall be switched to the Scrum Culture: In that case senior management, the personnel department, and all other departments have to pull together. They will only do this if they realize and share the urgency as well as face a joint crisis. Make all the people involved or affected aware of the reasons why the change to Scrum is so important for both them and the company.

You also should ensure commensurability. When in doubt, however, take all the points mentioned in this chapter into account. This is key to avoiding inadequate preparation and thereby missing the much needed target of change for your company. If you fail to produce a proper sense of urgency among those affected, your change initiative will fail.

6.4 Things You Should Remember

To create a sense of urgency in connection with a Scrum introduction, you should proceed as follows:

1. Determine who does want (or does not want) to introduce Scrum.
2. Analyze the company's situation and that of your competitors.
3. Make your company situation transparent and show everybody involved the urgency to change.
4. Analyze what is changing for the company through the introduction of Scrum.
5. Analyze what is changing for every individual through the introduction of Scrum.
6. Communicate the opportunities and risks that lie in the change to all parties. Point out to each individual what an insistence on the old processes means for her. Also show everyone the opportunities that the change offers for him or her.

⁶“Velocity” refers to the speed at which a Scrum team produces customer value. The time worked is not measured. Instead, the relative size of the customer relevant work that is completed is what counts. Unfinished features, bug fixes and support functions usually do not count towards velocity.

-
7. Live the consequences of the urgency as a role model. If there is a crisis, you have to behave accordingly.
 8. Make sure that the top management of the organization understands the urgency and exemplifies it in their behavior. All communication measures are worthless if management negates your statements through a contradictory behavior.
-

Reference

Kotter, J. (2012). *Leading change* (New ed.). Boston: Harvard Business Review.



To implement the changes in the organization, you need a guiding coalition. One person alone is usually overburdened with driving the change (cf. Kotter 2012, pp. 52). So you need a team. It must consist of the right people, be properly organized, and be constantly present for those affected by the change. Our focus is still on the Scrum introduction: Since you now know why you want to introduce Scrum (or what your urgency is), in this section you will learn who has to drive the changes in your organization.

7.1 Composition

Any change has to be driven from the “top”, otherwise it will fade away. The “top” level may differ from case to case though. For an enterprise-wide rollout of Scrum, the CEO is needed in the guiding coalition, while the introduction of test-driven development probably just requires the head of development. Also, the composition of the coalition depends on the introduction approach selected.

In the case of a submarine you have no other choice than to work with the people you have. Since in this situation you can only implement changes at Development Team level, the Development Team is most often the guiding coalition. That is sufficient. If your goals change, however, you may slip into a bottom-up implementation and have to adapt your guiding coalition.

In a bottom-up approach to introduce Scrum PRN, all managers should be part of the guiding coalition that carry responsibility in the development of your products. Usually, this includes development managers, product managers, and project managers, if necessary. Bring together those people who will make the decision whether to use Scrum for each project or not. You will gradually advance into the upper levels of management. In the end, you might arrive at a top-down scenario.

Here, executive management is involved as well and you can try to establish a Scrum Studio or even Profound Scrum. In any case, organization-wide changes become possible. By the time you want to implement such far-reaching changes,

you need the active support of management—therefore, these colleagues are required as members of your guiding coalition.

Let us get more specific. For the introduction of Sustainable Profound Scrum you need the CEO—at least as supporter, better still, as a member of your guiding coalition. All colleagues who are necessary because of their hierarchical position have to be added to the mix—usually the division or department heads. You also need a Scrum expert, who should be complemented by an existing process expert. It does not hurt if this expert is a Scrum-skeptic. On the contrary, his fears and objections should be considered a pulse monitor for the organization. These people should bring credibility (cf. Kotter 2012, p. 58), but sometimes they are too far away from action. A leader who has the confidence of the developers directly affected, and is perceived by them as being credible, should also be part of the coalition. Lastly, you need an organizational change expert. These people need to push the process of change—it will fail without them. Sometimes a single person can fill multiple roles. This depends very much on the individual though. Too large a group tends to spend more time discussing than acting. Therefore, you should not exceed a team size of nine people. In very large organizations, it may become necessary to directly include more people in the guiding coalition (cf. *ibid.*, p. 61). Then try to reduce meetings with the full group to a minimum and switch working and decision-making over to smaller groups.

At one of my clients, I encountered an already formed guiding coalition: the heads of all departments of the area as well as important key players had been organized to implement Scrum. This group of eight individuals also had a designated Scrum expert and someone with experience in change processes in their midst. With this strong base, I only had to transport some methodological refinements and raise the level of Scrum knowledge of the group. The starting point could not have been better for the desired changes.

Should you not be able to fill all of the roles above immediately, you should first try to develop your internal staff. It is always better with such important topics to rely on your own personnel than to hire them in from third party suppliers. If you do not succeed in finding suitable candidates at all or quick enough, you have no choice but to take a look at external service providers. Especially when it comes to the change experts, you should very carefully select your partner—too much time and money depend on this position so you cannot play it by chance.

Now that you know who should work in your guiding coalition, we also have to consider who should under no circumstance be part of the coalition: You do not need saboteurs and egomaniacs.¹ Saboteurs are people who, through their actions, prevent collaboration of the group as a whole. This can show through pointed remarks, repression (by more senior colleagues), or other anxiety-promoting actions. Egomaniacs are those characters that are so self-obsessed, that no one else in their opinion can add value. Other team members' ideas are flogged to death, ignored, or talked down. Talking time of other people is begrudged. Responsibility for important tasks is not delegated. When you encounter such a “show stopper”,

¹ Kotter (2012, pp. 60–61) calls these “egos that fill up a room” and “snakes”.

you have no other choice but to get rid of them. You can do this by persuasion or termination—allowing the evildoers to stay in the organization is not feasible though (ibid., p. 63). If you have clearly identified the urgency of the change, but the person is further blocking the endeavor without giving any reasons, then he will continue to spread his poison throughout the organization even if he is expelled from the guiding coalition. This can only be avoided by removing this person from the company.

In a small enterprise the CEO was omnipresent. He actively designed all areas of the business and was highly resistant to advice. This behavior had worked well for decades, because he was an extremely capable man and full of energy. Eventually, however, the company reached a size that necessitated the delegation of power and responsibility. Therefore it was necessary for him to trust his subordinate managers and be confident that they would fulfill their respective responsibilities. Unfortunately, the CEO was not capable of doing that for all of his subordinates. Instead, he never got tired of explaining in detail to his managers why they were incapable and too inexperienced to perform their duties according to his expectations. He tried to fulfill their responsibilities as well. As a result, his management team's motivation was ripped to shreds, employee turnover increased, and the results of the company as a whole severely suffered from the weakness of this CEO. It would have made sense to bring in a second director, who could have compensated for the weaknesses of his counterpart and could have enhanced the exceptionally positive traits of the CEO. Unfortunately, the attempt failed and the situation remained unchanged until the company failed economically.

7.2 Organization

Organize your guiding coalition with Scrum.² This way, you will learn how Scrum works, how it feels, and what outcomes it produces. Moreover, it gives you the ability to react quickly to changes and regularly deliver results in iterations. Generally, you should apply the elements of Scrum for a Scrum introduction like this:

1. Sprint

Each iteration is called Sprint and cannot exceed 1 month. Assess the availability of your coalition and the urgency of the project. The better the availability of your team and the more urgent the change, the shorter (1 week minimum) the Sprints need to be. In practice, the change teams often opt for 4-week Sprints, although 2 or 3 weeks would be more effective. Settle all Sprint dates 6–12 months in advance, so that the calendars of the guiding coalition do not become a constraint.

²The explanations in this chapter are very general. The concepts are more vivid in the case study of this book.

2. **Sprint Planning**

Perform planning for every Sprint. First, revisit your release plan, your vision, and your goals. Then decide on the next steps, based on the results of the last Sprint and your Product Backlog. Make sure the goals are achievable: If it is foreseeable that a goal cannot be reached, break it down into several smaller goals. Do not forget that the team determines the amount of work to take on for a Sprint—this is true for the guiding coalition as well.

3. **Daily Scrum**

The Daily Scrum should be held on a daily basis. The purpose behind this is to evaluate and re-plan the line of action on a daily basis. So it is not about reporting the status (even though this may be part of the Daily Scrum), but about minimizing risk. Strictly speaking, you are not doing Scrum if the Daily Scrum is not conducted every day. However, it may be necessary in the context of your specific situation to meet at less frequent intervals (e.g. twice a week). This usually suggests that the Scrum introduction is not the highest priority for the guiding coalition. Hence, people do not work every day on the tasks and do not have anything to share from day to day. Think about the signal you are sending to all employees before you decide to abandon Scrum rules.

4. **Sprint Review**

Open up your review for everybody interested. Invite everyone affected by the change; when in doubt it is better to invite all employees than to forget an important person. The goal of the review is to gather opinions from your stakeholders and to jointly plan the next steps. It is a collaborative working session. Begin by briefly highlighting the urgency and vision. Then continue by communicating the planned goals of the last Sprint and compare these with the actual achievements. The results should be tangible. If you produced documents, print them. If you provided a work environment, visit it physically. Allow the audience to try everything out. Such a review might even be fun!

5. **Sprint Retrospective**

Only the guiding coalition may be present in this meeting. Analyze your cooperation during the last Sprint and develop measures to improve it. Communicate the results to all interested parties. Show clearly that you are willing to make mistakes and learn from them. A simple way to communicate the results is through colored task cards on the backlog. For example, “normal” backlog items could be yellow, while measures from the retrospective could be red.

6. **Product Backlog**

The Product Backlog is owned by the Product Owner. All requirements that have to be fulfilled by the team (which is the guiding coalition) are entered into it. The form is secondary. However, it is recommended to use the same medium and format that you expect from your future product Development Teams. Make the Product Backlog transparent at all times for anyone interested. This works best when it is visibly posted up somewhere.

7. **Sprint Backlog**

The Sprint Backlog consists of all the requirements that have been selected for a Sprint, plus the tasks that are necessary to fulfill them. Use it the same way

you expect from your product Development Teams. Make it transparent. Every employee should and must know what the guiding coalition is currently working on. Break the tasks down into small chunks. Ideally, every team member can complete one task per day.

8. **Product Increment**

It is difficult for the guiding coalition to create a “potentially shippable” Product Increment. But it is possible. Arrange your requirements for dependencies and business value. Make sure to address short- and long-term tasks each Sprint. This allows you to not only work on long-running tasks that do not produce anything tangible within one Sprint. Produce results that demonstrate a clear benefit for the change activity, for the Scrum introduction, and thus for everybody involved.

9. **Product Owner**

The person with the greatest interest in changing the organization should be Product Owner. In Profound Scrum or a Scrum Studio, that is the managing director. Scrum PRN or a Virtual Studio usually require the involvement of the development manager. It is important that a clear signal is set for the guiding coalition, what strategy and vision are, and in which direction they are headed. These are tasks for the executive management. The Product Owner must be responsible for the Product Backlog—that does not mean, however, that he has to do all the backlog related work himself. Just as with a normal Scrum Team he can delegate tasks such as the formulation or preparation of backlog items. The selection of the most important elements, that is the arrangement of the Product Backlog, cannot be delegated though.

10. **Scrum Master**

The Scrum Master of the guiding coalition will have to explain how Scrum works to many colleagues in the course of a Scrum introduction. He should therefore have deep process knowledge. If in doubt he should get professional assistance until he has built up enough knowledge and experience himself. His job is to make sure that the Scrum rules are followed within the guiding coalition. He also is available to all colleagues in the company to answer questions about Scrum. Naturally, it is also his duty to make grievances transparent and to ensure their resolution. As part of the guiding coalition this should not pose a problem, as all required responsibility and authority is already assembled.

11. **Development Team**

You develop your business. This means you are a developer. This applies to all members of the guiding coalition except Product Owner and Scrum Master. You are doing the actual work and transforming the requirements from the Product Backlog into a Product Increment. By the way: in a normal Scrum Team the term “developer” does not only describe programmers, but all the people who contribute to the creation of the overall results—including tester, documenters, architects, etc.

Once you have filled all roles, you should find a name for your team. In practice, I have encountered names such as “Scrum Steering Committee”, “Agile Capture

Team”, and “Agile team.” “Enterprise Transition Community” and “Enterprise Transition Team” are also widely used. Whichever name you choose, retain it and make sure that everyone in your organization connects it with agile successes. Conduct trainings (better externally than internally) for the guiding coalition. In addition, start building trust (cf. Kotter 2012, pp. 63). All members of your team must be able to rely on each other and know this to be the case. The level of trust required usually exceeds the level of trust already present in a business environment (ibid.). Do not try to save money here—it is the wrong place. Organize carefully planned offsite events, have the group work together and celebrate together. Build on the urgency (see last chapter) to work out a vision for your guiding coalition and goals that must be achieved. These goals have to appeal to both hearts and minds of those involved. To achieve this, it is necessary that you understand the motivations of your colleagues: What motivates them? What drives them? Go for it!

7.3 Tasks

The guiding coalition is responsible for leading and managing the change endeavor. It is their job to develop and communicate a common vision. They inspire other colleagues to follow them by living their vision.

As a member of the guiding coalition, you should know exactly why you are seeking the change, what the resulting state looks like, and how to get there. Also, you should know the motivation of all your employees, and why they follow you on the change journey or not. In the context of a Scrum introduction, every member of the guiding coalition must know how Scrum works. They do not need to be experts (one expert in the team is enough)—but the basic concepts must be known. In general, the guiding coalition will not just enforce rules or impose draconian penalties, even though they possess the necessary power. Such measures are not needed if urgency and vision are clearly defined and well communicated. Instead, the guiding coalition lives by what they preach and does what is necessary. Think of your children: You probably have already noticed that your kids imitate you and simply ignore verbally pronounced behavioral instructions. If you eat with your fingers, the children will do that too. If you use foul language, you will hear the same words from your children soon. But if you always help your partner with the dishes after eating, showing great joy in doing so, then the small ones (at least until puberty) will also want to help. It works the same way in your company: If you exemplify the situations and values you communicated, your employees will follow you. If you preach water and drink wine, you are on your own.

It is also one of the tasks of the guiding coalition to always make the upcoming steps and the progress accomplished transparent. In other words: Create a Product Backlog for the changes, hang it on a wall in order to make it easily visible to all interested parties (e.g. in the cafeteria), and run a Review at the end of each Sprint, inviting all concerned parties. Present the results and turn “parties concerned” to “parties involved” by collecting and valuing their opinions. Also show what your team has learned and communicate the results of your Retrospectives. This clearly shows that a culture of learning is not only allowed but also desired. Focus the

coalition to such a high degree that it is clearly evident for all parties that the change has the highest priority in the company and will not simply be handled alongside daily business. Prevent the grapevine effectively by immediately facing rumors. Introduce, if necessary, a rumor wall.³

7.4 Things You Should Remember

To get your guiding coalition into place, you should proceed as follows:

1. Find the right people (cf. Kotter 2012, pp. 59–60):
 - 1.1. Colleagues with hierarchical power (general manager, area managers and/or department heads).
 - 1.2. An expert for the current processes.
 - 1.3. A Scrum veteran.
 - 1.4. All other people who could help or hinder your Scrum introduction.
 - 1.5. All these people must have a high credibility within the workforce.
2. Limit your team to a maximum of nine people or divide it into several sub-teams.
3. Get outside help if you need it.
4. Assign the role of Product Owner to the hierarchically highest colleague with the greatest interest in the Scrum introduction.
5. Assign the role of the Scrum Master to the best internal Scrum Master.
6. Give the team a name.
7. Create a high level of trust among the people involved, by working, celebrating, and conducting external events together (ibid., p. 63).
8. Perform all necessary trainings for your guiding coalition.
9. Develop a shared vision and goals that touch both heart and mind (ibid., p. 67).
10. Create a Product Backlog.
11. Make everything transparent that is planned or done by your guiding coalition.
12. Prevent and face rumors.
13. Communicate as clearly and often as you can.
14. Learn the motives of all the people involved to ensure you are followed.
15. Lead by example. Be a role model. Your colleagues should want to follow you.

Reference

Kotter, J. (2012). *Leading change* (New ed.). Boston: Harvard Business Review.

³ A rumor wall is a pin board that can be used by the staff (including you) to hang up rumors or uncertainties that have been heard. The guiding coalition visits this wall daily and answers or corrects the rumors on the same board. This usually leads to the fact that after a few weeks or months, no more rumors are circulating, but people come and ask first.



Quite often, employees are not initially convinced that radical changes are necessary at all. Even in the face of an outstanding urgency the majority of staff usually accepts the problems but does not accept that significant change is necessary to resolve the issues. They accept a vacation freeze or forced overtime rather than break their entrenched patterns. Even if you manage to make your employees willing to accept changes, frequent disagreement and confusion are still prevalent. Therefore, it is important to clearly define the direction of change. “Vision refers to a picture of the future with some implicit or explicit commentary on why people should strive to create that future” (Kotter 2012, p. 70). It allows your employees to follow your leadership.

8.1 Vision and Strategy in the Context of Leadership and Management

John Kotter is one of the few authors who clearly makes a separation between leadership and management. Kotter (2012) argues that leadership is concerned with visions and strategies, while management is rather dealing with actions at the level of plans and budgets. So leadership is concerned more with the strategic issues beyond the daily business, while management focuses on the operational issues. In this book, you have often read the term “leadership.” I used the term “leadership” in the spirit of Kotter and want to briefly explain the reasons why.

First remember your own career. Recall the people you encountered throughout your life so far. Who impressed you most? Did these people create detailed tasks, timetables, and budgets for the operative implementation of strategies, or were these people visionary personalities who inspired others with their ideas? Personally, I remember the visionaries with great joy—especially if they had weaknesses elsewhere. That makes them more human.

Now think back to difficult situations in your life. Whether in business or private life: What gave you courage and allowed you to stand up and fight? Was that a

checklist or was it the vision of a better future? Was the initial pulse a specific call to action, or a more abstract strategy?

Now put yourself into the situation of your employees: Your boss tells you that the company is in a crisis. She has told you clearly that you will be unemployed in 3 years if the company cannot find a way out of this crisis. She has certainly made it clear that no employee will be laid off—but you know from the press, how seriously such announcements are taken when it comes to the crunch. Although management meets regularly and has given their task force an important sounding name, so far no one has told you options exist to get out of the crisis. The rumor mill is in full spin. You want to break out of this demotivating vicious circle, but the only way you can think of is quitting your job. You do not want to do that: Your spouse and children most certainly would not approve. You want to help; you want to save the company. But you do not know how. What do you need right now: A vision that shows you the direction, or a detailed schedule?

At the end of the day you require both, of course. The first and most important step, however, is that you sense an ideal you can follow. Only if you can believe in the desired target state and find it worthwhile, are you likely to take a closer look at the road that leads there. Only if you trust the vision will you be able to trust the people who embody this vision. To say it in the words of Antoine de Saint-Exupery (2003): “If you want to build a ship, do not drum up people to collect wood and do not assign them tasks and work, but rather teach them to long for the endless immensity of the sea.”

8.2 How to Create Vision and Strategy

You should create your vision and strategy jointly within the guiding coalition (cf. Kotter 2012, p. 82). It is about the vision for change, not the vision of the guiding coalition itself, even though the two are usually closely linked. Start with the urgency when creating the vision. Write it down, ideally on a large sheet of paper, and hang it up somewhere so that it is constantly present. Then find all those affected by this urgency. Often these are not just a small group of people, but all employees, shareholders, and customers. Brainstorm and then develop the first draft of a concise vision. This vision needs to be short—often ranging from one to two sentences—but still create a clear picture of the vision in the reader’s mind (ibid., p. 84). Everyone must immediately be able to associate an image with this vision. This image must be worth striving for, by addressing the long-term interests of the people affected. The vision also has to be ambitious enough to tear the people out of their daily routines. Do not aim for slight improvements, but demand to be the leader in a particular area. To remain credible, the vision must also appear feasible, and thus contain realistic goals. Nobody likes fighting windmills or building castles in the air. It becomes more difficult to assess how specific the vision should be. On the one hand, it must be specific enough to guide decision making, for example in the creation of strategies and plans. On the other hand, it must be generic enough to allow leeway. The conditions are very likely to change and it is counterproductive if

you have to change your vision every 3 months. If you have managed to create a specific vision with room for maneuver, be aware of the communicability. You have to be able to explain your vision in a way that everybody understands within 5 min (*ibid.*, p. 80). If you find yourself faltering, you should revise your draft again.

Once you have created a good draft, you should analyze it again point by point. Check the conceivability by asking questions from different perspectives. How will this vision impact customers/employees/shareholders? Will the degree of satisfaction change within one of these groups? Does this vision appeal to both the hearts and minds of the people? Remember that you should not compromise at the expense of the parties concerned (*ibid.*, pp. 74–76). In today's competitive economy in every market you cannot afford to lose your staff or alienate your customers.

Check the feasibility of your vision with your guiding coalition. Do not be afraid to involve additional employees who were not directly affected so far. Make sure, however, that no half-truths and rumors are spread afterwards. You are still in the orientation phase—you should make this very clear.

Then ask critical questions whether your vision is designed open enough to allow room for employees' own initiatives and bandwidth for changing surrounding conditions (*ibid.*, pp. 78). Also question if your vision is focused enough to show your employees a clear direction, which makes it explicit what measures are important and which ones are forbidden. Only if your employees can relate to the vision they will follow it. If you cannot align all of the feedback, focus on the communicability: There is only value in what you can clearly communicate (*ibid.*, p. 79).

Often, the orientation phase is protracted because too many details are packed into the vision. Free yourself of it by directly writing down those points and setting them aside for the strategy creation. Refine your vision until the entire guiding coalition is satisfied. A foul compromise will take revenge on you later—do not allow that to happen. Also, do not expect to create the vision in one afternoon. It will need a number of iterations and a whole lot of energy (emotional and physical) in order to achieve a good result. Count more on 1 year than 1 day—especially if you do not yet possess an analysis of the available options and needs. If you need fast results, it may make sense to first propagate a small change project with a corresponding smaller vision and aim for a larger change in a second step. You should then communicate this accurately to ensure that the employees do not get the feeling that a flood of change projects are on their way to overwhelm them. This is a sure way to cause them to lose their confidence in you.

Write your vision onto a large sheet of paper and hang it up. You can start creating the strategy even before your vision is finalized. However, you can only finish and communicate the strategy when the vision is ready as well. List all the strategic points that clearly show the way to achieve the objective. Incorporate all the people you think are important, but at least the entire guiding coalition. Consider all the data you need and do not be afraid to ask. Your strategy should meet the same criteria as the vision. The scope can be wider and the content specific. You should be able to roughly explain your strategy within 5 min as well, though not in every detail. Remember: What you cannot communicate is

worth nothing. If you are slipping into detailed discussions too often, frequently ask yourself the question whether this level of detail is appropriate for a strategy, or whether you are discussing operational implementation issues. Postpone them in the latter case but make sure to write them down, so they are not forgotten.

8.3 Things You Should Remember

The vision and strategy for change form the foundation of your change process. If you do not fully meet the requirements of this phase, your endeavor will crumble to dust sooner or later and you will have to start over. Remember the following points, backed up by Kotter (2012, pp. 75/79/82/84):

1. Just because the urgency is accepted it does not mean that change is accepted as the solution. A vision for change shows a clear direction toward a desirable future.
2. Visions and strategies are a tool to lead your employees.
3. Vision and strategy are jointly created within the guiding coalition.
4. Start with the urgency in the creation process.
5. Make use of all available data. If data is missing, obtain it.
6. Match your vision to the needs of those affected.
7. Set challenging but achievable goals.
8. State the vision specifically enough to serve as a guide, but generic enough to allow leeway.
9. Put the focus on communicability. If you cannot communicate your vision within 5 min to those affected, you still have work to do.
10. Fundamentally question your vision from the perspective of those affected.
11. Focus on the essentials and record the rest for later use.
12. The creation of a vision is a lengthy process that can take up to 1 year.

References

Kotter, J. (2012). *Leading change* (New ed.). Boston: Harvard Business Review.
de Saint-Exupery, A. (2003). *Wisdom of the sands*. Hardcover. Mattituck: Amereon.



The communication of your vision is a clear leadership responsibility. Only someone who knows your vision has a chance to follow it. Most managers communicate far too little, especially in large change efforts. Sometimes they transport diverging or contradicting messages through different channels. Both mistakes cause your change project to fail (cf. Kotter 2012, p. 87).

9.1 What You Always Wanted to Know About Communication

Matthias Mehl published a study in July 2007 in the journal “Science”, examining the communication behavior of men and women. The result shows that both men and women speak an average of 16,000 words per day. Very communicative volunteers even managed up to 45,000 words a day. Communications in which the volunteers did not actually speak themselves, such as reading e-mail or listening to a lecture, was not included in these numbers.

Although I suppose that you are an above average communicator, I will keep it simple and stay with the average number of words of the study. Let us assume that half of your communication is professional. How many of your 8,000 words did you use today on the communication of the change effort? If the urgency really is that high and the change really is so important, you should probably grant these subjects a prominent position in your daily communication.

How much communication is sufficient for your change effort to be successful? This question cannot be answered easily because it depends directly on the size of the change and the clarity of your vision. The vision should at least meet the following criteria (cf. Kotter 2012, p. 91):

1. Simplicity: Do not use technical terms, stay with words that everyone knows. Even Scrum terminology counts as technical jargon, so avoid it in your vision. Besides, Scrum is not an end in itself.

2. Images: Use language or real images to help our brains to understand the information and store it more easily.
3. Apparent discrepancies: Avoid these, but go looking for them. If you cannot do without such discrepancies, you must explain them clearly.

Another complicating factor is the type of communication. People absorb information from different channels to different degrees. Therefore, you must use as many channels for your communication as possible. Once information arrived at the receiver, this does not mean that it is also stored. Our brain discards everything it deems unimportant within 24 h. Only by repeating the information it can be migrated to our long-term memory. Test yourself: Without looking it up, please state what you must do in order to build a successful guiding coalition.

Are you sure you recounted every important point? I am not sitting next to you, but I suspect that you could not list everything. This is unfortunate, but normal. The information I asked for is just two chapters earlier in this book, yet your brain has already deleted everything it deemed to be unimportant since you read it. It is the same with your employees.

Use all channels for communication available to you (cf. *ibid.*, pp. 95–97): presentations (e.g. working meetings), articles in the company newspaper, e-mail, face-to-face conversations (both formal and informal), discussions, workshops, posters, flyers, stickers, and everything else you can think of. Use these channels as often as you can. At the point when you are sick and tired of all the communication, the individual employees have not heard it often enough by a long way. The toughest, yet most effective interactions are bi-directional, i.e. those in which the recipient of the information has to get actively involved (cf. *ibid.*, pp. 101). This form of communication brings with it a perceived risk that your counterpart unveils deficiencies in your vision or putting it positively potential for improvement. You might fear that you subsequently have to start again from scratch. This risk should be understood as an opportunity: Better to fail early and having a chance for correction than investing a lot of time and work into a path that does not solve your problems. This feedback loop is a form of risk mitigation and does not pose a risk in itself.

In a large company a change initiative (which had nothing to do with Scrum, but concerned the larger company context) was kicked off with an all-staff meeting. The chairman of the company also visited each business unit location and gave a presentation on the importance and vision for change. For most people this was the first time they had seen the chairman at all. It was seen as an honor and an indication of the importance and urgency of the situation. In addition, each business unit had to conduct a change workshop with all employees, hosted by staff from the corporate headquarters. There, urgency and vision were sharpened and adapted by the employees for their own daily work. Very emotional elements were also included, which talked to the hearts of the participants.

Each issue of the company newspaper included at least one (but usually three or four) articles related to the urgency and demonstrated successes along the road to achieving the vision. On special occasions, the chairman sent out emails with motivational content to all employees. All managers were encouraged to include the key messages of the vision in their daily communication, which worked well for most of them. Posters and key lanyards accompanied the employees at every step. From the viewpoint of communication, this was probably the best change initiative I have ever seen. However, it took more than a year to get all employees involved in the change effort.

You can see that a lot of work is waiting for you. In addition, you need to show strong leadership and communicate your vision through your own exemplary behavior. Your actions and omissions show your employees more clearly than a 100 memos would do, on what you consider really important (cf. *ibid.*, pp. 97–99). Most organizations fail right here: The leaders do not exemplify the necessary changes, but remain in retracted behaviors. This guarantees with absolute certainty that the employees offer passive resistance. The fish rots from the head down. Managers shape their organizations through their own actions.

9.2 Things You Should Remember

The communication of the vision, and therefore of the change itself, is an essential part of your change process. Only someone who knows your vision can decide whether he wants to follow you on the difficult path that lies ahead (cf. Kotter 2012, pp. 91; 95; 97; 99; 101).

1. The communication of the vision is a clear leadership responsibility.
2. Keep your communication consistent across all channels.
3. Outstanding issues should occupy a prominent position in your communication.
4. A simple, clear vision that was created with the help of images and without apparent contradictions can be easily communicated and understood.
5. Communicate through all available channels.
6. Communicate constantly and daily.
7. Use bi-directional interactions, such as personal interviews or workshops, to better convey your message and learn from the feedback of your interlocutors.
8. Your behavior (and that of your managers) is the reference element in the communication of change.

References

- Kotter, J. (2012). *Leading change* (New ed.). Boston: Harvard Business Review.
- Mehl, M. R. (2007, July). Are women really more talkative than men? *Science*, 317, 82.



Empower Your Employees on a Broad Basis 10

Once you managed to communicate urgency and vision in a way that motivates your employees to follow you, it is time to let everyone participate in the change. In other words, you need help because you will not be able to implement everything yourself with just the guiding coalition. All-encompassing change does not happen all of a sudden, but rather person by person. Your task now is to turn affected parties into involved ones.

10.1 Transform Affected Parties into Involved Ones

The greater the change, the more people are affected. Accordingly, you need to convince many people of the necessity to change. Once these people finally want to help, it is your job to make sure they can. Here you will encounter various issues (e.g. cf. Kotter 2012, p. 105):

- Distrust
- Lack of skills
- Attitudes of your employees
- External inhibiting factors such as structures, systems, and supervisors

Overcoming distrust is particularly difficult. Often it is a completely new experience for the employees to suddenly shape processes and structures, rather than strictly conforming to the orders of management. Suddenly the concept of “empowerment” drifts through the hallways, but the employees smell a rat: Is this just a management ploy to squeeze more time and energy out of the staff? At this point at the latest it becomes clear why top management must be involved and present, because only leadership at top management level can ensure that all staff members truly believe they are allowed to actively live the vision and contribute their ideas. If you succeed, you will encounter the phenomenon that some people do not believe they have sufficient skills to live the vision. This attitude partly stems

from a lack of confidence, but sometimes it is simply true. You will have to invest in training if you want to be successful with your change endeavor. It involves both the technical skills of the participants—for example Scrum training as well as seminars that influence people’s attitudes. For example, if customer focus is part of your vision, then you should seriously consider providing training that sharpens this area (cf. *ibid.*, p. 112). Experience has shown that it makes sense to act from two angles for all training needs: the self-assessment of employees is a valuable pointer. On the other hand, external evaluations should not be neglected. There are many reasons¹ why some people try to avoid training. You are on the safe side if you generally train all employees in the core topics. Here, you must of course carry out a cost-benefit analysis.

Closely linked to the skills are the attitudes of your staff. There are always some colleagues who at first sit on their hands and wait, or who completely refuse the change. You will also meet colleagues who claim to indeed support the change, but their actions say the opposite. Every single point of your vision can run counter to the personal attitudes and values of your employees and trigger corresponding reactions. On the one hand, you can face this issue with training, as described above, although this is only one piece of the puzzle in changing the attitudes. Equally important, if not more so, is open communication and convincing people that the required changes are really important. The previous work on urgency, vision, and communication are now paying off—or retaliating if you have not done it carefully (cf. *ibid.*, p. 105).

It may be necessary for you to let individual employees go who are not willing to work on their attitudes.² On the one hand you will not want to lay off longer serving employees; on the other hand they also have the greatest potential to adamantly insist on their “earned” positions. Usually face to face talks can help clarify the situation, where the reasons for the insistence are discussed. Typically, a solution can be found, because the employee either can be persuaded, or training or other needs are revealed. Should you however have to deal with a person who is not willing to listen or to change, you might have no other choice but to part company with him. If this person remains in the company, he will hinder the change (cf. *ibid.*, p. 117) and spread his resistance to other colleagues like a virus. It is easier to wade through a swamp, than to succeed in a big change with a resistive group of colleagues.

Once you have overcome all these factors, you arrive at the part that makes up the lion’s share of your work: the external inhibitory factors. This refers to established processes, models, systems, and practices that are normal in your

¹The reasons can range from overwork through to a deeply rooted fear of failure. Since such analysis is beyond the scope of this book, I do not take it further here. If you speak German, I recommend you to read: Fritz Riemann, *Grundformen der Angst* (Riemann 1985).

²Laying off employees is absolutely the last option, once everything else has failed. Working professionally with criticism, responding to the employees, and motivating them are the primary tasks of management in changing organizations. Only if all else fails should you consider parting with an employee.

organization, but contradict your vision (cf. *ibid.*, p. 115). In the following section, I go more into detail using some relevant examples, which are often encountered in connection with a Scrum introduction. Solve these inhibitory factors in order to achieve your goal!

10.2 Typical Inhibitory Factors Regarding Scrum

You will encounter a number of inhibitory factors when introducing Scrum. Depending on which Scrum shape you want to achieve, you might choose to maintain an inhibitory Scrum factor because you consider it beneficial to your organization. That is okay. Evaluate both advantages and disadvantages individually in each and every case. It is also helpful to ask whether the sought benefits could be achieved without Scrum. Inspect the situation and adapt your behavior accordingly.

In this section you can find common inhibitory factors. Their severity depends on your specific context.

1. There is no single Product Owner/product responsibility and accountability are not clear

Far too often there is a group of senior managers who jointly decide on the direction of the product. This leads to nobody really feeling responsible and decisions taking a long time. In addition, sudden changes of direction are more likely in such an environment compared to an individual living up to his responsibility. These are the reasons why Scrum demands a single Product Owner who has the singular power to make product decisions. However, it is common sense that the Product Owner should usually base his decisions on objective criteria and his stakeholders' wishes.

2. Every project needs a project manager

Some organizations hold the view that every project must be led by a project manager. The project manager then has the sole responsibility and will of course manage it accordingly. This is a direct contradiction to Scrum, where responsibility is clearly distributed: The Product Owner is responsible for the product, the Scrum Master is responsible for the process, and the Development Team is responsible for the implementation, including the technical quality. A project manager will constantly cause trouble here. Depending on her abilities, she must choose one of these three roles, or find fulfillment in a different position like program or line management.

3. Career models for Scrum Masters and Product Owners do not exist

"I do not want to be a Product Owner. I am a product manager today; that is a far more important title!"—I often hear these and similar statements. The reason is that some individuals are heavily dependant on traditional career models that actually contradict the Scrum Culture. However, if the organizational culture places significant value on titles and this cannot be changed in the short run, companies have to provide functional careers for their specialists.

Connecting your hierarchy-focused career to Scrum will cause you to implement a management career path (for line management positions), as well as a specialist career path (for developers) and a Scrum career path (for Product Owners and Scrum Masters), which is somewhere in between leadership and specialist paths. For example, if a hierarchy-infected Scrum Master can get promoted from Junior Scrum Master to Senior Scrum Master, he is more likely to be motivated to use all of his skills productively. This productive response is highly unlikely if he can only rise through the ranks by switching over to a management career path, for example by filling the position of team leader. Also remember to make the various career paths permeable, so that switching between paths is possible at any time. Keep in mind that career paths as such oppose the Scrum Culture. However, if you cannot get rid of them, try the three paths approach. Besides, in a matured agile organization these titles most likely represent skill levels, not decision power or traditional hierarchy.

4. Recognition is primarily defined by the pay level

People born after 1980, so-called digital natives, demand different speed and types of recognition than previous generations did.³ The pay level must be high enough to prevent dissatisfaction but is pretty irrelevant otherwise (cf. Pink 2011). Far more important are short, timely feedback cycles. For example, a text message containing the praise, “Your presentation was great today. Well done!” contributes far more to employee satisfaction than feedback during the annual performance review. Knowledge workers especially have to be highly motivated to be productive. Your Scrum Master will do her best to ensure this—if you let her. Pizza once a month or a grant for a team bowling evening should be possible. Are your personnel and your controlling departments prepared for this?

5. There are no team facilities

Usually corporations introduce Scrum because they believe it increases the productivity of their teams. This is wrong. Scrum does not directly increase performance—it just makes problems transparent so you can solve them. Once they are solved, your productivity might rise. A dispersed team is at least 40 %—more likely 80 %—less productive than a collocated team. Just 30 m between offices are enough to cause this productivity loss. It is therefore necessary to place the whole Scrum team in one room—without other disturbing influences. Scrum teams discuss a lot and sometimes produce some noise; the team would therefore only disrupt other colleagues. It is the same the other way around—open plan offices do not work. You also have to make sure that there are enough meeting rooms and “quiet corners” where people can work in concentrated fashion without interrupting noise.

³There are certainly similar demands in other generations as well. However, these are far more common with digital natives. There are many articles and books about the topic, if you want to dig deeper.

6. The premises are under-equipped

Modern Scrum developers need more than just desks and chairs. Laptops with fast hard drives, plenty of memory, and a good processor are equally important as several large screens and a seamless backup infrastructure. In addition, each team needs at least one whiteboard and flipchart in their room. Moderation equipment like sticky notes, pens, cards, and sticking dots goes without saying. However, your team will only be truly content if they can design their space themselves. Give them an appropriate budget and do not question the team's decisions as long as they were made in consensus. No matter if this leads to posters, plants, table soccer, or a video game console—it is important that the team feels at home.

7. There are central corporate functions that want to control the Scrum teams in an inappropriate way

“The team organizes itself” is a guiding principle of Scrum. Every Scrum role has its own tasks and responsibilities. Some of these tasks might have been in the custody of central corporate functions before Scrum was introduced. They do not want to give up their power, which clearly causes conflicts. Typical examples of such central functions are concept development, quality assurance, software architecture, and process management. The Scrum teams solve these tasks. If the people from central departments still want to make a contribution, they have to join the Development Team as regular team members.

8. Administrivia eats up the teams

DeMarco (1997) coined the term Administrivia. It describes bureaucratic efforts that are serving themselves rather than the overall result. There are cases in which the preparation of documentation prior to the annual appraisal interview keeps the employee busy for a full working day. For the project, that is a day lost. The many small losses in daily life are much worse though: Here an application, there documentation, and just a few KPIs⁴ over here. Measure how much time your employees spend on such administrative tasks. Then decide whether this time is adding value or not. It is very likely that you will not need everything that is currently being “filled in”.

9. Budgets are defined and granted at the beginning of the year

With Scrum you produce your results in short iterations. You inspect and adapt the direction in which the product has to evolve every few weeks. You also verify every few weeks what changes must be done to the framework and processes in order to be more productive. However, if you collectively release all your budgets at the beginning of the year, the findings of the upcoming months are not included in the calculation. To exceed the budget is simply not possible in some companies. You are giving away valuable productivity—which also means money—in these cases, because your processes do not allow you to keep up with the most recent findings. You cannot exploit the

⁴ KPI stands for Key Performance Indicator—or simply metrics.

full potential of Scrum in this scenario. Quite often, this inhibitory factors appears together with the next one.

10. Projects are approved on the basis of time estimates

Processes following the waterfall or phase model usually first create a product requirements document and associated functional and technical specifications; then they release the project budget on the basis of these. Of course, in the technical specification, each task is listed, accurately estimated on an hourly basis, and recorded in a project plan. Unfortunately, this approach does not work for complex products such as software. Although this is well known to the general public, organizations hold on to this system since they are lacking a better approach. Scrum makes it clear that there is no absolute predictability in software development and we can only forecast the outcome. If the Development Team remains constant and trust exists between client and service provider, such a predictive approach is no longer needed. The pursuit of absolute predictive certainty is an expression of a deep mistrust caused by the software development experience of the last 30 years. Scrum does not forbid you to plan everything in advance. However, Scrum shows you very clearly that you are wasting a lot of time if you do. Try to approve projects on the basis of trust and empirical data, instead of doomed assumptions about the future. If you cannot manage to escape that situation, there are still options for you to move into the agile domain. One common method is for example to first have the Development Team estimate the whole requirements document, as you would for any fixed price project. Then you should add two rules to the package: If requirements are added or altered, they will be estimated and equal amount of work has to be removed from the backlog. This does not result in any costs for the customer. The second rule is that the customer can stop the project at any time, if he pays 20 % of the remaining project volume. So the customer's risk is reduced to 20 % while the agent has enough time to look for another project.⁵

11. Training is not approved

Training is expensive, especially because the people do not work on the project while they participate in a course. Usually, some training for individuals does get approved, but training for larger groups does not happen. The introduction of Scrum is a fundamental change in mindset, values, and procedures that must be engrained in the team. The company is facing the classic question (cf. Covey 2013): “Sharpen the saw, or cut down the tree?”

Absurdly, organizations often choose to continue working with a dull saw instead of investing in their employees. This is precisely the point: Whoever sees training as a cost rather than an investment will have a hard time introducing change. This is true for a Scrum introduction as well, of course.

⁵This concept is called “Money for nothing and changes for free” and was presented by Jeff Sutherland in 2008.

12. Development tools are mandatory and non-negotiable

If you want to deliver “done” software monthly or even weekly, you must have the appropriate tools. Continuous testing and delivery are just two aspects that need to be considered. These necessary tools are sometimes not known to the IT operations department, are not at all approved, or have to go through extremely time-consuming evaluation projects. You can compare this to a craftsman who is being told that he has to manually turn the screws into the wall, as the electric screwdriver needs to be tested and might, if he is lucky, be released in a couple of months. Try to accelerate such approval processes. Every day you have to wait for the proper tools will cost you a lot of money and employee motivation.

13. Technical requirements can not be fulfilled

A Scrum Team places higher demands on its environment than traditional Development Teams. This starts with communications infrastructure (dispersed teams at least will need web cams, headsets, a fast internet connection, and VPN access) and does not end with strong development servers that offer branching.⁶ If your company is unable or unwilling to fulfill these technical requirements for any reason (in Germany, you need the approval of the workers’ council if you want to install and use cameras, for example), your team cannot unfold its full productivity.

14. Developers work on multiple projects at the same time

Each switch between tasks requires time. Time to engage intellectually with the new task, time to recapitulate what has already been done, and of course, time to think again about the specific solution to this task. This is also true when working on several projects at the same time. You lose about 20 % of your productivity for each additional project. So if your employee is working on two projects simultaneously to 50 %, only 40 % of his effort is delivered to each project. The rest is lost in the switching process (cf. Weinberg 1991). Scrum does not require you to only pursue one project at a time. However, Scrum will show you very quickly and that this multitasking is painfully defocusing you, which reflects very negatively on your productivity.

15. Team compositions are changed

Organizations that view their employees as soulless resources like to move these resources from team to team to see more person-days reflected on the project sheet. This Taylorist thinking works very well in simple environments such as assembly lines, but is totally inappropriate in thought-intensive environments such as software development. A team is only more productive than the sum of its individual members, if an emotional bond exists between the members and a common mode of operation was established.⁷ This takes time

⁶ Branching means that a Development Team has their own development environment (called “branch”), which is derived but detached from the main system (called “trunk”). This allows the team to only add working, tested code to the trunk and thus protects the integrity of the system.

⁷ The team building process goes through different phases: forming, storming, norming and performing (and adjourning). cf. Tuckman (1965).

and cannot be controlled, it has to grow. Every time the composition is changed, you have to start from scratch again. In addition, the team will build a deep distrust against the arbitrariness of management, which results in decreased motivation.

16. Decisions over the heads of the team

In large organizations, decisions about work places, tools, team compositions, etc. are often made over the heads of those affected. Acceptance by the employees is therefore very difficult and can only be achieved by forcing it through hierarchy. This in turn reduces the motivation of the team and reduces the ability to self-organize. This is common sense, since there is no reason why the employees should think for themselves if all wisdom “falls down from above” anyway. You cannot derive many benefits from Scrum if self-organization is lacking, since the team will not participate in the continuous improvement of your processes. Scrum will slowly die out if this problem is not remedied.

17. Managers insist on directly accessing the developers

It is very convenient for managers to submit their wishes and ideas directly to the developers. This prevents long bureaucratic processes and spares you from forgetting what you wanted. Especially if everybody is doing it this way, this might be the only way for you to get your requirements implemented at all. If personal goals (that influence your bonus at the end of the year) are added to the mix, Scrum is quickly abandoned.

Scrum demands that all requirements are submitted into the process through the Product Owner. She is the only person who is allowed to sequence those requirements. So without a Product Owner, there is no Scrum, no focus on the greatest business value, and no delivery in short iterations, which in turn leads to an increased need to circumvent the process and submit your wishes directly to the developers. Break through this vicious circle by insisting on the Scrum rules.

18. Personal goals are dominating corporate goals

What is contributing more to your annual bonus: personal goals or corporate targets? Usually the personal goals are dominant. Thus, your personal pay depends not on doing what is best for the organization, but on doing what has been agreed on at an individual level. This is especially problematic if the objectives of several individuals contradict each other—which is particularly evident in product management, if manager A needs to implement feature one, while Manager B is required to introduce feature two, and both want to use the same team. You can save many endless discussions if you give up personal financial goals altogether. Focus on personal development and overall corporate success instead.

19. There is no product vision

Unfortunately, work is far too often prioritized based on the current day-to-day business rather than being founded on a sophisticated product vision. It is often not distinguished whether a feature is really valuable for the whole product or only for a very “loud” customer. This leads to losing the focus on “Total Cost of Ownership (TCO),” which is the consideration of the total cost

that the product incurs well beyond pure implementation. You are particularly successful with Scrum if your Product Owner bases his decision on a well-coordinated product vision.

20. Short-term results are valued higher than long-term success

How far in advance is your organization thinking? Do you think in months? Quarters? Years? Very few think strategically in 5- or 10-year increments. Your product will not only exist for a quarter or two. It will most likely be revenue generating for at least 5 years. Many companies hold the benefits of a single year against the costs of a single year of development. This is shortsighted. Instead, the full benefits over the remaining lifetime of the product should be calculated versus the total costs to be incurred up to that point. Many decisions would probably have been met differently. Think in products, not quarters.

21. Command structures are preferred to trust

Wherever there is a lack of trust in the employees, organizations try to build up an impenetrable control system. Responsibilities are defined, job descriptions are written, committees get installed, and so on. At the end of the day there is a clear hierarchy that delegates power and responsibility to the top. Unfortunately, operational decisions have to be made where they are needed: in the teams. Military structures do not help here. Decision-making authority needs to be assigned to the appropriate expertise instead. This is quite simple on the formal level, but it will pester you on the meta level for some time: you have to build trust in both directions. Management must not only trust the team, the team must also trust management.

22. The dominance of the KPIs

Is there anything better than meaningful metrics (Key Performance Indicator, KPI) to clearly assess the performance of a department, team, or individual? Unfortunately, this notion of controllability through measures is a myth. As soon as the information supplied by these metrics is no longer considered value-neutral, but instead is used to control whatever was measured, their effect is inverted (cf. Campbell 1976). In short, one will always find a way to improve the metric itself without improving the true performance. One example: Every now and then companies still use the number of written lines of code as a developer's productivity measure. As you know, all developers are able to produce as many lines of code as they want—without delivering a single additional feature. Producing the required lines of code takes time though. As a result, the metric improves, but the actual output drops—KPI perversion just struck you. In order to avoid this trap, you should capture as few indicators as possible, use them only in a general context, and consider them value-neutral. You are never allowed to let rewards or punishments follow these metrics, or they will immediately be rendered useless.

23. A high level of specialization prevents developers from sharing responsibility

In some teams, individual developers have highly specialized skills. This can be true for domain knowledge (e.g. feature areas), programming, or any other skill (e.g. architecture, testing, tools, databases, etc.). If the level of specialization is too high, dependences can occur, which slow the workflow down. In

addition, the individuals will not feel responsible for those aspects of the products that are outside their area of expertise. This weakens the team spirit and the overall product quality. To cover all product areas, a lack of generalists can, in some cases, lead to teams that exceed the ideal size of three to nine people by far. If this is the case in your situation, discuss the issue with your Development Team and ask them to suggest a solution. The case of dependencies can especially awake the team's interest and leads, with a high probability, to good suggestions. These could for example include working in pairs (pair programming) or training within the team. Try to avoid all suggestions that move in the direction of splitting the team based on their technical expertise. This only moves you further away from agility.

24. Departments and teams fight each other, there is no common mission

Perhaps you have experienced this yourself: Instead of providing the information you require, a neighboring department stalls you. This continues until you miss your deadline or escalate the incident. Unfortunately such power struggles are very common. In general, the bigger companies are, the more frequent and tougher these power struggles are. What you need here is a common overarching goal, a vision, and a mission statement. These should then be reflected in the employees' personal bonus objectives. All personnel must be aware that they are working towards the same goal—together for a better future, and not against each other.

At one of my customers' the programmers almost hated the quality assurance department. QA constantly highlighted new requirements that were perceived as patronizing by the programmers. From the programmers' perspective, the individuals within quality assurance behaved as if they were superior to their programming colleagues, making decisions over their heads. Conversely, the quality assurance staff felt threatened by the programmers because they simply ignored their orders and directives. It was only when I started to bring all parties around a table on a regular basis that mutual trust started to build. I simply asked them how they wanted to work together in the future to develop software that also matched the quality requirements of the company. Three months later, the departments worked productively together.

The list above is not exhaustive, of course. There are many other issues that may arise. I limited it to those problems I encounter very often. Please be aware that the occurrence of one of the above does not necessarily mean that you are acting "against Scrum". That is completely irrelevant. It is all about making your business more productive with your employees enjoying their daily work more. Incidentally, these problems probably already existed without Scrum and could also be solved without Scrum. If you implement Scrum however, you get the major advantage for free, revealing issues like those above which impact the process like sand in the gears. They will be made transparent by the Scrum Teams over and over again and the productivity reduction will be visible. This allows you to address and solve these issues—an opportunity that is not necessarily given to you by more traditional process models.

10.3 Things You Should Remember

After you successfully communicated urgency, vision, and goals, the real work starts (cf. Kotter 2012, pp. 105; 110; 115; 116).

1. To enable broad-based action by your employees, you need to solve four key issues:
 - Distrust
 - Lack of skills
 - Employee attitudes
 - External inhibiting factors such as structures, systems, and supervisors
2. Therefore it is absolutely necessary to successfully conclude the previous change phases. Particularly the formulation of urgency and vision, and their communication must have taken place.
3. Create confidence by having your entire management exemplify and lead the changes. In addition, you have to constantly encourage your employees to actively participate.
4. Meet lack of skills with training. When assessing the capabilities use both self-perception and external perception.
5. The attitudes of your staff can be modified by talks, education, and training. In individual cases, however, this may prove impossible. In such cases it may be necessary for you to let one of your employees go. This should be the last resort though.
6. External inhibitory factors such as structures, systems, and supervisors must be analyzed and adjusted.
7. Some inhibitory factors need to be tackled before a Scrum introduction, while others can be addressed at a later time, or even left as they are. Consciously make these decisions and clearly transport the reasons as well as the consequences to everybody involved.
8. Lead by being a role model!

References

- Campbell, D. T. (1976). *Assessing the impact of planned social change*. Occasional paper series. Reprinted with permission of The Public Affairs Center, Dartmouth College. December 1976, Paper #8, pp. 1–70.
- Covey, S. R. (2013). *The 7 habits of highly effective people* (Anniversary ed.). New York: Simon & Schuster.
- DeMarco, T. (1997). *The deadline: A novel about project management*. New York: Dorset House.
- Kotter, J. (2012). *Leading change* (New ed.). Boston: Harvard Business Review.
- Pink, D. H. (2011). *Drive: The surprising truth about what motivates us*. New York: Riverhead Books.
- Riemann, F. (1985). *Grundformen der Angst* (12th ed.). Paperback. E. Reinhardt.
- Tuckman, B. W. (1965). Developmental sequence in small groups. *Psychological Bulletin*, 63(6), 384–399.
- Weinberg, G. M. (1991). *Quality software management: Systems thinking*. New York: Dorset House.



You have analyzed your urgency, developed your vision down to the details, and communicated it so vividly that even the cleaning staff knows exactly where the journey is headed. You have also empowered your employees on a broad basis and removed the biggest obstacles. Excellent! If you now believe, however, the work is done, and you could sit back, then you are mistaken. Even if you really have done everything that is needed for your long-term success, this is only true for the moment. To keep the motivation of your employees high for the long run—and your own, of course—you must also ensure short-term success (cf. Kotter 2012, p. 123). In this chapter, you will learn why this is so important and what you should heed.

11.1 Why Dreamers Need an Alarm Clock

To create a vision that appeals to both heart and mind, you need dreamers. Only dreamers can imagine a world in a fantasized form that does not yet exist. The results include brilliant product ideas and strategies, a boon to your business. The problem with dreamers is that they can forget about the world around them while they are dreaming. This is harmful when trying to implement strategic actions—you need an alarm clock! Imagine your change initiative would only take 5 years to implement (which is unlikely): It is very valuable to have someone who can picture your 4-year-future organization in its improved form. Having someone who is sleeping for 5 years can be irritating though. I have asked you in previous chapters to follow Kotter's advice and create the vision together with your full guiding coalition. This way I asked you to dream. What do you think happens if the guiding coalition does not wake up and just keeps dreaming? Correct—your business sails without guidance through the turbulences of the market and will sooner or later run up on a reef. It happens either due to the rowers getting tired, or because the navigation direction is no longer valid. In all strategic considerations you must

never lose sight of reality. It is your privilege to move forward bravely and in a visionary manner—to carefully consider the present is your duty.

Slightly less pictorially expressed: Your vision must always be validated in the context of the current requirements. You must also ensure that the motivation of all employees involved does not wane. The only sensible way to achieve this goal is short-term success. “Short-term” may mean 6 months, but a few weeks or a few months should be preferred. The more achievements you can clearly visualize, the more successful you will be with your change.

With all the enthusiasm we should never forget that we can not change a company directly, but that we must inspire the people who work in a company. Only when these people have changed their behavior will the company culture gradually change as well. People get tired over time, especially if they work hard. Organizational changes are amongst the most stressful changes your employees possibly have to face. Short-term results provide strength to keep going (cf. Kotter 2012, pp. 123). Incidentally, not only your employees are strengthened, but you will be as well: Successes prove your efforts are valid, often cause a more rapid amortization of the costs, and also motivate the guiding coalition.

For the purposes of risk minimization you should also strongly emphasize short-term success: If your short-term achievements align with the context of your long-term goals, these achievements can show you if your vision is realistic or not (cf. *ibid.*, p. 126). In the context of a Scrum introduction for example, the transition of a single team to Scrum could prove—to you and all critics—that it works. The conversion of the entire software development department therefore becomes more realistic. If this first Scrum Team then starts to discuss the benefits with their colleagues, you have created multipliers who will give new impetus to your change project. The next teams will be more willing to try the new style of working, and may even ask on their own initiative to be allowed to work with Scrum.

11.2 Characteristics of Quick Wins

The most dominant criterion for short-term success is how rapidly these successes are achieved. The exact amount of time allowed depends on the size of the overall endeavor. For example, if you assume that a small reorganization will only take 6 months, you should realize your first quick win within 4–8 weeks rather than 5 months. John Kotter describes in his book “Leading Change” that quick wins could also be achieved after 18 months (pp. 123; 126). I disagree vehemently, since in our fast-paced world 6 months are perceived as an eternity. A year is usually considered long-term.

One of my customers implemented Scrum in one pilot team. The project was designed to run for six months. After just three months, the world had changed and the original target had become obsolete. The pilot team was disbanded; we had to start all over again. Despite the sudden interrupt, we could still deliver a working product with reduced capabilities. This was a huge step forward, which we would have missed if we had solely focused on long-term success.

Therefore, I urge you to generate many small successes, instead of a single large success over a longer period of time. Create unique results each quarter, if this is possible for you.

First reconsider the overall situation: The company is stuck in a crisis (or at least is facing a high urgency for change). A vision exists, showing a long and difficult journey out of this crisis. All eyes are set on the guiding coalition that will lead the organization out of the doldrums.

Being secretive doesn't help you here. On the contrary, self-marketing is essential for the guiding coalition. Every success must therefore be fully visible. Visible does not only mean that each employee has all the information about the success. It also means that this information has to be prepared in a way that everybody understands it. In addition, the data must be verifiable by the employees. No one will believe you if you only claim a success but do not make it "tangible". Once the employees have access to your accomplishments, they will put them to the acid test. Sometimes they will offer you friendly suggestions for improvements, sometimes harsh criticism is stated, and people might even vociferously call for your resignation. Therefore make sure the success you present is a clear one. There must be no obvious point of criticism (cf. Kotter 2012, p. 125). If the credibility of your successes falters, so does the guiding coalition and therefore the entire change process hangs by a thread.

Quick wins are important. However, many change initiatives are not successful, because too much emphasis is placed on short-term successes. Those who only think in the short term will never achieve their vision and make a real difference (ibid., p. 128). Therefore, it is extremely important that you always keep the vision in mind in everything you do, and carefully plan the short-term goals. Make sure your communication always connects the short-term goals to the long-term goals and your vision. Make it clear to everyone that you are still going to drive through with the full change.

"Transformation is not a process involving leadership alone; good management is also essential" (ibid., p. 133). In addition to transporting urgency and vision you need to achieve short-term results while always keeping the complete journey in mind. Do not underestimate the difficulty of this task! It helps to mutually support each other in the guiding coalition and, depending on the task at hand, placing greater emphasis on the guidance of the visionaries or the managers. The thin line between too much management and too much vision passes directly through that coalition.

11.3 Pilots

No matter which type of Scrum introduction you choose, you will need pilot projects. Only Scrum PRN introductions are exempt from this rule. Here, every project has the characteristics of a pilot project, even if it is the hundredth. Selecting the right project as a pilot is not easy. The proper preparation and subsequent implementation have their pitfalls as well. Below you will learn how you can avoid

the most common snares. It does not matter whether you are striving to establish a Scrum Software Studio or Profound Scrum—the procedure of setting up pilots stays essentially the same.

11.3.1 Identification

Suppose you got the permission to look for a Scrum pilot project—how do you proceed? What should you look for? It is good practice to first list all possible projects. These have usually not been started yet. Selecting projects that are already underway increases the difficulty, since the project members are already accustomed to certain processes. In addition, interdepartmental work, e.g. controlling or finance, might well be attuned making it more difficult to change. The larger your organization, the more potential project candidates will be on your list when you are done. Start to analyze these projects. State the budgets and people already assigned to the projects. The roles of the participants are interesting as well. Note down the goals of each project, determine which technologies are already decided, and visualize how critical the projects are for the organization. Start and end dates should certainly not be missed either. Now you can separate some of the projects out for the first time: you will need 1–3 months for your preparations. If projects have earlier start dates you will have to reschedule them. If that does not work, these projects must be rejected (exception: you already have completed the tasks listed in the “preparation” section). Projects with too long running lifetime are also poor pilot candidates. While they provide you much room for improvement, it takes too long to prove their success. The full success is only confirmed upon completion of the full project—all previous achievements are usually regarded as nice to have, but not crucial. If you have the chance, you should choose a project with 3–6 months duration.

The criticality of the project is also of considerable interest. If the project is not important enough your critics will dismiss all your successes with a shrug; after all, you would not have been successful with one of the really important projects from their point of view. While this argument is superficial and shortsighted, it is common and cannot be neglected. If the project is too important on the other hand, all managers of your organization will want to have their say. Your decisions will be questioned and significant pressure will be placed on you. However, you then have an ace up your sleeve: You will have every support you need to lead the project to success provided of course that you have gained management’s trust. It goes without saying that this is most valid for the most important projects. Therefore, go for one of the important projects. Gather your courage, prepare for tough times and go for it—you will not get more support or more visibility in any other project. There is an additional aspect for very important projects: The project members are more likely to be fully available for the project. Your goal should be to get all team members fully allocated (100 %) to your project. In practice, you unfortunately often need to share developers between projects. Project affiliations of 30 % and less are common. Be aware that your employees lose about 20 % of

their total productivity with each additional concurrent project. With more than four simultaneous projects you will not get any significant results. This is due to the time needed by your team to think their way into the other task, to change workplaces, to configure the work environment on their computers so that it fits to the new project, and so on. In addition, the time needed for meetings and processing emails increases significantly because each project generates a minimum communication requirement. Usually, even with two parallel projects, you are already missing an opportunity for a higher total productivity, by handling them in succession. I recommend you try this out yourself for several days by finishing all tasks in one context before focusing on the next one. You will probably be surprised.

Productivity can also be positively influenced if you select a project where the team members are already collocated in one location. It is much harder to lead distributed or dispersed teams¹ than to work with a team that is concentrated in one place. This affects productivity noticeably: Forty percent is rather the lower limit of what you gain in productivity if you bring your team together in one place. Although the threshold for working in a distributed fashion is about 30 m (cf. Kraut and Streeter 1995) between the team members, you have much better chances to gather your employees in one room if they only have to overcome a flight of stairs rather than continents! Of course you can also work with teams that are not collocated. You then have to accept a productivity loss.

Another factor might be the nationality of the project team members. The vast majority of people communicate faster, more efficiently, and in a more motivated fashion in their native rather than in a foreign language. If you have the opportunity, you should therefore look for a project that can very likely be conducted with developers speaking the same mother tongue.

