



# Retrospectives ANTIPATTERNS



Aino Vonge Corry

*Foreword by Diana Larsen*



# **RETROSPECTIVES**

# **ANTIPATTERNS**

*This page intentionally left blank*

# RETROSPECTIVES ANTIPATTERNS

---

Aino Vonge Corry

◆ 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 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.

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: 2020941957

Copyright © 2021 Pearson Education, Inc.

Cover and interior illustrations: Nikola Korac

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: 978-0-13-682336-0

ISBN-10: 0-13-682336-X

ScoutAutomatedPrintCode

**Publisher**

Mark L. Taub

**Executive Editor**

Greg Doench

**Assistant Editor**

Menka Mehta

**Managing Producer**

Sandra Schroeder

**Sr. Content Producer**

Julie B. Nahil

**Project Manager**

Aswini Kumar/  
codeMantra

**Copy Editor**

Carol Lallier

**Indexer**

Ken Johnson

**Proofreader**

Donna Mulder

**Cover Designer**

Chuti Prasertsith

**Compositor**

codeMantra

This book is dedicated to my best friend, Ellen Agerbo, who, based on her experience with me as we worked on our CS degrees, our common master's thesis, and part of the PhD program, never believed I would be able to write a book. To quote her: "You were always excellent at coming up with ideas, but writing has never been your strong suit."

*This page intentionally left blank*

# CONTENTS

---

Foreword		xv
Preface		xvii
Acknowledgments		xxxv
About the Author		xxxvii
	The Story Begins	1
Part I	Structural Antipatterns	2
Chapter 1	Wheel of Fortune	4
	Context	6
	General Context	7
	Antipattern Solution	7
	Consequences	7
	Symptoms	8
	Refactored Solution	9
	Online Aspect	12
	Personal Anecdote	13



<b>Chapter 2</b>	<b>Prime Directive Ignorance</b>	<b>16</b>
	Context	18
	General Context	20
	Antipattern Solution	21
	Consequences	21
	Symptoms	22
	Refactored Solution	22
	Online Aspect	23
	Personal Anecdote	23
<b>Chapter 3</b>	<b>In the Soup</b>	<b>26</b>
	Context	28
	General Context	28
	Antipattern Solution	29
	Consequences	29
	Symptoms	30
	Refactored Solution	30
	Online Aspect	34
	Personal Anecdote	34
<b>Chapter 4</b>	<b>Overtime</b>	<b>36</b>
	Context	38
	General Context	39
	Antipattern Solution	39
	Consequences	39
	Symptoms	40
	Refactored Solution	40
	Online Aspect	43
	Personal Anecdote	43
<b>Chapter 5</b>	<b>Small Talk</b>	<b>46</b>
	Context	48
	General Context	48
	Antipattern Solution	48
	Consequences	49

---

	Symptoms	49
	Refactored Solution	49
	Online Aspect	51
	Personal Anecdote	51
<b>Chapter 6</b>	<b>Unfruitful Democracy</b>	<b>54</b>
	Context	56
	General Context	56
	Antipattern Solution	57
	Consequences	57
	Symptoms	58
	Refactored Solution	58
	Online Aspect	60
	Personal Anecdote	61
<b>Chapter 7</b>	<b>Nothing to Talk About</b>	<b>62</b>
	Context	64
	General Context	64
	Antipattern Solution	64
	Consequences	65
	Symptoms	65
	Refactored Solution	65
	Online Aspect	70
	Personal Anecdote	71
<b>Chapter 8</b>	<b>Political Vote</b>	<b>74</b>
	Context	76
	General Context	77
	Antipattern Solution	77
	Consequences	78
	Symptoms	78
	Refactored Solution	78
	Online Aspect	79
	Personal Anecdote	79

---

<b>Part II</b>	<b>Planning Antipatterns</b>	<b>80</b>
<b>Chapter 9</b>	<b>Team, Really?</b>	<b>82</b>
	Context	84
	General Context	85
	Antipattern Solution	85
	Consequences	85
	Symptoms	86
	Refactored Solution	86
	Online Aspect	87
	Personal Anecdote	88
<b>Chapter 10</b>	<b>Do It Yourself</b>	<b>90</b>
	Context	92
	General Context	92
	Antipattern Solution	92
	Consequences	93
	Symptoms	93
	Refactored Solution	94
	Online Aspect	96
	Personal Anecdote	96
<b>Chapter 11</b>	<b>Death by Postponement</b>	<b>98</b>
	Context	100
	General Context	100
	Antipattern Solution	100
	Consequences	101
	Symptoms	101
	Refactored Solution	102
	Online Aspect	103
	Personal Anecdote	104
<b>Chapter 12</b>	<b>Get It Over With</b>	<b>106</b>
	Context	108
	General Context	108
	Antipattern Solution	108

---

	Consequences	109
	Symptoms	109
	Refactored Solution	109
	Online Aspect	110
	Personal Anecdote	111
<b>Chapter 13</b>	<b>Disregard for Preparation</b>	<b>114</b>
	Context	116
	General Context	117
	Antipattern Solution	118
	Consequences	118
	Symptoms	119
	Refactored Solution	120
	Online Aspect	122
	Personal Anecdote	123
<b>Chapter 14</b>	<b>Suffocating</b>	<b>124</b>
	Context	126
	General Context	127
	Antipattern Solution	127
	Consequences	127
	Symptoms	127
	Refactored Solution	128
	Online Aspect	128
	Personal Anecdote	129
<b>Chapter 15</b>	<b>Curious Manager</b>	<b>130</b>
	Context	132
	General Context	132
	Antipattern Solution	133
	Consequences	133
	Symptoms	133
	Refactored Solution	133
	Online Aspect	134
	Personal Anecdote	134

---

<b>Chapter 16</b>	<b>Peek-A-Boo</b>	<b>136</b>
	Context	138
	General Context	138
	Antipattern Solution	139
	Consequences	139
	Symptoms	140
	Refactored Solution	140
	Online Aspect	143
	Personal Anecdote	143
<b>Part III</b>	<b>People Antipatterns</b>	<b>146</b>
<b>Chapter 17</b>	<b>Disillusioned Facilitator</b>	<b>148</b>
	Context	150
	General Context	151
	Antipattern Solution	151
	Consequences	151
	Symptoms	151
	Refactored Solution	152
	Online Aspect	153
	Personal Anecdote	153
<b>Chapter 18</b>	<b>Loudmouth</b>	<b>156</b>
	Context	158
	General Context	158
	Antipattern Solution	158
	Consequences	159
	Symptoms	160
	Refactored Solution	160
	Online Aspect	162
	Personal Anecdote	162
<b>Chapter 19</b>	<b>Silent One</b>	<b>166</b>
	Context	168
	General Context	168
	Antipattern Solution	169

---

	Consequences	169
	Symptoms	169
	Refactored Solution	170
	Online Aspect	171
	Personal Anecdote	172
<b>Chapter 20</b>	<b>Negative One</b>	<b>174</b>
	Context	176
	General Context	176
	Antipattern Solution	176
	Consequences	177
	Symptoms	177
	Refactored Solution	177
	Online Aspect	179
	Personal Anecdote	180
<b>Chapter 21</b>	<b>Negative Team</b>	<b>182</b>
	Context	184
	General Context	184
	Antipattern Solution	184
	Consequences	185
	Symptoms	185
	Refactored Solution	186
	Online Aspect	187
	Personal Anecdote	187
<b>Chapter 22</b>	<b>Lack of Trust</b>	<b>188</b>
	Context	190
	General Context	191
	Antipattern Solution	191
	Consequences	191
	Symptoms	191
	Refactored Solution	192
	Online Aspect	197
	Personal Anecdote	197

---

<b>Chapter 23</b>	<b>Different Cultures</b>	<b>202</b>
	Context	204
	General Context	204
	Antipattern Solution	204
	Consequences	205
	Symptoms	205
	Refactored Solution	205
	Online Aspect	207
	Personal Anecdote	207
<b>Chapter 24</b>	<b>Dead Silence</b>	<b>210</b>
	Context	212
	General Context	212
	Antipattern Solution	213
	Consequences	213
	Symptoms	214
	Refactored Solution	214
	Online Aspect	216
	Personal Anecdote	216
	<b>Conclusion</b>	<b>219</b>
<b>References</b>		<b>221</b>
<b>Index</b>		<b>225</b>

# FOREWORD

---

In this book, Aino tells the story of how our relationship began. However, there's more to the story. There's a continually unfolding story about how our connection has continued, deepened, and matured through the ensuing years. Over time Aino has become more than someone I've mentored. She has become a highly respected colleague and friend.

Aino knows that, at times, I can feel overprotective about the practice of retrospectives facilitation. As an author and early proponent of team continuous learning and improvement, I want team meetings to deliver these valuable outcomes, every time. I encourage team leaders to invest in setting aside focused time for team retrospectives. I want teams (and their organizations) to receive ever-increasing benefits from their retrospective practice.

Unfortunately, I often hear stories of retrospectives-in-name-only, retrospectives held primarily to check the "retro box," or retrospectives limited to listing the answers to two or three questions, resulting in few if any actionable plans. The storytellers generally follow on with comments about teams that feel these meetings are a waste of time. I can't blame

---



them. Of course they feel that way. Those meetings take up time without providing the team with the benefits promised. In my overprotective state, I want to deny the name. Whatever these meeting are, I refute the idea that they are my idea of retrospectives.

As a consequence, I'm eager to engage with colleagues who communicate the word about leading effective retrospectives. It helps me know that I'm not alone. These days, whenever I see a conference or other event program with Aino Corry presenting on a retrospectives-related topic, I'm thrilled. I know I can relax. Those audiences (as well as the teams she works more closely with) will receive valuable information from Aino about the path to team improvement. I am happy to recommend her trainings to anyone who asks.

That's why I'm so happy to recommend this book. In it, Aino has shared a robust, curated list of antipatterns and how to avoid them. (And they will be familiar to every seasoned facilitator. I'm intimately acquainted with most of them.) *And* she has shared so much more than tips and techniques. If you read this book carefully, you will find a gold mine—with precious nuggets including her personal experiences, effective facilitation resources, and pointers for extracting yourself and your team when you're stuck in an antipattern.

Pick up this book. Study these antipatterns. Identify the ones that show up most often for you. Then make a plan for your next retrospective to include Aino's alternative solutions and improve your consequences. You'll be glad you did.

Best wishes for your future retrospectives,

Diana Larsen  
Coauthor, *Agile Retrospectives: Making Good Teams Great*  
Cofounder and Chief Connector, Agile Fluency Project LLC

# PREFACE

---

## THIS BOOK IS FOR YOU

I have wanted to write this book for you for a long time. Actually, I have worked on it for so long that it has become a running joke between my family and friends.

