The Addison Wesley Signature Series

A Mike Cohn Signature Book

Essential Scrum

A Practical Guide to the Most Popular Agile Process

Kenneth S. Rubin



Forewords by Mike Cohn and Ron Jeffries

Phone/WhatsApp +86 185 7567 8796
+1 (346) 638 0369

Praise for *Essential Scrum*

"Agile coaches, you're gonna be happy with this book. Kenny Rubin has created an indispensable resource for us. Do you have a manager who just doesn't 'get it'? Hand them this book and ask them to flip to Chapter 3 for a complete explanation of how Scrum is less risky than plan-driven management. It's written just for them—in management-speak. Want to help the team come to a common understanding of Scrum? The visual icon language used throughout this book will help you help them. These are just two ways this book can aid you to coach Scrum teams. Use it well."

—Lyssa Adkins, Coach of Agile Coaches, Agile Coaching Institute; author, *Coaching Agile Teams*

"One of the best, most comprehensive descriptions of the core Scrum framework out there! *Essential Scrum* is for anyone—new to or experienced with Scrum—who's interested in the most important aspects of the process. Kenny does an excellent job of distilling the key tenets of the Scrum framework into a simple format with compelling visuals. As a Scrum coach for many teams, I continually reference the material for new ways to help teams that are learning and practicing the framework. I've seen Scrum continually misinterpreted and poorly implemented by big companies and tool vendors for more than ten years. Reading this book will help you get back to the basics and focus on what's important."

—Joe Balistrieri, Process Development Manager, Rockwell Automation

"Corporate IT leadership, which has been slow to embrace agile methods, would benefit immensely from giving a copy of this book to all of their project and delivery managers. Kenny Rubin has laid out in this book all the pragmatic business case and process materials needed for any corporate IT shop to successfully implement Scrum."

—John F. Bauer III, veteran of technical solution delivery in large corporate IT shops

"Kenny's extensive experience as a consultant, trainer, and past managing director of the Scrum Alliance is evident in this book. Along with providing the basics and introduction to Scrum, this book addresses the questions of masses—what happens to project managers? *Essential Scrum* helps us understand the big picture and guides how organization leaders can support and be involved with their Scrum teams for successful agile transformations."

—Sameer S. Bendre, CSM, PMP, Senior Consultant, 3i Infotech Inc.

"If you're new to agile development or to Scrum, this book will give you a flying start. The examples and descriptions are clear and vivid, and you'll often find yourself asking a question just before the book addresses that very topic."

—Johannes Brodwall, Principal Solution Architect, Steria Norway

"Kenny's well-structured explanations have a clarity to them that echoes the sensibilities of Smalltalk—the development environment with which he worked for years and from which both Scrum and Extreme Programming were born. This book pulls together a thorough set of agile management principles that really hit the mark and will no doubt guide you toward a more effective agile approach."

—Rowan Bunning, Founder, Scrum WithStyle

"There are lots of books on Scrum these days, but this book takes a new angle: a reality check for software practitioners. Kenny uses real-world examples and clear illustrations to show what makes a solid foundation for successful agile development. Readers will understand the value of building quality in, and the reality that we can't get everything right up front; we must work incrementally and learn as we go. It might have 'Scrum' in the title, but the book leverages effective practices from the larger agile universe to help managers and their teams succeed."

—Lisa Crispin, coauthor, *Agile Testing*

"Kenny Rubin managed to write the book that I want everyone associated with Scrum development to read! He covers everything you'll need to know about Scrum and more!"

—Martine Devos, European Scrum Pioneer and Certified Scrum Trainer

"I've reviewed a number of agile books in the past few years, so the question of 'Do we really need another one?' always comes to my mind. In the case of Kenny's book, I very much believe the answer is 'yes.' Getting the benefit of different, experienced perspectives on commonly encountered and needed material is valuable. Kenny has one of those valuable perspectives. One unique aspect of the book is an interesting 'iconography'—a new icon language for Scrum and agile that Kenny has created. I believe you'll find value-added material in this book to expand your ideas for how Scrum can be applied."

—Scott Duncan, Agile/Scrum coach and trainer

"Anyone who has had Scrum training or has been part of a Scrum team will find *Essential Scrum* to be a great follow-up read. It dives into the details of how to become more agile through implementing Scrum processes, and it explains exactly how to break down complex projects into manageable initiatives (or 'sprints'). Kenny Rubin provides a wealth of relevant case studies on what worked—or what didn't—in a

Phone/WhatsApp: +86 135 7567 8796
+1 (346) 638-0369

variety of organizations. The simple layout and businesslike graphics make it easy to scan quickly and find specific topics. Any organization that is seeking to evolve from a traditional waterfall approach toward a more agile methodology will find *Essential Scrum* a definitive guidebook for the journey."

—Julia Frazier, product manager

"Developing software is hard. Adopting a new way of working while in a project is even harder. This book offers a bypass of many of the pitfalls and will accelerate a team's ability to produce business value and become successful with Scrum. I wish I had this kind of book when I started using Scrum."

—Geir Hedemark, Development Manager, Basefarm AS

"I am convinced that *Essential Scrum* will become the foundation reference for the next generation of Scrum practitioners. Not only is it the most comprehensive introduction to Scrum available today, but it is also extremely well written and easy on the eye with its fantastic new visual Scrum language. If that isn't enough, Kenny shares a range of his valuable personal insights and experiences that we can all certainly learn from."

—Ilan Goldstein, Agile Solutions Manager, Reed Elsevier

"Scrum is elegantly simple, yet deceptively complex. In *Essential Scrum,* Kenny Rubin provides us with a step-by-step guide to those complexities while retaining the essential simplicity. Real-world experiences coupled with enlightening illustrations make Scrum come to life. For senior managers and team members alike, this is a must-read book if you are starting or considering whether to implement Scrum in your organization. This will certainly be a book recommended to my students."

—John Hebley, Hebley & Associates

"Kenny unpacks a wealth of wisdom and knowledge in *Essential Scrum,* providing valuable and comprehensive insights to the practical application of agile/Scrum. Whether you're new to agile or are looking to reach a greater maturity of continuous improvement in your organization, this is a definitive handbook for your toolbox."

—David Luzquiños, Head of Agile Enablement, Agile Coach, Betfair

"Kenny Rubin continues to provide clarity and insight into adopting agile in a pragmatic way. In one hand he holds the formal or ideal Scrum definition, and in the other, the pragmatic application of it. He brings the wisdom of his workshops and years of experience to the table for you to read in his latest book. If you are about to start out on your agile adoption journey or are seeking guidance midcourse, grab a copy."

—Cuan Mulligan, freelance coactive Agile coach

Phone/What's App: +86 195 7567 8798
+1 (346) 628 0369

"A decade after publication of the first Scrum books, it is time to combine the essential aspects of the Scrum framework with the practical experiences and approaches of the last ten years. Kenny Rubin does so in a satisfying and nondogmatic way. The reader gets a pragmatic look at Scrum and learns when and how to best apply Scrum to achieve business benefits."

—Yves Stalgies, Ph.D., Director IT, www.etracker.com

"Adoption of Scrum is most successful when everyone involved—even peripherally—with product development has a good understanding of the fundamentals. *Essential Scrum* provides an ideal overview of both the big picture and the details in an accessible style. It is sure to become a standard reference."

—Kevin Tureski, Principal, Kevin Tureski Consulting

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 638-0369

Essential Scrum

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 638-0369

The Addison-Wesley Signature Series

Kent Beck, Mike Cohn, and Martin Fowler, Consulting Editors



Visit **informit.com/awss** for a complete list of available products.

The Addison-Wesley Signature Series provides readers with practical and authoritative information on the latest trends in modern technology for computer professionals. The series is based on one simple premise: Great books come from great authors. Titles in the series are personally chosen by expert advisors, world-class authors in their own right. These experts are proud to put their signatures on the covers, and their signatures ensure that these thought leaders have worked closely with authors to define topic coverage, book scope, critical content, and overall uniqueness. The expert signatures also symbolize a promise to our readers: You are reading a future classic.

Phone/WhatsApp: +86 135 7567 8796
+1 (346) 638-0369

Make sure to connect with us!
informit.com/socialconnect

Addison Wesley | informIT.com THE TRUSTED TECHNOLOGY LEARNING SOURCE | Safari Books Online

Essential Scrum

A Practical Guide to the Most Popular Agile Process

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 638-0369

Kenneth S. Rubin

Addison-Wesley
Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The author and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U.S. Corporate and Government Sales
(800) 382-3419
corpsales@pearsontechgroup.com

For sales outside the United States, please contact:

International Sales
international@pearson.com

Visit us on the Web: informit.com/aw

Library of Congress Cataloging-in-Publication Data
Rubin, Kenneth S.
Essential Scrum : a practical guide to the most popular agile process / Kenneth S. Rubin.
p. cm.
Includes bibliographical references and index.
ISBN 978-0-13-704329-3 (pbk. : alk. paper)—ISBN 0-13-704329-5 (pbk. : alk. paper)
1. Scrum (Computer software development) 2. Agile software development. 3. Project management. I. Title.
QA76.76.D47R824 2012
005.1—dc23

2012010892

Copyright © 2013 Pearson Education, Inc.