If the project staff has already been defined, you should check their enthusiasm. Are they pirates who love to sail into uncharted waters, to make rich pickings? Then they will be thrilled to try out Scrum as a pilot team. Are your colleagues rather skeptical and prefer to work in their routine processes? Are they satisfied with all processes just as they are today? Then you have a difficult time ahead of you. Maybe you should spend more time looking for that right pilot. A pinch of Scrum experience does not hurt your project staff either.

Another factor is the complexity of the technology involved. Is it a project where you have to integrate hardware, server software, client software, databases, and possibly more? This is already difficult with experienced teams. Look for something simpler for your pilot project. The underlying technologies are also relevant here. Are the developers familiar with the programming language, classes, development environment, and subcontracted deliverables, or is it all new?

¹ Your teams are distributed, if complete teams sit at different locations. For example, team red sits in the USA and team blue works in Germany. Your team is dispersed if the team members of one team are scattered around various locations. E.g. Peter in Bulgaria, Marc and Steffi in Germany, Uta in the USA, and so on.

Such projects are ideal as a second pilot project for a team that has formed itself, enjoys working together, and is already well versed in Scrum. You should avoid them as your first test project.

Still, you have to face the question of whether the project is actually suitable for Scrum. Scrum was created for complex situations. This means situations with many uncertainties. If you are able to precisely plan your project sequence of events and can also fairly confidently assume that everything will happen as planned, then it is obvious that you should not choose Scrum. In such projects, Scrum can not play to its strengths.

The final point you should look out for, are the vested interests of your stakeholders. If management and customers already have certain preferences, you should consider these. This applies even if other criteria are no longer met. You absolutely need the support of these people. Negotiate with them—but if no agreement emerges, back down. Without the support of your stakeholders even the simplest project will fail.

Briefly summarized, an optimal pilot project is:

An important, national, technologically not too complex project, that starts in 1–3 months' time and does not take longer than 6 months. The staff is already collocated, allocated full-time to the project, and having the spirit of pirates open to new ideas. Also, your stakeholders should favor it as one of their priorities.

Often you do not have the luxury of being able to choose your first pilot project. In this case, use the knowledge from this book in order to perform a risk analysis and to remove some of the existing barriers. For example, merging your project staff in a single team space is always a worthwhile endeavor.

11.3.2 Setting Up a Pilot Project

You can increase pilot projects' likelihood of success considerably with the right preparation. Unfortunately, I often meet companies during my daily work that make absolutely no preparations—and then are surprised when the projects have teething troubles. Scrum is not a universal cure for your problems and Scrum does not function without anybody doing anything. Hard work is necessary. The preparation of a pilot project is primarily about reducing its complexity² as far as possible and about ensuring the team's ability to work. After selecting your project, you need to assemble your team. The Product Owner is usually obvious: It is the person who knows the product best from the customer's viewpoint. This is usually a product manager. This function already fulfills many of the duties of a Product Owner—e.g. recording requirements, talking to clients, developing product strategies, and so

² I use the term “complexity” in the context of “uncertainty”. The higher the complexity, the more uncertainty there is. As a consequence, the precision of planning decreases and risks increase. Since you already have enough process uncertainty when introducing Scrum, it is worthwhile to reduce other complexity factors.

on. What is usually missing is the close collaboration with the Development Team and the constant revision of the Product Backlog. If you cannot win the product manager for your project, you have to look for someone who can fulfill the task as well as possible—developers are usually not the best choice. Then go looking for your Scrum Master. Since you are obviously actively dealing with the matter, *you* might be the right choice. Otherwise you need a person who knows Scrum well and has good psychological skills. The more important the project is, the better your Scrum Master must be. Keep in mind that your Scrum Master is responsible for the productivity of the Development Team. He will also negotiate process improvements with management and other departments. If one of your developers feels drawn to this role, this is an option. Make sure, however, that no one individual fills both Developer-Product Owner or Developer-Scrum Master-roles. Even if they have all the necessary skills, conflicts of interest are inevitable. Imagine your project is under pressure: Do you want your Developer-Scrum Master dual role to join in with the programming tasks or solve the outstanding impediments? Do you want your Developer-Product Owner double hat to prepare the next Sprint or progress the code? You can avoid such awkward situations.

Once you have chosen your Product Owner and Scrum Master, you can start putting together your Development Team. If some people already have been chosen, you just need to include them in the process and involve them in the selection of the other developers. You should choose all additional team members very carefully. Make sure that all technical skills (programmers, testers, database specialists, software architects, hardware specialists, designers, requirements engineers. . .) relevant for your product are represented in your team. In particular the testers should not be neglected since you can only deliver finished software if it was well tested. A group of people first needs to grow together into a team. You can speed up this process if you involve the existing team members in the selection of new staff. This does not mean that the developers must make the decision. It does mean however that the opinion of the developers is taken serious. In addition to technical ability, soft skills have to be considered. The colleagues must be comfortable and able to work with each other. Conflicts are normal and allowed as long as they are carried out professionally. Situations where people do not want to work together at all are difficult. In a pilot project, you have neither the time nor the nerves to solve such cases. You should therefore look to include other individuals in your pilot team. Intimate personal relationships between colleagues are also difficult. Personal ups and downs have a direct impact on the project and multiply, as other team members can rarely avoid being directly involved.

As described in the previous chapter, you should definitely insist on a single mother tongue and availability at a single site. The full-time allocation for the project is also very important for your team's productivity. The most important point however, is that the people want to participate in a pilot project with Scrum. You will definitely encounter difficulties in whichever case you select. If your team truly wants to represent the change pilot, it will consider these problems as a positive challenge and will overcome them with ease. If your team feels forced, every obstacle is an ordeal.

Once you have your team together, conduct a kick-off workshop with all team members 2–4 weeks before the project starts. Have the Scrum Master and Product Owner participate as well. The goal of this kick-off is not the start of the project, but to get to know each other. Let the Product Owner briefly familiarize everybody with the business and functional content. All team members should introduce themselves. It has proven successful to visit a restaurant or bar together after the official part of the workshop has ended. One can informally mingle in such a relaxed atmosphere, which would not be possible in a work environment. A good Scrum Master will reach peak performance here and allow communication to “buzz” between everybody.

Besides the team preparation the environment must also fit. Take care of the project budget. Make sure that you have a sum for special unplanned expenses that you can spend on additional interior decoration, IT equipment, or even pizza. A couple of thousand Euros are sufficient here and can achieve a lot. If possible, you should also ensure that no other people in the organization have to sign off on this part of the budget. It is pathetic if an executive manager has to sign off a €100—pizza order or the controlling department subsequently cancels your order.

There were very inflexible processes in a large company. They did the best they could and were successful on the business side. However, particularly the approval and ordering process resembled a centipede. To celebrate a successful software release pizza lunch was promised to the developers. The costs were around €240,-. This promise was made by a member of lower management and was approved by the department head. When the promised day came, the controlling department ensured that there was no pizza. It was deemed too expensive, and it was criticized that no budget had been allocated for it at the beginning of the year. The result of this rejection was that the developers suddenly lost all confidence in their management. Motivation decreased rapidly and colleagues wasted considerable time getting excited about controlling and management. On that day, the company lost a hundred times the cost of pizza due to people not working and even more during the following months due to reduced motivation. Even years later management statements of any kind were eyed very critically. The solution would have been simple: Let the order get through this time and find a process to satisfy the needs of controlling in the future.

Next you will need to attend to the matter of premises. You achieve the highest level of productivity if your entire team including Scrum Master and Product Owner are sitting together in one room. Other teams should not be housed there. In addition to an area for meetings there should be a separate quiet corner in which individuals can retreat when they need to work on a specific thought-intensive task. Windows should be a matter of course. Large white boards on the walls and mobile pin boards ease working on difficult problems. Moderation materials such as pens and sticky notes should also be obtained in advance. The last tweaks are plants and other decorative items. The best you can do here is to send the Development Team with cash in hand to a hardware store. This way you can ensure that the whole team will accept and feel at home in their own new surroundings.

In addition to the room, you must also provide the right equipment needed to achieve the results you and the team will be striving for. You would not expect a carpenter to work without hammer and saw. Interestingly, many companies expect

their developers to manage without adequate tools. Professional equipment can sometimes be expensive. These costs are quickly put into perspective when you consider that an increase of 1 % in productivity in a single Development Team is equivalent to about €1,000³ per month. The same applies to the developer's workstations. Higher performance of these systems quickly reaches break even. For example, if more processing power of the computer ensures that the build completes 1 min faster, you can save about 100 min per month, assuming five builds per day. In addition, developers are usually in love with their technology. Their motivation increases when they are allowed to work with the latest hardware. Your goal has to be that your developers are proud of their work environment. If you are successful here, they will love it, enjoy working in it, and take care of it.

Your goal should be to allow all developers to work smoothly from the very first day in the project. This means that you need computers (Notebooks are state of the art as they also facilitate meetings, relocations, and communication with other departments), with sufficiently large screens, a fast network connection (including excellent internet access), an appropriate development environment (your developers know what they need), and an integration environment that enables the perpetual integration of the code (continuous integration). These software components have to be installed, but do not necessarily have to be fully configured.

A customer had not given any thought to the entry of new employees into the company. Thus it happened that new colleagues had to sign the order application form for their computer on their first working day. Once the computer finally arrived—which often enough took six weeks or longer—many of the necessary permissions were lacking. For each permission and every service a separate application form had to be handed in. Even e-mail accounts were not provided automatically. In the best case, it took a full week until a new employee was reasonably fit for work. More commonly, it took three to four weeks.

Another customer had well-defined welcome processes. Whenever a new employee came on board, he received a one-hour introduction to his computer on his first day. Of course the computer had already been procured, was set up, and fully tested. The biggest problem for new colleagues was usually to think of a new password. They could work productively from the first day on.

If you have the misfortune of having to work with distributed or dispersed teams, you also need very good conference phones, web cams for all employees, headsets for everybody, and a mobile high resolution network camera, which can be used to show billboards, walls, and rooms. In such contexts, you will also need communication software that allows all team members to contact (video, sound, and text) each other via a single click.⁴ In addition, programs that allow your employees to

³ In my experience a Development Team with seven developers costs around €100,000 per month. Of course, this depends very much on whether you are working with internal staff or foreign workers. In addition, the hourly rates vary strongly. However, this amount has proven to be quite accurate as a rule of thumb.

⁴ I personally used Skype/Lync, Teamviewer and Hangout so far. Other products will work as well of course. Important, however, is that the psychological barrier to actually use the software must be low. This is not the case if you need dial-in codes, register processes, or other wasteful steps.

share content of their screens are important. The release approval of source code, documents, and services must of course also be clear in advance.

Another dimension of the preparation is communication. On the one hand you need to ensure that all stakeholders know how the pilot project will run, when it starts, what it is trying to achieve, and so on. On the other hand you have to ensure that product requirements reach the Product Owner through a well-defined channel. In many cases it is enough to clearly state that demands can only be introduced through this person. If this is not made transparent, both preparation and implementation of the project will be harder. Help the Product Owner to define the product vision, the release goals, and the Sprint goals in advance of the project. Make all project participants aware of the fact that the project success can only be achieved if the project input fits the goal. The individual feature-level requirements do not necessarily have to be available at this time. This usually happens in the preparation phase or the first Sprint. These preparatory activities, commencing before the pilot is started, therefore deal with the general strategic view.

You now have a project, a team, premises, and work equipment. In addition, all stakeholders are prepared for what comes next. Your Product Owner knows at least roughly what he wants. So now it's time to get started!

11.3.3 Implementation

The implementation of a pilot project with Scrum is no different from other Scrum implementations. Whether a pilot or not, you will face similar difficulties. You should expect that your team will need three Sprints to find themselves and become productive. The more complex the overall context, the longer this “finding phase” will be. A good Scrum Master can slightly reduce the amount of time needed. Three Sprints are a good general rule of thumb. At the beginning a short Sprint duration—1–2 weeks—should be chosen. This gives you the opportunity to learn faster and react to problems more quickly. Opposed to an experienced team, you should not let a pilot team decide the Sprint length by themselves at the beginning. Indicate a Sprint duration and ask the developers to try it for two or three Sprints before they change it as desired.⁵ This way you can avoid useless discussions⁶ (not based on historical experience) and create a safe environment for the project participants. Stick to the steps that you have learned through this book. Convey urgency and

⁵ Trying out different Sprint lengths does not make sense in any case. Often, Scrum Masters or Scrum coaches have a fairly good impression of what would make sense for a new team. An inexperienced team might lack this foresight if they have never tried Scrum before. It is still important to involve the team, value their opinions, and facilitate its self-organization. This can mean that you have to work in a sub-optimal setting from time to time.

⁶ The discussion most often heard is that “four weeks are far too short to deliver something of value”. This discussion usually vanishes quickly once the team actually starts working in Scrum. Do not fall for it.

strategy. Make it clear to the pilot team why they are so important. Let them feel that they are special. Ensure maximum transparency.

Also, remember that a pilot team has yet to learn how Scrum works. Therefore, plan for some training during the first Sprints. The regular Scrum meetings will also take a little bit longer than with an experienced team.

11.3.3.1 Set Up Phase

The goal of the set up phase is to get ready to start on the operational level. This preparation phase is called “Sprint 0” by some authors. In the Scrum community there are heated debates about whether something like a Sprint 0 should be allowed. What is special about the preparation phase is that it does not require the team to create a usable Product Increment for the customer. This contradicts the purpose of a Sprint, because the core elements of every Sprint are “focus on the customer” and “delivery of a done Product Increment”. The product of this phase is the Scrum Team’s ability to work—not something usable by the customer. Therefore, the term “Sprint” is not correct and in the worst case can lead to a wrong understanding of Scrum. Thus, you should refrain from calling the preparation phase “Sprint 0”.

It is very important that this time span is kept as short as possible—and not longer than 4 weeks in all cases. Some people have a tendency to extend this period in order to develop “broad” or “complete” concepts. This misses the point of the phase. In the preparation phase five tasks need to be covered:

1. The development environment, including all servers, must be configured so that it can be used productively for the project.
2. The Product Owner needs to plan her first three Sprints, create her Product Backlog, and have it estimated.
3. The Development Team has to put some first thoughts to the software structure, important technologies, and architecture.
4. The Scrum Master must begin to form a team out of a bunch of individuals.
5. Everything else that is needed to achieve the working ability of the Scrum Team.

The default configuration requires a local development environment, which can also be used to build and test the software locally. Basically, every developer must always stay fully operational, even if all server systems fail. To ensure a minimal level of issues when code from all developers is integrated, it is necessary to be able to run as many tests as possible locally. Ensure that a high degree of test automation is targeted. In a traditional approach, you only have one testing period at the end of the implementation phase. The tests can be conducted manually here, since they hardly ever have to be repeated. This is different with Scrum, because here you have to deliver fully tested software every Sprint—which means every 1–4 weeks. This is not possible solely with manual testing. Therefore, the development systems need to provide the capability for frequent builds and automated testing. When multiple teams are working on the same product, it is also advisable to provide a multi-level server infrastructure. You can then use one system to develop and test the product, while on the next stage only fully executable and stable software is available. This

system can then serve as an acceptance and reference system. A similar effect can be achieved by an appropriate branching strategy. Whatever you do: Make sure that you have a fully integrated and error-free software version at your disposal at all times. If you pass these requests on to your developers, they will probably quickly configure their system accordingly and maintain it from that point on. For this reason it is not necessary to have the system configured in advance of the pilot. The risk that the pre-configured systems do not meet the Development Team's requirements is high. As a result, the system would then have to be reconfigured.

At the beginning of a new project, often the Product Owner only has a diffuse idea of what she wants to have implemented. Her job during the preparation phase is to elicit all initial requirements from her stakeholders. These have to be evaluated in terms of business value and captured in the Product Backlog. Of course, the Product Owner can request help from the Development Team to do this. The subsequent ordering of the backlog however cannot be delegated. Each element of the Product Backlog must be assigned a unique position in the sequential list. It is good practice to agree with the stakeholders on criteria, upon which the ordering can be derived. Such an approach makes the Product Owner's decisions objective and avoids conflicts. Once the first version of the Product Backlog is finished, the Development Team needs to estimate it. It is normal to have three or four Estimation Meetings⁷ during the preparation phase. If it has not yet been done, you also need to create release and Sprint goals for the project. The closer you are to the respective date, the more specific the goals must be formulated. This means that during the preparation phase it must be clearly defined what should be achieved during Sprint 1 both in terms of goal and features. A rough headline however is sufficient for Sprint 5. Only if Product Owner, stakeholders, and Development Team understand what they can expect during the upcoming weeks and months, has the Product Owner done her preparation phase work properly.

Once the Development Team knows what is expected of them at the functional level, they must start to think about the technologies to be utilized. They must also put some thought into the software architecture. Be careful not to define a complete architecture up front. I have never seen a team that actually succeeded defining it comprehensively. It is much more important and meets the purpose to define an architecture concept that is easily expandable. New features have to be easily implemented or removed—a bit like Lego bricks. In a good Scrum project the architecture grows and changes throughout the whole lifecycle of the product. You will also not be able to make all technology decisions in advance. Start with the most important questions and stop when the Sprint is over. The other questions will have to be clarified at a later time—which is usually easier, since you will have experience with the product by then. Some developers try to argue that without extensive preliminary work, decisions would be irreversible. Thus, all technical issues had to be addressed in advance. This is nonsense. For one, you will notice at some point that the world continued to spin and your initial ideas were wrong.

⁷ Compare with the chapter "Estimation" Meeting in the appendix.

Secondly, we are talking about software development and not about house construction. Of course you can expand the foundation of the software at any time, if you are willing to bear the expenses for it. To keep these expenses as low as possible, it is especially important to design your software modularly. Be careful not to skimp on the cost of modularity—otherwise you will bitterly regret this later.

The Scrum Master usually kicks off the preparation phase by conducting a meeting in which the project’s objectives, requirements, and participants are presented again. People get to know each other and agree on how to work together. Especially in the early days of a pilot project, the Scrum Master faces the risk and the opportunity to make a great impression on his team. If he facilitates with thought and solves pressing issues already at this early stage, he will win the respect of his team and will have a greater influence as a consequence. If he shows himself as being reluctant and incompetent, he will leave a lasting bad impression with his team. This way, he will have a hard time fulfilling his duties in the preparation phase: namely to begin to form a team out of individuals. Although this process is rarely completed after one Sprint, the course is set here. How exactly he tackles this task depends very much on the particular situation. It has proven helpful to strongly facilitate discussions, to make the rules (of the team, Scrum, and the company) transparent, to show his human side, and to spend time with colleagues after work—preferably at the three B’s—billiards, bowling, and beer. Before he can ensure that rules are respected, he must of course first create them together with the rest of the Scrum Team. Of particular importance is the “Definition of Done”.⁸ Once the rules are defined and transparent, the Scrum Master must ensure compliance with them until the team agrees on new ones.

It greatly depends on the particular project context as to which additional measures are required to ensure the Scrum Team’s ability to work. Very often, trainings in Scrum and technical fields are necessary. The preparation phase provides optimal conditions to carry out these trainings, since no concrete Product Increment needs to be delivered.

Often extensive purchases are necessary as well. Maybe it is just keyboards and network cables—but if these are lacking, you are facing a serious problem.

Not to be underestimated is the communication with all project stakeholders. The pilot project can only be a success if these stakeholders are fully informed and their expectations are realistic. Therefore, communicate as much as you can—and then some more. There are good reasons why you find hints about communication in almost every chapter of this book.

Analyze what you are still lacking for a successful start to development. Also ask your team and your stakeholders. Please do not delay taking the necessary actions, but instead build trust by taking them immediately and seriously, addressing everything possible.

⁸ You can find more about the Definition of Done in the appendix.

11.3.4 Sprint 1

In the first Sprint, your team has to create something potentially shippable for the first time. The result is delivered to the Product Owner. She will decide whether the release should be shipped to the customer. The delivered product must always be usable by the customer (or the Product Owner, who is the customer's agent). Source code, tests, or PowerPoint slides are not normally shown.⁹ The executable software is the measure for assessing whether the Product Backlog items have been finished or not.

It is normal that relatively little functionality is delivered in the first Sprint. Most often a lot of organizational issues still have to be addressed. In addition, more effort is needed now for software architecture and concepts than at later points in time. Nevertheless, it is imperative that finished features are produced. You can only be sure that your team has thought of everything if the software is doing what it should. Almost always inexperienced teams argue that they could not deliver anything after the first Sprint, because first the “foundations” had to be created. This opinion stems from a deep-rooted traditional way of working, where the architecture is usually defined upfront and cannot subsequently be changed. This approach does not apply to Scrum. On the contrary, the constant improvement of the architecture is part of daily work. Scrum also has recognized that pre-defined architectures rarely can do everything that is needed by the software, and always contain elements that are not needed at all. Therefore, do not accept such objections from your team. Encourage them to tell their Product Owner what functionality they can deliver in the first Sprint. A “Hello World” might be acceptable in the first Sprint if this proves that all architecture layers communicate correctly.

One of my teams—like many before—at first went to the barricades during Sprint Planning when I refused to only do architecture for four weeks. There were many arguments; the word “impossible” dominated the discussion. I then explained to the team in detail what costs were incurred by such a choice and what revenue we would miss through a single month's delay. Then I asked them how sure they were that the architecture would really be ready in a month? The answer was sobering: The highest estimate was 60 %. I then asked the developers, how confident they were that the architecture would be completed in 3 months. No response was above 80 %. At last I asked how confident they were that the architecture would be perfect enough in six months' time, so we would not have to work on it anymore. Only the less experienced colleagues, freshly out of college, offered estimates above 80 %. We then discussed how we could prove the functionality of the fundamental architectural concept. The result was that the smallest feature of the product should be implemented with a temporary GUI across all layers. This was done successfully during the first Sprint. The team had understood the principle and began to expand its architecture in accordance with the respective requirements Sprint by Sprint. Of course, often parts of the architecture had to be rewritten (“refactored”), but this was still cheaper than implementing a comprehensive architecture up front.

⁹There may be very specific situations in which the Product Owner only needs a document. In research teams, this is more often the case. This is similar with source code and tests: There are exceptions, but usually it comes down to executable software.

During the first Sprint all Sprint meetings are being conducted¹⁰ for the first time as well. Start with a review of the results of the preparation phase. Were all objectives achieved? What work is still open? Then do a Retrospective. How well did the collaboration work so far? What improvements should be implemented for Sprint 1? Do not, however, allow the team to change Scrum before they fully tried it out for at least three Sprints.

Continue with a Sprint Planning. In this planning meeting, the Development Team again receives an overview of the release and Sprint goals. The features to be worked on are discussed and planned out in detail. If it has not been done before, the Definition of Done also has to be created. Prior to that, any commitment of the team is obsolete because no one knows exactly what such a commitment means. In addition, a task board is created by the team. The best results can be achieved with paper on the walls—use this knowledge well!

In a management workshop I had brought my moderation case with many colorful Post-it® notes. The managers present ridiculed me for this and asked for “proper” tools. I asked them to try paper for the first tasks and they agreed. The work was very productive. After lunch the call for a digital tool got louder again. I advised against it, because with a digital tool only one person writes, while all other people watch him do the work. My advice was not followed and everything happened as I had suggested it would. After only a few minutes the executive manager spoke a word of power: We went back to Post-its®. While one colleague continued typing, the other participants productively worked in parallel with paper. From that day on, this company uses Post-its® in every workshop. Digital tools are filled afterwards.

The real Sprint work starts now. The developers begin to transform the requirements into executable software. The Product Owner refines her Product Backlog, answers developers’ questions and prepares the next Sprint in detail. The most difficult task, however, lies with the Scrum Master: He has to ensure that all developers overcome their initial difficulties and grow together as a team. It is quite normal that the process does not run perfectly smooth at the beginning. In part, this is due to interfaces to other departments that are still designed to accommodate the old processes. In addition, people are still insecure regarding the process. Both topics need to be addressed by the Scrum Master. Sprint 1 is usually also complicated by the fact that the developers are still locked in the “storming” phase of the team development lifecycle. This means that everyone is trying to carve out their place in the team, testing the limits, and fighting instead of avoiding conflicts. These conflicts have to be resolved in a constructive manner. The moderation of this process is the Scrum Master’s job. Through these productive confrontations the team creates rules that are accepted by everybody. In addition, confidence is built so that, even if sometimes sparks may fly high, people will not be personally offended. Do not try to suppress these conflicts, because otherwise they will fester forever beneath the surface. Instead, support their rapid solution.

¹⁰ Also see chapter “events” in the appendix.

In addition you should celebrate achievements at the end of the Sprint. Praise the Scrum Team, buy some pizza, or host a team event (e.g. team climbing, raft building, or the like). This welds the team even more closely together and quickly pays off in terms of higher motivation and a shorter team forming phase.

Particular attention should be paid to the Retrospective. This instrument is particularly valuable in the beginning of the transition, because the people involved still have an unprejudiced view while they start their journey from the old to the new processes. The results of the first Retrospectives also bring problems to the surface about which the team will later say, “This is just the way it is”.

11.3.5 Sprint n

I have only a little advice for you on the subsequent Sprints. The procedure is always the same, only the Sprint length may vary (but does not have to). Every Sprint a finished, executable Product Increment has to be delivered. Every Sprint a retrospective is used to identify process improvements. You will notice that the Development Team’s focus gradually moves away from problems within the team towards impediments in the organization. This is good and shows that the team has found itself. If you use the same care that you have used on Sprint 1 for the following Sprints as well, your chances for success are excellent. However, if for example you have an unprepared Product Owner or Scrum Master who does not solve impediments as needed, then your prospects dwindle.

Allow mistakes, but learn from them. Stay open to new ideas. Listen to other people when they give you advice. Do not put just any advice into practice, however—judge carefully whether the proposal really applies to your situation. Ensure maximum transparency. This avoids surprises for you and your stakeholders. It also enables others to constructively help you and learn from their mistakes.

11.4 Common Problems

Most of the problems that occur when implementing pilot projects with Scrum are caused by the lack of a seasoned expert on board. People read a book or attend a training and believe they could do without experienced coaches. They try to save money by only asking for a few days of expert consultancy. While this approach works, you have to be ready to make more mistakes and move forward more slowly than is the case when benefitting from intensive expert assistance.

In a small business the executives visited several Scrum trainings and read two books that had been recommended to them as a reference. They even booked an experienced coach for a few days of support. Then they tried to introduce Scrum. This was done in a very casual manner and led to some success as well as failures. Problems rose to the surface (to a large part due to the external coach highlighting them), but they were not faced up to by management. This led to a rejection of external expertise. They started to rant about

these “book writers” and stated internally and externally that the Scrum standards were just not applicable to the special circumstances of their company. As soon as a consultant had the “odor” of an expert, he was rejected. The real world was described as being just not as clear as the theory in the books.

Although this is of course true, the main rule was violated: Learn from your mistakes and solve problems when they occur. After living chaotic processes for several years, the company ceased to exist by now.

The content of books or trainings is also misunderstood quite often. Without practical experience only exceptional individuals are capable of viewing their behavioral patterns—that were acquired by years of practice and used more or less successfully in the past—from a bird’s eye perspective and change them. Pilot teams are often so busy with the new processes that they do not even get the idea to raise their antenna and actively look for problems. Many obstacles therefore only appear to the pilot team after several Sprints when the novelty has worn off. Let us take a closer look at some of the problems I encounter in my daily work over and over again:

1. The belief that Scrum solves all problems

Although virtually every introduction to the subject clearly indicates that Scrum does not solve problems, but only makes them transparent, there are still many people who think that the mere introduction of Scrum would solve all their problems at once. This is a false assumption. Scrum is like your mother in law¹¹: she knows very well that their son/daughter could have found a better match. But since she is a nice person she will remind you of all your weaknesses and mistakes every day. This hurts. But it allows you to address your weaknesses. Of course, your mother in law will immediately identify new opportunities for improvement, so you are constantly exposed to high levels of criticism.

Even if your own mother in law does not match this picture, Scrum does: It identifies the inadequacies of your organization, without solving the problems itself.

Sharpen the expectations of all stakeholders (including yourself) to that effect. Make sure that nobody believes Scrum to be a panacea. Also, everyone must be aware that Scrum is very labor intensive.

2. Wrong/nonexistent Product Owner

Too often, people try to save money by eliminating the Product Owner role. The duties of the Product Owner will then be taken over by the Scrum Master or a developer. This leads to role conflicts and inadequate preparation of the product requirements. If it is unclear what should be created, nothing useful will come out of it. These projects have to deal with serious problems in most cases, because lacking a Product Owner, or having a sub optimal one surfaces in new symptoms every day.

¹¹ This example was part of the February 2012 courseware of the Professional Scrum Master course by Scrum.org.

Make sure that all stakeholders know that the Product Owner basically has to fulfill the tasks of a product manager and must lead the Development Team on the functional content side. You can compare the absence of a good Product Owner with the absence of a GPS navigation system in a foreign city. Of course you can drive without one, but where and when you will arrive is written in the stars. In a pilot project, the incorrect fulfillment of this role is even more significant. Since the other project participants are busy with team forming and new processes, they cannot even make a serious attempt to compensate for the deficiencies of the Product Owner.

Select the right Product Owner.¹² Ensure his availability for the project. This person must want the project to succeed. Make him aware of the urgency and reinforce his intrinsic motivators.

3. Wrong/nonexistent Scrum Master

Similarly, sometimes too little emphasis is placed on the proper selection of the Scrum Master. After having read the above chapters, you know which tasks a Scrum Master is responsible for. You also know, that the profile required largely revolves around soft skills—programming knowledge is not essential. Companies often decide to use a developer for the Scrum Master role. In individual cases, this choice may well be correct, but in practice it is rarely examined whether the person has the required psychological and sociological skills. A good Scrum Master is especially important for a pilot project. He makes a significant contribution to the project's success or failure. With an experienced team, the Development Team and the Product Owner can compensate for a poor Scrum Master. In a pilot project you do not have an experienced team—and therefore they cannot compensate for lacking abilities. The availability of the Scrum Master also has a direct impact on project success. At the latest when your Scrum Master reacts to team problems with the words, “I cannot help you, I have more important things to do”, you know that you have made a mistake.

Ensure in a pilot project that an experienced Scrum Master is in place and available full-time.

4. Lack of Development Team skills

Lack of skills is present whenever either no one in the team knows about the topic, or if the knowledgeable team members are not available when they are needed. This is for example the case if the database specialist is only available for 20 % of her time. When you need her, she is not there. Unfortunately, it is often not possible to allocate all team members 100 % at the start of the pilot project.

First of all identify what skills at all are needed in the Development Team. This might lead to roles. Fill all roles with capable people. Wherever possible fill each role with two developers. If there are still some skills missing despite your careful consideration, build these skills through training. If you buy in

¹² For a few tips on how to do this, refer to the appendix.

external consultants you might indeed solve the issue in the short term, but your problem will not be solved and probably will haunt you long term. Explain the implications on team productivity if a person needs to work on multiple projects at the same time to everybody involved (20 % loss (cf. Weinberg 1991) for each additional project).

5. Lack of training

“Who needs Scrum training? We sent the Scrum Master to a two-day course, after all!”

These and similar ways of thinking lead to the necessary training taking place during the project work. Every company has to pay for the learning curve of their employees. There is no way around it. You decide at what point you want to bear the costs incurred and how transparent you want them to be. However, be aware that a lack of training will lead to your team making more mistakes than necessary in the pilot phase and may even allow wrong behaviors to creep in, which you need to iron out again later. In any case, it is cheaper and more effective to train the entire team together.

Also specialized training should not be used to save money. If a developer does not know what a unit test is or how to perform a boundary case analysis, test training is probably due. The same applies to other topics. Again, you must pay the costs—what you can decide is when, and with how much interest. Interest particularly arises in technical subjects by your developers building up technical debt (usually without bad intentions). The longer you wait to teach the skills necessary, the higher the debt mountain. The more time that passes from the point the debt was created to the day it is paid back the higher the interest rates. This is because the developer has to think his way into the old code and must also adapt connected code areas. The average cost to remove technical debt is four times¹³ of what the avoidance would have cost.

Especially with pilot projects you have to make sure you have all the necessary skills covered. Scrum demands more from developers than traditional approaches. Of course, not everyone can do everything. But everybody has to master the basics of good development work.¹⁴ Your employees will have a much harder time if they lack these basic skills.

6. Testing outside the team

In phase model project contexts, first the project is planned, then the code is implemented, and at the end everything is tested. This means that the expected completion date of the project is unclear until the last phase is completed. In addition, the bugs are only found at the end, causing significant additional costs. Sometimes even systemic errors are detected, which can no longer be

¹³ You do not believe this? Now consider how costly it is to fix a bug during development. Or during the test phase. Or once it has reached the customer. To do this, use the numbers of your own organization—this is better than any statistics.

¹⁴ Wikipedia gives a good overview over this topic: http://en.wikipedia.org/wiki/Software_craftsmanship

corrected at this late date. Nevertheless, this approach remains entrenched in people's minds. As a consequence, processes are established that implement the code in a pseudo-agile manner, but only validate it in a subsequent test phase (remember Water-Scrum-Fall). This gives away some of the benefits of Agile. In fact you no longer do Scrum, because this would require you to provide "done" software each Sprint. "Done" in every case means tested as well.

Solve this problem by integrating the test experts into your Development Team. Even if this means that you have to test twice: once during the Sprint and once in the subsequent official testing phase. While this is often a waste of time and money,¹⁵ it usually is process compliant enough, to allow you to get permission to test during the Sprints as well. During a pilot phase, you should achieve the maximum productivity within the Scrum framework—if you can prove at the same time that other measures are no longer necessary, all the better.

7. Too much focus on frameworks and architecture

As previously indicated, software architects and other developers often tend to initially prepare the entire architecture ("big design upfront"), before any other code is checked in. Unfortunately this approach is not effective in most projects, since the requirements for the product constantly change, and thus the architecture must constantly change as well. The result is sometimes a very long period in which no business value is being produced in terms of potentially shippable software. Even if the company is willing to take these costs into account, the result is usually not sufficient to cover all the requirements of the final application. If you are using agile methods, your software architecture must also meet the agile requirements. It must therefore be possible to constantly change and expand it, without having to change the entire existing code.

Only implement the very basic things in advance. Verify with your architects, that your architecture design allows constant changes. Try to only create as much of the architecture as necessary for the features of the current release to function properly. This is especially important for relatively short pilot projects. If you do not heed this advice you run the risk to only present an architecture concept and not working software at the end of the project.

8. Too little focus on frameworks and architecture

Some teams tend to abuse the term "agile" when tackling architecture. They state that, since they are agile now, they no longer need to consider architecture. This is not true and leads to chaotic structures, which fall apart after just a couple of Sprints. Especially when ancient technology and existing monolithic structures that cannot be replaced are involved, architecture plays an important role. You should of course not start on a several-month-journey to define a full

¹⁵ Caution: There are contexts in which this is very useful. For example it might be necessary in safety-critical contexts or for tests which the Development Team cannot do in one Sprint. Or would you send your developers to Scandinavia for vehicle testing every Sprint?

architecture concept upfront. However, you definitely have to look at your product vision and release goals and then conduct a technical risk assessment together with your Development Team. With a rough architecture outline, for example no more than a couple of boxes drawn on a whiteboard, you can still identify the risk areas requiring further investigation. Try to reduce the major risks by creating features, which then in turn prove that the issue is resolved. Also, as part of this approach, be sure to create architectural documentation for these areas.

9. Adjusting Scrum

Pay close attention if someone claims that certain elements of Scrum do not fit the context of their company. Hundreds of thousands of companies worldwide use Scrum. This includes automobile manufacturers, companies from the military sector, medical device developers, small businesses, as well as large corporations. If Scrum works for these organizations, why should it not work for you? It is hard for people to admit they were wrong. This also applies to the internal business processes—after all, there was a lot of effort involved to design and implement these in the past, and they helped drive success to a given extent. Listen very carefully to the Scrum critics. Generally, they provide valuable information and point you to a systemic problem of the organization that is uncovered by Scrum. Let us take a look at a common example: “In our company Daily Scrums do not work, because our developers are spread across three time zones. So we only do Daily Scrums once a week.”

The statement that the Daily Scrum is not working properly across three continents is perfectly correct. The conclusion, however, that Scrum has to be changed, is wrong. The real problem is that the team is globally dispersed. You will also find with a traditional product development approach that the communication between people does not really work in such a situation. Scrum did not change anything here—but it made the problem obvious.

Recognize proposals to change Scrum as valuable hints for your change efforts. Such wishes are especially voiced in pilot projects, where everything seems new and difficult. Listen closely, stick to your guns, and solve the underlying problems.

10. Flexible Sprint duration

Who has not seen this: Just before the end of the Sprint the developers come along and stoutly maintain that they could finish all remaining work in just one extra day. So why not move the Review by a day?

If you follow the proposal now, you have lost. “Game over”, so to say. The team’s velocity is no longer comparable, thus your medium-term planning no longer applies. The team will consider rules as interpretable—especially if they are of a temporal nature. In addition, you have no assurance whatsoever that the team will really finish everything within the requested time. Scrum is about pausing, inspecting where you are, and adapting your approach accordingly. Do not waste this advantage!

Pilot projects are about learning Scrum. Stick to the rules. Do not permit a single minute of extra time. Conduct the Sprint Review as planned and transfer

unfinished work into the next Sprint, if appropriate. Teach the Development Team that they are responsible for their own work.

11. Treating a pilot project as a line project

A pilot project involves greater risks than a project that has been carried out multiple times in similar form. This is especially true for a Scrum introduction because Scrum does things differently to the way project participants are accustomed to. It becomes dangerous if people expect the same level of productivity and forecast accuracy from a pilot project as from line projects. This is like comparing apples to oranges, which immediately puts you, as the person responsible for the pilot, into an awkward position. Communicate clearly that this is a pilot project and that predictions are correspondingly uncertain. Therefore make sure that all interested parties are kept sufficiently informed at all times. This reduces their anxiety and concern for potential failure.

12. Lack of communication

Communication is of immense importance—especially in pilot projects. First, the pilot project exists in the context of an organizational change project. Second, the pilot has its own goals, which have to be achieved. Scrum is new for all participants, so tension and uncertainty are correspondingly high. If you are lucky, the attention for your project is much larger than would be the case for normal projects. This attention needs to be fed with information. If you do not communicate enough, then both the pilot project and the change endeavor are at risk. You will lose the support of your stakeholders. Rumors start doing the rounds and the atmosphere heats up.

Do yourself a favor and proactively tell everyone about the project's progress and your actions. Additionally, make more specific project information (Product Backlog, team velocity, release goals, burndown charts, etc.) readily available, so that all those interested can view it at any time. This may mean not using programs or tools that are not readily available to all (e.g. because additional activation procedures are necessary), since this increases the threshold to really look at the information. Also, do not forget to always point to the urgency and the vision of your change project.

13. Inadequate communication of successes

A special case of communication in pilot projects relates to success stories. A pilot project has the objective to generate quick results and gather experience for the future. You should make these successes transparent. This helps your change efforts, the pilot team, as well as you personally. Even if you do not enjoy self-marketing, you will need to recognize that success needs to be made transparent in order for the organizational development initiative to function properly. All too often, pilot teams are successful, but no one knows about them. Do things well and talk about them!

14. Insisting on fixed-price contracts

In many companies purchasing and controlling departments in particular are accustomed to request fixed-price projects. Scrum is better suited for time and material contracts, since only relatively short-term forecasts of deliverables are

made and the employees are trusted to do a good job. Scrum is also aware of the fact that people under pressure (which occurs automatically in fixed-price contracts) start saving—especially on the product quality side. If you already have an established team working on a well-known product, fixed-price contracts are possible. Unfortunately, you usually have neither the first nor the latter in the context of a pilot project.

First try to convince those requesting the product of the benefits of a service contract model. If this fails, you cannot solve the dilemma by yourself. You have no choice but to plan in a traditional upfront way that helps you to create a fixed price quote. When the project is running, you can then plan in an agile manner. When doing so, however, make sure that two clauses are incorporated into the fixed-price contract (cf. Sutherland 2008). On the one hand, the risk for the client must be minimized, that is, he must have the right to withdraw from the contract at any time (while paying 20 % of the remaining order volume). On the other hand, it must be ensured that changes to requirements can be accommodated. Therefore you should include a clause stating that the client may introduce changes at any time; but if he does, he has to remove another requirement of the same size (or effort or cost) from the list of the features still to be implemented (i.e. from your Product Backlog). This way you create trust and can perhaps handle the next project without a fixed-price quote.

15. Inadequate premises

By using the right facilities you can increase productivity in a sustainable fashion. This of course works both ways: you can sustainably reduce productivity through your mistakes as well. Properly setting up a room is a science in itself and is beyond the scope of this book. For starters, it is sufficient to follow the basic rules, so you can avoid any negative consequences. Put the entire Scrum Team in an area dedicated solely to the team. This room must have windows. Tables and chairs must correspond to ergonomic standards. There should be enough space for quiet work as well as for meetings. The meeting space can be made comfortable thus creating a pleasant working atmosphere. The occupants themselves should select plants, pictures, and the like. Involve team members in other aspects of facility planning as well. If the team wants something in particular, chances are that this change will influence motivation and productivity positively. Seek out the conversation and actively listen.

16. Inadequate work equipment

Although providing the right equipment should be a matter of course, companies frequently equip their employees with outdated tools. This can go so far that some people even bring their personal equipment with them to the company a system administrator's nightmare. You would not expect a craftsman to operate with a hand drill, even though this tool was state of the art at some point in the past. What is the reason then why some developers are expected to work with obsolete computers? The workstation is the most important motivating factor for a developer. If it is outdated and slow, the employee will be dissatisfied. If it is fast and up to date, the developer usually is excited (it is amazing what a solid state disk drive can achieve). This pays off

not only in terms of higher motivation, but also in terms of increased development speed.

In addition to the workstation itself, it is also worth investing time and money in mice, keyboards, monitors, web cams, conference phones, headsets, and so on. The more dispersed your team, the better the communication infrastructure needs to be. If the voice quality is too poor, the developers will just stop talking to each other. Also remember to provide adequate development servers. What exactly you need depends on your specific project context.

In addition to the hardware you need to take a look at the software as well. Word, Excel, and PowerPoint, are as important as other business applications. Perhaps your employees also need Visio for process flow charting, a document management system, or a program for depicting relationships or dependencies in a visual manner. Ask your team what they need, and obtain the appropriate licenses.

17. The belief that everybody means well with you

Although this statement is not very popular, you should still be clear about it: Not everyone means well with you. For various reasons, there will be people working against you. Sometimes they are afraid of losing power. In other cases it is a private vendetta. Further still other situations may be due to their own career ambitions. There are many other reasons, but the result is always the same: They are working against you and will try to sabotage your pilot or diminish your achievements.

I firmly believe you can handle this in principle. Important, however, is to actively recognize these attacks. If you are provided your project team members late, for example, the provider will usually state reasons for this. These might be genuine or could be a façade. You must ask yourself the question whether he is working for or against you. Remember, your pilot has a very high visibility within the organization and is a critical success factor for your change effort. If the pilot fails, your Scrum introduction will be the next to crumble. Therefore, continuously ask yourself the question which of your colleagues can be counted on as supporters and who should rather be considered as opponents.

18. Successful teams are torn apart

A successful pilot project has two results. First, the product itself. Second, the successful Scrum Team. People are not resources. Forging a bunch of individuals into a team is no small achievement, and sometimes fails. With such a successful team, you are most likely to be able to accomplish more projects successfully. If you allow the team to be torn apart, however, the chances are that you cannot combine them again in the same constellation. This means that you will have to start again from scratch with your next project. Keep track of your (former) team members and enable the team to stay together. Ideally, immediately start the next project with this team. If you have to split the team, try to jointly allocate as many of the colleagues as possible to another project. This way you at least have a nucleus for your next successful Scrum Team.

19. Other problems

There are many other issues. I learn something new with every customer. If you have identified an obstacle, use your primary Scrum tool—your own common sense. Analyze the situation and sit down with your team. Apply what you have learned in this book. If nothing else helps, consult with an expert. Make sure you always learn from your mistakes, they are your greatest teacher.

11.5 Measurement of Results and Reporting

There are Scrum consultants who demonize any form of monitoring and reporting. This attitude, however, ignores the basic needs of any changing organization: to know where the road leads and where it is on this path. It is of paramount importance that you both set targets and communicate what you have already achieved. For pilot projects, it is advisable to set time, cost, quality and implementation goals (features). Scrum should make it relatively easy for you to keep the time targets—you work in a timeboxed¹⁶ fashion after all, which means the final deadline is fixed. Also, the cost targets should pose no problem, because your team is defined once and should not change thereafter, since this only leads to productivity decreases in the short-term. Check with your stakeholders regarding their expectations of product quality. Obtain reference values from past projects. The metrics “bugs within 3 months after delivery to the customer” and “total number of open bugs” have proven useful here. Once you know the expectations of your stakeholders, you can work together with your team to create a Definition of Done that ensures compliance with the quality objectives. The unknown factor is the number of features to be achieved. Again, you should try to reference values from the past, which can be difficult. A function point analysis is fairly objective, but also costly. Even this method does not take special circumstances into account. The comparability with your team is therefore difficult. Focus instead on the team itself. Rate how much business value is being delivered by your team each Sprint. This of course assumes that the Product Owner evaluated the Product Backlogs Items in terms of business value. This KPI (Key Performance Indicator) literally shows you how much you get for your money. A second useful metric is the measurement of the team velocity.¹⁷ You must measure this value anyway to create a release plan. Remember that velocity is not comparable between teams, only the velocity increase of an individual team should be monitored. This way productivity gains can very simply be determined. But beware of charlatans: It is common practice to take the first Sprint’s velocity as a base metric and to calculate productivity gains from there. This is a farce however, since there is too much uncertainty in the first Sprint and the team is busy with other things such as basic architecture decisions

¹⁶ You can find an explanation of “timebox” in the appendix.

¹⁷ See the appendix for more about velocity.

and interior design of the team room. In addition, experience shows that the velocity becomes stable after about three Sprints. Only then is the team well-tuned and the number is useful as a reference.

Be very careful with the use of KPIs. If a measure is not used in a value-neutral fashion, but instead viewed judgmentally, the measured value will improve but the intended purpose worsens (cf. Campbell 1976). An example illustrates this better: If you measure the number of defects, this is first of all value-neutral information. As soon as you start to evaluate this information and decide that your staff must produce fewer bugs in the future (maybe you even incentivize this), they will ensure that the measured number of bugs decreases. However, your colleagues from that point on will only have the measured number in mind, not the cause of the problems. They will find the easiest ways to improve the KPI. Maybe bugs are no longer registered in the corresponding system. Alternatively bugs could be closed without being solved. There are even cases in which the developers posted their phone numbers into the error messages so that the support calls did not arrive in the support department, but directly with the developers. The figure looks great then—but the intended benefits are not achieved.

Follow these basic rules:

1. Measure as little as possible, but as much as necessary. In most cases two or three measured values are enough to derive four to six figures from them.
2. Let the Development Team evaluate what a KPI means and what actions must be derived from this meaning.
3. Do not try to motivate your staff through indicators. If you are unlucky, this will work.
4. Velocity, business value, and the number of bugs are a good starting point for your performance measurement system in an agile project.
5. Make the measured values transparent for all project participants.

Transparency is particularly important. To measure success is not enough. Your stakeholders must be able to verify for themselves whether these achievements are really true. It has proven useful to not only make the metrics visible by hanging them up in the team room, but also to communicate them in every Sprint Review meeting. In many companies, there are also digital dashboards that can be viewed by all management representatives from their workstations. In some companies, certain types of reports are expected. This can include certain figures or an agreed format. Do not make your life unnecessarily difficult and follow these guidelines during the pilot project. If, in your opinion, the metric requested is meaningless do not communicate it actively, and rather only provide it in the requested report. You should particularly not bother your Development Team with unnecessary numbers. Later you can replace the existing measurement systems with an agile one. However, you must first create trust. A pilot project provides wonderful conditions for this to happen. Prove that your metrics are meaningful and provide the essential data!

11.6 Things You Should Remember

You will not be able to successfully implement change with the vision alone. You will also need short-term success. Here it is important to remember (cf. Kotter 2012, pp. 125; 127; 133):

1. The more real results you can generate in a short time, the better.
2. Aim for being able to show first results within 6 months or quicker.
3. Use short-term success to examine whether your strategy and vision are viable.
4. People get tired. Awaken them again by achieving and communicating success!
5. Short-term success can justify your efforts—and their costs—to management.
6. The more success you generate, the fewer critics will try to block your way.
7. Achievements must be visible.
8. Successes must be genuine and unique, i.e. they must not show any blemish and not be a repeat of earlier efforts.
9. Success must be understandable and verifiable for the employees.
10. Every short-term success must be clearly related to the vision and strategy of your change endeavor.
11. This connection must always be clearly communicated.
12. You need both leadership and management to create short-term success in the context of long-term visions.
13. Pilots are an excellent means to generate success.

References

- Campbell, D. T. (1976). *Assessing the impact of planned social change*. Occasional paper series. Reprinted with permission of The Public Affairs Center, Dartmouth College. December 1976, Paper #8, pp. 1–70.
- Kotter, J. (2012). *Leading change* (New ed.). Boston: Harvard Business Review.
- Kraut, R. E., & Streeter, L. A. (1995, March). Coordination in software development. *Communications of the ACM*, 38(3), 69–81.
- Sutherland, J. (2008). *Agile contracts: Money for nothing and your change for free*. 25.10.2008. jeffsutherland.com. 23.03.2014. <http://scrum.jeffsutherland.com/2008/10/agile-contracts-money-for-nothing-and.html>
- Weinberg, G. M. (1991). *Quality software management: Systems thinking*. New York: Dorset House.



Consolidate Gains and Initiate Further Change

12

You have achieved initial success in your change initiative. Employees and colleagues are informed and follow your lead. Everybody is aware of the urgency and strategy. You have the feeling that everything runs by itself. You are exhausted, since you have made tremendous efforts in the last weeks and months. It is time to rest. A little vacation would be spot on right now (cf. Kotter 2012, p. 139). The sun's heat, a secluded beach, the warm sea water—wonderful!

While this may be true for you, it would be poison for your change effort. To ensure long-term success, you need to consolidate the short-term gains and even intensify your efforts.

12.1 Promotions and Other Felonies

The problem with organizational change is that it depends considerably on leadership (cf. Kotter 2012, pp. 147–148). Even if you have a powerful guiding coalition, success still depends on a few key players, until the new approaches are rooted into the corporate culture. If these people drop out, your change effort comes to a halt. People soon return to their old ways of playing the game and most progress is lost. There are many reasons why this reversion back to the earlier situation takes place: from illness and exhaustion through retirement and termination to promotion—anything is possible and quite likely to happen. In the case of a Scrum introduction it can additionally happen that the pilot team is dissolved, thus destroying the agile nucleus. Let us consider the context: The head of a change initiative could be either experienced and a little older, a “silverback”,¹ or a young hunter, hungry for first or further success. The silverback has most likely led the company for a long time.

¹ Male gorillas become silverbacks when they are approximately 12 years old. The term comes from their characteristic back fur coloring. Usually this term is only used for the leader of the gorilla group.

Now he recognizes the signs of the times, and wants to ensure his legacy is ready for the future. His next action therefore is to initiate an appropriate change process. This process of change however, may very well take 10 years to complete—time the older colleague maybe cannot or does not want to spend. Time poses just such a problem for the hunter, as he sees the change as an opportunity to distinguish himself, and to qualify for a higher position. If he does his job well and proves successful, the organization is usually quick to offer a promotion. It requires a significant amount of self-control to turn down such an opportunity. Of course there are not only these two extreme forms, but every successful leader will at some point be faced with the decision whether to move up in, or out of the organization. Organizational change requires many years of the highest personal dedication. In addition, you have to reinforce your message over and over again—for some reason people tend to forget things, even things as important as your urgency and vision. The need for constant reinforcement may have a grueling affect on the members of the guiding coalition and in the worst case might cause a leader to throw in the towel and actually quit. Fatigue-related downtime and resignations are not uncommon. The most frequent reason for the failure of such a change project in this phase however is an exaggerated self-importance. This is caused because people misinterpret the first positive results after an exhausting first period to mean that the process can work by itself now. Facing this situation, the communication of urgency and vision is often reduced, which very quickly leads to a reduction in pace of change with it eventually coming to a halt. This deceleration is further influenced by the fact that you might actually want your employees to have a break—they have been through so much, after all—so you do not intervene when you notice reduced efforts (cf. *ibid.*, p. 139). If the urgency is lost in the course of this change deceleration, it must be laboriously built up again. This usually does not succeed however, because the effort's credibility was abandoned along with the urgency. Either something is important or not—a true urgency never follows an “on-off”, “on-off” approach.

For agile introduction projects another component is important as well: Large companies often have a large number of parallel projects, tended to by a single team. If you free up the team to work on a single specific agile pilot project, their productivity almost immediately soars. This in turn makes the members of this team very attractive for selection in other projects within the organization. As soon as the pilot project is completed, the team is ripped apart and must continue to work in the established traditional structures of other projects. The progress achieved in the agile environment, especially on a personal level, can therefore be lost very quickly.

12.2 This Is Only the Beginning

Do not falter!

Once you have created the first successes, celebrate them—in a way appropriate to the urgency (cf. Kotter 2012, p. 137). If the urgency for example includes financial problems, avoid an over lavish party. A few snacks and sincere thanks

are probably sufficient. Use the opportunity to highlight the urgency and to clearly demonstrate the strategy again.

Your main task during the consolidation phase is to use the results to improve the motivation of the project participants and to use this motivation in turn to promote the changes, or even to start new change initiatives in the context of your urgency. Simply said, do not falter but instead focus on accelerating (ibid., p. 150).

The short-term successes have validated your strategy. They are incontestable and transparent for everyone. Even your critics will have to admit that progress has been made. Your supporters are highly motivated. This is the ideal breeding ground for your strategy. For example, in the context of the introduction of profound Scrum, you will come into the situation that a pilot team successfully uses Scrum. The members of this team are highly motivated to go to work, while management is pleased with the increased productivity and transparency. Rather than sitting back and watching the team work, it is time to set up another pilot team and to establish a standardized process for doing so. At the same time, you can more easily discuss the larger systemic problems of the organization if this success is backing you up.

In the development department of a large company a Scrum pilot team had been set up successfully. The team members needed the usual three Sprints to find themselves. From that date on they were engaged, motivated, and worked particularly productively. Due to external factors, the company decided to cancel the pilot project and to distribute the team members across other projects that seemed more important. The developers were scattered over several non-Scrum projects. The guiding coalition had no alternative but to start over - with new pilot projects and new developers. They decided to set up several pilot projects in parallel for their second initiative. In this case if a single project was halted, the others could still go ahead remaining solidly intact.

Chances are that you will soon find yourself in an environment where you have to set up several change projects simultaneously (cf. ibid., p. 146). Of course, all of them are aimed to fulfill the strategy and address the urgency. You cannot do everything alone, however. It will show now, if you have really empowered your employees successfully on a broad basis. Only then you can rely on their support with them not collapsing under the weight of ongoing demands. No matter how good you are: The sum of your colleagues' achievements will be greater than the results you can achieve alone. Focus on your leadership and leave team management to your employees.² Look for issues which your employees can not solve by themselves. Here you can offer specific, targeted help with training programs or by solving systemic problems. Such systemic problems make themselves visible either when several people are needed to make a decision, by decisions taking a very long period of time to be made, or by people not making decisions despite the attendance at many meetings. Often, all these situations are visible in parallel. Critically question whether identified systemic dependencies (cf. Kotter 2012, pp. 140–145)

²This is not only true for Scrum contexts.

are really necessary, or whether the reason for their existence can only be justified by past imperatives.

In an automotive company the divisional and departmental heads could not freely decide on expenditure, even after receiving their approved budgets. At the beginning of every year, the budgets had to be requested, justified, and approved. Further still, once this hurdle was taken, every expense still had to be approved by the controlling department, regardless of the amount. This approach was appropriate during times of crises in order to avoid unnecessary costs. Beyond crises however, these structures greatly paralyzed the company and also served as a sign of significant distrust of the staff. They quite simply incapacitated the managers of this company. An examination of these dependencies was appropriate. A simple solution would have been to let management live out their responsibilities to the full and give them free reign over their budgets, as long as the company was not in crisis mode.

12.3 Things You Should Remember

If the first results catapult you into an ecstatic rapture, you should still keep a cool head. In particular, the following aspects are noteworthy (partly based on Kotter 2012, pp. 139–146; 150):

1. Organizational changes stand or fall with the people leading them.
2. If these people drop out, for example because of illness, exhaustion, retirement, resignation, or promotion, the change effort is seriously threatened.
3. The disintegration of entire teams, for example the pilot teams of your Scrum introduction, has an extremely negative impact on your change effort.
4. Look for silverbacks and hunters. Bind them to the company and therefore to the change process.
5. Keep your arrogance in check and always keep your gaze firmly locked on the urgency.
6. Use the successes achieved to push for changes or to even start new change initiatives in the context of your urgency.
7. In the context of your urgency you will inevitably face several change projects that have to be carried out in parallel.
8. You have to rely on the support of your employees to cope with the large number of tasks. Hopefully you have empowered them already.
9. Solve problems, especially systemic ones that highlight unnecessary dependencies in your organization.

Reference

Kotter, J. (2012). *Leading change* (New ed.). Boston: Harvard Business Review.



Anchor New Approaches into the Corporate Culture 13

Once you have achieved success and your change process is in full swing, you face the task of making your work become sustainable. Only if you manage to firmly embed the new values, standards, and approaches into its culture, can the organization be so deeply affected that even your departure would not cause the changes achieved to be reverted (cf. Kotter 2012, p. 153).

We had successfully introduced Scrum with a client and saw good results. However, management struggled with their fate, because they could now exercise less direct control and the Development Teams had more responsibility. We decided at that time not to pay attention to their struggle. We believed that management would follow us as soon as they saw positive results.

Unfortunately, this assumption was wrong. Although the criticism became quieter, it continued in secrecy. The Scrum implementation project was successfully completed after about a year. Six months later, the organization had fallen back into their old patterns. Although Scrum was still officially being practiced, it was heavily distorted and made absurd through numerous monitoring instruments. Change had again become a mortal enemy. What had happened? We had failed to anchor the new approaches into the corporate culture. Victory had been declared too soon and changed into defeat within a short time. Through the absence of the coach the guiding coalition's driver was missing. These coalition members did not even notice that the original corporate culture had prevailed again.

13.1 Origins of Culture

You have read a lot about culture in the first part of this book. After having read how to introduce Scrum, it makes sense to look at some aspects again.

Corporate culture is something hardly tangible. It takes place on different levels (cf. Schein 2010). The visible behaviors, artifacts, rituals, etc. are just the tip of the iceberg. These are the manifestation of culture, so to speak. These visible characteristics are driven by a sense of doing the right thing. This feeling arises

from the sum of the individual values and personal characteristics of all people constituting the organization. Everybody has her own character, a concrete idea of the relationship to other people, and her own time and activity orientation (cf. Schein 2010). The collective culture arises because similar characteristics are combined and lead to success.

Once a culture has developed, consciously or unconsciously employees that comply with this culture are being facilitated. Disturbing the culture is a no go. Promotions and hiring criteria are then considerably dependent on whether an individual fits the corporate culture (cf. Kotter 2012, pp. 156–159).

At a job interview (I had applied for a job) in a large company, we were sitting together until late after 6 pm. The interview had been planned to last two hours, but right afterwards I was also presented to the entire department, and in a third pass the area head talked to me as well. At this time it had already taken five hours and I gradually lost the desire for additional talks. Three circumstances showed me that this corporate culture was not suitable for me: First, I was expected to waive all other appointments I might have had so I could fulfill the wishes of my superiors. After all, it would not have been a problem to come back another day—instead I had to not only stick around, but also the whole department had to cease all work and stay late in order to meet their boss's wish to evaluate me. Second, the interviewer immediately started the conversation by testing how I respond in a stress situation. This was done by immediately criticizing the knot in my tie and asking me who had taught me to tie it. Both he and I knew that such an observation was subjective criticism, not something that could be verified during an interview. It was only a test to see whether I could stay calm and confident in an awkward situation. A completely unnecessary measure, as an interview in itself causes a sufficient amount of stress anyway. The fact that he nevertheless found it necessary to put further pressure on me showed very clearly that a culture of pressure prevailed in the department as well. Third, my interviewers made a joke at the expense of a colleague during the interview: As he left the workplace around 6 pm, the question was thrown into the room, why the person was leaving so early? Because he had a part-time job, of course. Not the joke as such left an impression, but the seriousness with which it was presented and well received by everybody struck me. I had learned a lot about the culture of this company that day: Permanent and artificial stress, absolute subordination of ones private life to the interests of superiors, and overtime until exhaustion—this did not meet my expectations of a desirable workplace. Three offers of this company followed, of which I declined all. In my opinion, I would not have fit well into this culture, and vice versa.