I imagine you sitting on the sofa in the evening, frustrated with all the things that you experience as a retrospectives facilitator, and you want to know that someone shares your pain. If that is the case, this book is for you. I am about to tell you about all my mistakes and how I saw them repeating themselves to an extent that enabled me to write patterns about them.

Don't be discouraged by my describing them as antipatterns: each has a solution as part of the antipattern description as well. Most of the solutions entail planning things differently the next time around; for example, I might give you ideas on how to remember to explain the reason for an activity. But there are also some real-time solutions that can be applied spontaneously in response to events during the retrospective—for example, ideas on how to encourage people to talk when they are silent.

When reading this book, you may notice that most of the antipatterns I describe, in the setting of a retrospective, could be found in the wild among other types of meetings as well. I would claim that the solutions described in the antipatterns could hold for whatever type of meeting you are facilitating. Because you do have a facilitator for all your meetings, right? Jutta Eckstein, in *Retrospectives for Organizational Change* (2019), also made the supported claim that the structure of a retrospective can be used in more settings than the cyclic team check-in that my context describes. She describes how retrospectives can also help you implement an organizational change on a much larger scale than the Scrum team we often relate the concept of retrospectives to.

If you are very new to all this facilitation business, it might be a good idea to read my book, because then you will be aware of all the challenges you might run into. On the other hand, who are we kidding? We almost never learn something before we need it, so you will need to make mistakes of your own before you can start to appreciate this book. Reading it now might only give you a laugh: “Did she really *do* that?” “What on earth made her think it was a good idea to say that?” But a good laugh is important, so there still might be value in it for you.

## HOW THE BOOK CAME TO LIFE

At the beginning of this millennium, I was a co-organizer and program chair at the JAOO Conference (now known as GOTO) in Aarhus, Denmark, where Linda Rising was an invited speaker. Among many things, Rising is the coauthor, with Mary Lynn Manns, of *Fearless Change* (2005), and she introduced me to Norm Kerth’s seminal book *Project Retrospectives: A Handbook for Team Reviews* (2001). I always enjoy listening to talks by Rising, but this one was special. The idea of retrospectives intrigued me, and I could see how they could be very useful to a lot of the teams at our clients’ organizations as well as teams in our own company. Reading the book after the conference did not discourage me, and in 2007, I started facilitating a few retrospectives.

The next step in my journey was to attend a course about facilitating retrospectives with Diana Larsen as teacher. This really opened my eyes to the possibilities and the challenges. I was lucky enough to assist Diana in later courses in Denmark, and as a teacher you learn even more than as a student. Ever since, I have been fascinated by the thought of helping people and teams reflect and learn. I started facilitating more retrospectives, first for my colleagues in the IT industry and later in other companies and settings. In the last 10 years, I have facilitated hundreds of retrospectives in dozens of organizations and delivered talks on retrospectives at conferences, geek nights, and, frankly, wherever people didn't get a chance to run away.

I read many books and articles and watched numerous presentations, both online and at conferences. I learned, of course, a lot of activities, but I soon understood that being a good retrospective facilitator is not just about knowing which activities to use—there is much more to it. I spent some time learning about body language, and *The Definitive Book of Body Language* (Pease & Pease 2004) gave me many insights about the importance of eye contact, handshakes, and the effects of positioning yourself in relation to other people in different ways. I drew inspiration from numerous books that were not directly aimed at facilitating retrospectives; see the References section at the end of the book.

Prompted by my husband's reminder of my poor memory, I kept notes over the years on what I saw and heard while facilitating retrospectives. I noted in a little black book the techniques I tried, what worked, and what didn't work to help teams make forward progress and avoid getting stuck in a rut. If I improvised a way of redirecting a discussion that was going around in circles, I wrote a few sentences on it. When a group of developers needed a prod in the right direction to keep their meetings constructive and useful, I used the little black book to remind myself what I had already tried and whether it worked. If I invented or stole a silly game or an exercise that got people moving when their minds and bodies were weary from sitting and talking, I jotted it down. The retrospectives in these notebooks to date add up to 296 retrospectives facilitated for 68 different teams in 27 different companies, and I know that I did a lot before I started taking notes.

When I started using the notebook, it proved helpful not just for reminding me about what I had done that worked or did not work but also for following up on experiments a team may have decided on in previous retrospectives. I also wrote plans A and B for each retrospective so that I had all my plans in one place and could get inspiration from my earlier work. Some people might rely on their memory for past successes and blunders, but that is not an option for me. When I started facilitating online retrospectives, I used online tools and initially forgot to make use of the notebook. After some less-than-successful online retrospectives, I learned that the notebook was equally useful in an online retrospective because it gives me a quick overview and helps me remember.

This book, which I started writing in October 2013, is a distillation of my little black book—or rather, books. At least it’s a distillation of the bad parts, because this is a book of antipatterns. These are the traps people have fallen into, the mistakes I’ve made, and my best tips for getting out of the traps and fixing the mistakes. Facilitating retrospectives is never the same twice; if it is, then that’s an antipattern in itself. I never want to stop learning how to make meetings more useful and how to get teams to work better together. I also take great pleasure in showing skeptical software developers, who just want to be left alone to type in code, what they can gain from communicating with their colleagues for a small part of their working week.

Although the process of learning never ends and I am still learning, I want to share with a wider circle what I have learned so far. A lot of great material can be found in books and online about how to facilitate a good retrospective, but over the years, I have seen many people struggling with the same problems. That’s why I decided this book should be structured as a collection of antipatterns. Retrospectives are not easy to facilitate and are easily ruined, or at least made less efficient, and I think the time is ripe for a book of antipatterns. But don’t be put off by the *negativity* of antipatterns. Every antipattern contains a refactored solution that has worked for me and the teams I help.

Also, when you read the Refactored Solution sections in this book, remember to consider your own context before applying my proposal. As Diana Larsen used to say whenever I asked her what to do in a retrospective: “It depends,” by which she means that it depends on the context.

## PREREQUISITES

I assume that since you are reading this book, you are familiar with retrospectives and the role of the facilitator. If you need a refresher, you could spend some time reading about retrospectives on the Internet and, of course, read *Agile Retrospectives: Making Good Teams Great* (Larsen & Derby 2006).

I have added a lot of info boxes in the book to explain concepts that you might not know about or have forgotten. Feel free to skip them if you are already aware of what the different concepts cover.

## WHAT IS A RETROSPECTIVE?

A retrospective is a chance for a team to reflect and learn from the past within a structured meeting. The main aim is to inspect the situation and adapt to the reality. *Inspect and adapt* is the core of any agile process and was first popularized with the Japanese word *kaizen* in *The Machine That Changed the World* (Womack, Jones & Roos 1990). To get a true inspection and be able to adapt to a situation, we need to create an atmosphere of trust in which people can share what they have experienced. The facilitator makes sure that every voice is heard in some way and that the team decides together what to spend time discussing or doing cause analysis on. The outcome of a retrospective is often a few experiments that the team can make in order to improve how they work. Or, as Larsen and Derby (2006) put it, retrospectives are about “making good teams great!” A retrospective is also a time to share with your team how you have experienced different events since the last retrospective and to gain a clear understanding of each other. As my late father used to say, “To understand everything is to forgive everything.”

## THE FIVE PHASES OF A RETROSPECTIVE

Among the various definitions of retrospectives, *Agile Retrospectives* (Larsen & Derby 2006) describes the life of a retrospective as five phases:

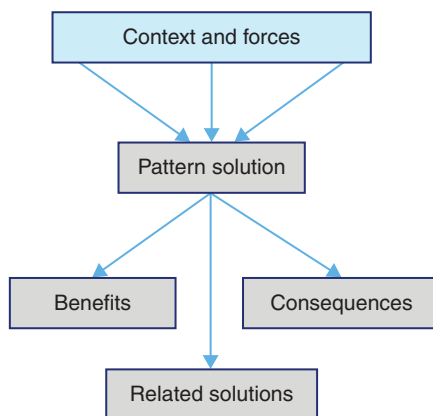
1. **Set the Stage:** The facilitator creates an atmosphere of trust, makes sure everyone's voice is heard, looks at earlier experiments, defines the theme for the retrospective, and manages any other tasks necessary to start the retrospective.
2. **Gather Data:** The team gathers data (on experiences, events, tests, sales, etc.) for the time that the retrospective is focused on.
3. **Generate Insights:** The team looks behind the data to find the stories and the causes behind them. This phase can be done as a free discussion or a cause analysis.
4. **Decide What to Do:** The team decides together what experiments to carry out to improve the way they work together.
5. **Close the Retrospective:** The team decides who is responsible for following up on the experiments. The facilitator wraps up what happened and perhaps provides an evaluation of the retrospective if he or she feels it would be valuable—retrospective over the retrospective.

Often, I hear people claim that for short sprint retrospectives, it does not make sense to go through all five phases. But that way of thinking is exactly what leads to premature decision making, as described in depth later in Chapter 1, **Wheel of Fortune**.

## WHAT IS A PATTERN?

A pattern is an abstract solution to an often recurring problem. Patterns are a means to disseminate experience in a literary form. Their names make up a common vocabulary for design, programming, or whatever domain for which the patterns describe solutions. A pattern can be a way to describe how things are done in an organization. A pattern contains a description of the context and the forces that define the problem you need to solve, the pattern solution, and the benefits and consequences of apply-

ing the pattern. A pattern also often refers to other patterns, because the consequences could be helped with a solution found in another pattern.



**Figure P.1** The elements of a pattern

The concept of patterns was originated by the building architect Christopher Alexander and his coauthors in *A Pattern Language: Towns, Buildings, Construction* (1977). Just over a decade later, patterns were introduced for use in software by Kent Beck in a *Smalltalk Report* article, “A Short Introduction to Pattern Language” ((1993) 1999), with a focus on communication. Two years later, the concept was made popular by *Design Patterns: Elements of Reusable Object-Oriented Software* (Gamma et al. 1995), now known as the GoF book because the authors became collectively known as the Gang of Four.

When working with patterns, such as the Observer, Composite, and Strategy (Gamma et al. 1995), it is useful to refer to the names of patterns instead of having to explain a design or concept from scratch. Patterns are effective because of the way the brain works with cognitive patterns and cognitive automation. When you learn something, the details of that new knowledge are “glued” together in your long-term memory as a cognitive pattern. Together with the cognitive pattern, which helps you recognize the situation as a situation in which you have learned how to act, the cognitive automations, or how to react to the pattern, are also drilled into the



brain. My former manager Michael Caspersen, who taught me almost everything I know about learning, includes many examples of and references to cognitive patterns and cognitive automations of the brain in his PhD dissertation (Caspersen 2007).

