

WITH 40+  
SIMPLE AND POWERFUL  
EXPERIMENTS

# ZOMBIE SCRUM SURVIVAL GUIDE

## A JOURNEY TO RECOVERY



**CHRISTIAAN VERWIJS  
JOHANNES SCHARTAU  
BARRY OVEREEM**

*Forewords by* **DAVE WEST & HENRI LIPMANOWICZ**

The Professional Scrum Series  Scrum.org

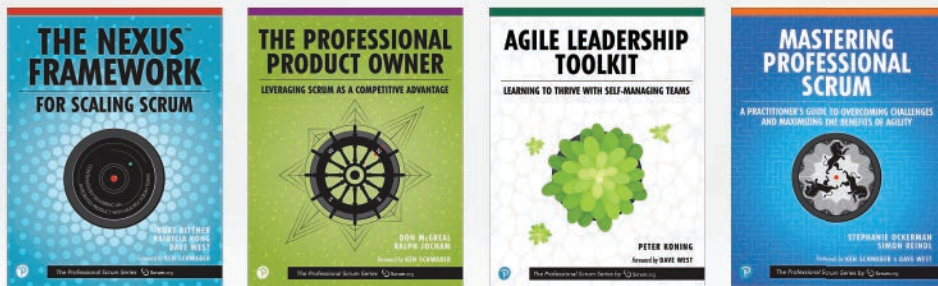
**FREE SAMPLE CHAPTER**

SHARE WITH OTHERS



# **Zombie Scrum Survival Guide**

# The Professional Scrum Series by Scrum.org



Visit [informit.com/scrumorg](http://informit.com/scrumorg) for a complete list of available publications.

**T**he **Professional Scrum Series** from Pearson Addison-Wesley and Scrum.org consists of a series of books that focus on helping individuals and organizations apply Scrum and agile leadership to improve the outcomes of customers and organizations. Approaching the challenge from different perspectives, each book provides deep insights into overcoming the obstacles that both teams and organizations face as they seek to reap the benefits of agility.

All Scrum.org proceeds from the series go to Year Up, an organization whose mission is to close the Opportunity Divide by providing urban young adults with the skills, experience, and support to empower them to reach their potential through professional careers and education.



Make sure to connect with us!  
[informit.com/socialconnect](http://informit.com/socialconnect)

---

# Zombie Scrum Survival Guide

**A JOURNEY TO RECOVERY**

---

Christiaan Verwijs

Johannes Schartau

Barry Overeem

◆◆ Addison-Wesley

Boston • Columbus • New York • San Francisco • Amsterdam • Cape Town  
Dubai • London • Madrid • Milan • Munich • Paris • Montreal • Toronto • Delhi • Mexico City  
São Paulo • Sydney • Hong Kong • Seoul • Singapore • Taipei • Tokyo

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 authors 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.

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at [corpsales@pearsoned.com](mailto:corpsales@pearsoned.com) or (800) 382-3419.

For government sales inquiries, please contact [governmentsales@pearsoned.com](mailto:governmentsales@pearsoned.com).

For questions about sales outside the U.S., please contact [intlcs@pearson.com](mailto:intlcs@pearson.com).

Visit us on the Web: [informit.com/aw](http://informit.com/aw)

Library of Congress Control Number: 2020944524

Cover and interior illustrations by Thea Schukken

Copyright © 2021 Pearson Education, Inc.

All rights reserved. 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. For information regarding permissions, request forms and the appropriate contacts within the Pearson Education Global Rights & Permissions Department, please visit [www.pearson.com/permissions/](http://www.pearson.com/permissions/).

ISBN-13: 978-0-13-652326-0

ISBN-10: 0-13-652326-9

ScoutAutomatedPrintCode

*Zombie Scrum Survival Guide* is dedicated to all the nameless victims  
and unsung heroes in the ongoing struggle against Zombie Scrum.  
We are here to support you.

*This page intentionally left blank*

---

# CONTENTS

---

Foreword by Dave West	xiii
Foreword by Henri Lipmanowicz	xvii
Acknowledgments	xix
About the Authors	xxi
Chapter 1 <b>Getting Started</b>	<b>1</b>
Purpose of This Book	4
Do You Need This Book?	5
How This Book Is Organized	6
No Time to Lose: Off You Go!	8
Chapter 2 <b>First Aid Kit</b>	<b>11</b>
<b>Part I        (Zombie) Scrum</b>	<b>13</b>
Chapter 3 <b>A Primer on Zombie Scrum</b>	<b>15</b>
The State of Scrum	17
Zombie Scrum	18
Symptom 1: Zombie Scrum Teams Don't Know the Needs of Their Stakeholders	19
Symptom 2: Zombie Scrum Teams Don't Ship Fast	20
Symptom 3: Zombie Scrum Teams Don't Improve (Continuously)	21

---



	Symptom 4: Zombie Scrum Teams Don't Self-Organize to Overcome Impediments	23
	It's All Connected	24
	Isn't This Just Cargo Cult Scrum or Dark Scrum?	24
	Is There Hope for Zombie Scrum?	24
	Experiment: Diagnose Your Team Together	25
	Steps	27
	Our Findings	28
	Now What?	29
<b>Chapter 4</b>	<b>The Purpose of Scrum</b>	<b>31</b>
	It's All about Complex Adaptive Problems	32
	Problems	33
	Complex, Adaptive Problems	34
	Complexity, Uncertainty, and Risk	35
	Empiricism and Process Control Theory	36
	Empiricism and the Scrum Framework	37
	What the Scrum Framework Makes Possible	38
	Scrum: An Evolving Set of Minimal Boundaries to Work Empirically	39
	Zombie Scrum and the Efficiency Mindset	40
	What about Simple Problems?	42
	Now What?	44
<b>Part II</b>	<b>Build What Stakeholders Need</b>	<b>45</b>
<b>Chapter 5</b>	<b>Symptoms and Causes</b>	<b>47</b>
	Why Bother Involving Stakeholders?	49
	Who Are the Stakeholders, Actually?	50
	Validating Assumptions about Value	51
	Why Are We Not Involving Stakeholders?	52
	We Don't Really Understand the Purpose of Our Product	52
	We Make Assumptions about What Stakeholders Need	55
	We Create Distance between Developers and Stakeholders	56
	We See Business and IT As Separate Things	59
	We Don't Allow Product Owners to Actually Own the Product	61
	We Measure Output over Value	63

---

	We Believe That Developers Should Only Write Code	64
	We Have Stakeholders Who Don't Want to Be Involved	66
	Healthy Scrum	68
	Who Should Get to Know the Stakeholders?	68
	When to Involve Stakeholders	69
	Now What?	71
<b>Chapter 6</b>	<b>Experiments</b>	<b>73</b>
	Experiments: Getting to Know Your Stakeholders	74
	Start a Stakeholder Treasure Hunt	74
	Create Transparency with the Stakeholder Distance Metric	76
	Give the Stakeholder a Desk Close to the Scrum Team	78
	Decorate the Team Room with the Product Purpose	80
	Experiments: Involving Stakeholders in Product Development	81
	Invite Stakeholders to a "Feedback Party"	81
	Go on a User Safari	84
	Guerrilla Testing	86
	Experiments: Keeping Your Focus on What Is Valuable	88
	Limit the Maximum Length of Your Product Backlog	88
	Map Your Product Backlog on an Ecocycle	90
	Express Desired Outcomes, Not Work to Be Done	94
	Now What?	96
<b>Part III</b>	<b>Ship It Fast</b>	<b>97</b>
<b>Chapter 7</b>	<b>Symptoms and Causes</b>	<b>99</b>
	The Benefits of Shipping Fast	102
	Complexity in Your Environment	102
	Complexity in Your Product	104
	The Bottom Line: Not Shipping Fast Is a Sign of Zombie Scrum	105
	Why Are We Not Shipping Fast Enough?	105
	We Don't Understand How Shipping Fast Reduces Risk	106
	We Are Impeded by Plan-Driven Governance	108
	We Don't Understand the Competitive Advantage of Shipping Fast	110
	We Don't Remove Impediments to Shipping Fast	113
	We Work on Very Large Items during a Sprint	114

---

	Healthy Scrum	116
	Deciding to Release (or Not)	117
	Releasing Is No Longer a Binary Action	118
	Shipping during a Sprint	120
	No More “Big-Bang” Releases	121
	Now What?	121
<b>Chapter 8</b>	<b>Experiments</b>	<b>123</b>
	Experiments to Create Transparency and Urgency	124
	Make a Business Case for Continuous Delivery	124
	Measure Lead and Cycle Times	126
	Measure Stakeholder Satisfaction	129
	Experiments for Starting Shipping More Often	131
	Take the First Steps to Automating Integration and Deployment	131
	Evolve Your Definition of Done	135
	Ship Every Sprint	137
	Ask Powerful Questions to Get Things Done	139
	Experiments for Optimizing Flow	141
	Increase Cross-Functionality with a Skill Matrix	141
	Limit Your Work in Progress	145
	Slice Your Product Backlog Items	148
	Now What?	150
<b>Part IV</b>	<b>Improve Continuously</b>	<b>153</b>
<b>Chapter 9</b>	<b>Symptoms and Causes</b>	<b>155</b>
	Why Bother Improving Continuously?	157
	What Is Continuous Improvement?	158
	Continuous Improvement or Agile Transformation?	161
	Why Are We Not Improving Continuously?	163
	In Zombie Scrum, We Don’t Value Mistakes	163
	In Zombie Scrum, We Don’t Have Tangible Improvements	166
	In Zombie Scrum, We Don’t Create Safety to Fail	168
	In Zombie Scrum, We Don’t Celebrate Success	171
	In Zombie Scrum, We Don’t Recognize the Human Factors of Work	172
	In Zombie Scrum, We Don’t Critique How We Do Our Work	175

