

BEST PRACTICE

SCRUM

A Smart Travel Companion

A Pocket Guide - 2nd edition

Gunther Verheyen

Scrum - A Pocket Guide
2nd edition

Other publications by Van Haren Publishing

Van Haren Publishing (VHP) specializes in titles on Best Practices, methods and standards within four domains:

- IT and IT Management
- Architecture (Enterprise and IT)
- Business Management and
- Project Management

Van Haren Publishing is also publishing on behalf of leading organizations and companies: ASLBiSL Foundation, BRMI, CA, Centre Henri Tudor, Gaming Works, IACCM, IAOP, IFDC, Innovation Value Institute, IPMA-NL, ITSq, NAF, KNVI, PMI-NL, PON, The Open Group, The SOX Institute.

Topics are (per domain):

IT and IT Management

ABC of ICT
ASL*
CATS CM*
CMMI*
COBIT*
e-CF
ISO/IEC 20000
ISO/IEC 27001/27002
ISPL
IT4IT*
IT-CMF™
IT Service CMM
ITIL*
MOF
MSF
SABSA
SAF
SIAM™
TRIM
VeriSM™

Enterprise Architecture

ArchiMate*
GEA*
Novius Architectuur
Methode
TOGAF*

Business Management

BABOK Guide*
BiSL* and BiSL* Next
BRMBOK™
BTF
EFQM
eSCM
IACCM
ISA-95
ISO 9000/9001
OPBOK
SixSigma
SOX
SqEME*

Project Management

A4-Projectmanagement
DSDM/Atern
ICB / NCB
ISO 21500
MINCE*
M_o_R*
MSP*
P3O*
PMBOK Guide*
Praxis*
PRINCE2*

For the latest information on VHP publications, visit our website: www.vanharen.net.

Scrum

A Pocket Guide 2nd edition

A Smart Travel Companion



Colophon

Title:	Scrum - A Pocket Guide - 2nd edition
Subtitle:	A Smart Travel Companion
Author:	Gunther Verheyen
Reviewers:	Ken Schwaber (Scrum co-creator, Scrum.org) David Starr (Agile Craftsman, Microsoft) Ralph Jocham (Agile Professional, effective agile.) Patricia M. Kong (Business Agility Enterprise Solutions, Scrum.org) Blake McMillan (Scrum Master – soulofscrum.com) Dominik Maximini (Agile Lead, NovaTec Consulting GmbH)
Publisher:	Van Haren Publishing, 's Hertogenbosch, www.vanharen.net
ISBN hard copy:	978 94 018 0375 5
ISBN eBook (pdf):	978 94 018 0376 2
ISBN ePUB:	978 94 018 0377 9
Edition:	Second edition, first impression, January 2019
Layout and typesetting:	Coco Bookmedia, Amersfoort - NL
Copyright:	Gunther Verheyen & Van Haren Publishing

Although this publication has been composed with most care, neither author nor publisher can accept any liability for damage caused by possible errors and/or incompleteness in this publication.

No part of this publication may be reproduced in any form by print, photo print, microfilm or other means without written permission by the publisher.

Foreword by Ken Schwaber

An outstanding accomplishment that simmers with intelligence.

Scrum – A Pocket Guide is an extraordinarily competent book. Gunther has described everything about Scrum in well-formed, clearly written descriptions that flow with insight, understanding, and perception. Yet, you are never struck by these attributes. You simply benefit from them, later thinking, “That was really, really helpful. I found what I needed to know, readily understood what I wanted, and wasn’t bothered by irrelevancies.”

I have struggled to write this foreword. I feel the foreword should be as well-written as the book it describes. In this case, that is hard. Read Gunther’s book. Read it in part, or read it in whole. You will be satisfied.

Scrum is simple, but complete and competent in addressing complex problems. Gunther’s pocket guide is complete and competent in addressing understanding a simple framework for addressing complex problems, Scrum.