My PhD dissertation (Cornils 2001)<sup>1</sup> focused on software patterns, and I have noticed that I always see patterns in things, which is not uncommon among humans.

## WHAT IS AN ANTIPATTERN?

An antipattern is a way to describe experience. The antipattern as a concept was first named and described by William J. Brown and his coauthors in *AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis* (1998). It is a description of a solution to a frequently occurring problem in which the consequences outweigh the benefits.

The antipatterns in this book are the result of a facilitator not knowing better or not having the time or opportunity to do the right thing. Maybe the solution worked once for the facilitator in another group because the group members had a different way of communicating or knew each other better, but then it unexpectedly did not work in a new context.

I set the scene of antipatterns by giving you two examples of antipatterns. The first is an old one that led to a famous disaster. Late in the evening of April 14, 1912, the RMS *Titanic* hit an iceberg, and in the early hours of April 15, she sank, killing more than 1,500 of the 2,224 people on board, both passengers and crew. To understand why the ship sank, you have to look at a number of small things that, together, led to a disaster. The thing I choose to focus on is the antipattern you could call “following orders from an uninformed superior.” If you look up *superior orders* on

---

1. Observant readers will realize that the name on my dissertation is not the name on this book. It is the name of my first husband, which was also my name at the time. Incidentally, this is also an antipattern: Do Not Change Your Name. At least not after publishing.

Wikipedia, you find, “Superior orders, often known as the Nuremberg defense, lawful orders, just following orders, or by the German phrase *Befehl ist Befehl* (‘an order is an order’), is a plea in a court of law that a person—whether a member of the military, law enforcement, a firefighting force, or the civilian population—not be held guilty for actions ordered by a superior officer or an official.” This antipattern has been identified in numerous places and times, and also, as it happens, in the middle of the tale of the *Titanic*.

The two wireless radio operators on the *Titanic* worked for the Marconi Wireless Telegraph Company. The operators’ orders were to relay passengers’ messages to and from friends and family on land in order to demonstrate the wireless communication service provided by the Marconi Company. It was also a factor that the company was paid for every message to and from the passengers, and thus their income increased by prioritizing them over the ship-to-ship courtesy messages. Almost from the beginning of the voyage, they had received warnings about icebergs and passed most of these messages to the bridge. Unfortunately, some of the messages sent to the *Titanic* were lost because the radio operators focused on following orders from their company. Their superior was the owner of the wireless company, not the captain.

This explains why, when at 9:40 p.m. the *Mesaba*, a ship sailing in the same waters as the *Titanic*, sent a warning of an ice field, the message never reached the bridge. At 10:55 p.m., another nearby cruise liner, the *Californian*, messaged that she had stopped after becoming surrounded by ice, but one of the radio operators on the *Titanic* scolded the *Californian* for interrupting him, since he was busy handling passenger messages. Consequently, the captain was not warned about the ice situation being worse than expected, and thus he continued to sail at full speed. It was not until 11:40 p.m., when an iceberg was spotted from the crow’s nest, that the ship altered its route. The bridge crew started to turn the *Titanic*, but since she was a large ship sailing at high speed, it was too late. The side of the *Titanic* scraped along the iceberg, and the ship ruptured. We know how the story ends.

Often, a pattern in one context can be an antipattern in a different context. In the case of the *Titanic*, for example, the wireless radio operators were obligated to follow the orders of their superiors. However, had they known that the context had changed, that the ship faced an emergency, they would not have followed orders blindly—to do so would have constituted an antipattern.

Patterns also can become antipatterns over time, as technologies and processes change and improve. When a good solution is replaced by a better solution, the original solution can come to be viewed as a bad solution to a recurring problem.

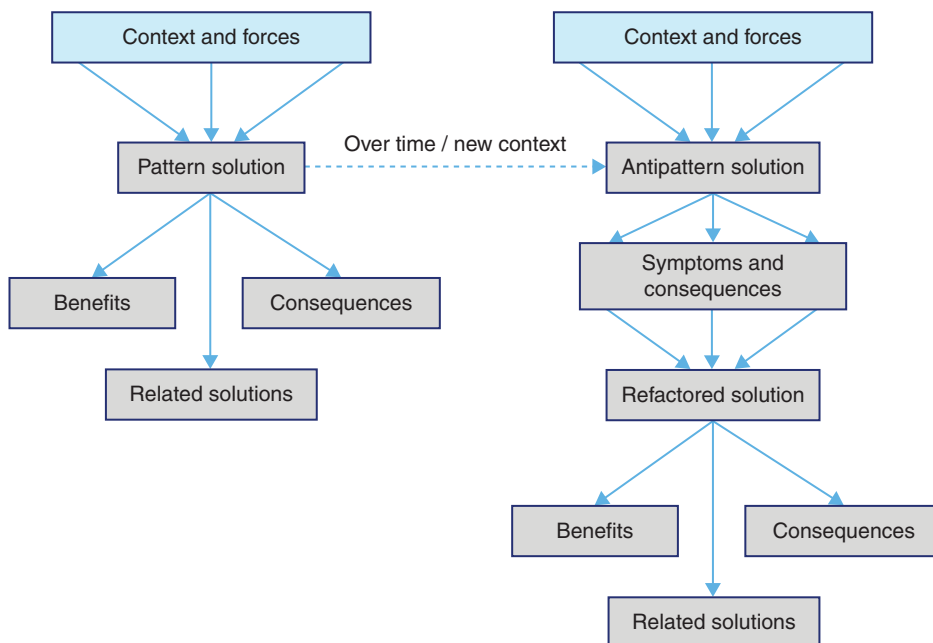
The second example is the Microservices pattern, which is a design described by Martin Fowler and James Lewis (2014) whereby developers create a set of small services, each with its own functionality. The Microservices pattern proved to be maintainable, flexible, and resilient in software architectures, and it was hailed as the best thing since sliced bread (and design patterns). This pattern promoted the development of independently deployable, reusable components, enabling developers to create scalable systems built with microservices. The success of these systems led to the conversion of many monolithic systems to microservices architectures.

What happened then was what sometimes happens with patterns: the pattern was overused.<sup>2</sup> Microservices can have negative consequences if the organization lacks the expertise required to maintain the system, a domain that would benefit from this implementation, or well-defined boundaries between the services. Increased complexity is a major consequence of a microservices architecture, and in the wrong circumstances, the system becomes a more complex monolith instead of a suite of smaller, well-defined microservices. All the expected benefits, such as scalability, independence, and reusability, are lost, and if used in the wrong context, the Microservices pattern becomes an antipattern. The context in

---

2. The Singleton pattern from the GoF book is a brilliant example of pattern overuse.

which you use a pattern is important, and there will always be a context in which the pattern solution does not apply, resulting in an antipattern.



**Figure P.2** A pattern becomes an antipattern when applied in the wrong context

An antipattern described correctly contains a general description, a list of the factors that led to the symptoms and how to recognize them, the consequences of the original solution, and a refactored solution that describes how to solve the current problems or at least how to do better next time.

All patterns have consequences. In some situations, it is a good idea to use a particular pattern, and in others, the same pattern becomes an antipattern. It is important to understand the context-dependent implications when you want to use a pattern. This helps you get the full picture, including the side effects of the antipattern solution. Like a pattern, an antipattern is not some abstract theory a person has invented but a series of causes and effects that he or she sees in often-recurring bad solutions.

You should read this book to learn to recognize antipatterns within retrospectives as (or perhaps even before) they happen to you. My goal in writing *Retrospectives Antipatterns* is to help you avoid making the same mistakes I have made so many times.

As an experienced retrospective facilitator, you may notice that you already know a lot of these patterns and how to deal with them. The added benefit of this book is that now you have a vocabulary to discuss it with other people, and it might be easier for you to recognize when you find yourself in an antipattern. If you share them with your colleagues, their memorable names can help to make you aware when you or your team is slipping into a **Wheel of Fortune** or **Prime Directive Ignorance**.

Lastly, you could read this book for the *schadenfreude*<sup>3</sup> because, as one of the authors of the original antipattern book, *Antipatterns: Refactoring Software, Architectures, and Projects in Crisis* (Brown et al. 1998), said at a presentation when that book was published, “[One’s own] happiness is good, but the misfortune of others is better.”

## HOW TO READ THIS BOOK

### THE OCTOPUS

You might wonder what an octopus has to do with retrospectives. The short answer is: nothing. But that is just the short answer.

I got intrigued by octopuses when I learned about their intelligence. They can learn tricks by watching other octopuses being rewarded for learning tricks without even getting a reward themselves. They can crawl out of an aquarium and through a tiny pipe that leads to the ocean. They can climb out of an aquarium, get down on the floor, cross the floor, mount another table, enter another aquarium, eat all the fish there, and then go back to their own aquarium *as if nothing happened*.

---

3. *Schadenfreude* is a German word meaning that you take pleasure in other peoples’ pain.

Most of all, I am stunned that 60 percent of an octopus's brain is in its eight legs, divided into eight individual little animals, almost, each with its own will and yet all working as a whole with the rest of the octopus in a synergy that is unique in the animal kingdom. I see a team with a facilitator at a retrospective as an octopus: the team and facilitator work together toward a common goal while still being individuals, with individual focus and strengths.

For every antipattern, there is an illustration with an octopus that captures the essence of the antipattern. In the antipattern **In the Soup**, the team works together to lift a weight that is still too heavy, because the problem they are trying to solve is **In the Soup** of things they cannot change but just need to accept. In **Prime Directive Ignorance**, the team starts blaming one person instead of trying to find the faults in the system. In the **Disillusioned Facilitator**, the team mocks the facilitator for trying out activities they find ridiculous.

### THE LITERARY FORM OF ANTIPATTERNS IN THIS BOOK

Based on the literary form for antipatterns found in *Antipatterns* (Brown et al. 1998), I have decided on a specific form to make it easier to read the antipatterns. As you will see, this form fits better with some antipatterns than with others. For example, sometimes the symptoms are obvious, and other times they are subtle and worth describing in detail. For brevity, I have left out the section on forces that you would normally see in a pattern. In my literary form, I have folded the forces into the context and related context descriptions. Forces could be, for example, haste, eagerness to be heard, or lack of people.

**Name:** The names of patterns are important because names allow you to extend your vocabulary about retrospectives and enable you to communicate efficiently with others about those patterns. An interesting thing about giving something a name is that the name can encompass a large set of concepts, processes, and conditions in a succinct way that organizes information so that your brain can quickly access all of the elements associated with that name.