---

	In Zombie Scrum, We Consider Learning and Work As Different Things	177
	Healthy Scrum	179
	Self-Critical Teams	181
	See the Forest and the Trees, Together	181
	Now What?	182
<b>Chapter 10</b>	<b>Experiments</b>	<b>183</b>
	Experiments for Encouraging Deep Learning	183
	Share an Impediment Newsletter throughout the Organization	184
	Ask Powerful Questions during Sprint Retrospectives	185
	Dig Deeper into Problems and Potential Solutions, Together	187
	Experiments for Making Improvements Tangible	190
	Create 15% Solutions	190
	Focus on What to Stop Doing	191
	Create Improvement Recipes	193
	Experiments for Gathering New Information	195
	Use Formal and Informal Networks to Drive Change	195
	Create a Low-Tech Metrics Dashboard to Track Outcomes	198
	Experiments to Create a Learning Environment	200
	Share Success Stories and Build on What Made Them Possible	200
	Bake a Release Pie	202
	Now What?	204
<b>Part V</b>	<b>Self-Organize</b>	<b>205</b>
<b>Chapter 11</b>	<b>Symptoms and Causes</b>	<b>207</b>
	Why Bother Self-Organizing?	209
	What Is Self-Organization?	210
	Self-Organization through Simple Rules	211
	Self-Organization through Self-Management	212
	Self-Organization Is a Survival Skill in a Complex World	214
	The Bottom Line	217
	Why Are We Not Self-Organizing?	217
	In Zombie Scrum, We Are Not Self-Managing Enough	218
	In Zombie Scrum, We Use Off-the-Shelf Solutions	220
	In Zombie Scrum, Scrum Masters Keep Resolving All Impediments	223

---

	In Zombie Scrum, Scrum Masters Focus Only on Scrum Team(s)	225
	In Zombie Scrum, We Have No Goals or They Are Imposed	227
	In Zombie Scrum, We Don't Use the Environment As External Memory	229
	In Zombie Scrum, We Are Impeded by Standardization	232
	Healthy Scrum: What Self-Organization Looks Like	234
	Scrum Teams Have Product Autonomy	234
	Management Supports Scrum Teams	237
	Now What?	238
<b>Chapter 12</b>	<b>Experiments</b>	<b>239</b>
	Experiments to Increase Autonomy	239
	Make the Cost of Low Autonomy Transparent with Permission Tokens	240
	Find Actions That Boost Both Integration and Autonomy	242
	Break the Rules!	245
	Experiments to Encourage Self-Organization	246
	Find a Minimum Set of Rules for Self-Organization	247
	Express Clear Requests for Help	249
	Observe What Is Happening	251
	Experiments to Promote Self-Alignment	254
	Create Better Sprint Goals with Powerful Questions	254
	Use a Physical Scrum Board	256
	Find Local Solutions	259
	Organize Scrum Master Impediment Gatherings	259
	Develop Local Solutions with Open Space Technology	261
	Now What?	263
<b>Chapter 13</b>	<b>The Road to Recovery</b>	<b>265</b>
	A Global Movement	266
	What If Nothing Helps?	267
	More Resources	268
	Closing Words	268
<b>Index</b>		<b>271</b>

---

# FOREWORD BY DAVE WEST

---

Scrum is cited by analysts and the press as the most widely used agile framework, with potentially millions of people applying it every day. To prove its impact, just wear a T-shirt with *Scrum* written on it and walk through an airport. People will stop you and ask you questions about Scrum and if you can help them do *x* or *y*. But there are many people using Scrum and not getting the most out of it. They are, as Christiaan, Johannes, and Barry describe, acting like zombies, mindlessly using the Scrum artifacts, events, and roles but not really getting the benefits from it.

But there is hope! The Zombie Scrum infection can be cured, with focus and perseverance. Christiaan, Johannes, and Barry have written this excellent *survival guide* to help teams and organizations improve their use of Scrum to achieve better results. It is a perfect complement to the other titles in The Professional Scrum Series, all of which focus on helping to improve the ability of Scrum Teams to deliver value in a complex and sometimes chaotic world.

Professional Scrum, the antithesis of Zombie Scrum, consists of two elements. First is Scrum, which is of course the framework as described in the Scrum Guide, but also the foundations that framework is based upon. Those foundations are empirical process; empowered, self-managed teams; and a

focus on continuous improvement. Surrounding the framework and its ideas are four additional elements:

- **Discipline.** To be effective with Scrum requires discipline. You have to deliver to gain learning; you have to do the mechanics of Scrum; you have to challenge your preconceived ideas about your skills, role, and understanding of the problem; and you have to work in a transparent and structured way. Discipline is hard and may at times seem unfair as your work exposes problem after problem and your efforts seem in vain.
- **Behaviors.** The Scrum values were introduced to the Scrum Guide in 2016 in response to the need for a supporting culture for Scrum to be successful. The Scrum values describe five simple ideas that when practiced encourage an agile culture. Courage, focus, commitment, respect, and openness describe behaviors that both Scrum Teams and the organizations they work within should exhibit.
- **Value.** Scrum Teams work on problems that deliver value to stakeholders when they are solved. Teams work for a customer who rewards them for that work. But the relationship is complex because the problems are complex; the customer might not know what they want, or the economics of the solution might also be unclear, or the quality and safety of the solution may also be unknown. The job of a professional Scrum Team is, to the best of their ability, to do the right thing for all these parties by delivering a solution that best meets their customers' needs within the constraints that have been placed on them. That requires transparency, respect for each other and for customers, and a healthy curiosity to uncover the truth.
- **Active Community Membership.** Scrum is a team sport where the team is small. That means that the team is often the underdog trying to solve problems that it barely has the skills and experience to solve. To be effective professional Scrum Teams must work with other members of their community to learn new skills and share experiences. Helping to scale the agility of the community is not completely altruistic, because the helper often learns valuable things that they can bring back to help their own team. Professional Scrum encourages people to form professional networks in which ideas and experiences that help teams can be exchanged.

Professional Scrum and Zombie Scrum are two mortal enemies in eternal locked combat. If you relax your guard for a moment, Zombie Scrum comes back. In this book Christiaan, Johannes, and Barry describe a guide for how to stay on your guard, providing practical tips for both identifying when you have become a zombie and how to stop this happening. Their humorous and very visual material is a must-have for any Zombie Scrum hunter.

Good luck in fighting Zombie Scrum!

—Dave West  
CEO, Scrum.org



*This page intentionally left blank*

---

# FOREWORD BY HENRI LIPMANOWICZ

---

Scrum is an excellent framework, but—there’s always a *but*, isn’t there?!—its users and practitioners are, like everyone, imperfect, diverse, and unpredictable. They will show up as they are, quiet or talkative, hesitant or interrupting, reckless or cautious, linear or creative, bossy or timid. And all, Scrum Masters included, bring reflexive habits of what they do routinely while working in a group. In other words, all the people ingredients that can make ordinary meetings dysfunctional are present in Scrum Events. This is why Scrum practitioners must be prepared to fortify the framework with suitable techniques to ensure that every event delivers its full potential, regardless of the personalities in the room. In short, every Scrum Event must be facilitated well enough to be productive, engaging, rewarding, and enjoyable.

Liberating Structures are ideal fortifiers of Scrum because they perfectly complement it. First, they are easy to use, flexible, efficient, and effective. Second and most important, Liberating Structures ensure that every participant is actively engaged and contributes. This makes the Scrum Events both productive and rewarding for all.

As Scrum Teams learn how to use a few Liberating Structures, they acquire tools that are universally and routinely useful in all manner of situations at work or outside work. For example, a simple “1-2-4-All” or “Impromptu Networking” can engage groups in deeper thinking during a Sprint Review, Sprint Planning, or Sprint Retrospective. “Min Specs” or “Ecocycle Planning” can help Product Owners work with stakeholders to order the Product Backlog. And structures like “Conversation Cafe,” “Troika Consulting,” and “Wise Crowds” can be used to navigate complex challenges and concerns and build trust. Throughout this book, you’ll notice many examples of how Scrum Teams can use Liberating Structures to overcome Zombie Scrum.

Barry, Christiaan, and Johannes have done a magnificent job of accumulating successful experiences and sharing their inspiring stories in this very practical book. They don’t shy away from telling it like it is, which is why their proposals are always useful, as they are grounded in reality.

—Henri Lipmanowicz  
Cofounder, Liberating Structures

---

# ACKNOWLEDGMENTS

---

Although this book has only three authors listed on the title page, it was made possible by a significantly larger group. We want to start by thanking Dave West, Kurt Bittner, and Sabrina Love from Scrum.org for their support, encouragement, and trust in this book about *Zombie Scrum*. Kurt Bittner, in particular, deserves a deep bow for his repeated reviews of our initially long-winded chapters. Like a Product Owner, he helped us focus on what mattered most and say “no” (even when it hurt) to the rest.

We also want to thank the team from Pearson, Haze Humbert, Tracy Brown, Sheri Replin, Menka Mehta, Christopher Keane, Vaishnavi Venkatesan, and Julie Nahil, for their time and effort. And for the trust they put in us when we suggested writing, reviewing, and editing the book in a more incremental fashion than what is customary in the publishing world. Another group that deserves a deep bow are the Scrum Masters who reviewed this book and supported us with their thorough feedback: Ton Sweep, Thomas Vitzky, Saskia Vermeer-Ooms, Tom Suter, Christian Hofstetter, Chris Davies, Graeme Robinson, Tábata P. Renteria, Sjors de Valk, Carsten Grønberg Lützen, Yury Zaryaninoy, and Simon Flossman. This book is much, much better because of you.

One person who made this book come alive in particular is Thea Schukken. She created all the beautiful, clever, and funny illustrations in this book and adds a much-needed visual perspective. And then there are all the reviewers in the community who helped us with their feedback and suggestions when we posted tidbits of it on our blogs.

Our work and thinking stand on the shoulders of giants. First of all, there are Ken Schwaber and Jeff Sutherland, the creators of the Scrum Framework. Their work changed our lives and those of many others. The same goes for Keith McCandless and Henri Lipmanowicz, who collected and invented Liberating Structures as a way to unleash and include everyone in groups of any size. Others who shaped and guided our work are Gunther Verheyen, Gareth Morgan, Thomas Friedman, and many Professional Scrum Trainers and stewards of Scrum.org.