Ken, 22 August 2013

Preface

The use of Agile methods continues to gain traction with Scrum as the most widely adopted definition of Agile. The general level of interest in Scrum is already huge, yet still growing, in and beyond software development.

Transforming an organization's way of working to Scrum represents quite a challenge. Scrum is not a cookbook 'process' with detailed and exhaustive prescriptions for every imaginable situation. Scrum is a *framework* of principles, rules and values that thrives on the *people* employing Scrum. A major potential of Scrum lies in the discovery and *emergence* of practices, tools and techniques and in optimizing them for each specific context.

The benefits realized through Scrum depend on the will to remove barriers, think across walls and separations, and embark on a journey of discovery. Scrum is more about behavior than it is about process.

The journey starts by understanding the rules of Scrum to start playing Scrum. This book aspires to be your companion along the way, all the way. This book shows how Scrum implements the Agile mindset, what the rules of the game of Scrum are, and how these rules leave room for a variety of tactics to play the game. The ambition of introducing all these aspects is to make this book a worthwhile read for people, teams, managers and change

agents, whether they are already doing Scrum or want to embark on their journey of Scrum.

In 2003 my journey took off. My path of agility started with eXtreme Programming and Scrum. It has inevitably been a cobblestone path. On my journey I have used Scrum with many teams, in various projects and initiatives, at different scales and at different organizations. I have worked with both large and small enterprises and have coached individual practitioners and teams as well as executive management. I created the first edition of this book, '*Scrum - A Pocket Guide*'. I was in the fortunate position of partnering with Ken Schwaber, co-creator of Scrum, at Scrum.org, while also shepherding their 'Professional' series of Scrum trainings, courseware and assessments. I am gratified to continue my journey of Scrum as an independent Scrum Caretaker.

Who would have figured that there was demand for a second edition of my pocket guide to Scrum, five years after the almost accidental creation of the first edition in 2013?

I consider how I described the Scrum Values in the first edition. In July 2016 they were added to the Scrum Guide. I described the traditional 3 questions as a good, but optional tactic to use at the Daily Scrum. That too was added to the Scrum Guide, in November 2017.

However, more and bigger challenges have surfaced since 2013. The balance of society keeps drastically and rapidly shifting from industrial (often physical) labor to digital (often virtual) work. In many domains of society, the unpredictability of work increases continually. The industrial paradigm is rendered useless, definitely. The need for the Agile paradigm is bigger than ever, and thus the need for the tangible framework of Scrum to help people and organizations increase their agility in performing complex work in complex circumstances.

Scrum is increasingly being discovered as *a simple framework to address complex challenges*, more than as a way to deliver complex (software) products. More and different people ask for guidance and insights on their journey of Scrum in domains beyond software development. It required a more generic description of the rules of Scrum, different words, other angles to the same set of rules. Organizations look for clear insights in the simple rules of Scrum as they envision re-emerging their structures and their way of working around Scrum. As the third Scrum wave is rising, this second edition introduces or refreshes the simplicity of Scrum for those that want to surf that wave. This second edition offers the foundational insights into Scrum for them and their organizations to properly shape their Scrum.

I thank Ken for the foreword and his review for the original (2013) edition as well as the other reviewers Dave Starr, Patricia Kong and Ralph Jocham for their feedback on that first edition. I thank Blake McMillan and Dominik Maximini for their much-appreciated review of this second edition. I thank all translators for their past and on-going efforts to spread my words in different languages. I thank all at Van Haren Publishing, and especially Ivo van Haren, for giving me the chance to express my views on Scrum with this book.

Enjoy reading, and... keep Scrumming.

Gunther, June 2013 / August 2018

Reviews

This Scrum Pocket Guide is outstanding. It is well organized, well written, and the content is excellent. This should be the de facto standard handout for all looking for a complete, yet clear overview of Scrum.