**Context:** This book is written as the learning journey of a retrospective facilitator from a Danish company. The story and the people will be introduced later, but for now it suffices to know that every antipattern will include a description of the context in which we find the antipattern in our story.

**General Context:** This section covers the environment you will be in if this antipattern occurs, a description of what may have led to this problem, and the urge to go for the antipattern solution. This is a more generic way of describing the situation in which you may find yourself tempted to implement the antipattern solution.

**Antipattern Solution:** This section explains the chosen path based on the problem described. The antipattern looked like a solution at the time, based on education, earlier experiences, time constraints, lack of courage, or simply orders from above. It might have been the right solution if it were not for the negative consequences in this context. If you are aware that you can end in this situation by choosing an antipattern solution, you might avoid making some of the numerous mistakes I have made. The antipattern solution is not to be confused with the refactored solution.

**Consequences:** All solutions and decisions have consequences. In the original book about design patterns (Gamma et al. 1995), they were listed as positive and negative consequences, and depending on context, one might outweigh the other. In antipatterns, the point is that the antipattern solution might fit well in another context, but in this context, the negative consequences are much larger than the benefits. I usually say that this listing of consequences is what makes patterns differ from mere methods or recipes found in other books.

**Symptoms:** Symptoms are the indicators enabling you to see that a particular retrospective antipattern is occurring. Symptoms might include comments you hear either outside of or during the retrospective, behaviors you observe, moods you sense, and so on.

**Refactored Solution:** The refactored solution suggests how to improve the current situation so that you and your team are gaining more benefits than negative consequences. In some of the antipatterns, you will learn

that it is not possible to implement the refactored solution once the situation has already arisen, and what you gain from such an antipattern is merely an awareness of how to avoid it next time.

**Online Aspect:** For some of the refactored solutions, there are differences in how you would overcome the challenges in online and offline settings. A growing number of the retrospectives I facilitate are online, so I share my experiences with this medium as well.

**Personal Anecdote:** In this section, I tell of my own experience in sighting the antipattern. Sometimes I found a way to refactor it while it happened, and sometimes it was a lesson learned for the next time.

## OUTLINE OF THE ANTIPATTERNS

### Structural Antipatterns

*Structural antipatterns* describe problems with the structure of the retrospective, such as how the activities are chosen, how the flow of the communication is facilitated, and where a change in agenda might solve the problem either at the present retrospective or at the next retrospective. These are the structural antipatterns:

**Wheel of Fortune** ...in which the team jumps to conclusions in the retrospective by solving symptoms instead of problems, and the facilitator makes the team members spend time on finding the causes behind the symptoms

**Prime Directive Ignorance** ...in which the team members ignore the Prime Directive—“Regardless of what we discover, we understand and truly believe that everyone did the best job they could, given what they knew at the time, their skills and abilities, the resources available, and the situation at hand” (Kerth 2001)—because they find it ridiculous, and the facilitator reminds them how important this mindset is for a successful retrospective

**In the Soup** ...in which team members discuss things that are outside their power to change, and the facilitator helps them focus their energy on what they can change and accept what they cannot change



**Overtime** ...in which the team gets sidetracked at the retrospective by talking about one development that is not the most important for the team as a whole, and the facilitator helps the team get back on track

**Small Talk** ...in which the team members spend time on small talk in small groups instead of focusing on sharing and learning, and the facilitator changes the activities to make them work together as a team again

**Unfruitful Democracy** ...in which, to the frustration of the minority in the team, democracy is used to decide what to discuss and what to do, and the facilitator finds other ways of deciding that makes everyone happier

**Nothing to Talk About** ...in which the team believes it has become so good that it doesn't need retrospectives, and the facilitator shows the team how it can learn to keep improving

**Political Vote** ...in which the team members wait until the last moment to vote in order to game the system, and the facilitator finds a way to make the voting system more fair

### **Planning Antipatterns**

*Planning antipatterns* describe problems with the planning of retrospectives. Whom do you invite to a retrospective? Who should facilitate the retrospective? When should you have a retrospective? How much time should you set aside for it? When you find yourself in a planning antipattern, you cannot change the current retrospective, so you need to be aware to plan differently next time.

**Team, Really?** ...in which the borders of the team are blurred, and the team members all help each other figure out who should attend the retrospective

**Do It Yourself** ...in which the facilitator is wearing several hats, which is suboptimal for both the facilitator and the retrospective, and the team finds other facilitators to take over at times

**Death by Postponement** ...in which the team is so busy with “real work” that the retrospectives are postponed again and again, and the facilitator helps the team see how valuable these retrospectives are and that they are real work

**Get It Over With** ...in which the facilitator rushes through the retrospective in order to “waste” as little time as possible for the team, and the facilitator finally decides that to have a decent retrospective, sufficient time must be allowed for discussions

**Disregard for Preparation** ...in which the facilitator initially misjudges how much preparation an online retrospective requires and later learns how to prepare for it wisely

**Suffocating** ...in which team members get tired and hungry and unfocused during the retrospective, and the facilitator makes sure to feed them and give them oxygen so that they can concentrate a bit more

**Curious Manager** ...in which a manager is curious about what happens at the retrospectives and wants to listen in on them, and the facilitator, in a nice but firm way, says no to the manager

**Peek-A-Boo** ...in which team members will not show their faces on the video in an online retrospective, and the facilitator learns why and finds ways to make it safer for people to show their faces

### **People Antipatterns**

*People antipatterns* describe problems with the people in the retrospectives. You often cannot anticipate these antipatterns because they can occur quite suddenly. Knowing the people will help you be aware of these antipatterns, and the refactored solutions described for these situations can help you navigate out of or around these antipatterns.

**Disillusioned Facilitator** ...in which the team mocks the facilitator for using ridiculous activities, and the facilitator explains why the activities are useful

**Loudmouth** ...in which a team member needs to hear him- or herself all the time, at everyone else’s expense, and the facilitator applies various tactics to ensure the rest of the team is heard

**Silent One** ...in which a team member chooses to be almost completely quiet, and the facilitator applies various tactics to make sure the Silent One is heard

**Negative One** ...in which one team member's attitude has great negative impact on a retrospective, and the facilitator shields the other team members from the negativity

**Negative Team** ...in which the team wants to talk only about the negative things because they think these are the only things they can learn from, and the facilitator shows them that a focus on positive aspects can be equally valuable

**Lack of Trust** ...in which the team members do not trust each other enough to share anything of importance in the retrospective, and the facilitator helps them build that trust

**Different Cultures** ...in which the assumptions the facilitator or the team members bring from their own culture are preventing them from seeing how the retrospective is experienced by others, and the facilitator finds ways to make them more aligned

**Dead Silence** ...in which the team members are completely silent, often in an online retrospective, and the facilitator uses various tactics to hear their opinions despite their reluctance to participate

Register your copy of *Retrospectives Antipatterns* on the InformIT site for convenient access to updates and/or corrections as they become available. To start the registration process, go to [informit.com/register](http://informit.com/register) and log in or create an account. Enter the product ISBN (9780136823360) and click Submit. Look on the Registered Products tab for an Access Bonus Content link next to this product, and follow that link to access any available bonus materials. If you would like to be notified of exclusive offers on new editions and updates, please check the box to receive email from us.

# ACKNOWLEDGMENTS

---

Foremost, I would like to thank my husband, Erik Corry, for believing in me and letting me spend months without income just to focus on the writing. On top of this, he also patiently (mostly) debugged my entire manuscript more than once. Any errors that remain are, of course, caused by stray cosmic rays hitting my computer. I am grateful to my three wonderful and interesting children, Maja, Sophus, and Vera, for doing their chores so that I had more time to write.

I also want to thank Mai Skou Wihlborg for putting me in a “writer’s prison” in her summer house and feeding me so that I could concentrate on writing. She is also the photographer behind the picture on the back cover.

Thank you to Kari Rye Schougaard, Simon Hem Pedersen, Henrik Madsen, and Jakob Roesgaard Færch, who, in the very early days of the book, listened to my ideas and suggested improvements.

I also want to thank the cozy coffee shop Hos Sofies Forældre in Aarhus, Denmark, where I have written most of my chapters while sipping decaf

cappuccino or getting a guilty pleasure from eating their wonderful burger with fries and chili mayo.

At Pearson, my editors Greg Doench and Menka Mehta helped navigate many issues in getting this book to publication. And thank you to Martin Fowler for making such a great introduction of me to Pearson.

I want to thank the people who took time to review my book and provide me with feedback, which improved the book immensely: Gary Harvey, Gregor Hohpe, Henrik Bærbak Christensen, Jimmy Nilsson, Joseph Pelrine, Jutta Eckstein, Karl Krukow, Linda Rising, and Toby Corballis. At RFG 2019 Therese Hansen encouraged me to share my unfinished book with the other participants, and their positive reaction made me share the first edition on Leanpub.

This book is masterly illustrated by Nikola Korać, who listened to all my ideas and made something much better than what I imagined in my head.

Finally, a big thank-you to all the people who dislike retrospectives, try to avoid them, and even sometimes destroy them. Without you, this book would not have existed.

# ABOUT THE AUTHOR

---



**Aino Vonge Corry** is an independent consultant who sometimes works as an agile coach. After gaining her PhD in computer science (CS) in 2001, she spent the next 10 years failing to choose between being a researcher/teacher in academia and being a teacher/facilitator in industry. She eventually squared the circle by starting her own company, Metadeveloper, which develops developers by teaching CS, teaching how to teach CS, inviting speakers to IT conferences, and facilitating software development in various ways. In her youth, she thought she would be a schoolteacher in mathematics, but the strange ways of the world moved her toward communication and facilitation, mainly in the software industry. She has facilitated for the past 15 years, during which time she has made all the mistakes possible in that field. And she would like to share the stories of those mistakes, and their resolutions, as a warning and a guide to other people wanting to facilitate meetings.

Aino has spent time in Stockholm and Cambridge, but she is now back in Aarhus, Denmark, where she lives with her family and a growing collection of plush cephalopods.

To get in touch, email [aino@metadeveloper.com](mailto:aino@metadeveloper.com), tweet [@apaiipi](https://twitter.com/apaiipi), or visit [metadeveloper.com](http://metadeveloper.com).

*This page intentionally left blank*

# The Story Begins

At Titanic Software A/S, a fictional Danish company specializing in navigation software for ships, changes must be made. The customers have been complaining about bugs in the software that make their ships sail to the wrong destinations, and the speed of software development is not satisfactory.

The chief technology officer (CTO) and some developers decide to go to a software conference to learn what other companies are doing to stay ahead of the game. They find a new approach called agile development, which promises to speed up development, reduce defects, and better align their software with the needs of their customers. In other words, this new approach might take them to where they want to be: rich and happy.