Because the elements of culture—for example, doing overtime every day—largely influence the criteria for hiring and promotions, soon all employees exercise these “required” behaviors. It is not a single person that provides for the establishment of culture, but the sum of all employees. Hundreds or even thousands of employees live the common values and act accordingly. This is all the more remarkable since the culture is not written down anywhere. While there are guidelines, standards, and other artifacts that have emerged from the corporate culture, the collective values cannot easily be derived from them. This works only in selective cases. Take a common example: Employees are always 15 min late to meetings. So it seems to be part of the corporate culture to be late. What does this say about the common values? There are a variety of options from which I'd like to pick out just two. It could be that in this organization the basic assumption exists that one must first really finish one thing before starting a new one. This should then

also lead to a high product quality and a lower than average level of hustle in the organization. On the other hand it could also be the case that there is an unspoken agreement that the self-importance is underscored by the amount of time one can allow himself to be late. In this case, you could probably observe that the colleagues arrive in the order of their rank, the highest in the hierarchy arriving last. While the observable behavior of delayed arrival alone does not allow you to draw conclusions regarding the underlying values without further indicators, it usually is a meaningful observation.

To understand corporate cultures it is important to know that these are only cemented by joint success. Only if the application of a cultural element has contributed to an improvement in the company's situation, will it become standard. If it is already part of the standard, it is validated and reinforced by success. Unfortunately it does not work the other way around: If the company fails in some aspect, usually no conclusions are drawn that point to the culture. Since it largely exists unconsciously, it is accepted without criticism and not changed. At the most some artifacts such as guidelines are adapted. As the corporate culture continues to exist unchanged, the new guidelines reflect the spirit of the existing culture as well. A real change does not take place.

13.2 Anchoring

To change an existing corporate culture, it must be replaced with a new one. You probably can already guess what this means for you: hundreds, if not thousands of people have to adjust their personal value systems. Precisely because culture is something very diffuse and subconscious, you cannot bring this change about with a few words. Even the deliberate choice of a particular practice is usually not viable—it is too easy to fall back into old patterns. People are emotional creatures. This means you need to appeal to their hearts and minds (cf. Kotter 2012, pp. 160–161). You can only achieve this through a vision that feeds on a corresponding urgency. This vision must be supported by a viable strategy and a series of successes. You have got to know the exact steps in the previous chapters. In addition, there are a few distinctive features that should be observed.

Analyze the present corporate culture and make it transparent to everyone involved (i.e. this might well mean the whole company) how this culture developed. There was a time when this culture was absolutely necessary and helpful; otherwise it would not have developed. Clearly display these circumstances. You can now complete this information by adding today's relevant practices and explaining their advantages (ibid., p. 163). Generate successes and prove them with the new cultural elements. For example, you can use the first deliverables from your Scrum Team. Ensure that only those employees are promoted who clearly live the new culture. Employees who oppose the new behavior patterns may not, in any case at all, be promoted. This is true regardless of other merits and seniority, because that would transport a completely false picture: The old values are still paramount, demanding changes are unnecessary. Also, be careful when hiring new employees. Only select those who fit the new corporate culture. This is a particularly difficult task, since the

old values and practices have been deeply entrenched into us and also act unconsciously. It can help to create an interview questionnaire in advance or to even hire a service provider to support the process. Once you start with such considerations, you should also think about the succession of your key players. This includes all members of the guiding coalition, and especially your CEO. No one has as much influence on the development of your corporate culture as the Managing Director. While he cannot guarantee the success of your change efforts, he most certainly can let them fail. It is therefore important to choose successors wisely and to begin early with the transition. Only those who live the new values from their hearts are suitable successors (*ibid.*).

In connection with Scrum introductions the Scrum values especially should be noted: commitment, openness, respect, focus, and courage as well as teamwork, trust, collaboration, transparency, honesty, communication, and the readiness to make mistakes and learn from them are at the forefront. These values must be established and practiced. Visible characteristics of the culture, such as policies and procedures must be focused on these core values. Ask yourself these questions: Are all activities transparent in your organization? Are you allowed to make mistakes? Who gets ahead—seemingly immaculate winners, or transparent misfits who learn from their mistakes?

Scrum also puts people at the center of all action. Are you still buying in resources, or have you already realized that it is actually people who work at your company? What do you do for your employees¹ in addition to paying them? Have you ever verified whether work results are better after 8 h of work rather than ten? Presumably you would be surprised by the results. Always remember—you can lead people, but you can only procure resources.

13.3 Things You Should Remember

To ensure that your efforts were not in vain, you have to anchor the changes sustainably within the organization. This only works through the corporate culture. Therefore in everything you do you must strive to reinforce the new behavior. You should especially note the following points (cf. Kotter 2012, pp. 160–161; 163):

1. Corporate culture is something hardly tangible. It takes place at different levels (cf. Schein 2010).
2. The visible manifestations of culture are driven by a sense of doing the right thing.
3. This feeling arises from the sum of the personal values and characteristics of all individuals involved.

¹ André Häusling and Boris Gloger wrote a German book (Gloger and Häusling 2011), which I can recommend in regards to this topic.

4. Past achievements have shaped the corporate culture. Each culture was at least once upon a time yielding a return for those who lived in and with it.
5. Make it clear how the current culture was created and point to the advantages of the new paradigm.
6. The sum of all employees forms the culture, not a single person.
7. Since culture is so difficult to grasp, it is rarely identified as the cause for failures.
8. To change an existing corporate culture, it must be replaced with a new one.
9. People are emotional creatures. Therefore, you must appeal to both heart and mind.
10. Align recruitment and promotion processes to the new values. Only someone who represents these is allowed to advance.
11. Also adjust the succession of your key players in terms of whether the candidates live for and are ignited by the new values.
12. Scrum puts people at the center. In addition, it demands commitment, openness, respect, focus, and courage as well as teamwork, trust, collaboration, transparency, honesty, communication, and the willingness to make mistakes and learn from them.
13. All the previous steps described in this book are essential in order to change the organizational culture. If you omit one, you will most likely fail.

References

- Gloger, B., & Häusling, A. (2011). *Erfolgreich mit Scrum – Einflussfaktor Personalmanagement*. München: Carl Hanser.
- Kotter, J. (2012). *Leading change* (New ed.). Boston: Harvard Business Review.
- Schein, E. H. (2010). *Organizational culture and leadership* (4th ed.). San Francisco: Jossey-Bass.



Time and time again, there are situations in which a very large project team or even an entire company is looking to transition to Scrum. Quite often, these teams are not only large, but also scattered around the globe. A Scrum introduction in such an environment is extremely demanding for the change agent and all those involved in change management. You need to know the specifics of such an endeavor and coordinate across all the departments and staff, whether they are involved or just impacted. Mistakes made in this respect will immediately multiply themselves.

14.1 Special Circumstances

The larger a team, the greater the amount of communication needed. Two people for example have one channel of communication: from A to B. With three people you increase to three channels: from A to B, from B to C, and from A to C. Twenty team members face a significantly higher number of 190 channels of communication, while 100 people have to contend with an overwhelming 4,950 channels! This is no longer manageable. You will have probably already noticed that a group of people starts to break into sub-teams once it reaches eight to ten people. Although they might still officially be referred to as one team, they will in fact work in other groupings. If it is not managed, this division into sub-groups usually happens chaotically and leads to people looking for professional affinity, i.e. people with similar professional interests working together. This means in software projects, you will find software developers who are experts on the same application layer working together. The database experts keep to themselves, the server developers join forces, and the user interface gurus work separated from the rest. These groups of people can relate to and understand each other. Some companies have even aligned their organizational structures according to this principle, in the hope that the interfaces between departments can be managed more easily than the sum of all people as such. Unfortunately, this is not efficient either: not even a calendar full of meetings would cover the amount of communication required. Mistakes and

misunderstandings cannot be avoided. The rude awakening usually comes when all software components are being integrated. Blaming each other rather than working towards a solution then dominates the interaction.

Your main task in the context of large teams is to shape the relationships between people in such a way that they can efficiently work together. Another difficulty is that more people have political or other interests in this large group of people, than is the case with a small team. Your stakeholder circle has simply grown significantly. All parties want to be informed and your task of gaining buy-in requires much more effort. In addition, everyone of course has their own goals and interests that must be satisfied.

The larger the organization, the more cumbersome it usually becomes. The employees have settled into their comfort zones and feel at home there. Therefore, you should expect to have to part company with more people during a Scrum introduction in large groups than in small ones. This is because the shift towards openness and transparency can be very painful for some individuals. Partly because the change itself is exhausting. Partly because individual shortcomings of some people become clearly visible. In such contexts, it is also possible that you do not want to have some employees on your team. This relates to both developers and managers who cannot or do not want to identify with the new values. You must first try to win these people for your project. Actively involve them in, or at least communicate to them, the work the guiding coalition is driving. Warmly recommend the new urgency and advantages of the approach to them. Ask for their support. Help these colleagues to get along with the new realities. If all else fails, you must exclude the person from the transformation of the group for now and try again later. If this is not possible, it is best to help the person understand that other career alternatives could better suit them—the current one will not lead to happiness anyway, and once the change begins it will progress rapidly.

Another unique feature of introducing Scrum into large teams is, that with increasing team size the dependence of global corporate business processes increases significantly. While a team of seven people can still manoeuvre around the standard processes, this is no longer possible with a 100 developers. On the contrary, you are probably even glad to be able to rely on a few fundamental processes. Unfortunately, you will not like all of them. There are always courses of action that are inappropriate for agile approaches. You can try, together with your guiding coalition, to transform these unsuitable processes. It is more likely, however, that you have to live with them in the beginning and must first undoubtedly prove that improvement is needed. While this is wasting productivity (and therefore money), this course of action is often necessary to gain support. You can describe organizational development with an analogy from seafaring: you cannot stop an ocean tanker easily and rebuild it on the open sea. You can make small improvements and pull it in a different direction with a tug, but this all takes time. If you take it into a dry dock, the ship can be rebuilt but does not earn any money for its owners. The same applies to your organization: if you paralyze it, no money is earned. This approach would also cut off the very revenue source that is obviously the basis for your own salary. Instead choose an incremental approach,

which keeps your organization intact and operating at the same time as making change in an iterative fashion.

14.2 Basic Rules for Scaling

“Scaling” describes the application of multiple rather than of single teams. Most literature talks about “the team” and transports the notion of a single team working on a product. My preference is to talk about “large teams”. In practice, this effectively means many teams working in parallel to develop a single product, a so-called “scaled environment”. To ease your work, I will provide you with a couple of basic rules for scaling. These are no substitute for specialized literature, and are intended to help with taking the first steps.

The scaling rule number one is: Don’t do it! Very few people use more than two or three teams and remain successful. You can achieve a lot of movement with many people, but movement is not the same as progress. Most companies increase their costs linearly when scaling, but effectively reduce their output, because the additional coordination and management effort devours the increase in available capacity.

The second rule of scaling is: Whatever you have before you scale, you will have more of it after scaling. If you have rubbish when you start, you will have a huge pile of rubbish when you are done. Therefore the first step has to be establishing a well tuned single team. Only when your team is working at this level, should you think about adding a second one. Only when both teams are working like a charm, should you consider a third one, etc. Ideas like “let’s add five teams to the first one” usually don’t work well.

If you have to scale, don’t scale up beyond 10 teams working on the same product. Up to 10 teams, well-established approaches like Nexus or LeSS work quite well. Beyond that number, this isn’t the case and you won’t find a working agile blueprint. You will have to develop your own solution. In any case, you will need an additional steering instrument to coordinate more than 10 teams. By doing this, you will also most likely exceed your cell size (cf. Sect. 1.4) of 150 people, which in turn drastically reduces agility in your organization.

The last basic rule is: Know everything, follow nothing. This means you need to know all popular scaling models to make qualified decisions—but you should not follow any of them blindly. Right now, the most popular models are Nexus (scrum.org), LeSS (Larman & Vodde), SAFe (Leffingwell) and Scrum@Scale (Sutherland). Nexus and LeSS (Large-Scale Scrum) follow similar revolutionary approaches and were derived from practical experience of the authors. Scrum@Scale is a meta framework, listing different models for different situations. You can view it as a method tool box for experiments. These three approaches follow agile principles from my point of view and are suitable for daily use. SAFe (Scaled Agile Framework) keeps many roles and processes from the traditional world and only uses Scrum on the team level. It was created with a strong RUP (Rational Unified Process) mindset and—from my personal point of view—is not

really agile when fully put into practice. The charming aspect of it is the evolutionary approach, making it more appealing to big corporations because it is less painful to start with. It can be used as a “starter drug” for these enterprises. Unfortunately, I know neither an organization that put SAFe fully into practice, nor one that has used it for more than 4 years.

All the models discussed above offer interesting ideas and methods that can be very useful, depending on your context. In order to judge this correctly, you need to be familiar with all approaches. You can then apply the aspects that are most appropriate for your situation. Beware of people who tell you to follow “that book” or “this method” and claim that other options do not contain any useful ideas for you.

14.3 Direct Comparison of Small and Large Introductions

You have already learned a lot in the previous chapters about how to initiate and manage a change process. It all applies to large teams as well. Let us compare a Scrum introduction, with simpler steps in a small business, seven developer context, to an introduction in a larger context with 3,000 developers. Let us call these companies “Small Ltd.” and “Big Inc.”.

The CEO of Small Ltd. has the idea to introduce Scrum. The reason is that the competition is delivering new desired functionality to the customer more rapidly and at a lower price, and this will clearly pose a major problem in the long run (urgency). The managing director grabs the head of product development and the product manager, and discusses alternatives with them. A plan results that describes the target state (vision) and how Scrum should be introduced (strategy). The three of them (guiding coalition) walk up to the Development Team and explain the situation (communication). The developers are then trained in Scrum and other required topics (empowering employees on a broad basis) and an external coach is brought on board. The first goal is to ship a new release in 3 months—which always took a year so far (achieve quick success). They succeed, and both sales and product management are enthusiastically re-organized to also follow the Scrum values (consolidate gains and initiate further changes). The personnel representative develops a concept that probes to what extent agile values are lived and shared among employees and new applicants. Hiring and promotions are made dependent on these values (new approaches are anchored into the organizational culture).

You can see that introducing Scrum in a small context is relatively simple. Although it is not enough to simply get started, following Kotter’s eight steps for organizational change will lead you relatively quickly into the success zone. This is not as easy with Big Inc. In this example, I take the same conditions for Big Inc., as I did for Small Ltd., for granted. Although this is unrealistic, it does make the examples more easily comparable.

The managing director of Big Inc. has the idea to introduce Scrum. The reason is that the competition is delivering new desired functionality to the customer more frequently and at a lower price, and this will clearly pose a problem in the long run

(urgency). She grabs her heads of development and product management and discusses the alternatives. Together, the issue is discussed in the executive committee. A 2-day workshop to analyze the problem and to create a vision is carried out (vision). Some people are assigned who have to attend to the issue (guiding coalition). These people involve an outside expert, develop a concept, and present this to the executive committee. With a few changes, it is accepted (strategy). At a staff meeting urgency, vision, and strategy are communicated to all employees. After that the internal communications department takes care of further information distribution (communication). Together with the appropriate managers, a first pilot project is chosen to be transitioned to Scrum. Employees who will work on this project are provided. These employees receive training and an external coach is hired to support them (empowering employees). For everybody not directly involved, information sessions and Scrum crash courses are provided. The pilot project runs over a period of 6 months. While it is successful (achieve quick success), a large number of systemic problems were identified. These are processed by the guiding coalition and discussed in the executive committee. The problems found are selectively solved (initiate further changes). To validate the findings, another pilot project is launched. It confirms the initial observations (consolidate gains). Further changes are approved. Every change must be extensively communicated to all employees. Training and workshops are needed to keep everyone on the change path. Finally, after 2 years, it is decided to establish a standardized process for Scrum projects. Every project head now has the freedom to choose between traditional and agile approaches. Agile training is included in the regular education program of the organization. Statistics are gathered from all projects to monitor success and satisfaction rates. After an additional 3 years the decision is made to only permit traditional approaches in exceptional cases. These exemptions are well defined and communicated. Every 6 months there are workshops in which the agile values are developed and taught. Job advertisements include a sentence that makes it clear that transparency and openness are fundamental values of this company. This is validated in assessment centers. These criteria are included in the annual performance appraisals¹ as well, which must then be assessed by the supervisors. The extent to which executives live up to the agile values is reviewed as part of an annual 360-degree feedback. Promotions are only possible if the reviews are positive (new approaches are anchored into the organizational culture).

This example is greatly simplified. In practice, you will encounter many more problems and must do much more in terms of communication. In particular, staff involvement in the change endeavor requires much more effort. Nevertheless, even this short example shows that in large organizations, change takes place in several stages and is more difficult than in smaller contexts. Furthermore it is clear that the same steps are necessary, regardless of the size of the change effort—although with

¹ A truly agile enterprise will not conduct performance reviews on an annual basis. Feedback is expected in short cycles, preferably daily.

varying characteristics. The case study in this book shows a little more closely what needs to be done to sustainably introduce Scrum.

14.4 Coordination

If you transition a large group to Scrum, your most difficult task is coordination. In addition to the change itself, which is described throughout this book, it is also about the coordination of the different interest groups.

Split your projects by products. It is important that your teams can work as independently as possible on these products and still produce something saleable for the company. A framework team, a server team, and a user interface team are a poor alternative because the company only earns money when the “products” of these groups are integrated with one another. You are better off organizing your developers in teams that combine all of the required skills to deliver the final product. No team has to rely on another in order to deliver. To stay with the example above, this means putting framework, server, and user interface experts into every team. The team size should be chosen so that the developers can, on the one hand, work without problems due to lack of skillset, and on the other hand do not disintegrate into sub-teams. This is usually the case with a team size of three to nine² people. Scrum Master and Product Owner do not count into this number. Usually no special precautions are necessary up to a size of three Scrum Teams. Teams talk with each other by themselves. Dependencies are not so big that you continuously step on each other’s toes. If you have more teams, it gets more involved. Consider each product for itself if you have managed to define independent products. Since the different products do not depend on each other, only a small need for coordination exists. The need for coordination is immense, however, if you did not succeed in this separation and have in fact only one huge product. For every three to five teams, you need a daily coordination meeting, the “Scrum of Scrums”.³ In this round, team representatives discuss what they plan to accomplish by the following day, in which code areas they are working, and how common problems should be solved. Once you have more than five teams, you need to think about extending the Scrum of Scrums and introduce a “Scrum of Scrum of Scrums”. Be careful though: It very much depends on your specific situation, if this is necessary and a good idea. The enlargement only works if your employees are disciplined and experienced, that is they know code and processes well. The Scrum of Scrum of Scrums is a discussion round in which one representative from each Scrum of Scrums explains the essence of the respective Scrum of Scrums. Decisions are usually not made, but an optimal flow of information back to the

² Depending on the specific task and situation at hand your team may already decay from seven people. Try it out and involve your developers in the decision making process.

³ Depending on the chosen scaling model, this coordination meeting works differently. “Scrum of Scrums” is a very general term, originating from the early days of Scrum.

teams needs to be ensured. This meeting overhead⁴ is unfortunately necessary because otherwise miscommunication leads to problems, such as software bugs and lacking productivity. The more teams you have, the bigger your overhead. I highly recommend you to only use three to five teams to work on one product, and never more than ten. Invest time in the separation of your products, so that the teams can work independently. The productivity gains with each additional team get smaller and smaller until the costs finally outweigh the benefits. You are in many cases more productive if you reduce the total number of developers involved in one product. Let the other employees work on another product instead.

In addition to your developers, your Scrum Masters and Product Owners also need to synchronize. You should only have one Product Owner for a single product. He will be overwhelmed in large projects, however, and needs support. In many cases, organizations resort to proxy Product Owners.⁵ Alternatively, a Chief Product Owner can be used, who works with the other Product Owners. For a large monolithic product this will not work, because the Product Owners then have no decision rights and in the end only act as proxies. It is better to keep your Product Owners solely responsible and to support them with requirement engineers or tools that reduce their workload. No matter which model applies to you, your Product Owners must gather information daily, so that the total product progress can be managed and is transparent to everyone. This works the same way as the Scrum of Scrums and can be called “Scrum of Product Owners”⁶ (“Scrum of Scrum Masters” for the Scrum Master). Moreover, it may be necessary that certain specialists in the teams have an additional coordination need. This is often the case with architects and user interface developers. These then come together in specialized rounds to discuss important topics at hand. This does not always happen every day, and not always throughout the entire project lifetime. In effect, the Development Teams must determine for themselves what coordination needs they have and how to meet them adequately. Always remember to provide a Scrum Master for moderation purposes—this makes meetings significantly more efficient and helps everyone to focus.

It may be useful at the beginning of a large Scrum introduction to impose the same processes on all teams. This includes tools and documentation. By doing this, complexity is reduced and all start from a common and transparent baseline. Once your teams are experienced, this is different however. It is normal that every team uses slightly different tools and processes. It is also normal that not all teams in an organization do Scrum. Maintenance teams prefer Kanban for example and simple projects can also effectively be solved using traditional approaches. You should

⁴“Overhead” refers to expenses for administrative or other tasks that have nothing to do with the actual product development. Every company should strive to minimize these costs as such.

⁵ A proxy Product Owner is an information relay of the Product Owner. He does not have decision-making powers and only forwards information coming from the top. Some information is lost or distorted. This is not a desirable state and should be avoided.

⁶ Again, I am relying on old terms here in order not to favor any specific scaling approach. Depending on your choice, you will find other terms like “community of practice” or “guild”.

banish undefined processes from your portfolio. If the process stays undefined, transparency is lost and optimization cannot commence. Rest assured that Scrum includes enough leeway for every creative mind. The cornerstones⁷ of Scrum are, on the other hand, immovable.

14.5 The Right Time

When is the right time for introducing Scrum into a large setting? This question is easy to answer: once you have done your homework in terms of the change process and gathered extensive experience with single-team Scrum.

The homework is complete when urgency, vision, and strategy are clear and being constantly communicated. A guiding coalition must take care of the change. You must be clear about how you want to empower your employees to support the change and become part of a Scrum project. Initial successes in terms of completed pilot projects have to prove beyond doubt that Scrum is good for everybody. In these pilots, you can also build up your Scrum expertise.

If just one of these criteria is not fully met, you should try it on a small scale first. Otherwise the probability of failure is very high. A significant point is that you will probably not immediately fail, but rather after 1 or 2 years of effort. Prior to that your employees will “play along”, even if they were not entirely convinced at the outset.

14.6 Things You Should Remember

The process sequence of introducing Scrum in large contexts does not significantly differ from introducing it in small projects. However, the effort needed and the exact specifics of the individual steps are fundamentally different. If you have to deal with organizing a large number of people using Scrum as your operating process, you should consider these points:

1. Large groups fall apart into sub-teams. This usually happens once the group exceeds seven to nine people.
2. The larger the context of your Scrum introduction, the larger your circle of stakeholders.
3. Avoid scaling if possible and never scale beyond ten teams per product.
4. After scaling you have more of what you had before. A little muck grows into a huge heap of muck.
5. Know all popular scaling approaches and follow none blindly.

⁷The Scrum guide defines the rules of Scrum. In these 17 pages (July 2016) are the cornerstones (Schwaber and Sutherland 2016) defined.

6. Some people will quit, especially in large companies. This is due to the pain caused by leaving their comfort zones and due to the transparency generated by Scrum.
7. The more people involved, the greater is your dependence on global business processes.
8. Divide your projects by products.
9. Each team must be able to work without the others. This excludes the possibility that one team can only work on specific layers of the software.
10. A Development Team consists of three to nine people. Adding Scrum Master and Product Owner to the mix completes the Scrum Team.
11. If you have three teams or less, you generally do not need special measures for their coordination.
12. The more teams you have, the more overhead is required.
13. Use a Scrum of Scrums to coordinate multiple teams. Analogous constructs work for multiple teams (e.g. Scrum of Scrum of Scrums), the Scrum Masters (e.g. Scrum of Scrum Masters), and the Product Owners (e.g. Scrum of Product Owners).
14. Use the same process in all novice teams at the beginning. Allow experienced teams to freely choose their own processes. Make sure, however, that the basic principles of Scrum are adhered to, if the team wants to use Scrum.
15. Only introduce Scrum into many teams if you have done your homework in terms of the change process. You also need extensive experience with Scrum to succeed here.

Reference

Schwaber, K. & Sutherland, J. (2016, July). *The scrum guide*. Scrum.org. 28.08.2017. <http://scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf#zoom=100>

Part IV
Case Study



The following case study is drawn from real life, although it is no accurate reproduction of any single one of my projects. It is included here to make the often abstract theory outlined in this book more tangible and accessible. Depending on the circumstances you, the reader, are facing, certain aspects of it might seem exaggerated or unrealistic. However, you might also see yourself reflected in other elements of it. This case study is presented in a way that conforms to the eight steps of Kotter (cf. Kotter 2012).

Should you find it difficult to keep track of the many protagonists in the following story, simply turn to Sect. 15.11 for a brief list of all the main players.

15.1 The Hardest Part: Starting Out

Peter is the head of development at a large company. He has long been fighting with the problem that his team of 100 developers need more and more time to deliver fewer and fewer functions. Product releases keep getting postponed, and quality had long seen better days. His developers are spending 80 % of their precious time not on productive work, such as developing new functions, but stuck in meetings, fixing bugs, and coming up to speed. On average there are two projects—meaning two products—for every single developer. Although the company has gotten by quite well for the last few years, the competition is not asleep: For the last three quarters, there has been a steady trickle of customers going elsewhere. In this situation, Peter took the time to check the figures and predict their future trends. All things being equal, the business will be in major trouble in a mere 2 years. He takes this troubling news to Stephen, his CEO and a well-known “we can do it” decision maker, who immediately calls the entire executive team in and asks Peter to repeat what he had just told him. The effect: embarrassed silence all round. If truth be told, everybody had known about the situation, but nobody dared mention it out loud. Now that the cat is out of the bag, what would happen? Stephen takes to the stage and asks his

colleagues to check the situation in their individual areas and to collect all relevant data. He wants to make sure whether there is indeed a crisis. He wants proof.

Seven days pass, and the executive team meets again. It turns out that the situation is even more dire than assumed. Sarah, the head of controlling, reveals that clients are not only leaving the company in droves, as Peter had noticed, but that the remaining clients apparently need to be “kept happy” with greater and greater discounts and goodwill payments. She too confirms that the business will become unsustainable within 2 years. This is when Frederick from the legal office steps in: “It won’t even take that long. We’ve been at the receiving end of more and more litigations for the last 3 months. Our software is making our customers lose money. And we’re more often than not liable to make up for their losses. This is costing us 20 million a month in legal costs, settlements, and penalties. If we add that to our data, we have a maximum of 18 months before we stop being a force in our markets.” Anne, representing the HR department, speaks up as well: “People, we’ve been fixated with money for the last hour. Remember, we have human beings working here—and they are getting more and more dissatisfied all the time. Staff turnover has reached 15 % already, and it was only 8 % a year ago.”

Stephen is shocked by the scope of the problems. He demands suggestions.

“Step up the pressure! We need to get rid of developers who introduce bugs in their code.”

“More people. We just need more people.”

“Outsourcing software development is best. To India. They have excellent training over there.”

“Cut the bonuses and freeze the salary. That’s the only way we can consolidate our finances.”

“We need to scupper the legacy code and start programming from scratch. Else we won’t ever get rid of the core of the problem!”

This is the time when Peter breaks his silence: “Guys, pressure and less money in their pockets won’t keep our people on board, let alone improve the quality of our products. All that does is create a climate of fear. Adding more people might help in the long run, but the problem cannot wait. It might take a year for them to be trained up and ready—we haven’t got that sort of time. Going to India is also only a long term option, which in addition would mean handing off our core competency—developing software—to another company. Our fate would then be in their hands. No, I say, no—we mustn’t look to our developers for blame. We need to take a long hard look at ourselves. If we screwed up, we need to accept the consequences.”

Stephen takes a minute to think: “Peter, you’re right. The buck stops here. We won’t get rid of our problems by getting rid of our people. Let’s be constructive.”

The debate starts up and dies down again without any real outcome. A decision is taken to take the weekend to think about possible solutions. The team is asked to bear the following principles in mind for that: Customer satisfaction, staff motivation, product quality, productivity, and the stability of the software.

At 7 a.m. on Monday morning, the team reassembles, with everybody looking slightly frayed at the edges for lack of sleep. Everybody grabs a coffee, and the discussion starts again. Every executive has one or two ideas to contribute, which

are written down on slips of paper and pinned to the wall. Peter alone is surprisingly quiet. He looks particularly tired, with dark rings around his eyes and his third cup of coffee in his hands. When Anne points this out, he seems to wake up again and mentions that he had spent the entire weekend with Marc, one of his top developers. Marc had amazing things to contribute. It turned out that he knows the processes and models very well indeed. He actually named all of the issues off the cuff that Peter and his colleagues needed the entire crisis meeting to identify. What came out was a review of the software development process, a gap analysis, and a list of specific improvement suggestions.

Everybody is suitably impressed, and Marc is immediately requested to join the meetings—after all, his work was most productive. Anne states that she found out that most people are actually quite satisfied and motivated to actively contribute to the future wellbeing of the business. Although Marc, who had only ever seen the executives as distant bosses, is somewhat uncertain at first, his caution soon disappears when he can present his analysis. After his presentation of the gap analysis, Marc points to a word underlined in red with visible embarrassment. “Antiquated processes—this is how my grandma programmed code!” it says. Marc apologizes: “I’m sorry, I didn’t have the time to change that. But it’s like that: We’re really not moving with the times anymore. We are still working in a way that stopped being successful in the 1980s.”

Stephen looks at Peter: “Is that true?”

Peter gives a slight nod. Yes, it is true. He never noticed before, but there are solutions for their problems that he had for some reason missed. In essence, the old phase-by-phase model is not suitable for the type of complex development work the company is doing. Today, such projects are usually approached with the so-called “agile” methods. With the exception of Marc, nobody at the company has any experience with the term. Stephen turns to Marc: “So, can we make our processes agile?”¹

Marc scratches his head, takes a minute to think, and finally says: “The most popular version of these agile methods is Scrum. Introducing Scrum is not that hard at the beginning. You need to assign a few new roles and keep track of some regular meetings. You can read all of that in the Scrum Guide. But that won’t solve our problems. Scrum just shows us what the problems are, but it does not make them go away. The hard part is overcoming the flaws in the system. We need to dig deep into our business for that.”

Debates flare up all around Marc. Stephen has to work hard to get people focused again. He asks Marc to give everybody a brief introduction to Scrum the next day. The other people are asked to read this “Scrum Guide”, and Peter is asked to get a list of consultants who could help introduce Scrum, if that is indeed the way forward.

¹ In reality, one would have to check in detail whether agile methods—and Scrum in particular—are the right fit for the given situation.

The brief introduction eventually takes 2 h to complete. There are limits to the understanding of what it can do for the group, as Marc simply cannot answer certain questions. He knows the developers' perspective very well, but the management's point of view is simply unknown and also unusual for him. Despite these limitations, the executive team recognizes the potential of Scrum and decides to buy the services of an external consultant for a "real" introduction. Peter seems wary: "Which consultant should we go with? There's thousands of them out there. All of them are saying that they can do Scrum, and all of them are offering introduction courses. Which criteria should we use to help us decide?"

With Marc's help, the team settles on a list of criteria that the consultant has to fulfill: He or she has to have at least 3 years experience with Scrum. There should be meaningful reference projects in the area, and he or she should know how change processes work in business. The consultant should also have more indirect proof of his or her abilities: publications, forum postings, or anything of that nature. In the end, Marc adds another point: "Before I forget: When we introduced Scrum at my last employer, we first picked the wrong consultant. He stayed for 2 weeks, said that the introduction was done, and then upped sticks and left. That only left us with a massive headache, because we kept finding problems that we then had to solve on our own. So we decided to book a consultant for a longer-term project. On a daily basis to start with, and then for 2 days a week after a few months. He only moved on to the next project when we knew we were happy with it."

This is turned into the criterion: "Does not only spend 2 weeks on the easy part, but stays for the hard part."

With these criteria to guide him, Peter makes a shortlist and sends an email with the details about their situation to the ten most promising candidates. He closes the email with a simple question: How would the consultant solve the problem? The result is disappointing: Two emails are never answered; five are answered with a standard "Everything will be fine once we introduce Scrum."

The remaining three consultants offer more detailed answers. They email or phone the company to ask more about the situation. One of the contenders says that he would be intrigued by the project, but would not have time for it, as he is still committed to another assignment. The other two consultants meet Peter to discuss their approach in person. To him, they seem very competent, and they distinguish between the company's short and long-term success. Peter decides to ask both of them to give separate, 1-day introduction courses, telling both contenders openly that there are two people in the running. Neither seems surprised and both offer their services for the workshop initially for free, or only to be billed if not selected for the actual project.

The introductions go well, and both consultants can answer all of their potential client's questions. The second consultant has a far tougher job, as the participants' questions are much more sophisticated, after having already completed the first workshop. Nonetheless, he seems more appealing on a personal level, although nobody can give the exact reasons. After the offers from both consultants are checked in detail, the executive team indeed decides to work with the second consultant.

Scheduling is more difficult, but some creative calendar work means that the project can already start a week later.

The coach begins by explaining the different target shapes, such as Scrum PRN or the Scrum Software Studio. He reviews Kotter's eight steps and makes the group understand how much work there is ahead of them. The first day ends with two seemingly simple assignments: Find a name for the project group, and work on the first of Kotter's steps, creating a sense of urgency, before the next session a week later. Stephen sums up the proceedings: "So, we're the guiding coalition. There's more than one shape of Scrum, and every concept has its advantages and disadvantages. We need to take many decisions and complete a lot of work in the time ahead of us. Well, at least we're already on the way with this sense of urgency! Now, what shall we call ourselves?"

The guiding coalition slowly begins to realize that the identified problems can indeed be solved. There is a real sense of motivation and energy in the room, and the participants suddenly come up with dozens of possible names. In the end, they all agree: The guiding coalition will be known as the "Emergency Power Team".

15.2 Urgency

When the "Emergency Power Team" meets again a week later, all participants come in with a set of documents and their laptops. They have collected mountains of data and masses of facts. The coach is excited by this commitment and calls for a search for the root causes:

"If there is an urgent reason why you're not doing your normal jobs today, I need to hear it now."

One by one, the answers are coming in.

"We're as good as broke."

"Our product quality is poor."

"Our customers are leaving."

"We're wasting millions on litigation".

Stephen sees where the coach is trying to go and intervenes: "People, all of these reasons are correct, but what you all are showing us is your own, personal version of urgency. We cannot communicate that. We need a single, coherent message for all of our employees."

The team begins its analysis. It does not take long for them to agree that everybody wants to introduce Scrum. Marc adds that he is sure that the developers would be on board as well. No important manager has been left out, although the managers on the ground have not yet been involved. The coach reminds the Emergency Power Team that it is these line managers that will be most affected by Scrum and might lose power as well as authority as a result. This means resistance, unless they too see the benefits. Stephen makes a note on the whiteboard, before asking his colleagues: "Okay, we know that we want Scrum. We also know who might not want it. But what do we want to change with Scrum?"

“That all depends on the situation at your company, and on the Scrum shape that you are choosing,” the coach explains. “I can tell you something about our situation.” Sarah comes in with her presentation, loaded with data. “Our competitors are getting to market faster than us, and doing it with a better product. Our customers know this and are shifting over to them. In response, we pile on the pressure. All of that constant pressure means that our good people are leaving. We need to get out of this vicious circle, or we will be insolvent in 18 months.” Everybody is shocked by this plain and simple truth. The catastrophe has to be averted! The group begins to work on what is needed next: a concept to make the crisis transparent and understandable for everybody. People are also unsure about what introducing Scrum will mean for the company as a whole and the people in it. The coach tries to help by again explaining the various Scrum shapes. Although everybody listens intently, Stephen says out loud what they are thinking: “That is all well and good, but we need a solution, fast. We don’t have the time to turn our organization upside down. We’re also just too big for that. What we need now is to pull ourselves out of this mess by our bootstraps. Then we can start thinking about where we want to go.”

The coach agrees. “We should start with the most critical project, get it back on track, and then work on the other projects. So, what we are talking about in essence is a top-down Scrum PRN introduction. We still have an option for a Scrum Software Studio or even Profound Scrum. We simply have to make sure that we don’t lose what we learn from our PRN projects, or else we are doomed to repeat our mistakes over and over again.”

The critical project is first given a new structure, adding the roles of Product Owner and Scrum Master, but removing the project manager’s role in essence. It is immediately obvious that this will have major implications: Only Claudia, a product manager, is a suitable choice as Product Owner, since she solely knows what the market expects from the product. However, this also means that she alone is responsible for the entire product, and she alone can and must take decisions about the product. This used to be her supervisor’s responsibility, shared with a number of other high-profile executives. These are now relegated to roles as “stakeholders”, instead of decision-makers. A Scrum Master is also required for the project. Marc would be available, but what happens with the current project manager? He knows little to nothing about Scrum and he prefers an authoritarian leadership style that does not fit the bill for Scrum. What role is left for him? That question remains unresolved for the moment.

The coach makes it clear to the people in his charge that the Development Team will still run into problems: Up to now, decisions were mostly taken by management. Scrum forces people to take charge of the entire technical responsibilities. Indeed: Scrum demands that the team organizes itself—and most team members do not know how this could or would happen. In their private lives, they might be building and buying houses or raising children, but at work, they have spent years and decades letting others organize their lives. Making such a change can easily overwhelm some people. To make matters worse, Scrum does not include any team leaders and even treats architects as regular developers—an affront to many of the

people in such high-status roles. Their competences are still in great demand, but the Development Team has no place for any hierarchies or signs of status.

The Emergency Power Team considers these challenges, but reaches no clear result. Peter steps in and says: “We have 20 developers and two team leaders working on our most critical project. Why don’t we just go and speak to them?”

“Shouldn’t we first finish our urgency statement?”, Anne retorts. “If we start speaking about how we want to introduce Scrum without actually explaining why we need to do so, we won’t get people on board.”

All nod in agreement and get back to the task. A hard day’s work later, and the results are in:

Our company is facing the most severe crisis of its history. Our competitors are getting to market faster with a better product than us. Our customers know this and are leaving us in droves. Our people are unhappy and quitting their jobs. This vicious circle means that we will have to file for insolvency protection in 18 months, unless we find a way out of it. We, the Emergency Power Team, are working hard to avoid this disaster. We need your support.

The team meets again the next morning and invites the two team leaders of the most important product to join them. Marc notices that his fellow developers are not present. Aren’t they what this is all about? When he mentions this, he is told: “Our developers are under pressure. We don’t have time to invest whole days for workshops. As long as we, the team leaders, are here, we can take all necessary decisions.”

Stephen takes to the stage: “Gentlemen, our company is facing a grave crisis. It does not matter whether your teams spend another day with the old ways. Please go and get your colleagues. I’ll explain later.”

With a worried look on their faces, the two team leaders leave the room and return with their developers in tow. Stephen points to the urgency statement on the board: “Ladies and gentlemen, these are serious issues. If we continue to work like this, we’ll be out of business in 18 months. And you’ll be out of work. None of us want that. That’s why you are here. We have analyzed the situation and we are sure that we’ll come out of it stronger and better than ever before with Scrum on our side. We want you to understand what Scrum will change for you. Your opinion matters, because we still have other options.”

The statement leads into a lengthy discussion with lots of questions going back and forth. Will there be redundancies? Will people get their Christmas bonuses? What does Scrum mean? The coach gives everybody a high-speed introduction into the topic, focusing on the opportunities and risks. He keeps the big shock for last: “Scrum means no more team leaders, because the teams will manage themselves. And all team members are equally important.”

The team leaders and team members react in very different ways: Some developers are relieved and happy about being given an official opportunity to take the decisions that they had been taking unofficially all along. Others are worried what this “self-organizing” might entail. The team leader’s responses are the most ambivalent. While one of them is excited that he would finally “only do things that really matter” as Scrum Master—if that will indeed become his role—

his opposite number gets very emotional: “My job matters. You can’t simply plan me away! No team can organize itself. You haven’t heard the end of this!”

This is the point at which Stephen intervenes: “No threats, please. We value your work a lot. We need to understand which other activities you might find interesting, even in a Scrum setup. We always need good developers for example. And maybe you would like to become Scrum Master. Of course, you can also stay team leader, but of a different team. For the moment, all we are speaking about is organizing 20 people and their jobs differently in line with Scrum principles.”

This calms the team leader down again. He likes the idea of returning to actual development work, which he had been missing for the last few years. “Give me time to think about it”, he says, with a much more conciliatory tone in his voice.

The coach then mentions the basic rule of Scrum: “We’ve got 20 developers. Scrum works with teams of no more than nine people. Should we start with two separate teams?”

This is again the object of much heated debate, which goes on until one of the developers steps up and reminds the group that everybody had been working on at least two, often three projects at the same time. If all the work actually invested into the critical project were added up, it would mean a maximum of eight people working on it. The problem had been that the product managers could access the developers directly—which Scrum would change. Not everybody would have to be at the product managers’ mercy if eight people were indeed assigned to the project full time (instead of 20 in differing degrees of part time). Claudia nods slowly: “If this means that other, just as capable people have more time to give to other products, then we product managers could happily give up individual people or our direct access rights. Solving this crisis matters more.”

The assembled developers are asked to read up on Scrum and to choose a team of eight people. The same assignment is given to the team leaders, although they should propose their own team structures independent from the developers’ suggestion. Both the developers and the team leaders are also asked to think about where they could see themselves in a Scrum-based organization.

Stephen turns to the guiding coalition again: “Now, how can we get the message about our crisis to the other employees? We’ve seen here today that people have lots of questions that deserve proper answers. And we’ve seen that they will have concerns and anxieties that we need to respond to. What people are expecting are solutions about how to overcome this crisis. Do you have suggestions for a communication concept?”

This is Anne’s turn to shine: “That’s a normal job for us in HR. Luckily, we have hierarchical structures to work with. I would propose that we first invite all managers to an info event. After that, we could call a general works assembly and tell our other people. People should have an opportunity to ask their questions to their supervisors, or to us in the Emergency Power Team. We should also sit down with marketing and prepare some posters, articles in our in-house journal, or some emails. There is no one-size-fits-all answer for our problems—we need to be clear about that.”

The suggestion is carried unanimously. Stephen adds: “We are in the midst of a crisis, and we should behave like that. Sarah, you make sure that we stop all unnecessary spending. That starts with us here: From today on, we do not spend money unless absolutely necessary. We have to lead by example. That means: no more pointless printouts, second class fares for business trips, and cheaper hotels. And it means testing all ideas and projects to see whether they actually contribute value. That should have been done long ago.”

15.3 The Guiding Coalition

The staff assembly takes place only days later. At the same time, the PR department produced a press release to stop any unfounded rumors from filtering out into the press. The guiding coalition meets in the cafeteria, where the coach raises another issue: “We have done great work. We’ve done most of it right. But one issue still leaves me concerned: We ourselves, as the Emergency Power Team. Is everybody needed on the team? We are also not actually following Scrum principles ourselves, but we want to solve problems with Scrum. There is lots of work ahead of us. Let’s get to work first thing tomorrow.”

Frederick opens the proceedings on the next morning: “I thought about whether I belong here on the Emergency Power Team. I love working with you, but I think I have reached the end of my usefulness. I could help you with the urgency issue, but I don’t think I can do more than help you from time to time when it comes to introducing Scrum.”

The group takes a moment to think about his decision, and then thanks Frederick for the work he has done. The coach reminds the group that there will be Sprint Reviews of the first team soon and that Frederick should attend them in any case. His opinions and his expertise are important for the team. Anne starts fidgeting, and slowly begins to say something: “Well, to be honest, HR has not all that much to do with software development either. Is there anything I can contribute still?”

Peter immediately intervenes: “Anne, without your instinct, and your experience, we would never have managed to communicate our sense of urgency. I think you’re an absolute asset for the team. We want you on board, definitely!”

The soft murmur of agreement gets louder. The coach gets up and writes a first point on the flipchart: “Hierarchical power”. He looks at the group and says: “It seems as if we have the CEO and all department managers on board. From where I am standing, the guiding coalition has the authority and power it needs.”

He then writes down the second point: “Expert in the current processes.” When he asks who this might be, there is an embarrassed silence. “Okay, so we might need somebody else still. Who could that be?”

Stephen seems unsure: “Our processes are Christina’s job, normally. Why isn’t she here with us?”

Thomas, who has kept quiet so far, speaks up: “Christina reports to me as head of quality management. I have been keeping her informed about our meetings, but I

didn't think it necessary for her to be involved. But you might be right: She should be here."

A short phone call later, and Christina joins the group. After a brief run-down of what the Emergency Power Team is doing, she seems passionate about getting involved. "I've always thought that we need to change something. Count me in!"

The coach turns to the flipchart and writes down a third point: "A Scrum veteran".

He explains: "Right now, this is my role, because I live Scrum every day. But in future, somebody of your own staff needs to take over. Marc. What about you?"

Marc agrees: "I've spent the last few weeks thinking about this a lot. I am sure that I could become a good Scrum Master. I want to change things—so I might be the right person for the job."

Stephen and Peter nod in unison. The coach continues with the point "All people who could support or hinder the introduction of Scrum". This point leads to considerable discussions: Who can offer support? Who will stand in the way? Has everybody been considered? At the end of the discussion, two more people have been identified: Hannah, the head of IT operations, and Larry, a software architect of long standing who is respected by all developers, but known to be generally critical. Stephen calls both of them. Hannah arrives soon later and listens to the explanations of the others. Larry, however, does not come. Stephen calls him again and lets the others listen in. Larry is explicitly and categorically against getting involved. Neither praise for his competences in terms of the planned changes nor another description of the negative state of the company can change his mind. Everybody seems very upset about his reaction.

Marc scratches his beard: "I think we need to sit down with him. We've always gotten along well. I like him as a colleague. Maybe I can find out more about his reasons."

The coach adds the issue of credibility to the flipchart. At first, the group tries to claim the point as being fulfilled already, but the coach remains persistent: If credibility is as good as people are saying, how can Larry say that processes do not matter? How credible is it if there is this one person forming his own little fiefdom in the organization? Are there any other points, where credibility might not be as high as it seems? The team ends the meeting with the task to go and look for other indications that speak for or against the presumed high credibility.

The team meets again 2 days later, but this time, at the request of the coach, in a conference center over 50 miles from the company's headquarters. This guarantees that everyday work does not get in the way of the crisis intervention. The team first starts with some playful exercises that help build the group and also make for an enjoyable experience. Then, the real work starts. The analysis of the last few days has shown that the guiding coalition is seen as very credible throughout the company. Larry alone seems to have a different opinion. Marc knows more about this: "Larry did not change his opinions even when I spoke to him in private. He sticks by his point that our problems are not real and that he alone is responsible for the success of the company. He doesn't think much of teamwork. Other than that, he has no new arguments."

Anne intervenes: “We have a skilled mediator here. I will speak to him and arrange a meeting with Larry. There must be a reason for why he is behaving like this.”

The coach expresses his thanks here and then continues, “I have written down who’s on the team now.”

He points to the flipchart on the wall, which says:

Emergency Power Team

Stephen (CEO)

Peter (Head of Development)

Anne (Head of HR)

Marc (Scrum Developer and designated Scrum Master)

Sarah (Head of Controlling)

Claudia (Product Manager)

Thomas (Head of Quality Management)

Christina (Corporate Processes)

Hannah (Head of IT Operations)

“So, there’s nine people on the team. I myself am a temporary visitor, so I don’t count”, the coach explains. “And we have Frederick as assistant for legal matters. We’re not certain about Larry, and about how to handle him. The formal requirements in terms of team size are met. Now we should start by organizing ourselves according to Scrum to stay credible and to actually learn the process.”

The team’s roles are soon allocated: Stephen will act as Product Owner for the Emergency Power Team, Marc volunteers as Scrum Master. However, he asks for (and is promised) active support from the coach until he is really fit for the job. When everybody knows his or her role, the coach asks a provocative question. “Dear Product Owner: What is your product?”

Stephen takes a minute to consider his answer: “That’s a hard question. It’s not software. Product means something that’s produced. But we’re not producing a traditional product. It’s processes. So, our product is corporate processes.”

The coach is impressed. Normally, it takes much longer for such a realization to come about. He praises the team: “Excellent! That’s it exactly. It means that we need to deliver incremental improvements to processes with every Sprint. In actual fact, we have multiple products, since we’re not dealing with a single process, but with many. Our starting point, however, is the software development process. We now know our product increment artifact. What we need now is a vision and goals that reach people’s hearts and minds. Actually, we need two of each: For the Emergency Power Team and for the crisis in general.”

The team splits into two smaller groups for more effective work. One sub-team looks into the vision for the Emergency Power Team, while the other concerns more global issues. The groups agree to meet again 1 h later and to then compare the results. The work proceeds quickly, and the first group can present its interim results at the end of the hour. Hannah explains: “For us as a team, the vision seems clear: We want to rescue our company!”

This simple expression is met with applause. Stephen comments: “The vision is perfect to communicate and reaches people’s hearts. But what about their minds? Maybe your goals can help.”

Hannah nods and introduces the individual targets:

1. Customer satisfaction needs to be improved substantially. According to Sarah’s figures, we can stomach 50 customers leaving per month, not 250. That means increasing customer satisfaction by 500 %.
2. Satisfied employees are an invaluable asset. Anne has shown that 8 % staff fluctuation have been normal per year, but we’re currently at 15 %. We want to be better than we used to be, so we need to improve staff satisfaction by a factor of 3. This means a fluctuation rate of 5 % per year.
3. We are too slow compared to our competitors. The data shows that we currently need a full 9 months to deliver on a customer request. Our aim is to be better than the market, so customer’s queries need to be answered in full in less than 2 months.
4. Our productivity is too low. We don’t actually know how productive we are. Our goal is to be able to measure productivity in objective terms and then to triple it.

“Those are some ambitious targets”, Thomas says. Stephen nods, but also mentions that targets have to be ambitious if people want to really change something. He says: “Well done. If we manage to reach all of these goals, we will be safe. How we get from here to there is another matter. But the second group might be able to help.”

Peter stands up: “The vision and the targets for the company are actually quite closely interrelated. We only added that Scrum belongs to them as well.”

The discussion continues late into the afternoon. In the end, the guiding coalition decides to use the same vision and targets that they have chosen for the company itself. The end result is:

Our company is experiencing a grave crisis. To rescue it from the threat of insolvency in 18 months, we need to:

- Improve customer satisfaction by a factor of 5 (fewer than 50 customers leaving per month, compared to today’s 250).
- Increase staff satisfaction by a factor of 3 (5 % fluctuation per year, compared to today’s 15 %).
- Respond faster to customers’ wishes and implement their needs in our software in less than 2 months (instead of today’s 9 months).
- Measure our productivity objectively and triple it.

For these ambitions to become possible, we will be transitioning our most critical software development project over the next month to Scrum, an agile product development framework. Scrum enables us to respond quickly to new challenges and to uncover all the problems that stand in the way of our ambitions.

We will plan and publicize how we will be going forward from there. The Emergency Power Team is in charge of leading and supporting this transition.

Everybody seems happy with this draft, although the coach mentions that the targets will have to be revised constantly over the next few months and that this version will certainly not be the final draft. The members of the guiding coalition check in to their rooms and end the day at the hotel's bar. The next morning, all come together again in the conference room, with expectant looks on their faces. The coach takes a few post-its from his kit and distributes them to the attendees: "We have decided to work according to Scrum. We know our targets. Now we need a Product Backlog. Please write down everything we need to or should do to reach those targets."

The participants are not used to this type of working, but it seems effective. A few minutes later, the group is split into three. Each team is asked to prepare new post-its and consolidate the finished ones. After this phase, the teams pin their results on the board. The outcome is a colorful mass of post-its with many different ideas. This is the time for the coach to step back in: "Well done. Now, I'd like you to sort these ideas into a definite order in which we can process them. You can add new ones if there are any gaps. Your Product Owner has the final say."

The task leads to lots of debates and needs a lot of time. After the lunch break, the coach stops the work: "We have agreed on the most important tasks. We can come back and look at the less important ones at a later date. For now, we just need something to start working on."

Everybody looks at the results. The first nine elements are:

1. Plan our Sprint
2. General Scrum training for the guiding coalition
3. Product Owner training for Stephen
4. Scrum Master training for Marc
5. Convince and involve Larry
6. Communicate the vision and goals to the company at large
7. Make our Product Backlog transparent for everybody
8. Transition the project to Scrum
9. Monitor project progress

This is followed by many other points, but the coach decides to stop here for the time being. He pins the nine items to the flipchart and says: "That's a good start. We are now already in the midst of planning our Sprint, so item one is under way. Let's estimate the effort for these items to see how much work we have ahead of us."

He hands a pack of cards around and tells the group that it is time for a round of "Planning Poker". "We don't know how long we will need for every individual point. Even if we knew it, the time might change over the course of our rescue mission, for instance when we encounter new problems or if we simply speed up in our work. If we estimated our work in absolute time units, we would have to re-estimate the entire backlog, so we're doing this in relative figures. The cards in the deck have the numbers 1, 2, 3, 5, 8, 13, 21, 40, and 100 on them. Up to 21, they

follow the Fibonacci sequence to help you think how big each element actually is—it often happens that people go up by doubling the estimates without any sense or purpose. Above 21, the values get really large, so Fibonacci does not make sense anymore. Let me start with a wild guess: Planning our Sprint will get a score of three, because it probably is the “smallest” element. Please estimate the relative size for the other items by comparison to that baseline. If you believe that the general Scrum training will be twice as big, then pick either a five or an eight. Turn your cards over all at once before you start discussing them. Let’s have a go. Marc, as our Scrum Master, you are not allowed to take part. But then again, you’re also a team member—that shouldn’t happen in a Development Team, but it’s normal in a guiding coalition. So, you can add your estimate in your function as team member. Stephen, as Product Owner, you’re also out of the running. But you can answer questions and take decisions.”

The group is slightly insecure when it begins working on the first estimate. The second Product Backlog item is explained again by the Product Owner, and everybody puts the card face down on the table. The cards are turned over, and the coach checks the outcome: “So, we have three 5s, three 8s, one 21, and one 1. Anne, you picked the 21. Could you tell us why?”

“I’d be happy to. We will finish the planning of the Sprint latest by tomorrow evening. The training will take 2 days. Two days times nine people makes 18, which is why I picked 21.”

“Careful there, Anne. This is not about estimating man-days, but about weighing up the individual elements. Lots of people make that mistake—just try to remember it next time. Sarah, you picked a 1. Why is that?”

Sarah: “Well, I thought that both items are about the same size, because both need 2 days for all of us. But I actually don’t think that we will need 2 days. With our coach, we’ve got a head start, so a single day should be enough. That’s the 1, then.”

Stephen seems happy: “Great! So we’re already saving time for the second item.”

Everybody’s amused. The coach explains: “Yes, that could work. You see how the planning poker is less about perfect estimates and more about discussing the matter. Our estimate has just uncovered an ambiguous element that the Product Owner needs to clear up. And we also understand what is actually required. The estimates will just define themselves in the end. Your Product Owner has made it clear that you won’t need a full standard course, but actually just one that matches your unique needs. As your coach, I can tell you that you’ll need 1 day, given that we’re continuously working together already. So, back to work again.”

All team members pick their cards, again face down. When they turn them over, the values range between one and two. The coach decides on the final results: “If we are all in that range, let’s go for the bigger number. One level higher or lower is no major issue during Planning Poker, since the estimates will only become actually predictable when we know the team’s speed or what we like to call its velocity. If the estimates rise, the velocity also increases, so we don’t have to worry about this now.”

All items are estimated by the group, which is finding it easier to do now. The first results are available within half an hour:

1. Plan our Sprint—3
2. General Scrum training for the guiding coalition—2
3. Product Owner training for Stephen—1
4. Scrum Master training for Marc—1
5. Convince and involve Larry—2
6. Communicate the vision and goals to the company at large—40
7. Make our Product Backlog transparent for everybody—1
8. Transition the project to Scrum—100
9. Monitor project progress—100

Although many of the items seem very obvious, others are more cause for concern. The fact that the vision and the goals need to be communicated constantly and perseveringly is obvious—but how this can be done is not yet known. What it takes to transition the project to Scrum is another great mystery for everybody except the coach. Stephen suggests breaking down some items into smaller packages. This does not work for the project transition, but there are more options for the communication part. After a brief discussion, the Product Backlog looks as follows:

1. Plan our Sprint—3
2. General Scrum training for the guiding coalition—2
3. Product Owner training for Stephen—1
4. Scrum Master training for Marc—1
5. Convince and involve Larry—2
6. Produce a communication concept—3
7. Make the Product Backlog transparent for everybody—1
8. Communicate the vision and goals to the company at large—40
9. Transition the project to Scrum—100
10. Monitor project progress—100

The Emergency Power Team decides in favor of this Product Backlog. Everybody understands that much work is still needed on the details, but that can be done in parallel to actually solving the first few items.

As a final assignment for the day, the coach asks all participants to write down their reasons for contributing to the Emergency Power Team on post-its. He asks them to be open and honest about this.² The results are surprising for everybody concerned: “Saving the company” is mentioned as a reason, but it is not the primary

² Normally, such reasons are explored in confidential meetings, since the necessary trust for honest answers is often still missing at this early stage in the process. To simplify the story, we are assuming a deep founded trust between the actors.

motivation. Far more people mention aspects like “career”, “a challenge”, “stability”, or even “fun”. It seems as if every member of the team has his or her very own reasons for committing so much work to saving the business. Another surprising feature is that every personal reason can be squared with or even reinforced by the vision the group developed (and its allocated goals). This seems to be the reason why the vision has such great appeal for the members of the Emergency Power Team. Having been made to think about their reasons, the team leaves for the day.

On the third day, the team gets to work on planning their Sprint. It decides in favor of a 2-week Sprint and estimates that the first seven items on the Product Backlog can be finished in that time. It also produces a precise plan for tackling each item. The day ends quickly after a few more discussions about possible dates and appointments.

The next day, Stephen’s cell phone is ringing at 7 in the morning. Frederick is on the line, and he is audibly excited: “Stephen, great to have you back! The last 3 days were absolute chaos. Somebody started telling horror stories about the end of our company. The rumor mill is going into overdrive. We need to act now!”

Stephen assembles his executive team to look into the causes for the unrest. The picture that emerges is obvious: Larry seems to have followed his usual destructive tendencies and spread some half-truths to his colleagues during lunch hour. As it happens, the stories were taken up and made to sound worse and worse, until the rumor was that the company would be sacking half of its workforce by the end of the quarter. The stories have spread like cancer and need an urgent response.

The Emergency Power Team thinks about a suitable answer, and it reaches a decision: To counter the current rumors, Stephen will be sending an email to all members of staff, in which he will invite them to the next Sprint Review in 2 weeks’ time. The Product Backlog and the Sprint Backlog will be posted for everybody to see in the cafeteria. A dedicated rumor wall will be installed to stop the spread of unfounded rumors in future. This wall will be checked for new stories every day, and the correct answers are posted on it. The first rumor on the wall is: “Sacking people by the end of the quarter”, which is immediately followed by the truth: “This is nonsense. There will be no sackings this year unless the commercial situation of the company takes a turn for the worse.”

The foundations for open communication are in place.

15.4 Vision and Strategy

The following 14 days are given over to hard and intensive work. Training courses are being planned and conducted, the communication concept is worked and re-worked, and Larry is talked to many times. The coach has also asked the team to continue working on the strategy, mostly in terms of the vision and how it will be communicated. When the team meets again, the coach starts: “We already have a first draft of our vision and strategy, but as I said, that first draft needed some more work. After all, these two points are the cornerstones of the entire structure we are trying to build.”

All eyes turn to the poster with the vision, which the coach reads out again: “Our company is experiencing a grave crisis. To rescue it from the threat of insolvency in 18 months, we need to:

- Improve customer satisfaction by a factor of 5 (fewer than 50 customers leaving per month, compared to today’s 250).
- Increase staff satisfaction by a factor of 3 (5 % fluctuation per year, compared to today’s 15 %).
- Respond faster to customers’ wishes and implement their needs in our software in less than 2 months (instead of today’s 9 months).
- Measure our productivity objectively and triple it.

For these ambitions to become possible, we will be transitioning our most critical software development project over the next month to Scrum, an agile product development framework. Scrum enables us to respond quickly to new challenges and to uncover all the problems that stand in the way of our ambitions. We will plan and publicize how we will be going forward from there. The Emergency Power Team is in charge of leading and supporting this transition.”

He gives the vision a moment to sink in again before continuing: “Ladies and gentlemen, a vision is meant to be an expression of a specific direction into a desirable future. Is our vision desirable?”

“Of course it is. We’ll be out of business in 18 months’ time without it.”

Marc shakes his head: “That is not actually a problem for our better employees. They’ll find a new job anywhere. It needs more than that for the vision to be actually desirable for everybody.”

Peter adds: “We are not being transparent enough.”

The Emergency Power Team has completed all of its training courses by this point in the process, so the group has begun to understand Scrum and its advantages. The group discusses the relevant points, leading to a new poster, headed “We want Scrum, because we want:

- Clear responsibilities.
- A team that can focus on the work at hand every single iteration.
- A Development Team that can decide for itself how to approach its requirements.
- A troubleshooter who clears the way for the Development Team.
- A Scrum Team that can improve its own processes.
- A dedicated contact person for the client.
- Clients who can state their wishes and expectations whenever they need to.
- The opportunity to respond immediately to the clients’ wishes.
- A new focus on product quality that is not watered down by time constraints.”