(Ken Schwaber, Scrum co-creator, August 2013)

Gunther has expertly packaged the right no-nonsense guidance for teams seeking agility, without a drop of hyperbole. This is the book about agility with Scrum I wish I had written.

(David Starr, Agile Craftsman, June 2013)

During my many Scrum training activities I often get asked: “For Scrum, what is the one book to read?” In the past the answer wasn’t straightforward, but now it is! The Scrum Pocket Guide is the one book to read when starting with Scrum. It is a concise, yet complete and passionate reference about Scrum.

(Ralph Jocham, Agile Professional, effective agile., June 2013)

“The house of Scrum is a warm house. It’s a house where people are WELCOME.” Gunther’s passion for Scrum and its players is evident in his work and in each chapter of this book. He explains the Agile paradigm, lays

out the Scrum framework and then discusses the ‘future state of Scrum.’ Intimately, in about 100 pages.

(Patricia M. Kong, Business Agility Enterprise Solutions, Scrum.org, June 2013)

I recommend reading *Scrum – A Pocket Guide* early in your Scrum journey to help you gain a deeper understanding of why Scrum works and how the values and principles can positively impact the lives of your team as well as the health of an organization. Reading it later in your journey is great too... except for the feeling of regret wishing you had read it earlier.

(Blake McMillan, Scrum Master – Soulofscrums.com, August 2018)

It is hard to find concise, to the point literature about Scrum. Most authors circle around the core topics instead of naming them. Gunther chose to break this pattern, enlightening us with the knowledge of the truly important parts of Scrum. When starting on your Scrum journey, make sure to take a copy of this guide along with you.

(Dominik Maximini, Agile Lead, NovaTec Consulting GmbH, August 2018)

Table of contents

1	THE AGILE PARADIGM	15
1.1	To shift or not to shift.	15
1.2	The origins of Agile	20
1.3	Definition of Agile	21
1.4	The iterative-incremental continuum	24
1.5	Agility can't be planned.	27
1.6	Combining Agile and Lean.	30
2	SCRUM	39
2.1	The house of Scrum	39
2.2	Scrum, what's in a name?	40
2.3	Is that a gorilla I see over there?	43
2.4	Framework, not methodology	47
2.5	Playing the game.	49
2.6	Core principles of Scrum.	63
2.7	The Scrum values	73
3	TACTICS FOR A PURPOSE	79
3.1	Visualizing progress.	80
3.2	The Daily Scrum questions	82
3.3	Product Backlog refinement	83
3.4	User Stories	84

3.5	Planning Poker	85
3.6	Sprint length	86
3.7	How Scrum scales.....	88
4	THE FUTURE STATE OF SCRUM	95
4.1	Yes, we do Scrum. And... ..	95
4.2	The power of the possible product.....	97
4.3	The upstream adoption of Scrum	99
	ANNEXES.....	105
	Annex A: Scrum Glossary	105
	Annex B: References	109
	About the author	113

1

The Agile paradigm

1.1 TO SHIFT OR NOT TO SHIFT

The software industry was for a long time dominated by a paradigm of *industrial* views and beliefs. This was in fact a copy-paste of old manufacturing routines and theories. An essential element in this landscape of knowledge, views and practices was the Taylorist¹ conviction that ‘workers’ can’t be trusted to intelligently, autonomously and creatively perform their work. They are expected to only carry out pre-defined, executable tasks. Their work must be prepared, designed and planned by more senior staff. And then still, hierarchical supervisors must vigilantly oversee the execution of these carefully prepared tasks. Quality is assured

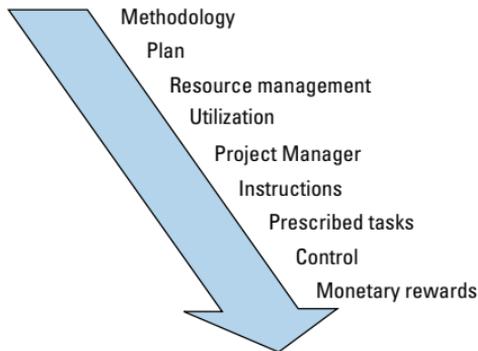


Figure 1.1 The industrial paradigm

by admitting the good and rejecting the bad batches of outputs. Monetary rewards are used to stimulate desired behavior. Unwanted behavior is punished. The old ‘carrots and sticks’ strategies.