The CTO sends everybody to a scrum master course. Let us zoom in on one little team consisting of six people: Bo, Peter, Rene, Kim, Sarah, and Andrea. Sarah, who was previously the project leader, is the one most eager to become a scrum master, and since eagerness is an important quality in filling this role, she is named scrum master for the team.

Job titles have to be adjusted to the agile way of doing things. Peter, who was the business analyst, becomes the product owner. In her new role as scrum master, Sarah is, among other things, responsible for making sure the daily standup meeting takes place. When issues come up during the daily standup that require talking to others in the company, Sarah makes sure to do that. Also, as scrum master, she is the master of ceremonies and, as such, makes sure that the team has regular retrospectives.



# Structural Antipatterns

**Wheel of Fortune** ...in which the team jumps to conclusions in the retrospective by solving symptoms instead of problems, and the facilitator makes the team members spend time on finding the causes behind the symptoms

**Prime Directive Ignorance** ...in which the team members ignore the Prime Directive—“Regardless of what we discover, we understand and truly believe that everyone did the best job they could, given what they knew at the time, their skills and abilities, the resources available, and the situation at hand” (Kerth 2001)—because they find it ridiculous, and the facilitator reminds them how important this mindset is for a successful retrospective

**In the Soup** ...in which team members discuss things that are outside their power to change, and the facilitator helps them focus their energy on what they can change and accept what they cannot change

**Overtime** ...in which the team gets sidetracked at the retrospective by talking about one development that is not the most important for the team as a whole, and the facilitator helps the team get back on track

# Part I

**Small Talk** ...in which the team members spend time on small talk in small groups instead of focusing on sharing and learning, and the facilitator changes the activities to make them work together as a team again

**Unfruitful Democracy** ...in which, to the frustration of the minority in the team, democracy is used to decide what to discuss and what to do, and the facilitator finds other ways of deciding that makes everyone happier

**Nothing to Talk About** ...in which the team believes it has become so good that it doesn't need retrospectives, and the facilitator shows the team how it can learn to keep improving

**Political Vote** ...in which the team members wait until the last moment to vote in order to game the system, and the facilitator finds a way to make the voting system more fair

# Wheel of Fortune

*... in which the team jumps to conclusions in the retrospective by solving symptoms instead of problems, and the facilitator makes the team members spend time on finding the causes behind the symptoms*

# Chapter I



## CONTEXT

When planning her first retrospective, Sarah finds herself short of time. She needs to find a recipe for facilitating one, and her only source of knowledge about how to facilitate a retrospective comes from the scrum master course. The course covered all of the components of the Scrum framework, but its broad scope limited the amount of time spent teaching any one topic, including how to facilitate retrospectives. Sarah decides to do the Start-Stop-Continue activity, where the team brainstorms on what to start doing, what to stop doing, and what to continue doing. She uses this activity as the heart of the retrospective, and in the Decide What to Do phase, the team votes on the three different topics and finds things to start doing, stop doing, and remember to continue doing.

### **Start-Stop-Continue**

For this activity, the facilitator prepares by creating three posters on flip paper or whiteboards, one labeled Start, one labeled Stop, and one labeled Continue. The team members are given Post-it Notes and pens. Some people prefer that all the notes be the same color; others prefer notes of three different colors, such as traffic-light colors, so the notes can be removed from the boards without losing the reference to one of the three choices. The facilitator can also ask people to write Start, Stop, or Continue at the top of the note. The team members are given about 10 minutes to write on the Post-it Notes what they think should be started, stopped, and continued. They then place the Post-it Notes on the boards. The team must now choose their action points based on these, and how that is done depends very much on the facilitator, the content of the Post-it Notes, and the people on the team.

At Sarah's first retrospective, the Start board has a Post-it Note saying "more pair programming." This is an easy thing to address: the team simply makes a schedule requiring everyone to do 3 hours of pair programming each day. Everybody is happy after the retrospective, but in the next sprint, they find that the amount of pair programming being done has not increased significantly. In the first week, Bo and Peter did some pair

programming—though not the 15 hours the new schedule called for. Kim and Rene did even less pair programming. Both seem uncomfortable with it, so neither wants to initiate it.

## GENERAL CONTEXT

As a retrospective facilitator, you sometimes encounter people with the attitude that a retrospective takes time away from real work. This viewpoint might persuade you to spend as little time as possible on the retrospective in order to make everyone happy, so a quick, fix-it-all activity seems like the best idea.

## ANTIPATTERN SOLUTION

The easiest way to do a retrospective is to put up three posters labeled Stop, Start, and Continue. The next step is to have the team brainstorm, write on Post-it Notes, and put the notes on the appropriate poster. The last step is even easier. You start doing the items listed on the Start poster, stop doing those on the Stop poster, and continue those on the Continue poster. The retrospective is finished in 15 minutes, and the team can go back to their “real” work. Even if you added a Do More Of poster, everyone would be back at their desks in short order.

Actually, this approach is not the easiest way to run an effective retrospective. I have heard of “retrospectives” (and I use the term *retrospective* loosely here) where the team went into a meeting room, the scrum master or the project leader asked if anything needed to be changed, and when no one said anything, they all went back to work.

## CONSEQUENCES

The benefit of this antipattern solution is that it is a very fast way of conducting a retrospective, but the negative consequences can be severe. Sometimes, you are lucky and the Post-it Notes identify the real problems. To fix them, the team simply stops doing the Stop items and does more of the

Do More Of items. But often, the Post-it Notes merely describe the symptoms of bigger problems that need a more fundamental change than just addressing the symptoms can provide, so you need to stop the symptoms from occurring in the first place by removing the underlying problem.

In our example from Titanic Software A/S, this antipattern works fine if the only reason for not doing more pair programming is that people simply forget and revert to earlier habits. However, the team's difficulty may well be a symptom of a problem that is more interesting and harder to solve.

Had they spent time during the retrospective exploring the cause of the problem, they might have found reasons that the team was not doing more pair programming. For example, perhaps the developers couldn't see the benefit of it and had never been asked if they wanted to do pair programming. Or perhaps they saw the benefit but most of them were introverted and thus needed hours on their own to reflect on information before sharing it with their coworkers. Another reason might be that the team members don't know how to do pair programming, so they need to learn how to do it right before they can actually start doing it. On our little team, where Rene is a somewhat negative **Loudmouth** and Kim is a **Silent One**, there could also be a lack of psychological safety. None of these problems can be solved just by forcing people to work together.

## SYMPTOMS

If you start hearing things like “Why do we always talk about this at the retrospective?” and “The retrospectives aren't working—nothing changes except the color of the problems,” then you should wonder if you are in a **Wheel of Fortune**.

As in a real-world wheel of fortune, you sometimes hit a win when you turn the wheel—if the wheel lands on the actual problem and not symptoms of the problem, that is. And the chance of that happening is about the same as if you spun a wheel of fortune at a fair. Consequently, another symptom of this antipattern is that you hear the same issues being

discussed repeatedly at the retrospectives for the simple reason that you never solve the problems, you just put Band-Aids on the symptoms.

## REFACTORED SOLUTION

It is tempting to go directly from problems to solutions, and most developers are trained to do exactly that. Unfortunately, since the issues that come up at the retrospectives are sometimes not understood immediately, you need to examine them before you start working on the solutions. After data is collected, you need to look at the causes behind the problems. This is called the Generate Insights phase, as described in the introduction, and it can't be skipped.

Several activities can be applied in this stage. A simple one is to ask for the story behind the Post-it Note in order to learn about what led to this issue. Other activities are the Fishbone (see Figure 1.1 and description) and 5 Hows. Previously, I used an activity called 5 Whys to make the cause analysis, but because of insights I gained from John Allspaw (2014), I now use 5 Hows.

### 5 Hows

*5 Hows* is an iterative interrogative technique used to explore the cause-and-effect relationships underlying a particular problem. The primary goal of the technique is to determine the root cause of a defect or problem by repeatedly asking, How did this happen? Each answer forms the basis of the next question. The five in the name derives from an anecdotal observation on the number of iterations needed to resolve the problem.

Few problems have a single root cause. To uncover multiple root causes, the method can be repeated, asking a different sequence of questions each time.

The method provides no hard-and-fast rules about what lines of questions to explore or how long to continue the search for additional root causes. Thus, even when the method is closely followed, the outcome still depends on the knowledge and persistence of the people involved.



Allspaw (2014) explains that asking why can lead to trying to find one cause for a problem and even to blaming. Asking how instead could lead to a narrative, a number of causes. Making it a blame game is obviously not desired, and searching for one cause in a system as complex as software development and teamwork is optimistic bordering gullible. You could even do this on a personal level: try asking yourself how instead of why next time something doesn't go the way you intended.

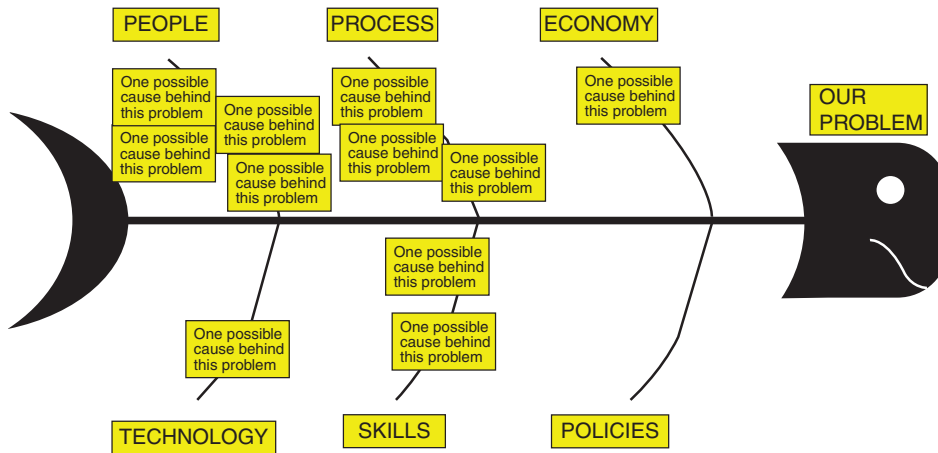


Figure 1.1 Fishbone activity for finding several causes

## Fishbone

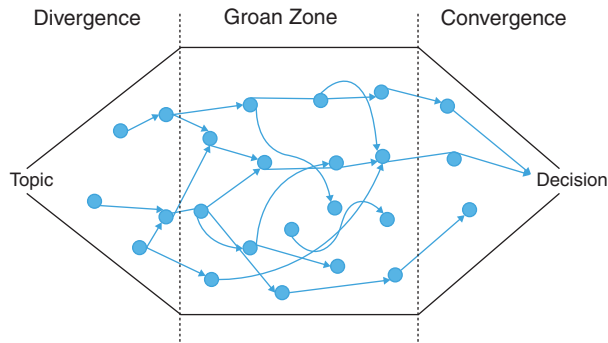
A Fishbone is a cause-analysis activity in which the facilitator draws a fish skeleton with different themes, such as People, Process, Economy, and Technology, as the ribs to visualize the different types of causes behind a problem. Then the facilitator asks team members to write Post-it Notes for the different causes they can think of and to place them on the fishbone. The result is a fish displaying many different ideas about what might be causing the problem. Perhaps you'll find that most of the ideas cluster in a particular category, such as process or technology, and that gives you input for what cause to work on. This technique is generally used in the Generate Insights phase of a retrospective. It is also called an *Ishikawa diagram analysis*.<sup>1</sup>

1. [https://en.wikipedia.org/wiki/Ishikawa\\_diagram](https://en.wikipedia.org/wiki/Ishikawa_diagram)

All Generate Insights activities for finding causes involve digging down into what is written on the Post-it Notes to find the real stories behind them.