The guiding coalition decides to include these points in the vision. However, Anne immediately notices one major omission: “We keep talking about Scrum, and

we know what is meant by it. But remember: Our colleagues might not. We should definitely offer short introduction courses for everybody.”

The coach agrees: “Other companies tend to offer one or two hours of introductory training per week, usually at a fixed time. That allows all people to get familiar with the issue and learn about its basics at least.”

A decision is taken that Marc should plan and conduct these courses, with the coach helping him for the first session. That approach also has the added advantage that Marc can get a first-hand insight into the cares and concerns of the workforce. Christina has something to add: “Are we really only moving one team to Scrum, or are we speaking about the entire organization? Now that we have learned so much about Scrum, I guess that we could get so much more out of it if we take the entire company along.”

“Christina, I believe you are right”, Stephen says. “But I don’t want our people to get worried. Changing the entire organization will be such a major event that most colleagues will get scared. It is easier for them to swallow if we only have one pilot team transitioning to Scrum.”

Peter speaks up: “We are expected to embody our vision. Openness—and that means honesty—is a basic principle of Scrum. Should we not state clearly what is going on?”

A quite heated debate breaks out. Terms like honesty, integrity, and perspectives are flung around, but in the end, the team agrees to change a few phrases in the vision:

For these ambitions to become possible, we need to change for good. We will be transitioning our most critical software development project over the next month to Scrum, an agile product development framework, to test the applicability of the concept in our company.

The group is happy with the work. However, people acknowledge that the targets will still change over the next few months. For the time being, a single project is transitioned to Scrum, but others will follow, depending on the company’s experience with the concept. Scrum PRN will probably not suffice, and a Scrum Software Studio or Profound Scrum might be in order, which would again affect the goals or even change the vision. The Emergency Power Team agrees to check both the vision and the goals on a regular, 2-weekly basis during their Retrospectives. The coach gets the group to focus again: “Very well, for the moment, we are happy with the vision and the goals. Now let us see whether we have actually complied with the criteria I mentioned. Does our vision meet the needs of all the people it affects?”

After a brief hesitation, Anne says: “Well, basically, everybody wants to be master of his or her own fate. Scrum helps in that sense, because it puts the responsibility with the Development Team itself. We have made that plain by adding it to our vision. So: Yes.”

But Sarah counters: “What about security? If people do not feel safe in their jobs, all other aspects are worthless.”

Stephen proposes another item for the list:

- Avoid all layoffs to protect the most valuable asset of the company—its people.

There is general agreement about this. The team goes on to consider whether all criteria are fulfilled, but does not seem to have any new ideas. A decision is taken to introduce changes once the first feedback has come in from the workforce. The coach asks another question: “Are the targets challenging, but realistic?”

It is immediately obvious to everybody that the targets are indeed a challenge. Several 100 % improvements in so many key areas is not an easy proposition. The question of feasibility is harder: Everybody has a sense that it should be realistic and feasible, but there is no data to actually prove this. Such data would, however, be essential to make the targets sound and credible. Everybody sits down to take another close look at the targets:

- Improve customer satisfaction by a factor of 5 (fewer than 50 customers leaving per month, compared to today’s 250).
- Increase staff satisfaction by a factor of 3 (5 % fluctuation per year, compared to today’s 15 %).
- Respond faster to customers’ wishes and implement their needs in our software in less than 2 months (instead of today’s 9 months).
- Measure our productivity objectively and triple it.
- Avoid all layoffs to protect the most valuable asset of the company—its people.

Anne speaks out: “We could cover that last point if Sarah and her controlling department go and check again if we can indeed cope with no compulsory layoffs for the next 18 months. All things being equal, I mean. We could take that to the works council and arrange some sort of formal agreement. Something of the type: If the workforce supports our attempts at consolidating the business and if the commercial situation does not take a turn for the worse, then nobody will be made redundant.”

There is a brief discussion about the idea, but most people seem happy with it. However, verifying the other points seems impossible, because no reference data is available. Stephen seems disappointed: “We need to find some reference companies from our sector that has the data we need for our targets and that is actually happy to share it with us.”

The group decides that all of its members should produce a list of suitable companies for the next day, which can then be used in a comparison. The coach adds insights from his experience, but he is bound by confidentiality to not give out too much information from his reference projects. Concrete figures are only available for the point “productivity”. Beyond that, the team is on its own. After a short breather, the group gets back to work: “Is our vision explicit enough to work as a guideline, but generic enough to give people room to maneuver?”

Marc: “We are saying that we are introducing Scrum and that we are trying to improve customer satisfaction, staff satisfaction, productivity, and responsiveness. This is all quite explicit. And we’re leaving room for interpretation, because we mention Scrum, but nothing else about how we are planning to get there. I think we’ve fulfilled every point.”

Although everybody agrees with Marc, they also know that they should gather experiences and see what happens. The team wants to test how well the concept can be communicated, but the discussion drags on until Peter has had enough of it: “People, we aren’t getting anywhere. I say: Let’s just try it out. Let’s see if we can get our vision and our ambitions across to somebody in less than 5 min. What about Frederick? We wanted to give him an update about what we’re doing, anyway.”

The proposal seems absolutely pragmatic and helpful, and Frederick is immediately asked to join the team. Hannah starts off and explains the vision and the goals. After only a minute, Frederick stops her: “I get what you are saying, except for this term ‘Scrum’ that you used twice. What’s that about—Why Scrum?”

Hannah: “We only explain what Scrum actually is in passing, in this ‘agile product development framework’ part. We are planning to give people an option to join a weekly Scrum crammer course.”

“Let’s write that down”, Frederick suggests. “Do we have the dates?”

Marc nods and adds another final paragraph: “Anybody who is interested in learning more about Scrum is invited to come to Room 357 for a Scrum crammer course from 2 p.m to 4 p.m. every Friday. For any other questions and comments, simply call Marc on extension -333.”

Frederick is happy with this. He takes the initiative and calls in a colleague from the legal office, a department that is traditionally not deeply involved with the software-side of the business. Hannah explains the vision and the targets again. Although the colleague has a number of questions, they mostly concern the practical side and not the actual vision. The guiding coalition is convinced enough for the coach to close the proceedings: “Normally, the last step would be to challenge the entire vision again from the point of view of the people affected by it, but in a sense we have already done this. I would suggest that we do this again in 2 weeks’ time and include our newest impressions then.”

Everybody agrees. The meeting room empties soon afterward, but the group leaves a few sheets of brown paper with its vision behind for everybody to see:

Our company is experiencing a grave crisis. To rescue it from the threat of insolvency in 18 months, we need to:

- Improve customer satisfaction by a factor of 5 (fewer than 50 customers leaving per month, compared to today’s 250).
- Increase staff satisfaction by a factor of 3 (5 % fluctuation per year, compared to today’s 15 %).
- Respond faster to customers’ wishes and implement their needs in our software in less than 2 months (instead of today’s 9 months).
- Measure our productivity objectively and triple it.
- Avoid all layoffs to protect the most valuable asset of the company—its people.

For these ambitions to become possible, we need to change for good. We will be transitioning our most critical software development project over the next month to Scrum, an agile product development framework, to test the applicability of the concept in our company. Scrum enables us to respond quickly to new challenges

and to uncover all the problems that stand in the way of our ambitions. We will plan and publicize how we will be going forward from there. The Emergency Power Team is in charge of leading and supporting this transition.

We want Scrum, because we want:

- Clear responsibilities.
- A team that can focus on the work at hand every single iteration.
- A Development Team that can decide for itself how to approach its requirements.
- A troubleshooter who clears the way for the development team.
- A Scrum Team that can improve its own processes.
- A dedicated contact person for the client.
- Clients who can state their wishes and expectations whenever they need to.
- The opportunity to respond immediately to the clients' wishes.
- A new focus on product quality that is not watered down by time constraints.

Anybody who is interested in learning more about Scrum is invited to come to Room 357 for a Scrum crammer course from 2 p.m to 4 p.m. every Friday. For any other questions and comments, simply call Marc on extension -333.

15.5 Communication

After some well-deserved rest and recuperation over the weekend, the Emergency Power Team starts with fresh energy into the next week.

The coach begins: "We have already mentioned how important communication is for the change effort. People who don't know what is happening cannot take part in it. We have also spoken about the fact that communication is a leader's job—which makes it a job for you, the guiding coalition."

"Yes, we know all of that", Thomas interjects. "Aren't we communicating already?"

"Yes, you've started to, but it's unplanned and insufficient as yet. When did you last communicate something to your teams?"

"Last week", Stephen says.

"See what I mean? The changes are the most important topic for the company, because we will all be out of a job in 18 months if it fails. That means that communication should also be as important. We shouldn't communicate once a week, but many times every day. I want you all to think of ways to communicate the changes at least three times per day. You have 10 min to think of something."

The team gets to work on that new aspect. Ten minutes later, the first results are in. It is obvious that there are many good ideas. The executives think that they can include the vision in their normal, everyday meetings. Peter has a specific example: "I just had my weekly status update with a development team. I used that to mention the current situation of the company and to spread the sense of urgency for the project. I also told them that our top targets are improving employee and customer

satisfaction, and I asked them how they could go about these targets in their work. They had some good ideas—even without Scrum.”

This is followed by another few examples from the team. Everybody is sure that they could manage the necessary three instances of communication per day, but the coach does not let up: “This is a start. We’ve covered this channel of communication now. Then there’s the cafeteria with our backlogs and rumor wall. But which other channels are there?”

The list grows and grows:

- Emails
- Flyers
- Posters
- T-shirts
- Stickers
- Pens
- Personal letters
- Private conversations
- Informal chats during coffee breaks
- Large-audience events
- Workshops
- Designated contact people for questions
- Article in the in-house magazine
- Round-table discussions
- Intranet forums

“T-shirts and pens?” Hannah seems surprised. “What good are they? You can’t fit anything on them.”

Claudia is not worried: “We often use them in product management when we work with marketing. If you get the details of the message across via the other channels, a simple logo or a single word are often enough to get people’s memory working. They remember what they’ve heard. Yes, even a pen can do some magic.”

The group discusses the right choice of channels for a while, until it reaches the decision to simply try out all possibilities. The details will be defined in the communication concept that is supposed to be finished by the end of the week. Particular care will be invested into the interactive workshops for employees, since they are often easiest to recall and can also add more invaluable input about how to proceed.

Sarah is still not absolutely happy with the results: “How can we make sure that everybody gets the same message? I mean: that the information is consistent across all channels?”

“Let’s collect all questions and answers that we come across. We could put an FAQ page up on the intranet”, Hannah suggests. “That would kill two birds with one stone: We have all the information at our fingertips, and the employees can read up on it, instead of having to listen to rumors.”

The coach starts to speak: “Your behavior will be the essential piece in the puzzle when it comes to communicating the changes. I know that I don’t have to tell you, but it is important that we are all on the same page. Do you agree with me, that employees will only follow our example if that is what we are offering them: convincing examples?”

His statement is met with full agreement. He continues: “Then let’s live like this. Whenever we do or say anything, let’s first think about whether it matches our vision and helps the company.”

15.6 Empowering Broad-Based Action

The communication efforts about urgency, vision, and targets get off to a good start. The team not only produces a viable communication concept, but also begins to put it into practice right away. In addition to his normal everyday communication and representation duties, Stephen somehow finds the time to write an article for the in-house magazine, and Claudia gets to work with the marketing team on a set of flyers and posters. The questions of the workforce, whether spoken or not, are answered in public, and the rumor wall in the cafeteria starts to fill up. The Emergency Power Team meets every day to discuss how to move forward. Larry is, of course, a frequent topic of their debates. All attempts to change his attitude—even conversations guided by a professional mediator—come to nothing. The opposite is true: Larry feels confirmed in his attitude because of the sudden attention he is receiving and begins to assemble a group of critics around him. He even starts calling the group “the Resistance”. Two days before their Sprint Review, Anne raises the issue again after the Daily Scrum. “I don’t think we’ll ever win Larry over. He is actively working against us, and he is getting people on his side.”

All other attempts at reaching out to Larry end in a deadlock. In the end, and despite everybody’s unhappiness with his decision, he decides to resign from the company.

On the day of the Sprint Review, over 100 employees decide to take up the invitation and come to the cafeteria. The Emergency Power Team begins by explaining the urgency and the vision again. Marc then takes a moment to explain how the team has organized itself, and then looks at the Sprint Backlog:

1. Plan our Sprint—3
2. General Scrum training for the guiding coalition—2
3. Product Owner training for Stephen—1
4. Scrum Master training for Marc—1
5. Convince and involve Larry—2
6. Produce a communication concept—3
7. Make the Product Backlog transparent for everybody—1

The team then takes turns to explain the results and progress of the recent Sprint. There are some questions from the audience, but no actual discussion until item 5 is

reached. Suddenly, the audience seems restless. Anne takes to the stage: “Larry is a great architect, and we all appreciated his work. Unfortunately, his social skills left something to be desired. He not only said that he was unhappy about our plans—that is everybody’s right—but he was rude and actively worked against the company. In a crisis like the one we are facing now, that sort of behavior cannot be tolerated. In the end, the matter escalated, and Larry’s resignation might have only stopped him from getting fired in turn.³”

The following discussion is quite heated, but generally remains professional in nature. People demand explanations and examples, and they want to know what was done to keep Larry. However, there are also voices that seem quite relieved about the departure of the “selfish” Larry. At the end of the debate, everybody appreciates that everything was done to persuade him to stay, or indeed that much more was done than was ever done before for a person who behaved like that. Nobody thinks that the executive team treated Larry incorrectly or unfairly, and people feel a renewed sense of trust in the guiding coalition that is seen to act transparently and openly.

The presentation of the communication concept leads to a productive and professional discussion. Many of the attendees have good ideas to contribute, which the team writes down and posts on their boards. At the end, the team agrees: The communication concept could be improved collaboratively. The meeting closes with the coach asking the participants which items they would like to see the Emergency Power Team executing. Every attendee is asked to write down one preferred item and post it on the board, giving his or her name for follow-up questions. Eventually, there is a list of 30 items, some of which fit the context of the Scrum introduction. The coach gestures towards the board and tells Stephen: “So, this was your first stakeholder survey as Product Owner. Some input for your Product Backlog, and hopefully just the first survey of many.”

To plan the next Sprint, the group is asked by the coach to conduct another full-scale workshop. “Now that we have spread the message about the urgency, our vision, and our goals, we need to get to the real work. The communication concept is in place, and we can wait another 2 weeks for the next Sprint Planning meeting. We all have more to do until then than just communicate. We need to empower our people.”

“What do you mean by empower?” Thomas asks. “Our people are empowered, aren’t they? We want them to contribute, don’t we?”

³ This example is exaggerated deliberately. This is not meant to “demonize” Larry, but to start a dialogue with the other employees and to make the implications of working against the company clear. The credibility of the guiding coalition would be at risk if it had been seen to accept such misdemeanors. Incidents of this nature are always a balancing act, in which people’s trust needs to be protected and maintained. Losing somebody without damaging the trust of the people who are staying is only possible if everything humanly possible was done—and seen to be done—to keep Larry on board.

“Yes and no”, the coach answers. “You are correct from our point of view, but from their point of view, the situation might look different. So, who needs to be empowered?”

Marc has an idea: “Well, we want to introduce Scrum PRN in a single product team first. Wouldn’t it be enough to empower that team?”

Peter adds: “But we should not forget about the other areas that belong with that team, like product management and IT operations. We need them to be able to help.”

The coach takes over again: “We have four key problems that we need to master: Mistrust, insufficient competences, the attitudes of people, and other inhibiting factors.”

“We already checked the trust issue. Apart from Larry, everybody seems sold”, Sarah says. “I don’t see any need for intervention there.”

Stephen agrees: “That’s right. If we continue to embody what our vision demands, then we should be on the safe side. But what about competences? Our developers know how to code, don’t they?”

Peter has the answer: “Of course, but never according to Scrum principles. They will need new skills for that. And then there’s customer focus—a completely new experience for many of them.”

“Okay, I’ll include Scrum and customer focus in our backlog as training needs. We can discuss the priority later on”, Stephen agrees. “Now, about the next point: I think that the attitude of our chosen team is quite positive. Anybody?”

Nobody has anything to say, so Stephen continues: “Okay, so that’s on hold as well. But what are inhibitory factors?”

The coach explains: “What I mean by that term is structures, systems, or individual managers who might stand in the way of the changes. In our case, that means everything in the way of introducing Scrum. One point we know already: the fact that many developers are involved in multiple projects at the same time. Claudia wanted to spread all of them out equally to the projects in order to get one fully available team for our primary project. We could have started without that, but then the project team would not have been as productive as it can be now.”

The discussion goes on for a few minutes, before the group starts brainstorming. Everybody contributes every “inhibiting factor” that he or she can think of, which are then grouped. Some are rated as “not currently relevant” and handed to Marc as the guiding coalition’s Scrum Master to be addressed later. The following list remains:

- We currently have programmers and architects on the team, but no software testers.
- The eventual team make-up is still unknown.
- The development environment still lacks the necessary infrastructure.
- New development servers are urgently needed, but cannot be ordered, because the budget for the year has been allocated already.
- The developers are still new to Scrum.
- The developers use desktop computers, not laptops.

- There is no room in which all developers can work at the same time.
- There are no rules about how the Development Team can assure the quality of their work results or get it into production.

Thomas intervenes: “Why do we need testers on the team? Our process is clear about one thing: The final approval is done by my people.”

Peter: “The Development Team needs to deliver finished software every Sprint. Do you consider untested software finished?”

“Of course not”, Thomas says in his defense. “But we could have the output of each Sprint tested in the next one.”

“But that means only finding bugs and flaws in the next period. Fixing them would force the team to return to a job they had thought finished. That takes too long—and costs money”, Claudia adds. “Thomas, if I in product management have to give up my best assets, then you should do so too. I suggest that you put your best people right into the Development Team. They would still be your direct reports.”

“I can’t see how that would work. I wouldn’t have access to them. Their other work would go unfinished.” Thomas is set against the idea.

Claudia argues: “If finishing the software is our top priority, then you need to make sacrifices. Like myself you might not have direct access to your people, but you will be able to adjust. Have you forgotten about the state of this company?”

“Of course not. But I don’t like it.”

The coach speaks up: “Thomas, I can see where you are coming from. It feels like we are taking something away that matters to you. But don’t worry. Nobody is taking anything away. The opposite is true: You are getting more important responsibilities. If the operative work is done in the Development Team, you will have more time for strategic issues. For instance, we definitely need a sophisticated testing concept for the product. The old one does not match our new challenges. We also need to define how each piece of code will make its way from the developer’s computer to actual production.”

Thomas’ face lightens up again: “I wanted to revise that anyway. The old concept is a legacy from the 1990s. We never had time to change it because of all the manual testing we need to do.”

“Very well”, Stephen says. “Then go and find some good people who can become full-time members of our first Scrum team. By the way: We need to get that team staffing fixed.”

The coach asks: “Why don’t we let the developers decide for themselves? After all, they need to work with each other. We only have to tell them the terms and conditions, like the maximum team size.”

Finally, an opportunity to try the phenomenon of self-organization with something tangible. Peter and Thomas are asked to inform the developers and the testers. The coach has some useful hints for them: “The Development Team should not have more than nine members. Seven would be perfect. Anything above that, and there is too much communication, and the team will dissolve into smaller sub-teams. The team must also be able to deliver finished software at the end of each Sprint. The chosen team members need to be aware of that and they need to

have people for all required roles. Experience tells us that we need about as many testers and programmers on the team, although the testers also need to know how to code in the sense of producing automated tests. Give the developers those guidelines, and they will be able to form a team in less than an hour.”

The plan is put into action. After a short break, Peter and Thomas go and inform their teams and ask them to pick the people for the Scrum Team. It does not take long for all of them to come together again.

Hannah starts: “In IT operations, we have no development environment available for the team. Our current tools are simply not suitable for constant integration and testing. We also can’t produce code branches or test them in any meaningfully automated way. And we lack servers. We simply cannot provide the computing power we need. And we haven’t yet evaluated, let alone procured, the necessary tools. It could be months until we are sure that the new tools will match our IT environment. We don’t want to break anything.”

Stephen is unconvinced: “I understand. I also want to keep our IT intact. But the team needs its wings—you know how urgent this is. Which software components are we talking about, anyway?”

“Well, we’ve got server applications to manage our source code, and an integrated development environment, what we call an IDE, for development support on every workstation”, Hannah answers. “That needs to work properly with the sever components. And then, there’s quality monitoring tools, which also need to be compatible.”

“If we buy new servers anyway, could we take them off the grid and keep any potential damage away from the rest of the IT infrastructure?”, Stephen asks.

Hannah takes a moment to think about the idea. “That could work for the servers, but our people need access to our file system and to their emails. They need access to our normal network. I am sorry, but there will be side-effects.”

“Okay, I see. But we need to be up and running as soon as possible—by which I mean next week. Let me suggest something: I will see to it that you get your order cleared for the servers and the software. In the meantime, you find a solution, so that we can start without any lengthy evaluation. I prefer starting now with the second-best solution than starting in 6 months’ time with the perfect tools.”

Marc joins the discussion: “Our developers are already complaining about their big old desktops. They can’t have proper meetings with them, and they are stuck where they are if they need to find a quiet spot for their work. We need laptops that can handle the load.”

With a slight shrug, he adds: “Of course, we didn’t plan for that. I had a look at some and asked the other developers. I am talking about systems that can cost up to €4,000.”

Stephen seems surprised: “€4,000 times nine developers. That’s €36,000. On top of the servers and the software licenses. I don’t think I can get that past our advisory board. Is there any other solution?”

“There might be one”, Peter suggests. “I am thinking that two laptops per team are enough. Anybody who needs one can take it. It might not feel like your personal computer, but it will be enough for the time being. We just need to set them up so

that our developers don't need to arrange everything afresh again when they swap users."

"Don't worry. We're already used to that", Hannah smiles at him. "We've had lots of problems in the past, but we now do it automatically whenever a computer logs into our network."

"Two laptops can be ordered without having to get clearance from the advisory board, I guess. I'll see to that", Stephen promises.

"Speaking of quiet corners", Anne mentions. "We still have that room issue. Where can we put our developers?"

Sarah interrupts: "They are already sitting around somewhere. Why do we need to speak about these trivialities?"

Anne cannot accept that: "Everybody knows that a group of people can only become a team if it is actually physically in the same location. Otherwise, we're speaking of workgroups. If we just repeat the mistakes from the past, then we'll have information not being shared, people working by themselves, and people getting frustrated. No! We need a team room for the entire team. No exceptions. A team room that is large and open-plan, so that everybody can see and work with everybody else. At the same time, we need corners for quiet work if people need to work on particularly complex pieces of code. And we need room for discussions, meetings, and phone calls without everybody else listening in. The Development Team should be involved in the details, because that will make them feel at home in the end."

Stephen seems lost: "I can't pull that room out of the hat now. I just can't."

The coach does not accept that: "Do the others agree?"

Everybody starts to think about solutions. In the end, Peter has an idea: "We have a few spare rooms in the basement. They are a bit out of the way, but they are free and they are large enough. That used to be a storeroom, which we now have on the ground floor. The rooms have been empty ever since. A bit of paint and some furniture, and it should be alright. I'll have a look. Anne, do you want to come?"

"Love to!" Anne seems excited. "Wouldn't want to miss it."

Hannah and Claudia are also interested, but the tour is postponed to the afternoon, because the coach has another question: "Our developers are still new to Scrum. The training was a start, but it did not solve the entire problem. Please take 5 min and think about what else we could do to get our team off to a flying start."

The problem and its solution are the topics of a lengthy discussion. The coach lets it go on for a few more minutes before intervening: "It's great that you have so many ideas. Let's sit down and organize them. We can then prioritize them, so that we know what we will do and what we will leave."

The discussion continues, but with much more focus than before. Some new ideas are still identified and immediately written down. After about an hour, the Emergency Power Team has settled on five points:

- Close, daily support from the coach
- A direct contact person from the Emergency Power Team for the team
- Literature about Scrum for the team

- The formation of a “community of practice” to share experience, which can include people from outside the company
- Small, tailor-made learning nuggets for the team, no more than 1 h per week

“By the way”, the coach says after looking at the finished list. “There is an established community of practice in town. It’s called ScrumTable and meets on a monthly basis. It’s attended by people from the entire region. Is that what you were thinking about?”

Marc is ecstatic: “It’s great. I’ve been there once. There’s people from all corners, which makes it even more interesting. You can teach and you can learn at the same time. The next meeting is on Wednesday, by the way.”

“Great, I’ll tell my team”, Peter says. “I wanted to go and see how they are getting on with forming the team anyway.”

“Okay, then let’s end the workshop for today. Stephen needs more time for the Product Backlog, and Peter needs to check back with his team. All of you, think about what we’ve been doing today. If you can think of any other issues, any obstacles hindering the Scrum introduction, then write them down. We will see whether they need to be resolved right away or whether they can wait. See you tomorrow!”

Tired and preoccupied, but quite happy with their work, the team separates for the day.

The next meeting starts at the customary early hour.

Peter has news: “We have a team!”

The coach seems happy: “Great! We’re going as fast as this, because we defined and communicated our notion of urgency and our vision. If we hadn’t done that, we’d have to start over again. I am proud of you!”

Peter recapitulates the events from the previous afternoon. The team is particularly impressed by his description of how the A-team—as the new group is calling itself—immediately began to organize itself when it realized that it could do so without management involvement. When Peter returned from the workshop, he was met by seven excited developers who were already discussing how to proceed with their work.

“We have one architect, four programmers, and two testers on that team”, Peter concludes his story. “The A-team is sure that that’s the most productive arrangement.”

“Well, that’s a super start!” The coach is full of praise. “Now, let’s get on with planning our Sprint.”

The second round of planning is much easier for the Emergency Power Team than the first session 2 weeks earlier. After only 90 min, the team has already picked and produced the estimates for selected Product Backlog items.

1. Providing the facilities for the A-team—13
2. Scrum training for the developers on the A-team—1
3. Training to improve customer focus among the A-team’s developers—1
4. Expanding the development environment, including servers—13

5. Actively supporting the A-team in Scrum matters—3
6. Procuring two laptops that are suitable for the developers' work—5
7. Producing a flyer to communicate the vision—3
8. Producing posters to communicate the vision—2

The sum total of the estimates for the individual elements is much higher than in the first Sprint, but the team agrees that these items can be done. The Emergency Power Team gets to work with renewed energy.

15.7 Generate Short-Term Wins

The Emergency Power Team meets again a week later. By now, the rooms in the basement have been filled with furniture and technical equipment. The A-team has been invited to a Scrum course in 2 days' time, although no date for the customer focus training has yet been found. The other points are also still unresolved. The guiding coalition has checked up on progress every day for the last week, so all of the information is known to everybody. The coach starts the proceedings: "We are making headway, and we are helping each other when there are any problems. We should use today to take a closer look at the change process. Every change process needs quick wins. Imagine a process that takes 3 years in total: Without one or two motivating achievements every few months, nobody would go the distance. And these successes also show the company's management and the other stakeholders that all of the efforts are worth it."

Nods on all sides.

"The scope of the change project also influences the need for quick wins. Take our own project, for instance: If we only think of setting up one Scrum Team, it would presumably be enough to get it up and running as quickly as possible and then to prove that productivity increases. But if we think about transitioning all development teams, that is, the entire organization, then one productive A-team would only be a quick win for the short term."

Stephen keeps digging: "What do you mean by short term? That could mean anything, depending on the context."

"Precisely", the coach confirms. "The duration depends on the overall scope of the change venture. As a rule of thumb, it helps to achieve concrete results every 3–6 months. In our case, we should aim for progress every 1 or 2 months, since we only have another 17 months left. If we can go on like we have so far, our work will mean that the A-team can function productively within the first month."

"So, we have had successes", Hannah says. "Why are we spending so much time on this? It's working."

"You are right", the coach admits. "But successes do not come from nothing. We need to plan them and work towards them. If you leave success up to chance, you will not have any."

Stephen takes a minute to think. Then he says: "Let's plan some achievements. One per month would be good. What do you think?"

The group quickly agrees to some brainstorming. The aim is to agree on a list of six possible achievements. Before work starts, the coach gives a word of warning: “Careful: The achievements need to be unambiguous and waterproof. They also need to fit our vision. Everybody needs to be able to see and to verify every achievement.”

The brainstorming takes almost an hour. Items keep getting thrown out, because they do not match the criteria. There is also some uncertainty about what constitutes an unambiguous and transparent target. In the end, the group agrees to a set of criteria and ends the discussion on a positive note. Sarah sums up: “Before we all go to a well-earned coffee break, let’s recapitulate: We’ve agreed to only consider criteria that are measurable. They also need to actually help rescue our company. The following points are agreed:

1. The A-team is working full-time and according to Scrum principles on our most important product.
2. The A-team delivers measurable results.
3. The productivity of the A-team is at least 50 % higher than the previous month.
4. The members of the A-team are more satisfied with their work than before.
5. The A-team establishes a pace it can hold and its forecasts generally hold true.
6. The A-team is so satisfied and happy that other development teams will approach us on their own initiative to ask for a similar arrangement.

Have we missed anything?”

The other members of the Emergency Power Team shake their heads, although Anne adds: “We have picked staff satisfaction as a criterion several times already. If we want to be able to measure it, we now need to start monitoring it. We need a reference score. HR has some experience with this. Marc, would you like to help?”

Marc: “Sure, but let’s have a breather—I need a coffee.”

The coach starts again: “We have defined a number of short-term goals. I would like to ask you to think of the vision as well: Improving customer satisfaction fivefold, staff satisfaction by 300 %, reducing our reaction time to 2 months, increasing productivity by another 300 %, and not making anybody redundant. Our short-term goals do not fully achieve these. Customer satisfaction, for instance, is only just touched upon by our attempts at creating sounder prognoses. Let’s use our short-term achievements to make sure whether our bigger vision is actually feasible.”

People start speaking to each other, until Peter brings it to the point: “I was so focused on our short-term goals that I completely lost sight of the bigger picture. Good thing that you keep that perspective in mind.”

Stephen agrees: “My words exactly. We have another week to reach our first short-term goal. But we definitely still need a working development environment for our A-team. Hannah, how can we help you?”

“Don’t worry. The servers have been ordered, and the licenses for the development environment. The only thing we can do is wait.”

“Wait?” Claudia is annoyed. “If we don’t get a new development environment before Monday, the A-team will just be sitting there, twiddling their thumbs. We can’t let it come to that. Can we ask for express delivery?”

Stephen is already on the phone to call procurement. After a brief conversation, he hangs up with a worried look on his face. “Well, that’s not good. They just told me that these things cannot be speeded up. They’ve tried already.”

Blank faces all round.

Finally, Hannah has an idea: “Well, if we shuffle some virtual machines around and if the environment does not have to be the most powerful one, it might work still. A number of other systems might suffer, but we can switch over immediately when the new servers are here.”

“What do you mean by ‘full power’ and ‘other systems’, Hannah?” Stephen seems slightly nervous.

“Emails would be sent and received less frequently. Our workflow system would be obviously slower. And the likelihood of crashes would increase”, the head of IT operations comments. “Checking source code in and out would also take forever for the A-team. But we cannot know how much performance we’re losing until we actually try it.”

The team discusses the pros and cons, but decides in favor of trying the proposal. After all, slow work is better than no work at all. Stephen informs the workforce in an email about the possible problems and uses that opportunity to mention the urgency of the situation and the vision again. Hannah and her team then get to work.

The end of the week arrives. The A-team has started its first Sprint, is based in its new rooms, and manages to work with the interim equipment. Surprisingly it does not actually notice the lost speed in its everyday work, as people state that the new development environment seems faster and better than the old one. The A-team seems to be proud of its special role and of the attention it is enjoying. The team members seem to see it as a privilege that they can work full-time on one project only. They are also not above rubbing their special status in their colleagues’ faces. With a note of amusement in her voice, Christina comments at the start of the guiding coalition’s workshop: “At least everybody knows now that we have a Scrum Team that is actually working. We cannot complain about a lack of visibility.”

“That’s correct, but we need to communicate it properly”, Stephen suggests. “I guess that the A-team tends to exaggerate from time to time. I’d like to put it in the right context.”

“Good idea”, the coach agrees. “But first let’s check our quick-win criteria. Christina has shown us that they are visible. What about them being unambiguous? Is there any reason to doubt our achievements?”

Marc weighs up his answer first: “The team is able to work, but it is working with an interim solution in terms of the equipment.”

The coach does not let go: “Correct. So, we’ve missed our target?”

“Not if you take it literally”, Marc replies. “But in spirit, yes, we have. As long as we don’t know whether the team is slower or faster than before, our critics will always find a fly in the ointment.”

“There’s no way we can change that anymore”, Hannah says. “The new hardware will be delivered. They promised it for Wednesday in 8 days’ time. We can at least start the next A-team Sprint with the full computing power.”

The coach takes over again: “Is our success transparent and tangible for all employees?”

Marc laughs: “Well, they just need to come downstairs and see us at work!”

“That’s right”, Peter admits. “Did anybody come yet?”

Marc is surprised by the question. “No, we only had two people coming in, but they came to see us about a question.”

“Maybe we should give them an official invitation?” Anne suggests. “People might be more willing to come if they do not feel like they might be intruding.”

“Good idea. I’ll speak to the A-team.”

The coach takes over again. “Is this success related to the vision or the strategy?”

Sarah explains: “We want to achieve all of our targets, like productivity boosts, customer satisfaction, and so on, with Scrum. But we can only do that if we actually introduce Scrum—and that’s what we’re doing.”

The coach accepts this. “But did you make this clear to others?”

All eyes turn to Stephen, who looks back at his peers and seems slightly embarrassed. “Well, not yet, really.”

“Well, let’s get going”, the coach encourages him. “With organizational change, nothing matters more than good communication. Even tiny wins need to be made public to have a real impact.”

“How about all of us taking this back to our departments?” Claudia suggests. “That’s more personal and direct than an email.”

Hannah replies: “I’d still go with the email as well. Then, our people have something they can read and come back to.”

“Let’s just do both”, Stephen says to stop the discussion in its tracks. “We’ve learnt that we need to communicate on as many channels as possible at the same time.”

Anne intervenes: “Well, we can go on then if that’s all settled. I really, really want to tell you about our survey! Marc and I worked with the other HR folks to produce this questionnaire. A single page was enough. We concentrated on questions that people could simply answer by ticking a box on a scale, just from plus to minus. I took that questionnaire to the A-team to complete, so we could use their feedback to improve it. This morning, I went around and put a questionnaire in everybody’s inbox. We’ll have data for the entire organization, and we’ll see how the A-team measures up.”

“What a great idea!” Stephen is pleased: “Do I have one waiting for me?”

“No, yours is right here!”

She distributes the questionnaires to all members of the group and adds: “Let’s see your answers. After all, we are all employees as well.”

Claudia is the first to finish and uses the moment to say something: “Seeing that we are distributing good old printed paper, I’ve got something for you as well.”

She literally jumps out of her seat and gets flyers and a poster from her bag. “Let me introduce you to: Our vision!”

The other members of the Emergency Power Team are excited and gather around her to take a closer look at the colorful images. There is an immediate, positive discussion about how the posters could be improved further. In the end, everybody seems happy. Claudia ends the workshop: “I’ll be revising the flyers and the poster today. Then, we should be ready for the review.”

15.8 Consolidating Achievements and Initiating Further Change

The Sprint Review of the Emergency Power Team again begins in the company’s cafeteria. Around 50 people are present and interested in the goings-on, including the entire A-team. Stephen begins by reiterating the urgency and the vision before stating the purpose of the Sprint in his role as Product Owner:

1. Expanding the facilities for the A-team—13
2. Scrum training for the developers on the A-team—1
3. Training to improve customer focus among the A-team’s developers—1
4. Expanding the development environment, including servers—13
5. Actively supporting the A-team in Scrum matters—3
6. Procuring two laptops that are suitable for the developers’ work—5
7. Producing a flyer to communicate the vision—3
8. Producing posters to communicate the vision—2

Stephen explains these points in more detail: “Our ambition was to set up a team in our most critical project and get them to work according to Scrum principles. You have just met this team. It picked its own name. If that answers your questions for now, let me now explain what we have achieved in the Sprint.”

Stephen continues to describe the activities that were taken to develop the facilities for the A-team. Faced with constant questions, he realizes that speaking about a room is not as good as seeing it in person, so he simply moves the Review to the new A-team’s room. The attendees are less interested in the official equipment than in the plants and posters that litter the room. The absolute highlight is the football table, which attracts lots of attention from the developers.

“Where did you get this?”, a flabbergasted Stephen asks.

Marc has to smile: “The janitor had it lying around, gathering dust. We got it back from the storeroom, and its been a top attraction at lunchtime ever since. It’s absolutely great for the team, and it gets your mind off things. Come and have a go yourself during the next lunch hour.”

“It must have been 20 years since I played it last. . . But I’m not saying no.”

“Let’s get on with it. So, we’re here for the Sprint Review. This room is the first Product Backlog item that we have finished. If anybody has any improvements to suggest, now is the time to say so.”

There are indeed a number of ideas, but it is soon obvious that the discussion should involve the A-team. The A-team itself keeps quiet—after all, they were

involved in all the work during the Sprint itself. Stephen therefore takes the decision that all suggestions can be brought to the A-team, who can then pass them back to the Emergency Power Team whenever they see fit. Since Marc is in charge of the A-team as Scrum Master, there is a direct line of communication. Stephen continues: “Our second point was conducting Scrum training with the A-team. Tell us: Did you finish that?”

One of the developers confirms it. “We’ve learned a lot, although we only had time to just mention some aspects like emergent architectures or test-driven development. None of us has any experience with these, so another 1 or 2 days of training might be worthwhile.”

Stephen makes a note of this on a colored post-it. “Okay, I’ve got that. We’ll have a look when we plan our Sprint.”

He looks at his notes again and goes on: “The next item was our customer focus training. Did that go well?”

“No. We didn’t have any. We don’t need it, because none of us has anything to do with clients”, one of the developers admits.

Stephen responds immediately: “You have nothing to do with clients YET. That will change. Even if you never meet a client in the flesh, our clients will be using your products. That means it’s only good and proper to keep that weird creature, the client, in mind from time to time. After all, he’s paying your salary in the end.”

Claudia butts in: “Honestly, the trainer simply wasn’t available at such short notice. We do have a course, but it’s only in 8 days’ time.”

“Okay, so we’ll record this as not done. We should keep it for the next Sprint. How about the development environment?”

“That’s great!” The A-team is full of praise. “Fast, fully integrated, and much better than the old one.”

Stephen is surprised. “Are the new servers online?”

Hannah admits: “No, not yet. But that will happen in 2 weeks, as planned.”

“So, is this done or not done?”

The coach intervenes: “You put down some hard criteria for completion. We don’t know yet what’s happening in terms of performance or with multiple teams. We are, in a sense, working with an unfinished prototype. My recommendation is: Define that item as not done and carry it over into the next Sprint.”

Stephen looks at Hannah, who is nodding along. “I agree. It’s working right now, but that’s more or less by chance. We’re not really finished yet.”

“Pity. I had hoped to get that issue out of the way.”

The Review continues. The laptops have not arrived, but the flyers and posters are ready and are immediately handed out to the attendees. The ideas and suggestions of the group are recorded and passed to Claudia for evaluation. Stephen ends the review: “Claudia, you check these ideas, and then we’ll sit down together and rate them. We cannot decide now whether or not we need to change the flyers or posters for the next round. So, what’s left? Calculating our velocity. Let’s see. . .”

A few seconds of calculation later, he arrives at a velocity of 22.

“Hey, that’s almost twice that of the last sprint!” Hannah seems pleased. “We only had 13 then.”

Stephen's face suddenly lightens up. "You know what: You are right. 22 is not quite double 13, but it's a great step up. Seventy percent more. Wow!"

To celebrate that achievement, the Emergency Power Team challenges the A-team to a round of table soccer (and loses, naturally). After that, the team returns to its favorite meeting room and starts planning their next Sprint.

The work progresses well for the next 4 weeks. Both the Emergency Power Team and the A-team are cooperating more and more and are enjoying their work. The A-team even manages to produce a working piece of software in the very first Sprint. A new, albeit minor feature is added to the product, which is a visible sign of the vision working. In the second Sprint, the A-team even achieves a twofold increase in its productivity. More mean-spirited colleagues suggest that this is only due to the many training courses and all the preparation work in the first Sprint, but the coach reminds everybody that the team's speed will only reach its final and stable level after about three Sprints. Anything before that is not a reliable indicator. He calls another workshop for the next day.

Hannah and Sarah arrive about 10 min too late, because they had some more urgent business to attend to. The coach uses exactly this excuse as the point of the workshop. The problem is that the sense of urgency is lost, often by imperceptible little steps. Whenever that happens, the entire change process begins to slow down. Everyday routines start taking over, and people start to return to their old ways. "Ask yourself: Have there been any situations in the last 2 weeks, in which you could have acted differently?" Every team member can immediately think of one or two occasions. Sometimes, sheer laziness stopped people from doing as much as they could have done; sometimes, less important issues were given priority. The reasons vary from case to case, but there are two essential forces at work: Change fatigue and the lack of energy it brings, or the belief that no more work is needed—i.e. a loss of the sense of urgency.

"That happens to all of our people", the coach explains. "Anybody will be affected by a certain fatigue, sometimes sooner, sometimes later. And then they lose that sense of urgency. We need to keep bringing in new facts and figures that remind us of the urgency, and we need to make every step in our strategy transparent."

He stops for a minute to let this sink in. "We are the guiding coalition. We must not let up. It is essential that we stay on the ball. If we do not take the lead and embody what we want to achieve, nobody has any reason to follow us."

This time, the team members wait in vain for the coach to continue. Marc takes a deep breath: "Guys, staying on the ball might be a problem. I had an offer from another company yesterday. They are offering me far more money, and I could continue working as Scrum Master, with the prospect of becoming internal Scrum coach in 2 years. It's enticing, I must admit."

Stephen is completely taken aback by this and is struggling with keeping his composure. "Marc, I have to say, I am surprised. Aren't you happy here anymore?"

"I am", Marc tries to justify himself. "But the prospects are better, in the end."

“If you leave us, we’ll be losing our internal Scrum expert and a great Scrum Master. It will rip a hole into our guiding coalition, and we can’t simply plaster over that. It will endanger the entire project.”

Stephen stops to catch his breath and then makes Marc an offer: “Marc, listen to me: You stay for another year. Then, we’ll know whether we can turn this thing around. We can discuss your salary, but we’ll do that in private. If we can get out of this hole, the entire organization will go Scrum. And then we’ll need a Scrum coach ourselves. And frankly, if we fail, you will have one more year of invaluable experience and you’ll be even more attractive for other companies. What do you say?”

“Okay. That sounds okay. Let me sleep on it, and we’ll see”, Marc agrees. This is the point for the coach to start speaking again: “Resignation, retirement, and promotion are the mortal enemies of change. We need to stand up to them. What would you think about making a risk analysis regarding this topic?”

The idea is received enthusiastically by the group. New risks are found immediately. It is much harder to find mitigation strategies or other responses. The strained finances of the company do not leave much room for monetary incentives. In the end, the group has to accept the promise of every member of the Emergency Power Team to stay for at least another year.

“These points are not the only things that could cause our key actors to drop out. There’s also illness or exhaustion”, the coach continues his train of thought. “I won’t use the term burnout, because that has become a fad. But there is the undisputed fact that too much stress will make you sick. What about these risks?”

“Nonsense!” Thomas blurts out. “There is no pressure, and we are working at a sustainable pace. Nobody is burning out.”

Anne takes a long look at him: “Thomas, have you got any data for that statement?”

“Well, it’s just obvious”, the quality manager says in his defense.

Anne’s gaze becomes even more intense: “Since we started with our change efforts, absenteeism has indeed gone down by 6 %. But that’s no reason to celebrate. Three years ago, when the company was still doing fine, we had an average of 7 sick days per person per year. Today, it’s 15, and that’s after the improvements.”

Stephen steps in: “Careful, let’s not lose focus. I believe you meant the risks for the guiding coalition, not for all employees, am I right?”

His look is directed at the coach, who raises his hands in defense: “Yes and no. It’s about the consequences of change. Of course, that’s primarily for the key actors. But the key actors are not only here in the Emergency Power Team, but spread out across the entire organization. How about your sick days, by the way?”

Anne looks at her laptop and analyses the figures, while the other colleagues are thinking. It is obvious that people are not sure whether they are off more or less than before. Anne finally solves the issues: “By comparison to 3 years ago, we have 10 % more sick days. By comparison to last year, there’s no change.”

“Okay”, the coach goes on. “So we know three things: First, fewer people were off sick before the crisis than today. Second, the change projects have not yet led to

any more absenteeism. And third, this applies to everybody, including the Emergency Power Team.”

He gives his statement time to sink in, before continuing: “That’s great news, but there’s also a lot of risk in it. We know that the entire workforce is working under a lot of pressure. If we make a mistake and add to that pressure, we might be facing a complete breakdown.”

Hannah agrees: “Our colleagues have to want to change!”

“And we need to take better care of ourselves”, Stephen adds. “We must never work ourselves into the ground, because the change needs us as its engine.”

“People, I have to say that I am proud of you!” The coach is beaming with pride. “There are so many companies that never get these mechanisms. Let me sum up: Organizational change lives or dies depending on the people leading it. If they fall, the change will most likely fall as well. These people need to be committed to the company, which purely financial incentives will rarely ever achieve. And the pilot teams are also key factors. We need to make sure that it stays intact. It is the seed that will grow into the bigger change. And, remember, people are watching it.”

He ends: “We have learned something new today. We must never be lazy or self-complacent. We have to remember how urgent our mission is. Otherwise, the entire efforts will go to waste.”

With that in mind, everybody leaves for a long lunch.

Refreshed from the lunch, the coach comes in with an unexpected idea. “Let’s take a walk. We can talk about how we continue then.”

“Why should we ever do that?”, Thomas seems unconvinced. “What’s wrong with our meeting room?”

“Oh, come on”, Anne interrupts. “Walking is healthy. A breath of fresh air has never done anybody any harm.”

The team gets up and leaves. After about a mile, it reaches a hillock with some trees and low shrubs, which offers a great view of the company’s headquarters. Everybody gets in a half circle and looks at the building in the distance.

Stephen is the first to speak: “It is strange, really. All these years, and I never came up here. I’ve never seen the company from this viewpoint.”

“That’s my point”, the coach responds. “If we see the company from a different vantage point, we might also change how we see our problems.”

The team enjoys the view for a few minutes. Everybody is lost in thought when the coach continues: “So, do you think the change will succeed, seen from up here?”

Marc: “It’s going well right now. Better than I had hoped. I think we have a 100 % chance of success.”

Christina is not so sure: “Marc, I think you might be a bit over-optimistic. We’ve only started a mini project, the A-team. To save all that down there, we need to change every part of it. And that means 20 projects, there and then. It’s a completely different dimension.”

Thomas groans: “Goodness, how do you expect us to cope with 20 projects? That’s just impossible!”

Stephen shakes his head vigorously: “Not impossible. Tough, but not impossible. We won’t manage it if we go it alone.”

“So what do you expect us to do? Hire people for the change project? That never works.” Doubting Thomas remains true to his name.

“You are right about that”, Anne admits. “But we’ve actually got everything we need: A guiding coalition that’s actually guiding things. And people who want to help, who want to save the business.”

Thomas is not convinced: “People can’t do it on their own either. We constantly need to make sure that the milestones are there, else it will all go to waste!” Stephen turns to the group. “Let’s not argue. We need to cope with the people we’ve got, because there’s no cavalry coming to save us. And remember: These are the people who raise children, build houses, and coach football teams—Why shouldn’t they manage other tiny projects like that? We have to remember that leading and managing are two different things. Even with 20 projects, we just need to make sure that everybody knows where we want to go and that everybody wants to get there as well. That’s leading. We cannot afford being at their side for every step along the way. We need to trust our people. Our job is to empower them to live up to what we ask of them.”

The coach agrees: “You’ve hit the nail on the head! If you want to be successful, you need to trust your people and let go of some of your responsibilities. Even if that’s hard. Otherwise, you’ll just wear yourselves down trying to control every detail.”

This leads to a brief discussion about the right time to involve other people. At the end, everybody agrees: Better sooner than later.

Stephen has a suggestion: “What do you think about starting a competition? All teams who want to go Scrum need to apply. The best and most original application wins, and the winner can start as the second pilot team.”

“Great idea!” Everybody is convinced, and the group makes its way back to the company to start working on that idea.

Back inside, Marc begins: “Well, we will need a Scrum Master, a Product Owner, and the right equipment for the second pilot team as well. And a room for them to work together.”

“That’s the hard part”, Sarah says. “We simply don’t have any rooms left that are large enough. And all rooms are occupied already.”

Marc suggests: “Let’s tear down some walls. The old places of the A-team must be free now?”

“It’s not as easy as that”, Sarah hesitates.

“Why not?” Marc stands up to draw a layout on the whiteboard. “So, we just need to shuffle these five people around, remove that partition, and we’re done. Easy-peasy.”

Stephen sees the problem: “Well, these five people include two managers. We have a culture in which personal offices are status symbols. There have been lots of conflicts in the past, just because somebody moved a chair—What do you think will happen if we take away somebody’s entire office? But you’re still right, Marc. We’re not here to mollycoddle people’s egos, but to save our company. Let’s start with us. Would there be enough space if we open up my office?”

Stephen struggles with calming the resulting fracas. “Listen! Please! We’ve all heard that we need to lead by example. How can you expect our people to give up their offices if we fight tooth and nail for our own? We need to practice what we preach. I’ll be the first to tear down these walls, in every sense of the term.”

The debate flares up again, but Stephen stops it immediately. “However”, he goes on, “we need to plan carefully. We’ve got files that need to go somewhere; we need meeting rooms where we can receive visitors in private. And so on. I think we’re talking about a completely new layout.”

Anne speaks up: “Stephen, I don’t know if that’s going to be as simple as that. But we have a specialist for this on my team. I’ll get him involved.”

The discussion continues for a few minutes before the coach intervenes: “We have just found our first systemic problem. Let’s play this out a bit more. Imagine what would have happened if you hadn’t noticed it. First, people would have been set against the changes, because they’d lose their status symbols. Then, they’d be working against their colleagues who avoided such losses. In the end, all executives would be working against each other, just to avoid losing more such symbols. Everybody would care about this and this alone. If we changed something at one end, it would affect the other end as well, without us wanting that. But it’s all hypothetical.”

There is a general murmur of agreement in the group.

“The further we get ahead, the more such systemic problems we will encounter. Very few will be as obvious and straightforward as the room issue. Keep your eyes peeled!”

With these words, the workshop ends, and everybody goes back to work, preoccupied with what they have just heard.

15.9 Embedding New Concepts in the Corporate Culture

The work of the guiding coalition continues. During every Sprint, all of its members are working hard for the success of the pilot team and the survival of the company, with the coach adding essential support. Six months later, the team has overcome such problems as the qualifications of the developers, the layout of the rooms, the retention of the key employees, and the delegation of responsibilities. New problems have been encountered, have been solved or are still being worked on. All in all, the Emergency Power Team is working well and effectively. At one of its workshops, the coach praises the team: “I am so proud of you. You are doing excellent work. If we go on like this, the company will be saved. We’re making headway. You are about to enter the last stage of the change process, so I’ve prepared something new for today. An organization changes in eight stages. We’ve come through all of them so far, and now we are at the last stage: Incorporating the changes for good, so that people would not revert to old habits even if you weren’t here anymore.”

The coach goes on to explain more, making much use of the whiteboard to make his meaning more obvious. He explains that a company’s culture is something

intangible that works on many different levels. People's practices are driven by a sense of doing the right thing. This sense, in turn, is the sum total of all people's values and personality traits. If these values are used successfully, a culture can be solidified—among all people, not just individuals. Since such a culture is intangible, however, it is not normally seen or treated as the cause of failures or setbacks. In order to break that barrier, the culture needs to be replaced by a new one, which only happens by reaching people's hearts and minds.

Hannah speaks up: "Did we ever do anything that went against our culture? I guess we haven't changed anything."

The coach does not allow any debate: "Thanks, Hannah, that's a good point. I would like you to take your post-its and to write down all things that either go against your old culture or are new elements in it. Remember: Whenever things got hard or conflicts arose, it's worth to take a closer look. Often, that's where the cultural change is invisibly at work."

Everybody gets to work. The unusual task makes for slow progress at first. After 10 min, the group checks its results, with an interesting outcome:

- Status symbols have been removed/Room layout has been changed.
- It is okay to make mistakes and learn from them.
- More openness for new ideas.
- Absolute transparency about all activities.
- Direct involvement of employees.
- Lots of leadership and not as much management.
- Developers can take decisions themselves.
- Punctuality matters.
- More staff orientation.
- Working towards a common goal, instead of against each other.
- Working can and should be fun.
- Caring about quality.
- Strong customer service mindsets.
- Individual heroes count less than teams that work well.

"Well, that's quite a bit", Stephen says. "I'm sure that we could find even more if we tried."

"You're right", the coach agrees. "Our job as the guiding coalition is to identify all points. We need to make people see the old ways and the new ways and show them how we got from there to here. Above all, it's important to know what led to the old cultural elements and what advantages the new ways bring."

The group is annoyed. "Oh boy, I knew this would end in a lot of work. Damn!"

The coach smirks and adds another point to the list:

- We all love extra work if it helps us reach our higher goals.

The groaning intensifies, but the coach continues: “When we’ve made that transparent, everybody can face up to the pros and cons of what we are trying to do. Everything we have done to date is only the groundwork for that.”

Peter blurts out: “You mean, we’ve only been working on the culture?”

“Of course not”, Stephen is quick to add. “I think what our coach is trying to say is that the achievements we have had make our new concepts more credible in the first place. If we had not done what we’ve done, people wouldn’t listen. Am I right?”

“Right”, the coach admits. “Now that we’ve shown people that the new ways work better than the old, we need to make sure that all people are with us. That can be quite unpleasant, because we need to change how we promote or recruit people to match these new values. We can only promote somebody if he or she actively stands up for the new values. That can mean that some very qualified and long-standing employees have nowhere else to go. But it’s essential, and we must not whitewash the fact. Otherwise, you are telling people that the values and your decisions are up for negotiation. Stay firm, even if it’s hard.”

Stephen takes a long look at the team: “It is hard. Damn hard. I can immediately think of two people who’ve been promised promotion, but who are not actually involved in our change efforts. So, they are out of the running, I guess.”

He seems to take this realization hard. Anne tries to help: “It might not be as bad as that. If they realize that they need to get behind the new values, they will, I think. Very few people will actually resist.”

The coach takes over again: “Correct. As long as everybody knows what’s expected, people are free to decide how they will act. Help them, because it could be that they find it hard to square the new values with their own personal beliefs. By the way: This also goes for succession management. Every candidate has to be 100 % committed to the new values.”

The members of the guiding coalition discuss the impact of this new information for a few minutes before the coach closes the workshop: “I have told you everything you need to know to save your business. There won’t be any more surprises from my end. All you need to do is to keep introducing new change projects until the new values and ideas are really a fixed part of the corporate culture. When that has been done and done sustainably, you are there. It can take years. I’m here if you need me, but I don’t think you will. What can I say: Good luck!”

15.10 Beyond the Case Study

Let’s consider the simulated company’s fate:

Two years have passed. Business is going strong. It has long come back from the red, and its customers are back to their usual satisfied selves. There are regular postings in the cafeteria, which relate the new business figures to the values developed at the time of the crisis. The current customer satisfaction index shows that there are now only 30 customers leaving per month, compared to a full 250 in the worst period. Staff fluctuation is down from 15 % to a sustainable 5 %. The fact that no redundancies were needed even during the crisis

helped enormously here. The company can now respond to customers' request within an average of two months, and some urgent cases are developed within a mere four weeks. Although more complex orders still take longer, that figure used to hover around the nine-month mark only two years ago. All of these great figures are accompanied by one term in bold red letters: Velocity. The figure next to it immediately shows how far the company has come since the crisis: Its productivity has increased almost fivefold. Stephen gives the coach a hearty pat on the back: "There's a long road ahead of us, but the company is safe now. We do still see that some people don't get or don't follow our new values. The Emergency Power Team has lots of work to do, but I don't mind. It would be a pity to stop working with these people. We're like a family. I never enjoyed my work as much as I do now. Thank you for that!"

The result seems perfect, and it was not long in the making. Even productivity has increased fivefold. Too bad that this case study is a product of imagination and hypothesis. The approach and the results do reflect the experience in actual real-life projects though. Naturally, no two change initiatives are ever alike, but the basic approach described in this case study can be found in most of them. The biggest difference to real life is the general absence of resistance and problems. With the single exception of Larry, nobody is resisting the proposals; the members of the guiding coalition are all pulling in the same direction, and everything they do ends in success. In actual practice, one should expect one or two hurdles along the way. The characters in this case study are exaggerated archetypes as well: Stephen appears as some kind of wise and benevolent leader, who guides his knights in shining armor to a final victory—without fear or favor, and always with the right solution in his pocket. The coach also seem to be beyond mistakes. He remains in control and can resolve any conflict. Coming without a name, he has a certain mystical appeal, as if he had come down to Earth from the distant planet “Scrum” to save the company. Larry, by contrast, is so stubborn that one could only hope to never meet a Larry in real life. His stubbornness allows an uncompromising black-or-white view of the situation with an unambiguous solution.

You should never expect to find such a group of heroes in real life. It is much more likely that you will be encountering normal human beings with all their faults and weaknesses. This is a good thing, as it breeds creativity and produces better solutions than unchallenged harmony. Adjust your expectations accordingly and enjoy the most exciting project of your career: Changing the organization around you.

15.11 The Actors in Brief

When writing this book, it helped me to keep a list of all the active participants in the case study, which includes no fewer than 11 unique characters. I am attaching some details about the participants to help understand who's doing what:

The members of the Emergency Power Team:

Peter is the head of development at the company. He was the first to realize and draw attention to the crisis. Introducing Scrum was his idea in the first place.

Stephen is the CEO of the company and a decision-maker with a can-do attitude. He has lots of experience and likes to take quick, but careful action.

As the head of HR, Anne has a special sense for the needs of her people. Her empathy and her analytical abilities often help the group along.

Marc is one of the company's software developers. He has some experience with Scrum and wishes to become a Scrum Master.

Sarah manages the controlling team. Reliable figures and a critical sense make up her worldview, which can come in quite handy for the team.

Claudia is a product manager who knows the ins and outs of her products. She also has a good connection with her clients.

Thomas is the head of quality management. Critically minded by nature, he is not the person who would welcome innovation wholeheartedly. His loyalty and his experience are great assets for the company.

Christina is a member of Thomas' team, where she is responsible for process development. She knows all processes of the company like the back of her hand, and she has an acute analytical mind.

Hannah is the head of IT operations. She makes sure that all computers and other IT systems are working properly. As a result, she cares most about a smooth operating process, which is reflected in her conciliatory nature.

The coach is left nameless to give him a certain aura of mystery. He is an expert for Scrum and organizational development. He teaches the guiding coalition how to approach the change effort.

Other players who are not part of the guiding coalition:

Frederick is the head of the legal office. He helps the Emergency Power Team as good he can, although he is not part of it. His work revolves around law books and case notes, not around product development.

Larry is a software architect working under Peter. He is a natural loner who thinks he is irreplaceable. Unfortunately, all attempts at winning him over for the idea come to nothing. The situation escalates, and Larry leaves the company.

Reference

Kotter, J. (2012). *Leading change* (New ed.). Boston: Harvard Business Review.



You have learned about the different shapes of Scrum. You know how you can get to these different target states. Furthermore, you are aware of the eight steps you have to take from an organizational development perspective in order for your Scrum introduction to succeed. The case study illustrated the practical application of what you have learned. The comments in the appendix regarding roles, artifacts, events, and methods may give you some more food for thought.

Now it is your turn: Get down to your Scrum project and lead it to success!

I wish you nerves of steel, patience, and a clear vision, but also fun, companions as bold as you, and a little bit of good luck.

If you liked this book, I look forward to receiving your feedback. As always, I welcome suggestions and constructive criticism. You can contact me via my website <http://scrumorakel.de>.

Part V

The Scientific Part of the Scrum Culture



As explained in the preface of the second edition, many professional readers found it too difficult to work their way through the scientific descriptions. Therefore, I moved them down to here, where you now find them for yourself in this book. The academic reader will locate the full study description here and get the additional benefit of being really close to the appendices, which hold all the statistical analyses, questionnaire details and more. The only part that is missing, are the main conclusions. These are to be found earlier in Sects. 1.3 and 1.4.

17.1 Problem Definition

Although the understanding of Scrum from a “corporate culture” point of view is highly important, as described above, there have only been minor attempts to clarify this question. Michael Spayd (2010) analyzed the culture of agile methods (including Scrum) with a ten-question questionnaire, positioning “agile” in the “collaboration” quadrant of William Schneider’s (1999) corporate culture model, with strong influences from the “cultivation” quadrant. Spayd even referenced certain “cultural levers”, which correlate certain characteristics of culture, to specific cultural classifications.

However, as of today the methodology has not been fully disclosed and the questionnaire has not been made public. Therefore, these results are of limited scientific use. So far there has been no other attempt to classify “agile” or “Scrum” into a specific corporate culture model.