The serious flaws of the old paradigm in software development are known and well documented. In particular, the Chaos reports of the Standish Group [Standish, 2011; Standish, 2013] have over and over revealed the low success rates of traditional software development. Many shortcomings and errors resulting from the application of the industrial paradigm in software development are well beyond reasonable levels of tolerance. The unfortunate response seems to have been to lower expectations. It became accepted that only 10-20% of software projects were successful. The definition of ‘success’ in the industrial paradigm is made up of the combination of on-time, within budget and including all scope. *Although these criteria for success can be disputed, it is the paradigm’s promise.* It became accepted that quality is low, and that over 50% of features of traditionally delivered software applications are never used [Standish, 2002; Standish, 2013].

Although it is not widely and consciously admitted, the industrial paradigm did put the software industry in a serious crisis. Many tried to overcome this crisis by fortifying the industrial approach. More plans were created, more phases scheduled, more designs made, more work was done upfront, hoping that the actual work would be executed more effectively. The exhaustiveness of the upfront work was increased. The core idea remained that the ‘workers’ needed to be directed, but with even more detailed instructions. Supervision was increased and intensified. *As the success rates did not increase, the industrial paradigm assumes that the instructions are not clear and detailed enough.*

Yet, little improved. Many flaws, defects and low quality remained and had to be tolerated.

It took some time, but inevitably new ideas and insights started forming upon observing the significant anomalies of the industrial paradigm.

The seeds of a new world view were already sown in the 1990's. But it was in 2001 that these resulted in the formal naming of 'Agile', a turning-point in the history of software development. A new paradigm was born, in the realm of the software industry but in the meantime expanding to other domains of society. It is a paradigm that thrives upon heuristics and creativity, a paradigm that thrives upon the (restored) respect for the creative nature of the work and the intelligence of the 'workers'.

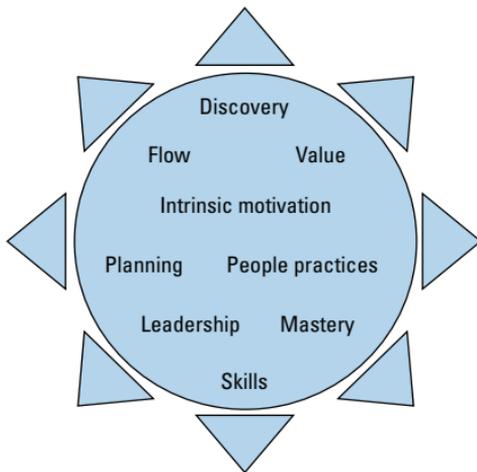


Figure 1.2 The Agile paradigm

The software industry has good reasons to keep moving to the new paradigm; the existing flaws are significant, widely known and the presence of software in society grows exponentially, making it a critical aspect of our modern world. However, by definition, a shift to a new paradigm takes time. And the old paradigm seems to have deep roots and a considerable half-life time. An industrial approach to software development continues to be taught and promoted as the most appropriate one.

Many say that Agile is too radical and they, therefore, propagate a gradual introduction of Agile practices within the existing, traditional frames. However, there is reason to be very skeptical about such gradual evolution, a slow progression from the old to the new paradigm, from waterfall to Agile.

The chances are high that a gradual evolution will never go beyond the surface, will not do more than just scratch that surface. New names will be installed, new terms and new practices will be imposed, but the fundamental thinking and behavior of people and organizations remain the same. Essential flaws remain untouched; especially the disrespect for people that leads to the continued treatment of creative, intelligent people as mindless ‘workers’, as ‘resources’.