Every meeting, including a retrospective, has a life cycle of divergence and convergence (see Figure 1.2), as described in *Facilitator's Guide to Participatory Decision-Making* by Sam Kaner (2007). We start, hopefully, in a small space of understanding the purpose of the meeting and the outcomes we expect; at the least, the topic of the meeting should be agreed upon. Then we go into the divergent phase, where discussions open up for disagreements and clashes of opinion. Next comes the groan zone, where people further discuss the issues not in search of agreement but to learn more about the topic. After spending some time in the groan zone, the meeting needs to go into a convergence phase where we try to narrow down the scope and/or find agreement. If agreement cannot be reached, at least the group can agree to disagree and establish the terms and grounds of the disagreement.

To start with, the agreement on the topic resembles the Set the Stage phase of the retrospective. The divergence part of a retrospective is the Gather Data and Generate Insights phases. It is not an easy part of a retrospective but a very important one. The groan zone can be related to the last part of the Generate Insights phase and the start of the Decide What to Do phase. Convergence is the essence of the Decide What to Do phase, and when you reach the decision at the end of the retrospective, you Close the Retrospective. Make sure to go through all five phases of a retrospective and generate insights instead of jumping to conclusions and thus getting premature convergence just to get to the point quickly.



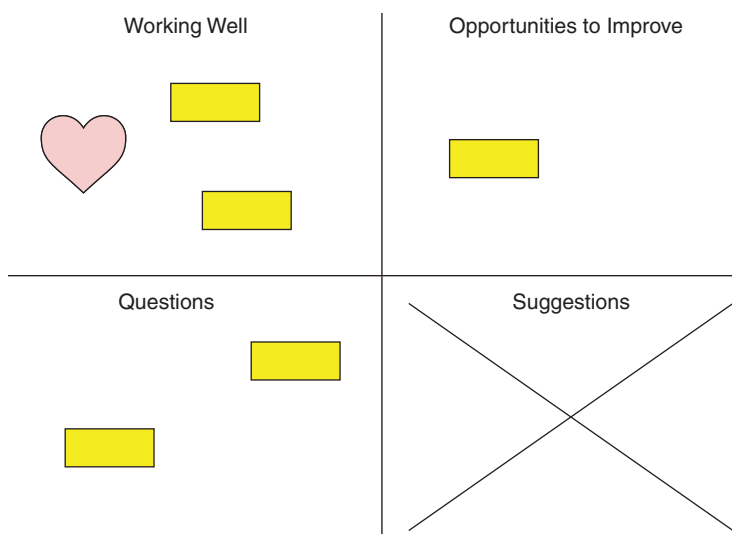
---

**Figure 1.2** Divergence and convergence in a meeting

## ONLINE ASPECT

If the retrospective is done online, you can use an online tool that forces the team to enter and stay in the Generate Insights phase before they move on to problem solving. If you are creating your own online document, you can visualize what phase of the retrospective you are in at all times. I often use a Google Drawings diagram for the Generate Insights phase and cross out the suggestions box. In that way, team members are visually reminded not to add a suggestion for an experiment before we get to the Decide What to Do phase (see Figure 1.3).

You can also visualize the phase of the retrospective by wearing a different hat for each phase. If people still forget not to jump to conclusions, you can point at your hat and smile.



**Figure 1.3** An online retrospective document with the suggestion box crossed out

## PERSONAL ANECDOTE

I was facilitating a team in a company in Denmark that had been acquired by a large organization from abroad. The company had its own way of doing things and liked it like that. Unfortunately, the new organization decided to divide the original company into three teams. At one of the retrospectives not long after the acquisition, one of the things that was put on the board was “less work in silos.” We all understood what that meant: the team members wanted to work together instead of as three separate teams working in parallel without helping each other.

Since we understood the issue, it was the action point they decided, without much discussion, to do at that retrospective. And at the next retrospective. And the next retrospective. Finally, we decided to look at the cause of the siloing. It turned out the key performance indicators made for each team got in the way of cooperation. Since every team had its own goals, it became a zero sum game where one team’s win was based on another team’s loss. Also the planning meetings for the teams were with

different people in the new organization. It would have been good to acknowledge that from the beginning, to understand whether this was something they could change or something they would have to learn to live with. (See also Chapter 3, **In the Soup**.)

Now for something completely different: working mainly with IT people, I always find it hard to make people stay in the open discussion, or the groan zone. It feels almost physically hard to keep the options open before I allow them to rush to the solution phase. All minutes spent in the open discussion phase are well spent, though, in my experience. This is where new insights and ideas often come up. Dave Snowden has written often about premature convergence as a part of complexity theory, and he describes what we might miss out on: “holding things open, allowing them to break down into more finely grained objects, then seeing them recombine and co-evolve” (Snowden 2015).

When I had considerable experience with IT teams, I was asked to do idea generation for people working at an art museum. I was not prepared for what happened; it was extremely easy to get them into the open phase where everything is possible and where they should not fixate on a solution or track. I felt that it was going very well and that I had done a good job of getting them into that way of working.

But then my problems started. It was next to impossible to get them to make decisions and to cut down on ideas and remove possibilities. I found myself running from group to group and peeling down Post-it Notes, which they put up again the moment I turned toward another group. When I heard someone say, “Should we even have an art museum?” I felt a slight panic, since we were here to brainstorm for an app, not have an existential discussion of whether we needed an art museum. I had to adapt to this new crowd, and I had to do it quickly.

I interpreted the situation as working with a different species, almost. I had to be much more direct than was necessary with other groups I'd worked with, and I had to spend more energy in making this group go to the last phase of a meeting and get to some kind of convergence. In the end, we succeeded in agreeing on a focus for the next period only by promising that the other ideas would be kept for the future. It is interesting how cultural this behavior, or thought pattern, is. This is also described in Chapter 23, **Different Cultures**. In some cultures, dealing with conflict or differences of opinion is something you try to avoid. People from such a culture (and it could be within a specific country, company, or team) will seek consensus as quickly as possible, thus perhaps losing the chance for novel ideas to arise from their interaction.

## Numbers

1–2-3 vote method, 79  
1, 2, 4, All technique, 171  
5 Whys, 9–10

## A

Action points, 111–112  
activation phenomenon, 170, 214  
agile development, job titles, 1  
Agile Fluency, 70, 110  
*Agile Retrospectives: Making Good Teams Great*  
(2006), 18, 103, 170  
Allspaw (2014), 10  
amygdalae, 195  
anonymity  
    Different Cultures, 205–206  
    anonymous voting, time management, 41  
antipattern solutions  
    Curious Manager, 133  
    Dead Silence, 213  
    Death by Postponement, 100  
    Different Cultures, 204  
    Disillusioned Facilitator, 151  
    Disregard for Preparation, 118  
    Do It Yourself, 92–93  
    In The Soup, 29  
    facilitating retrospectives, 7–8  
    Get it Over With, 108–109  
    Lack of Trust, 191

Loudmouths, 158–159  
Negative Ones, 176  
Negative Teams, 184  
Nothing to Talk About, 64  
Peek-A-Boo, 139  
    planning retrospectives, attendance, 85  
Political Voting, 77  
Prime Directive, 20–21  
Silent Ones, 169  
small talk, 48  
Start-Stop-Continue, 7–8  
Suffocating Slowly, 127  
time management, 39  
Unfruitful Democracies, 57  
attendance, planning retrospectives  
antipattern solutions, 85  
    consequences, 85–86  
    context, 84–85  
    online retrospectives, 87  
    personal anecdotes, 88  
    refactored solutions, 86–87  
    symptoms, 86  
avatars, 121

## B

Ben Franklin effect, 142–143  
benevolence, topology of trust, 194  
blame  
    5 Whys, 10  
    facilitating retrospectives, 10

body language, Lack of Trust, 192  
 Breakers, 159–160  
 breaks, taking, 127

**C**

Champion Skeptics, 178–179  
*The Checklist Manifesto* (2011), 170  
 Circles and Soup activity, 30–34  
 circles of influence, 30–34  
 cognitive dissonance, 142  
*Collaboration Explained* (2006), 94  
 competence, topology of trust, 194  
 conferences (video)  
     avatars, 121  
     Disregard for Preparation  
         antipattern solutions, 118  
         consequences, 118–119  
         context, 116–117  
         Google Drawings, 120  
         message boards, 120  
         online retrospectives, 122  
         personal anecdotes, 123  
         refactored solutions, 119–121  
         reminders, 120  
         round robins, 121–122  
         shared documents, 118–119  
         symptoms, 119  
     message boards, 120  
 Peek-A-Boo  
     antipattern solutions, 139  
     Ben Franklin effect, 142–143  
     cognitive dissonance, 142  
     consequences, 139–140  
     context, 138  
     personal anecdotes, 143–145  
     refactored solutions, 140–143  
     Six Thinking Hats, 145  
     symptoms, 140  
     proxies, 121  
     reminders, 120  
     shared documents, 118–119  
 consensus decision making, 58  
 context, facilitating retrospectives, 7  
 continuous retrospective timelines, 102–103  
 convergence  
     facilitating retrospectives, 11–12  
     Start-Stop-Continue, 11–12  
 CTO (Chief Technology Officers), 1  
 cultures  
     Different Cultures  
         anonymity, 205–206  
         antipattern solutions, 204  
         building trust, 206  
         consequences, 205  
         context, 204