Agile visual icon language copyright © Kenneth S. Rubin and used with permission.

All rights reserved. Printed in the United States of America. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. To obtain permission to use material from this work, please submit a written request to Pearson Education, Inc., Permissions Department, One Lake Street, Upper Saddle River, New Jersey 07458, or you may fax your request to (201) 236-3290.

ISBN-13: 978-0-13-704329-3
ISBN-10: 0-13-704329-5
Text printed in the United States on recycled paper at Edwards Brothers Malloy in Ann Arbor, Michigan.
First printing, July 2012

Phone/WhatsApp: +86 185 7587 8796
+1 (346) 638-0369

To my wife, Jenine, for all your loving support
To my sons, Jonah and Asher, for inspiring me
To my father, Manny, for teaching me the value of hard work
To my mother, Joyce, for showing me what real courage looks like
(may her memory be a blessing)

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 638-0369

This page intentionally left blank

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 638-0369

Contents

List of Figures xxv
Foreword by Mike Cohn xxxi
Foreword by Ron Jeffries xxxiii
Preface xxxv
Acknowledgments xxxix
About the Author xliii

Chapter 1 Introduction 1
What Is Scrum? 1
Scrum Origins 3
Why Scrum? 4
Genomica Results 4
Can Scrum Help You? 5
Complex Domain 8
Complicated Domain 8
Simple Domain 8
Chaotic Domain 9
Disorder 9
Interrupt-Driven Work 9
Closing 10

PART I Core Concepts 11

Chapter 2 Scrum Framework 13
Overview 13
Scrum Roles 14
Product Owner 15
ScrumMaster 16
Development Team 16
Scrum Activities and Artifacts 16
Product Backlog 18

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 638-0369

Sprints 20
Sprint Planning 21
Sprint Execution 23
Daily Scrum 23
Done 25
Sprint Review 26
Sprint Retrospective 27
Closing 28

Chapter 3 Agile Principles 29
Overview 29
Variability and Uncertainty 32
Embrace Helpful Variability 32
Employ Iterative and Incremental Development 33
Leverage Variability through Inspection, Adaptation, and Transparency 35
Reduce All Forms of Uncertainty Simultaneously 36
Prediction and Adaptation 37
Keep Options Open 37
Accept That You Can't Get It Right Up Front 38
Favor an Adaptive, Exploratory Approach 39
Embrace Change in an Economically Sensible Way 40
Balance Predictive Up-Front Work with Adaptive Just-in-Time Work 43
Validated Learning 44
Validate Important Assumptions Fast 45
Leverage Multiple Concurrent Learning Loops 45
Organize Workflow for Fast Feedback 46
Work in Process (WIP) 48
Use Economically Sensible Batch Sizes 48
Recognize Inventory and Manage It for Good Flow 49
Focus on Idle Work, Not Idle Workers 51
Consider Cost of Delay 52
Progress 54
Adapt to Real-Time Information and Replan 54
Measure Progress by Validating Working Assets 54
Focus on Value-Centric Delivery 55
Performance 56
Go Fast but Never Hurry 56
Build In Quality 56
Employ Minimally Sufficient Ceremony 57
Closing 58

Buy@WhatsApp: +86 185 7567 8796
+1 (346) 638-0369

Chapter 4 Sprints 61
Overview 61
Timeboxed 62
Establishes a WIP Limit 62
Forces Prioritization 62
Demonstrates Progress 62
Avoids Unnecessary Perfectionism 63
Motivates Closure 63
Improves Predictability 64
Short Duration 64
Ease of Planning 64
Fast Feedback 64
Improved Return on Investment 65
Bounded Error 65
Rejuvenated Excitement 65
Frequent Checkpoints 66
Consistent Duration 67
Cadence Benefits 67
Simplifies Planning 68
No Goal-Altering Changes 69
What Is a Sprint Goal? 69
Mutual Commitment 69
Change versus Clarification 69
Consequences of Change 70
Being Pragmatic 72
Abnormal Termination 72
Definition of Done 74
What Is the Definition of Done? 74
Definition of Done Can Evolve Over Time 76
Definition of Done versus Acceptance Criteria 77
Done versus Done-Done 77
Closing 78

Chapter 5 Requirements and User Stories 79
Overview 79
Using Conversations 81
Progressive Refinement 82
What Are User Stories? 83
Card 83
Conversation 84
Confirmation 85

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 638-0369

Level of Detail 86
INVEST in Good Stories 88
Independent 88
Negotiable 89
Valuable 90
Estimatable 91
Sized Appropriately (Small) 92
Testable 92
Nonfunctional Requirements 93
Knowledge-Acquisition Stories 93
Gathering Stories 95
User-Story-Writing Workshop 95
Story Mapping 96
Closing 98

Chapter 6 Product Backlog 99
Overview 99
Product Backlog Items 100
Good Product Backlog Characteristics 101
Detailed Appropriately 101
Emergent 102
Estimated 102
Prioritized 103
Grooming 104
What Is Grooming? 104
Who Does the Grooming? 105
When Does Grooming Take Place? 106
Definition of Ready 108
Flow Management 110
Release Flow Management 110
Sprint Flow Management 111
Which and How Many Product Backlogs? 112
What Is a Product? 113
Large Products—Hierarchical Backlogs 114
Multiple Teams—One Product Backlog 115
One Team—Multiple Products 117
Closing 118

Chapter 7 Estimation and Velocity 119
Overview 119
What and When We Estimate 120
Portfolio Backlog Item Estimates 121

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 638-0369

Product Backlog Estimates 121
Task Estimates 122
PBI Estimation Concepts 123
Estimate as a Team 123
Estimates Are Not Commitments 124
Accuracy versus Precision 125
Relative Size Estimation 125
PBI Estimation Units 128
Story Points 128
Ideal Days 128
Planning Poker 129
Estimation Scale 130
How to Play 131
Benefits 133
What Is Velocity? 133
Calculate a Velocity Range 134
Forecasting Velocity 135
Affecting Velocity 135
Misusing Velocity 137
Closing 138

Chapter 8 Technical Debt 139
Overview 139
Consequences of Technical Debt 141
Unpredictable Tipping Point 142
Increased Time to Delivery 142
Significant Number of Defects 142
Rising Development and Support Costs 142
Product Atrophy 143
Decreased Predictability 143
Underperformance 143
Universal Frustration 144
Decreased Customer Satisfaction 144
Causes of Technical Debt 144
Pressure to Meet a Deadline 144
Attempting to Falsely Accelerate Velocity 145
Myth: Less Testing Can Accelerate Velocity 145
Debt Builds on Debt 147
Technical Debt Must Be Managed 148
Managing the Accrual of Technical Debt 149
Use Good Technical Practices 149
Use a Strong Definition of Done 149
Properly Understand Technical Debt Economics 150

PhoneWhatsApp: +86 185 7567 8796
+1 (346) 638-0369

Making Technical Debt Visible 153
Make Technical Debt Visible at the Business Level 153
Make Technical Debt Visible at the Technical Level 154
Servicing the Technical Debt 155
Not All Technical Debt Should Be Repaid 157
Apply the Boy Scout Rule (Service Debt When You Happen Upon It) 158
Repay Technical Debt Incrementally 159
Repay the High-Interest Technical Debt First 160
Repay Technical Debt While Performing Customer-Valuable Work 160
Closing 162

PART II Roles 163

Chapter 9 Product Owner 165
Overview 165
Principal Responsibilities 166
Manage Economics 167
Participate in Planning 168
Groom the Product Backlog 169
Define Acceptance Criteria and Verify That They Are Met 169
Collaborate with the Development Team 170
Collaborate with the Stakeholders 171
Characteristics/Skills 171
Domain Skills 171
People Skills 172
Decision Making 173
Accountability 173
A Day in the Life 174
Who Should Be a Product Owner? 176
Internal Development 176
Commercial Development 177
Outsourced Development Project 180
Component Development 180
Product Owner Combined with Other Roles 181
Product Owner Team 182
Product Owner Proxy 183
Chief Product Owner 183
Closing 184