The preservation of the traditional foundations will keep existing data, metrics and standards in place, and the new paradigm will be measured against those old standards. Different paradigms by their nature however consist of fundamentally different concepts and ideas, generally mutually exclusive. No meaningful comparison between the industrial and the Agile paradigm is possible. It requires the honesty to accept the serious flaws of the old ways. It requires leadership, vision, entrepreneurship and persistence to embrace the new ways, thereby abandoning the old thinking.

A gradual shift is factually a status-quo situation that keeps the industrial paradigm intact.

There is overwhelming evidence that the old paradigm doesn't work. Much of the evidence on Agile used to be anecdotal, personal or relatively minor. The Chaos report of 2011 by the Standish Group [Standish, 2011] marked a turning point, holding clear research results for the first time that were confirmed in all later Chaos reports. Extensive research was done in comparing traditional projects with projects that used Agile methods. The report shows that an Agile approach results in a much higher yield, even against the old expectations that software must be delivered on time, on

budget and with all the promised scope. The report shows that the Agile projects were three times as successful, and there were three times fewer failed Agile projects compared to traditional projects. For large projects however, the changes in success rates were less outspoken, which is likely more about starting with the wrong expectations in the large, i.e. the combination of time+budget+scope. Against the right expectations, with a focus on active customer collaboration and frequent delivery of value, the new paradigm would be performing even better, with vertical slices of value, frequently delivered, to overcome the volume problem.

Yet, Agile is a choice, not a must. It is one way to improve the software industry. Research shows it is more successful.



Scrum helps.

The distinct rules of Scrum help in getting a grip on the new paradigm. The small set of prescriptions allows immediate action and results in a more fruitful absorption of the new paradigm. Scrum is a tangible way to adopt the Agile paradigm. Using Scrum, people do develop new ways of working; through discovery, experimentation-based learning and collaboration. They enter a new state of being, a state of *agility*; a state of constant change, flux, evolution and adaptation. This process helps their organizations transform towards such a *state* of agility, freeing up time, people and energy for being innovative (again).

Nevertheless, despite its practicality, experience shows that adopting Scrum often represents a giant leap. This may be because of the uncertainty induced by letting go of old certainties, even if those old certainties have proven not to be very reliable. It may be the time that it takes to make a substantial shift. It may be the determination and hard work that is required. Over and over again it is shown that Scrum is simple, not easy.

1.2 THE ORIGINS OF AGILE

Despite the domination of the plan-driven, industrial views, an evolutionary approach to software development is not new. Craig Larman has extensively described the historical predecessors of Agile in his book *'Agile & Iterative Development, A Manager's Guide'* [Larman, 2004].

But the official label 'Agile' dates from February 2001, when 17 software development leaders gathered at the Snowbird ski resort in Utah. They discussed their views on software development in times when the failing waterfall approaches were replaced by heavy-weight RUP implementations ('Rational Unified Process'), which did not lead to better results than the traditional processes. These development leaders were following different paths and methods, each being a distinct implementation of the new paradigm; Scrum, eXtreme Programming, Adaptive Software Development, Crystal, Feature Driven Development, etc.

The gathering resulted in assigning the label 'Agile' to the common principles, beliefs and thinking of these leaders and their methods. They were published as the *'Manifesto for Agile Software Development'* [Beck, et.al., 2001].

Often the desire "to do Agile" can be overheard. And all too often it is the desire for a magical solution, another silver bullet process that solves all problems. It makes me often state that *"Agile does not exist"*. Agile is not one fixed process, method or practice. Agile is the collection of principles that the methods for Agile software development have in common. Agile refers to the mindset, the convictions and the preferences expressed in the Manifesto for Agile Software Development.

The manifesto does help to grasp the ideas underpinning Agile. If you use it as a source to gain a deeper understanding of Agile, then I strongly advise looking at the 12 principles behind the 4 value statements, see <http://agilemanifesto.org/principles.html>.