Other shoulders we've stood on are those of our partners, Gerdien, Fiona, and Lianne, as well as our families. They supported us throughout as we had to withdraw yet another evening into our home offices to write this book.

But the most important acknowledgment is to all the Scrum Masters, Product Owners, and Development Teams out there who are working hard to deliver value to their stakeholders—in particular those who are carrying on despite severe Zombie Scrum. We are indebted to your persistence. This book is here for you.

---

# ABOUT THE AUTHORS

---



**Christiaan Verwijs** is one of the two founders of The Liberators, together with Barry Overeem. The mission of The Liberators is to unleash organizational superpowers with Scrum and Liberating Structures. Somewhere in a dusty drawer, he has degrees in organizational psychology and business information technology. He has over twenty years of experience as a developer, Scrum Master, and trainer and steward for Scrum.org, in both small and large organizations. In those years he has seen his share of severe *Zombie Scrum*, as well as how many of those teams found the road to recovery. Christiaan loves to write (posts and code), read, and play games. There's also a weird fascination with LEGO and squeezing as much of it as possible into his home office. You can follow his writing online at [medium.com/the-liberators](https://medium.com/the-liberators).



**Johannes Schartau** is a consultant, trainer, and coach for agile product development and organizational improvement. His interests in ethnology (with a focus on Amazonian shamanism), psychology, technology, integral thinking, complexity science, and stand-up comedy finally coalesced when he was introduced to Scrum in 2010. Since then he has dedicated

himself to exploring organizations from all possible angles together with the people working in them. His mission is to bring life and meaning back to the workplace by spreading Healthy Agile and Liberating Structures around the world. Aside from his work, he is passionate about cast iron (both in the gym and the kitchen), mixed martial arts, and humor. Being a proud husband and the father of two wicked boys gives his life meaning and beauty.



**Barry Overeem** is the other founder of The Liberators. In line with the mission of The Liberators, Barry liberates organizations from outdated modes of working and learning, using Scrum and Liberating Structures as sources of inspiration. Although becoming a journalist and teacher was his original plan, he ended up with a degree in business administration. He spent the first half of his twenty-plus-year professional journey being an application manager and IT project manager. In 2010, working in software development environments, he started his first experiments with Scrum. In the past ten years, Barry has worked with a wide variety of teams and organizations. Some got stuck with Zombie Scrum; others managed to recover. In 2015, he joined Scrum.org as a trainer and, together with Christiaan, created the Professional Scrum Master II class. When not fighting Zombie Scrum, he enjoys reading and writing, walking long distances, and spending time with his kids, Melandri, Guinnessa, and Fayette.

## ABOUT THE ILLUSTRATOR

**Thea Schukken** is the founder of the company Beeld in Werking. As a visual facilitator, she transforms complex information into simple and attractive illustrations, animations, and infographics. She combines her drawing skills with more than twenty-five years of experience in IT and management. For this book, Thea translated our story into simple, powerful visuals that underscore our message of how to recognize and recover from Zombie Scrum.



Thea Schukken, the founder of Beeld in Werking, created over fifty illustrations for *Zombie Scrum Survival Guide*.



*This page intentionally left blank*



## **EXPERIMENT: DIAGNOSE YOUR TEAM TOGETHER**

Throughout this book, you'll find many experiments and interventions that you can do with your team. They are all designed to help create transparency around what is happening, to allow inspection and encourage adaptation. Every experiment follows a similar pattern. We start with the purpose. Then we explain the steps and give direction on what to watch out for.

This first experiment is all about creating transparency and starting a conversation around Zombie Scrum (see Figure 3.6). This is a critical first step towards recovery and to confront the truth that work is needed. This experiment helps you progress on the first three steps of the First Aid Kit (Chapter 2): take responsibility, assess the situation, and create awareness.

This experiment is based on the Liberating Structure “What, So What, Now What?”<sup>1</sup> It is a good way to build confidence, celebrate small successes, and build the muscle to get through the hard stuff.

### Skill/Impact Ratio

<b>Skill</b>		No skill is required for filling in a survey and inspecting the results together with your team.
<b>Impact on survival</b>		This experiment creates transparency around what is going on in your team (and around it) in terms of Zombie Scrum. It’s a crucial first step on your way to recovery.



**Figure 3.6** Team diagnoses in progress

1. Lipmanowicz, H., and K. McCandless. 2014. *The Surprising Power of Liberating Structures: Simple Rules to Unleash a Culture of Innovation*. Liberating Structures Press. ASN: 978-0615975306.

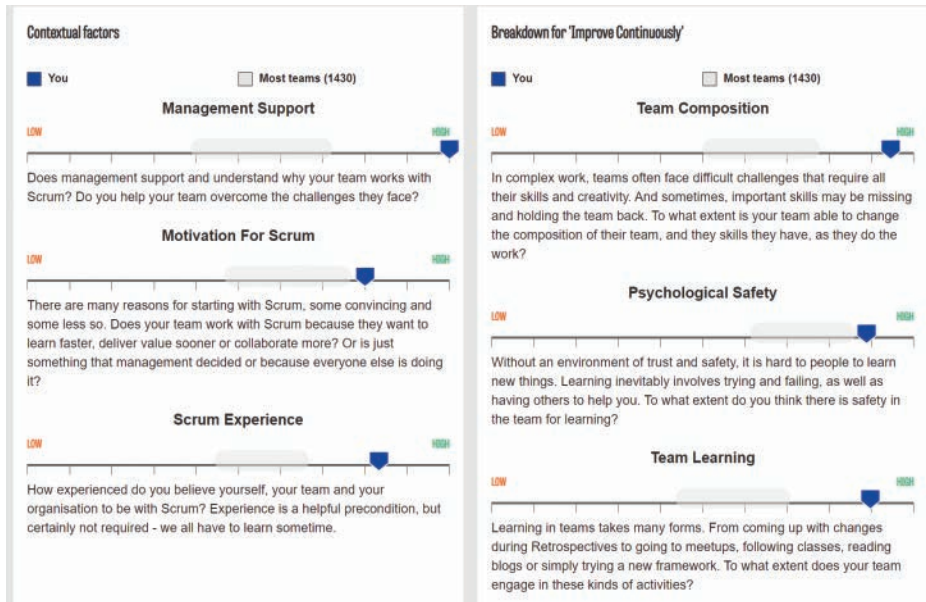
## STEPS

The following steps help you do this experiment:

1. Go to [survey.zombiescrum.org](http://survey.zombiescrum.org) and fill out the extensive free survey for your Scrum Team. Invite others from your team to join your “sample” as instructed. To protect others’ privacy and avoid abuse of the survey, scores from individual members are only shown to each survey taker.
2. When you’ve completed the survey, you’ll receive a detailed report (see Figure 3.7). The report will be updated every time someone joins the sample. In the report, you’ll find results for the four symptoms of Zombie Scrum, as well as a more detailed breakdown. The report also gives feedback and recommendations based on the results.
3. When everyone has participated, schedule a one-hour workshop to inspect the results together. We recommend doing this with only the Scrum Team: the Product Owner, the Scrum Master, and the Development Team.
4. Prepare for the workshop. You can print the report and hand out copies, put prints on the walls, or simply put up the profile on a screen.
5. Start the workshop by reiterating the purpose clearly and emphasizing what will happen with the outcomes (and what won’t). Make sure to emphasize that improvement is always a gradual, incremental, and often messy process and that this workshop is a step in that process.
6. Invite everyone to inspect the results silently and note down observations. Ask: “What do you notice in the results?” Encourage people to stick to the facts, and avoid jumping to conclusions, for the first round. After a few minutes, ask people to share their observations in pairs for another couple of minutes and notice similarities and differences. If you have eight or more people, ask pairs to join another pair and take a few minutes to share observations and notice patterns. Ask the small groups to share their most important insights with the whole group, and capture them in a way that remains visible to everyone present.
7. Following the pattern outlined in the previous step, repeat twice more with different questions. For round two, ask people “So, what does this mean for our work as a team?” For round three, ask people “Where do we

have the freedom and autonomy to improve as a team? What are small, first steps we can commit to?” Make sure to keep capturing the most salient outcomes.

8. Put the most important actionable improvement on the Sprint Backlog for the next Sprint. Involve others as needed to keep making progress.



**Figure 3.7** Part of the report you'll receive after completing the Zombie Scrum Survey

## OUR FINDINGS

- It can be tempting to identify dozens of potential improvements and end up doing nothing at all. Instead, keep a strong focus on improving one thing first before moving onto something else. If that improvement is too big to commit to doing it in a single Sprint, make it smaller.
- When you ask people to participate in this survey, you're asking them to trust you with their honest answers. Be deeply respectful of that. Don't spread reports to people outside of the team or forward them to

management unless you have clear and unambiguous approval from everyone involved.

- Don't use the report to compare teams. Doing so will erode trust much faster than you can rebuild it.

## **INCREASE CROSS-FUNCTIONALITY WITH A SKILL MATRIX**

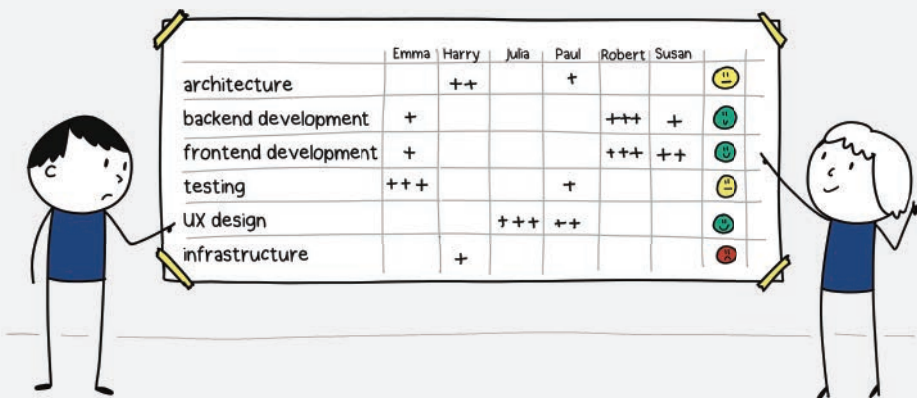
Is your team experiencing bottlenecks because only one person is capable of testing work? Is a developer on your team struggling to implement something that is blocking everyone else until she is done? Do team members start work

---

on unrelated and low-value tasks simply because they have nothing else to do? These symptoms arise when teams are not cross-functional enough, causing work to pile up for some people and creating delays for others.