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 638 0369

Chapter 10 ScrumMaster 185
Overview 185
Principal Responsibilities 185
Coach 185
Servant Leader 186
Process Authority 186
Interference Shield 187
Impediment Remover 187
Change Agent 187
Characteristics/Skills 188
Knowledgeable 188
Questioning 188
Patient 189
Collaborative 189
Protective 189
Transparent 189
A Day in the Life 190
Fulfilling the Role 191
Who Should Be a ScrumMaster? 191
Is ScrumMaster a Full-Time Job? 192
ScrumMaster Combined with Other Roles 192
Closing 193

Chapter 11 Development Team 195
Overview 195
Role-Specific Teams 195
Principal Responsibilities 196
Perform Sprint Execution 196
Inspect and Adapt Each Day 197
Groom the Product Backlog 197
Plan the Sprint 197
Inspect and Adapt the Product and Process 197
Characteristics/Skills 198
Self-Organizing 198
Cross-Functionally Diverse and Sufficient 200
T-Shaped Skills 201
Musketeer Attitude 203
High-Bandwidth Communications 204
Transparent Communication 205
Right-Sized 206
Focused and Committed 207

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 638-0369

Working at a Sustainable Pace 208
Long-Lived 209
Closing 211

Chapter 12 Scrum Team Structures 213
Overview 213
Feature Teams versus Component Teams 213
Multiple-Team Coordination 218
Scrum of Scrums 218
Release Train 220
Closing 223

Chapter 13 Managers 225
Overview 225
Fashioning Teams 227
Define Boundaries 227
Provide a Clear Elevating Goal 228
Form Teams 228
Change Team Composition 229
Empower Teams 230
Nurturing Teams 231
Energize People 231
Develop Competence 231
Provide Functional-Area Leadership 232
Maintain Team Integrity 233
Aligning and Adapting the Environment 233
Promote Agile Values 233
Remove Organizational Impediments 234
Align Internal Groups 234
Align Partners 234
Managing Value-Creation Flow 235
Take a Systems Perspective 235
Manage Economics 236
Monitor Measures and Reports 236
Project Managers 237
Project Management Responsibilities on a Scrum Team 237
Retaining a Separate Project Manager Role 239
Closing 243

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 638-0369

PART III Planning 245

Chapter 14 Scrum Planning Principles 247
Overview 247
Don't Assume We Can Get the Plans Right Up Front 248
Up-Front Planning Should Be Helpful without Being Excessive 248
Keep Planning Options Open Until the Last Responsible Moment 249
Focus More on Adapting and Replanning Than on Conforming to a Plan 249
Correctly Manage the Planning Inventory 251
Favor Smaller and More Frequent Releases 252
Plan to Learn Fast and Pivot When Necessary 254
Closing 255

Chapter 15 Multilevel Planning 257
Overview 257
Portfolio Planning 259
Product Planning (Envisioning) 259
Vision 259
High-Level Product Backlog 259
Product Roadmap 260
Release Planning 261
Sprint Planning 264
Daily Planning 264
Closing 265

Chapter 16 Portfolio Planning 267
Overview 267
Timing 267
Participants 268
Process 268
Scheduling Strategies 270
Optimize for Lifecycle Profits 270
Calculate Cost of Delay 271
Estimate for Accuracy, Not Precision 274
Inflow Strategies 275
Apply the Economic Filter 275
Balance the Arrival Rate with the Departure Rate 276
Quickly Embrace Emergent Opportunities 278
Plan for Smaller, More Frequent Releases 279

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 638-0369

Outflow Strategies 280
Focus on Idle Work, Not Idle Workers 281
Establish a WIP Limit 281
Wait for a Complete Team 282
In-Process Strategies 283
Use Marginal Economics 283
Closing 285

Chapter 17 Envisioning (Product Planning) 287
Overview 287
Timing 287
Participants 288
Process 290
SR4U Example 290
Visioning 291
High-Level Product Backlog Creation 294
Product Roadmap Definition 295
Other Activities 298
Economically Sensible Envisioning 299
Target a Realistic Confidence Threshold 300
Focus on a Short Horizon 302
Act Quickly 302
Pay for Validated Learning 303
Use Incremental/Provisional Funding 304
Learn Fast and Pivot (aka Fail Fast) 305
Closing 306

Chapter 18 Release Planning (Longer-Term Planning) 307
Overview 307
Timing 308
Participants 308
Process 309
Release Constraints 311
Fixed Everything 311
Fixed Scope and Date 312
Fixed Scope 313
Fixed Date 313
Variable Quality 314
Updating Constraints 314
Grooming the Product Backlog 315
Refine Minimum Releasable Features (MRFs) 316

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 638-0369

Sprint Mapping (PBI Slotting) 316
Fixed-Date Release Planning 318
Fixed-Scope Release Planning 323
Calculating Cost 325
Communicating 326
Communicating Progress on a Fixed-Scope Release 327
Communicating Progress on a Fixed-Date Release 329
Closing 330

PART IV Sprinting 333

Chapter 19 Sprint Planning 335
Overview 335
Timing 335
Participants 335
Process 336
Approaches to Sprint Planning 338
Two-Part Sprint Planning 338
One-Part Sprint Planning 339
Determining Capacity 340
What Is Capacity? 340
Capacity in Story Points 342
Capacity in Effort-Hours 342
Selecting Product Backlog Items 343
Acquiring Confidence 344
Refine the Sprint Goal 346
Finalize the Commitment 346
Closing 346

Chapter 20 Sprint Execution 347
Overview 347
Timing 347
Participants 348
Process 348
Sprint Execution Planning 349
Flow Management 349
Parallel Work and Swarming 350
Which Work to Start 352
How to Organize Task Work 352

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 638-0369

What Work Needs to Be Done? 353
Who Does the Work? 354
Daily Scrum 354
Task Performance—Technical Practices 355
Communicating 356
Task Board 356
Sprint Burndown Chart 357
Sprint Burnup Chart 359
Closing 360

Chapter 21 Sprint Review 363
Overview 363
Participants 364
Prework 365
Determine Whom to Invite 366
Schedule the Activity 366
Confirm That the Sprint Work Is Done 367
Prepare for the Demonstration 368
Determine Who Does What 368
Approach 368
Summarize 369
Demonstrate 370
Discuss 371
Adapt 371
Sprint Review Issues 372
Sign-offs 372
Sporadic Attendance 372
Large Development Efforts 373
Closing 373

Chapter 22 Sprint Retrospective 375
Overview 375
Participants 377
Prework 378
Define the Retrospective Focus 378
Select the Exercises 379
Gather Objective Data 379
Structure the Retrospective 380
Approach 380
Set the Atmosphere 382
Share Context 382

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 638-0369

Identify Insights 385
Determine Actions 387
Close the Retrospective 390
Follow Through 391
Sprint Retrospective Issues 392
Closing 393

Chapter 23 The Path Forward 395
There Is No End State 395
Discover Your Own Path 396
Sharing Best Practices 396
Using Scrum to Discover the Path Forward 397
Get Going! 398

Glossary 401

References 423

Index 427

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 638-0369

This page intentionally left blank

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 638-0369

List of Figures

Figure 1.1	Agile development overview	2
Figure 1.2	Scrum benefits	6
Figure 1.3	Cynefin framework	7
Figure 2.1	Scrum practices	14
Figure 2.2	Scrum roles	15
Figure 2.3	Scrum framework	17
Figure 2.4	Product backlog	19
Figure 2.5	Product backlog grooming	19
Figure 2.6	Product backlog item sizes	20
Figure 2.7	Sprint characteristics	21
Figure 2.8	Sprint planning	21
Figure 2.9	Sprint backlog	22
Figure 2.10	Sprint execution	23
Figure 2.11	Daily scrum	24
Figure 2.12	Sprint results (potentially shippable product increment)	25
Figure 2.13	Sprint review	27
Figure 2.14	Sprint retrospective	27
Figure 3.1	Waterfall process	30
Figure 3.2	Categorization of principles	31
Figure 3.3	Defined process	32
Figure 3.4	Scrum uses iterative and incremental development.	34
Figure 3.5	Scrum process model	36
Figure 3.6	Make decisions at the last responsible moment.	38
Figure 3.7	Plan-driven requirements acquisition relative to product knowledge	39
Figure 3.8	Historical cost of exploration	40
Figure 3.9	Significant late cost of change with sequential development	41
Figure 3.10	Self-fulfilling prophecy	42
Figure 3.11	Flattening the cost-of-change curve	43
Figure 3.12	Balancing predictive and adaptive work	44
Figure 3.13	Learning loop pattern	46
Figure 3.14	Component integration	47
Figure 3.15	How utilization affects queue size (delay)	52