        Lack of Trust, 206  
         online retrospectives, 207  
         personal anecdotes, 207–208  
         refactored solutions, 205–206  
         symptoms, 205  
     of trust, 197  
 Curious Manager antipattern  
     antipattern solutions, 133  
     consequences, 133  
     context, 132  
     Do It Yourself, 97  
     online retrospectives, 134  
     personal anecdotes, 134–135  
     refactored solutions, 133–134  
     symptoms, 133  
 Cynefin framework, 121

**D**

de Bono, Edward, 56–57  
 Dead Silence antipattern  
     activation phenomenon, 214  
     antipattern solutions, 213  
     consequences, 213  
     context, 212–213  
     Google Drawings, 216–217  
     online retrospectives, 212, 216  
     personal anecdotes, 216–218  
     refactored solutions, 214–215  
     round robins, 214  
     symptoms, 214  
 Death by Postponement antipattern  
     antipattern solutions, 100  
     consequences, 101  
     context, 100  
     online retrospectives, 103  
     personal anecdotes, 104–105  
     refactored solutions, 102–103  
     symptoms, 101  
 debriefing, time management, 40  
 Decide What to Do phase, 11–13, 40–41  
 defining trust, 193  
 Democracies antipattern (Unfruitful)  
     antipattern solutions, 57  
     consensus decision making, 58  
     consequences, 57–58  
     context, 56  
     dictatorships, 58  
     dot voting (dotmocracies), 58  
     minority vote, 59  
     online retrospectives, 60  
     parallel thinking, 57  
     personal anecdotes, 61  
     refactored solutions, 58–60  
     Six Thinking Hats, 56–57  
     symptoms, 58



- timeboxes, 60
  - UX/UI experts, 61
  - Derby, Esther, 65
  - DevOps, 72–73
  - dictatorships, 58
  - Different Cultures antipattern
    - anonymity, 205–206
    - antipattern solutions, 204
    - building trust, 206
    - consequences, 205
    - context, 204
    - Lack of Trust, 206
    - online retrospectives, 207
    - personal anecdotes, 207–208
    - refactored solutions, 205–206
    - symptoms, 205
  - Disillusioned Facilitators antipattern
    - antipattern solutions, 151
    - consequences, 151
    - context, 150–151
    - online retrospectives, 153
    - personal anecdotes, 153–154
    - refactored solutions, 152
    - symptoms, 151
  - Disregard for Preparation antipattern
    - antipattern solutions, 118
    - consequences, 118–119
    - context, 116–117
    - Google Drawings, 120
    - message boards, 120
    - online retrospectives, 122
    - personal anecdotes, 123
    - refactored solutions, 119–121
    - reminders, 120
    - round robins, 121–122
    - shared documents, 118–119
    - symptoms, 119
  - divergence
    - facilitating retrospectives, 11–12
    - Start-Stop-Continue, 11–12
  - Do Food pattern, 128
  - Do It Yourself antipattern
    - antipattern solutions, 92–93
    - consequences, 93
    - context, 92
    - Curious Manager, 97
    - facilitator rotations, 96–97
    - online retrospectives, 96
    - personal anecdotes, 96–97
    - refactored solutions, 94–95
    - symptoms, 93
  - Docs (Google), 207–208
  - documents (shared), Disregard for Preparation, 118–119
  - dot voting (dotmocracies), In the Soup
    - antipattern, 58
    - antipattern solutions, 29
    - Circles and Soup activity, 30–34
    - consequences, 29
    - context, 28
    - defined, 28
    - general context, 28
    - influence, circles of, 30–34
    - online retrospectives, 34
    - personal anecdotes, 34–35
    - problems, 28
    - refactored solutions, 30–34
    - symptoms, 30
  - Drawings (Google)
    - anonymity, different cultures, 205–206
    - Dead Silence, 216–217
    - Disregard for Preparation, 120
    - online retrospectives, 12
    - Political Voting, 76, 79
    - team radar retrospectives, 76
- E**
- Eckstein, Jutta, 60
- F**
- facilitating retrospectives
    - 5 Whys, 9–10
    - antipattern solutions, 7–8
    - blame, 10
    - consequences, 7–8
    - context, 7
    - convergence, 11–12
    - Decide What to Do phase, 11–13
    - divergence, 11–12
    - General Insights phase, 9
      - 5 Whys, 9–10
      - convergence, 11–12
      - divergence, 11–12
      - groan zones, 11
    - Generate Insights phase, 12
    - groan zones, 11
    - personal anecdotes, 13–15
    - refactored solutions, 9–12
    - Set the Stage phase, 11
    - Start-Stop-Continue, 6
      - antipattern solutions, 7–8
      - consequences, 6
      - context, 7
      - General Insights phase, 9–12
      - refactored solutions, 9–12
      - symptoms, 8–9
      - Wheel of Fortune, 8–9
    - symptoms, 8–9
    - Wheel of Fortune, 8–9, 19
  - Facilitating with Ease!* (2005), 94

facilitator rotations, 96–97  
*Facilitators' Guide to Participatory Decision-Making* (2007), 11, 94  
*Fearless Change: Patterns for Introducing New Ideas* (2005), 128, 178  
 feet (body language), Lack of Trust, 192  
 Fishbones, 10  
 Franklin, Benjamin, 142–143  
 futurespectives, 66–67, 207

**G**

general context, facilitating retrospectives, 7  
 General Insights phase, 9  
   5 Whys, 9–10  
   convergence, 11–12  
   divergence, 11–12  
   groan zones, 11  
 Generate Insights phase, 12  
 Get it Over With antipattern  
   Agile Fluency, 110  
   antipattern solutions, 108–109  
   consequences, 109  
   context, 108  
   online retrospectives, 110  
   personal anecdotes, 111–113  
   refactored solutions, 109–110  
   regretpectives, 109  
   symptoms, 109  
*Getting Value Out of Agile Retrospectives: A Toolbox of Retrospective Exercises* (2014), 68  
 Goncalves, Luis, 132  
 Google Docs, 207–208  
 Google Drawings  
   anonymity, different cultures, 205–206  
   Dead Silence, 216–217  
   Disregard for Preparation, 120  
   online retrospectives, 12  
   Political Voting, 76, 79  
   team radar retrospectives, 76  
 Google Hangout, 145  
 groan zones  
   facilitating retrospectives, 11  
   Start-Stop-Continue, 11  
 ground rules, small talk, 49–50  
*Groundhog Day*, 29

**H**

Hangout (Google), 145  
 Hawthorne effect, 192  
 Hem, Simon, 187

**I**

influence, circles of, 30–34  
 integrity, topology of trust, 194  
 In The Soup antipattern, 58  
   antipattern solutions, 29  
   Circles and Soup activity, 30–34  
   consequences, 29  
   context, 28  
   defined, 28  
   general context, 28  
   influence, circles of, 30–34  
   online retrospectives, 34  
   personal anecdotes, 34–35  
   problems, 28  
   refactored solutions, 30–34  
   symptoms, 30  
 invitations, planning retrospectives  
   antipattern solutions, 85  
   consequences, 85–86  
   context, 84–85  
   online retrospectives, 87  
   personal anecdotes, 88  
   refactored solutions, 86–87  
   symptoms, 86  
 Ishikawa diagram analysis. *See* Fishbones

**J**

job titles, agile development, 1

**K**

Kahneman, Daniel, 42  
 Kaner, Sam, 11  
 Kerth, Norm, 18  
   Get it Over With, 110  
   Offer Appreciations exercise, 153  
   Prime Directive, 16, 18, 20

**L**

Lack of Trust antipattern, 19–20  
   antipattern solutions, 191  
   body language, 192  
   consequences, 191  
   context, 190–191  
   culture of trust, 197  
   defining trust, 193  
   Different Cultures, 206  
   Hawthorne effect, 192  
   Measure Safety activity, 190  
   online retrospectives, 197  
   Performance versus Trust, 197

- personal anecdotes, 197–200
  - QuizBreaker, 195–196
  - refactored solutions, 192–197
  - symptoms, 191–192
  - team performance, 195–197
  - temperature readings, 190
  - topology of trust, 194
  - trusting stance, 194
  - Two Truths and a Lie, 195–196
  - Larsen, Diana, 65, 70
  - Line Managers Want to Attend* (2019), 132
  - Loudmouths antipattern, 8
    - antipattern solutions, 158–159
    - Breakers, 159–160
    - consequences, 159–160
    - context, 158
    - online retrospectives, 162
    - personal anecdotes, 162–164
    - refactored solutions, 160–162
    - Storytellers, 159–161
    - symptoms, 160
- M**
- Making Good Teams Great* (2006), 65
  - managers, Curious Manager antipattern
    - antipattern solutions, 133
    - consequences, 133
    - context, 132
    - Do It Yourself, 97
    - online retrospectives, 134
    - personal anecdotes, 134–135
    - refactored solutions, 133–134
    - symptoms, 133
  - managing time. *See also* small talk
    - anonymous voting, 41
    - antipattern solutions, 39
    - consequences, 39–40
    - context, 38
    - debriefing, 40
    - Decide What to Do phase, 40–41
    - general context, 39
    - Kahneman, Daniel, 42
    - online retrospectives, 43
    - parking lots, 41
    - personal anecdotes, 43–44
    - positivity retrospectives, 44
    - refactored solutions, 40–42
    - symptoms, 40
    - TCI, 44
    - time between retrospectives, 102–103
    - timeboxes, 60
  - Mann, Mary Lynn, 128
  - Measure Safety activity, 190
  - message boards, video conferencing, 120
  - minority vote, 59
  - Morgan, J.P. 65
- N**
- Negative Ones antipattern, 176
    - antipattern solutions, 176
    - Champion Skeptics, 178–179
    - consequences, 177
    - online retrospectives, 179
    - personal anecdotes, 180–181
    - Prime Directive, 178
    - refactored solutions, 177–179
    - symptoms, 177
  - Negative Teams antipattern, 44
    - antipattern solutions, 184
    - consequences, 185
    - context, 184
    - online retrospectives, 187
    - personal anecdotes, 187
    - positivity retrospectives, 186
    - refactored solutions, 186–187
    - symptoms, 185
  - Never Split the Difference: Negotiating as If Your Life Depended on It* (2017), 198
  - Niebuhr, Reinhold, 31
  - Nothing to Talk About antipattern
    - Agile Fluency, 70
    - antipattern solutions, 64
    - consequences, 65
    - context, 64
    - DevOps, 72–73
    - futurespectives, 66–67
    - online retrospectives, 70
    - personal anecdotes, 71–73
    - positivity retrospectives, 66
    - refactored solutions, 65–70
    - Ship retrospective, 67–68
    - symptoms, 65
    - team radar retrospectives, 66
- O**
- Offer Appreciations exercise, 153
  - online retrospectives, 197
    - attendance, 87
    - Curious Manager, 134
    - Dead Silence, 212, 216
    - Death by Postponement, 103
    - Different Cultures, 207
    - Disillusioned Facilitator, 153
    - Disregard for Preparation, 122
    - Do It Yourself, 96