The Scrum Framework is built on cross-functional teams because they are better able to overcome the unpredictable challenges that arise when working on complex problems. Your team is cross-functional enough when items flow smoothly through your workflow. Cross-functionality does not mean that everyone can perform any kind of task or that you must have at least two experts for every kind of skill on your team. Often, just having another person who has a particular skill, even when they are slower and less experienced at it, already improves flow enough to prevent most problems.

This experiment offers your team practical strategies to help them improve their cross-functionality (see Figure 8.4).



**Figure 8.4** Increase cross-functionality with a skill matrix.

### Effort/Impact Ratio

**Effort**



This experiment aims at one of the toughest causes of Zombie Scrum. You may have to deal with resignation and cynicism.

**Impact on survival**



Finding ways to distribute skills in your team not only improves flow, it is also good for morale.



## Steps

To try this experiment, do the following:

1. With your team, map the skills you need during a typical Sprint. Together, create a matrix on a flip chart where you plot the members of your team against the skills you identified. Invite people to decide for themselves what skills they possess and to self-rate their proficiency with it using plus signs (+, ++, and +++).
2. When you're done with the matrix, ask "What do you notice about how the skills on our team are distributed? What is immediately obvious?" Invite people to reflect on this question individually for two minutes, then for a few minutes in pairs. With the whole group, capture important patterns on a flip chart.
3. Ask "What does this mean for our work as a team? Where should we focus our improvements?" Let people reflect on this question individually, then in pairs for a few minutes, and then capture the biggest insights on the flip.
4. Ask "Where should we start improving? What first step is possible for us without needing approval from others or resources we don't have?" Let people reflect on this individually, then in pairs for a few minutes, and then capture the biggest insights on the flip chart. Use the strategies as described in the next section as inspiration when people struggle to see possibilities.
5. Keep the skill matrix in your team room and update it frequently. You can tie it to flow-based metrics such as throughput and cycle time, which should improve over time as cross-functionality increases. See the experiment "Limit Your Work in Progress" to learn how to do this.

There are many strategies for improving cross-functionality on your team.

- You can add people to your team who already have skills that you need. Although a seemingly obvious solution, adding skilled people isn't always possible. It's also doubtful how structural this solution really is, as it can cause "Skill Whac-A-Mole," where other skills then become bottlenecks and you have to add even more specialized people. Instead of maintaining
-

---

high degrees of skill specialization, it's often more effective to distribute skills.

- You can automate tasks that require scarce skills. For example, creating a backup of a database or deploying a release are critical tasks that are often performed by database specialists and release engineers. When you automate these tasks, you improve not only the speed of the activity, but also how frequently these tasks can be performed, while also removing the constraint.
- You can purposefully limit your team's work in progress, putting constraints on how much new work can be started, to encourage cross-functionality. Instead of starting a new Product Backlog item, because there isn't anything else to do, ask "How can I help others complete their current work?" or "How can others help me complete this work?" The Daily Scrum is a natural opportunity to offer and request help.
- You can encourage people to pair on tasks that only a few people can perform. When you pair experienced and inexperienced people, the less experienced people develop new skills, and both people find better ways to support each other. For example, pairing developers who typically work on the front end with developers who work on the back end makes it easier for them to support each other when bottlenecks occur.
- You can use approaches such as "Specification by Example"<sup>5</sup> to allow customers, developers, and testers to work together to develop automated test cases. In a similar vein, front-end frameworks (e.g., Bootstrap, Material, or Meteor) can make it easier for designers and developers to work together with a common design language for elements.
- You can organize skill workshops where people who are skilled at a particular task demonstrate how they perform it and help others perform it.

## **Our Findings**

- When Scrum Teams have been affected by Zombie Scrum for a long time, they may have come to believe that nothing ever changes. You may even face understandable cynicism. If this is the case, start with the smallest

---

5. Adzic, G. 2011. *Specification by Example: How Successful Teams Deliver the Right Software*. Manning Publications. ISBN: 1617290084.

possible improvements to show people that change is possible and worth the time spent making it happen.

- When the skills of team members are narrowly specialized, they may struggle to see how broadening their skills will benefit the team. They may also fear losing their uniquely visible contribution to the team. Make an effort to celebrate the successes of the team to emphasize the collective outcomes over individual contributions.

## SHARE AN IMPEDIMENT NEWSLETTER THROUGHOUT THE ORGANIZATION

The impediments that make it hard for Scrum Teams to work empirically often involve people across the organization. Helping these people understand the impediments and the problems they cause creates awareness that enables double-loop learning, which can lead to systemic improvements.

### Effort/Impact Ratio

---

**Effort**



This experiment calls for nothing but courage and a dash of tact.

**Impact on survival**



Although painful, this experiment is a great way to create urgency around the biggest problems.

---

### Steps

To try this experiment, do the following:

1. With your Scrum Team, ask everyone to silently write down impediments they see that are making it hard for them to build what stakeholders need or ship fast(er), or both. What skills are missing? Where is protocol getting in the way? Which people do they need, but don't have access to? After a few minutes, invite people to pair up to share and build on their individual ideas. Together, share all impediments and pick the three to five impediments that are most impactful (e.g., with dot-voting).
  2. For the biggest impediments, ask "What is lost because of this? What would we and our stakeholders gain when this impediment is removed?" Capture the consequences for the various impediments.
  3. For the biggest impediments, ask "Where do we need help? What would help look like?" Collect the requests for help for the various impediments.
  4. Compile the biggest impediments, including their consequences and requests for help, in a format that you can easily distribute to everyone who has a stake in your work. It could be a mailing, a paper newsletter, a
-

blog post on your intranet, or a poster that you put in a heavy-traffic corridor. Include the purpose of your team and how to contact you. You can also include the accomplishments of your team, of course.

### **Our Findings**

- Make sure to include (higher) management and consider informing them up front. Also, they will probably appreciate a shorter, more concise version of the newsletter.
- Transparency can be painful. Be honest but tactful in your messaging, and don't blame others or be negative. State what is happening and make clear requests for help.
- If you are planning to do this experiment frequently, make sure to include the accomplishments of your team as well. What is going well? What has changed since the previous newsletter? And most important: from whom did you receive (unexpected) help?

---

## MAKE THE COST OF LOW AUTONOMY TRANSPARENT WITH PERMISSION TOKENS

The autonomy of teams decreases as their dependencies on external people increase. Some dependencies are explicit, such as when a Scrum Team needs someone outside the team to do something for them. Other dependencies are more implicit. Having to ask for permission or approval from someone outside the team in order to proceed is a good example. This experiment is about making transparent where and how often permission is required (see Figure 12.1).



**Figure 12.1** Without considering all the things that constrain Scrum Teams, it's easy to expect miracles from them.

### Effort/Impact Ratio

<b>Effort</b>	☆☆☆☆	This experiment requires only a jar, some tokens, and a few minutes during your Sprint Review.
<b>Impact on survival</b>	☆☆☆☆☆	Even in the most zombified environments, regaining some sense of control makes people sigh with relief.

---

---

## Steps

To try this experiment, do the following:

1. Find an empty jar, or another container, and place it in the team room. Somewhere near the Sprint Backlog is the best spot.
2. Give everyone on your team a bunch of permission tokens. You can use marbles, LEGO bricks, magnets, or stickies. Use different colors for the various permission categories. For example, the permission to release something, to move an item to another column on your Scrum Board, or to change your tools or environment. We recommend a limit of five categories to keep things simple.
3. During the Sprint, put an approval token in the jar every time someone on the Scrum Team has to ask permission from someone outside the team. For example, put a token in the jar when an external architect needs to approve that an item is done. Or when the Product Owner has to vet an item with an external manager. Put a token in the jar when you need permission from office management to purchase stickies. And put a token in the jar when you need a configuration to be changed by an external administrator. Aside from requests for permission, also add a token every time you need someone outside the team to perform a specific action as well.
4. During the Sprint Review, and with stakeholders present, share the number of tokens in the jar. Ask: “How does this affect our ability to quickly adapt in the moment and do what is the most valuable? Where can we simplify things?” Invite people to first consider this question for themselves and in silence, then in pairs for two minutes, and then paired with another pair for four more minutes. Capture the most salient improvements with the whole group. The Sprint Retrospective is a great opportunity for digging into potential improvements.

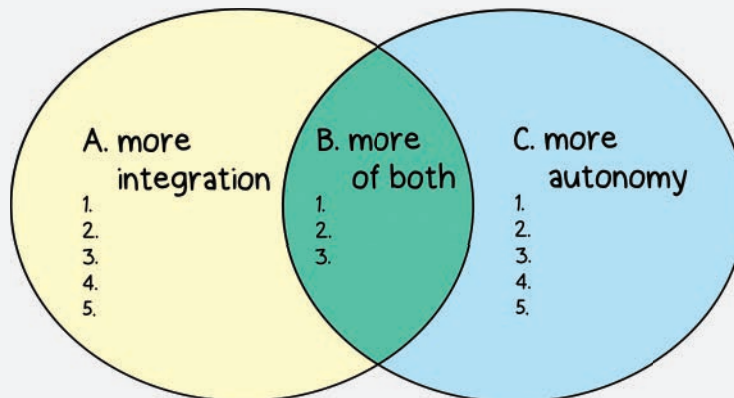
## Our Findings

- For another perspective, you can use different colors for everyone on your team. This allows you to identify who is most often in need of permission.

- 
- If you want to focus on the amount of organizational bureaucracy, don't add permission tokens for requests from direct stakeholders such as customers, users, or people who otherwise invest significant money or time in your product.
  - The experiment "Break the Rules!" elsewhere in this chapter is great to test where asking for permission matters, and where it just gets in the way of doing the right thing.