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 638-0369

Figure 3.16	Deliver high-value features sooner.	55
Figure 3.17	Ceremony scale	58
Figure 4.1	Sprints are the skeleton of the Scrum framework.	61
Figure 4.2	The benefits of timeboxing	63
Figure 4.3	The benefits of short-duration sprints	64
Figure 4.4	Excitement over time	65
Figure 4.5	Checkpoint comparison	66
Figure 4.6	Cumulative investment at different states	71
Figure 4.7	Deciding on the next sprint length after sprint termination	73
Figure 5.1	Scrum uses placeholders for requirements.	81
Figure 5.2	A user story template and card	83
Figure 5.3	User story with additional data attached	84
Figure 5.4	User story conditions of satisfaction	85
Figure 5.5	User story abstraction hierarchy	87
Figure 5.6	Example epic	87
Figure 5.7	Example theme	88
Figure 5.8	Highly dependent stories	89
Figure 5.9	Example technical story	90
Figure 5.10	Undesirable technical story	91
Figure 5.11	Nonfunctional requirements	93
Figure 5.12	Knowledge-acquisition story	94
Figure 5.13	Story map	97
Figure 6.1	The product backlog is at the heart of the Scrum framework.	99
Figure 6.2	Product backlog items	100
Figure 6.3	Product backlog items are different sizes.	102
Figure 6.4	Product backlog items are estimated.	103
Figure 6.5	Product backlog items are prioritized.	104
Figure 6.6	Grooming reshapes the product backlog.	105
Figure 6.7	Grooming is a collaborative effort.	106
Figure 6.8	Outside-of-primary-flow grooming with sequential projects	107
Figure 6.9	When grooming happens	108
Figure 6.10	Definition of ready	109
Figure 6.11	Release-level view of the product backlog	111
Figure 6.12	The product backlog as a pipeline of requirements	112
Figure 6.13	The product backlog is associated with the product.	113
Figure 6.14	Hierarchical product backlogs	115
Figure 6.15	Team-specific view of the product backlog	116
Figure 6.16	Scenarios for multiple product backlogs	117

Phone/WhatsApp: +86 185 7567 8796
+1 (316) 638-0369

Figure 7.1	The relationship among size, velocity, and duration	120
Figure 7.2	What and when we estimate	121
Figure 7.3	Product backlog item estimating concepts	123
Figure 7.4	The full Scrum team participates in estimation.	124
Figure 7.5	Effect of committing on estimates	124
Figure 7.6	Effort versus accuracy when estimating	126
Figure 7.7	Relative size estimation	126
Figure 7.8	Absolute versus relative size estimation	127
Figure 7.9	Planning Poker concepts	129
Figure 7.10	Planning Poker uses binning.	130
Figure 7.11	Innolution Planning Poker cards	131
Figure 7.12	Calculating and using a velocity range	134
Figure 7.13	A team's velocity over time	136
Figure 7.14	The effect of overtime on velocity (based on a figure from Cook 2008)	137
Figure 8.1	Consequences of technical debt	141
Figure 8.2	Cost-of-change curve affected by technical debt	143
Figure 8.3	Pressure to meet a deadline can lead to technical debt.	145
Figure 8.4	Accruing technical debt to meet unreasonable fixed scope and date	146
Figure 8.5	The myth, reality, and good practice of how testing affects velocity	146
Figure 8.6	As technical debt increases, velocity decreases.	147
Figure 8.7	Activities for managing technical debt	148
Figure 8.8	Example technical debt economic analysis	150
Figure 8.9	Ways to make technical debt visible at the technical level	154
Figure 8.10	Approaches for servicing technical debt	156
Figure 8.11	A technique for managing technical debt when using Scrum	161
Figure 9.1	The product owner faces two directions simultaneously.	165
Figure 9.2	Principal product owner responsibilities	166
Figure 9.3	The product owner manages economics.	167
Figure 9.4	Comparison of customer or business engagement over time	170
Figure 9.5	Product owner characteristics	172
Figure 9.6	A day in the life of a product owner	174
Figure 9.7	Example of a product owner on internal development	177
Figure 9.8	Example of a product owner on commercial development	178
Figure 9.9	Pragmatic Marketing framework	179
Figure 9.10	Example of a product owner on outsourced development	180
Figure 9.11	Example of a product owner on component development	181
Figure 9.12	Same person as product owner of more than one Scrum team	182
Figure 9.13	Hierarchical product owner role	184

Phone/WhatsApp: +1 (340) 638-0569 +1 (346) 785 7967 8796

Figure 10.1	Principal ScrumMaster responsibilities	186
Figure 10.2	ScrumMaster characteristics	188
Figure 10.3	A day in the life of a ScrumMaster	190
Figure 10.4	Same person as ScrumMaster of more than one team	193
Figure 11.1	Development team responsibilities with respect to Scrum activities	196
Figure 11.2	Development team characteristics	198
Figure 11.3	Flocking isn't the result of top-down planning.	199
Figure 11.4	Flocking: simple rules and frequent feedback	200
Figure 11.5	Team diversity	201
Figure 11.6	T-shaped skills	202
Figure 11.7	Team members must act as if they are all in the same boat.	204
Figure 11.8	The cost of multitasking	208
Figure 11.9	Sustainable pace over time	209
Figure 12.1	One product and multiple component teams	214
Figure 12.2	Two products and multiple component teams	215
Figure 12.3	Combined feature team and component teams	217
Figure 12.4	Scrum of scrums	219
Figure 12.5	Release train structure	221
Figure 13.1	Greatest concerns about adopting agile	225
Figure 13.2	Functional manager responsibilities in a Scrum organization	226
Figure 13.3	Managers define the boundaries.	227
Figure 13.4	Functional managers collectively create Scrum teams.	228
Figure 13.5	Teams rarely have fully connected communication channels.	240
Figure 13.6	Teams frequently form collaboration clusters.	241
Figure 13.7	Funneling coordination through a project or program manager	242
Figure 13.8	Project manager on complex, multiparty development	243
Figure 14.1	Scrum planning principles	247
Figure 14.2	Big up-front Gantt chart	250
Figure 14.3	When the map and the terrain don't agree, believe the terrain.	251
Figure 14.4	Single-release economics	253
Figure 14.5	Multi-release economics	253
Figure 15.1	Different levels of planning	257
Figure 15.2	Scrum Alliance website product roadmap	261
Figure 15.3	A release line in the product backlog	262

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 638-0369

Figure 15.4	Product roadmap releases mapped to the product backlog	263
Figure 15.5	A release can encompass one or more sprints.	263
Figure 15.6	Each sprint has a sprint backlog.	264
Figure 15.7	Hierarchical Scrum planning	266
Figure 16.1	Portfolio-planning activity	268
Figure 16.2	Portfolio-planning strategies	269
Figure 16.3	Cost-of-delay profiles	273
Figure 16.4	Applying the economic filter	276
Figure 16.5	Balancing inflow and outflow in the portfolio backlog	277
Figure 16.6	The value of many emergent opportunities decays rapidly.	279
Figure 16.7	Large products in the portfolio backlog create a convoy.	280
Figure 16.8	Teams are the unit of capacity for establishing the product WIP limit.	282
Figure 16.9	In-process product decision flow based on marginal economics	284
Figure 17.1	Envisioning is an ongoing activity.	288
Figure 17.2	Envisioning (product-planning) activity	289
Figure 17.3	Areas of stakeholder value	292
Figure 17.4	Fixed, periodic releases	296
Figure 17.5	SmartReview4You product roadmap	297
Figure 17.6	SR4U knowledge-acquisition sprint storyboard	298
Figure 17.7	Guidelines for economically sensible envisioning	300
Figure 17.8	Consequences of setting the confidence threshold bar too high	301
Figure 17.9	Decision making under the illusion of certainty	303
Figure 17.10	Incremental/provisional funding	304
Figure 18.1	Different release cadences	307
Figure 18.2	When release planning happens	309
Figure 18.3	Release-planning activity	310
Figure 18.4	Fixed date and fixed scope playing a game of chicken	312
Figure 18.5	Mapping product backlog items to sprints	317
Figure 18.6	Sprint calendar for SR4U Release 1.0	319
Figure 18.7	Product backlog ready for release planning	321
Figure 18.8	Determining the range of features on a fixed-date release	322
Figure 18.9	Location of must-have features relative to the range of deliverable features	322
Figure 18.10	Results of fixed-scope planning	325
Figure 18.11	Fixed-scope-release burndown chart	327
Figure 18.12	Fixed-scope-release burnup chart	328