A multitude of scientists have created corporate culture models. Examples include Harrison (1972, 1987), Schneider (1999), Deal and Kennedy (2000a, b), and Cameron and Quinn (2011). These models are very different, specifically in the reduction criteria upon which they try to simplify the complex topic of organizational culture. Some authors like Martin (2001) and Schein (2010) have elaborated on that and tried to work out the general concept of corporate culture. There is no dominant “right” or “wrong” model. Most authors point out that there is never a

corporate culture fitting any model 100 %, but there usually is one dominant culture, permeating all subcultures (cf. e.g. Harrison 1972; Schneider 1999). In order to find such a “dominant” culture inherent to Scrum, the existing models first have to be analyzed and compared. A more limited number of models then have to be chosen for further research.

It has to be mentioned that national and industrial culture are also relevant. Ken Schwaber and Jeff Sutherland, U.S. Americans working in the IT industry, invented Scrum. It is probable that this cultural context had a major impact on Scrum’s nature since national, industrial and organizational cultures are interdependent (cf. Fayolle et al. 2010, pp. 708). In addition, it is not obvious if Scrum belongs to the category of industrial or organizational culture. I leave it to scientists to research those aspects. For the remainder of this book, the origins of Scrum’s inventors are neglected. The organizational culture stance is assumed since many companies from different industries can apply Scrum and it impacts primarily companies, not industries.

Some authors have pointed out specific implications of Scrum on corporate culture (e.g. Cohn 2009; Gloger and Häusling 2011; Schwaber 2004). However, there has not been any attempt to summarize these implications and formulate them into a cultural definition. Instead, the professional discussion largely revolves around the Agile Manifesto (Beck et al. 2001), without going into depth. This is neither helpful for management professionals who want to understand the impact of a Scrum introduction, nor does it move the scientific community forward in understanding the difference between “Scrum” and more traditional product development approaches. After all, the starting point for every cultural change effort is to know where you are and where you intend to end up. Therefore it is essential to analyze the nature of Scrum.

17.2 Goals of the Research Project

The goal of the research project that provides the base for this book was to analyze and summarize the nature of Scrum, from a corporate culture point of view. This will help enterprises to globally determine whether the introduction of Scrum can be expected to have a negative impact on their current culture, or whether it will be a smooth, rapid and successful transition. In addition, the results from my research project will help international researchers to better understand and culturally classify Scrum. Other aspects (e.g. motivation of employees; reasons for a Scrum introduction; productivity; product quality) are not considered in this work.

In order to establish a corporate culture view on Scrum, all major existing corporate culture models first have to be identified. It is essential to analyze these models and to determine the significant characteristics that are inherent to a specific culture. A classification of Scrum is only possible if specific characteristics can be identified, because existing corporate culture models tend to overly simplify. This results in two questions:

1. Which major corporate culture models do exist?

Once the existing models have been analyzed, the focus moves onto Scrum. The cultural characteristics found in other models might lead to conclusions for Scrum, so a second question must be answered:

2. What cultural characteristics are inherent to Scrum?

When questions 1 and 2 have been answered, Scrum can be classified and described from a corporate culture point of view. This will help to answer a third, additional question and produce the anticipated benefit of this work.

3. What is the classification of Scrum from a “corporate culture” point of view?

17.3 Scientific Approach

Scientists have researched the topic of organizational culture since the early 1970s (e.g. Harrison 1972). During the 1980s and 1990s, this research was intensified (e.g. Harrison 1987; Schneider 1999) and several models were created. Even though the topic of organizational culture is very complex and these models only represent a simplified and incomplete view of this complexity, a literature review of such models was essential. As a starting point, Harrison (1972, 1987), Schneider (1999), Deal and Kennedy (2000a, b), and Cameron and Quinn (2011) were examined. Other authors such as Martin and Schein supplemented the research on organizational culture models.

Since the early years of the twenty-first century, software professionals have tried to pinpoint cultural implications of Scrum. Although it is clear that none of the authors has produced a comprehensive cultural typology of Scrum, additional literature review was necessary to understand what different views on cultural factors already existed. This secondary research will start with Schwaber (2004, 2007), Cohn (2009), Spayd (2010), and Gloger and Häusling (2011). Other authors will be added, if appropriate.

While this secondary research produced useful results, it consisted mostly of individual author’s opinions. Most authors writing about Scrum build their opinions on personal experience (e.g. Cohn 2009) and case studies (e.g. Schwaber 2007), not on a larger data sample. Therefore, a questionnaire had to be developed that helped to classify the nature of Scrum from an organizational culture point of view. This questionnaire was based on the findings from the secondary research and addressed the identified cultural characteristics. Therefore, a mainly quantitative approach was used.

Seasoned agile professionals answered the questionnaire on conferences and via an online form.

17.4 Expected Results

It was expected that the first secondary research would produce a multitude of different organizational culture models as well as different characteristics, which those models considered significant. This research also brought up different methods for diagnosing organizational culture, which were used to derive a questionnaire. All models, characteristics and methods are briefly described and summarized in this book.

The second secondary research identified many indications from professional Scrum practitioners about the cultural impact of Scrum. Those indications of impact were compared to the information already gathered about organizational culture. On the one hand, this extended the summarized results while on the other hand this helped classify Scrum into the examined models. It was expected that those results would be fairly incomplete and would not allow for a proper classification of Scrum. However, it did bring up a “corridor” of options that could be applicable to Scrum. The findings are presented in an appropriate way, both as tabular and graphical representations within the earlier identified corporate culture models.

The primary research (questionnaire) was expected to lead to deeper insights about the nature of Scrum. It allowed for an objective view on the topic. Not only cultural classifications were possible, but unique cultural characteristics were also identified. These final results are described textually in detail and supplemented by figures where useful. Certain cultural models that fit well are being displayed, including a graphical representation of Scrum’s culture according to these models. The expectation also was that if a new model had to be forged, it would be described and displayed as well.

In short, it was assumed that the final result of this study would describe the typical characteristics of the Scrum Culture, using one or more existing models of corporate culture. The goal was to help professionals and researchers alike to understand the nature of Scrum and to anticipate the implications of a Scrum introduction into their enterprises.

References

- Beck, K. et al. (2001). *Manifesto for agile software development*. <http://agilemanifesto.org/>. 01.01.2013.
- Cameron, K. S., & Quinn, R. E. (2011). *Diagnosing and changing organizational culture* (3rd ed.). Hoboken, NJ: Wiley.
- Cohn, M. (2009). *Succeeding with Agile. Software development using scrum*. Amsterdam: Addison-Wesley.
- Deal, T. E., & Kennedy, A. A. (2000a). *Corporate cultures. The rites and rituals of corporate life* (Revisedth ed.). New York: Basic Books.
- Deal, T. E., & Kennedy, A. A. (2000b). *The new corporate cultures. Revitalizing the workplace after downsizing, mergers and reengineering* (First Paperbackth ed.). New York: Basic Books.

- Fayolle, A., Basso, O., & Bouchard, V. (2010, November–December). Three levels of culture and firms' entrepreneurial orientation: A research agenda. *Entrepreneurship & Regional Development*, 22(7–8), 707–730.
- Gloger, B., & Häusling, A. (2011). *Erfolgreich mit Scrum – Einflussfaktor Personalmanagement*. München: Carl Hanser.
- Harrison, R. (1972). Understanding your organization's character. *Harvard Business Review*, 50(3), 119–128.
- Harrison, R. (1987). Harnessing personal energy: How companies can inspire employees. *Organizational Dynamics*, 16(2), 5–20.
- Martin, J. (2001). *Organizational culture: Mapping the Terrain*. Thousand Oaks: Sage.
- Schein, E. H. (2010). *Organizational culture and leadership* (4th ed.). San Francisco: Jossey-Bass.
- Schneider, W. E. (1999). *The reengineering alternative. A plan for making your current culture work* (Special reprintth ed.). New York: McGraw Hill.
- Schwaber, K. (2004). *Agile project management with scrum*. Redmond: Microsoft Press.
- Schwaber, K. (2007). *The enterprise and scrum*. Redmond: Microsoft Press.
- Spayd, M. (2010). *Agile & culture: The results*. 06.07.2010. Collective Edge Coaching. 01.01.2013. http://collectiveedgecoaching.com/2010/07/agile__culture/



This chapter describes the approach taken in my research. It shows the considerations taken, the approach chosen, and how the research was conceived and conducted.

18.1 Different Approaches

Organizational culture can be approached from different directions. Martin (2001) conducted some extensive research on the topic. She identified three theoretical perspectives in studies on organizational culture and named them “integration”, “differentiation”, and “fragmentation” (cf. Table 18.1). No perspective is in itself “right” or “wrong”. They all express different worldviews and have diverse advantages and disadvantages.

In short, integration studies focus on the perception that all mentioned cultural aspects are consistent and reinforce each other (cf. Martin 2001, p. 95). If deviations are found, they are seen as shortcomings that must be remedied. In contrast, differentiation studies “focus on cultural manifestations that have inconsistent interpretations” (Martin 2001, p. 101).

This means, “the integration perspective focuses on those manifestations of a culture that have mutually consistent interpretations. An integration portrait of a culture sees consensus (although not necessarily unanimity) throughout an organization. From the integration perspective, culture is that which is clear; ambiguity is excluded. [...] The differentiation perspective focuses on cultural manifestations that have inconsistent interpretations, such as when top executives announce a policy and then behave in a policy-inconsistent manner. From the differentiation perspective, consensus exists within an organization—but only at lower levels of analysis, labeled ‘subcultures.’ Subcultures may exist in harmony, independently, or in conflict with each other. Within a subculture, all is clear; ambiguity is banished to the interstices between subcultures. [...] The fragmentation perspective conceptualizes the relationship among cultural manifestations as neither clearly

Table 18.1 How three theoretical perspectives complement each other

	Integration perspective	Differentiation perspective	Fragmentation perspective
Orientation to consensus	Organization-wide consensus	Subcultural consensus	Lack of consensus
Relation among manifestations	Consistency	Inconsistency	Not clearly consistent or inconsistent
Orientation to ambiguity	Exclude it	Channel it outside subcultures	Acknowledge it

Source Martin (2001, p. 95)

consistent nor clearly inconsistent. Instead, interpretations of cultural manifestations are ambiguously related to each other, placing ambiguity, rather than clarity, at the core of culture. In the fragmentation view, consensus is transient and issue specific” (Martin 2001, p. 94). Generally, people working with an integration perspective have managerial interests in mind. Differentiation scholars are taking a critical stance and fragmentation researchers are not taking an explicit interest position (Martin 2001, p. 174).

Another aspect that has to be considered in analyzing organizational culture is that of specialist studies. “Specialist studies assume that one or a few manifestations can stand in for, or represent, an entire culture because interpretations of more types of manifestations would be consistent” (Martin 2001, p. 60). So while the integration, differentiation, and fragmentation perspectives describe the level of conformity sought for, the specialist aspect means that a few analyzed people or companies allow the researcher to make conclusions from that small data set and extrapolate onto a larger population like the whole company or industry. This stance poses the risk of overrating findings without having a truly representative data sample and thus drawing wrong conclusions.

Additionally, the method to gather data has to be chosen. Long-term ethnographies based on participant observation, short-term qualitative studies, textual and discourse analysis, and analyses of visual artifacts such as photographs are, according to Martin, counted as qualitative methods. Experiments, surveys, archival studies of large data sets, and content analysis (counts of categories of qualitative data) are considered quantitative research. The method chosen also has a huge impact on (and is sometimes impacted by) the type of study participants: “Whereas quantitative study participants are sampled so that they will be statistically representative of some larger population, qualitative study participants, called informants, are chosen because of their experience, lucidity, and willingness to talk openly with the researcher” (Martin 2001, p. 220).

The inclusion of those aspects is important because in general, methods choices, theoretical perspectives and interest orientations are correlated. “Quantitative studies usually assume the integration perspective and adopt a managerial orientation. In contrast, qualitative studies are more likely to assume differentiation or fragmentation perspectives and to adopt a more critical orientation” (Martin 2001, p. 234).

My study was written with a managerial interest and focuses primarily on the integration perspective, using mainly quantitative data. It also takes a specialist stance and assumes that from a relatively small data sample conclusions can be drawn to accurately describe the nature of Scrum. While some qualitative data was used as well, it was not the focus. As described above, the intended major outcome of this research was to find a culture model that accurately describes Scrum in order to help managers and researchers alike to better understand its implications. This is also reflected in the choice of literature: Harrison, Schneider, Deal and Kennedy, Cameron and Quinn, and Schein all take an integration perspective stance (cf. Martin 2001, p. 100). Following this focus, a suitable definition of culture had to be chosen. There are many different definitions available, where Schneider offers the most intuitive one: “Organizational culture is the way we do things in order to succeed” (1999, p. 128). This definition is used throughout this work.

18.2 Model Selection

A multitude of organizational culture models can be found in literature. One of the first who created a thorough model based on empirical data was Harrison. He defined four different “organization ideologies” (1972) and named them “power orientation”, “role orientation”, “task orientation”, and “person orientation”. In a later publication (1987), he used the term “culture” beside the term “orientation” and renamed “task orientation” to “achievement culture” as well as “person orientation” to “support culture” (cf. Fig. 18.1). Harrison defines a power-oriented enterprise as “an organization that [...] attempts to dominate its environment and vanquish all opposition. [...] And within the organization those who are powerful strive to maintain absolute control over subordinates” (Harrison 1972, p. 121). A

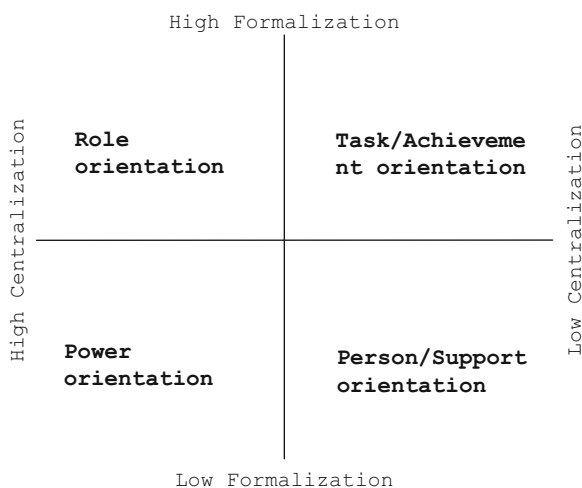


Fig. 18.1 Harrison’s culture model (Based on Harrison 1972, 1987)

power-oriented organization is further described as “competitive and jealous” (*ibid.*, p. 121); compliance is more highly valued than performance.

Power orientation can be found in companies with a background of family ownership or which are newly founded.

“An organization that is role-oriented aspires to be as rational and orderly as possible. [...] Competition and conflict [...] are regulated or replaced by agreements, rules, and procedures. [...] While there is a strong emphasis on hierarchy and status, it is moderated by the commitment to legitimacy and legality” (Harrison 1972, pp. 121–122). This means that in both the power- and the role-oriented enterprise all power is centralized, but while a power-oriented company exerts this power on a personal level, the role-oriented company has highly formalized processes and work instructions to apply this power. Harrison states, “most organizations we know, live with, and work in are a combination of the power-oriented and role-oriented models, with larger organizations tending toward the bureaucratic [role-oriented] mode” (Harrison 1987, p. 8).

“In the organization that is task-oriented, achievement of a superordinate goal is the highest value. The goal need not be economic. [...] The important thing is that the organization’s structure, functions, and activities are all evaluated in terms of their contribution to the superordinate goal. Nothing is permitted to get in the way of accomplishing the task. If established authority impedes achievement, it is swept away” (Harrison 1972, p. 122). Appropriate knowledge and competence is needed to gain authority. Harrison also describes this culture as fostering “deep personal satisfaction” (1987, p. 9) as well as evoking “strong personal commitment” in “high energy work situations” and links them to “new business and new plant startups, nuclear test shots, intensive care units, combat teams, and political and community organizing campaigns”. He also points to “social service organizations, research teams, and high-risk businesses” (1972, p. 122). Task forces and project teams are also mentioned.

“Unlike the other three types, the person-oriented organization exists primarily to serve the needs of its members. The organization itself is a device through which the members can meet needs that they could not otherwise satisfy by themselves. [...] Authority in the role- or power-oriented sense is discouraged. When it is absolutely necessary, authority may be assigned on the basis of task competence [...]. Instead, individuals are expected to influence each other through example, helpfulness, and caring” (Harrison 1972, pp. 122–123). Harrison redefined this culture later as “an organizational climate based on mutual trust between the individual and the organization. In such an organization, people believe they are valued as human beings, not just as cogs in a machine” (1987, p. 13). He gives examples of small groups of professionals who have joined together for research and development as well as some consulting companies.

Harrison also states, “the pure support culture tends not to thrive in business unless it is balanced by a drive for success—an achievement orientation” (1987, p. 14).

Those “organizational ideologies”, as Harrison called them back in 1972, are usually not found as pure types. However, usually a company focuses primarily on a single one.

Another pair of authors, who had a major impact on the field of organizational culture, especially when viewed from the managerial angle, are Terrence E. Deal and Allan A. Kennedy. They originally published their first book ‘Corporate Culture: The Rites and Rituals of Corporate Life’ in 1982. This book centered on the newly coined term ‘corporate culture’ and sparked a “firestorm of controversy” (Deal and Kennedy 2000b, p. 1), which again brought the concept of organizational culture to the attention of a wide audience.

Deal and Kennedy state in their original work, that “each company faces a different reality in the marketplace depending on its products, competitors, customers, technologies, government influences, and so on. [...] In short, the environment in which a company operates determines what it must do to be a success” (2000a, p. 13). This is notable, because the authors state that culture is shaped by outside influences rather than by the individuals inside the company, as most other authors suggest. This outside focus reflects in their corporate culture model, as can be seen in Fig. 18.2.

The process culture is defined as “a world of little or no feedback where employees find it hard to measure what they do; instead they concentrate on how it’s done” (Deal and Kennedy 2000a, p. 208). “How neatly and completely workers do something is often more important than what they do. [...] People who are valued in this culture are those who are trying to protect the system’s integrity more than their own” (ibid., p. 120). When looking for examples, the authors point to “banks, insurance companies, financial-service organizations, large chunks of government, utilities, and heavily regulated industries like pharmaceutical companies” (ibid., p. 119).

The work hard/play hard culture is described as a “world of small risks [...] and quick, often intensive feedback. Activity in this world is everything. [...] Success comes with persistence” (ibid., p. 113). “If the tough-guy culture is built on ‘find a



Fig. 18.2 Deal and Kennedy’s culture model (Based on Deal and Kennedy 2000a, b)

mountain and climb it,' then work hard/play hard rests on 'find a need and fill it'" (ibid.). "While anyone who succeeds in a tough-guy culture becomes a star; here the team beats the world because no individual really makes a difference. The team produces the volume" (ibid., p. 114). The authors give some examples, which include primarily sales organizations such as real estate, automotive distributors, mass consumer-sales companies, office-equipment manufacturers, and all retail stores.

Deal and Kennedy define the tough-guy (also called "macho" or "stars") culture as "the most grueling of all business cultures" (ibid., p. 108). The stakes are high and the feedback is quick. "Tough-guy, macho cultures tend to be young ones with a focus on speed, not endurance. Not taking an action is as important as taking one" (ibid., p. 109). People in this culture require a "tough attitude" and internal competition is high. This is a "world of individualists" where "outlaw heroes are the norm" (ibid., p. 110). Examples of this culture include construction, cosmetics, management consulting, venture capital, advertising, and publishing. Police departments and surgeons are described as the essence of this type of culture since the stakes there are often ones of life or death.

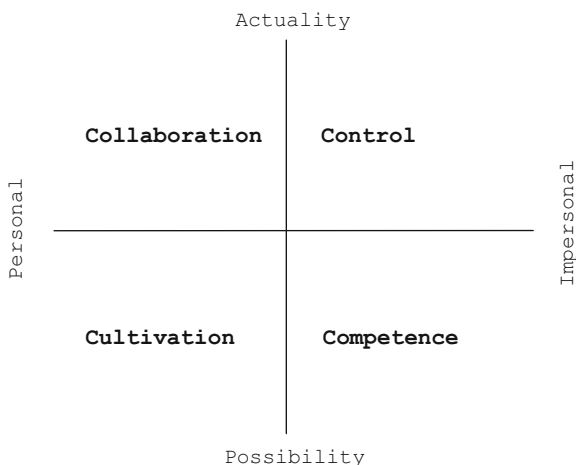
Bet-your-company cultures have to endure "high risk, but slow feedback" (Deal and Kennedy 2000a, p. 116). "Slow here doesn't mean less pressure; instead it means pressure that is as persistent as low-drip water torture" (ibid.). "Instead of putting their careers on the line—as tough guys would—corporate bettors often risk the future of the entire company" (ibid., p. 117). "Decision-making comes from the top down—once all the inputs are in. [. . .] The values of this culture focus on the future and the importance of investing in it" (ibid.).

Industries exemplifying an inhibition of a bet-your-company culture include capital goods, mining and smelting, investment banks, and computer-design companies.

While many similarities between Harrison and Deal and Kennedy can be found, there are also—sometimes subtle—differences. William Schneider tried to work out those differences (and also those of other authors) to find a generally accepted and universal corporate culture model (cf. Schneider 1999, pp. 149–153). He builds on the work of other authors, amongst whom Harrison as well as Deal and Kennedy can be found. He also defines a four-square-matrix to describe his culture model (cf. Fig. 18.3).

The author describes the cultivation culture as "one of faith", that "heralds a system of beliefs or expectations that the organization and its people will accomplish what it deems valuable. [. . .] This culture trusts unquestioningly in success, in its people and in the organization" (Schneider 1999, p. 82). The individual's commitment and the fulfillment of worthwhile purposes create the energy and vitality of the cultivation culture. Schneider gives some industry examples as well: "Organizations dedicated to aesthetics are often cultivation cultures: symphony orchestras, theaters, artistic organizations, and some entertainment, advertising, and media graphics enterprises" (1999, p. 88). On top of that, Schneider mentions religious enterprises as additional examples.

Fig. 18.3 Schneider's culture model (Based on Schneider 1999)



“The collaboration culture springs from the family” (ibid., p. 44). Its “way to success is to put a collection of people together, to build these people into a team, to engender their positive affective relationship with one another and to charge them with fully utilizing one another as resources” (ibid., p. 45). This means that destructive behavior and excessive self-interest do not go well with this culture. “Status and rank take a back seat” (ibid., p. 50). Examples include service organizations (such as health care organizations, especially hospitals), many family-owned and -operated businesses, nursing, entertainment, and many personal service enterprises.

In contrast, “control cultures prize objectivity. Emotions, subjectivity, and ‘soft’ concepts take everyone’s eye off the ball and potentially get the organization in trouble. Empiricism and the systematic examination of externally generated facts are highly valued” (ibid., p. 30). Important values in control cultures are order and predictability, as well as maintaining stability. “Decision-making is highly detached and impersonal” (ibid., p. 35). Examples mentioned by Schneider are energy companies, resource companies, defense, manufacturing companies, commodity or commodity-like enterprises, enterprises that have to do with matters of life and death as well as companies in mature markets.

In describing the competence culture, Schneider heavily refers to McClelland (1961). He argues “the competence culture is based in the achievement motive, discovered by McClelland in his research on individuals and societies and defined as man’s need ‘to compete against a standard of excellence’” (Schneider 1999, p. 63). Schneider continues to explain that, “the need to achieve has to do with accomplishing more and doing better than others” (ibid.). In a competence culture, being superior or the best is paramount. This can mean having the best product, service, process or technology in the marketplace. “This culture gains its uniqueness by combining possibility with rationalism. What might be and the logic for getting there are what count” (ibid., p. 65). Fundamental values are knowledge and

information. Formalities and emotional considerations are not important compared to proven accomplishment. “A competence culture values competition for its own sake even though it is not necessarily more competitive than other core cultures. There is a love of challenge; people like to be told that ‘it can’t be done’” (ibid., p. 68). Universities are described as being a natural competence culture prototype, which is also true for research and development organizations, many consulting firms, accounting firms, think tanks, and engineering construction firms.

Schneider provides a questionnaire (20 questions) in his book to classify any given enterprise into this culture model. However, this questionnaire was not statistically validated and therefore is of little scientific use (cf. Schneider 1999, p. 18).

Cameron and Quinn present a statistically validated and widely used tool to diagnose culture. It is called “Organizational Culture Assessment Instrument”, or “OCAI”¹ and is based on the Competing Values Framework, which is founded in the work of Quinn and Rohrbaugh (1983) (Fig. 18.4).

The culture model presented by Cameron and Quinn (2011) places organizations in a continuum of four core values, called Flexibility, Stability, Differentiation, and Integration. “What is notable about these four core values is that they represent opposite or competing assumptions. Each continuum highlights a core value that is opposite from the value on the other end of the continuum” (Cameron and Quinn 2011, p. 40). The authors have named the quadrants (cf. Fig. 18.5) in a way that resonates well with managers and researchers alike who have some knowledge in organizational culture frameworks. “It is important to note that these quadrant names were not randomly selected. Rather, they were derived from the scholarly literature that explains how, over time, different organizational values have become associated with different forms of organizations. We [Cameron and Quinn] discovered that the four quadrants that emerged from these analyses [Clan, Adhocracy, Hierarchy, Market] match precisely the main organizational forms that have developed in organizational science. They also match key management theories about organizational success, approaches to organizational quality, leadership roles, and management skills” (ibid.).

Hierarchy cultures emerge, because “the environment was relatively stable”. Due to that fact, “tasks and functions could be integrated and coordinated, uniformity in products and services was maintained, and workers and jobs were under control. Clear lines of decision-making authority, standardized rules and procedures, and control and accountability mechanisms were valued as the keys to success” (ibid., p. 42). A company with such an organizational culture is a “formalized and structured place to work. Procedures govern what people do. [...] Formal rules and policies hold the organization together”. “The long-term concerns of the organization are stability, predictability, and efficiency” (ibid.). In such an environment, “effective leaders are good coordinators and organizers. Maintaining a smoothly running organization is important”. Examples

¹ ©Kim Cameron, University of Michigan.

Fig. 18.4 The competing values framework (Based on Quinn and Rohrbaugh 1983)

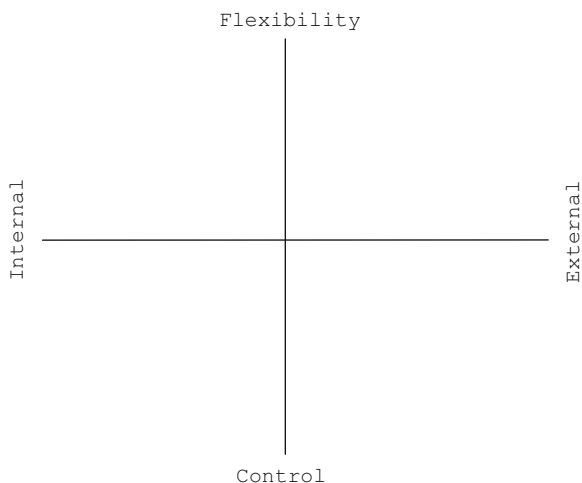
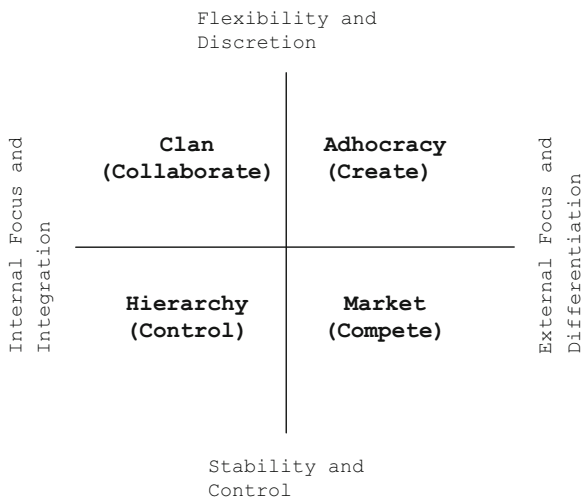


Fig. 18.5 Cameron and Quinn culture model (Based on Cameron and Quinn 2011, p. 39)



include “large organizations and government agencies [which] are generally dominated by a hierarchy culture, as evidenced by large numbers of standardized procedures, multiple hierarchical levels (Ford has 17 levels of management), and an emphasis on rule reinforcement” (ibid.). In general, “hierarchy cultures are characterized by a controlling environment” (ibid., p. 43).

In contrast to the stable environment assumption of the hierarchy culture, “the basic assumptions in a market culture are that the external environment is hostile rather than benign, consumers are choosy and interested in value, the organization is in the business of increasing its competitive position, and the major task of management is to drive the organization toward productivity, results, and profits. It

is assumed that a clear purpose and an aggressive strategy lead to productivity and profitability” (ibid., p. 45). A market culture therefore has to be a “results-oriented workplace”. “Leaders are hard-driving producers and competitors who are tough and demanding. The glue that holds the organization together is an emphasis on winning, [and] the long-term concern is on competitive actions and achieving stretch goals and targets. Success is defined in terms of market share and penetration [while] outpacing the competition and market leadership are important” (ibid., p. 46).

The tough and demanding leader of the market culture will not last long in a clan culture. A more team-oriented approach is needed: “Basic assumptions in a clan culture are that the environment can best be managed through teamwork and employee development, customers are best thought of as partners, the organization is in the business of developing a humane work environment, and the major task of management is to empower employees and facilitate their participation, commitment, and loyalty” (ibid.). Sharing the same values, beliefs, and goals is paramount, especially in rapidly changing, turbulent environments. In general, the clan culture is “typified by a friendly place to work where people share a lot of themselves. It is like an extended family. Leaders are thought of as mentors and perhaps even as parent figures” (ibid., p. 48). Those leaders hold the organization together by loyalty and tradition, which leads to a high commitment. “Success is defined in terms of internal climate and concern for people. The organization places a premium on teamwork, participation, and consensus” (ibid.).

The fourth organizational form described by Cameron and Quinn is called adhocracy. “The root of the word adhocracy is *ad hoc*—implying something temporary, specialized, and dynamic” (ibid., p. 49). Adhocracies can be found in environments that are even more turbulent than those in which clan cultures thrive. “A major goal of an adhocracy is to foster adaptability, flexibility, and creativity if uncertainty, ambiguity, and information overload are typical” (ibid.). The authors found a number of characteristics that are common in this type of organization: No organizational charts due to the frequently and rapidly changing structure, temporary physical space, temporary roles and responsibilities depending on changing client problems, as well as creativity and innovation were the most visible ones. “In sum, the adhocracy culture [. . .] is characterized by a dynamic, entrepreneurial, and creative workplace. People stick their necks out and take risks. Effective leadership is visionary, innovative, and risk oriented. The glue that holds that organization together is commitment to experimentation and innovation” (ibid., p. 51). Quite often, “the emphasis is on being at the leading edge of new knowledge, products, and services. Readiness for change and meeting new challenges are important. The organization’s long-term emphasis is on rapid growth and acquiring new resources. Success means producing unique and original products and services” (ibid.).

Cameron and Quinn also found that “new or small organizations tend to progress through a predictable pattern of organization culture changes” (ibid., p. 64), starting in the adhocracy quadrant, evolving into a clan, and then a hierarchy culture until it finally settles into a market form, as shown in Fig. 18.6:

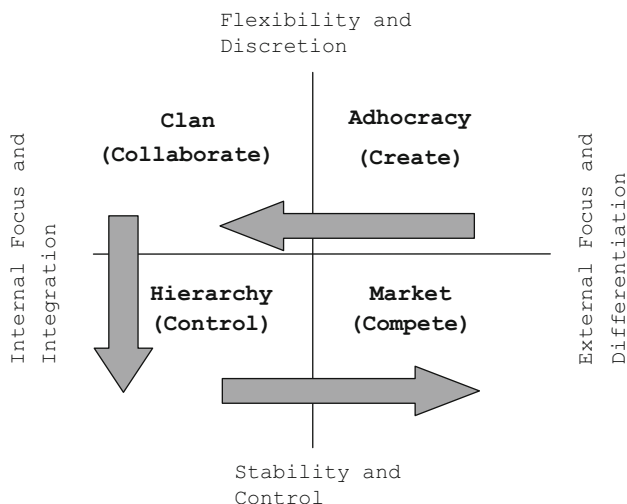


Fig. 18.6 Culture change over time (Based on Cameron and Quinn 2011, pp. 64–65)

This cultural evolution is more or less inevitable. However, if properly managed, elements of other quadrants can be used to soften the weaknesses of the market or hierarchy culture. The necessary starting point for such action is to know what the company believes to be important today. That is, while the company as such may represent one of these quadrants, it might “indeed have a strong secondary component. This is also the case at the department/group level” (Tharp 2009, p. 5). It is rare however to “have companies that share equal traits of all four culture types—with no dominant or barely dominant type” (ibid.). Therefore, the use of this model might lead to a more sophisticated (and complex) result than the pure positioning in a single quadrant of a four square matrix.

The four models shown above describe different aspects of culture. Harrison focuses on how processes are conducted and decisions are made within a culture, that is, if they are centralized and formalized, or not. Deal and Kennedy focus on what kinds of decisions have to be made—are the stakes high and how quickly does the decision-maker know if the decision was right? Schneider focuses more on the general way of thinking in the decision making process. Does the decision-maker primarily think about people or the company? Is he focusing on the present or the future? Cameron and Quinn introduce the element of cultural evolution and focus on the values held dear by the organization: Flexibility, stability, differentiation, or integration.

It is hard to choose between these models in order to evaluate the cultural nature of Scrum. In particular Schneider, whose work was already used by another researcher (cf. Spayd 2010) to analyze Scrum, looks promising. However, Schneider’s work is not validated and the author no longer uses his own questionnaire to analyze corporate cultures (as far as I know). Due to the fact that validated

Table 18.2 Decision matrix for model selection

	Harrison	Deal and Kennedy	Schneider	Cameron and Quinn
Name of quadrants	Achievement	Bet-Your-Company	Cultivation	Adhocracy
	Person	Work Hard/Play Hard	Collaboration	Clan
	Power	Process	Control	Hierarchy
	Role	Tough-Guy	Competence	Market
Primary focus	Process conduction and decision making	Kinds of decisions	General way of thinking in the decision making process	Values held dear by organization
X-axis	High/low centralization	High/low risk	People/company orientation	Internal/external focus
Y-axis	High/low formalization	Fast/slow feedback	Actuality/possibility orientation	Flexibility vs. Stability
Includes questionnaire by author	Yes (Harrison and Stokes 1992)	No	Yes	Yes
Questionnaire is statistically validated	No	n.a.	No	Yes
Central database exists for further research	Yes	n.a.	No	Yes
Model is still in practical use today	No	Yes	Yes	Yes
Model has been used to analyze Scrum	No	No	Yes	No
Is the author still basing his work on the model?	Unknown	Yes	No	Yes

quality, actuality, the availability of a central database and the generally sophisticated approach, the Cameron and Quinn model is chosen for this research project (Table 18.2).

18.3 A Broader View on Cultural Dimensions

Even though Cameron and Quinn created a sophisticated model that will lead to valuable insights, it might prove to not be sufficient. In order to understand the full complexity of the inherent cultural characteristics of Scrum, more than just a typology might be needed. “The value of typologies is that they simplify thinking and provide useful categories for sorting out the complexities we must deal with when we confront organizational realities. [...] The weakness of culture typologies is that they oversimplify these complexities and may provide us categories that are incorrect in terms of their relevance to what we are trying to understand. They limit our perspective by prematurely focusing us on just a few dimensions, they limit our ability to find complex patterns among a number of dimensions, and they do not reveal what a given group feels intensely about” (Schein 2010, p. 175). So with the Organizational Culture Assessment Instrument (OCAI), the dimensions of analysis might have been narrowed.

Broadening the horizon of this research could happen through a survey, expert interviews, or group workshops. In Schein’s opinion, “culture cannot be assessed by means of surveys or questionnaires because one does not know what to ask, cannot judge the reliability and validity of the responses, and may not want to influence the organization in unknown ways through the survey itself” (Schein 2009, p. 101). These concerns are well founded, but do not fit to the situation at hand. Since not one individual organization is assessed, there is no risk to influence an organization by means of a questionnaire. A survey is never reliable, but neither are interviews or workshops. However, by gathering a large enough data sample (e.g. 200), the significance can be statistically verified. To find out which questions to ask, one can consult literature: Even Schein himself (2009, 2010) gives ample examples of what to ask.

One additional issue eases the decision even more: that of pragmatism. The OCAI questions will be asked in a survey. Since the people answering that survey will be scattered all around the world and will be answering a questionnaire anyway, it is easy to add some more questions. Therefore, a survey approach is chosen. The only open issue is what questions to ask.

Schein divides what culture is about into three areas, of which the first one is obvious (and well documented in the case of Scrum):

“External Survival Issues

- Mission, strategy, goals
- Means: structure, systems, processes
- Measurement: error-detection and correction systems

Internal Integration Issues

- Common language and concepts
- Group boundaries and identity
- The nature of authority and relationships
- Allocation of rewards and status

Deeper Underlying Assumptions

- Human relationships to nature
- The nature of reality and truth
- The nature of human nature
- The nature of human relationships
- The nature of time and space
- The unknowable and uncontrollable” (Schein 2009, pp. 39–40)

These aspects are a good starting point, which is backed by other authors as well. Martin states for example, that one of the first manifestations of culture an outsider encounters when entering it, is language that only cultural insiders are able to decipher (2001, p. 77), which matches Schein’s hint to “common language and concepts” (2009, p. 40). Martin also differentiates between technical and emotional jargon. “Technical jargon is task oriented and appears to be emotionally neutral. In contrast, emotionally laden jargon is more overtly concerned with feelings. For example, ‘idea hamsters’ on the ‘bleeding edge’ are metaphors of life and death in Silicon Valley, the U.S. Mecca for high-technology entrepreneurship” (2001, p. 77). Asking for technical jargon is easy: It manifests in all the acronyms and special statements only insiders understand. Emotional jargon is usually hidden, but it does surface in the form of humor (ibid., p. 81).

Group boundaries are sometimes blurred. However, in some cultures membership badges, uniforms, special symbols or privileges are used (cf. Schein 2009, p. 55). In addition, people almost always have a fair understanding of who is an “insider” and who is an “outsider” of their culture. To find out what a group believes to be true about the nature of authority and relationships, more subtle questions should be asked to prevent people from answering in a socially acceptable way. Aside from inquiring how people are addressed, it should be investigated how discussions commence, and whose opinion is valued most in group meetings. The way in which disagreement with one’s boss is voiced—if at all—is an important indicator as well (cf. ibid., p. 57). This should tell a lot about the underlying beliefs when considering the nature of authority.

In any given group, rewards and status have to be distributed, or as Schein puts it: “Every group must work out its pecking order, its criteria and rules for how someone gets, maintains, and loses power and authority. Consensus in this area is crucial to help members manage feelings of aggression” (2009, p. 94). In most companies, the primary way to get power and improve one’s status is by way of a promotion. On a smaller scale, rewards and punishments are relevant. To find out more about this issue, it should be asked what kind of behavior is rewarded or punished and how one knows (cf. ibid., pp. 58).

The aspects of ‘human relationship to nature’ and ‘the nature of reality and truth’ are deeply rooted in the national cultures in which a company operates (cf. Schein 2009, p. 61). Since this study is not trying to analyze national cultures but rather cultural characteristics of Scrum, those aspects are not investigated. Human nature, however, definitely is relevant. The major question is whether people want to work, or do not want to work. McGregor, Herzberg, and others found that financial incentives might decrease motivation, but not increase it above a certain point.

Instead, personal challenges and the opportunity to use one's talents are needed (cf. McGregor 2000; Herzberg 2003). Commonly referred to as "Theory X" and "Theory Y", the first one assumes that people only work when "carrots and sticks" are used, while the latter believes that people are intrinsically motivated. In addition, it is generally assumed that people matching "Theory Y" are highly motivated and like coming to work, while those fitting "Theory X" do not. It is easy to ask people how they feel at work.

Assumptions about human relationships as such are difficult to inquire. While people might espouse group values, they might actually follow a more individualistic approach. While direct questions are risky, it is still helpful to ask for the espoused values. Questions regarding the leadership style and the focus of the company as such might reveal some useful information in that regard more indirectly.

Other cultural aspects, such as assumptions about space, can be identified more easily. "Architecture, interior décor, and dress norms are particularly powerful cultural clues, in part because they are so easy to see" (Martin 2001, p. 83). While it is not expected to find a single common architecture that permeates all Scrum organizations on the planet, there might be clues about office design. This information could be supplemented by asking for the perceived noise level: Is there a buzz of communication in an open-plan office or silent working behind closed doors? When looking at space, assumptions about time should not be forgotten. Is it perceived as controllable? This is especially important, since "planning time as used by most managers assumes that one can speed things up or slow them down according to the needs of the moment. If something needs to be done soon, we 'work around the clock' to meet the deadline. On the other hand, the R&D department is more likely to be working on 'development time,' [...] implying that the development of certain processes cannot be speeded up" (Schein 2009, pp. 70–71). The importance lies here in the fundamentally divergent concept of time, which could lead—if different amongst members—to conflicts in enterprises. Asking for overtime encouragement and monitoring intervals should reveal the underlying thought concept.

To finish this line of thought, it had to be investigated how Scrum deals with the unknowable and uncontrollable. This was straightforward. In addition, people were allowed to report on any visible artifacts or general ideas that might not have been covered by the other questions. Table 18.3 shows a summary of all identified questions.

With these questions supplementing those of the OCAI, a broad view on the cultural implications of Scrum could be gathered. While the questionnaire was longer than originally expected, this extension was necessary since "culture is a multidimensional, multifaceted phenomenon, not easily reduced to a few major dimensions" (Schein 2010, p. 91). That in mind, we can dig into the existing Scrum literature and mine some cultural gems.

Table 18.3 Questions to deepen cultural insights

Schein's category	What to find out	What to ask
Common language and concepts	Technical jargon	Specify all jargon and acronyms that might be common in a perfect Scrum company
Common language and concepts	Emotional jargon	What jokes are common in a perfect Scrum company?
Group boundaries and identity	Dress norms	What dress code is dominating in a perfect Scrum company?
Group boundaries and identity	Badges, Uniforms, symbols or privileges	How are different degrees of status symbolized? Are there any sort of uniforms, badges, and so on?
Group boundaries and identity	Insider and outsider	Who is considered an "insider" or "outsider" in a perfect Scrum company?
Nature of authority and relationships	Formal or informal relationship between people	Are people in a perfect Scrum company addressing each other on a first name basis or differently?
Nature of authority and relationships	Formal or informal relationship with bosses	Are people in a perfect Scrum company addressing their bosses on a first name basis or differently?
Nature of authority and relationships	Pecking order in meetings	How would you describe behavior in group meetings?
Nature of authority and relationships	Source of authority	Whose opinion is valued most in group meetings?
Nature of authority and relationships	Openly voiced criticism	If you disagree with the boss, do you feel encouraged or discouraged to voice your disagreement face-to-face?
Nature of authority and relationships	Openly voiced criticism	Is it OK to disagree in front of others, or do you have to seek the boss out and disagree privately?
Allocation of rewards and status	How to gain power	How does promotion ("climbing up the ladder") look like in a perfect Scrum company?
Allocation of rewards and status	What is rewarded	What kind of behavior is rewarded in a perfect Scrum company?
Allocation of rewards and status	What is punished	What kind of behavior is punished in a perfect Scrum company?

(continued)

Table 18.3 (continued)

Schein's category	What to find out	What to ask
Allocation of rewards and status	Reward mechanisms	How do you know when you have been rewarded or punished in a perfect Scrum company?
The nature of human nature	Are people intrinsically or extrinsically motivated	In a perfect Scrum company: Does management believe that people want to work (intrinsic) or do they believe people need external (extrinsic) motivators to work (e.g. money)?
The nature of human nature	Like people coming to work	How does work feel in a perfect Scrum company?
The nature of human relationships	Espoused values	What values are espoused in a perfect Scrum company?
The nature of human relationships	Focus	What is a perfect Scrum company focusing on?
The nature of human relationships	Leadership style	How would you describe the leadership style in a perfect Scrum company?
Nature of Space	Office design	What does the working space look like in a perfect Scrum company?
Nature of Space	Communication amount in the environment	How would you describe the noise level in a perfect Scrum company?
Nature of Time	Overtime encouragement	Is working overtime encouraged or despised in a perfect Scrum company?
Nature of Time	Monitoring intervals	How long is an employee left alone without being monitored in a perfect Scrum company?
Unknowable and Uncontrollable	How is it dealt with	How does a perfect Scrum company deal with the Unknowable and Uncontrollable?
General	Missed artifacts	What artifacts ("important tangibles") are visible in a perfect Scrum company?
General	Missed ideas	What else do you want to point out in regard to the nature of Scrum?

References

- Cameron, K. S., & Quinn, R. E. (2011). *Diagnosing and changing organizational culture* (3rd ed.). Hoboken, NJ: Wiley.
- Deal, T. E., & Kennedy, A. A. (2000a). *Corporate cultures. The rites and rituals of corporate life* (Revised ed.). New York: Basic Books.
- Deal, T. E., & Kennedy, A. A. (2000b). *The new corporate cultures. Revitalizing the workplace after downsizing, mergers and reengineering* (First Paperback ed.). New York: Basic Books.
- Harrison, R. (1972). Understanding your organization's character. *Harvard Business Review*, 50(3), 119–128.

- Harrison, R. (1987). Harnessing personal energy: How companies can inspire employees. *Organizational Dynamics*, 16(2), 5–20.
- Harrison, R., & Stokes, H. (1992). *Diagnosing organizational culture*. San Francisco: Jossey-Bass/Pfeiffer.
- Hertzberg, F. (2003, January). One more time: How do you motivate employees? *Harvard Business Review*, 81(1), 87–96.
- Martin, J. (2001). *Organizational culture: Mapping the Terrain*. Thousand Oaks: Sage.
- McClelland, D. C. (1961). *The achieving society*. Princeton, NJ: D. Van Nostrand.
- McGregor, D. (2000, Fall). The human side of enterprise. *Reflections*, 2(1), 6–15.
- Schein, E. H. (2009). *The corporate culture survival guide* (New and Revised ed.). San Francisco: Jossey-Bass.
- Schein, E. H. (2010). *Organizational culture and leadership* (4th ed.). San Francisco: Jossey-Bass.
- Schneider, W. E. (1999). *The reengineering alternative. A plan for making your current culture work* (Special reprint ed.). New York: McGraw Hill.
- Spayd, M. (2010). *Agile & culture: The results*. 06.07.2010. Collective edge coaching. 01.01.2013. http://collectiveedgecoaching.com/2010/07/agile__culture/
- Tharp, B. M. (2009). *Four organizational culture types*. Haworth. Organizational Culture White Paper. 04.09.
- Quinn, R. E., & Rohrbaugh, J. (1983, March). A spatial model of effectiveness criteria: Towards a competing values approach to organizational analysis. *Management Science*, 29(3), 363–377.



19.1 The Origins of Scrum

In order to understand a culture, one has to know where its origins are and how it evolved. Ken Schwaber and Jeff Sutherland presented Scrum for the first time at the OOPSLA conference in Austin, Texas, in 1995. At this point they had already been working together for several years. Both had huge experience as software developers, project managers, line managers, and owners of IT companies in the United States of America. They were struggling with the status quo of dealing with software development, specifically waterfall project management approaches. At that time, projects and companies they were involved in were failing and due to the pressure they were feeling, they went in a different direction. At the same time, lean management and empirical process control as well as iterative and incremental development practices were emerging and impressed Ken and Jeff. Especially the works of Babatunde Ogunnaike¹ as well as Takeuchi and Nonaka (1986) influenced them (cf. Schwaber and Sutherland 2012, p. 27).

Takeuchi and Nonaka described the core elements of success in new product development (self-organizing project teams, overlapping development phases, multi-learning, and transfer of learning) and referred to the game of Rugby when they summarized it as “moving the Scrum downfield”. Ken and Jeff accordingly labeled their approach “Scrum”. They continued working together refining their ideas. In 2001, the first book about Scrum was published. Together with Mike Beedle, Ken wrote “Agile Software Development with Scrum”—followed by “Agile Project Management With Scrum” in 2004—and laid out the ideas in a more detailed way than before. One year later (2002), Ken founded the Scrum Alliance, aiming at providing worldwide Scrum training and certification. Scrum continued to evolve and so did Ken’s wish for the direction of the Scrum Alliance. He moved on and founded Scrum.org in 2009. Shortly thereafter, in 2011, Ken and

¹ Especially Ogunnaike and Ray (1992).

Jeff, as co-creators of Scrum, wrote and published the Scrum Guide. This 16-page document has since then been the official definition of Scrum. It is jointly maintained and updated by Ken and Jeff.

It is also noteworthy, that Scrum exists in the context of so-called “Agile” methods. In February 2001, a group of thought leaders from the IT industry got together in a ski resort in the Wasatch mountains of Utah and created the “Agile Manifesto” (Beck et al. 2001), a set of 4 values and 12 principles describing how they envision the development of software. Ken and Jeff were amongst the participants, who afterwards named themselves and founded the “Agile Alliance”. This Agile Manifesto is the first hint to a cultural definition of Scrum since it represents the expression of the shared beliefs of a group of people, connected in the domain of agile software development. Even though this is not a direct definition of Scrum, it nevertheless represents some of the values shared by the inventors of Scrum and thus is loosely associated evidence. The Manifesto for Agile Software Development (“Agile Manifesto”) reads:

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more. We follow these principles:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity—the art of maximizing the amount of work not done—is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.

- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.” (Beck et al. 2001)

The stated values and principles give some indication about focus, collaboration, transparency, technical excellence, authority, continuous improvement, communication, and sustainable pace. However, they are not imperatively a part of a “Scrum culture” since they were defined at the broader “Agile” level. It has to be researched what respectable authors say about those characteristics in the context of Scrum.

19.2 General Cultural Characteristics of Scrum

In search of “respectable” Scrum authors one stumbles across Mike Cohn, Ken Schwaber, and others. Mike and Ken are the single most influential people in Agile and Scrum (cf. Smith 2012). Therefore, Schwaber (2004, 2007) and Cohn (2009) have been chosen for analysis. Even though Jeff Sutherland is just the ninth influential person in the domain of Agile, he should also be part of the analysis since he is the co-creator of Scrum. Jeff had only written one book (together with Ken Schwaber) when this study was conducted, so Schwaber and Sutherland (2012) was added to the list. While some of what is written in the original book about Scrum is already outdated, it still does not hurt to take a look at Schwaber and Beedle (2002). On top of that, The Scrum Guide must be part of the study, since it is the “definitive guide to Scrum” and states “the rules of the game” (Schwaber and Sutherland 2016, p. 1).

Scrum is defined as “a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value” (ibid., p. 3). The nature of a framework is its incompleteness. Similar to a picture frame hung up on a wall, a framework only makes sense when filled with useful content. Of course, it always exists in the context of something larger—a wall in the case of a picture, an organization in the case of Scrum. This means that Scrum does not give all the answers in a detailed prescriptive fashion. The reason for that is that solutions for product development vary widely between teams and companies due to differing demand (cf. Schwaber 2007, p. xi).

What Scrum does prescribe are three roles (Scrum Master, Product Owner, Development Team), three artifacts (Product Increment, Product Backlog, Sprint Backlog), and five events (Sprint, Sprint Planning, Daily Scrum, Sprint Review, and Sprint Retrospective). These are bound together by a couple of rules and principles. Briefly explained, a Sprint is an iteration of 1 month or less that starts with a Sprint Planning meeting and ends with a Sprint Retrospective meeting. Within this Sprint, the Development Team creates a fully functional Product Increment out of the requirements of the Product Owner (which are stored in the Product Backlog). In order to stay on track, the Development Team reevaluates its progress every 24 h in their Daily Scrum and re-plans their work in the Sprint Backlog. The Scrum Master is there to coach the other roles and to remove issues

that impede the progress of the team (they are therefore called “impediments”). The whole approach is highly collaborative, focused on frequent delivery, continuous improvement, and on clear responsibilities.²

For a deeper insight into the cultural aspects of Scrum, the selected literature is studied for answers to the questions mentioned in Sect. 18.3.

Since vocabulary is an important aspect of culture, it has to be considered how this appears to a Scrum outsider. Whenever someone first comes across Scrum, they are immersed in a whole broad new vocabulary. Even the name “Scrum” is new and meaningless to the outsider, while the insider connects it to the view “that a product development team should behave much like a rugby team—a group of individuals moving the ball down a field as a unit” (Cohn 2009, p. 201). The official standard vocabulary around that view describes the different elements of Scrum (Sprint, Sprint Planning, Sprint Review, Sprint Retrospective, Daily Scrum, Product Owner, Scrum Master, Development Team, Scrum Team, Product Backlog, Sprint Backlog, and Product Increment) and is explained in every single Scrum literature on the market.

Other jargon is borrowed from empirical process control and includes complexity, transparency, inspection, adaptation, embracing change, good enough, the art of the possible, failing early/fast, and iterations. On that, the “skeleton” of Scrum was formed: iterative incremental practices (cf. Schwaber 2004, p. 5), also described as “sashimi” (ibid., p. 55). In addition, the team is emphasized as being self-managing, self-organizing, cross-functional, and responsible for its work, which is described as shipping an “increment of potentially shippable product functionality”, or “Increment.” When talking about teams, the authors also emphasize a “sustainable pace of work”, or “sustainable pace”, and “purge” people “to the bench” instead of firing them off a team (cf. Schwaber 2007, p. 79). The person accountable for the product, the Product Owner, is nicknamed “single, wringable neck” (Schwaber 2007, p. 6), highlighting his extraordinary importance and responsibility.

On the product side, the focus is on quality as expressed by using a tool called “Definition of Done” (roughly resembling a quality management plan), and often talking about “being done” or producing “done product”. The progress towards the “Sprint goal” is tracked via so-called “burndown charts”, also described as “the collision of reality [. . .] with what is planned, or hoped for” (ibid., p. 109). To reduce complexity and keep everybody focused, all meetings (and agenda elements within meetings) are “timeboxed”, meaning that they cannot take more than a maximum amount of time.

Next to efficiency, there is also a lot of wording in the context of business. Productivity, Return on investment (ROI), total cost of ownership (TCO), metrics, business value (BV), and even not delivering anything more “when the opportunity value is greater than the marginal value of the next increment” (ibid., p. 86), all summed up under the umbrella of “value driven development” (VDD).

²For a deeper explanation of Scrum, refer to other authors, e.g. Schwaber (2004, 2007), Schwaber and Sutherland (2012), Cohn (2009).

Even though the Scrum Guide does not even mention the word “software” once, the legacy of software development cannot be denied. The literature is full of examples, case studies, and vocabulary from this domain. Some of those are especially noteworthy: technical excellence, keeping the code well factored, simple design, automated testing, and early detection of errors (cf. Cohn 2009, p. 169). Problems faced outside the code are called “impediments”, describing the fact that they are impeding the progress of the development team towards the Sprint goal.

The terminology describing Scrum adversaries revolves largely around predictive process modeling also called “defined” processes. From that wording, the arch-nemesis of Scrum emerges: “the tyranny of waterfall” (Schwaber 2007, p. 21) with a “command-and-control” management style, which contrasts the desired “servant leadership” style.

Scrum also introduces the pet names “pigs” and “chickens” for the Scrum Team and their stakeholders. This terminology represents emotional jargon and is based on the joke of pigs and chickens: “A chicken and a pig are together when the chicken says, ‘Let’s start a restaurant!’ The pig thinks it over and says, ‘What would we call this restaurant?’ The chicken says, ‘Ham n’ Eggs!’ The pig says, “No, thanks. I’d be committed, but you’d only be involved!” (Schwaber and Beedle 2002).

The distinction between those groups is made to ensure that “those who are responsible for the project have the authority to do what is necessary for its success and that those who aren’t responsible can’t interfere unnecessarily” (Schwaber 2004, p. 7). This discrimination often led to unnecessary fights between management (chickens) and Scrum Teams (pigs), because it was taken too literally. As a consequence, the metaphor was removed from the Scrum Guide (cf. Porter 2011) and is no longer officially part of Scrum.

Another example of emotional jargon is the description of the Product Owner as “single wringable neck” (Schwaber 2007, p. 6). This pays tribute to the fact that the Product Owner serves as a deflector of outside influence towards the Development Team. All stakeholder contact goes through the Product Owner and thus the anger of dissatisfied individuals is directed at this single person.

No other jokes were found in the analyzed literature. However, there are many stories and case studies, sometimes with humorous anecdotes, although these represent individual experience and not common “Scrum lore”.

The search for a Scrum “dress code” did not return any results. The topic is not even mentioned in the Scrum literature. This is different for different degrees of status symbols. The Scrum Guide clearly states, “Scrum recognizes no titles for Development Team members other than Developer, regardless of the work being performed by the person; there are no exceptions to this rule” (Schwaber and Sutherland 2016, p. 6). No uniforms, badges or other status symbols are mentioned in an appreciative way, they are at best neutral. When reading through literature, one gets the perception that being on a good team rather than sitting on the “bench” (cf. Schwaber 2007, p. 79) is the only representation of status available in Scrum. Since not even work descriptions (e.g. programmer, tester, architect, etc.) are allowed, it becomes clear that status symbols are despised.

It is interesting to research the topic of who is considered an “insider” or “outsider” in Scrum. In the early Scrum literature, the borders are clear: Product Owner, Scrum Master, and Development Team are the insiders while everybody else, especially management, are outsiders. This stance is clearly articulated by expressions like “All management responsibilities in a project are divided among these three roles” (Schwaber 2004, p. 6), “chickens have no direct authority over the project’s execution or progress” (ibid., p. 15), “the Scrum Master is a leader, not a manager” (ibid., p. 30), or “the chickens must be kept in line” (ibid., p. 36).

This distinction becomes a bit more sophisticated in later publications. Those emphasize that Scrum is about acting as a team and self-organizing rather than being a lone hero or being ordered around by management (cf. Schwaber 2007, pp. 6). Instead of attacking management as a whole, traditional project managers are teased from time to time. For examples it is expressed that “project managers might also believe that lying saves time” (ibid., p. 27). While this does not mean that all project managers are liars, it puts the spotlight on the dysfunctions often visible in complex projects that are approached with waterfall procedures in traditional hierarchy-oriented enterprises. It becomes clear that “outsiders” are no longer everybody around the Scrum Team but rather those that “seek easy answers and simple solutions to complex problems” (ibid., p. 101). The edges around the Scrum Team as insiders are unclear and blurred, however.

Mike Cohn follows this logic and gets a bit more precise. He does not explicitly name people as insiders or outsiders, but draws a line between being agile and not being agile. He puts a strong focus on teams and team players, criticizing that traditional project managers are left “to wonder what their role is” (Cohn 2009, p. 420). One of the major criteria to decide if somebody is agile—and thereby an insider—is, if they are continuously improving or not (ibid., p. xxvii). This improvement is seen totally independent from the level of expertise, quality, etc. already achieved.

In the latest literature, the frontier is softened even more. The removal of the pigs and chicken metaphor in 2011 (cf. Porter 2011) marked the beginning of a much broader inclusion of people in an organization. The book “Software in 30 Days” by Schwaber and Sutherland directly addresses management functions in enterprises, while all literature prior to that always addressed the Development Team or the Scrum Master roles. Neither line nor project management is attacked or explicitly excluded. However, old “waterfall” thinking is still despised and comes closest to the definition of an outsider. Schwaber and Sutherland clearly state, “The world is uncertain. Software development is uncertain”, and offer to “help people relax their desire for certainty” (2012, p. 53).

Summing up, insiders seem to be people who think in an agile way, especially the members of the Scrum Team while outsiders are people who think in traditional “waterfall” dimensions and strive for absolute certainty.

Determining if people are addressing each other and their bosses on a first name basis or differently is not straightforward. This is not explicitly mentioned anywhere. However, the case studies written by the investigated authors address the protagonists on a first name basis. This is valid for both peers and bosses, even

though it is hard to figure out who is a “boss” and who is not. While this is not clear proof, it is an indication for the consistent use of first names.

Behavior in group meetings can only be analyzed indirectly. After all, books do not show group dynamics and pecking orders of meetings. Whenever literature describes group meetings, a collaborative team approach is described. “There is no ‘my work’ and ‘your work’ on a Scrum team; there is only ‘our work’” (Cohn 2009, p. 201). Decisions are collectively determined and clarified through creative team discussions (Schwaber 2004, p. 6; 35). There is no hint that any individual opinion outweighs that of any other individual. The only exceptions are the clear responsibilities of the Product Owner for the product requirements and the Scrum Master for the Scrum framework. However, the Scrum Master rarely orders anybody around. Instead, he acts as a facilitator to nurture team decisions. This facilitation also includes keeping the team on track, e.g. by the use of timeboxing and “the art of the possible”, which means reaching good enough decisions fast rather than trying to get the perfect solution (cf. Schwaber 2004, p. 37).

One aspect of the intense collaboration in meetings is that of openness and honesty—even if this disappoints somebody.³ Since the Product Owner can be viewed as the “boss” in product requirement questions and the Scrum Master can be viewed as the “boss” in Scrum process questions, it is encouraged to voice disagreement with the boss face to face and on the spot. Literature does not say anything about voicing disagreement with traditional line management, though. This question cannot be thoroughly answered by literature.