## FIND ACTIONS THAT BOOST BOTH INTEGRATION AND AUTONOMY

Organizations with self-managing Scrum Teams face the difficult challenge of balancing their autonomy while keeping their work integrated with the rest of the organization. Because both of these aspects are equally desirable, and we can't simply make an either-or decision, we are faced with what is called a "wicked question." Instead of letting the pendulum swing entirely to one side, this experiment is about finding ways of supporting both sides. With this approach, you help groups move from "either-or" to "yes-and" thinking. This experiment and its corresponding worksheet (see Figure 12.2) are based on the Liberating Structure "Integrated~Autonomy."<sup>1</sup>



---

**Figure 12.2** A simple worksheet for Integrated~Autonomy<sup>2</sup>

1. Lipmanowicz, H., and K. McCandless. 2014. *The Surprising Power of Liberating Structures: Simple Rules to Unleash a Culture of Innovation*. Liberating Structures Press. ASN: 978-0615975306.
  2. Source: Lipmanowicz and McCandless, *The Surprising Power of Liberating Structures*.
-



---

## Effort/Impact Ratio

---

**Effort**



This experiment greatly benefits from tight facilitation and asking powerful questions to help the group move out of deadlocks.

---

**Impact on survival**



As people start seeing that autonomy and integration are not opposed, more of both will be possible.

---

## Steps

To try this experiment, do the following:

1. Invite people who have a stake in either increasing the autonomy of Scrum Teams or keeping them integrated with work done elsewhere. This includes the Scrum Teams themselves, departments they depend on (and vice versa), and management.
2. Begin by helping people make tensions between autonomy and integration tangible. Ask “For the Scrum Teams, where in their work is there tension between the desire for autonomy and the desire for integration?” Start with a minute of silent thinking (one minute), then invite people to share their ideas in pairs (two minutes). Capture salient examples from the whole group (five minutes). For example, there can be tension between the autonomy that Scrum Teams have over their Sprint Backlog and the need to be able to pick up urgent issues from people outside the team that emerge during a Sprint. There can be tension between the autonomy of a Product Owner to order the Product Backlog and keeping that ordering aligned with corporate strategy. Or between allowing Scrum Teams to pick their own tools and having mandated tools that are safe for corporate environments.
3. The next step is to explore actions that promote integration. For this step, the participants work with the Integrated~Autonomy worksheet shown in Figure 12.2. It shows three columns with space for writing down ideas that lead to either more integration (A), more autonomy (C), or both (B). The group will focus on column A first. Ask “What actions boost integration of the Scrum Teams’ activities with what is happening

---

elsewhere?” Start with a minute of silent thinking (one minute), then invite people to share their ideas in groups of four (five minutes). Capture the most salient actions from the small groups on the left side of the worksheet (ten minutes).

4. As a follow-up, explore actions that promote autonomy. Ask “What actions boost the autonomy of Scrum Teams?” Capture them in the right column of the worksheet. Start with a minute of silent thinking (one minute), then invite people to share their ideas in groups of four (five minutes). Capture the most salient actions from the small groups on the right side of the worksheet (ten minutes).
5. Now that you have actions that each address one side of the wicked question, help the group move into yes-and thinking. Ask “Which actions boost both integration and autonomy?” Capture them on the worksheet in the middle. Start with a minute of silent thinking (one minute), then invite people to share their ideas in groups of four (five minutes). Capture the most salient actions from the small groups on the middle of the worksheet (ten minutes).
6. Now that people have experience identifying actions that serve both sides, investigate earlier actions to see if they can be shifted to the middle. Ask “Which actions on the left or the right of the worksheet can be creatively modified to boost both integration and autonomy?” Start with a minute of silent thinking (one minute), then invite people to share their ideas in groups of four (five minutes). Capture the most salient actions from the small groups on the middle of the worksheet (ten minutes).
7. Order actions by their ability to promote both integration and autonomy and identify 15% Solutions for the most impactful ones (see Chapter 10).

## **Our Findings**

- Coming up with specific and tangible actions can be difficult. Keep asking “How would you do that for us?” or “What would that look like here?” in order to move groups beyond abstract ideas and platitudes (such as “more communication”).
- If you have a large group, you can make each group of four responsible for one of the actions you identified during step 2. Let them fill in the entire worksheet in their small group from the perspective of that action.

- 
- You can replace the sides—integration and autonomy—with other wicked challenges. For example, there is also tension between responding to change as quickly as possible and preventing huge mistakes. Or the tension between standardization on the one hand and customization on the other. Work with whatever wicked challenge makes the most sense!

---

# INDEX

---

## Numbers

- “1-2-4-All” Liberating Structure, 74–75, 95, 195–198, 263
- “15% Solutions”, Liberating Structure, 131–134, 190–191, 263
- 96 *Visualization Examples* (Janlén), 256–259

## A

- accountability, 235–237
- adaptation, 37
- agile software development. *See also* Scrum Framework
- agile transformation process, 17
- versus continuous improvement, 161–163
- qualities of, 32
- Andreessen, Marc, 60
- antifragility, 214–215
- “Appreciative Interviews” Liberating Structure, 200–202
- Argyris, Chris, 158

- Ask Powerful Questions during Sprint Retrospectives experiment, 184–185
- Ask Powerful Questions to Get Things Done experiment, 139–141
- assessment, importance of, 11
- automation, 131–135
- autonomy of teams, increasing, 234–237
- characteristics of healthy Scrum, 234–237
- integrated autonomy, 242–245
- with permission tokens, 240–242
- testing of rules, 245–246
- awareness, creating, 12

## B

- backlogs
- Product Backlog
- cycle time, 113–114, 126–129
- experiments related to, 88–96

- expressing desired outcomes in, 94–96
- lead time, 126–129
- limiting maximum length of, 88–90
- mapping on ecocycle, 90–94
- Product Owner input into, 61–62
- refinement of, 70–71, 148–150
- stakeholder involvement in, 70–71
- Sprint Backlog, 43
  - carrying over to next Sprint, 21–22
  - failures in, 20
  - improvements in, 158. *See also*
    - continuous improvement
    - refinement during, 70–71, 148–150
- Bake a Release Pie experiment, 202–204
- biases, 174
- “big-bang” releases, avoidance of, 104, 106, 121, 164–165
- Birth phase, Ecocycle Planning, 91
- Black Swans events, 214
- boards, Scrum, 256–259
- Bootstrap, 144
- bottlenecks, 113
- Break the Rules! experiment, 245–246
- Business, IT versus, 59–61
- C**
- Cargo Cult Scrum. *See* Zombie Scrum
- case studies
  - continuous improvement
    - healthy Scrum, 179–180
    - Zombie Scrum, 155–157
  - self-organization, 207–209
  - shipping fast
    - healthy Scrum, 119–120
    - Zombie Scrum, 99–101
  - stakeholder needs, 48
  - Zombie Scrum symptoms, 16
- celebrating improvement, 12
- change, incremental, 12
- Chaos Monkey, 215
- checklist, for Zombie Scrum
  - diagnosis, 5–6
- “Checklist Scrum”, 17–18
- churn rate, 111
- CMS (content management system)
  - case study, 112
- cognitive dissonance, 174
- collaboration, stakeholder, 56–59
- complaints, avoiding, 12
- complex, adaptive problems
  - environmental complexity
    - self-organization for, 214–217
    - shipping fast and, 102–104
  - product complexity, 104–105
  - Scrum Framework as solution to, 31–39
    - detection of complexity, 42–44
    - Empirical Process Control Theory, 36–37
    - empiricism in Scrum Framework, 37–40
    - uncertainty and risk, 35–36
- complex environments
  - self-organization for, 214–217
  - shipping fast and, 102–104
- complimenting success, 171–172
- confidence intervals, 129
- conflicts, group functioning and, 174
- content management system (CMS)
  - case study, 112

- 
- continuous delivery, business case for, 124–126
  - continuous improvement, 161–163.
    - See also* improvement-related experiments
    - versus agile transformation, 161–163
    - benefits of, 157–158
    - case studies of
      - healthy Scrum, 179–180
      - Zombie Scrum, 155–157
    - deep learning, encouragement of
      - double-loop learning, 159–163
      - impediment newsletters, 184–185
      - problem/solution analysis, 187–189
      - questions during Sprint Retrospectives, 185–187
    - definition of, 158–160
    - Force Field Model of, 161–163
    - in healthy Scrum
      - case study of, 179–180
      - self-critical teams and, 181
      - viewing system as whole, 181–182
    - lack of, reasons for, 21–22
      - failure to celebrate success, 171–172
      - failure to critique how work is done, 175–177
      - failure to recognize human factors of work, 172–175
      - failure to value mistakes, 163–165
      - lack of safety to fail, 168–171
      - lack of tangible improvements, 166–168
      - learning and work considered different things, 177–179
      - learning environments, creating
        - release pies, 202–204
        - success stories, 200–202
      - new information, gathering
        - formal and informal networks for, 195–198
        - metrics dashboards, 198–200
      - organizational culture and, 169–171
      - prevalence of failures in, 157
      - tangible improvements, creating
        - 15% solutions, 131–134, 189, 190–191
        - improvement recipes, 193–195
        - unproductive tasks, elimination of, 191–193
    - control, locus of, 235–237
    - “Conversation Café” Liberating Structure, 74–75
    - conversations, writing PBIs (Product Backlog items) as, 94–96
    - Create 15% Solutions experiment, 190–191
    - Create a Low-Tech Metrics Dashboard to Track Outcomes experiment, 198–200
    - Create Better Sprint Goals with Powerful Questions experiment, 254–256
    - Create Improvement Recipes experiment, 193–195
    - Create Transparency with the Stakeholder Distance Metric experiment, 76–78
    - Creative Destruction phase, Ecocycle Planning, 92
    - criticisms, self-censorship of, 173
    - cross-functionality, improving with skill matrix, 141–145
    - CRUD (create, read, update, delete) interactions, 150