Phone/WhatsApp: +86 185 7561 8796
+1 (340) 693-0369

Figure 18.13	Variable-scope-release burnup chart	329
Figure 18.14	Fixed-date-release burnup chart (with inverted product backlog)	330
Figure 19.1	When sprint planning happens	336
Figure 19.2	Sprint-planning activity	337
Figure 19.3	Two-part sprint-planning approach	339
Figure 19.4	One-part sprint-planning approach	340
Figure 19.5	Development team capacity in a sprint	341
Figure 19.6	Sprint backlog showing PBIs and task plan	345
Figure 20.1	When sprint execution happens	347
Figure 20.2	Sprint execution activity	348
Figure 20.3	Cost of multitasking	350
Figure 20.4	Mini waterfall during sprint execution—a bad idea	352
Figure 20.5	Subset of Extreme Programming technical practices	355
Figure 20.6	Example task board	356
Figure 20.7	Sprint burndown chart	358
Figure 20.8	Sprint burndown chart with trend lines	359
Figure 20.9	Sprint burnup chart	360
Figure 21.1	When the sprint review happens	363
Figure 21.2	Sprint review prework	366
Figure 21.3	Sprint review activity	369
Figure 22.1	Edward Bear illustrating the need for a retrospective	376
Figure 22.2	When the sprint retrospective happens	376
Figure 22.3	Sprint retrospective prework	378
Figure 22.4	Sprint retrospective activity	381
Figure 22.5	Aligning perspectives to create a shared context	383
Figure 22.6	Sprint event timeline	384
Figure 22.7	Emotions seismograph	385
Figure 22.8	Retrospective insight card wall	386
Figure 22.9	Insight cards clustered into similarity groups	386
Figure 22.10	Insight cards placed into predetermined groups	387
Figure 22.11	Example of dot voting	388
Figure 22.12	Sprint retrospective issues	391

Phone/WhatsApp: +86 185 7567 8796
+1 (316) 638-0369

Foreword
by Mike Cohn

I had lunch today at a Burger King. A sign on the wall proclaimed the restaurant the "Home of the Whopper" and then proceeded to tell me there were over a million different ways to order a Whopper. If various combinations of extra or no pickles, tomatoes, lettuce, cheese, and so on can lead to over a million ways to make a hamburger, there must be billions of possible ways to implement Scrum. And while there is no single right way, there are better and worse ways to implement Scrum.

In *Essential Scrum*, Kenny Rubin helps readers find the better ways. His isn't a prescriptive book—he doesn't say, "You must do this." Instead, he teaches the essential principles underlying success with Scrum and then gives us choices in how we live up to those principles. For example, there is no one right way for all teams to plan a sprint. What works in one company or project will fail in another. And so Kenny gives us choices. He describes an overall structure for why Scrum teams plan sprints and what must result from sprint planning, and he gives us a couple of alternative approaches that will work. But ultimately the decision belongs to each team. Fortunately for those teams, they now have this book to help them.

An unexpected benefit of *Essential Scrum* is the visual language Kenny introduces for communicating about Scrum. I found these images very helpful in following along with the text, and I suspect they will become commonplace in future discussions of Scrum.

The world has needed this book for a long time. Scrum started as a small concept. The first book to talk about it—*Wicked Problems, Righteous Solutions* in 1990 by DeGrace and Stahl—did so in six pages. But in the more than 20 years since that book appeared, Scrum has expanded. New roles, meetings, and artifacts have been introduced and refined. With each new piece that was added, we were at risk of losing the heart of Scrum, that part of it that is about a team planning how to do something, doing some small part of it, and then reflecting on what the team members did and how well they did it together.

With *Essential Scrum*, Kenny brings us back to the heart of Scrum. And from there teams can begin to make the decisions necessary to implement Scrum, making it their own. This book serves as an indispensable guide, helping teams choose among the billions of possible ways of implementing Scrum and finding one that leads to success.

—Mike Cohn
Author of *Succeeding with Agile, Agile Estimating and Planning,* and *User Stories Applied*
www.mountaingoatsoftware.com

Rights@NetSAPP: +86 135 7557 8796
+1 (346) 638 0369

This page intentionally left blank

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 638-0369

Foreword
by Ron Jeffries

When Kenny asked me to write a foreword for *Essential Scrum,* I was thinking, "This will be quick and easy; it must be a short book going straight to a simple description of what Scrum is." I knew Kenny's work, so I knew it would be a good read, and short, too. What could be better!

Imagine my surprise and delight when I found that this book covers just about everything you'll need to know about Scrum, on the first day or years into your use of Scrum. And Kenny doesn't stop there. He starts with the central ideas, including the agile principles that underlie all the agile methods, and a quick view of the Scrum framework. Then he drills in, deeper and deeper. It's still a good read, and it's quite comprehensive as well.

Kenny covers planning in good detail, looking at requirements, stories, the backlog, estimation, velocity. Then he takes us deeper into the principles and helps us deal with all the levels of planning and all the time horizons. He describes how sprints are planned, executed, reviewed, and improved. And throughout, he gives us more than the basics, highlighting key issues that you may encounter as you go along.

My own focus in Scrum and agile is on the necessary developer skills to ensure that teams can deliver real, running, business-focused software, sprint after sprint. Kenny helps us understand how to use ideas like velocity and technical debt safely and well. Both of these are critical topics, and I commend them to your attention.

Velocity tells us how much the team is delivering over time. We can use it to get a sense of how much we're getting done and whether we're improving. Kenny warns us, however, that using velocity as a performance measure is damaging to our business results, and he helps us understand why.

Technical debt has become a very broad term, referring to almost everything that could go wrong in the code. Kenny helps us tease apart all the various meanings and helps us understand why we care about these seemingly technical details. In particular, I like his description of how putting a team under pressure will inevitably damage our prospects of getting a good product on time.

Scrum, like all agile methods, relies on an exploratory approach with rapid feedback. Kenny tells a story of his brief use of punch cards, and it reminded me of my earliest experience with computing, many years before Kenny saw his first punch card.

As a college student, I was lucky enough to get a job as a sort of intern at Strategic Air Command headquarters in Omaha. In those days all computing was on cards. My

phone/WhatsApp: +86 185 7557 8796
T (346) 5861369

cards got sent down several floors underground at SAC HQ and run on the computer that would run the war, if we ever had one. I was lucky to get one or two runs a day.

As soon as my security clearance came through, I would go down to the computer room in the middle of the night. I would sweet-talk Sergeant Whittaker into letting me run my own programs, sitting at the console of the machine—yes, the machine whose main job was to launch a nuclear attack. Rest easy, though: The red button was not in that room.

Working hands-on with the machine, I got ten times as much work done as when I had to wait for my cards to be taken down and my listings to be brought back up. Feedback came faster, I learned faster, and my programs worked sooner.

That's what Scrum is about. Instead of waiting months or even years to find out what the programmers are doing, in Scrum we find out every couple of weeks. A Scrum product owner with a really good team will be seeing actual features taking shape every few days!

And that is what Kenny's book is about. If you're new to Scrum, read it through from beginning to end. Then keep it nearby. If you've been doing Scrum for a while, scan it, then keep it nearby.

When you find yourself thinking about something that's happening to your team, or wondering about different things to try, pick up this book and look around. Chances are you'll find something of value.

—Ron Jeffries

Phone/WhatsApp: +86 185 1567 8796
+1 (346) 638-0369

PREFACE

This book discusses Essential Scrum—the things you have to know if you're going to be successful when using Scrum to develop innovative products and services.

What Is Essential Scrum?

Scrum is based on a small set of core **values**, **principles**, and **practices** (collectively the **Scrum framework**). Organizations using Scrum should embrace the Scrum framework in its entirety, perhaps not through the entire organization all at once, but certainly within the initial teams that will use Scrum. Embracing all of Scrum does not mean, however, that organizations must implement Scrum according to some cookie-cutter, one-size-fits-all formula. Rather, it means that organizations should always stay true to the Scrum framework while choosing an appropriate blend of **approaches** for their Scrum implementations.

Essential Scrum combines the values, principles, and practices of Scrum with a set of tried-and-true approaches that are consistent with, but not mandated by, the Scrum framework. Some of these approaches will be appropriate to your situation; others will not. Any approach will need to be inspected and adapted to your unique circumstances.

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 653-0369

Origins of This Book

As an agile/Scrum coach and trainer, I am frequently asked for a reference book for Scrum—one that provides a comprehensive overview of the Scrum framework and also presents the most popular approaches for applying Scrum. Because I have been unable to find a single book that covers these topics at a level deep enough to be useful to today's practitioners, I found myself recommending a collection of books: a few that discuss the Scrum framework but are out of date or incomplete; several highly regarded agile books that do not focus solely on Scrum; and a handful that are focused on a specific aspect of Scrum or a specific approach but do not cover the full Scrum framework in depth. That's a lot of books for someone who just wants a single, stand-alone resource that covers the essentials of Scrum!