How to become a “boss” or—more generally speaking—what a promotion looks like—is another question of interest. In short, there is no promotion and no ladder to climb; at least not in the traditional (originating from military tradition) sense of rising through the ranks. Of course, a development team member can fill the role of Scrum Master. However, “the authority of the Scrum Master is largely indirect; it springs mainly from the Scrum Master’s knowledge of Scrum rules and practices, and his or her work to ensure that they are followed” (Schwaber 2004, p. 25). So, strictly speaking, the Scrum Master is not really a “boss” and filling this role is therefore not really a promotion. People will only listen to her, if she earns their respect. The same is true for the Product Owner role. It is a different role, but not necessarily a step up the ladder. When looking at compensation schemes, the team focus becomes clear again: While the Scrum Guide does not say anything about it, Schwaber recommends to allocate all incentive and bonus funds to whole teams, based on the teams’ performance (cf. Schwaber 2007, p. 6).

In total, people get rewarded by greater salary and by greater responsibility (ibid., p. 81). This means that from project to project more is expected from them in their respective areas of expertise, more important projects might be assigned or a person might be assigned to a team to solve an important problem (cf. Cohn 2009,

³ This is well documented, for example in Schwaber (2007, p. 85) and Schwaber and Sutherland (2016, p. 11).

p. 411). They will not earn different titles or gain higher hierarchical positions however.

Knowing what a career might look like, the question arises what kind of behavior is rewarded or reprimanded in Scrum and how people get their actions mirrored. Firstly, “Scrum Teams succeed together and Scrum Teams fail together” (Cohn 2009, p. 201). This puts the focus strongly on the team. Team members are expected to actively participate, engage in discussions, and to be helpful. Secondly, team members should not degrade performance or productivity of the team (Schwaber 2007, p. 79), damage transparency, or violate the Scrum rules in any other way. Both positive and negative behavior is immediately communicated and acted upon by the other team members. In addition, the Scrum Master might point out rule violations. If bad behavior continues, people can be excluded from meetings or even removed from the team (cf. Schwaber 2004, p. 166). While there are most probably other reasons and ways to reward and punish, the Scrum literature is not explicit about those.

Explicitly mentioned is the type of motivation common in Scrum. All three well respected authors point to the fact that “people always do the best they can” (Schwaber and Sutherland 2012, p. 29; cf. Cohn 2009, p. 216). Even though money is mentioned from time to time, it is never used as an individual motivator. The impression prevails that work itself is the primary motivator, not money. This equals McGregor’s “Theory Y”, the belief in intrinsic motivation of people.

Intrinsically motivated people tend to like coming to work. In addition, this is an explicit goal of Scrum. The lives of the development team shall be improved and people should look forward to coming to work (cf. Schwaber 2004, p. 36; 115). This high job satisfaction is primarily attributed to the sustainable pace of work, having more control over one’s work, the high level of teamwork, producing products that delight customers, and seeing the results of one’s work quickly (cf. Cohn 2009, p. 13).

One would expect to find teamwork as one of the explicitly espoused values in the Scrum literature. Unfortunately—even though mentioned in almost every chapter—this is not the case. The officially espoused values are commitment, focus, openness, respect, and courage (Schwaber and Beedle 2002, pp. 147). In addition, the authors emphasize some values that are not explicitly named “Scrum values”, one of which is honesty (cf. Schwaber 2007, p. 85). Visibility or transparency (which relates to openness), emergence, inspection and adaptation are values derived from empirical process control. This also includes the mantra to “embrace change” (Schwaber 2007, p. 4) and to look for “the art of the possible” (Schwaber 2004, p. 37; 2007, p. 19). Empirical process control, especially in conjunction with teamwork, is closely connected to continuous improvement and learning (cf. e.g. Cohn 2009, p. xxvii). Self-organization, empowerment, and collaboration go directly into the direction of valuing teamwork, intensified by face-to-face communication, and the “involvement of customers, Product Owner, management, etc.” (Schwaber 2004, p. 57). There is also a strong urge to act instead of over-planning the task, to focus on delivering a working product instead of documentation. In fact, Ken Schwaber reflects all of the values of the Agile Manifesto (2004, p. 26). While

interesting, it is very unfortunate that the Scrum Guide, which is the “Definitive Guide to Scrum”, does not state any Scrum values. To find those, one has to read the well-established Scrum literature.

As can be derived by the documented values, Scrum is focusing on continuous improvement, teamwork, and the product as such. Looking at the definition of Scrum⁴ itself, the additional focus on people and value becomes obvious, while the product focus is reinforced. The people focus is very subtly implemented in literature, barely visible in the choice of words. For example, the common term “resource” is never used to describe people.

The definition of Scrum, as stated in the Scrum Guide, revolves around roles and meetings designed to foster collaboration around the product increment. All meetings are connected to planning; therefore, a strong planning focus is obvious. However, this planning is not predictive but rather a lean or ad-hoc planning style. One interesting tidbit is the subtitle of “Software in 30 Days” by Ken Schwaber and Jeff Sutherland. It reads: “How Agile Managers Beat the Odds, Delight Their Customers and Leave Competitors in the Dust.” Although the language is a bit flowery, a strong customer and market focus becomes obvious. All of the mentioned focus areas are resembled in literature as well.⁵

To support this focus, especially in the people and team dimensions, a servant-leadership style is employed. Leaders—who can be technical experts, the Scrum Master, the Product Owner, and line management—have to recognize themselves as serving their internal customers instead of having them fulfill their own bidding. The Scrum Guide even explicitly mentions three types of services the Scrum Master delivers to Product Owner, Development Team, and the organization as such. The Scrum Team is described as being self-organizing and should be left alone as much as possible while developing the product (cf. e.g. Schwaber 2004, pp. 6–7). Management has to create the room and space for the team to do so and then must step back to the servant-leadership position, meaning they have to remove barriers hindering the work of the team, teach, facilitate, mentor, and coach them whenever necessary, and provide the necessary transparency for the team to make the right decisions. The role of management is even described as that of a parent: “to grow their people so that they are mature and self-managing” (Schwaber 2007, p. 7). Since maturity cannot be ordered, Scrum relies on study and experience instead. Due to the complex worldview of Scrum, the belief prevails that there is no one best answer but rather individual solutions for every team and every enterprise. Consequently, it employs decentralized decision-making as close as possible to the level where the issue impedes work (cf. Schwaber 2007, p. xi).

⁴ Scrum: A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value (Schwaber and Sutherland 2016, p. 3).

⁵ For example in the following books: Schwaber (2004, p. 12; 114), (2007, p. 6; 86; 106); Cohn (2009, p. xxvii; 283).

No matter how a team is managed, it is always physically located somewhere. Of course, this is true for Scrum as well. Due to the strong emphasis of collaboration and face-to-face communication, a team room is recommended. While Schwaber states that “this is not yet collocated space” (2007, p. 124), he also states that it is a “full-time room within which they can work”. This effectively means that the team works together in one room, constantly collaborating and communicating. This also makes offshore development difficult (cf. Schwaber 2007, pp. 140–141).⁶ Even though it is not forbidden by Scrum, it is often at odds with the Scrum values and is only feasible if collocated, cross-functional teams in each location work together on the same product across the globe (cf. Schwaber 2007, pp. 140–141). Mike Cohn goes a bit more into detail how a workspace for Scrum teams should look like. He pictures a caves-and-commons workspace where everybody sits together (“commons”) but has the freedom to temporarily move to a quiet or meeting room (“caves”) whenever necessary (cf. Cohn 2009, p. 413). This does not mean, however, that a cafeteria-style open-plan office is created. A maximum number of 20 people in one room—including Product Owner and Scrum Master—are recommended for large projects while making sure that people sit together in teams rather than functions (cf. *ibid.*, p. 415). It is common for Scrum Teams to own movable desks in a large open workspace which they can arrange as they see fit, multiple times a day. No matter how big or small a room is—the walls are plastered with information. Charts about the projects progress, Sprint and Product Backlog, whiteboards, etc. are fundamental for transparency and thus always visible (cf. *ibid.*, pp. 418–420).

The noise level in such a setting hinges on the discipline of the team. However, the investigated authors have said nothing about it.

They are more forthcoming on the question of overtime. Scrum clearly favors the practice of “sustainable pace of work” (Schwaber 2007, p. 4), which means that people should work at a pace that is productive and sustainable into eternity. Actually, this contradicts the name “Sprint” for Scrum iterations, since in sport, human beings are exhausted after sprinting, while in Scrum, people are expected to be fresh and able to continue at the same speed. Usually, overtime does not fit well with sustainable pace. The underlying thinking to this is that “people have many creative moments during downtime” (Schwaber and Sutherland 2012, p. 29) and under pressure, quality is reduced or even dropped completely. Since code quality is essential for the maintainability and extensibility of software, and creativity is indispensable to solve complex problems, waiving overtime should increase the long-term profits of an enterprise. In addition, it increases job satisfaction of employees (cf. Cohn 2009, p. 13).

The monitoring interval also has an important influence on job satisfaction. In Scrum there are three kinds of monitoring. The first one is done by the Development Team for the Development Team on a daily basis in the Daily Scrum meeting as

⁶By the way: Communication is extremely important in complex work, whether you are using Scrum or not. Offshore development is a challenge independent of the methods used.

well as during the day-to-day work. The second kind of monitoring happens at the end of each Sprint when all stakeholders inspect the outcome of the Sprint in the Sprint Review and collaborate on what to do next. The third one is a status update to management and can happen at any time, but is solely managed by the Product Owner. The Development Team is spared of this activity and may only need to change its plan for the next Sprint as a result of the meeting. Basically, the team is “left alone to make its best effort for the rest of the iteration” (Schwaber 2004, p. 6) after the Sprint Planning meeting is over. The rationale behind this is that people work best when they are not interrupted by issues not directly related to their actual work.

The important part is that Scrum does not view “monitoring” in the traditional sense. For Scrum, estimates are not a contract (cf. Schwaber 2004, p. 73) and deviations from plan are normal due to the complex nature of product development. Controlling in Scrum does not mean creating what was predicted but rather “controlling the process to guide the work toward the most valuable outcome possible” (Schwaber 2007, p. 102). Scrum also provides full transparency at any time about every aspect of the project, which usually satisfies all information needs of stakeholders.

This transparency helps the team to deal with the unknowable and uncontrollable. First of all, Scrum accepts the world as being exactly that—unknowable and uncontrollable—and does not even try to achieve certainty (cf. Schwaber and Sutherland 2012, p. 53). To deal with it, Scrum embraces change and “hangs all of its practices on an iterative, incremental process skeleton” (Schwaber 2004, p. 2). This allows Scrum to run a series of experiments (at least one each Sprint, or even one each day), learn from them, and adapt accordingly (cf. Cohn 2009, p. 283). By confining these experiments to small timeboxes, the complexity is reduced as far as possible to simplify the problem. Once such a small portion of complexity is spotted, Scrum forces the team to act instead of long periods of analysis and thinking it over (cf. Schwaber 2007, p. 8). Basically, Scrum meets uncertainty “head on with determination and wit” (ibid., p. 101), together as a team.

While tackling uncertainty, Scrum Teams usually keep a couple of artifacts visible. That is, it has them on a wall, placed openly in the team room, or showing them actively around. It is stated that transparency of Scrum’s artifacts is necessary in order that everybody has the same understanding of the key information (Schwaber and Sutherland 2016, p. 12). Included are the Product Backlog, Sprint Backlog, and the Increment. To track progress across time, usually a burndown chart is added to the mix. Since Scrum teams are self-organizing, they can show anything they want and deem helpful. This could be charts, feedback devices such as lava lamps to show if the build was successful, whiteboards, food and drink, etc. (cf. Cohn 2009, pp. 418–420).

This analysis already provides a deep insight into the cultural characteristics of Scrum. To dig even further and shed some light on more specific aspects, additional literature must be reviewed.

19.3 Specific Cultural Aspects of Scrum

When looking for additional literature about specific aspects of Scrum, one finds a lot about software development practices, scaling Scrum from a couple of teams to many teams, and how to spread several teams around the globe. Those books do not touch culture, though. So far, there seems to be only one culturally relevant area covered by Scrum literature: personnel management. Unfortunately, sources are still scarce there.

André Häusling, together with other authors, published several articles and a German book about agile personnel management. His work often touches cultural aspects of Scrum, which are summarized in this chapter.

Gloger and Häusling (2011, p. 21) contrast traditional with agile organizations to highlight the cultural differences (cf. Table 19.1). They confirm some of the earlier findings: There are no positions or titles, but roles to fill; a flat hierarchy and equality render the position of team lead unnecessary while a Scrum Master takes over coaching of the team; project managers no longer exist but a Product Owner takes over the work of product management; planning for short timeboxes instead of predicting long time spans; full transparency instead of hiding information; involvement of the customer; self-management instead of control; and people enjoy coming to work rather than just earning money.

The authors also stress some existing aspects more and introduce new ones. While all Scrum literature talks about cross-functional teams and the wish for team members with more than one skill, Gloger and Häusling express that Scrum looks for generalists rather than experts. This is a logical consequence due to the fact that in Scrum the whole team focuses on the same goal and all solutions emerge through collaborative work. The one best expert only has one voice in such situations and

Table 19.1 Contrasting traditional and agile organizations

Traditional organization	Agile organization
Position	Role
Expert	Generalist
Team lead	Scrum Master
Product/project manager	Product Owner
Responsibility of line management: Team	Responsibility of line management: Individual
Passiveness	Activeness
Planning of uncertainty over a long time horizon	Planning for a short and clear time horizon
Intransparency	Transparency
Presence	Accomplishment
Customer as alien	Involvement of customers
Delegation of responsibility	Adoption of responsibility
Control	Self-responsibility—positive idea of man
Job	Passion

Based on Gloger and Häusling (2011, p. 21)

needs to build up knowledge of other areas if he wants to be heard on different questions. Being a generalist does not mean of course that an individual does not possess any special expertise. It does mean however, that this special expertise is shared freely and other knowledge is willingly absorbed while the person strives for basic knowledge of all aspects of work (cf. 2011, p. 30).

Since teams are self-organizing and have a Scrum Master to coach them, they actually do not need traditional line management. However, management is still important for the individual employee. Working out individual development paths outside the current project focus, getting individual feedback aside from the teams', and maybe preparing role changes are still important. Therefore, the responsibility of line management shifts from being responsible for the performance of whole teams towards being solely responsible for individual employees with a strong focus on their development.

This duty is performed in a very active way. The expectation to everybody in the organization is that they do not wait for things to happen but rather drive them head on. This means in the case of line management that employees are actively approached and get suggestions rather than waiting for the employee to come to management and make suggestions by themselves. As stressed before, leaders are serving their people. This implies a high level of proactivity.

Viewing proactivity as so important, fulfilling ones responsibilities instead of delegating them is a natural fit. If you love your job and proactively serve your employees, you do not just walk up to one of them and tell her to take over your responsibility. Instead, you focus on it yourself. Even if help is sought—for example due to work overload—the accountability always sticks. In Scrum, nobody will ever get blamed for not fulfilling a delegated responsibility in the expected way. It will always be the person originally responsible who gets her neck wrung and who does not forget this fact. Of course, instead of blame, skill extension and coaching will be provided to improve the likelihood of successful delegation in the future.

This high level of proactivity and awareness of responsibility combined with the strong product focus described above leads to a paradigm shift in terms of what is considered performance. While traditional organizations often equate presence to performance—which becomes apparent for example when late in a project, overtime is demanded even though it is clear that this reduces product quality in a way that will take even more time to fix it again—Scrum emphasizes true accomplishments. While this is obvious in terms of Sprint and release goal achievement, it is also true for evaluating individual or team performance. Nobody gets rewarded for staying late and violating the paradigm of sustainable pace. People are being rewarded for getting the job done.

Gloger and Häusling also emphasize the strong need for management to focus on strategy instead of daily routine. By allowing self-organization to happen, management is freed of precious time to perform other valuable tasks like refining strategy (cf. 2011, p. 23). This fits well with other Scrum literature that most often is written with the assumption that strategy has already been defined elsewhere, and the Product Owner picked it up from there.

A unique nugget from Gloger and Häusling is the sketching of an organizational structure for Scrum (*ibid.*, pp. 29–33). While the traditional line organization has a top and a bottom alongside with divisions—often described as “silos”—Scrum is stated as a specialized form of network organization called “permanent project organization” (Gloger and Häusling 2011, p. 30). Since cross-functional teams without titles, hierarchies, or status symbols are asked to collaborate in the creation of one common product, potentially competing departments or divisions are not helpful.

The Scrum organizational structure consists of a “meta-organization” containing the managers and “benches” of all departments or “job families”, including a Scrum job family from which Scrum Masters can be recruited. A bench is the place where an employee goes to if he is excluded from his team or the team is disbanded. As soon as an employee is part of another team, he leaves the bench and joins the team. The members of the meta-organization are responsible for crafting the company strategy and managing the corporate culture. The managers are also disciplinarily responsible for their employees, spread across the development teams. The cross-functional development teams form the main body of the organization. All necessary members from all relevant functional areas (not only development!) are banded in teams that are kept as stable as possible for as long as needed. These teams are Scrum Teams. That is, they include Developers, a Scrum Master, and a Product Owner. Everybody who contributes to the work is called “Developer” in Scrum, regardless of his main area of expertise. So it doesn’t matter if the person comes from programming, quality assurance, finance or personnel, as long as he contributes (Fig. 19.1).

The important aspect of this organizational model is the constant communication and collaboration amongst all parts of the construct. While the work focus might differ, a common vision prevails and forges closer relationships.

In such an organization, the Scrum values of commitment, focus, openness, courage, and respect of course still apply. In addition—from the perspective of personnel—the following values result: employee orientation, pragmatism, and collaboration in partnership (cf. Gloeden and Häusling 2013, p. 3). Employee orientation shows respect towards the employees and directs the focus of the personnel department. Pragmatism means a solution-driven stance, the commitment to find solutions and the courage to keep them simple. Collaboration in partnership describes the deep respect people and personnel exhibit towards each other, resulting in transparency and loyalty.

This close collaboration for example means that employees are directly involved in the process of hiring new employees (cf. Gloeden and Häusling 2013, p. 11; Gloger and Häusling 2011, p. 55). They will have to work together, so they should have a feeling for how good that might work out as well.

Another aspect of leadership in an agile company is that of feedback. In a traditional setting, employees usually get feedback once a year during their annual appraisal. There are times when this is done in a hurry and quite often the value to the employee is very limited. An agile context demands that bosses talk to their employees frequently, weekly at best (cf. Gloeden and Häusling 2013, p. 30).

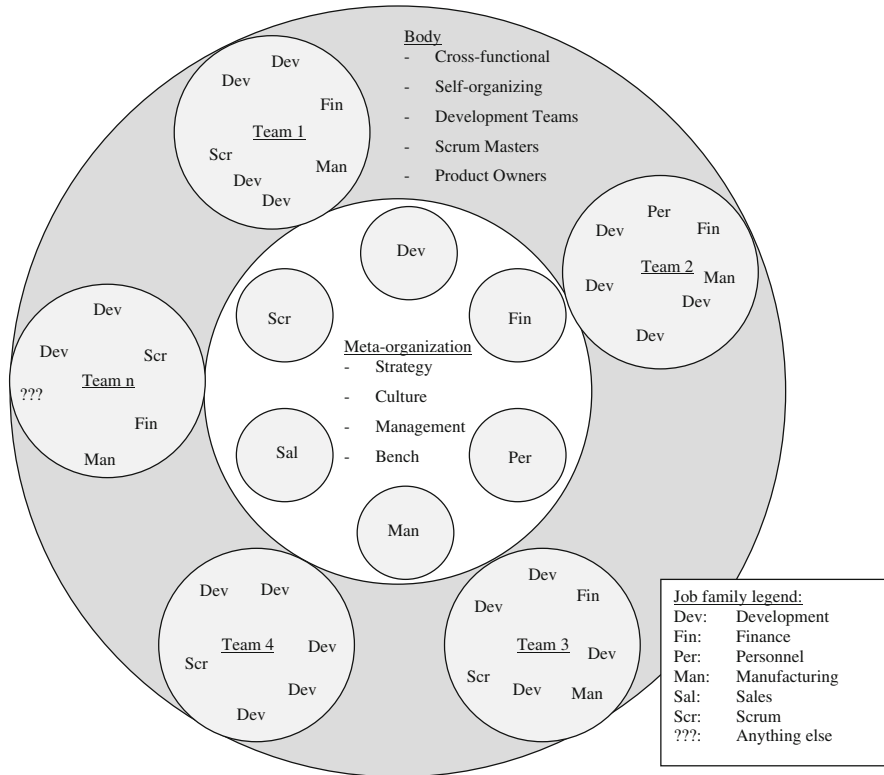


Fig. 19.1 Agile organizational structure (Based on Gloger and Häusling 2011, p. 31)

Mutual feedback is provided and the principle of inspection and adaptation is applied to the people themselves. Depending on the personalities involved, this can happen over lunch, not even demanding additional time slots from the involved parties.

While this literature review confirms the prior findings, it also highlights some additional aspects of an agile culture. It is now time to group and consolidate the findings. A summary can be found in the appendix.

References

- Beck, K. et al. (2001). *Manifesto for agile software development*. <http://agilemanifesto.org/>. 01.01.2013.
- Cohn, M. (2009). *Succeeding with Agile. Software development using scrum*. Amsterdam: Addison-Wesley.
- Gloger, B., & Häusling, A. (2011). *Erfolgreich mit Scrum – Einflussfaktor Personalmanagement*. München: Carl Hanser.
- Gloeden, D. & Häusling, A. (2013, February). *Recruiting 3.0 – Ein Wegweiser für die Suche nach agilen Spezialisten*. eBook. Köln: HR Pioneers.

- Ogunnaike, B. A., & Ray, W. H. (1992). *Process dynamics, modeling and control*. Oxford: Oxford University Press.
- Porter, S. (2011). *Chickens and pigs*. Scrum.org. 17.11.2013. <https://www.scrum.org/About/All-Articles/articleType/ArticleView/articleId/90/Chickens-and-Pigs>
- Schwaber, K. (2004). *Agile project management with scrum*. Redmond: Microsoft Press.
- Schwaber, K. (2007). *The enterprise and scrum*. Redmond: Microsoft Press.
- Schwaber, K., & Beedle, M. (2002). *Agile software development with scrum* (Pearson Internationalth ed.). Upper Saddle River: Prentice Hall.
- Schwaber, K., & Sutherland, J. (2012). *Software in 30 days*. Hoboken: Wiley.
- Schwaber, K. & Sutherland, J. (2016, July). *The scrum guide*. Scrum.org. 28.08.2017. <http://scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf#zoom=100>
- Smith, C. (2012). *The most influential people in Agile*. InfoQ. 17.11.2013. <http://www.infoq.com/news/2012/04/agile-influential-people>
- Takeuchi, H., & Nonaka, I. (1986, January–February). The new new product development game. *Harvard Business Review*, 64(1), 137–146.



Primary Research: The Nature of Scrum Survey

20

20.1 Study Setup

To get an impression of the result quality of the survey, a brief summary of the setup and general results is provided in this chapter. To get the full statistical analysis details, please refer to Sect. 23.3.

The questions for “The Nature of Scrum Survey” were selected following the logic stated in Sect. 18.3 (see Sect. 23.1 for the detailed questions and layout). In order to answer the second research question “What cultural characteristics are inherent to Scrum?” the questions were put in a general, prototypic way and asked in the context of a “perfect Scrum company.” The intention was to prevent respondents from describing their individual experiences with flawed and incomplete Scrum implementations and to state instead what Scrum itself would result in, if no opposition or alteration was present. The goal was to find the inherent cultural characteristics of Scrum and not specific cultural attributes of specific corporations.

To get this deep insight, general questions about the respondents were asked, followed by the OCAI questions to get a broad classification of Scrum. A mix of open and closed questions regarding the dimensions described by Schein (cf. Sect. 18.3) ensued to fill in the cultural picture.

The survey was published both digitally and in paper form in English and German. The digital version was available from 5th of May 2013 to 21st of October 2013. The paper version was handed out and collected the same day at two agile conferences: The Scrum Day Berlin (11th and 12th of June 2013) and the AgileTour Stuttgart (16th of October 2013). The online survey was started 390 times, but there were only 229 qualified responses, meaning that at least the personal information and the questions from the OCAI were completely answered. All replies that did not comply with these criteria were deleted and are thus not part of the study results. The open questions were optional and many respondents did not answer every single one of them. During the survey availability period (cf. Fig. 20.1), 98 qualified responses were gathered digitally and 131 responses

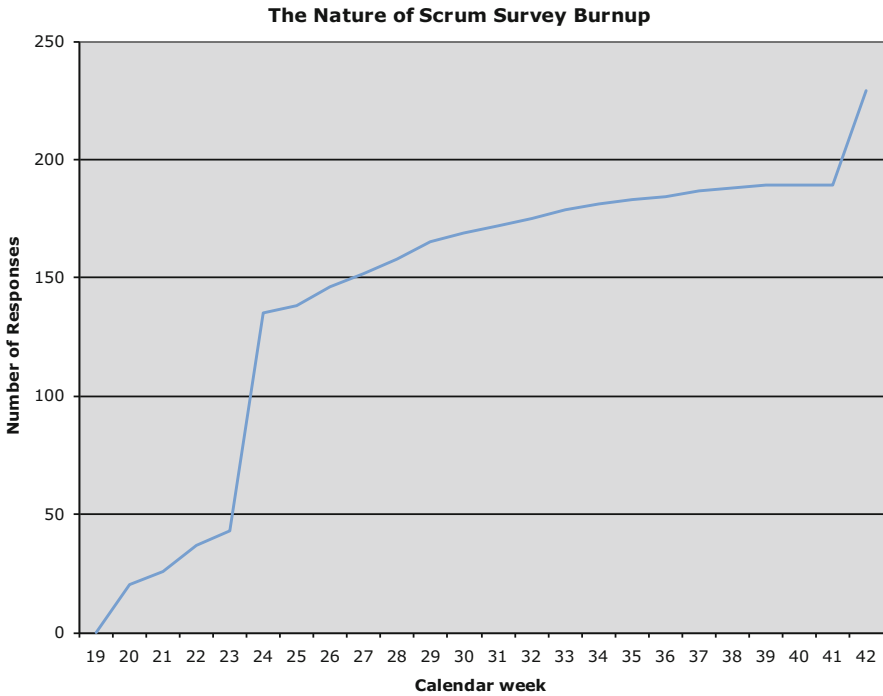


Fig. 20.1 Cumulated responses over time

came in on paper questionnaires, divided in 91 responses from the Scrum Day and 40 responses from the AgileTour.

The digital version was distributed via online networks (XING, Twitter, LinkedIn) and mailing lists connected to Scrum. The personal network of the author and the high number of conference responses results in two biases. Firstly, there is a selection bias: Almost 60 % of the people who responded were experienced or highly experienced in Scrum, equaling or exceeding 3 years of agile experience. Only one quarter of the respondents had 1 year or less of agile experience. This is actually a wanted effect, since the sought after result is a summary of cultural characteristics of Scrum, which should be provided by people experienced in the matter and not just reciting what they have read. Secondly, there is a national culture bias since 165 of the 229 respondents are German and the second largest national group was Dutch, contributing 17 replies. It has to be checked if significant differences between nationalities can be found.

Surprisingly, almost one quarter (50 out of 229) of the respondents were female. This constitutes an overrepresentation compared to the 14 % of female employees

¹Christiaan Verwijs helped by refining the questions in a way that made them statistically analyzable and by running the analysis.

in the German IT sector (cf. BMFSFJ 2005) and offers the opportunity to check if Scrum is seen differently between genders.

The questions were collected with Limesurvey on the author's homepage (<http://scrumorakel.de/surveys/>) and the statistical analysis¹ was performed with SPSS 21. The analysis questions were refined several times throughout the process, whenever more was learned. Three statistical outliers were identified and excluded from the final analysis. The detailed results can be found in Sect. 23.3.

20.2 Findings from the Organizational Culture Assessment Instrument^{© 2}

Before the answers given on the OCAI questions were reviewed closely, the data had to be checked for quality. This included analyzing correlation between the OCAI dimensions and the validity of the measures for this particular sample. Correlation was significant, but relatively low (below 0.85). The strongest correlation seemed to exist between the clan and market cultures with r being -0.68 . This meant the scales (Clan, Adhocracy, Market, Hierarchy) were actually measuring different things.

Validity was also not too high. The Cronbach's alpha scores of Clan (0.72), Adhocracy (0.66), Market (0.63), and Hierarchy (0.7) were all below 0.8, so the reliability could only be considered acceptable. This was expected though, since the OCAI assumes that every company usually has scores on all four culture options and people therefore rarely choose to distribute all their points onto a single alternative. Taking all aspects into account, the results were meaningful and valid given the sample, so the following data is valuable.

The summed scores for all 226 respondents resulted in a clearly dominating "Clan" classification for the perfect Scrum culture, followed by a strong "Adhocracy" influence. "Market" and "Hierarchy" were far behind (cf. Table 20.1, Fig. 20.2). This was still strikingly visible when following the scoring instructions by Cameron and Quinn (2011, p. 33), dividing the scores by 6 to make them more easily comparable with the individual dimensions.

Looking at the details (cf. Table 20.2), some interesting aspects became apparent. First, in the "Dominant characteristics" section, Clan and Adhocracy scored closely to one another. This means that people were picturing a Scrum organization as being both a very personal place, like an extended family, and a dynamic, entrepreneurial place where people are willing to take risks. People are less competitive or achievement oriented. Hardly anybody described a Scrum organization as being controlled, structured or governed by formal procedures.

The leadership in such an organization was described as being mainly nurturing, facilitating, and mentoring. However, both Adhocracy and Hierarchy scored high as well. So innovation, entrepreneurship, and risk taking were important aspects,

² ©Kim Cameron, University of Michigan.

Table 20.1 Means of all OCAI dimensions

Means for	Clan	Adhocracy	Market	Hierarchy
Sum of all dimensions	26.3	17.2	8.2	8.3
Scored (divided by 6)	4.38	2.87	1.37	1.38

Fig. 20.2 Scrum in Cameron’s and Quinn’s culture model (Based on Cameron and Quinn 2011, p. 39)

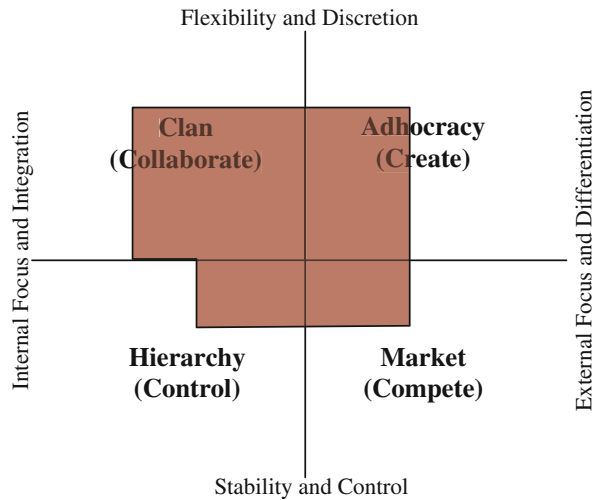


Table 20.2 All OCAI means

Means for	Clan	Adhocracy	Market	Hierarchy
Scored	4.38	2.87	1.37	1.38
Dominant characteristics	3.44	3.48	2.27	0.82
Organizational leadership	4.21	2.68	0.77	2.35
Management of employees	5.39	2.60	0.75	1.24
Organizational glue	4.75	2.52	1.85	0.81
Strategic emphasis	4.76	3.22	0.9	1.12
Criteria of success	3.78	2.71	1.64	1.87

supplemented by coordinating, organizing, and efficiency. The respondents did not see leadership in a Scrum organization as exemplifying a no-nonsense, aggressive, results-oriented focus.

The category of employee management gained the highest single score (5.39) of all dimensions. This indicated that the respondents very strongly felt that management in Scrum is about teamwork, consensus, and participation. Also relevant—although only scoring half of the Clan points—were Adhocracy elements: Individual risk taking, innovation, freedom, and uniqueness. Far less—if at all—important

were competitiveness, achievement, conformity, predictability, and stability in relationships.

The glue that holds Scrum organizations together was described as loyalty, mutual trust, and commitment, supplemented by commitment to innovation, development, and being on the cutting edge. Achievement and goal accomplishment were also seen as important, while formal rules and policies were largely rejected.

Strategic emphasis was described as being put on human development, high trust, openness, and participation. Trying out new things and actively searching for opportunities were also highly valued. The respondents paid little attention to hitting stretch targets, winning in the marketplace, and emphasizing permanence or stability.

Diverging from the overall picture, the criteria of success were described as being quite balanced. Although Clan aspects like development of human resources, teamwork, employee commitment, and concern for people were stressed, high scores were also distributed amongst Adhocracy, Market, and Hierarchy aspects. Having unique products and being an innovator was emphasized, followed by competitive market leadership, and dependable delivery, smooth scheduling, and low-cost production. So while human aspects are promoted, traditional success criteria are important as well. Scrum corporations are still profit-oriented enterprises, as it seems.

This ordering does not fundamentally change with growing experience. However, the differences became more distinct when comparing the experienced survey participants with the inexperienced respondents, strengthening the Clan focus even more and further reducing the Hierarchy elements (Table 20.3).

While experience did have a small effect on the results, gender did not. Comparing the answers of experienced women with that of experienced men did not show any difference. However, there were three times more male respondents than female in the sample and on top of that the females were generally less experienced (on average 2.3 years of experience with agile methodologies compared to 3.7 years). With a larger data sample the results could have been different. For

Table 20.3 OCAI results, experience comparison

Means for	Clan		Adhocracy		Market		Hierarchy	
	≤1	>5	≤1	>5	≤1	>5	≤1	>5
Experience in years								
Dominant characteristics	3.44	3.47	3.21	3.65	2.44	2.15	0.93	0.79
Organizational leadership	3.86	4.47	2.42	2.91	0.77	0.82	2.96	1.82
Management of employees	4.93	5.35	2.72	2.79	0.81	0.82	1.51	1.00
Organizational glue	4.23	5.21	2.77	2.21	1.91	1.79	1.16	0.79
Strategic emphasis	4.33	5.18	3.33	3.15	1.05	0.79	1.28	0.88
Criteria of success	3.40	3.88	2.70	2.71	1.81	1.79	2.09	1.65

this study however, the hypothesis held that the perceived culture of Scrum is not dependent on gender.

Comparing the responses of different nationalities also did not show any difference. This finding was meaningless though, since the vast majority of respondents were German. It could neither be statistically proven nor refuted that people from different countries had different opinions about the inherent culture of Scrum.

20.3 Findings from the Open Questions

Apart from the general and the OCAI questions, the respondents had the opportunity to answer 26 other questions relating to Schein (cf. chapters above), plus two questions making sure nothing was forgotten. Most of them were open questions, even though in some instances a choice had to be made between several options. Whenever open questions were answered, the results were translated, classified, and grouped by me. All references to “low” or “high” experience mean “up to three years” or “more than three years” of experience in agile methods respectively, as stated by the respondents in the questionnaire (so the first two quartiles and the last two quartiles as described in Table 22.2). All results are briefly discussed in this chapter; statistical details and numbers can be found in Chap. 23. All questions are referenced with their original numbers as used in the paper version of the questionnaire (cf. Sect. 23.1).

94.4 % (n = 215) of the respondents stated that people are addressing each other on a first name basis (question 17). Only four said that the family name should be used and eight mentioned that there is no rule for what to use. This clear picture is equally relevant when addressing one’s management (question 18): 89.3 % (n = 215) of the respondents stated that managers are also addressed on a first name basis. Only two respondents saw a formal title as preferable.

Question 19 asked for acronyms and jargon common in a Scrum corporation. The answers largely revolved around Scrum and agile jargon, but also put great emphasis on quality, planning, and learning. It also became clear that people not familiar with Scrum, agile, and the other associated processes would quickly lose track when confronted with a Scrum team. Taking into account that knowing something and having experienced it are two different things, it will be even more difficult for a newcomer to understand the nuances in the meaning, which might lead to misunderstandings or even conflicts.

The dress code (question 20) in a Scrum company was described as being very casual or non-existent. Only very few people mentioned that smart casual (no jeans, regular shirt but no tie) was the preference or that a suit without a tie (business casual) was required. Nobody mentioned a formal business dress. On the contrary, some respondents explicitly stated that a formal dress was forbidden in Scrum. However, it was also mentioned that cleanliness is mandatory and thought is necessary, meaning that when directly facing customers, clothing has to be appropriate according to the company, industry, and culture. At the end of the day, the dress code in Scrum seems to be directly opposing traditional dress norms by

excluding suits from daily work and replacing them with casual clothing and common sense.

The artifacts (question 21) mentioned by the respondents revolved largely around providing transparency for the Scrum process (Task Board, Burndown Charts, Product Backlog, Sprint Backlog, Product Increment). In addition, general process, learning, and quality artifacts were mentioned often. The most interesting fact is that it seems to be completely normal for Scrum teams to have artifacts, to maintain them, and to plaster the walls with them.

This became very obvious when looking at the working space (question 22) in a perfect Scrum company. Half of the respondents mentioned that the environment was personalized and the walls were used for information radiators, posters and so on. It also was strongly emphasized that the team was sitting together in an open space with calm, meeting, and lounge areas. The working environment was also described as being friendly, comfortable, motivating, spacious, bright, creativity promoting, inspiring, communicative, collaborative, and sometimes even chaotic.

Question 23 asked for jokes being told in a Scrum environment. Only 30 % of the respondents entered anything there, even fewer knew some jokes themselves. What became apparent though is that jokes *are* being told and belong to a Scrum environment. The specific jokes mentioned can be found in Table 20.4.

The predominant thing people seem to joke about are other people who are violating the rules of Scrum, who “do not get it”, or who prefer the “waterfall” way of working. The chicken and pig story (cf. Sect. 19.2) reappeared and the squirrel burger was mentioned; both were and are part of the “official” Scrum literature and training material. The other jokes are not distributed officially and thus must have emerged in some teams by themselves. It is worth noting that a chasm seems to exist between “us” (those who are doing Scrum) and “them” (those who are not doing or at least not understanding Scrum). This chasm becomes obvious and might even be deepened by sarcasm expressed in jokes.

The next question asked (24) concerned overtime. Respondents could answer, “despised” or “encouraged”—few paper questionnaires showed “it depends” as well. 76 % of the respondents shared the opinion that overtime is despised in Scrum. Only 15 % saw Scrum encouraging overtime. This is a pretty clear picture conforming to the practice of “sustainable pace” common in Scrum and Agile.

The noise level (question 25) in a perfect Scrum environment was described as being a low but constant buzz of communication, occasionally louder in meetings or discussions. Experienced respondents were significantly more likely to choose “occasionally loud” and significantly less likely to choose “low noise” here, which strengthens the constant communication hypothesis. This finding stresses the importance of having quiet and meeting areas as stated in the findings of question 22. Skipping this and reverting to a full open space environment instead would not be well accepted by a Scrum team. Consequently, only 11 % of the respondents mentioned a loud environment as being part of a perfect Scrum company.

This comfortable noise level contributes to a good feeling at work (question 26). 98 % of all respondents stated in some way that working in a Scrum environment

Table 20.4 Jokes in a Scrum environment

Joke	Rough meaning
“If you find some horse meat in your beef, blame the processing company. But, if it’s a mix of pigs and chicken, call the Scrum Master.”/standard pig and chicken	Reference to commitment and the pigs and chicken story (cf. Sect. 19.2)
Dilbert cartoons	The worse the project, the more “Dilberts” on the walls, usually pointing to the managerial project defects at hand
“How do you catch a pack of wolves? Catch the first one, then the rest!”	Illustrating the incremental and iterative way of working in Scrum
Old stories from waterfall times	Illustrating the dysfunctions of the past and highlighting the advantages of the new approach
“Only 10 types of people exist—those who Scrum and those who don’t—it’s a binary thing”	IT joke with a Scrum reference, dividing the world into two types of people—Scrum followers and Scrum adversaries
“It has always been like that!”	Looking back at the old organization before Scrum, enjoying the fact that change is possible and successful
“I am the biggest fan of Scrum, but please adhere to the gate process”	Since there are no gate processes in Scrum, this illustrates waterfall thinking
“TEAM—toll, ein anderer machts” (that translates into TEAM—great, somebody else does it!)	Obviously, if nobody takes responsibility, there is no team. A real team will make fun of that since it is unthinkable for them
“Reporting to the Scrum Master”	Nobody ever “reports” to the Scrum Master.
“I just completed planning the whole project”	In Scrum, a project only lasts one Sprint. Longer time periods are usually referred to as “product releases”. So when a Scrum Team plans the “whole project”, it actually planned one Sprint. If a traditional project manager planned the “whole project”, he usually talks about a much longer timeframe
“The CFO says, ‘Here is my plan for Q1 through Q4’”	Scrum embraces change and lives in the complex world, accepting that many things are unknown and therefore cannot be planned. A Scrum Team would therefore consider a plan for four quarters highly unstable
“Who laughs last has the highest ping”	Not Scrum related. Standard IT joke about latency in thinking
“I’ll go ahead and get you a Gantt chart for that”	Gantt charts are artifacts of waterfall project planning. In Scrum, burndowns, backlogs and task boards are used
“Chuck Norris is allowed to extend timeboxes”	A timebox may never be extended. Only Chuck Norris can do the impossible!
“Today, am I Scrum Master, Developer, or Manager?”	The roles are very different and each requires full-time attention. If you want to control everything, or if your organization does not understand the demands, role conflicts can emerge

(continued)

Table 20.4 (continued)

Joke	Rough meaning
“50 % team member”	Either you are on the team or not. 50 % is possible but unlikely and points to the thinking in terms of “resources” rather than “people”
Squirrel burger	A story taught in Scrum trainings about a fast food server serving a squirrel to a client instead of what he ordered. The client cannot pay for what he ordered. The server changes the order without making the product change transparent, so the client suffers. This happens often in IT projects when a fixed-price bid is requested and provided even though the contractor clearly knows about the risks and uncertainties

feels very positive and motivating. 11 % explicitly stated that one is eager to come to work again.

One aspect that might contribute to this positive feeling is that of status symbols (question 26). According to the survey participants, status symbols do not play any role. If at all, the team agrees upon the use of symbols, often in the form of gamification. What truly shows the status of a team member in a Scrum company is the behavior of colleagues. People respected highly are sought for counsel and discussion; their opinion is highly valued. Outside symbols like roles, titles, and money on the other hand were only mentioned by 2.6 % of the respondents and can therefore be considered insignificant.

This could be connected to the espoused values in a Scrum environment. Openness, trust, teamwork, and respect led the ranking, closely followed by transparency, honesty, courage, and commitment. Interestingly, communication and collaboration were also named as values by approximately 10 % of all respondents. While it can be discussed if these are values or not, it became very obvious that constant interaction rather than individual heroism is paramount.

Since openness is the most important value, the answers to the following questions (29 and 30) were no surprise. It was stated by 94.3 % of the respondents that disagreement with one’s manager should always be voiced face-to-face in a Scrum environment. 84.1 % also replied that it was okay to state that disagreement in front of others. However, some respondents emphasized that for personal matters a private surrounding is preferred.

The values mentioned above also influence behavior in group meetings (question 31). It was described as being focused and goal oriented, open, respectful, involving, constructive, collaborative, and committed with heated discussions about the best solutions every now and then. In these focused meetings, the opinion (question 32) of everybody was equally valued (57.8 %) while some respondents preferred the opinion of the person considered an expert or most experienced with the topic at hand (23.1 %). This does not mean however, that all decisions have to be made in

full consensus: Only 8 % of the participants mentioned that. Only one person emphasized that the manager's opinion was paramount.

Looking at the insiders and outsiders in a Scrum environment (question 33) it is interesting to see that 56.5 % of all respondents stated that there are no outsiders at all. This number rises to 66.5 % if "team-players" and "those who choose to be insiders" are added. Only 37.3 % stated that outsiders could exist within the company, where "people who are not in the Scrum Team" (38.3 % of the outsiders within the company responses/14.3 % of all responses) and "Someone who does not work with the team" (16.7 %/6.2 %) were the dominant answers. This shows that Scrum is perceived as very inclusive. Pretty much everybody who wants to be part of it can be. However, an undercurrent became transparent, excluding people who are not directly connected to the Scrum Team. This is not only dangerous for project success but also harmful for values such as openness and trust.

Investigating the responses to question 34, the focus of a Scrum organization is described as customer (delight), market, as well as the product. Third place was taken by people and their happiness, followed by a focus on results, value, and continuous improvement. Return on investment and quality also scored above 10 %.

However, instead of rewarding focus (question 35), it is teamwork (41.5 %), openness (15.3 %), and continuous improvement (10.9 %) that are rewarded. Reprimands are delivered for uncooperative, competitive, or antagonistic behavior (50.3 %). Acting against Scrum principles (28.2 %) and lone-wolfing³ (25.4 %) are also prone to punishment. Interestingly, 14.4 % of all respondents mentioned that nothing is punished at all. While this certainly points towards continuous improvement and good team spirit, it most likely does not reflect reality. People do get direct feedback (question 37) about their actions and if they misbehaved, they will be told. Being criticized is the simplest form of punishment—people do not always have to take pay cuts or even worse, lose their jobs.

Switching over to the question of whether management believes people want to work or need external motivators (question 38), the answer was crystal clear. 88.3 % of the respondents stated that the primary motivation of people in Scrum is intrinsic. Only 12.2 % stated that motivation is mixed or extrinsic. Of course, money is important—but in the eyes of the survey respondents only as a hygiene factor, not as the primary reason why people work. This invokes a corresponding leadership style (question 39), primarily described as servant leadership (68.4 %), supported by democratic (35.5 %) and transformational (18.4 %) aspects. Authoritarian, task oriented, and transactional leadership was only described by a very small minority as being present in the perfect Scrum company.

For the inexperienced reader, the question of monitoring (40) was answered in a confusing way. 66.1 % of all respondents stated that monitoring is very rare or is not

³ Lone-wolfing means hunting alone instead of with the pack. While one individual wolf might be a very efficient hunter, the pack always beats it. This applies to complex product development work as well.

happening at all. At the same time, 38.8 % of the respondents said seemingly the opposite, namely that there is some sort of constant monitoring going on. Some even said both. The reason most likely is that there are two levels of monitoring in Scrum: one being team-internal and one being external, for example by line management. Then the responses make sense again: There is basically no management monitoring directed at the Development Team. However, the team tracks its progress themselves on a daily basis, latest during the Daily Scrum. In addition, all progress is tracked on a team basis, not on an individual basis.

If everything is team-oriented, what does promotion look like (question 41)? According to the respondents, promotion is primarily about developing one's own personality and skills. It is not commonly agreed whether there is a traditional promotion system (25.4 %), or not (37.3 %). This concept seems to be hardly understood. More than 10 % of the respondents even explicitly stated that they did not know how promotion takes place in Scrum. So for now, it can be said that more people think there is nothing that equates to promotion in Scrum companies, in comparison to the number of people who believe there is.

The uncertainty about promotion is not the only one people might face. Every enterprise will have to deal with uncertainty in some form (question 42). According to the survey participants, it is dealt with empirically and accepted or even embraced. Only three people stated that Scrum was not dealing with the unknowable in some way, was neglecting that something uncertain might exist, or was dealing with it as every other enterprise does.

A summary of all findings can be found in the appendix.

References

- BMFSFJ. (2005). *Gender-Datenreport - Horizontale Segregation: Frauen- und Männerdomänen*. Bundesministerium für Familie, Senioren, Frauen und Jugend. 2. Fassung, November 2005. 08.12.2013. <http://www.bmfsfj.de/doku/Publikationen/genderreport/2-Erwerbstaetigkeit-arbeitsmarktintegration-von-frauen-und-maenner/2-8/2-8-1-horizontale-segregation-frauen-und-maennerdomaenen,seite=2.html>
- Cameron, K. S., & Quinn, R. E. (2011). *Diagnosing and changing organizational culture* (3rd ed.). Hoboken, NJ: Wiley.



The conclusions are outlined in Sects. 1.3 (Scrum Culture Elements) and 1.4 (Consequences for Enterprises) of this book. Appendix 2 offers even more details for you to dive into. As a result, this chapter only describes the limitations of this study and opportunities for further research.

21.1 Limitations of This Study

The conducted study is subject to some limitations that are discussed in this chapter. While they should not diminish the overall results of the work, it is important to be aware of them.

When this work started, I could not have formulated the questions I needed to ask. Only by extensively analyzing organizational culture literature could an understanding of the topic be evolved. The very choice of literature could represent a limitation since other important other authors or approaches could have been missed. It is possible that a model providing a perfect fit or a questionnaire perfectly matching the study's needs could have been discovered and utilized. While I do not believe this to be the case, it is still possible.

Closely connected to the choice of literature is the choice of questions for the questionnaire. These were derived from literature and the vast majority was not invented by myself. It is possible that important and helpful questions were not identified and used. It is also possible that individual questions may overlap, with a risk of statistically influencing each other or making one another obsolete. Similarly, some questions may not be meaningful for the task at hand. Due to the open nature of the answers and the resulting challenges this poses, this was not statistically tested. In addition, general corporate culture questions were used. If there are special questions necessary for an analysis of Scrum, they were not used.

In addition the scientific community points out issues around definition, measurement, and dimensions of culture. Naturally, since these concepts are not

commonly agreed upon, biases might have been introduced. Where possible, the choices have been made transparent in this work.

Even if the choice of questions was good, I translated them to German. During the course of this translation some language nuances could have been lost. This naturally has the potential to influence the results. The same is true for the respondents: Most of them were German and never saw the English version of the questionnaire. Those who did had to translate it for themselves while filling it in, which again could have led to some differences in interpretation of individual questions or suboptimal phrasing of the answers.

Generally, the response group was relatively small and quite homogenous. This means that a selection bias is possible. Two conferences, which both were co-organized by me, mean a level playing field for everybody, potentially resulting in homogenous responses. Respondents got to know each other and could share their opinions about the survey before and during completion. It is assumed that this happened in some cases. The online respondents also represent a relatively homogenous group since most of them were directly or indirectly connected to the author in some way. Most knew somebody who knew the author and might have been attracted towards completing the survey because they shared a passion for Scrum. While this situation does represent a bias, it was actually desired. The result of this study is a general classification of Scrum, not a representation of existing companies. For this, people were needed who know Scrum well enough to be able to answer the questionnaire with a reasonable level of confidence.

The response group though poses another bias in terms of gender, nationality, and experience. The average survey participant was German, male, and had 3 years or more of experience. Women were over-represented compared to their numbers in business, but under-represented in terms of experience. Therefore, it could not be properly analyzed whether there is a difference between responses from male and female participants. Even though no evidence of data variations by gender distribution could be found, it is certainly possible.

The strong focus on German respondents (72 %) could have introduced national cultural ideas into the results. Unfortunately, there were not enough responses from other countries to statistically analyze significant differences. This is an important circumstance that has to be evaluated in future research.

Independent from gender or nationality, the high level of experience with agile methods could have influenced the overall results as well, even though it was a wanted effect.

Looking at the results, the OCAI answers showed a moderate reliability. Reliability is an indicator of how consistently a metric measures what it intends to measure. The reliability scores were lower than they should have been, following Cameron and Quinn (2011, pp. 175–178). The scores for both market (0.63) and adhocracy (0.66) were especially low. While this does not invalidate the findings, it is a reason to take a closer look with a larger sample. It is unlikely that the OCAI itself is not valid, so other factors might have played a role. Potentially the respondents were having trouble translating the questions—or the translated version lost some nuance. Maybe the nationality of the respondents was an issue since

questions were interpreted differently by Germans than they would have been by Americans. Maybe it is hard to imagine a perfect Scrum organization—which is certainly true—or maybe the respondents were not paying attention when filling in the survey. The ipsative scale of the questions might have supported this. The reasons cannot be identified clearly. Since the weight of the analysis lies on the open questions and those validate the findings of the OCAI, the reliability issue is considered minor.

Looking at the results from the open questions, additional issues can be found. The results were entered by hand at the conferences; when interpreting handwriting, mistakes might have occurred. A small group of respondents did not adhere to the categories printed on the questionnaires—the online participants did not have that opportunity. It is possible that different categories would have produced different results.

All results were translated and grouped by the author, which might have led to poor translations or suboptimal groupings. Even though great care was taken, the author does not claim to be perfect.

What increases the likelihood of a bias here is my profession. I am working as a Scrum trainer and Scrum consultant. On the one hand it is possible that my expertise increased the quality of the groupings. On the other hand it might have led to prejudiced groupings. Since it did not make sense to have somebody unfamiliar with Scrum create groups for the grouping of a Scrum culture, another Scrum expert conducted a crosscheck. This did not produce major corrections but could have also been due to the prejudiced view of that person.

Also, some response categories were overlapping since the respondents did not try to make their answers mutually exclusive.

To avoid such issues, different methods like closed Likert-scales should be used in the future. Unfortunately, this was not possible for this work since it was absolutely unclear what to ask. With the results in your hand, it should be possible now.

The conclusions drawn from the result might also be biased due to the author's expertise. Based on the results, other researchers should check if they would have come to the same conclusions. If not, a sound scientific discussion is desirable.

21.2 Opportunities for Further Research

While this research project provides some valuable insight into the Scrum Culture, it also shows some opportunities for further research. A questionnaire should be developed that allows collecting data without having to ask open questions. This will reduce possible bias and simplify data analysis. Of course, such a questionnaire has to be specific and focus on the right questions.

One of those questions is whether different nationalities see the Scrum Culture differently. A huge data sample from different regions would be required to validate the findings of this study. For example an endeavor could be undertaken to gather 200 responses each from North America, Europe, Asia, Africa, and South America.

For even better results, the continents could be split into individual countries. Each country would have to provide at least 200 responses though.

Such a large data sample would also allow a closer examination of responses with different experience levels. In particular, re-running the OCAI and deep diving into noise levels would provide a useful insight. (Depending on experience levels, this study showed significantly different responses.)

Another question worth answering is if the Scrum Culture is seen differently across genders. Interesting aspects could then be highlighted, for example if female Scrum Masters tend to fill that role better or if male Product Owners should be preferred. Similarly, discovering that there are no differences between male and female respondents would also be a valuable finding.

To increase the value for businesses, the question of promotion systems in an agile context should be studied. This however, was the question that the study participants least understood, and the answers were not consistent. In addition, more experienced respondents showed a tendency to give answers that diverged from less experienced ones, even though the significance threshold was not fully reached. Being able to answer what a well designed, accepted, and culturally matching promotion system looks like could ease the transition to Scrum in many enterprises. Large corporations in particular could benefit here.

Now that the Scrum Culture is defined, its characteristics can be compared to other models. For example, Schein (2010, pp. 366–371) and other authors described the “learning organization” that seems to share some elements of the Scrum Culture. Other models, e.g. that of the networked organization exist which might be compatible. Also, the findings of Spayd (2010) could be revisited and compared to the findings of this study, while taking into account that the scientific value of Spayd’s work is limited.

Once the theoretical side has been thoroughly analyzed, the real world should be checked as well. Real organizations that already have implemented Scrum should be surveyed to find out if they match the Scrum Culture. If they do not match, it has to be evaluated if conflicts are apparent that can be tied back to the diverging and possibly opposing culture. If companies managed to overcome such conflicts in the past, this could be a valuable indication towards transition paths for enterprises. Also, if Scrum works well and without conflict in those organizations, and the Scrum Culture is not fully present, this insight can possibly be used to adjust the Scrum Culture model.

Today two additional topics are moving companies all over the world and especially in Germany. One is that of virtual teams (both distributed and dispersed¹), the other is that of temporary workers, supplied by temporary-employment agencies or consulting firms. Both topics should be analyzed to find out how they impact the Scrum Culture. Legal aspects might be at odds with the

¹ Distributed basically means that two teams are working together, each team at a different location. Dispersed describes the fact that the members of one teams are working from different places.

requirements of productive agile working styles. If that is the case, it must be studied what the most productive way of working is in a complex environment. This must include both monetary and non-monetary aspects (e.g. happiness of employees). In the case that the Scrum Culture is more successful than other approaches, a suggestion for national politics has to be developed in order to change the legal side.

Additional topics can of course be studied and analyzed. Those will have to be identified by other researchers though.

References

- Cameron, K. S., & Quinn, R. E. (2011). *Diagnosing and changing organizational culture* (3rd ed.). Hoboken, NJ: Wiley.
- Schein, E. H. (2010). *Organizational culture and leadership* (4th ed.). San Francisco: Jossey-Bass.
- Spayd, M. (2010). *Agile & culture: The results*. 06.07.2010. Collective edge coaching, 01.01.2013. http://collectiveedgecoaching.com/2010/07/agile__culture/
- Surowiecki, J. (2005). *The wisdom of crowds* (New ed.). Trade Paperback. New York: Anchor Books.
- Sutcliffe, A., Dunbar, R., & Bunder, J. (2012, May). Relationships and the social brain: Integrating psychological and evolutionary perspectives. *British Journal of Psychology*, 103(2), 149–168.

Part VI

Additional Information



Appendix 1: Why John Kotter's Model Was Chosen

22

There are many organizational change models out there. Most only describe change efforts in a general way, but some are specifically applied to a Scrum context. This book focuses very much on Kotter's approach and I want to explain the reasons. In order to do that, I will first briefly describe four different models, starting with two specific to Scrum. Then I will compare all of them and explain my choice.

22.1 How Mike Cohn ADAPTs to Scrum

In his excellent book "Succeeding With Agile" Mike Cohn describes his own transition model for Agile adoptions. The five letters in ADAPT¹ stand for

- **Awareness** that the current process is not delivering acceptable results
- **Desire** to adopt Scrum as a way to address current problems
- **Ability** to succeed with Scrum
- **Promotion** of Scrum through sharing experiences so that we remember and others can see our successes
- **Transfer** of the implications of using Scrum throughout the company

Awareness, Desire, and Ability are overlapping phases; Promotion and Transfer are described as happening continuously throughout the change effort. Cohn emphasizes that the whole ADAPT cycle continues indefinitely since an enterprise living Scrum will always improve and thus always change. The essence of the five phases is highlighted in Table 22.1 (all content was taken from Cohn 2009).

¹ Cohn (2009, p. 21): "The five activities of ADAPT are based on ADKAR (Hiatt 2006), a general model of change that includes the steps of Awareness, Desire, Knowledge, Ability, and Reinforcement."

Table 22.1 The ADAPT essence

Phase	Important statements
Awareness	<ul style="list-style-type: none"> • Change begins with awareness that the status quo is no longer desirable • However, becoming aware that what worked in the past is no longer working can be extremely difficult • Common reasons why individuals can be slow to develop an awareness of the need to change: <ul style="list-style-type: none"> – The company is doing well at the moment – A lack of exposure to the big picture – A refusal to see what’s right in front of us – Confusing motion with progress – Listening to our own propaganda • Tools for developing awareness include <ul style="list-style-type: none"> – Communicate that there’s a problem – Use metrics – Provide exposure to new people and experiences – Run a pilot project – Focus attention on the most important reasons to change
Desire	<ul style="list-style-type: none"> • Beyond being aware of the need to change, one must also have the desire to change • Reasons why developing the desire can be hard include <ul style="list-style-type: none"> – We have been educated to prefer a sequential approach, both through our schooling and years of experience – We have worked hard to get the right boss and the right team—Scrum would change that <ul style="list-style-type: none"> – The timing might not be right • Tools for increasing desire include <ul style="list-style-type: none"> – Communicate that there is a better way – Create a sense of urgency – Build momentum – Get the team to take Scrum for a test drive – Align incentives (or at least remove disincentives) – Focus on addressing fear – Help people let go – Don’t discredit the past – Engage employees in the effort
Ability	<ul style="list-style-type: none"> • All of the awareness and desire in the world won’t get a team anywhere if it does not also acquire the ability to be agile • Some of the larger challenges Scrum teams will face include: <ul style="list-style-type: none"> – Learning new technical skills – Learning to think and work as a team – Learning how to create working software within short timeboxes • Tools for developing ability include <ul style="list-style-type: none"> – Provide coaching and training – Hold individuals accountable – Share information – Set reasonable targets – Just do it. Don’t stall, waiting to know all the answers before you start

(continued)

Table 22.1 (continued)

Phase	Important statements
Promotion	<ul style="list-style-type: none"> • There are three goals during promotion <ul style="list-style-type: none"> – Lay the groundwork for the next pass through the ADAPT cycle – Reinforce agile behavior on existing teams by spreading the news of the good things those teams have achieved – Create awareness and interest among those outside the groups directly involved in adopting Scrum • Groups outside Scrum (e.g. human resources, sales, marketing, operations, facilities, etc.) can have a dramatic influence on the success of your transition • Tools for promoting Scrum include <ul style="list-style-type: none"> – Publicize the success stories – Host an agile safari [employees join an agile team for a couple of weeks] – Attract attention and interest
Transfer	<ul style="list-style-type: none"> • The implications of Scrum must be pushed far enough into other parts of the organization so that the entire transition is not pulled back by organizational gravity • Sources of organizational gravity include <ul style="list-style-type: none"> – Human resources – Facilities – Marketing – Finance • These groups do not need to use Scrum, but they will need to adapt how they interact with the development group

Based on Cohn (2009, pp. 23–40)

Cohn also describes different “patterns” for adopting Scrum (cf. Cohn 2009, pp. 43). He compares starting with one to starting with many teams, and using “stealth” approaches to doing it publicly. The author also describes the use of a transition backlog, an enterprise transition community (ETC) on management level, improvement communities (IC) on the operational level, and having the ETC as well as the ICs work in Sprints.