---

- culture, organizational, 169–171
- cycle time, 113–114
  - definition of, 127, 146
  - measurement of, 126–129
- cynicism, avoiding, 12, 267
- D**
- Daily Scrum, 2, 18–19, 48, 139, 140–141, 158, 179, 252
- Dark Scrum. *See* Zombie Scrum
- dashboards, metrics, 195–198
- Decorate the Team Room with the Product Purpose experiment, 80–81
- deep learning, encouragement of
  - double-loop learning, 159–163
  - impediment newsletters, 184–185
  - problem/solution analysis, 187–189
  - questions during Sprint Retrospectives, 185–187
- Definition of Done, 21, 95, 135–137, 257
- delivery, continuous, 124–126. *See also* shipping fast
- denial of Zombie Scrum, 25
- deployment, automation of, 131–135. *See also* shipping fast
- Develop Local Solutions with Open Space Technology experiment, 261–263
- Development Team, 64–65
  - autonomy of, 234–235
  - collaboration with stakeholders, 56–59
  - control and accountability by, 235–237
  - distance between developers/ stakeholders, 56–59
  - “I’m only here to code” attitude among, 64–65
  - paired developers, 144
  - pairing, 144
  - resolving impediments in, 223–225
  - stakeholders’ relationship with, 64–65
- DevOps, 43–44
- Diagnose Your Team Together experiment, 25–29
- diagnosing Zombie Scrum, 268
  - checklist for, 5–6
  - Diagnose Your Team Together experiment, 25–29
  - Symptoms Checker, 6
  - Zombie Scrum Survey, 25–29, 268
- Dig Deeper into Problems and Potential Solutions, Together experiment, 187–189
- “Discovery and Action Dialogue” Liberating Structure, 187–189, 263
- dissonance, cognitive, 174
- distance metric, stakeholder, 76–78
- “done,” definition of, 21, 95, 135–137, 257
- Done Increments. *See* incremental releases
- double-loop learning, 159–163
- doubt
  - acceptance of, 168–171
  - self-censorship of, 173
- Dutch railways, incident-management project for, 35
- E**
- Ebers, Anja, 186
- Edmondson, Amy, 172–173
- efficiency mindset, 40–42, 232
- Empirical Process Control
  - stakeholder satisfaction, measurement of, 129–131
  - theory of, 36–37

- environments
  - environmental complexity, 102–104
    - self-organization for, 214–217
    - shipping fast and, 102–104
  - as external memory, 229–231
  - learning environments, creating
    - release pies, 202–204
    - success stories, 200–202
- Etsy, 106
- Evolve Your Definition of Done
  - experiment, 135–137
- experiments
  - for continuous improvement
    - Ask Powerful Questions
      - during Sprint Retrospectives, 184–185
    - Bake a Release Pie, 202–204
    - Create 15% Solutions, 190–191
    - Create a Low-Tech Metrics Dashboard to Track Outcomes, 198–200
    - Create Improvement Recipes, 193–195
    - Dig Deeper into Problems and Potential Solutions, Together, 187–189
    - Focus on What to Stop Doing, 191–193
    - Share an Impediment
      - Newsletter throughout the Organization, 184–185
    - Share Success Stories and Build on What Made Them Possible, 200–202
    - Use Formal and Informal Networks to Drive Change, 195–198
  - Diagnose Your Team Together, 25–29
  - versus initiatives, 164
  - for self-organization
    - Break the Rules!, 245–246
    - Create Better Sprint Goals with Powerful Questions, 254–256
    - Develop Local Solutions with Open Space Technology, 261–263
    - Express Clear Requests for Help, 249–251
    - Find a Minimum Set of Rules for Self-Organization, 247–248
    - Find Actions That Boost Both Integration and Autonomy, 242–245
    - Make the Cost of Low Autonomy Transparent with Permission Tokens, 240–242
    - Observe What Is Happening, 251–254
    - Organize Scrum Master Impediment Gatherings, 259–261
    - Use a Physical Scrum Board, 256–259
  - for shipping fast
    - Ask Powerful Questions to Get Things Done, 139–141
    - Evolve Your Definition of Done, 135–137
    - Increase Cross-Functionality with a Skill Matrix, 141–145
    - Limit Your Work in Progress, 145–148
    - Make a Business Case for Continuous Delivery, 124–126
    - Measure Lead and Cycle Times, 126–129
    - Measure Stakeholder Satisfaction, 129–131
    - Ship Every Sprint, 135–139



- Slice Your Product Backlog
  - Items, 148–150
- Take the First Steps to Automating Integration and Deployment, 131–135
- for stakeholder relationships, 74
  - Create Transparency with the Stakeholder Distance Metric, 76–78
- Decorate the Team Room with the Product Purpose, 80–81
- Express Desired Outcomes, Not Work to Be Done, 94–96
- Give the Stakeholder a Desk Close to the Scrum Team, 78–79
- Go on a User Safari, 84–85
- Guerrilla Testing, 86–88
- Invite Stakeholders to a “Feedback Party”, 81–83
- Limit the Maximum Length of Your Product Backlog, 88–90
- Map Your Product Backlog on an Ecocycle, 90–94
- Start a Stakeholder Treasure Hunt, 74–76
- Express Clear Requests for Help experiment, 249–251
- Express Desired Outcomes, Not Work to Be Done experiment, 94–96
- external memory, environment as, 229–231
- external stakeholders, 59–61
- Extreme Programming, 43–44
- F**
- Facebook, 106
- feedback loops, 24
- feedback parties, stakeholder, 81–83
- field experiences
  - continuous improvement
    - healthy Scrum, 179–180
    - Zombie Scrum, 155–157
  - guerrilla testing, 88
  - release strategies for shipping fast, 119–120
  - self-organization, 207–209
  - shipping fast
    - healthy Scrum, 112
    - Zombie Scrum, 99–101
  - stakeholder needs, 48
  - user safaris, 85
  - Zombie Scrum symptoms, 16
- Find a Minimum Set of Rules for Self-Organization experiment, 247–248
- Find Actions That Boost Both Integration and Autonomy
  - Find Actions That Boost Both Integration and Autonomy experiment, 242–245
- First Aid Kit, 11–12, 268
- fixed-price/fixed-scope projects, 224
- flow optimization
  - cross-functionality, improving with skill matrix, 141–145
  - refinement workshops, 148–150
  - work in progress, limits on, 145–148
- focus, maintaining
  - Product Backlog
    - expressing desired outcomes in, 94–96
    - limiting maximum length of, 88–96
    - mapping on ecocycle, 90–94
    - questions to assist with, 139–141
  - Focus on What to Stop Doing experiment, 191–193

- Force Field Model, 161–163
- formal networks, driving change with, 195–198
- fundamental attribution errors, 174
- G**
- Give the Stakeholder a Desk Close to the Scrum Team experiment, 78–79
- Go on a User Safari experiment, 84–85
- goals
  - creation of, 254–256
  - importance of, 40
  - imposed, 227–229
  - lack of, 22, 227–229
  - questions to help teams focus on, 139–141
  - shared, 228
- governance, plan-driven, 108–110
- Grassé, Pierre-Paul, 229
- groups
  - forming, 12
  - groupthink, 173
- guerrilla testing, 86–88
- Guerrilla Testing experiment, 86–88
- H**
- “Happy-Clappy Scrum”, 166–168
- HealthCare.gov launch, 164–165
- help, finding, 12
- human factors of work, 172–175
- humor, benefits of, 12
- I**
- “I’m only here to code” attitude, 64–65
- impediment gatherings, 259–261
- impediment newsletters, 184–185
- Impediment Pyramid, 224–225
- imposed goals, 227–229
- “Impromptu Networking” Liberating Structure, 95
- improvement, continuous. *See* continuous improvement
- improvement recipes, 193–195
- improvement-related experiments
  - Ask Powerful Questions during Sprint Retrospectives, 184–185
  - Bake a Release Pie, 202–204
  - Create 15% Solutions, 190–191
  - Create a Low-Tech Metrics Dashboard to Track Outcomes, 198–200
  - Create Improvement Recipes, 193–195
  - Dig Deeper into Problems and Potential Solutions, Together, 187–189
  - Focus on What to Stop Doing, 191–193
  - Share an Impediment Newsletter throughout the Organization, 184–185
  - Share Success Stories and Build on What Made Them Possible, 200–202
  - Use Formal and Informal Networks to Drive Change, 195–198
- Increase Cross-Functionality with a Skill Matrix experiment, 141–145
- incremental change, 12
- incremental releases, 118–119, 211
- informal networks, driving change with, 195–198
- initiatives, experiments versus, 164
- inspection, 37
- integrated autonomy, 242–245
- “Integrated Autonomy” Liberating Structure, 242–245
- integration, automation of, 131–135

internal stakeholders, 59–61  
 interviews, appreciative, 200–202  
 Invite Stakeholders to a “Feedback Party” experiment, 81–83  
 IT, Business versus, 59–61

**J-K**

Janlén, Jimmy, 259  
 Kanban, 43–44, 178  
*The Kanban Guide for Scrum Teams* (Scrum.org), 145  
 KPIs (key performance indicators), 200  
 Kurtz, Cynthia, 44

**L**

large items of work, breaking down, 114–116  
 lead time  
   definition of, 127  
   measurement of, 126–129  
 learning  
   deep  
     impediment newsletters, 184–185  
     problem/solution analysis, 187–189  
     questions during Sprint Retrospectives, 185–187  
   double-loop, 159–163  
   learning environments, creating  
     release pies, 202–204  
     success stories, 195–198  
   single-loop, 159–163  
   as work, 177–179  
 Lewin, Kurt, 161  
 Liberating Structures, 74–75, 81–82, 178  
   “1-2-4-All”, 74–75, 95, 195–198, 263

“15% Solutions”, 131–134, 190–191, 263  
 “Appreciative Interviews”, 200–202  
 “Conversation Café”, 74–75  
 “Discovery and Action Dialogue”, 187–189, 263  
 Ecocycle Planning, 90–94  
 “Impromptu Networking”, 95  
 “Integrated Autonomy”, 242–245  
 “Min Specs”, 95, 247–248  
 “Myth Turning”, 186–187  
 “Shift & Share”, 81–82, 193–195  
 “Social Network Webbing”, 195–198  
 “TRIZ”, 191–193  
 “Troika Consulting”, 261  
 “User Experience Fishbowl”, 74–75  
 “What, So What, Now What?”, 25–28, 199, 263  
 “What I Need from You”, 249–251  
 “Wise Crowds”, 148, 259–261  
 Limit the Maximum Length of Your Product Backlog experiment, 88–90  
 Limit Your Work in Progress experiment, 145–148  
 limiting Producted Backlog length, 88–90  
 LinkedIn, 106  
 local solutions, development of  
   with Open Space Technology, 261–263  
   Scrum Master impediment gatherings, 259–261