The originators of Scrum (Jeff Sutherland and Ken Schwaber) do have a Scrum-specific publication called "The Scrum Guide." This short document (about 15 pages) is described by its authors as the "definitive rule book of Scrum and the

documentation of Scrum itself" (Schwaber and Sutherland 2011). They equate their document to the rules of the game of chess, "describing how the pieces move, how turns are taken, what is a win, and so on." Although useful as a Scrum overview or rule book, "The Scrum Guide" is by design not intended to be a comprehensive source of essential Scrum knowledge. Extending the authors' analogy, giving a new Scrum team just "The Scrum Guide" and expecting good results would be like giving a new chess player a 15-page description of the rules of chess and expecting her to be able to play a reasonable game of chess after reading it. It just isn't a stand-alone resource.

This book, *Essential Scrum,* is an attempt to be the missing single source for essential Scrum knowledge. It includes an in-depth discussion of Scrum's principles, values, and practices—one that in most cases agrees with other agile thought leaders and "The Scrum Guide." (Where this book offers a different perspective from what is widely promoted elsewhere, I point it out and explain why.) This book also describes approaches that are consistent with the Scrum framework and that have been used successfully by me and teams I have coached. I did not intend for this book to replace other books that provide a deep vertical treatment of a given Scrum practice or approach. Such books are complementary to and extend this book. Rather, think of *Essential Scrum* as the starting point on the journey of using Scrum to delight customers.

Phone/WhatsApp: +86 185 1467 8796
+1 (346) 638-0369

Intended Audience

For the many thousands of people who have taken my Working on a Scrum Team, Certified ScrumMaster, and Certified Scrum Product Owner classes, and the many teams I have coached, this book will refresh and perhaps even clarify topics we have already discussed. And for the even larger number of people with whom I have not yet had the pleasure of working, this book will either be your first introduction to Scrum and agile or it will be a chance to look at Scrum in a different light and perhaps even improve how you perform Scrum.

I did not write this book for any one specific role—this is not a book specifically for product owners, or ScrumMasters, or members of the development team. Instead, it is a book intended to give everyone involved with Scrum, from all the members of the Scrum team to those with whom they interact in the organization, a common understanding of Scrum based on a core set of concepts with a clear vocabulary for discussing them. With this shared foundation my hope is that your organization will be in a better position to successfully use Scrum to deliver business value.

I imagine that every Scrum team member would have a copy of this book on her desk open to a chapter relevant to the work at hand. I also envision managers at all levels of the organization reading it to understand why Scrum can be an effective approach for managing work and to understand the type of organizational change that may be necessary to successfully implement Scrum. Organizations using or

planning to use an agile approach other than Scrum will also find the information relevant and helpful to their specific agile adoption.

Organization of This Book

This book begins with a brief introduction to Scrum (Chapter 1) and concludes with a discussion of where you might go next (Chapter 23). The remaining chapters are organized into four parts:

- Part I—Core Concepts (Chapters 2–8): Scrum framework, agile principles, sprints, requirements and user stories, product backlog, estimating and velocity, and technical debt
- Part II—Roles (Chapters 9–13): product owner, ScrumMaster, development team, Scrum team structures, and managers
- Part III—Planning (Chapters 14–18): Scrum planning principles, multilevel planning, portfolio planning, envisioning/product planning, and release planning
- Part IV—Sprinting (Chapters 19–22): sprint planning, sprint execution, sprint review, and sprint retrospective

How to Use This Book

As you would expect, I wrote the book assuming that most people would read it linearly from front to back. If you are new or newer to Scrum, you should take this approach because the chapters do tend to build on one another. That being said, if you are looking for one place to get an end-to-end overview of the Scrum framework (a highly visual Scrum primer), read and reference Chapter 2.

For those who are more familiar with Scrum, you can use this book as a Scrum reference guide. If you're interested in sprint retrospectives, jump directly to Chapter 22. If you are interested in exploring the nuances of the product backlog, jump directly to Chapter 6. I highly recommend, however, that everyone, even those familiar with Scrum, read Chapter 3 in its entirety. The principles laid out there form the foundation of the Scrum framework and the rest of the book. It is not simply a restatement of the values and principles of the Agile Manifesto (Beck et al. 2001) that is common in many other written descriptions of Scrum.

Visual Icon Language

I am proud to include in this book a new visual language for describing Scrum. This language is composed from a vocabulary of icons that have been designed to capture essential Scrum roles, artifacts, and activities. This visual Scrum language is an

Phone: What's App: +86 185 7507 8706
TL (346) 538-0369

effective way to communicate concepts and improves the overall shared understandability of Scrum. If you are interested in obtaining and using the new full-color visual Scrum language art (this book is printed in two colors), visit www.innolution.com for details. This website will also host a variety of resources and discussions related to the book.

Let's Get Started

So, whatever your role, whatever your situation, you have picked up this book for a reason. Spend a little time getting to know Scrum. In the pages that follow you just might find a powerful framework that you can make your own, allowing you to substantially improve the way you develop and deliver products and services to delight your customers.

Phone/WhatsApp: +86 185 7567 9790
+1 (346) 638-0369

Acknowledgments

This book would not have been possible without the input of many people, including the thousands of people who have attended my agile-related classes and coaching sessions. By mentioning some people by name, I run the risk of failing to mention others. To those whose names I fail to mention, please know that all of our discussions and email exchanges have been invaluable to me and have definitely influenced this book!

There are three people in particular I would like to thank: Mike Cohn, Rebecca Traeger, and Jeff Schaich. Without the unique involvement of each, this book would be a mere shadow of itself.

Mike Cohn has been a friend and colleague since we first worked together at Genomica in 2000. He was gracious enough to include my book in the Mike Cohn Signature Series; by being affiliated with Mike and the other prestigious authors in that book series, "I look good by the company that I keep," as my parents would say. Mike was my go-to person whenever I wanted to bounce around ideas or discuss book strategies. He always made time in his insane schedule to review each chapter and give me his thoughtful feedback. Working with Mike all these years has been a very rewarding experience—one that I hope will continue long into the future.

Rebecca Traeger has been my personal editor on this book. We have worked together since my days as managing director of the Scrum Alliance in 2007. At that time Rebecca was the editor of the Scrum Alliance website and through that work (and much more since) became the industry's foremost editor on agile-related materials. Early on in writing this book I reached out to Rebecca and asked if she would work with me again, and to my good fortune, she agreed. Nobody saw any chapter unless Rebecca had seen it first. At times her feedback would make me blush, because she frequently improved how I said something, making it sound both more understandable and approachable. If you just love a section of this book, you can be sure Rebecca had her hands in it. If you don't, I probably foolishly chose to ignore her recommendations.

Jeff Schaich is an artist/designer extraordinaire. We have worked on so many different art projects that I can't recall them all. Early on in the formulation of this book I wanted to create an agile/Scrum icon vocabulary to use as the basis for my training presentations and many of the over 200 figures in the book. I knew that I needed a great designer to pull off this feat. Jeff agreed to take on the challenge. There are times when this book seemed like two different projects—writing the content and creating

Phone/WhatsApp: +86 185 7567 8799
+1 (346) 6320569

the artistic concepts. I'm honestly not sure which took more time. I am sure, however, that without Jeff's artistic input, this book would have suffered immeasurably.

I am deeply honored to have both Mike Cohn and Ron Jeffries, two luminaries in the agile community, write forewords for the book! In their own unique ways each has done a great job of properly placing the book in context and opening the door for a discussion of Essential Scrum. Also, Mike, stop eating at Burger King, and Ron, thanks for not pushing the red button!

I'd also like to thank the many people who took time out of their busy schedules to review chapters and send me their feedback. Let me start by mentioning reviewers who provided extensive feedback: Joe Balistrieri, Johannes Brodwall, Leyna Cotran, Martine Devos, Scott Duncan, Ilan Goldstein, John Hebley, Geir Hedemark, James Kovacs, Lauri Mackinnon, Robert Maksimchuk, and Kevin Tureski.

In addition, I would like to thank other reviewers who provided excellent feedback on select chapters: Lyssa Adkins, John Bauer, Sameer Bendre, Susan Briscoe, Pawel Brodzinski, Rowan Bunning, Josh Chappell, Lisa Crispin, Ward Cunningham, Cornelius Engelbrecht, Julia Frazier, Brindusa Gabur, Caroline Gordon, Drew Jemilo, Mike Klimkosky, Tom Langerhorst, Bjarne Larsen, Dean Leffingwell, Maurice le Rutte, David Luzquiños, Lv Yi, Shay McAulay, Armond Mehrabian, Sheriff Mohamed, Cuan Mulligan, Greg Pease, Roman Pichler, Jacopo Romei, Jens Schauder, Bill Schroeder, Yves Stalgies, Branko Stojaković, Howard Sublett, Julie Sylvain, Kevin Tambascio, Stephen Wolfram, and Michael Wollin.