22.2 Schwaber's Playbook from 'Software in 30 Days'

Ken Schwaber and Jeff Sutherland included “a playbook for achieving enterprise agility” in the appendix of their book “Software in 30 days” (pp. 153–183). This playbook not only describes how Scrum can be introduced but also includes some hints regarding how to scale it. The whole document primarily addresses CXOs who want to introduce Scrum. The following steps (described as “plays”) are described (all content was taken from Schwaber and Sutherland 2012) (Table 22.2):

Schwaber and Sutherland also highlight that the CXOs responsible for the change initiative first have to learn about the business and cultural benefits of Scrum and agility. Then they should apply Scrum to optimize the company's processes. In doing this, the CXO is acting as the “organizational Scrum Master

Table 22.2 The CXO playbook

Play	Important statements
Play 0—overview, assessment and pilot preparation	<ul style="list-style-type: none"> • The objective of the first play is to prepare the playing field for the activities ahead by <ul style="list-style-type: none"> – Assessing the organization's readiness for agility – Providing initial training for the early participants and – Building the Product Backlog for the initial projects • This also includes establishing metrics measuring the success of the use of Scrum and the value derived from it
Play 1—pilot project(s)	<ul style="list-style-type: none"> • The objective of this play is to experience Scrum on one or more real projects in order to demonstrate the positive benefits of improved software agility within the organization • One or more pilots are now executed • The duration should be 3–6 months • After that time a retrospective and re-planning should be conducted to review the Scrum introduction approach
Play 2—organizational expansion	<ul style="list-style-type: none"> • Based on successful pilots, the objective of this play is to expand the usage of Scrum and its benefits to a significant subset of the development organization • Steps to achieve this include: <ul style="list-style-type: none"> – Scrum master training – Product owner training – Developer training – Scrum/agility training [a 2–4 h quick introduction] – Information radiators such as whiteboards, task boards, Product Backlogs, release backlogs, and burndown charts – Reading and providing a list of recommended readings – CXO led seminars, preferably informal ones, to communicate often and openly – Feedback and war stories from the pilots
Play 3—achieving impact	<ul style="list-style-type: none"> • As the pilot projects have proven that real value will be delivered through an agile approach, the objective of this play is to achieve a more significant impact to the bottom line, which can only be demonstrated through more and larger projects • At this point, as much as 25 % of the organization should be involved in the implementation of Scrum • Effective change should now be occurring inside and outside the development organization
Play 4—measure, assess and adjust	<ul style="list-style-type: none"> • The objective of this play is to assess the organization's progress and to establish a broader set of metrics to serve as a basis for further expansion • This also might include letting go of current metrics • The primary metric for agile software development is whether or not working software actually exists, and is demonstrably suitable for use in its intended purpose. In Scrum, that key indicator is determined empirically, by demonstration, at the end of every single Sprint • All other metrics are subordinate to that objective and its constant mantra of “delivering working software more frequently”

(continued)

Table 22.2 (continued)

Play	Important statements
	<ul style="list-style-type: none"> • There are two types of metrics that may be applied: <ul style="list-style-type: none"> – Process metrics, primarily measuring effectiveness of teams, processes, and meetings – Project metrics, aiming at quality and productivity of individual projects
Play 5—expand and win	<ul style="list-style-type: none"> • With these activities behind the organization, and with a defined set of metrics to guide and evaluate future progress on an organization-wide basis, it is now time to expand the use of Scrum across the entire organization • The activities in this phase of the implementation are focused on the further scaling of Scrum within the organization • Steps include <ul style="list-style-type: none"> – Introduce the remaining teams to Scrum – Further refine and share existing practices – Adjust the strict rules of Scrum to better match the need to the organization [don't do this earlier!] – Invite customers to participate in the implementation as Product Owners or Scrum Masters • This phase continues until all teams have adopted Scrum

Based on Schwaber and Sutherland (2012, pp. 166–173)

for continuous improvement” (2012, p. 163) and relies on a backlog of impediments to decide what to tackle next, based on the ordering done by his sponsor.

It is also noteworthy that this playbook was created together with John Kotter.

22.3 Leading Change Like Kotter

You already learned a lot about Kotter throughout this book. I do not intend to repeat over what has already been discussed. However, I will summarize Kotter’s eight steps and highlight some important statements (all taken from Kotter 2012) he made in their context (Table 22.3).

I am sure you are aware that Kotter goes into much more detail. If you start on your own transformation journey, I strongly recommend you to read his book.

Table 22.3 Kotter's approach

Step	Important statements
Establish a sense of urgency	<ul style="list-style-type: none"> • The main goal of creating a sense of urgency is to overcome organizational complacency • The main sources of complacency are <ul style="list-style-type: none"> – The absence of a major and visible crisis – Too many visible resources – Low overall performance standards – Organizational structures that focus employees on narrow functional goals <ul style="list-style-type: none"> – Internal measurement systems that focus on the wrong performance indices – A lack of sufficient performance feedback from external sources – A kill-the-messenger-of-bad-news, low-candor, low-confrontation culture – Human nature, with its capacity for denial, especially if people are already busy or stressed – Too much happy talk from senior management • Ways to raise the urgency level include: <ul style="list-style-type: none"> – Create a crisis by allowing a financial loss, exposing managers to major weaknesses vis-à-vis competitors, or allowing errors to blow up instead of being corrected at the last minute <ul style="list-style-type: none"> – Eliminate obvious examples of excess – Set revenue, income, productivity, customer satisfaction, and cycle-time targets so high that they can't be reached by conducting business as usual – Stop measuring subunit performance based only on narrow functional goals. Insist that more people be held accountable for broader measures of business performance <ul style="list-style-type: none"> – Send more data about customer satisfaction and financial performance to more employees, especially information that demonstrates weaknesses vis-à-vis the competition – Insist that people talk regularly to unsatisfied customers, unhappy suppliers, and disgruntled shareholders – Use consultants and other means to force more relevant data and honest discussion into management meetings – Put more honest discussions of the firm's problems in company newspapers and senior management speeches. Stop senior management "happy talk." – Bombard people with information on future opportunities, on the wonderful rewards for capitalizing on those opportunities, and on the organization's current inability to pursue those opportunities • A majority of employees, perhaps 75 % of management overall, and virtually all of the top executives need to believe that considerable change is absolutely essential
Creating the guiding coalition	<ul style="list-style-type: none"> • Because major change is so difficult to accomplish, a powerful force is required to sustain the process • No one individual, even a monarch-like CEO, is ever able to develop the right vision, communicate it to large numbers of

(continued)

Table 22.3 (continued)

Step	Important statements
	<p>people, eliminate all the key obstacles, generate short-term wins, lead and manage dozens of change projects, and anchor new approaches deep in the organization’s culture</p> <ul style="list-style-type: none"> • A strong guiding coalition is always needed—one with the right composition, level of trust, and shared objective • Building such a team is always an essential part of the early stages of any effort to restructure, reengineer, or retool a set of strategies • Four key characteristics seem to be essential to effective guiding coalitions: <ul style="list-style-type: none"> – Position power – Expertise – Credibility – Leadership • Steps that help to create a guiding coalition include: <ul style="list-style-type: none"> – Find the right people – Create trust – Develop a common goal that appeals to head and heart
<p>Developing a vision and strategy</p>	<ul style="list-style-type: none"> • Vision refers to a picture of the future with some implicit or explicit commentary on why people should strive to create that future • Clarifying the direction of change is important because, more often than not, people disagree on direction, or are confused, or wonder whether significant change is really necessary • An effective vision and back-up strategies help resolve these issues • Characteristics of an effective vision include: <ul style="list-style-type: none"> – Imaginable – Desirable – Feasible – Focused – Flexible – Communicable • The first draft [of a vision] often comes from a single individual • In successful transformations, these ideas are then discussed at length with the guiding coalition • Vision creation is almost always a messy, difficult, and sometimes emotionally charged exercise
<p>Communicating the change vision</p>	<ul style="list-style-type: none"> • The real power of a vision is unleashed only when most of those involved in an enterprise or activity have a common understanding of its goals and direction • Key elements in the effective communication of vision include: <ul style="list-style-type: none"> – Simplicity: All jargon and “technobabble” must be eliminated – Metaphor, analogy, and example: A verbal picture is worth a 1,000 words

(continued)

Table 22.3 (continued)

Step	Important statements
	<ul style="list-style-type: none"> – Multiple forums: Large and small meetings, memos and newspapers, formal and informal interaction—all are effective for spreading the word – Repetition: Ideas sink in deeply only after they have been heard many times – Leadership by example: Behavior from important people that is inconsistent with a vision overwhelms other forms of communication – Explanation of seeming inconsistencies: Unaddressed inconsistencies undermine the credibility of all communication – Give-and-take: Two-way communication is always more powerful than one-way communication
Empowering employees for broad-based action	<ul style="list-style-type: none"> • Major internal transformation rarely happens unless many people assist • Yet employees generally won't help, or can't help, if they feel relatively powerless • Steps to empower people include: <ul style="list-style-type: none"> – Communicate a sensible vision to employees – Make structures compatible with the vision – Provide training employees need – Align information and personnel systems to the vision – Confront supervisors who undercut needed change
Generating short-term wins	<ul style="list-style-type: none"> • Major change takes time, sometimes lots of time • Zealous believers will often stay the course no matter what happens • Most of the rest of us expect to see convincing evidence that all the effort is paying off • Running a transformation effort without serious attention to short-term wins is extremely risky • A good short-term win has at least these three characteristics: <ul style="list-style-type: none"> – It's visible – It's unambiguous – It's clearly related to the change effort • If these characteristics are present, short-term wins: <ul style="list-style-type: none"> – Provide evidence that sacrifices are worth it – Reward change agents with a pat on the back – Help fine-tune vision and strategies – Undermine cynics and self-serving resisters – Keep bosses on board – Build momentum
Consolidating gains and producing more change	<ul style="list-style-type: none"> • Major change often takes a long time, especially in big organizations • Many forces can stall the process far short of the finish line: turnover of key change agents, sheer exhaustion on the part of leaders, bad luck • With complacency up, the forces of tradition can sweep back in with remarkable force and speed • Whenever you let up before the job is done, critical momentum can be lost and regression may follow

(continued)

Table 22.3 (continued)

Step	Important statements
	<ul style="list-style-type: none"> • Important steps include: <ul style="list-style-type: none"> – More change, not less – More help – Leadership from senior management – Project management and leadership from below – Reduction of unnecessary interdependencies
Anchoring new approaches in the culture	<ul style="list-style-type: none"> • Culture will revert all your changes if the new approaches are not anchored in it • The reasons why culture is powerful include: <ul style="list-style-type: none"> – Individuals are selected and well indoctrinated – Because the culture exerts itself through the actions of hundreds or thousands of people – Because all of this happens without much conscious intent and therefore is difficult to challenge or even discuss • Anchoring change in a culture <ul style="list-style-type: none"> – Comes last, not first – Depends on results – Requires a lot of talk – May involve turnover – Makes decisions on succession crucial

Based on Kotter (2012, pp. 41–165)

22.4 How Cameron and Quinn Use the OCAI for Change

Cameron and Quinn shape their change approach around the OCAI^{©,2} of course. They offer a nine-step process that aims at fostering involvement and minimizing resistance (all content was taken from Cameron and Quinn 2011) (Table 22.4).

The OCAI can be used in most steps to increase transparency and effectiveness of the change approach. It is a useful tool, even if another change approach is followed.

² ©Kim Cameron, University of Michigan.

Table 22.4 Cameron's and Quinn's nine step approach

Step	Important statements
Reach consensus regarding the current organizational culture	<ul style="list-style-type: none"> • Identify a set of key individuals in the organization who have a perspective of the overall organizational culture • Be sure to involve people who will be engaged in implementing change initiatives and whose acceptance is necessary for ensuring a successful change effort • Each of these individuals should complete the OCAI • Discussion and consensus regarding the culture must follow
Reach consensus on the preferred future organizational culture	<ul style="list-style-type: none"> • Repeat the process in step 1, this time focusing on the preferred or desired culture • Keep the discussion of current culture separate from that of the preferred culture so that the two are not just reflections of one another • Don't conduct the discussion on an ethereal or blue-sky basis, but ground the discussion on informed projections, specific examples, and verifiable data
Determine what the changes will and will not mean	<ul style="list-style-type: none"> • Plot the current and preferred culture profiles [...] and highlight the discrepancies • Every individual should do this for himself • Keep in mind that trying to move toward one particular type of culture does not mean that other culture types should be abandoned or ignored • The team should reach consensus on the key factors that team members have listed
Identify stories illustrating the desired future culture	<ul style="list-style-type: none"> • The key values, desired orientations, and behavioral principles that are to characterize the new organizational culture are usually more clearly communicated through stories than in any other way • In this step [...] the team should identify an actual incident or event that illustrates the key values they want to permeate the future organizational culture • These incidents or events should be associated with the organization itself so that members can identify with the values being illustrated • Articulate clearly the lessons to be learned and the morals of the stories
Identify a strategic action agenda	<ul style="list-style-type: none"> • With a shared understanding of what it means and doesn't mean to change the organization's culture, as well as what values are to be reinforced, the fifth step involves identifying the strategic agenda that will foster the desired change • A few key strategic actions should be identified in each culture quadrant • As a team, reach consensus on what should be started, what should be stopped, and what we should do more of in order for the culture change process to begin

(continued)

Table 22.4 (continued)

Step	Important statements
	<ul style="list-style-type: none"> • Several key actions must be considered: <ul style="list-style-type: none"> – Create readiness [for change] – Explain why [to change] – Focus on processes – Generate social support: Build coalitions of supporters for the change and empower them – Provide information
Identify immediate small wins	<ul style="list-style-type: none"> • The task associated with this step is to develop a list of a limited number of key action steps that you can execute right away • Identify something that can be implemented immediately—tomorrow morning—to begin the process of change • Out of necessity, these will be small, incremental changes or activities, but they are crucial for starting the process of change • The rule of thumb is: Find something easy to change, change it, then publicize it • Each small change represents a victory • Support is also created because victories built positive energy and enthusiasm
Identify the leadership implications	<ul style="list-style-type: none"> • One leadership implication involves leadership development so that leaders have the wherewithal to lead the process of change and can help create the consensus and collaboration needed to accomplish the goal • Second, leaders must identify ways in which their personal commitments and some types of visible indications of individual responsibility are made obvious
Identify metrics, measures, and milestones to maintain accountability	<ul style="list-style-type: none"> • An important part of the culture change process is to identify measures of what constitutes success, metrics of the key indicators, and milestones to mark progress along the way • Taking a measure means simply to quantify or demarcate what constitutes success • Metrics help individuals to remain accountable
Identify a communication strategy	<ul style="list-style-type: none"> • This final step is to decide on ways in which the message will be spread throughout the organization, what symbols or icons will be developed, and the ways in which commitment among all participants can be ensured • Among the most important changes that accompanies culture change is a change in symbols: logos, letterhead, bumper sticker slogans, signage, colors, or other visible and easily identifiable representations of the organization • New symbols indicate a new future, although symbols cannot change over and over again and be effective

Based on Cameron and Quinn (2011, pp. 102–120)

22.5 Why Kotter?

This should have given you a brief high level overview of these four models. There are of course other models available and it is most certainly up to you which one you choose for your Scrum introduction. They all work. As you can see in the tables above, all models cover each other's steps with small exceptions. The reason for choosing Kotter is the comprehensiveness of his model, the wide adoption throughout the industry, and its scientific acceptance.

While Cameron and Quinn are scientifically well accepted, their model for change does not appear as thorough as Kotter's. While they certainly know how to lead a change program, the reader might not, and thus might interpret too much or too little when trying to apply their approach. The OCAI is nevertheless a powerful tool that can aid a Scrum introduction regardless of the transition model used.

Schwaber and Sutherland are the "fathers of Scrum" and are very well known in the agile community. Every agile practitioner knows them and trusts at least one of them. However, their scientific reputation—and that of Scrum—is small. Both authors have academic degrees and have published books and papers, but their primary goal is not science. They focus on practical implementation. In addition, organizational leaders just embarking on their agile journey might not know them. If they know them, they might not yet trust them—they are advocates of the "new agile stuff" after all, which might potentially threaten these business leaders. The likelihood that such managers know and trust leadership gurus such as John Kotter is higher. On top of that, Schwaber's and Sutherland's model was built with Kotter's help and is based on Kotter's model. In order to fully understand and learn how to apply it, the reader might want to dig into the original literature, which is Kotter's "Leading Change". If on the other hand you have already read Kotter to better understand the playbook, you can start with that model right away.

Mike Cohn is the most influential person in the agile domain, beating even Ken Schwaber in the rankings (cf. Smith 2012). So what is true for Ken and Jeff in terms of being known and trusted applies to Mike as well. Unfortunately, the skepticism of business leaders might also apply to him. While he has written numerous books—including "Succeeding With Agile", which is one of the best books I have ever read about the topic—he is always closer to the technology than to the business. This is good for people with a technology background, such as programming or testing. It is not optimal for people with a business background, because the technical side might not inspire them enough. With a lack of interest, there is a risk they will "switch off" and not gain the full benefits of the learning opportunity the book provides. In addition, managers usually have limited time to read—which puts the comprehensive work of Cohn at a disadvantage.

I want to stress that I enjoyed the books of all authors highlighted above. They did a wonderful job and pushed the agile world a step ahead. My choice for Kotter certainly does not diminish any other work. On the contrary: If you want to become an expert in the domain of organizational change, you will not get around reading all of them. I just had to choose one model for the purpose of this book—otherwise the complexity would have been too great. So there is the answer to the

“why”-question: I chose a single model to reduce complexity and stuck to Kotter because I believe him to resonate best with business leaders.

References

- Cameron, K. S., & Quinn, R. E. (2011). *Diagnosing and changing organizational culture* (3rd ed.). Hoboken, NJ: Wiley.
- Cohn, M. (2009). *Succeeding with Agile. Software development using scrum*. Amsterdam: Addison-Wesley.
- Hiatt, J. (2006). *ADKAR: A model for change in business, government and our community*. Loveland: Prosci Learning Center.
- Kotter, J. (2012). *Leading change* (New ed.). Boston: Harvard Business Review.
- Schwaber, K., & Sutherland, J. (2012). *Software in 30 days*. Hoboken: Wiley.
- Smith, C. (2012). *The most influential people in Agile*. InfoQ. 17.11.2013. <http://www.infoq.com/news/2012/04/agile-influential-people>



23.1 The Nature of Scrum Survey Questions

On the following pages you can see the text and questions of “The Nature of Scrum Survey”. Open questions can be identified by the open textbox. The online version of the questionnaire contained the same questions in the same order.

Please note that the Organizational Culture Assessment Instrument (OCAI) is copyrighted by Kim Cameron, University of Michigan. You are not allowed to use the corresponding questions without his written consent. The other questions were taken from Schein. Even though they are not copyrighted, you will have to provide proper citing to both his and potentially this work, if you want to use it.

The Nature of Scrum Survey

Dear survey participant,

Thank you very much for taking the time to complete this questionnaire. By participating, you not only help science (the results of this survey will constitute a large portion of my Master's thesis), but also enterprises around the world to better understand Scrum. This will especially help companies that are transitioning to Scrum.

This survey consists of 3 parts and 44 questions. First, you will be asked to provide some personal data that will allow me to perform some statistical analysis. The second part consists of the questions from the Organizational Culture Assessment Instrument (©Kim Cameron, University of Michigan), as presented by Kim S. Cameron and Robert E. Quinn in "Diagnosing and Changing Organizational Culture" (2011). It is one, if not the, most widely used tool to diagnose organizational cultures. Last, you are asked to answer a couple of questions, most of which are free text. They rely on (and are partly taken from) Edgar H. Schein's "The Corporate Culture Survival Guide" (1999, 2009).

Analyzed together, those questions should provide a fairly complete picture of the inherent cultural characteristics of Scrum.

If you fill in your name and email address, you will get the survey results, free of charge, once they are available. If you don't want the results, you may leave the name and email fields empty. In any case, your personal information will only be used for this survey and not shared with anybody.

It should take you about 25 min to complete this survey.

If you have any questions, don't hesitate to ask the conference staff or me.

Dominik Maximini

1: Personal Information (will only be used for the purpose of this study)

Name* : _____

Email address: _____

1 Gender Male Female

2 Nationality: _____

3 Total work experience _____ years

4 Total experience with agile methods _____ years

5 Work experience as Scrum Master _____ years

6 Work experience as Product Owner _____ years

7 Work experience as Developer _____ years

8 Work experience as Manager _____ years

9 Work experience as agile Coach _____ years

10 How would you rate your expertise in Scrum?

Poor	Moderate	Acceptable	Good	Excellent

* The fields name and email address are optional. You only have to complete them, if you want to receive the survey results

2: Organizational Culture Assessment Instrument¹

©Kim Cameron, University of Michigan.

This survey is trying to pinpoint the inherent cultural characteristics of Scrum. Since those are difficult to map to organizational culture models, please keep the following question in mind:

Imagine a hypothetical and perfect Scrum company, that has fully adopted Scrum and thrives in it. If you were an employee in that company, how would you answer the following questions?

These questions are not for a single department—focus on the whole organization.

Attention: Please divide 10 points among all options for the questions in this section. Consider the following example:

Question: What ice cream do you like best?

	Option	# of points
A	Strawberry	3
B	Vanilla	1
C	Chocolate	6
D	Peppermint	0

As you can see, the 10 points have been divided between all options. While it is possible to put all 10 points on a single answer, you most likely will want to spread your points. Usually, different organizations—or even a framework like Scrum—will have characteristics of several options, not just those of a single one.

¹ Source: Kim S. Cameron & Robert E. Quinn (2011), “Diagnosing and Changing Organizational Culture”, John Wiley & Sons, pp. 30–32.

11. Dominant Characteristics

distribute 10 points

	Option	# of points
A	The [perfect Scrum] organization is a very personal place. It is like an extended family. People seem to share a lot of themselves	
B	The [perfect Scrum] organization is a dynamic and entrepreneurial place. People are willing to stick their necks out and take risks	
C	The [perfect Scrum] organization is very results oriented. A major concern is with getting the job done. People are very competitive and achievement oriented	
D	The [perfect Scrum] organization is a very controlled and structured place. Formal procedures generally govern what people do	

12. Organizational Leadership

distribute 10 points

	Option	# of points
A	The leadership in the [perfect Scrum] organization is generally considered to exemplify mentoring, facilitating, or nurturing	
B	The leadership in the [perfect Scrum] organization is generally considered to exemplify entrepreneurship, innovation, or risk taking	
C	The leadership in the [perfect Scrum] organization is generally considered to exemplify a no-nonsense, aggressive, results-oriented focus	
D	The leadership in the [perfect Scrum] organization is generally considered to exemplify coordinating, organizing, or smooth-running efficiency	

13. Management of Employees

distribute 10 points

	Option	# of points
A	The management style in the [perfect Scrum] organization is characterized by teamwork, consensus, and participation	
B	The management style in the [perfect Scrum] organization is characterized by individual risk taking, innovation, freedom, and uniqueness	
C	The management style in the [perfect Scrum] organization is characterized by hard-driving competitiveness, high demands, and achievement	
D	The management style in the [perfect Scrum] organization is characterized by security of employment, conformity, predictability, and stability in relationships	

14. Organization Glue*distribute 10 points*

	Option	# of points
A	The glue that holds the [perfect Scrum] organization together is loyalty and mutual trust. Commitment to this organization runs high	
B	The glue that holds the [perfect Scrum] organization together is commitment to innovation and development. There is an emphasis on being on the cutting edge	
C	The glue that holds the [perfect Scrum] organization together is the emphasis on achievement and goal accomplishment	
D	The glue that holds the [perfect Scrum] organization together is formal rules and policies. Maintaining a smoothly running organization is important	

15. Strategic Emphases*distribute 10 points*

	Option	# of points
A	The [perfect Scrum] organization emphasizes human development. High trust, openness, and participation persist	
B	The [perfect Scrum] organization emphasizes acquiring new resources and creating new challenges. Trying new things and prospecting for opportunities are valued	
C	The [perfect Scrum] organization emphasizes competitive actions and achievement. Hitting stretch targets and winning in the marketplace are dominant	
D	The [perfect Scrum] organization emphasizes permanence and stability. Efficiency, control, and smooth operations are important	

16. Criteria of Success*distribute 10 points*

	Option	# of points
A	The [perfect Scrum] organization defines success on the basis of development of human resources, teamwork, employee commitment, and concern for people	
B	The [perfect Scrum] organization defines success on the basis of having unique or the newest products. It is a product leader and innovator	
C	The [perfect Scrum] organization defines success on the basis of winning in the marketplace and outpacing the competition. Competitive market leadership is key	
D	The [perfect Scrum] organization defines success on the basis of efficiency. Dependable delivery, smooth scheduling, and low-cost production are critical	

3: General questions about culture²

This survey is trying to pinpoint the inherent cultural characteristics of Scrum. Since those are difficult to map to organizational culture models, please keep the following question in mind:

Imagine a hypothetical and perfect Scrum company, that has fully adopted Scrum and thrives in it. If you were an employee in that company, how would you answer the following questions?

These questions are not for a single department—focus on the whole organization.

Keep your answers brief. However, if the allocated space is not sufficient, please use the reverse side of the page.

17. Are people in a perfect Scrum company addressing each other on a first name basis or differently?

First name	Family name	Formal title	Other (please specify)

18. Are people in a perfect Scrum company addressing their bosses on a first name basis or differently?

First name	Family name	Formal title	Other (please specify)

19. Please specify all jargon and acronyms that might be common in a perfect Scrum company (aside from PO, SM, Team, PBL, PBI).

20. What dress code is dominating in the perfect Scrum company?

² Mainly based on: Edgar H. Schein (2009), “The Corporate Culture Survival Guide”, New and revised edition, Jossey-Bass.

21. What artifacts (“important tangibles”) are visible in a perfect Scrum company?

22. What does the working space look like in a perfect Scrum company?

23. What jokes are told in a perfect Scrum company? Please note them down, if you know some.

24. Is working overtime encouraged or despised in a perfect Scrum company?

Encouraged	Despised

25. How would you describe the noise level in a perfect Scrum company?

26. How does work feel in a perfect Scrum company?

27. How are different degrees of status symbolized? Are there any sort of uniforms, badges, and so on?

28. What values are espoused in a perfect Scrum company?

29. If you disagree with the boss, do you feel encouraged or discouraged to voice your disagreement face-to-face?

Encouraged	Discouraged

30. Is it OK to disagree in front of others, or do you have to seek the boss out and disagree privately?

It is ok	You have to seek out your boss privately

31. How would you describe behavior in group meetings?

32. Whose opinion is valued most in group meetings?

33. Who is considered an “insider” or “outsider” in a perfect Scrum company?

34. What is a perfect Scrum company focusing on?

35. What kind of behavior is rewarded in a perfect Scrum company?

36. What kind of behavior is punished in a perfect Scrum company?

37. How do you know when you have been rewarded or punished in a perfect Scrum company?

38. In a perfect Scrum company: Does management believe that people *want* to work (intrinsic) or do they believe people need external (extrinsic) motivators to work (e.g. money)?

39. How would you describe the leadership style in a perfect Scrum company?

40. How long is an employee left alone without being monitored in a perfect Scrum company?

41. How does promotion (“climbing up the ladder”) look like in a perfect Scrum company?

42. How does a perfect Scrum company deal with the Unknowable and Uncontrollable?

43. What else do you want to point out in regard to the nature of Scrum?

44. Is there anything you want to tell me about this survey in general?

Thank you very much for your help!

23.2 Statistical Analysis: Descriptives

In this section, the sample is described. Statistical tests and results can be found in the next chapter.

Survey language:	English, German
Form:	Both digital and paper
Availability period:	5th of May 2013 to 21st of Oct. 2013
Distribution:	Scrum Day Berlin (11–12th of June 2013) AgileTour Stuttgart (16th of Oct. 2013) XING, Scrumorakel.de, Scrum email lists, etc.
Surveys started:	390
Surveys completed:	229 (98 digitally, 131 paper)
Conference completion:	91 Scrum Day, 40 AgileTour
Male respondents:	177
Female respondents:	50
Gender not specified:	2
Total # of outliers:	3
ID of outliers:	15, 130, 277
Population after outlier removal (n):	226

The answers were translated, classified, and grouped by the author (expert classification). The full raw data is available upon request. The final grouped data can be found below. If the answers were not grouped into broader categories, the total number of answers and the cap is given.

Geographical distribution of respondents:

Geographical distribution of respondents

Nationality	# of respondents
German	163
Dutch	17
British	5
Danish	5
U.S. American	4
Belgian	3
Canadian	3
Croatian	3
Australian	3
Austrian	2
French	2
Italian	2
Romanian	2
Spanish	2
Swedish	2
Asian	1

(continued)

Nationality	# of respondents
Bosnian	1
Finnish	1
Georgian	1
New Zealand	1
Polish	1
Southern American	1
Turkish	1
Total	226

Experience with agile methods:

Experience with agile methods and quartiles

Experience in years	# of respondents	Quartile	#	Quartile definition
0	19	1	57	0–1 year
0.5	2			
1	36			
2	34	2	83	>1–3 years
2.5	1			
3	48			
4	21	3	52	>3–5 years
5	31			
6	16			
7	2	4	34	>5 years
8	3			
9	1			
10	6			
12	3			
13	2			
15	1			
Total	226		226	

Survey question 19: Please specify all jargon and acronyms that might be common in a perfect Scrum company (aside from PO, SM, Team, PBL, PBI):

Jargon distribution

Type of jargon (n = 131)	# of respondents
Scrum vocabulary and jargon	74
General agile jargon	45
Quality centered jargon	44
Planning focused jargon	36

(continued)

Type of jargon (n = 131)	# of respondents
Extreme programming jargon	32
Learning focus	31
Value centered jargon	25
Software development jargon	24
Team/collaboration centered jargon	22
Delivery centered jargon	19
Business jargon	17
Customer centered vocabulary	11
Lean jargon	10
Self-organization focus	9

Survey question 20: What dress code is dominating in the perfect Scrum company?

Dress code

Dress code (n = 206)	# of respondents
Casual	100
None	75
Smart casual	22
Business casual	9

Survey question 21: What artifacts (“important tangibles”) are visible in a perfect Scrum company?

72 different responses were given and no groupings were made. This table only shows answers given by more than 9 people.

Artifacts

Artifacts (n = 191)	# of respondents
Scrum-Board/Task Board	80
Burndown Charts	59
Product Backlog	53
Sprint Backlog	46
Product Increment	31
Impediment Backlog	26

(continued)

Artifacts (n = 191)	# of respondents
Vision/Vision Board	24
Whiteboard/Flipcharts	21
Release plan	17
Sticky notes	13
Definition of Done	13
Product KPIs	12
Build status/integration status	11
Goals	11
Pictures, posters, toys, table kicker	10

Question 22: What does working space look like in a perfect Scrum company?

Working space in Scrum

Working space (n = 197)	# of respondents
Personalized workspace, including gadgets, walls, posters, etc.	97
Team is sitting together	79
Open/open-space	61
Calm, meeting, and lounge areas	59
Friendly, comfortable, motivating, spacious, bright, creativity stifling, inspiring	52
Communicative, collaborative, chaotic	35
Tidy, clean, structured	20
Mobile, flexible environment	19

Question 23: What jokes are told in a perfect Scrum company?

Jokes

Jokes (n = 68)	# of respondents
Joking as such	24
Jokes about people who “do not get Scrum”/waterfall	18
IT jokes	12
Chicken and pig	9
Joking about oneself/the team	8
Squirrel burger	2
Scrum jokes	1
No jokes about others	1
Jokes about roles overstepping the border (e.g. PO influences estimate, etc.)	1

Question 24: Is working overtime encouraged or despised in a perfect Scrum company?

Overtime encouragement

Overtime (n = 210)	# of respondents
Despised	159
Encouraged	31
It depends	20

Question 25: How would you describe the noise level in a perfect Scrum company?

Noise level

Noise level (n = 198)	# of respondents
Constant buzz/low background noise	94
Low	66
It depends/occasionally loud/loud in meetings	51
Loud/open space	22
A level the team likes/reasonable	10

Question 26: How does work feel in a perfect Scrum company?

How work feels

Work feels (n = 205)	# of respondents
Very good and motivating, including looking forward to come to work again	179
Eager to come to work again	20
Work feels somehow negative or just “normal”	6

Question 27: How are different degrees of status symbolized? Are there any sort of uniforms, badges, and so on?

Status symbols

Status symbols (n = 191)	# of respondents
No symbols/no status at all	150
Team decides on symbols, gamification, etc.	25
Behavior of colleagues shows status	19
Outside symbols, roles, titles, money, etc.	5
It is possible and depends	5

Question 28: What values are espoused in a perfect Scrum company?
(88 different values mentioned, only values with more than 9 mentions included below)

Values

Values (n = 204)	# of respondents
Openness	72
Trust	49
Teamwork/team spirit	48
Respect	41
Transparency	37
Honesty	30
Courage	29
Commitment	27
Communication	22
Value/results/goal orientation	21
Collaboration	20
Accountability/responsibility	17
Innovation	16
Focus	15
Tolerance towards mistakes	12
Customer focus	11
Reliability	11
Good/quality products	10
(human) Appreciation	10
Learning/employee development	10
Openness to criticism	10
Self-organization	10

Question 29: If you disagree with the boss, do you feel encouraged or discouraged to voice your disagreement face-to-face?

Disagree with boss

Voice disagreement face-to-face (n = 212)	# of respondents
Encouraged	200
Discouraged	6
It depends	6

Question 30: Is it OK to disagree in front of others, or do you have to seek the boss out and disagree privately?

Open feedback to boss

Disagree in front of others (n = 214)	# of respondents
It is ok	180
You have to seek out the boss privately	28
It depends	6

Question 31: How would you describe behavior in group meetings?

(52 different behaviors mentioned, only those with more than 9 responses are included below)

Behavior in group meetings

Behavior in group meetings (n = 197)	# of respondents
Focused/no distractions/goal oriented	67
Open	56
Respectful/no interruptions	51
Everyone participates/involving	36
Confident/constructive	34
Interactive/collaborative	27
Committed/dedicated/involved/engaged	25
(heated) Discussions about the best solutions/controversial	25
Having equal rights/no one is superior	24
Effective use of time/punctuality/timeboxing	22
To the point/objective	19
Moderated/facilitated	15
Relaxed/humorous/warm/informal	14
Appreciating criticism	11
Honest	11

Question 32: Whose opinion is valued most in group meetings?

(19 different aspects mentioned, only those with more than 9 responses are included below)

Opinion valued most

Opinion valued most (n = 199)	# of respondents
Everybody's	115
The person who is best/most experienced at that topic	46
The best thought out idea	23
The one of the team/consensus/majority	16

Question 33: Who is considered an “insider” or “outsider” in a perfect Scrum company?

Insider and outsider

Insider and outsider (n = 161)	# of respondents
<i>Insider</i>	
All are insiders/no outsiders	91
Scrum-team members are insider	22
Dev-team members are insiders	11
Team-players are insider	10
<i>Outsider</i>	
People not in the Scrum team	23
People outside the company	13
Someone who doesn't work with the team	10
<i>Broad categories for insiders</i>	
Team-players; everybody, who chooses; etc.	107
Anybody directly attached (or inside) the Scrum-team or the project	36
<i>Broad categories for outsiders</i>	
Outsiders within the company	60
Outsiders outside the company	23

Question 34: What is the perfect Scrum company focusing on?

(59 different aspects mentioned, only those with more than 9 responses are included below)

Focus

Focus on (n = 185)	# of respondents
Customer (delight)/market	56
Product/working software	40
People/happiness	34
Results/delivery	33
Value	32
Continuous improvement/inspection and adaptation	30
ROI/value for the enterprise	25
Quality	22
Vision/goals	15
Innovation	13
Team	11

Question 35: What kind of behavior is rewarded in the perfect Scrum company?
(66 different aspects mentioned, only those with more than 9 responses are included below)

Rewarded behavior

Rewards for (n = 183)	# of respondents
Teamwork/collaboration	76
Open mindset/openness	28
Improving	20
Supportive behavior/helping others	18
Focus/goal orientation	16
Innovation	16
Initiative/engagement	14
Goal reaching	10
Learning	10
Transparency/feedback	10

Question 36: What kind of behavior is punished in the perfect Scrum company?

Punished behavior

Punishments for (n = 181)	# of respondents
Uncooperative, competitive, antagonistic	91
Acting against Scrum principles	51
Lone-wolfing/heroes	46
Nothing is punished	26
Harming output and productivity	22
Favoring “waterfall” behavior	16
Harming the product	13
Harming the company	11

Question 37: How do you know when you have been rewarded or punished in a perfect Scrum company?

Knowing about rewards and punishment

Knowing about rewards and punishment (n = 147)	# of respondents
Direct communication/feedback	107
One can feel it	22
Your pay depends on it (bonus, raise, etc.)	9
Reward: People value your opinion	8
Punishment: People don't talk to or despise you	7
Team/sprint goal reached	7
Reward: Special team event (pizza, beer, . . .)	4
Punishment: You are voted out of the team	3

Question 38: In a perfect Scrum company: Does management believe that people want to work (intrinsic) or do they believe people need external (extrinsic) motivators to work (e.g. money)?

Nature of motivation

Nature of motivation (n = 202)	# of respondents
Largely intrinsic	174
Mixed or extrinsic	24
It depends on the people: 80 % of people want to work, 20 % do not want to work	7

Question 39: How would you describe the leadership style in a perfect Scrum company?

Leadership style

Leadership style (n = 188)	# of respondents
Servant leadership	104
Democratic	54
Transformational	28
Transactional	10
Authoritarian	2
Task oriented	1

Question 40: How long is an employee left alone without being monitored in a perfect Scrum company?

Monitoring intervals

Monitoring intervals (n = 183)	# of respondents
None or rare monitoring	121
Some sort of constant monitoring (team not explicitly mentioned)	47
The team is always there	24

Question 41: How does promotion (“climbing up the ladder”) look like in a perfect Scrum company?

Promotion

Promotion (n = 160)	# of respondents
Personal development in some way	67
There is not really something like a promotion	50
In some way a “promotion” exists, including functional career	34
More pay/money	26
I don't know	17

Question 42: How does a perfect Scrum company deal with the Unknowable and Uncontrollable?

(38 different aspects mentioned, only those with more than 15 responses are included below)

Dealing with the unknowable and uncontrollable

Dealing with the unknown (n = 191)	# of respondents
Empirical approach/learn/try things out	100
It is dealt with/faced	34
It is accepted	32
It is embraced	24
React quickly/flexible	17
It is evaluated	16

23.3 Statistical Analysis: Data Preparation

In this section, all statistical analyses and results are presented. Since you will only refer to this section if you have affinity for statistics, only a few descriptions were added.³ The more easily interpretable version can be found in the previous chapters.

Outliers can easily dampen the power of significance tests by increasing the chances that the assumption of normality is violated. The assumption of normality is highly important for the statistical methods used. In order to find outliers, the responses on the OCAI scales were examined. All respondents with at least three extreme scores in different responses were removed. This led to three outliers and a resulting sample size of 226. Skew and kurtosis were calculated before and after the removal of the outliers:

Outlier identification

Skew		Kurtosis	
Initial	After correction	Initial	After correction
0.625	0.442	1.249	-0.007
-0.319	-0.246	0.271	0.246
0.311	0.299	-0.552	-0.572
0.521	0.444	0.049	-0.092

23.3.1 The OCAI Analyses Follow from Here

OCAI reliability is moderate:

OCAI reliability

Scale	Cronbach's alpha
Clan	0.72
Adhocracy	0.66
Market	0.63
Hierarchy	0.70

³ Christiaan Verwijs, who also helped me formulate the questions in a way that they could be analyzed, performed most of those analyses. In addition, he was a great help in teaching me how to interpret the results.

OCAI Correlation:

OCAI correlations

	Clan	Adhocracy	Market	Hierarchy
Clan	–	–0.272	–0.680	–0.450
Adhocracy	–0.272	–	–0.218	–0.553
Market	–0.680	–0.218	–	0.273
Hierarchy	–0.452	–0.553	0.273	–

All OCAI correlations are significant ($p < 0.01$)

Hypothesis 1 Determining the ordering of OCAI scores.

Summed scores for all respondents:

OCAI scores

	Clan	Adhocracy	Market	Hierarchy
Mean	26.3363	17.2124	8.1770	8.3009
Std. Dev.	6.98584	5.68656	4.62501	5.06208
n	226	226	226	226

The sums are significantly different as calculated with an ANOVA ($F = 972.151$, $df = 3$, $p = 0.01$). A Tukey’s HSD post-hoc test also shows that all individual sums are significantly different ($p < 0.01$), except for Market > Hierarchy. Therefore, OCAI ordering is Clan > Adhocracy > Hierarchy > Market.

Hypothesis 2 Individual dimensions are the same as the general OCAI ordering.

Scores on individual OCAI dimensions

	Clan		Adhocracy		Market		Hierarchy	
	Means	Std. Dev.	Means	Std. Dev.	Means	Std. Dev.	Means	Std. Dev.
Dominant characteristics ($p < 0.01$)	3.44	1.911	3.48	1.708	2.27	1.729	0.82	1.010
Organizational leadership	4.21	1.761	2.68	1.498	0.77	0.967	2.35	1.542
Management of employees	5.39	1.826	2.60	1.538	0.75	1.002	1.24	1.253
Organizational glue	4.75	1.812	2.52	1.473	1.85	1.345	0.81	1.130
Strategic emphasis	4.76	1.656	3.22	1.495	0.90	1.101	1.12	1.261
Criteria of success	3.78	1.861	2.71	1.609	1.64	1.518	1.87	1.646

For “dominant characteristics”, all means are significantly different ($F = 134.428$, $df = 3$, $p < 0.001$). MSw is 2.645. A Tukey HSD shows that all means are individually different ($p = 0.01$), except for Clan > Adhocracy.

For “organizational leadership”, all means are significantly different ($F = 211.435$, $df = 3$, $p < 0.001$). MSw is 2.164. A Tukey HSD shows that all means are individually different ($p = 0.01$).

For “management of employees”, all means are significantly different ($F = 473.907$, $df = 3$, $p < 0.001$). MSw is 2.068.

For “organizational glue”, all means are significantly different ($F = 294.321$, $df = 3$, $p < 0.001$). MSw is 2.135. Again, Tukey HSD shows that all means are also individually and significantly different ($p = 0.01$).

For “strategic emphasis”, all means are significantly different ($F = 390.835$, $df = 3$, $p < 0.001$). MSw is 1.945. With Tukey HSD, all means are significantly different ($p = 0.01$), except for Market > Hierarchy.

For “criteria of success”, all means are significantly different ($F = 76.764$, $df = 3$, $p < 0.001$). MSw is 2.766. All means are individually different ($p = 0.01$), except for Market > Hierarchy.

Although minor deviations from the hypothesis could be found, the overall pattern is well matched. Again, OCAI ordering is Clan > Adhocracy > Hierarchy > Market.

Hypothesis 3 Experience does not influence OCAI ordering.

First, the experience dimension has to be chosen. Therefore, correlations of some promising experience indicators are calculated.

Correlation of different expertise measures

	Work experience	Experience in agile methods	Experience as Scrum master	Self-rated expertise
Work experience	–	0.236*	0.162**	–0.005
Experience in agile methods	0.236*	–	0.636*	0.551*
Experience as Scrum master	0.162**	0.636*	–	0.506*
Self-rated expertise	–0.005	0.551*	0.506*	–

*Significant at 0.05 ($n = 226$)

**Significant at 0.01

Self-rated expertise, experience with agile methods, and experience as Scrum Master correlate well. Self-rated expertise only correlates weakly with agile experience, so people do not rate themselves according to their true experience. Therefore, experience with agile methods is chosen as variable for further analysis. See Table 22.2 for the quartile definition.

OCAI scores after experience

Quartile	1 (n = 57)		2 (n = 83)		3 (n = 52)		4 (n = 34)	
Sum	Means	Std. Dev.	Means	Std. Dev.	Means	Std. Dev.	Means	Std. Dev.
Clan	24.19	6.43	27.59	7.15	25.88	7.55	27.59	4.77
Adhocracy	17.16	5.62	17.54	5.74	16.62	5.98	17.42	4.33
Market	8.79	4.81	7.40	4.59	8.75	4.60	8.18	5.38
Hierarchy	9.93	4.93	7.47	4.69	8.73	5.55	6.94	5.81

The scoring on OCAI dimensions is comparable across all four experience quartiles. The scores for Clan ($F = 3.190$, $df = 3$, $p = 0.25$) and Hierarchy ($F = 3.792$, $df = 3$, $p = 0.011$) are different. This means that more experienced respondents give higher scores to Clan than less experienced respondents, while giving lower scores to Hierarchy. The overall ordering is still the same though (Clan > Adhocracy > Hierarchy > Market).

Next it has to be checked if that holds true for the individual dimensions as well:

Individual OCAI dimension scores after experience

	Clan	Adhocracy	Market	Hierarchy
Inexperienced (0–1 years of experience with agile methods)				
Dominant characteristics	3.44	3.21	2.44	0.93
Organizational leadership	3.86	2.42	0.77	2.96
Management of employees	4.93	2.72	0.81	1.51
Organizational glue	4.23	2.77	1.91	1.16
Strategic emphasis	4.33	3.33	1.05	1.28
Criteria of success	3.40	2.70	1.81	2.09
Experienced (>5 years of experience with agile methods)				
Dominant characteristics	3.47	3.65	2.15	0.79
Organizational leadership	4.47	2.91	0.82	1.82
Management of employees	5.35	2.79	0.82	1.00
Organizational glue	5.21	2.21	1.79	0.79
Strategic emphasis	5.18	3.18	0.79	0.88
Criteria of success	3.88	2.71	1.79	1.65

An ANOVA on these scores with “experience with agile methods” as a factor reveals that the scores do not significantly differ for all dimensions, except for one: The hierarchy scores for organizational leadership ($F = 5.048$, $df = 3$, $p = 0.002$). This shows that more experienced respondents put less emphasis on hierarchical styles of leadership than inexperienced respondents. The overall ordering of the dimensions is still the same though (Clan > Adhocracy > Hierarchy > Market).

Hypothesis 3 was confirmed with one minor deviation on the dimension of organizational leadership.

Hypothesis 4 There are no gender differences in how the perfect Scrum culture is perceived.

In the sample (after outlier removal), there are 49 women and 175 men. In addition, the men in the sample are significantly more experienced with Scrum (experience with agile methods) than women ($F = 10.235$, $df = 1$, $p = 0.002$). This has to be controlled in order not to skew the results. See the means and standard deviations (in brackets) below:

Comparison of responses for gender and experience

	Men		Women	
	All levels (n = 175)	High experience (n = 74)	All levels (n = 49)	High experience (n = 17)
Clan	26.36 (7.10)	25.76 (6.63)	26.16 (6.62)	24.12 (5.20)
Adhocracy	16.77 (5.97)	17.05 (5.79)	18.65 (4.35)	18.12 (4.03)
Market	8.44 (4.69)	8.51 (4.71)	7.41 (4.35)	8.77 (4.37)
Hierarchy	8.46 (5.26)	8.74 (5.35)	7.82 (4.20)	9.12 (3.62)

While women score significantly higher on adhocracy, this effect disappears when controlling for high experience. Therefore, the significance could be an effect of the small sample size. All other dimensions are not significantly different. For this study, hypothesis 4 has to be accepted and can be revisited with a larger data sample at a later time.

23.3.2 Now the Analyses of the Open Questions Follow

The answers were translated, classified, and grouped by the author (expert classification). Generally, the response categories were analyzed with cross tabulations, experience with agile methods was entered as predictor. To simplify the process, quartiles were summed up: The lower two quartiles and the higher two quartiles resulted in “low experience” (≤ 3 years) and “high experience” (> 3 years). Significance was tested with Cramer’s V (nominal \times nominal), because it is robust against both nominal and ordinal data. Individual subsets were tested with a Bonferroni corrected p-test in SPSS 21. In the following tables, numbers in brackets represent the absolute number of responses. Brackets in the headline show the total number of survey participants in that experience category who entered anything that is being reported. This means, that n here can be different from the total n for that question because some answers did not make it into a cluster. Percentages are always mapped to those numbers. If somebody did not enter anything, that person is excluded from these numbers.

Hypothesis 5 Noise level is a constant buzz but never too loud.

Noise levels

Response category	Experience with agile methods		
	Low (120)	High (78)	Total (198)
Depends/occasionally loud/loud meeting ^a	20.8 % (25)	33.3 % (26)	25.8 % (51)
Reasonable/A level the team likes ^b	5.8 % (7)	3.8 % (3)	5.1 % (10)
Constant buzz/low background noise ^b	46.7 % (56)	48.7 % (38)	47.5 % (94)
Loud/open space ^b	10.8 % (13)	11.5 % (9)	11.1 % (22)
Low ^c	39.2 % (47)	24.4 % (19)	33.3 % (66)

^aSignificantly different ($X^2 = 3.863$, Df = 1, $p = 0.49$)

^bNo significant difference for experience

^cSignificantly different ($X^2 = 4.664$, Df = 1, $p = 0.31$)

Experienced respondents make a stronger distinction between silent work and occasionally loud outbursts or meetings. More experienced respondents also emphasize that noise is “low” less often. Hypothesis 5 is clearly confirmed, since both support the finding of a constant buzz.

Hypothesis 6 Work in a perfect Scrum organization feels good.

Work feelings

Response category	Experience with agile methods		
	Low (110)	High (72)	Total (182)
Very good and motivating ^a	98.2 % (108)	98.6 % (71)	98.4 % (179)
Eager to come to work again ^a	14.5 % (16)	5.6 % (4)	11.0 % (20)
Work feels somehow negative or just normal ^a	2.7 % (3)	4.2 % (3)	3.3 % (6)

^aNo significant difference for experience

Hypothesis 6 confirmed.

Hypothesis 7 Status symbols are irrelevant in Scrum.

Status symbols

Response category	Experience with agile methods		
	Low (115)	High (76)	Total (191)
Behavior of colleagues shows status ^a	10.4 % (12)	9.2 % (7)	9.9 % (19)
Team decides on symbols, gamification ^a	13.0 % (15)	13.2 % (10)	13.1 % (25)
Outside symbols, roles, titles, money ^a	0.9 % (1)	5.3 % (4)	2.6 % (5)
It is possible and depends ^a	3.5 % (4)	1.3 % (1)	2.6 % (5)
Status symbols do not play any role ^a	77.4 % (89)	80.3 % (61)	78.5 % (150)

^aNo significant difference for experience

The hypothesis could clearly be confirmed.

Hypothesis 8 In Scrum, intrinsic motivation is stronger than extrinsic motivation.

Intrinsic and extrinsic motivation

Response category	Experience with agile methods		
	Low (124)	High (73)	Total (197)
Largely intrinsic ^a	86.3 % (107)	91.8 % (67)	88.3 % (174)
Mixed or extrinsic ^a	14.5 % (18)	8.2 % (6)	12.2 % (24)

^aNo significant difference for experience

Hypothesis 8 could clearly be confirmed.

Hypothesis 9 The leadership style is not authoritarian.

Leadership styles

Response category	Experience with agile methods		
	Low (88)	High (64)	Total (152)
Task oriented ^a	0.0 % (0)	1.6 % (1)	0.7 % (1)
Authoritarian ^a	1.1 % (1)	1.6 % (1)	1.3 % (2)
Democratic ^a	37.5 % (33)	32.8 % (21)	35.5 % (54)
Transactional ^a	9.1 % (8)	3.1 % (2)	6.6 % (10)
Transformational ^a	19.3 % (17)	17.2 % (11)	18.4 % (28)
Servant leadership	62.5 % (55)	76.6 % (49)	68.4 % (104)

^aNo significant difference for experience

Hypothesis 9 is clearly confirmed. Servant leadership traits were identified most often, followed by democratic and transformational aspects. Hardly anybody named authoritarian traits.

Hypothesis 10 There is no traditional promotion.

Promotion

Response category	Experience with agile methods		
	Low (77)	High (57)	Total (134)
Personal development in some way ^a	51.9 % (40)	47.4 % (27)	50.0 % (67)
In some way, "promotion" exists, including a functional career ^a	23.4 % (18)	28.1 % (16)	25.4 % (34)
There is not really something like promotion ^a	32.5 % (25)	43.9 % (25)	37.3 % (50)
More pay/money ^a	20.8 % (16)	17.5 % (10)	19.4 % (26)

^aNo significant difference for experience

Hypothesis 10 is confirmed. However, the concept of promotion seems not to be well understood and commonly agreed upon. Only very few respondents actually answered something here.

Hypothesis 11 The unknown is embraced.

Dealing with the unknown and uncontrollable

Response category	Experience with agile methods		
	Low (90)	High (53)	Total (143)
Empirical approach/learn/try things out ^a	71.1 % (64)	67.9 % (36)	69.9 % (100)
It is dealt with/faced ^a	24.4 % (22)	22.6 % (12)	23.8 % (34)
It is embraced ^a	13.3 % (12)	22.6 % (12)	16.8 % (24)

^aNo significant difference for experience

The hypothesis can neither be proven nor disproven. 16.8 % of all respondents see Scrum as embracing change. What can be said is that 69.9 % of all respondents see Scrum as dealing empirically with the unknowable. The main reason for the unclear answer is the diverse data. It was difficult to group, so many individual answers are available, most of them not clustered. However, there were only three answers stating that Scrum does not deal with the unknowable at all, it is dealt with as in every enterprise but without risk management and that there is nothing unknowable in Scrum. So one thing is clear: Participants believe that Scrum accepts the uncontrollable, faces it and solves it.