**M**

Make a Business Case for Continuous Delivery experiment, 124–126

- 
- Make the Cost of Low Autonomy Transparent with Permission Tokens experiment, 240–242
  - managers, self-organization supported by, 237
  - Map Your Product Backlog on an Ecocycle experiment, 90–94
  - Material framework, 144
  - Maturity phase, Ecocycle Planning, 91
  - Maximini, Dominik, 224–225
  - Measure Lead and Cycle Times experiment, 126–129
  - Measure Stakeholder Satisfaction experiment, 129–131
  - Mechanical Scrum. *See* Zombie Scrum
  - meetups, Zombie Scrum Resistance, 266
  - Meteor, 144
  - metrics dashboards, 195–198
  - “Min Specs” Liberating Structure, 95, 247–248
  - minimal group membership, 174
  - minimum set of rules, determination of, 247–248
  - mise en place*, 70
  - mistakes, value of, 163–165
  - Morgan, Gareth, 41, 190
  - “Myth Turning” Liberating Structure, 186–187
- N**
- negativity, fear of, 169, 267
  - Netflix Chaos Monkey, 215
  - network building, 12
  - new information, gathering
    - formal and informal networks for, 195–198
    - metrics dashboards, 198–200
    - observation techniques, 251–254
  - newsletters, impediment, 184–185
  - No Man’s Sky*, 164–165
- O**
- observation techniques, 251–254
  - Observe What Is Happening experiment, 251–254
  - off-the-shelf solutions, 218–220
  - online Scrum resources, 268
  - Open Space Technology, 261–263
  - Order Makers, Product Owners as, 62
  - organizational change, Force Field Model of, 161–163
  - organizational culture, 169–171
  - On Organizational Learning* (Argyris), 158
  - Organize Scrum Master Impediment Gatherings experiment, 259–261
  - outcomes, expressing in Product Backlog, 94–96
  - output, value versus, 63–64
- P**
- pairing developers, 144
  - peer pressure, 174
  - permission tokens, 240–242
  - physical Scrum boards, 256–259
  - plan-driven governance, as
    - impediment to shipping fast, 108–110
  - positivity, importance of, 12
  - Poverty Trap, 92
  - powerful questions, asking
    - during Daily Scrum, 139–141
    - goal creation and, 254–256
-

- during Sprint Retrospectives, 184–185
  - problems
    - complex
      - detection of complexity in, 42–44
      - Empirical Process Control, 36–37, 129–131
      - Scrum Framework as solution to, 31–39
      - uncertainty and risk in, 35–37
    - definition of, 33
    - problem/solution analysis, 187–189
    - simple, 42–44
  - Product Backlog
    - cycle time, 113–114
      - definition of, 127
      - measurement of, 126–129
    - experiments related to
      - Ecocycle Planning, 90–94
      - expressing desired outcomes, not work to be done, 94–96
      - limited Producted Backlog length, 88–90
    - expressing desired outcomes in, 94–96
    - lead time
      - definition of, 127
      - measurement of, 126–129
    - limiting maximum length of, 88–90
    - mapping on ecocycle, 90–94
    - Product Owner input into, 61–62
    - refinement of, 70–71, 148–150
    - stakeholder involvement in, 70–71
  - product complexity, 104–105
  - product delivery, 124–126. *See also* shipping fast
    - product development, stakeholder involvement in, 69
  - Product Owner
    - assumptions by, 55
    - autonomy of, 61–63
    - control and accountability by, 235–237
    - as Order Maker, 62
    - relationship with stakeholders, 68–69
    - role of, 53–54
    - value of work to, 63–64
    - Zombie Scrum symptoms in, 48
  - product purpose
    - stakeholder involvement in, 69
    - understanding
      - importance of, 52–54
      - team rooms, decorating with product purpose, 80–81
  - psychological safety, lack of, 172–175
- Q**
- QA (Quality Assurance), 100
  - Qua, Fisher, 186
  - quality
    - definition of, 177
    - QA (Quality Assurance), 100
- R**
- rationalism, 37
  - redundancy, 233
  - refinement of Product Backlog, 70–71, 148–150
  - release pies, 202–204
  - releases, 106. *See also* continuous improvement; shipping fast
    - “big-bang” releases, avoidance of, 106, 121, 164–165
    - case study of, 119–120
    - deciding to release, 117–118

- incremental, 118–119, 211
  - as non-binary action, 118–119
  - release pies, 202–204
- Renewal phase, Ecocycle Planning, 91
- requests for help, expression of, 249–251
- resources, Scrum, 268
- responsibility, taking, 11
- responsiveness, 176
- retrospectives, 139, 147–148, 185–187
  - fear of failure in, 169
  - questions to ask during, 185–187
  - self-criticism in, 181
  - viewing system as whole in, 181–182
  - Zombie Scrum symptoms in, 2
- Rigidity Trap, 92
- risk
  - challenges of, 35–36
  - Empirical Process Control
    - stakeholder satisfaction, measurement of, 129–131
    - theory of, 36–37
  - reducing by shipping fast, 106–108
- robustness, 214–215
- rules
  - minimum set of, 247–248
  - self-organization through, 211–212
  - testing, 245–246
- S**
- safety to fail, 168–171
- sarcasm, avoiding, 12
- scatter plots, 129
- Schein, Edgar, 169–170
- Schwaber, Ken, 37, 268
- Scrum Alliance, 17
- Scrum boards, 146, 256–259
- Scrum Framework, 40–42. *See also* Zombie Scrum
  - agile transformation process in, 17
  - challenges of, 44
  - control and accountability in, 235–237
  - definition of, 33
  - dynamic natures of, 40
  - empirical approach of, 37–40
  - as evolving set of minimal boundaries, 39–40
  - feedback loops in, 24
  - “Happy-Clappy Scrum”, 166–168
  - history of, 37–38
  - human factors of, 172–175
  - impediments to, 268–269
  - incremental releases in, 211
  - output versus value in, 63–64
  - pillars of, 37
  - popularity of, 17
  - purpose of, 37–39
  - as solution to complex, adaptive problems, 31–39
    - detection of complexity in, 42–44
  - Empirical Process Control Theory and, 36–37
  - empiricism in Scrum Framework, 37–40
  - uncertainty and risk in, 35–36
  - as solution to seemingly simple problems, 42–44
  - superficial implementation of, 17–18
- Scrum Guide, 37–40
  - online resources for, 268
- Scrum defined in, 33

- Sprint Review explained in, 70
  - stakeholders defined in, 50
  - Scrum Masters
    - control and accountability by, 235–237
    - impediment gatherings, 259–261
    - as impediment to self-management
      - Scrum Masters focused only on Scrum Team(s), 225–227
      - Scrum Masters resolving all impediments, 223–225
  - Scrum Teams, 56–59. *See also*
    - continuous improvement;
    - Product Owner; shipping fast
  - autonomy of, 234–237
  - control and accountability in, 235–237
  - Development Team, 64–65
    - autonomy of, 234–235
    - collaboration with stakeholders, 56–59
    - control and accountability by, 235–237
    - distance between developers/ stakeholders, 56–59
    - “I’m only here to code” attitude among, 64–65
    - paired developers, 144
    - resolving impediments in, 223–225
    - stakeholders’ relationship with, 64–65
  - diagnosing Zombie Scrum in, 268. *See also* Zombie Scrum
  - checklist for, 5–6
  - Diagnose Your Team Together experiment, 25–29
  - Symptoms Checker, 6
  - Zombie Scrum Survey, 25–29, 268
  - focus, maintaining, 139–141
  - increasing cross-functionality on, 141–145
  - management support for, 237
  - output versus value in, 63–64
  - questions to help focus in, 139–141
  - Scrum Masters focused only on, 225–227
  - self-critical, 181
  - stakeholder relationship with. *See* stakeholders
  - understanding purpose of product in, 52–54
- scrumguides.org, 268
  - Scrum.org, 17, 145, 268
  - self-alignment, promotion of
    - with physical Scrum boards, 256–259
    - through goal creation, 254–256
  - self-critical teams, 181
  - self-management, 212–214
  - self-organization, 23. *See also*
    - self-organization experiments
    - autonomy of teams, increasing
    - characteristics of healthy Scrum, 234–237
    - integrated autonomy, 242–245
    - with permission tokens, 240–242
    - testing of rules, 245–246
  - benefits of, 209–210
  - building confidence for
    - clear requests for help, expression of, 249–251
    - minimum set of rules, determination of, 247–248
    - observation techniques, 251–254
  - case study of, 207–209
  - definition of, 210

- in healthy Scrum
  - autonomy of Scrum Teams, 234–237
  - management support, 237
- impediments to
  - lack of goals, 227–229
  - limited ability to self-organize, 218–220
  - off-the-shelf solutions, 220–222
  - physical environment does not reinforce external memory, 229–231
  - Scrum Masters focused only on Scrum Team(s), 225–227
  - Scrum Masters resolving all impediments, 223–225
  - standardization, 232–234
- local solutions, development of
  - with Open Space Technology, 261–263
  - Scrum Master impediment gatherings, 259–261
- prevalence of failures in, 209
- self-alignment, promotion of
  - with physical Scrum boards, 256–259
  - through goal creation, 254–256
- stigmergy, 229–231
- as survival skill for complex environments, 214–217
- through self-management, 212–214
- through simple rules, 211–212
- self-organization experiments
  - Break the Rules! 245–246
  - Create Better Sprint Goals with Powerful Questions, 254–256
  - Develop Local Solutions with Open Space Technology, 261–263
  - Express Clear Requests for Help, 249–251
  - Find a Minimum Set of Rules for Self-Organization, 247–248
  - Find Actions That Boost Both Integration and Autonomy, 242–245
  - Make the Cost of Low Autonomy Transparent with Permission Tokens, 240–242
  - Observe What Is Happening, 251–254
  - Organize Scrum Master Impediment Gatherings, 259–261
  - Use a Physical Scrum Board, 256–259
- Sevенеves* (Stephenson), 217
- Share an Impediment Newsletter throughout the Organization experiment, 184–185
- Share Success Stories and Build on What Made Them Possible experiment, 200–202
- “Shift & Share” Liberating Structure, 81–82, 193–195
- Ship Every Sprint experiment, 135–139
- shipping fast. *See also* shipping-related experiments
  - benefits of, 102–105
  - response to environmental complexity, 102–104
  - response to product complexity, 104–105
- case studies of
  - healthy Scrum, 112
  - Zombie Scrum, 99–101
- continuous delivery, business case for, 124–126