I would also like to thank the staff at Pearson who were great partners in this project. They tolerated my delays with patience and always offered encouragement. Special thanks to Chris Guzikowski, who oversaw the whole thing from soup to nuts. He was there from my first Pearson meeting at a pub in Lexington, MA, through the final production. I would also like to thank Olivia Basegio for adeptly handling logistics and Julie Nahil who did a fantastic job overseeing the project. In addition, thanks to Barbara Wood for the great job of helping polish the manuscript and Gail Cocker for pulling all of the art together into a coherent and beautiful whole.

I am also grateful to my assistant, Lindsey Kalicki, to whom I was able to offload many important tasks so that I could stay focused on book development. I am lucky to be able to work with such a skilled professional.

Most of all, I would like to acknowledge my family—Jenine, Jonah, and Asher—and the critical role that they played. I have asked so very much from them during the long effort of creating this book. No amount of gratitude can make up for the family pressure it caused and our lost time together.

Jenine is my loving soulmate and has stuck by me through all of the ups and downs of writing this book. The sacrifices she made so that I could write would double the size of this book if I tried to list them all. I couldn't have done it without her!

Funny thing is, a year after we were married in 1993, I published my first book, *Succeeding with Objects*. At that time Jenine made me promise that I would never write another book again. Luckily for me, after 15 years memories fade and the

crushing workload doesn't seem as bad in hindsight, so when she urged me to write this one I was surprised to say the least! She hasn't yet told me I can't do book number three, but I suspect it might be 15 more years before the memory of this one fades enough for either of us to want me to write another one!

I also deeply appreciate the loving support from my sons, Jonah and Asher. They gave up time with their dad so that I could write. They were always there to bounce around ideas and to give input on the book. A number of their content and art suggestions have made their way into the book—and it's better because of them! I hope they learned the value of perseverance and that even the most daunting work can be completed if you take it a step at a time and don't give up.

Finally, I would like to acknowledge my mom, Joyce Rubin (Genesha Esther bat Avrahm), for all of the love and support she gave me. Without her influence this book would never have been possible. Sadly, she did not survive to see its publication. Her passing in January 2012 left a void in my life and the lives of her family that can never be filled. She was a very special person to the many whose lives she touched. Mom, I miss you more than I can possibly express.

Phone/WhatsApp: +86 185 7467 8796
+1 (346) 638-0369

This page intentionally left blank

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 638-0369

About the Author

Kenny Rubin provides Scrum and agile training and coaching to help companies develop products in an effective and economically sensible way. A Certified Scrum Trainer, Kenny has trained over 18,000 people on agile and Scrum, Smalltalk development, managing object-oriented projects, and transition management. He has coached over 200 companies, ranging from start-ups to Fortune 10.

Kenny was the first Managing Director of the worldwide Scrum Alliance, a nonprofit organization focused on the successful adoption of Scrum. In addition to this book, Kenny is also the coauthor of the 1995 book *Succeeding with Objects: Decision Frameworks for Project Management*. He received his B.S. in Information and Computer Science from the Georgia Institute of Technology and his M.S. in Computer Science from Stanford University.

Kenny's background is rooted in the object-oriented technology community. He started as a Smalltalk developer on a NASA-funded project back in 1985 and developed the first blackboard expert system outside of LISP. In 1988 he was fortunate to join ParcPlace Systems, a start-up company formed as a Xerox PARC spin-off, whose charter was to bring object-oriented technology out of the research labs and release it to the world. As a Smalltalk development consultant with many different organizations in the late 1980s and throughout the 1990s, Kenny was an early adopter of agile practices. His first use of Scrum was in 2000 for developing bioinformatics software.

In the course of his career, Kenny has held many roles, including successful stints as a Scrum product owner, ScrumMaster, and member of development teams. In addition, he has held numerous executive management roles: CEO, COO, VP of Engineering, VP of Product Management, and VP of Professional Services. He has also overseen the development of five commercial software product suites, generating over $200M in aggregate revenue. In addition, he has been directly involved in raising over $150M in venture capital funding and assisted in taking two companies public on the NASDAQ.

His multifaceted background gives Kenny the ability to understand (and explain) Scrum and its implications equally well from multiple perspectives: from the development team to the executive board.

Prof. Wolfgang Pohl: +49 (0) 2302 91466-18796
+1 (310) 630-0569

This page intentionally left blank

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 638-0369

Chapter 1
INTRODUCTION

On June 21, 2000, I was employed as Executive Vice President at Genomica, a bio-informatics company in Boulder, Colorado. I remember the date because my son Asher was born at one o'clock that morning.

His birth was a good start to the day. Asher was actually born on his predicted due date (in the United States this happens about 5% of the time). So we (really my wife, Jenine) had finished our nine-month "project" on schedule. And to top things off, Asher had a very high Apgar score, indicating that we had produced a healthy, good-quality result! Our biggest stakeholder, our older son, Jonah, was thrilled to have a younger brother. On time, high quality, and delighted stakeholders—it truly was a good day!

After a brief nap, I checked email and saw that the CEO of Genomica had sent an urgent message asking me to be at a board of directors' meeting at 8:00 a.m. that same day. Begrudgingly, I left the hospital and went to the meeting.

When I arrived, I was told that the VP of Engineering had been fired the night before and I had now inherited the 90-person engineering team. I wasn't surprised. For several months the executive team and board had been discussing Genomica's inability to deliver valuable products on time and with acceptable quality, and the VP of Engineering was at the center of that discussion.

It was now my responsibility to oversee the effort of substantially improving the results of our product development organization. I remember being struck by the irony of that day's successful delivery and my new responsibilities.

Because I was already quite busy overseeing sales and marketing, I was told that at my discretion I could hire a new VP of Engineering to report to me. The person I chose to hire was Mike Cohn (Cohn 2004; Cohn 2006; Cohn 2010), and Scrum was the approach that we decided to use.

What Is Scrum?

Scrum is an **agile** approach for developing innovative products and services. Figure 1.1 shows a simple, generic, agile development approach.

With an agile approach, you begin by creating a **product backlog**—a prioritized list of the features and other capabilities needed to develop a successful product. Guided by the product backlog, you always work on the most important or highest-priority items first. When you run out of resources (such as time), any work that didn't get completed will be of lower priority than the completed work.

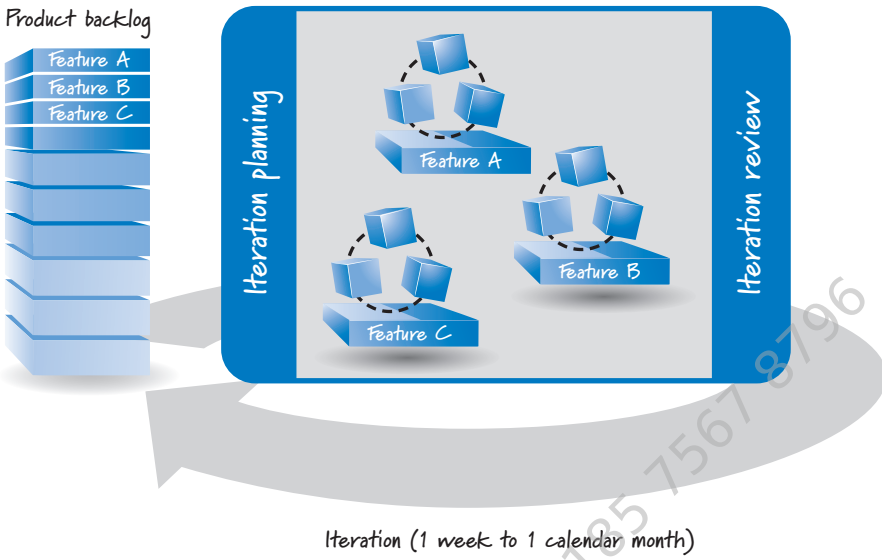


FIGURE 1.1 Agile development overview

The work itself is performed in short, timeboxed **iterations**, which usually range from a week to a calendar month in length. During each iteration, a self-organizing, **cross-functional team** does all of the work—such as designing, building, and testing—required to produce completed, working features that could be put into production.

Typically the amount of work in the product backlog is much greater than can be completed by a team in one short-duration iteration. So, at the start of each iteration, the team plans which high-priority subset of the product backlog to create in the upcoming iteration. In Figure 1.1, for example, the team has agreed that it can create features A, B, and C.

At the end of the iteration, the team reviews the completed features with the stakeholders to get their feedback. Based on the feedback, the product owner and team can alter both what they plan to work on next and how the team plans to do the work. For example, if the stakeholders see a completed feature and then realize that another feature that they never considered must also be included in the product, the product owner can simply create a new item representing that feature and insert it into the product backlog in the correct order to be worked on in a future iteration.

At the end of each iteration, the team should have a potentially shippable product (or increment of the product), one that can be released if appropriate. If releasing after each iteration isn't appropriate, a set of features from multiple iterations can be released together.