23.4 Summary of Scrum’s Cultural Characteristics According to Literature

After reviewing the standard and specialized literature and discussing the contents, the findings can be mapped on to the initially identified categories of Schein as follows:

Summary of Scrum’s cultural characteristics according to literature

Schein’s category	What to find out	Finding from literature review
Common language and concepts	Technical jargon	– Team related jargon (Scrum, self-managing, self-organizing, cross-functional, responsible, sustainable pace, purge people to the bench) – Empirical process control jargon (Complexity, transparency, inspection, adaptation, embracing change, good enough, the art of the possible, failing early/fast, iterations, iterative, incremental, sashimi) – Product related jargon (Potentially shippable, done, quality) – Planning related jargon (Burndown chart, Sprint goal, timeboxing) – Business jargon (Value Driven Development, Return on Invest,

(continued)

Schein's category	What to find out	Finding from literature review
		Total cost of ownership, business value, productivity) – IT jargon (Technical excellence, keeping the code well factored, simple design, automated testing, early detection of errors) – Leadership jargon (Servant leadership, Product Owner, Scrum Master, single wringable neck, impediments) – Predictive process modeling jargon (adversaries) (Waterfall, predictive, defined, command-and-control)
Common language and concepts	Emotional jargon	– Pigs and chickens – Single, wringable neck
Group boundaries and identity	Dress norms	– None mentioned
Group boundaries and identity	Badges, uniforms, symbols or privileges	– No titles other than Developer – No uniforms or badges – No status symbols – Being on a good team is a privilege
Group boundaries and identity	Insider and outsider	– Insider: Scrum Team and people who think “agile” – Outsider: People with old “waterfall” thinking or striving for absolute certainty
Nature of authority and relationships	Formal or informal relationship between people	– Informal relationship
Nature of authority and relationships	Formal or informal relationship with bosses	– Informal relationship
Nature of authority and relationships	Pecking order in meetings	– Collaborative team approach – Creative team discussions – Facilitation of discussions – Decisions are collectively determined
Nature of authority and relationships	Source of authority	– All opinions weigh equal – Product Owner has the last say on the requirements – Scrum Master has the last say on the Scrum process
Nature of authority and relationships	Openly voiced criticism (peers)	– Openness and honesty are wanted, even if it disappoints somebody – Voicing criticism is encouraged

(continued)

Schein’s category	What to find out	Finding from literature review
Nature of authority and relationships	Openly voiced criticism (boss)	<ul style="list-style-type: none"> – Openness and honesty are wanted, even if it disappoints somebody – Voicing criticism is encouraged – Line management: The standard literature doesn’t say anything about it. Specialized literature hints to frequent mutual feedback though
Allocation of rewards and status	How to gain power	<ul style="list-style-type: none"> – There is no promotion (titles) – There is no ladder to climb/flat hierarchies – Change of role is possible – Reward by greater salary – Reward by greater responsibility, e.g. more important projects
Allocation of rewards and status	What is rewarded	<ul style="list-style-type: none"> – Teamwork – Active, engaging participation – Helpful behavior – Getting the job done
Allocation of rewards and status	What is punished	<ul style="list-style-type: none"> – Behavior that degrades performance or productivity of the team – Violation of the Scrum rules
Allocation of rewards and status	Reward mechanisms	<ul style="list-style-type: none"> – Immediate communication by team members – Rule violations are communicated by Scrum Master – Exclusion from meetings or team can follow
The nature of human nature	Are people intrinsically or extrinsically motivated	<ul style="list-style-type: none"> – Intrinsic motivation/theory Y
The nature of human nature	Like people coming to work	<ul style="list-style-type: none"> – People look forward coming to work
The nature of human relationships	Espoused values	<ul style="list-style-type: none"> – Named values: commitment, focus, openness, respect, courage – Additional values: honesty, visibility/transparency, emergence, inspection, adaptation, embrace change, the art of the possible, teamwork, continuous improvement, learning, self-organization, empowerment, collaboration, face-to-face communication, involvement, act – Values of the Agile Manifesto – Values from a personnel perspective: Employee orientation, pragmatism, collaboration in partnership
The nature of human relationships	Focus	<ul style="list-style-type: none"> – Continuous improvement – Teamwork – Product/product increment – People – Value – Planning (agile)

(continued)

Schein's category	What to find out	Finding from literature review
		<ul style="list-style-type: none"> – Customer – Market – Accomplishment rather than presence
The nature of human relationships	Leadership style	<ul style="list-style-type: none"> – Decentralized decision-making at the front – Servant-leadership – Self-managing teams – Empowerment – Facilitation – Mentoring and coaching – Like a parent: Growing people so that they are mature and self-managing – Line management cares for the development of each individual employee – Frequent mutual feedback loops (e.g. weekly) with employees should be established – Active living of ones responsibility – Management focuses on strategy instead of micro-management
Nature of Space	Office design	<ul style="list-style-type: none"> – Team rooms – Collocated space – Offshore development only with cross-functional teams in each location – Caves-and-commons: Open team room plus silent areas and meeting rooms Maximum number of 20 people per room, including Scrum Master and Product Owner – Movable desks – Teams arrange their furniture themselves, multiple times a day if needed – Walls are plastered with information – Whiteboards and charts are everywhere
Nature of space	Communication amount in the environment	<ul style="list-style-type: none"> – Communication amount is high – Nothing has been found about the noise level
Nature of time	Overtime encouragement	<ul style="list-style-type: none"> – Overtime is discouraged – Sustainable pace of work – Creativity and high quality work require downtime
Nature of time	Monitoring intervals	<ul style="list-style-type: none"> – The Development Team monitors itself on a daily basis – At the end of the Sprint, the stakeholders monitor the outcome – Status updates can be requested any time from the outside but are solely answered by the Product Owner – Estimates are not viewed as a contract – Deviations from plan are normal due to the complex nature of product development – Full transparency into the real status replaces the urge to control adherence to a predictive plan

(continued)

Schein’s category	What to find out	Finding from literature review
Unknowable and uncontrollable	How is it dealt with	<ul style="list-style-type: none"> – The world is seen as unknowable and uncontrollable – Certainty cannot be achieved – Meet uncertainty with determination and wit – Embrace change – Iterative, incremental process – Running experiments to find the right solution – Reduce complexity by working in small timeboxes – Acting instead of thinking it over again and again
General	Missed artifacts	<ul style="list-style-type: none"> – All artifacts are kept visible on the wall or table – Product Backlog – Sprint Backlog – Product increment – Burndown charts – Anything the team deems helpful, e.g. – Lava lamps to show the built success
General	Missed ideas	<ul style="list-style-type: none"> – Generalists instead of specialists – An agile organizational structure resembles a networked or permanent project organization

23.4.1 Summary of Survey Findings

After reviewing the survey responses, the findings can be mapped on to the initially identified categories of Schein as follows:

Summary of Survey findings

Schein’s category	What to find out	Findings from survey
Common language and concepts	Technical jargon	<ul style="list-style-type: none"> – Team related jargon (Development Team) – Empirical process control jargon (Inspect and adapt, Agile) – Product related jargon (Definition of Done, Done, Product Increment) – Planning related jargon (Burndown, Teamboard/Taskboard, Release, Definition of Ready, Grooming/Estimation Meeting, Sprint Goal, User Story, Epic, Task,

(continued)

Schein’s category	What to find out	Findings from survey
		acceptance criteria, Story Points) – Business jargon (Stakeholder, Business Value, customer) – IT jargon (Continuous Integration, Continuous Delivery, Test Driven Development) – Leadership jargon (Product Owner, Scrum Master, impediment) – Scrum jargon (Sprint, Retrospective, Sprint Planning, Daily Scrum, Sprint Review, Product Backlog)
Common language and concepts	Emotional jargon	– Pigs and chickens – “If you find some horse meat in your beef, blame the processing company. But, if it’s a mix of pigs and chicken, call the Scrum Master.” – Dilbert cartoons – “How do you catch a pack of wolves? First the first one, then the rest!” – “There exist only 10 types of people—those who Scrum and those who don’t—it’s a binary thing.” – “It has always been like that!” – “I am the biggest fan of Scrum, but please adhere to the gate process.” – “TEAM—great, somebody else does it!” – “Reporting to the Scrum Master” – “I just completed planning the whole project” – “Says the CFO: ‘Here is my plan for Q1 through Q4’” – “Who laughs last has the highest ping” – “I’ll go ahead and get you a Gantt chart for that.” – “Chuck Norris is allowed to extend timeboxes” – “Am I today Scrum Master, Developer, or Manager?” – “50 % team member” – Squirrel burger
Group boundaries and identity	Dress norms	– Casual dressing – Common sense used (clean clothes, appropriately dressed when having customer contact) – Antipathy towards suits
Group boundaries and identity	Badges, uniforms, symbols or privileges	– No status symbols – Team decides on symbols and gamification – Behavior of colleagues shows status

(continued)

Schein’s category	What to find out	Findings from survey
Group boundaries and identity	Insider and outsider	<ul style="list-style-type: none"> – Insider: Everybody who wants to be an insider; there are no outsiders – Strong undercurrent showing that people who are not in the Scrum Team might be considered outsiders.
Nature of authority and relationships	Formal or informal relationship between people	<ul style="list-style-type: none"> – Informal relationship – First name basis
Nature of authority and relationships	Formal or informal relationship with bosses	<ul style="list-style-type: none"> – Informal relationship – First name basis
Nature of authority and relationships	Pecking order in meetings	<ul style="list-style-type: none"> – Focused, goal oriented discussions – Open communication – Respectful behavior – Everyone participates – Constructive and collaborative behavior – People are engaged – Sometimes heated discussions about the best solutions
Nature of authority and relationships	Source of authority	<ul style="list-style-type: none"> – All opinions weigh equal – The person who is best or most experienced at that topic has more weight – The best thought out idea wins – Consensus is not always paramount
Nature of authority and relationships	Openly voiced criticism (peers)	<ul style="list-style-type: none"> – Openness is wanted – Voicing criticism is encouraged
Nature of authority and relationships	Openly voiced criticism (boss)	<ul style="list-style-type: none"> – Openness is wanted – Voicing criticism is encouraged – It is OK to disagree with one’s boss in front of others
Allocation of rewards and status	How to gain power	<ul style="list-style-type: none"> – There is no traditional promotion (titles, etc.) – Promotion means personal development in some way – The topic is not commonly and equally understood
Allocation of rewards and status	What is rewarded	<ul style="list-style-type: none"> – Teamwork – Openness – Improvement – Supportive behavior
Allocation of rewards and status	What is punished	<ul style="list-style-type: none"> – Uncooperative, competitive, and antagonistic behavior – Acting against Scrum principles – Lone-wolfing/heroes
Allocation of rewards and status	Reward mechanisms	<ul style="list-style-type: none"> – Immediate feedback

(continued)

Schein's category	What to find out	Findings from survey
The nature of human nature	Are people intrinsically or extrinsically motivated	– Intrinsic motivation/theory Y
The nature of human nature	Like people coming to work	– People look forward coming to work – Work feels very good and motivating
The nature of human relationships	Espoused values	– Openness – Trust – Teamwork – Respect – Transparency – Honesty – Courage – Commitment – Communication – Value/results/goal orientation – Collaboration
The nature of human relationships	Focus	– Customer (delight)/market – Product/working software – People/happiness – Results/delivery – Value – Continuous improvement – Return on Invest/Business Value – Quality
The nature of human relationships	Leadership style	– Servant leadership – Democratic aspects – Transformational elements (vision etc.)
Nature of Space	Office design	– Personalized workspace, including gadgets, walls, posters, etc. – Team is sitting together – Open area – Calm, meeting, and lounge areas – Friendly, comfortable, motivating, spacious, bright, creativity promoting, inspiring – Communicative, collaborative, chaotic – Tidy, clean, structured – Mobile, flexible environment
Nature of space	Communication amount in the environment	– Constant buzz/low background noise – Low noise – Occasionally loud/loud in meetings
Nature of time	Overtime encouragement	– Overtime is despised
Nature of time	Monitoring intervals	– There is none or rare monitoring by management – There is some sort of constant monitoring by the team itself

(continued)

Schein's category	What to find out	Findings from survey
Unknowable and uncontrollable	How is it dealt with	<ul style="list-style-type: none"> – Empirical approach/try things out and learn – It is faced and dealt with – It is accepted – It is embraced
General	Missed artifacts	<ul style="list-style-type: none"> – Scrum Teams keep their artifacts visible and shape their environment – Scrum-Board/Task Board – Burndown Charts – Product Backlog – Sprint Backlog – Product Increment – Impediment Backlog – Vision/Vision Board – Whiteboard and Flipcharts

23.5 Findings Comparison

After reviewing both the literature findings and the survey responses, the findings can be compared based on the basis of the categories initially identified as follows:

Findings comparison

Schein's category	What to find out	Findings from literature review	Findings from survey
Common language and concepts	Technical jargon	<ul style="list-style-type: none"> – Team related jargon (Scrum, self-managing, self-organizing, cross-functional, responsible, sustainable pace, purge people to the bench) – Empirical process control jargon (Complexity, transparency, inspection, adaptation, embracing change, good enough, the art of the possible, failing early/fast, iterations, iterative, incremental, sashimi) – Product related jargon (Potentially shippable, done, quality) – Planning related jargon (Burndown chart, sprint goal, timeboxing) 	<ul style="list-style-type: none"> – Team related jargon (Development Team) – Empirical process control jargon (Inspect and adapt, Agile) – Product related jargon (Definition of Done, Done, Product Increment) – Planning related jargon (Burndown, Teamboard/Taskboard, release, Definition of Ready, Grooming/Estimation Meeting, Sprint Goal, User Story, Epic, Task, acceptance criteria, Story Points) – Business jargon (Stakeholder, Business value, customer) – IT jargon (Continuous Integration,

(continued)

Schein’s category	What to find out	Findings from literature review	Findings from survey
		<ul style="list-style-type: none"> – Business jargon (Value Driven Development, Return on Invest, Total Cost of Ownership, Business Value, productivity) – IT jargon (Technical excellence, keeping the code well factored, simple design, automated testing, early detection of errors) – Leadership jargon (Servant leadership, Product Owner, Scrum Master, single wringable neck, impediments) – Predictive process modeling jargon (adversaries) (Waterfall, predictive, defined, command-and-control) 	<p>Continuous Delivery, Test Driven Development)</p> <ul style="list-style-type: none"> – Leadership jargon (Product Owner, Scrum Master, impediment) – Scrum jargon (Sprint, Retrospective, Sprint Planning, Daily Scrum, Sprint Review, Product Backlog)
Common language and concepts	Emotional jargon	<ul style="list-style-type: none"> – Pigs and chickens – Single wringable neck 	<ul style="list-style-type: none"> – Pigs and chickens – “If you find some horse meat in your beef, blame the processing company. But, if it’s a mix of pigs and chicken, call the Scrum Master.” – Dilbert cartoons – “How do you catch a pack of wolves? First the first one, then the rest!” – “There exist only 10 types of people—those who Scrum and those who don’t—it’s a binary thing.” – “It has always been like that!” – “I am the biggest fan of Scrum, but please adhere to the gate process.” – “TEAM—great, somebody else does it!” – “Reporting to the Scrum Master” – “I just completed planning the whole project”

(continued)

Schein's category	What to find out	Findings from literature review	Findings from survey
			<ul style="list-style-type: none"> - "Says the CFO: 'Here is my plan for Q1 through Q4'" - "Who laughs last has the highest ping" - "I'll go ahead and get you a Gantt chart for that." - "Chuck Norris is allowed to extend timeboxes" - "Am I today Scrum Master, Developer, or Manager?" - "50 % team member" - Squirrel burger
Group boundaries and identity	Dress norms	- None mentioned	<ul style="list-style-type: none"> - Casual dressing - Common sense used (clean clothes, appropriately dressed when having customer contact) - Antipathy towards suits
Group boundaries and identity	Badges, uniforms, symbols or privileges	<ul style="list-style-type: none"> - No titles other than developer - No uniforms or badges - No status symbols - Being on a good team is a privilege 	<ul style="list-style-type: none"> - No status symbols - Team decides on symbols and gamification - Behavior of colleagues shows status
Group boundaries and identity	Insider and outsider	<ul style="list-style-type: none"> - Insider: Scrum Team and people who think "agile" - Outsider: People with old "waterfall" thinking or striving for absolute certainty 	<ul style="list-style-type: none"> - Insider: Everybody who wants to be an insider; there are no outsiders - Strong undercurrent showing that people who are not in the Scrum Team might be considered outsiders
Nature of authority and relationships	Formal or informal relationship between people	- Informal relationship	<ul style="list-style-type: none"> - Informal relationship - First name basis
Nature of authority and relationships	Formal or informal relationship with bosses	- Informal relationship	<ul style="list-style-type: none"> - Informal relationship - First name basis
Nature of authority and relationships	Pecking order in meetings	<ul style="list-style-type: none"> - Collaborative team approach - Creative team discussions - Facilitation of discussions 	<ul style="list-style-type: none"> - Focused, goal oriented discussions - Open communication - Respectful behavior - Everyone participates - Constructive and

(continued)

Schein's category	What to find out	Findings from literature review	Findings from survey
		<ul style="list-style-type: none"> – Decisions are collectively determined 	<ul style="list-style-type: none"> collaborative behavior – People are engaged – Sometimes heated discussions about the best solutions
Nature of authority and relationships	Source of authority	<ul style="list-style-type: none"> – All opinions weigh equal – Product Owner has the last say on the requirements – Scrum Master has the last say on the Scrum process 	<ul style="list-style-type: none"> – All opinions weigh equal – The person who is best or most experienced at that topic has more weight – The best thought out idea wins Consensus is not always paramount
Nature of authority and relationships	Openly voiced criticism (peers)	<ul style="list-style-type: none"> – Openness and honesty are wanted, even if it disappoints somebody – Voicing criticism is encouraged 	<ul style="list-style-type: none"> – Openness is wanted – Voicing criticism is encouraged
Nature of authority and relationships	Openly voiced criticism (boss)	<ul style="list-style-type: none"> – Openness and honesty are wanted, even if it disappoints somebody – Voicing criticism is encouraged – Line management: The standard literature doesn't say anything about it. Specialized literature hints to frequent mutual feedback though 	<ul style="list-style-type: none"> – Openness is wanted – Voicing criticism is encouraged – It is OK to disagree with one's boss in front of others
Allocation of rewards and status	How to gain power	<ul style="list-style-type: none"> – There is no promotion (titles) – There is no ladder to climb/flat hierarchies – Change of role is possible – Reward by greater salary – Reward by greater responsibility, e.g. more important projects 	<ul style="list-style-type: none"> – There is no traditional promotion (titles, etc.) – Promotion means personal development in some way – The topic is not commonly and equally understood
Allocation of rewards and status	What is rewarded	<ul style="list-style-type: none"> – Teamwork – Active, engaging participation – Helpful behavior – Getting the job done 	<ul style="list-style-type: none"> – Teamwork – Openness – Improvement – Supportive behavior
Allocation of rewards and status	What is punished	<ul style="list-style-type: none"> – Behavior that degrades performance or productivity of the team – Violation of the Scrum rules 	<ul style="list-style-type: none"> – Uncooperative, competitive, and antagonistic behavior – Acting against Scrum principles – Lone-wolfing/heroes

(continued)

Schein's category	What to find out	Findings from literature review	Findings from survey
Allocation of rewards and status	Reward mechanisms	<ul style="list-style-type: none"> – Immediate communication by team members – Rule violations are communicated by Scrum Master – Exclusion from meetings or team can follow 	<ul style="list-style-type: none"> – Immediate feedback
The nature of human nature	Are people intrinsically or extrinsically motivated	<ul style="list-style-type: none"> – Intrinsic motivation/ theory Y 	<ul style="list-style-type: none"> – Intrinsic motivation/ theory Y
The nature of human nature	Like people coming to work	<ul style="list-style-type: none"> – People look forward coming to work 	<ul style="list-style-type: none"> – People look forward coming to work – Work feels very good and motivating
The nature of human relationships	Espoused values	<ul style="list-style-type: none"> – Named values: commitment, focus, openness, respect, courage – Additional values: honesty, visibility/ transparency, emergence, inspection, adaptation, embrace change, the art of the possible, teamwork, continuous improvement, learning, self-organization, empowerment, collaboration, face-to-face communication, involvement, act – Values of the Agile Manifesto – Values from a personnel perspective: Employee orientation, pragmatism, collaboration in partnership 	<ul style="list-style-type: none"> – Openness – Trust – Teamwork – Respect – Transparency – Honesty – Courage – Commitment – Communication – Value/results/goal orientation – Collaboration
The nature of human relationships	Focus	<ul style="list-style-type: none"> – Continuous improvement – Teamwork – Product/product increment – People – Value – Planning (agile) – Customer – Market – Accomplishment rather than presence 	<ul style="list-style-type: none"> – Customer (delight)/ market – Product/working software – People/happiness – Results/delivery – Value – Continuous improvement – Return on invest/ business value – Quality

(continued)

Schein's category	What to find out	Findings from literature review	Findings from survey
The nature of human relationships	Leadership style	<ul style="list-style-type: none"> – Decentralized decision-making at the front – Servant-leadership – Self-managing teams – Empowerment – Facilitation – Mentoring and coaching – Like a parent: Growing people so that they are mature and self-managing – Line management cares for the development of each individual employee – Frequent mutual feedback loops (e.g. weekly) with employees should be established – Active living of ones responsibility – Management focuses on strategy instead of micro-management 	<ul style="list-style-type: none"> – Servant leadership – Democratic aspects – Transformational elements (vision etc.)
Nature of space	Office design	<ul style="list-style-type: none"> – Team rooms – Collocated space – Offshore development only with cross-functional teams in each location – Caves-and-commons: Open team room plus silent areas and meeting rooms – Maximum number of 20 people per room, including Scrum Master and Product Owner – Movable desks – Teams arrange their furniture themselves, multiple times a day if needed – Walls are plastered with information – Whiteboards and charts are everywhere 	<ul style="list-style-type: none"> – Personalized workspace, including gadgets, walls, posters, etc. – Team is sitting together – Open area – Calm, meeting, and lounge areas – Friendly, comfortable, motivating, spacious, bright, creativity promoting, inspiring – Communicative, collaborative, chaotic – Tidy, clean, structured – Mobile, flexible environment
Nature of space	Communication amount in the environment	<ul style="list-style-type: none"> – Communication amount is high – Nothing has been found about the noise level 	<ul style="list-style-type: none"> – Constant buzz/low background noise – Low noise – Occasionally loud/loud in meetings

(continued)

Schein's category	What to find out	Findings from literature review	Findings from survey
Nature of time	Overtime encouragement	<ul style="list-style-type: none"> – Overtime is discouraged – Sustainable pace of work – Creativity and high quality work require downtime 	<ul style="list-style-type: none"> – Overtime is despised
Nature of time	Monitoring intervals	<ul style="list-style-type: none"> – The development team monitors itself on a daily basis – At the end of the Sprint, the stakeholders monitor the outcome – Status updates can be requested any time from the outside but are solely answered by the Product Owner – Estimates are not viewed as a contract – Deviations from plan are normal due to the complex nature of product development – Full transparency into the real status replaces the urge to control adherence to a predictive plan 	<ul style="list-style-type: none"> – There is none or rare monitoring by management – There is some sort of constant monitoring by the team itself
Unknowable and uncontrollable	How is it dealt with	<ul style="list-style-type: none"> – The world is seen as unknowable and uncontrollable – Certainty cannot be achieved – Meet uncertainty with determination and wit – Embrace change – Iterative, incremental process – Running experiments to find the right solution – Reduce complexity by working in small timeboxes – Acting instead of thinking it over again and again 	<ul style="list-style-type: none"> – Empirical approach/try things out and learn – It is faced and dealt with – It is accepted – It is embraced
General	Missed artifacts	<ul style="list-style-type: none"> – All artifacts are kept visible on the wall or table – Product backlog – Sprint backlog – Product increment – Burndown charts – Anything the team deems helpful, e.g. 	<ul style="list-style-type: none"> – Scrum teams keep their artifacts visible and shape their environment – Scrum-board/task board – Burndown charts – Product backlog – Sprint backlog – Product increment

(continued)

Schein's category	What to find out	Findings from literature review	Findings from survey
		– Lava lamps to show the built success	– Impediment backlog – Vision/Vision Board – Whiteboard and Flipcharts
General	Missed ideas	– Generalists instead of specialists – An agile organizational structure resembles a networked or permanent project organization	

References

- Cameron, K. S., & Quinn, R. E. (2011). *Diagnosing and changing organizational culture* (3rd ed.). Hoboken, NJ: Wiley.
- Schein, E. H. (2009). *The corporate culture survival guide* (New and Revised ed.). San Francisco: Jossey-Bass.



Scrum is a very simple framework, consisting of just a few meetings, artifacts, roles, and some rules that bind them together. The Scrum Guide is the official definition of Scrum (cf. Schwaber and Sutherland 2016) and contains only 16 pages. This section gives a brief overview and stresses the most important aspects. For a more in-depth view into Scrum, other professional literature is recommended (e.g. Verheyen 2013). One important thing to know is that Scrum views itself merely as a framework, existing in the context of an organizational environment, and being filled with meaning by other methods that complement it (for some examples, see Fig. 24.1). No project—or product development endeavor—will be successful by using only Scrum—complimentary practices are always needed.

24.1 Scrum Roles

Scrum only recognizes three roles: Scrum Master, Product Owner, and Development Team. Together, they form the Scrum Team. The customer is often mentioned but is not defined in Scrum. Management is only mentioned on the outskirts of Scrum and—depending on the author—sometimes even said to be unimportant or not necessary. As described above, management is indispensable, even though its roles and definition might have to change. Please be aware that every project succeeds or fails through the people working on it. This is of course true for Scrum as well. As everywhere else in any company, each position has to be filled with qualified people. You wouldn't want a car mechanic to do a heart transplant, right?

Therefore, the following sections do not only describe the mandatory Scrum roles (Product Owner, Development Team, Scrum Master), but also other roles I observed to be important in the context of Scrum and especially its introduction (management, change manager, Scrum consultant).

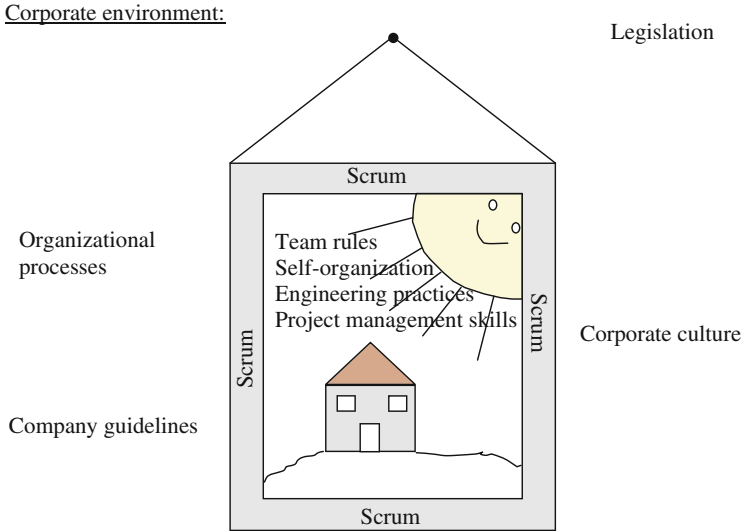


Fig. 24.1 Scrum viewed as a framework

24.1.1 Product Owner

The Product Owner is the most important role in Scrum.

A single team member who is not productive will be balanced by the other team members. An incompetent Scrum Master can be compensated for by his team, management, or a Scrum coach. Without a capable Product Owner, the project will certainly fail. The Product Owner owns the product. He is solely responsible for making all decisions that affect the future of the product. His tasks overlap significantly with those of a product manager:

- Competitor and market analysis
- Product improvements and maintenance
- Identify requirements and make sure they are well formulated
- Create concepts for market introduction
- Introduce the product into the market
- Support and educate sales
- Customer care
- Product strategy development
- Planning, coordinating, and implementing strategic measures
- Feasibility studies
- Estimation of expected sales volume
- Creation of the product roadmap
- Shaping of the introduction and product lifecycle processes
- Product retirement
- Optimizing return on investment (ROI)

- Reduction of the total cost of ownership (TCO)
- Optimizing customer value

So a Product Owner accompanies the product throughout its whole lifecycle in the same way a product manager would. This is one reason why Scrum often incurs organizational change: A product lifecycle does not start and end at product development, but touches almost every part of the organization. If supportive processes like personnel and purchasing (because these are important for employee retention and productivity) are added to the mix, almost no process is left untouched by Scrum.

The optimal Product Owner is usually the single person knowing the requirements from the customers' perspective the best. This usually is not a technical expert, so not a senior developer or architect. He might join in parallel during the actual development, but there is a strong likelihood that this will impede his ability to properly fill the Product Owner role. The tasks mentioned above can fully load even the best employees. It is important not to install a so-called Proxy Product Owner. This term describes a Product Owner who is working a lot with the development team, is available for requests, and takes over many of the tedious tasks like spelling out requirements. However, this person does not have any decision making power over those very requirements. Such constructs lead to information loss since the Proxy has a different understanding of any specific detail to that of the decision maker. In turn the development team interprets that information differently as well. At the end of the day, this leads to results that were never wanted by the decision maker, Product Owner, let alone the customer. Also, if the Product Owner only sees the results once the Sprint is over, he will most likely be disappointed and will have change requests. This procedure is inefficient.

Most certainly the product manager and Product Owner are both very busy and support is of course allowed. This could happen in the form of a technical analyst who is part of the Development Team. However, the Product Owner is never allowed to give up ownership of his product—if he does, power and accountability also have to be passed on. Without power and accountability, they are no longer Product Owners.

The Product Owner's strongest skills should be communication and the ability to motivate others. A product manager taking on the Product Owner role should be aware that he would have the opportunity, and maybe the obligation, to release and deliver product regularly. This is new for some since not everybody is used to think in terms of product absorption¹ of his product.

¹ Product absorption is the ability and will of a customer to install and use a new product version. This means the customer has to recognize a benefit for himself and the product installation has to work smoothly with low implementation costs.

24.1.2 Scrum Master

As the Product Owner manages the products, the Scrum Master manages the process. Her remit is contained to the Scrum process.² This power means that the Scrum Master can theoretically force the other process participants to adhere to the Scrum rules. In practice, she often lacks the formal authority to do so and might, on rare occasions, need management support. Usually, she is respected due to her skill level, personality, proven successful track record, and process participants listen to her, ignoring formal authority. This sometimes can create conflict since it puts in question the existing formal enterprise hierarchies.

The Scrum Master's tasks are:

- Increasing productivity of the Scrum Team by removing impediments
- Making sure the Scrum process is understood and adhered to
- Making problems (called “impediments”) transparent and making sure they are being solved
- Training project participants in Scrum
- Focusing project participants on the project goals
- Facilitating events
- Communicating with anyone interfacing with Scrum

The impediments usually build on one another and follow some sort of lifecycle. First, a team has to learn about Scrum and looks for the right methods to properly fill the framework. Often at the same time, a new team espouses team conflicts because the team members go through stages of forming and storming. So the Scrum Master has to teach Scrum, help the team find the right methods to apply, and work with the team through their conflicts. Once this is successful, the impediments normally shift towards processes that are directly connected to what the team already does. This often includes requirements management, quality assurance, and deployment. At this point, the team can already work at a high speed and will be quite satisfied. There are more impediments though, coming from processes not directly connected to development. These might be from personnel, sales, purchasing, management or other company processes. If they hinder the team, the Scrum Master has to help remove them. By this stage in the project lifecycle, the Scrum Master has effectively become an organizational change agent, influencing the whole company. Of course, a team might at any time experience issues from other stages in the lifecycle, but usually is able to resolve them without much help from the Scrum Master if they successfully have gone through it together already.

² Scrum is a product development framework. Therefore, the term “process” is not a perfect match. For simplicity, this term is used and includes roles, artifacts and events, as described in the Scrum Guide.

The Scrum Master does not have any technical or domain work to do. That means she needs neither a developer nor a product management background. She needs soft skills. These include the following³:

- Communication skills
- Conflict resolution skills
- Negotiation skills/diplomacy
- Ability to build and sustain a good network throughout the company
- Openness
- Courage
- Moderation/facilitation skills
- Motivational skills
- Expert Scrum knowledge
- Basic domain knowledge and being able to speak the development language
- Good time management
- Discipline
- Reflection skills (for herself and others)
- Ability to take criticism
- Process affinity
- Knowledge of team dynamics
- Knowledge of how the surroundings influence motivation and productivity
- Psychological basics
- Sociological basics
- Desire and ability for constant learning and improvement

It is helpful if the Scrum Master has been through similar projects before. This helps her understand connections and problems more quickly. However, it is far more important to possess the skills mentioned above. IT knowledge can be learned, some soft skills cannot. Scrum Master is a leadership position without disciplinary power. Do not underestimate it.

A Scrum Master is—just like a Product Owner—a leader. So if you are looking for a new Scrum Master, you should start evaluating your managers for this role. Check the skills of the candidates very critically: Whoever ascended through the “Peter Principle”,⁴ may not have the necessary traits. It is also important that the Scrum Master neither had disciplinary power over the Development Team in the past, nor has it now in the present. People have a hard time trusting somebody who, in the case of a conflict has the power to reprimand them. Without the team’s confidence, a Scrum Master cannot fulfill her duties and another solution for the placement needs to be sought.

³ This list was created by some of my training participants and includes both traits and skills.

⁴ cf. Peter and Hull (2011): In a hierarchy every employee tends to rise to his level of incompetence. In other words: If one is good at his job, he will be promoted to the next higher position. Eventually, the skilled worker becomes a manager. If he is no longer suitable for his position and does a bad job, he will no longer get promoted, but instead remains at his post.

24.1.3 Development Team

The Development Team is responsible for transforming the product requirements into a finished (or “done”) product. This happens iteratively and incrementally, so the team has to deliver something finished every couple of weeks. It is not able to engage in lengthy analysis or planning tasks for months on end. The team is self-organizing and usually solves impediments themselves. They are truly empowered and do not shift responsibility elsewhere. Should they fail, the Scrum Master helps. The resulting self-confidence helps them to select the right amount of work for each Sprint and to say “no” when no more work can be accommodated.

Your specific situation determines which competencies to introduce into the Development Team. Both domain expertise and personal aspects are important. It is always a good idea to involve the Product Owner, Scrum Master, and existing team members in the selection process for new developers. The term “developer” describes everybody in the team who helps develop the product. This usually includes programmers, testers, documentation experts, analysts, UI designers, etc. In many teams a healthy mix includes the same number of programmers as testers.

The problem of selecting new team members is eased if the developers have been working together for some time in a Scrum Team. In this case, usually a culture of openness has been established, in which the team members trust each other enough to also express dissatisfaction with one another. Here the Development Team is able to decide who they want to have on their team and who they would prefer not to include. It is beyond the scope of this book to describe the composition of a team in detail. However, I would like to share two tips to give you a hand if you do not know where to start.

Tip 1: The basic software team composition

If you do not know what technical skills you need to begin with, then you can start with the following rule of thumb: Include as many testers as programmers in the team, supplemented by an architect. Start with an experienced architect, two programmers and two testers. Use the Retrospectives to identify further staffing needs.

Tip 2: The selection method (cf. DeMarco and Lister 2013)

When you invite potential new employees for an interview, let the candidates prepare a 5-min presentation of their current work. Let the candidates present this in front of their potential future colleagues and encourage discussion with the candidate. Once the applicant has left, discuss the results with the team and seriously consider their impressions. This helps to better assess the candidate’s skills and ensures that—if you hire her—she is integrated more quickly into the Development Team than she would have been without the team involvement.

24.1.4 Management

Management is a role often underestimated in Scrum introductions. In the past some advisers even stated that no managers were needed at all. Please do not make

this mistake yourself! It is the core task of management to manage the organization, to advance it, and to ensure the company's success. This of course also includes shaping the business processes. One has to be naive to believe these processes could be changed without management support. The management team can be your greatest ally when it comes to solving problems and to shape the environment.

Managers are usually intelligent but very busy people. You should support them by preparing decisions well, by only conducting essential meetings (always with clear objectives and a well thought out agenda), and by explicitly highlighting the benefits of your endeavor for the organization as well as at an individual level. Do your homework and involve the management team. Always take objections seriously and question their origins—you might have missed an important point. At the end of the day you're all in the same boat. It might therefore help to all row in the same direction.

24.1.5 You as Change Manager

As a change manager, your job is to change the organization to increase its competitiveness. Be conscious of personal biases and preferences as these need to be eliminated or circumvented. Ensure that the organization recognizes its need for change and implements all the necessary steps. Help the company to stay on track. Such changes always bring stifling conflicts, which are often based on individual fears or change fatigue. It is also part of your job to identify and resolve such conflicts.

You want to introduce Scrum, because you see this as an opportunity for your organization. To introduce Scrum, ensure its long-term success, and enjoy its benefits however, you very likely have to change the organization itself. You will be Scrum Master, psychologist, kinder-garden teacher, midwife, and change manager all in one. Since you brought up the idea of agile, people will approach you with questions about Scrum. They will regard you as Scrum expert, whether you want them to or not. Accordingly, you should have solid knowledge in the area of agile methods. Alternatively, you can introduce an agile coach into the organization and delegate appropriate requests—in any case you still need a certain fundamental knowledge. If you do not already have this know-how, you have to remedy this without delay.

As change manager you also need considerable knowledge about how change in organizations takes place and how people react to it. Tact and political skill will be your most needed skills. The professional respect of your colleagues should be secured thus guaranteeing you a certain “standing” within the enterprise. Only if people listen to you fully, is it possible for what you say to matter to them. Only if what you say is brought forward in a way that your listeners understand and recognize a benefit for themselves, will you be able to achieve the necessary impact. You have chosen the most difficult task in changing the company. This is an excellent opportunity—one that will inspire and provide you with a worthwhile challenge.

24.1.6 The Scrum Consultant

The Scrum consultant has two main functions during a Scrum introduction: On the one hand he has to transport the agile domain knowledge into the company, on the other hand he needs to reveal unpleasant truths. He is best positioned to do this, as he is only bound to the company for a very limited period of time. He is not a “prophet in his own land”; people listen to him and value his opinion. He is, after all, a proven expert, usually with a proven record, certificates, and an exorbitant hourly rate. If someone is so expensive, people often think twice about having lengthy discussions with this person. The organization will also try to get rid of this person and related bills as soon as possible. This can only be achieved, however, once the goal is reached, for which the consultant was hired. This in turn requires a certain level of acceptance of his proposals. These proposals may very well uncover the most unpleasant issues,⁵ because the consultant does not have to fear for his job. He is after all just “passing through”. In addition, he does not know the organization that well—so people often forgive a very direct approach in his questioning. Internal employees would not get away with such openness in some corporate cultures—especially in large companies.

You should be very careful when choosing your advisor. As in any profession, there are different specializations and skills available. It may well be that in the course of your Scrum introduction you need different consultants for different tasks. Make sure that you find a very capable person to assist you in the initial stages to gain alignment with the long term vision. This person should be an expert in organizational development and Scrum—a “normal” Scrum consultant will not help you much. Do not try to save money in the wrong place. Also consider that introducing profound Scrum may take up to 10 years. We are not talking about a 2-week engagement of an expert who disappears after 2 weeks. At the start you need to plan an engagement that covers several months, followed by monthly visits. In the concluding phase, from around year three you should plan workshops in longer intervals. This list does not include other consultants you might need to improve development practices (Test-Driven Development (TDD), Pair Programming, Continuous Integration (CI) etc.) or to advance other special topics such as career path planning or agile budgeting approaches. These deployments are required only selectively and are less important since they can be corrected on an ongoing basis.

How to find “your” consultant? This is an extremely difficult question. Scrum consultants are as numerous as pebbles on the beach. Many after gaining first experience as Scrum Master, feel obliged to switch to a consultant role. Some of them actually are pretty good. Some are not. Start with thorough research. Both Scrum.org and scrumalliance.org have publicly available online lists of their

⁵ Imagine for a moment the consultant would reveal that in the past the decisions of the product managers were driven by individual annual bonuses rather than by the company’s goals. This would be most unpleasant for the management team, since they built these bonus schemes into the contracts, did they not?

trainers/coaches. Consider it a starting point. Go through the announcements of agile conferences and take a look at the listed speakers. Better yet, listen to the presentations and assess their quality yourself. Look for literature on the subject. Ask for recommendations. If you have a few names, run them through a search engine. You will find a lot on the well-known people and only very little on the “normal” consultants. Go through the results carefully and decide on their relevance. Take notes for later comparison. Make sure that you find more than just training advertisements: Anyone who trains others every week will have little time to be available with you onsite. Once you finally have your shortlist, write to or speak with the individual consultants. Ask a few well-prepared questions about the topic and document their answers. Finally, you should invite your favorite(s) for a personal interview. While this statement is unpopular, the “gut feeling” still has to fit. It is particularly helpful for the selection process to confront the consultant with a group of “skeptics”. Role-play it, if you do not have true skeptics. Either way, use this to convince yourself of the consultants’ Scrum introduction approaches. Pay close attention to the concept: Do you recognize the steps described in this book (possibly with different names), or does the concept only talk about the Scrum roles? Make your decision carefully and do not rush yourself. A mistake at this point can lead to the failure of your organizational development endeavor.

24.1.7 Approach to Fill the Roles

With the exception of the Scrum consultant you should, wherever possible, rely on internal personnel. They already know the company and the product. In particular with a Product Owner assignment, it is especially hard for a new employee to immediately be able to make all product related decisions. The transition to the new way of working is already difficult enough. Do not make it even harder by confronting her with a new company, new colleagues, as well as new products. You should also refrain from rotating roles.⁶ In a hospital, the heart surgeon does not change places with the immunologist. At least if they did nobody would expect the same performance level. When assigning the roles you are of course limited by the availability of your staff. Apart from this, however, you should focus on the factors “motivation” and “skill”—in this order. If an employee is highly motivated and wants to work with Scrum, he will be open to change and willing to overcome obstacles. In the beginning of a Scrum introduction this is even more important than perfect professional skills. These are of course also important, but even the best horse in your stable will not win a race if it does not want to Sprint. You can build up specific hard skills, such as technical and methodological knowledge over time,

⁶There are teams that for example rotate the Scrum Master role between team members or who appoint a different Product Owner every release. This is usually a huge mistake. Only very experienced and already successful teams can use such approaches in certain situations to increase motivation or solve a specific problem.

through training. Wherever you have no leeway because of corporate management decisions or due to the conditions in the company, you should look for and promote individuals based on their motivation towards the change. Forge a team!

24.2 Artifacts

Scrum only knows three artifacts: The Product Backlog, the Sprint Backlog, and the Product Increment. All artifacts are mandatory in Scrum. Their goal is to reduce the amount of documentation—and the associated effort—to the appropriate level. In addition, they help optimize the process through constant inspection and adaptation. All artifacts have to be transparent at all times. This means that they must be easily accessible (i.e. not hidden in some “information freezer” within a knowledge database where nobody ever looks—and by the time it is reviewed, the information is either outdated or the password to access it is no longer active. How much information in your freezer has passed its use by date?). It also means that everybody who looks at it has a common understanding of the content. It is normal in Scrum to continuously work with all process participants, which automatically leads to an equal understanding. In addition, artifacts are usually presented openly in the team room so everybody can easily keep up to date.

24.2.1 Product Increment

Scrum demands that the Development Team delivers a “done” Product Increment at the end of every iteration. Done means the product could be released to the customer with no additional work. Beyond that threshold it is perfectly clear to everyone what work is missing, so if a release is not possible, any nasty surprises are avoided. It is not sufficient to ship something to the quality assurance department so “they can fix it”. The responsibility always stays with the Development Team. To create something potentially shippable is difficult if some skills are missing in the team—for example testers. This is one reason why Scrum demands cross-functional teams with all the skills necessary to complete the Increment. Something like a “testing team” or a “concept team” simply does not exist. Make sure that every (!) Sprint a finished product is created.

24.2.2 Product Backlog

The Product Backlog contains the sum of all product requirements that need to be implemented by the Development Team. The Product Owner is solely responsible for this backlog. Scrum does not prescribe a specific form for this artifact (e.g. User Stories) and each entry in this list is simply called a “Product Backlog Item”. The main goal is for the Product Owner to create a reminder for the Development Team so they do not forget what he wants them to do. Usually, a Product Backlog is not

set up to be self-explanatory. Instead, a continuous collaboration between Product Owner and Development Team makes sure this artifact is commonly understood and the right things are implemented. This is a deviation from traditional requirements management where specifications are written to save on talking time. The Product Backlog is always ordered so at all times, it is transparent what needs to be worked on next. The items, which have been chosen for the next one or two Sprints, are usually well understood and more detailed than the items planned in later Sprints. There are no other sources of work for the Development Team.

24.2.3 Sprint Backlog

The Sprint Backlog consists of the Product Backlog Items forecasted by the Development Team in the current Sprint. It also contains the breakdown of tasks⁷ to show how to turn these items into a “done” Product Increment. The Development Team creates it during the Sprint Planning meeting and has sole responsibility for it. People outside the team do not have a say in its creation or usage. The Sprint Backlog is a living artifact, potentially being refined, updated, and corrected every single day. It always shows the exact status of the Sprint. Since Scrum views each Sprint as a separate project, it can be compared to a project plan. If it doesn’t change, it is not a real Sprint Backlog. The quality and actuality of this artifact is often a direct reflection of the maturity of the Development Team. The information transported by this backlog is value neutral. That is, no matter at what pace the work gets done, nobody has to answer to this point. Only after any situation is analyzed is it possible to learn about the root cause, and whether the Sprint goal can still be reached or not. The Sprint Backlog may never be used as a tool to apply pressure on the Development Team.

24.2.4 Definition of Done

While the Definition of Done (DoD) is not a mandatory Scrum artifact, I so far have not met any highly productive team without one. The development organization can create the DoD. If it doesn’t do that, the Development Team is responsible for creating one. The Product Owner defines the outcome of quality, that means what requirements exist for product quality. The Development Team defines in their DoD the measures taken to achieve this level of quality. When the Definition of Done is reached, the Product Backlog Item is considered “done”. It is the most important tool the Scrum Team has to ensure quality. It can be improved every single Sprint, but it may never be worsened. So for example adding automated

⁷ A task is a piece of work that can be completed from one day to the next. Some teams choose to work without tasks, for different reasons. When you start with Scrum however, I strongly recommend you use tasks.

regression testing is allowed, while removing documentation after it has been in the DoD for at least one Sprint, is not. Therefore it makes sense to start small and grow slowly. All agreed upon measures in the Definition of Done have to be adhered to at all times. If just one single item is not fulfilled at the end of the Sprint, the Product Increment is considered “not done” and will be rejected by the Product Owner. Typical DoD items are:

- Complete all tasks with no work remaining
- Every task undergoes the four eyes principle
- All acceptance criteria of the Product Backlog Item must be fulfilled
- Verify the acceptance criteria through automated tests
- Completely integrate the code
- Make sure all tests, old and new, run through successfully
- Update the documentation
- Adhere to the coding guidelines
- Refactor the code

These items are just an example. Each DoD has to be specifically created for the demands of the product at hand. The Development Team is responsible for both content and adherence. The Product Owner in return makes sure the Development Team has enough time to adhere to it. This way, everybody talks about the same thing when they say “done”.

The only unforgivable mistake in Scrum is the so-called “ethical breach”. This occurs when the Development Team claims that they complied with the DoD, even though they know that they did not. Thus, they are consciously lying to the Product Owner. The Product Owner is usually not in the position to judge whether the statement is true due to a lack of technical knowledge and has to rely on the team’s statement. Through this ethical breach, trust is so fundamentally damaged that further cooperation with this Development Team usually does not make sense. The team has to be disbanded. In addition, ethical breaches always come to the surface sooner or later. It often takes some time, but in the end all sins come to light. For the Development Team this simply means that they have to openly and honestly tell the Product Owner at all times if they have not complied with the DoD. The worst thing that can happen to them is that the unfinished Product Backlog items have to be completed during the next Sprint. Honesty is the best policy.

24.3 Events

Officially, the Scrum meetings are called “events”. All the following described events are mandatory in Scrum and part of the framework. Their main goal is to reduce the total meeting time by structuring the Sprint in a way that makes other meetings obsolete. All events are timeboxed, meaning that they cannot take more than a maximum amount of time. When the time is up, the achieved outcome is inspected; either a new timebox is scheduled, the approach is changed, or the

outcome is considered good enough to continue working with it. Most timeboxes can end early though, as soon as the goal has been reached. The only exception is the Sprint itself, since switching its duration every now and then would increase complexity in terms of meeting logistics and velocity calculation.

24.3.1 The Sprint Itself

The Sprint is an iteration of 30 calendar days or less. At the end of each iteration, a “done” Product Increment has to be delivered. There is no “free” time between Sprints—one Sprint seamlessly follows the other. These iterations set the pace and rhythm in Scrum. They are like a heartbeat.

1. Delivery of a done Increment at the end of each Sprint potentially allows the Product Owner to release it straight away and generate an immediate return for the customer. The Scrum Team sets the exact duration. It is important for the Sprint duration to stay constant across multiple Sprints in order to gather a meaningful velocity.⁸ While the maximum duration is 4 weeks (or up to 30 calendar days), there is no minimum duration defined by Scrum. However, in practice the minimum duration is usually set to 1 week. The most common durations are 2 and 3 weeks. After all, most people cannot remember what they had for lunch last week, so they are most certain to have difficulty remembering what was planned 4 weeks ago in the Sprint Planning meeting.
2. My recommendation is for you to start with 2-week Sprints. If there are any serious problems, you should switch to 1-week Sprints. This gives you more opportunities to analyze the root causes of the problems and to take action more rapidly. If the team is well-tuned and wants to work in longer Sprints, use three-week iterations.
3. The main reason to choose a specific Sprint length is the speed of learning you want to achieve. The faster you need to learn, the shorter the Sprint length.
4. If Scrum has recently been introduced, there is often pressure from the Development Team and management to extend the Sprint duration beyond this 4 week limit. The argument goes along the lines of, “we can not get anything done in just 4 weeks”. Do not give in to this pressure! I know no single case in which the above assertion was based on a solid foundation. Instead, ask the following question: “When will you be home tonight?” You will instantly get answers. For example: “Around six o’clock.” Dig deeper then: “Not around six o’clock. When exactly? Be precise to the minute, please!” You may even get an answer. Then ask: “When exactly will be you at home Wednesday in 4 weeks?”

⁸ Velocity is a term representing the sum of all estimates of all Product Backlog items that were completed in one Sprint. Across multiple Sprints the average is calculated and used to forecast the future speed of the team.

5. The goal of this exercise is to show that we cannot precisely plan even relatively simple activities. How should we then be able to plan something as complex as software development, which is in addition happening in the (far) future? Then ask the final key question: “What do we have to do so that we are able to deliver a finished product together within a single Sprint?”
6. You will receive many suggestions from your team which you should review and refine together collaboratively. These findings should then, as appropriate, be included as commitments to continuously improve your process for future Sprints. At the end of the day you need to know what it takes to at least deliver an initial working solution in a Sprint, knowing that the overall requirements evolve over time. Use this approach to help the team understand how this different approach can help them achieve their development goals and ask them to align on an appropriate Sprint length.

24.3.2 Sprint Planning

The Sprint Planning meeting is the first thing that happens in the cycle and is the official start of a Sprint. It pursues two goals, which are often represented in two parts of this meeting. The first goal is to reach agreement between Product Owner and Development Team what is wanted in the next Sprint. This usually ends with a forecast by the Development Team, telling the Product Owner what to expect by the end of the current Sprint. The second goal is for the Development Team to plan this next Sprint, often by creating tasks of 8 h or less. The final outcome of this meeting is the Sprint Backlog that shows in a very fine granularity what the team is going to tackle and how they aim to achieve this.

Usually, only the Scrum Team takes part in this meeting. However, if somebody else can add value, they are of course welcome and will be invited. This could be true for domain experts who can explain particular requirements in very specific detail or development experts who are needed for just one Sprint (they will be guests in the Development Team in that case). People outside the Scrum Team who cannot contribute should not be part of the meeting.

24.3.3 Daily Scrum

The Daily Scrum is not a status meeting but a planning meeting. The Development Team plans how to meet the Sprint goal every day, taking into account the recent progress. Of course, the Scrum Master can support the team by facilitating this daily 15-min timebox. Everybody’s status is disclosed—not for the sake of knowing the status but to make the right decisions in changing the Sprint Backlog and thus the overall plan. The Daily Scrum is also a great opportunity to observe the Development Team motivation and self-organization. Guests are not allowed; usually not even the Product Owner participates since the discussions are focused at a technical level, which he does not understand. It does help however when he is available right

after the meeting to answer questions that arose. If the need for individual discussions is identified—with the Product Owner or another team member—this is usually deferred to a later point in time, often right after the Daily Scrum. The Daily Scrum is an instrument that greatly reduces project risk. If it was done less frequently, weekly for example, it would take far longer to identify deviations or that the team is running in the wrong direction. A single week usually incurs costs of more than 25,000 \$ for an average team. That's a lot of money which can be invested more effectively by creating transparency in just 15 min every day.

24.3.4 Sprint Review

The Sprint Review is open for anybody who is interested in the Sprint outcome. Its goal is to collaborate with everybody present on what to do next, to gather inspiration and identify the need for changes. It is a planning meeting, aimed more at the long-term plan. Ideally, customer attendees try the product out themselves and do not just watch somebody else doing it. Lengthy presentations are forbidden—it is not about slide shows. Also, the primary goal is not to formally accept the outcome of the Sprint. This can be done, but is very bad practice since this usually means that the Product Owner knows at the same point in time as his stakeholders and customers what was done and how well it has been achieved. This can lead to awkward situations for the Product Owner in front of his clients. Usually, the Product Owner inspects the outcome of Product Backlog Items as soon as they are completed during the Sprint. If done correctly, the Product Owner can save a lot of personal traveling time by properly facilitating this meeting to ensure customer and stakeholder attendance and participation.

24.3.5 Sprint Retrospective

Usually the Sprint Retrospective is the official end of a Sprint. During this meeting the process is inspected and measures to improve it are identified. There are many ways to conduct such a meeting (e.g. described in Derby and Larsen 2006). The important thing is that concrete measures are defined—just whining about the status quo is not enough. A few tangible tasks are better than a great amount of fuzzy ideas. Only the Scrum Team is present in this meeting, no guests are allowed. It is important to conduct the Retrospective every single Sprint and to have it run by an experienced Scrum Master. This helps the participants to focus and creates a steady pace of continuous improvement. If run well, this meeting can uncover great productivity improvement opportunities, which, if implemented, can lead to tangible gains. Scrum in itself does not raise productivity automatically. It just makes problems transparent and allows the organization to solve them. This in turn does indeed raise productivity. If the Sprint Retrospective is skipped, so are productivity improvements.

References

- DeMarco, T., & Lister, T. (2003). *Waltzing with bears: Managing risk on software projects* (Paperback ed.). New York: Dorset House.
- DeMarco, T., & Lister, T. (2013). *Peopleware: Productive projects and teams* (3rd ed.). Reading, MA: Addison-Wesley Professional.
- Derby, E., & Larsen, D. (2006). *Agile retrospectives: Making good teams great*. Raleigh: Pragmatic Bookshelf.
- Peter, L. J., & Hull, R. (2011). *The Peter principle: Why things always go wrong* (Reprint ed.). HarperBusiness.
- Verheyen, G. (2013). *Scrum – A pocket guide. A smart travel companion*. Zaltbommel: Van Haren.
- Schwaber, K. & Sutherland, J. (2016, July). *The scrum guide*. Scrum.org. 28.08.2017. <http://scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf#zoom=100>



The following methods are not prescribed by Scrum but complementary and need to be selected depending on your specific context and situation. Many teams find them valuable for their daily work and use them. Of course, there are many more methods available as well. I just wanted to share a small sample of the most common ones with you.

25.1 Planning Poker

Planning Poker¹ is a method to estimate relative sizes. Each team member receives a set of playing cards that have the numbers 1, 2, 3, 5, 8, 13, and 21 printed on them (Many card sets also include the values 0, 40, 100, and ?. The “?” stands for “I have a question, which prevents me from estimating this.”). Historic estimates are used as the reference against which new Product Backlog items are relatively estimated. If you are starting with this technique and nothing has been estimated so far, you can arbitrarily define the seemingly smallest element as a “3”. This then becomes your reference value against which you assess the other Product Backlog items. The Product Owner then presents the next Product Backlog item and the Development Team members can ask clarification questions. Finally, the item is estimated by each team member by choosing and holding a card face down. All cards are then turned and shown simultaneously. The individuals who have the highest and lowest estimates state the reasons to support their respective choices. The Scrum Master keeps discussions short while the Product Owner provides requirement clarifications. Once the opinions are exchanged, everyone makes a new decision and re-estimates. The process then repeats as per the first estimation round. If no consensus is reached after three rounds, as a rule the highest value is taken. In some cases (depending heavily on the team), the number appearing most often in the last

¹ Planning Poker[®] is a registered trademark of Mountain Goat Software, LLC.

round can also be chosen as the estimate for the Product Backlog item. This way, you avoid absolute time estimates (e.g. “three person days”), and instead have a set of relative estimates based on an overall size of the requirement from the Development Team’s perspective. You only know that a “13” is almost three times as much effort as a “5”. Only in conjunction with the specific team-velocity can these numbers result in an indication of how much time a “5” really takes. The great advantage of this method is that it is not only very quick, but also ensures that every element does not have to be re-estimated if the environment changes. These changes can take the form of people changing teams, the team learning new skills and operating more efficiently, new problems surfacing, old problems are suddenly solved, automation is implemented, etc. Instead, only the velocity changes because the relative size of the elements is usually not impacted by these environment changes. This avoids considerable waste by preventing re-estimation and can therefore be viewed as a “lean” method. The biggest advantage of Planning Poker, however, is the way in which discussion and communication can be kept focused. This leads to a clarity in overall team understanding which far outweighs the actual time savings.

Since Planning Poker is usually used to estimate requirements that are formulated as User Stories, the sizes are usually estimated in the unit of Story Points.

25.2 Planning Poker for Absolute Numbers

Planning Poker can also be used to estimate absolute values. If you do so, you no longer try to determine whether an item is greater or smaller than another, but you try to estimate in real time units. If you do so, the team decides on a unit of measure, e.g. person hours, and then uses the cards to estimate the actual number of hours they individually believe each requirement will take during implementation. This will only work well in cases where the estimate must only be valid for a short time period, for example when Sprint tasks or acute impediments are to be judged. Do not make the mistake of trying to estimate the Product Backlog items in absolute numbers—you will not succeed!

The reason why you use Planning Poker for absolute estimates lies in its synchronicity. The fact that all team members have to form an opinion and these opinions are revealed simultaneously, leads to everybody getting actively involved. So more reserved individuals get to share their thoughts and you get to discussions that otherwise would not have surfaced.

25.3 Estimation Meeting

Scrum demands that the imminent portion of your Product Backlog must always be estimated. Unfortunately Scrum does not say anything about how you should perform the estimation process. The most common method for this is the

so-called Estimation Meeting (in the most recent version of the Scrum Guide it is referred to as “refinement”). Ideally, the Product Owner invites everybody to the Estimation Meeting as part of the Sprint Planning meeting. This means the team knows when they need to participate and can plan accordingly. The entire Scrum Team and—if needed—specialists from other teams attend the Estimation Meeting. Customers and stakeholders do not participate. The timebox should not exceed 2 h, since concentration then starts to deteriorate leading to less accurate estimates. The Product Owner brings his backlog items that have not yet been estimated to the working meeting. He successively presents the requirements and answers questions of the Development Team. Technical questions are not answered, since these have to be solved by the developers. Once all questions have been answered, the items are relatively estimated, for example using a method such as Planning Poker. Of course, other methods are equally valid. The main objective of the Estimation Meeting is that all developers understand what is involved in each Product Backlog item. This facilitates communication and prevents misunderstandings. Through this clarity, estimation of the elements is greatly simplified and almost becomes a by-product. There is no restriction on the frequency of Estimation Meetings. The Product Owner has to decide whether he wants to withdraw his team from working on solutions in order to estimate the backlog, or whether his backlog is sufficiently prepared. He must be aware that the Scrum Master is allowed to and will stop the Sprint Planning meeting if the Product Owner shows up without a sufficiently estimated Product Backlog. In addition, forecasts of the possible product completion dates and associated costs are only possible with meaningful estimates.

25.4 Timebox

A timebox is a specified period of time, which may not be exceeded. Scrum uses such timeboxes everywhere: Every meeting and every Sprint is time-constrained. Even most of the individual operations during a meeting are strictly limited (I for example often set timeboxes during workshops, of say 3–5 min, depending on the activity). A timebox is never extended because this would jeopardize the discipline. Instead, new timeboxes can be negotiated.

25.5 Velocity

Velocity represents the speed of your team. It results from the sum of the estimates of the finished features of a Sprint and is averaged over several Sprints. An example: At the beginning of a Sprint, a team estimates that it will create the three features A, B, and C. These features have been estimated using Planning Poker. A was estimated as three, B as eight, and C as 21 Story Points. After the first Sprint, the team delivers Product Backlog items A and B fully, but they did not fully complete C. Unfinished work is not even proportionally included in the velocity. Therefore, the velocity sums up to 11 this Sprint. In the next Sprint, the team

manages to deliver the fully completed feature C, but nothing else. So in that Sprint the velocity is 21. The average velocity of the team is 16 $((11 + 21)/2 = 16)$.

Since every team estimates differently, you cannot compare estimates from different teams easily. In such cases, normalization is required. Such a normalization usually is only wished-for in big corporations and neither sustainable nor simple to come by. Therefore, I do not cover it in this book.

The velocity is used to create release and Sprint plans. If all desired features (or Product Backlog items) are estimated in Story Points you can determine which features will be available and with what probability at which time, by extrapolating the average, best, and worst velocities out over time. This approach is the basis of your strategic planning.

Usually you only count Product Backlog items towards the velocity that are both completely done and generate business value. Bugs and purely technical requirements without customer benefit do not count towards velocity. Meetings or organizational tasks do not count either.

Index

A

Agile Manifesto, 182, 206, 212, 287, 297
Artifacts, 9, 33, 115–117, 143, 177, 188, 201,
203, 207, 215, 227, 228, 262, 269, 270,
289, 293, 299, 301, 310–312

C

Cameron, K.S., 181, 183–189, 223, 224, 234,
249–252, 255, 256, 258
Case study of Scrum culture, 125–126,
133–177
Change initiative, 52, 68, 69, 83, 85, 111, 113,
114, 175, 243, 250
Change manager, 301, 307
Cohn, Mike, 3, 182, 183, 207–215, 241–243,
252
ADAPT, 241–243
Common problems, 98–107
Communication, 6, 9, 13, 15, 36, 53, 67–69, 72,
77, 81, 85, 87, 90–92, 95, 104, 106, 112,
118, 119, 121, 124, 125, 140, 147, 148,
153–156, 158, 165, 167, 201, 203, 214,
218, 227, 229, 247, 248, 251, 272, 275,
287, 288, 291, 292, 296, 297, 303, 305,
318, 319
Competing values framework, 5, 9, 194, 195
Complexity, 8, 34–36, 87, 88, 127, 183, 199,
208, 215, 252, 253, 289, 293, 299,
313
Cultural aspects, 201, 208, 216–219
Cultural change, 42, 173, 182

D

Deal, T.E., 181, 183, 189, 191, 192, 197, 198
Decision matrix, 198
Dress code, 202, 209, 226, 261, 269

F

Feedback, 7, 8, 43, 65, 68, 69, 74, 125, 151,
165, 177, 191, 192, 198, 215, 217, 219,
230, 244, 246, 273, 275, 287, 288, 291,
296–298

G

Gender, 223, 225, 234, 236, 257, 267, 282
Geographical distribution, 267

H

Harrison, Roger, 181, 183, 189, 190, 198

I

Impediment-in-depth-analysis, 49–50
Insider, 200, 202, 208, 210, 230, 264, 274, 286,
291, 295
See also Outsider
Introduction styles
bottom-up, 22, 30–32, 55
submarine, 29, 31, 32, 41, 42, 55
top-down, 5, 7, 29–30, 32, 55, 138

J

Jargon, 5, 48, 67, 200, 202, 208, 209, 226, 247,
261, 268, 269, 285, 286, 289, 290, 293,
294
Jokes, 5, 116, 202, 227–229, 262, 270
chicken and pig, 209, 210, 227, 228, 270,
286, 290, 294

K

Kennedy, A.A., 181, 183, 189, 191, 192, 197, 198

- Kotter, John, 24, 41, 43, 47, 48, 51, 52, 55, 56, 60, 61, 63, 64, 66, 67, 69, 71, 81, 83–85, 109, 111–118, 124, 133, 137, 241–252
- Kotter's eight steps
 anchor new approaches, 41, 115–119, 247
 communicating the change vision, 67–69, 247
 consolidate gains, 41, 111–114, 124
 empower your employees, 71–81, 128
 generate quick wins, 83–109
 guiding coalition, 41, 44, 55–61, 64–66, 68, 71, 84, 85, 111–113, 115, 118, 122–125, 128, 137, 140–149, 152, 153, 155–157, 162, 164, 168, 169, 171–176, 246, 247
 urgency, 55, 57, 58, 60
 vision and strategy, 56, 58–61, 64–66
- L**
- Large teams, 121–129
- Leadership, 5, 10, 14, 38, 63–64, 67, 69, 71, 74, 85, 109, 113, 138, 173, 194, 196, 201, 203, 209, 213, 218, 223–225, 230, 247–249, 251, 252, 259, 260, 265, 276, 279–281, 284, 286, 288, 290, 292, 294, 298, 305
- Limitations, 136, 233–235
- M**
- Martin, Joanne, 181, 183, 187–189, 200, 201
 differentiation perspective, 187, 188
 fragmentation perspective, 187, 188
 integration perspective, 187–189
- Monitoring interval, 201, 203, 214, 276, 288, 292, 299
- Motivation, 7, 30, 31, 57, 60, 77, 78, 83, 84, 90, 91, 98, 105, 106, 113, 137, 148, 182, 200, 212, 230, 276, 284, 287, 292, 297, 305, 309, 310, 314
- Multi-change, 43–44
- N**
- Noise level, 201, 203, 214, 227, 236, 262, 271, 283, 288, 298
- O**
- Organizational Culture Assessment Instrument (OCAI), 5, 194, 199, 201, 221, 223–226, 234–236, 249–252, 255, 256, 278–281
- Organizational culture/corporate culture, 9, 27, 119, 124–126, 182–184, 187–203, 223–226, 233, 250, 255, 256, 258, 261
- Organizational structure, 4, 10, 12, 15, 20, 26, 27, 121, 218, 219, 246, 289, 300
- Outsider, 5, 6, 8, 200, 202, 208, 210, 230, 264, 274, 286, 291, 295
See also Insider
- Overtime, 8, 63, 116, 201, 203, 214, 227, 262, 271, 288, 292, 299
- P**
- Pilot projects, 85–92, 95, 98, 100–108, 112, 113, 125, 128, 242, 244
 Sprint 0, 93
- Playbook, 243–245, 252
- Promotion, 7, 111–112, 114, 116, 119, 124, 169, 174, 200, 202, 211, 231, 236, 241, 243, 265, 276, 284, 287, 291, 296
- Punishments, 7, 79, 200, 230, 275
- Q**
- Quinn, R.E., 181, 183, 189, 194–199, 223, 224, 234, 249–252, 256, 258
- R**
- Rewards, 7, 38, 79, 199, 200, 202, 203, 212, 246, 248, 275, 287, 291, 296, 297
- S**
- Schein, E.H., 4, 115, 116, 118, 181, 183, 189, 199–203, 221, 226, 236, 255, 256, 261, 285–300
- Schneider, W.E., 4, 181–183, 189, 192–194, 197, 198
- Schwaber, Ken, 3, 19, 22, 35, 128, 182, 183, 205, 207–215, 243, 245, 252, 301
- Scrum culture, 3–15, 24–26, 38, 52, 73, 74, 184, 207, 223, 235–237, 282
- Scrum shapes, 25, 26, 138
 Façade Scrum, 23, 27, 31, 50
 Profound Scrum, 23–27, 31, 42, 55, 56, 59, 86, 138, 150, 308
 Scrum PRN, 19–20, 26, 31, 41, 55, 59, 85, 137, 138, 150, 157
 Scrum Software Studio, 20–23, 26, 27, 86, 137
 Sustainable Profound Scrum, 24, 25, 27, 56

Virtual Scrum Software Studio, 20, 21, 26
Scrum values, 118, 124, 213, 214, 218
Stakeholder management, 36
Status symbols, 171–173, 201, 209, 218, 229,
271, 283, 286, 290, 295
Suitability of Scrum, 38
Sutherland, Jeff, 3, 19, 22, 76, 105, 123, 128,
182, 205, 207–215, 243, 245, 252, 301

T

Theory X, 201

Theory Y, 201, 212, 287, 292, 297
Things you should remember, 52–53, 61, 66,
69, 81, 109, 114, 118–119, 128–129
Transformation, 85, 122, 245, 247, 248

V

Velocity extrapolation, 50–51

W

Working space, 6, 20, 203, 227, 262, 270