- cycle time, 113–114
  - definition of, 127, 146
  - measurement of, 126–129
- Definition of Done, evolving, 135–137
- flow optimization
  - cross-functionality, improving
    - with skill matrix, 141–145
  - refinement workshops, 148–150
  - work in progress, limits on, 145–148
- in healthy Scrum
  - “big-bang” releases, avoidance of, 104, 106, 121
  - case study of, 112
  - release strategies for, 117–120
  - shipping during Sprints, 120–121
- impediments to, 20–21, 105
  - failure to remove, 113–114
  - failure to understand
    - competitive advantages, 110–111
  - failure to understand risk reduction, 106–108
  - large work items in Sprints, 114–116
  - plan-driven governance, 108–110
  - prevalence of, 101
- lead time
  - definition of, 127
  - measurement of, 126–129
- prevalence of failures in, 101
- risk reduction through, 106–108
- shipping more often
  - automation in, 131–135
  - Definition of Done, evolving, 135–137
  - questions to help teams focus on goals, 139–141
  - shipping during Sprints, 137–139
  - stakeholder satisfaction, measurement of, 129–131
  - throughput, 146
- shipping-related experiments
  - Ask Powerful Questions to Get Things Done, 139–141
  - Evolve Your Definition of Done, 135–137
  - Increase Cross-Functionality with a Skill Matrix, 141–145
  - Limit Your Work in Progress, 145–148
  - Make a Business Case for Continuous Delivery, 124–126
  - Measure Lead and Cycle Times, 126–129
  - Measure Stakeholder Satisfaction, 129–131
  - Ship Every Sprint, 135–139
  - Slice Your Product Backlog Items, 148–150
  - Take the First Steps to Automating Integration and Deployment, 131–135
- silos-thinking, 19
- simple problems, detecting complexity in, 42–44
- single-loop learning, 159–163
- situational assessment, 11
- skill matrix, 141–145
- Slice Your Product Backlog Items experiment, 148–150
- Snowden, Dave, 44
- “Social Network Webbing”
  - Liberating Structure, 195–198

- sociotechnical systems (STS)
  - approach, 213
- SpaceX (Space Exploration Technologies), 215
- “Specification by Example”, 144
- speed of shipping. *See* shipping fast
- Spotify Model, 220–222
- Sprint Backlog, 43
  - carrying over to next Sprint, 21–22
  - improvements in, 158. *See also* continuous improvement
  - large items of work in, 114–116
  - refinement during, 70–71, 148–150
- Sprint Goals
  - creation of, 254–256
  - importance of, 40
  - imposed, 227–229
  - lack of, 22, 227–229
  - questions to help teams focus on, 139–141
  - shared, 228
- Sprint Planning
  - refinement during, 71
  - Zombie Scrum symptoms in, 22, 48
- Sprint Retrospectives, 139, 147–148, 185–187
  - fear of failure in, 169
  - questions to ask during, 185–187
  - self-criticism in, 181
  - viewing system as whole in, 181–182
  - Zombie Scrum symptoms in, 2
- Sprint Review, 139
  - celebration of success in, 171–172
  - purpose of, 70
  - Zombie Scrum symptoms in. *See* Zombie Scrum
- Stacey, Ralph, 44
- stakeholder distance metric, 76–78
- stakeholder satisfaction, 129–131
- stakeholder-related experiments, 74
  - Create Transparency with the Stakeholder Distance Metric, 76–78
  - Decorate the Team Room with the Product Purpose, 80–81
  - Express Desired Outcomes, Not Work to Be Done, 94–96
  - Give the Stakeholder a Desk Close to the Scrum Team, 78–79
  - Go on a User Safari, 84–85
  - Guerrilla Testing, 86–88
  - Invite Stakeholders to a “Feedback Party”, 81–83
  - Limit the Maximum Length of Your Product Backlog, 88–90
  - Map Your Product Backlog on an Ecocycle, 90–94
  - Start a Stakeholder Treasure Hunt, 74–76
- stakeholders
  - assumptions about, 51–52, 55–56
  - case study of, 48
  - definition of, 50
  - failure to involve, reasons for
    - assumptions about stakeholder needs, 55–56
    - belief that only developers should write code, 64–65
    - distance between developers/stakeholders, 56–59
    - distinctions between “Business” and “IT”, 59–61
    - lack of autonomy of Product Owners, 61–63
    - measurement of output over value, 63–64

- misunderstanding of product purpose and, 52–54
  - unmotivated stakeholders, 66–67
  - failure to understand, 19–20, 48–49
  - getting to know
    - stakeholder desk close to Scrum Team, 78–79
    - stakeholder distance metric, 76–78
    - stakeholder treasure hunts, 74–76
    - team involvement in, 68–69
    - team rooms, decorating with product purpose, 80–81
  - in healthy Scrum
    - stakeholder involvement, 69–71
    - stakeholder/team interaction, 68–69
  - identification of, 49
  - importance of, 49
  - internal versus external, 59–61
  - satisfaction of, 129–131
  - when to involve, 49
    - during creation of product purpose, 69
    - during guerrilla testing, 86–88
    - during Product Backlog refinement, 70–71
    - during product development kickoff, 69
    - during Sprint Review, 70
    - during stakeholder feedback parties, 81–83
    - during user safaris, 84–85
  - standardization, 232–234
  - Start a Stakeholder Treasure Hunt experiment, 74–76
  - Stephenson, Neil, 217
  - stigmergy, 229–231, 256–259
  - STS (sociotechnical systems) approach, 213
  - success, celebration of, 171–172
  - success stories, creating, 200–202
  - superficial Scrum, 17–18
  - survey, Zombie Scrum, 25–29, 268
  - survivors, finding, 12
  - Sutherland, Jeff, 37, 268
  - Symptoms Checker, 3, 6
- T**
- Take the First Steps to Automating Integration and Deployment experiment, 131–135
  - Taleb, Nassim, 214, 233
  - tangible improvements
    - creation of
      - 15% solutions, 131–134, 189, 190–191
      - improvement recipes, 193–195
      - unproductive tasks, elimination of, 191–193
    - lack of, 166–168
  - Tavistock Institute of Human Relations, 213
  - team diagnoses. *See* diagnosing Zombie Scrum
  - team rooms, decorating with product purpose, 80–81
  - teams. *See* Scrum Teams
  - testing
    - guerrilla, 86–88
    - of rules, 245–246
    - user acceptance, 95
  - theory of resolution of invention-related tasks (TRIZ), 191–193
  - throughput, 146
  - tokens, monitoring permission requirements with, 240–242

- TPS (Toyota Production System), 213
- transparency
- of required permissions, 240–242
  - of Scrum Framework, 37
  - in shipping
    - continuous delivery, business case for, 124–126
    - lead and cycle times, measurement of, 126–129
    - stakeholder satisfaction, measurement of, 129–131
- treasure hunts, stakeholder, 74–76
- “TRIZ” Liberating Structure, 191–193
- “Troika Consulting” Liberating Structure, 261
- U-V**
- uncertainty
- acceptance of, 168–171
  - challenges of, 35–36
  - Empirical Process Control
    - stakeholder satisfaction, measurement of, 129–131
    - theory of, 36–37
- unproductive tasks, elimination of, 191–193
- urgency in shipping, creating. *See also* shipping fast
- continuous delivery, business case for, 124–126
  - lead and cycle times, measurement of, 126–129
  - stakeholder satisfaction, measurement of, 129–131
- Use a Physical Scrum Board experiment, 256–259
- Use Formal and Informal Networks to Drive Change experiment, 195–198
- user acceptance tests, 95
- “User Experience Fishbowl” Liberating Structure, 74–75
- user safaris, 84–85
- value
- assumptions about, 49
  - definition of, 177
  - output versus, 63–64
- W**
- “What, So What, Now What?” Liberating Structure, 25–28, 199, 263
- “What I Need from You”, Liberating Structure, 249–251
- WIP (work in progress), limits on, 145–148
- “Wise Crowds” Liberating Structure, 148, 259–261
- work
- critique of, 175–177
  - human factors of, 172–175
  - learning as, 177–179
  - work in progress, limits on, 145–148
- workshops, refinement, 71, 148–150
- X-Y-Z**
- Zombie Scrum. *See also* Scrum Framework
- case study of, 16
  - causes of, 267–269
  - denial of, 25
  - diagnosis of, 268
    - checklist for, 5–6
    - Diagnose Your Team Together experiment, 25–29
    - Symptoms Checker, 6
    - Zombie Scrum Survey, 25–29, 268
  - efficiency mindset in, 40–42, 232

- experiments for. *See* experiments
- First Aid Kit for, 11–12, 268
- inability to recover from, 267
- prevalence of, 3
- recovery from,
  - C01.0157-C01.0159, 24–25
- resources for, 268
- steps for fighting, 11–12
- symptoms of, 18–23
  - checklist for, 5–6
  - failure to self-organize, 23
  - failure to ship fast, 20–21
  - failure to understand
    - stakeholder needs, 19–20
  - lack of continuous improvement, 21–22
  - Symptoms Checker, 1–3, 6
  - synonyms for, 24
- Zombie Scrum Meetup, 266
- Zombie Scrum Resistance, 8–9, 266
- Zombie Scrum Survey, 25–29, 268
- zombiescrum.org, 96, 150–151, 204, 263–264, 268