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 638 0369

As each iteration ends, the whole process is begun anew with the planning of the next iteration.

Scrum Origins

Scrum's rich history can be traced back to a 1986 *Harvard Business Review* article, "The New New Product Development Game" (Takeuchi and Nonaka 1986). This article describes how companies such as Honda, Canon, and Fuji-Xerox produced world-class results using a scalable, team-based approach to **all-at-once product development**. It also emphasizes the importance of empowered, self-organizing teams and outlines management's role in the development process.

The 1986 article was influential in weaving together many of the concepts that gave rise to what today we call Scrum. Scrum is not an acronym, but rather a term borrowed from the sport of rugby, where it refers to a way of restarting a game after an accidental infringement or when the ball has gone out of play. Even if you are not a rugby aficionado, you have probably seen a scrum where the two sets of forwards mass together around the ball with locked arms and, with their heads down, struggle to gain possession of the ball.

Takeuchi and Nonaka used the metaphors of rugby and the scrum to describe product development:

> *The . . . "relay race" approach to product development . . . may conflict with the goals of maximum speed and flexibility. Instead a holistic or "rugby" approach—where a team tries to go the distance as a unit, passing the ball back and forth—may better serve today's competitive requirements.*

In 1993, Jeff Sutherland and his team at Easel Corporation created the Scrum process for use on a software development effort by combining concepts from the 1986 article with concepts from object-oriented development, empirical process control, iterative and incremental development, software process and productivity research, and complex adaptive systems. In 1995, Ken Schwaber published the first paper on Scrum at OOPSLA 1995 (Schwaber 1995). Since then, Schwaber and Sutherland, together and separately, have produced several Scrum-specific publications, including *Agile Software Development with Scrum* (Schwaber and Beedle 2001), *Agile Project Management with Scrum* (Schwaber 2004), and "The Scrum Guide" (Schwaber and Sutherland 2011).

Though Scrum is most commonly used to develop software products, the core values and principles of Scrum can and are being used to develop different types of products or to organize the **flow** of various types of work. For example, I have worked with organizations that have successfully used Scrum for organizing and managing the work associated with hardware development, marketing programs, and sales initiatives.

Phone/WhatsApp: +86 185 7507 8790
+1 (346) 038-0369

Why Scrum?

So what made an agile approach like Scrum a good choice for Genomica? First, it was clear that Genomica's previous approach to development simply wasn't working. That was the bad news; the good news was that most everyone agreed.

Genomica operated in a complex domain where more was unknown than known. We built products that had never been built before. Our focus was on bleeding-edge, continuously evolving, state-of-the-art, discovery informatics platforms that research scientists would use to help discover the next blockbuster molecule. We needed a way of developing that would allow us to quickly explore new ideas and approaches and learn fast which solutions were viable and which were not. We had a strategic corporate partner to whom we needed to show working results every few weeks or so to get feedback, because our product had to integrate with its core line of DNA sequencers. This need for rapid exploration and feedback did not mesh well with the detailed, up-front planning we had been doing.

We also wanted to avoid big up-front architecture design. A previous attempt to create a next generation of Genomica's core product had seen the organization spend almost one year doing architecture-only work to create a grand, unified bioinformatics platform. When the first real scientist-facing application was put on top of that architecture, and we finally validated design decisions made many months earlier, it took 42 seconds to tab from one field on the screen to the next field. If you think a typical user is impatient, imagine a molecular biologist with a Ph.D. having to wait 42 seconds! It was a disaster. We needed a different, more balanced approach to design, which included some design up front combined with a healthy dose of emergent, just-in-time design.

We also wanted our teams to be more cross-functional. Historically Genomica operated like most organizations. Development would hand off work to the test teams only after it was fully completed. We now had a desire for all team members to synchronize frequently—daily was the goal. In the past, errors were compounded because important issues were being discussed too late in the development effort. People in different areas weren't communicating frequently enough.

For these reasons, and others, we determined that Scrum would be a good fit for Genomica.

Genomica Results

When we chose to embrace Scrum, it was not well known; the first Scrum book didn't appear until the following year (Schwaber and Beedle 2001). However, we pulled together the available information and did the best we could, which was substantially better than we had done before (see Table 1.1).

From an effort perspective, with Scrum development we required one-tenth the amount of effort (calculated in person-months) compared to our previous use of a

TABLE 1.1 Genomica Scrum Results

Measure	Waterfall	Scrum
Effort	10x	1x
Velocity	1x	7x
Customer satisfaction	Poor	Excellent

plan-driven, **waterfall-style** approach to develop a comparable amount of product functionality. Equally important, the Scrum development progressed at seven times the velocity of the waterfall development, meaning that per unit of time, the Scrum development produced about seven times more valuable features than the waterfall development. Even more compelling was that we delivered the software to our partner in a time frame that met the expectations for the launch of its new hardware platform. This enabled us to reinforce a long-term partnership that substantially increased the shareholder value of Genomica.

Can Scrum Help You?

The Genomica pre-Scrum experience of building features that nobody wanted and delivering those features late and with poor quality is not uncommon. Genomica, like many other organizations, had survived by being no worse than its competitors. I saw the same problems when I first started working in commercial software development in the mid-1980s. And for many, after nearly 30 years, the situation hasn't improved.

Today, if you gathered together your business people and developers and asked them, "Are you happy with the results of our software development efforts?" or "Do you think we deliver good customer value in a timely, economical, and quality manner?" what would they say?

More often than not, the people I meet during my worldwide training and coaching answer both questions with a resounding "No." This is followed by a chorus of "Project failure rate is unacceptably high"; "Deliverables are late"; "Return on investment frequently falls short of expectations"; "Software quality is poor"; "Productivity is embarrassing"; "No one is accountable for outcomes"; "Employee morale is low"; "Employee turnover is too high." Then there's the under-the-breath snicker that accompanies the tongue-in-cheek "There must be a better way."

Yet even with all this discontent, most people seem resigned to the fact that dissatisfaction is just part of the reality of software development. It doesn't have to be.

Organizations that have diligently applied Scrum are experiencing a different reality (see Figure 1.2).

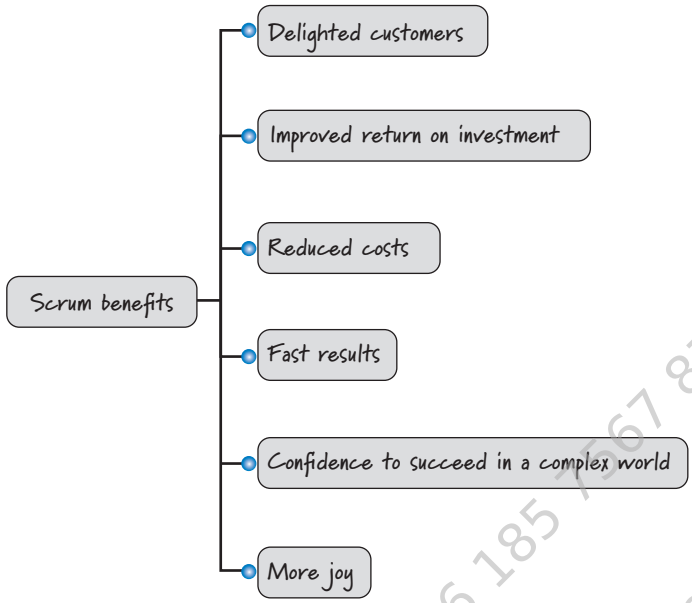


FIGURE 1.2 Scrum benefits

Phone/WhatsApp: +86 185 7567 8796
+1 (346) 638-0369

These organizations are repeatedly delighting their customers by giving them what they really want, not just the features they might have specified on the first day when they knew the least about their true needs. They are also seeing an improved return on investment by delivering smaller, more frequent **releases**. And, by relentlessly exposing organizational dysfunction and waste, these organizations are able to reduce costs.

Scrum's focus on delivering working, integrated, tested, business-valuable features each iteration leads to results being delivered fast. Scrum is also well suited to help organizations succeed in a complex world where they must quickly adapt based on the interconnected actions of competitors, customers, users, regulatory bodies, and other stakeholders. And Scrum provides more joy for all participants. Not only are customers delighted, but also the people doing the work actually enjoy it! They enjoy frequent and meaningful collaboration, leading to improved interpersonal relationships and greater mutual trust among team members.

Don't get me wrong. Though Scrum is an excellent solution for many situations, it is not the proper solution in all circumstances. The **Cynefin** framework (Snowden and Boone 2007) is a sense-making framework that helps us understand the situation in which we have to operate and decide on a situation-appropriate approach. It defines and compares the characteristics of five different domains: **simple**, **complicated**, **chaotic**, **complex**, and a fifth domain, **disorder**, which occurs when you don't