- Generate Insights phase, 12
  - Get it Over With, 110
  - Google Drawings, 12
    - In the Soup, 34
  - Loudmouths, 162
  - Negative Ones, 179
  - Negative Teams, 187
  - Nothing to Talk About, 70
  - Peek-A-Boo
    - antipattern solutions, 139
    - Ben Franklin effect, 142–143
    - cognitive dissonance, 142
    - consequences, 139–140
    - context, 138
    - personal anecdotes, 143–145
    - refactored solutions, 140–143
    - Six Thinking Hats, 145
    - symptoms, 140
  - Political Voting, 79
  - Prime Directive, 23
  - Silent Ones, 171
  - small talk, 51
  - Suffocating Slowly, 128
  - time management, 43
  - Unfruitful Democracies, 60
  - Overtime antipattern. *See also* small talk
    - anonymous voting, 41
    - antipattern solutions, 39
    - consequences, 39–40
    - context, 38
    - debriefing, 40
    - Decide What to Do phase, 40–41
    - general context, 39
    - Kahneman, Daniel, 42
    - online retrospectives, 43
    - parking lots, 41
    - personal anecdotes, 43–44
    - positivity retrospectives, 44
    - refactored solutions, 40–42
    - symptoms, 40
    - TCI, 44
    - time between retrospectives, 102–103
    - timeboxes, 60
- P**
- parallel thinking, 57
  - parking lots, time management, 41
  - pedagogical patterns. *See* round robins
  - Peek-A-Boo antipattern
    - antipattern solutions, 139
    - Ben Franklin effect, 142–143
    - cognitive dissonance, 142
    - consequences, 139–140
    - context, 138
    - personal anecdotes, 143–145
    - refactored solutions, 140–143
    - Six Thinking Hats, 145
    - symptoms, 140
  - Pelrine, Joseph, 121
  - people antipatterns
    - Dead Silence
      - activation phenomenon, 214
      - antipattern solutions, 213
      - consequences, 213
      - context, 212–213
      - Google Drawings, 216–217
      - online retrospectives, 212, 216
      - personal anecdotes, 216–218
      - refactored solutions, 214–215
      - round robins, 214
      - symptoms, 214
  - Different Cultures
    - anonymity, 205–206
    - antipattern solutions, 204
    - building trust, 206
    - consequences, 205
    - context, 204
    - Lack of Trust, 206
    - online retrospectives, 207
    - personal anecdotes, 207–208
    - refactored solutions, 205–206
    - symptoms, 205
  - Disillusioned Facilitator
    - antipattern solutions, 151
    - consequences, 151
    - context, 150–151
    - online retrospectives, 153
    - personal anecdotes, 153–154
    - refactored solutions, 152
    - symptoms, 151
  - Lack of Trust
    - antipattern solutions, 191
    - body language, 192
    - consequences, 191
    - context, 190–191
    - culture of trust, 197
    - defining trust, 193
    - different cultures, 206
    - Hawthorne effect, 192
    - Measure Safety activity, 190
    - online retrospectives, 197
    - Performance versus Trust, 197
    - personal anecdotes, 197–200
    - QuizBreaker, 195–196
    - refactored solutions, 192–197
    - symptoms, 191–192
    - team performance, 195–197
    - temperature readings, 190
    - topology of trust, 194

- trusting stance, 194
- Two Truths and a Lie, 195–196
- Loudmouths
  - antipattern solutions, 158–159
  - Breakers, 159, 160
  - consequences, 159–160
  - context, 158
  - online retrospectives, 162
  - personal anecdotes, 162–164
  - refactored solutions, 160–162
  - Storytellers, 159–161
  - symptoms, 160
- Negative Ones, 176
  - antipattern solutions, 176
  - Champion Skeptics, 178–179
  - consequences, 177
  - online retrospectives, 179
  - personal anecdotes, 180–181
  - Prime Directive, 178
  - refactored solutions, 177–179
  - symptoms, 177
- Negative Teams, 44
  - antipattern solutions, 184
  - consequences, 185
  - context, 184
  - online retrospectives, 187
  - personal anecdotes, 187
  - positivity retrospectives, 186
  - refactored solutions, 186–187
  - symptoms, 185
- Silent Ones
  - 1, 2, 4, All technique, 171
  - activation phenomenon, 170
  - antipattern solutions, 169
  - consequences, 169
  - context, 168
  - online retrospectives, 171
  - personal anecdotes, 172–173
  - refactored solutions, 170–171
  - symptoms, 169
  - Think, Pair, Share technique, 170
- Performance versus Trust, 197
- personnel, planning retrospectives
  - antipattern solutions, 85
  - consequences, 85–86
  - context, 84–85
  - online retrospectives, 87
  - personal anecdotes, 88
  - refactored solutions, 86–87
  - symptoms, 86
- planning retrospectives
  - attendance
    - antipattern solutions, 85
    - consequences, 85–86
    - context, 84–85
    - online retrospectives, 87
  - personal anecdotes, 88
  - refactored solutions, 86–87
  - symptoms, 86
- personal anecdotes, 88
- refactored solutions, 86–87
- symptoms, 86
- Curious Manager
  - antipattern solutions, 133
  - consequences, 133
  - context, 132
  - Do It Yourself, 97
  - online retrospectives, 134
  - personal anecdotes, 134–135
  - refactored solutions, 133–134
  - symptoms, 133
- Death by Postponement
  - antipattern solutions, 100
  - consequences, 101
  - context, 100
  - online retrospectives, 103
  - personal anecdotes, 104–105
  - refactored solutions, 102–103
  - symptoms, 101
- Disregard for Preparation
  - antipattern solutions, 118
  - consequences, 118–119
  - context, 116–117
  - Google Drawings, 120
  - message boards, 120
  - online retrospectives, 122
  - personal anecdotes, 123
  - refactored solutions, 119–121
  - reminders, 120
  - round robins, 121–122
  - shared documents, 118–119
  - symptoms, 119
- Do It Yourself
  - antipattern solutions, 92–93
  - consequences, 93
  - context, 92
  - Curious Manager, 97
  - facilitator rotations, 96–97
  - online retrospectives, 96
  - personal anecdotes, 96–97
  - refactored solutions, 94–95
  - symptoms, 93
- Get it Over With
  - Agile Fluency, 110
  - antipattern solutions, 108–109
  - consequences, 109
  - context, 108
  - online retrospectives, 110
  - personal anecdotes, 111–113
  - refactored solutions, 109–110
  - regretospectives, 109
  - symptoms, 109
- Peek-A-Boo
  - antipattern solutions, 139
  - Ben Franklin effect, 142–143

- cognitive dissonance, 142
  - consequences, 139–140
  - context, 138
  - personal anecdotes, 143–145
  - refactored solutions, 140–143
  - Six Thinking Hats, 145
  - symptoms, 140
  - Suffocating Slowly
    - antipattern solutions, 127
    - consequences, 127
    - context, 126–127
    - online retrospectives, 128
    - personal anecdotes, 129
    - refactored solutions, 128
    - symptoms, 127
    - taking breaks, 127
  - Political Voting antipattern, 76–77
    - 1-2-3 vote method, 79
    - antipattern solutions, 77
    - consequences, 78
    - Google Drawings, 76, 79
    - online retrospectives, 79
    - personal anecdotes, 79
    - refactored solutions, 78
    - symptoms, 78
  - pooling facilitators, 96–97
  - positivity retrospectives, 44, 66, 186
  - Postponement antipattern, Death by
    - antipattern solutions, 100
    - consequences, 101
    - context, 100
    - online retrospectives, 103
    - personal anecdotes, 104–105
    - refactored solutions, 102–103
    - symptoms, 101
  - predictability, topology of trust, 194
  - Preparation antipattern, Disregard for
    - antipattern solutions, 118
    - consequences, 118–119
    - context, 116–117
    - Google Drawings, 120
    - message boards, 120
    - online retrospectives, 122
    - personal anecdotes, 123
    - refactored solutions, 119–121
    - reminders, 120
    - round robins, 121–122
    - shared documents, 118–119
    - symptoms, 119
  - Prime Directive
    - defined, 18
    - Ignorance antipattern
      - antipattern solutions, 21
      - context, 18–20
      - general context, 20–21
      - online retrospectives, 23
    - personal anecdotes, 23–24
    - refactored solutions, 22–23
    - symptoms, 22
  - Kerth, Norm, 16, 18
  - Lack of Trust, 19–20
  - Negative Ones, 178
  - timelines, 19
  - product owners, 1
  - Project Retrospectives: A Handbook for Team Reviews* (2001), 18
  - proxies, 121
- Q**
- QuizBreaker, 195, 196
- R**
- real-time retrospectives, 102–103
  - regretrospectives, 109
  - reminders, video conferencing, 120
  - required personnel, planning retrospectives
    - antipattern solutions, 85
    - consequences, 85–86
    - context, 84–85
    - online retrospectives, 87
    - personal anecdotes, 88
    - refactored solutions, 86–87
    - symptoms, 86
  - Retrospectives Antipatterns*, 151
  - Rising, Linda
    - Fearless Change* (2005), 128
    - real-time retrospectives, 102
  - rotating facilitators, 96–97
  - round robins, 121–122, 214
  - rules (ground), small talk, 49–50
  - Rushed Retrospective* (2017), 108
- S**
- safety, Measure Safety activity, 190
  - Santanga, Paul, 195
  - School of Play website (2019), The, 150
  - scrum masters, 1
  - Serenity Prayer, 31
  - Set the Stage phase, 11
  - shared documents, Disregard for Preparation, 118–119
  - Ship retrospective, 67–68
  - Shore, James, 70
  - sidetracked discussions. *See* time management
  - Silence antipattern, Dead
    - activation phenomenon, 214
    - antipattern solutions, 213
    - consequences, 213
    - context, 212–213

- Google Drawings, 216–217
- online retrospectives, 212, 216
- personal anecdotes, 216–218
- refactored solutions, 214–215
- round robins, 214
- symptoms, 214
- Silent Ones antipattern, 8
  - 1, 2, 4, All technique, 171
  - activation phenomenon, 170
  - antipattern solutions, 169
  - consequences, 169
  - context, 168
  - online retrospectives, 171
  - personal anecdotes, 172–173
  - refactored solutions, 170–171
  - symptoms, 169
  - Think, Pair, Share technique, 170
- Sinek, Simon, 197
- Six Thinking Hats, 56–57, 145
- Skilled Facilitator* (2002), *The*, 94
- Skywalker, Luke, 22
- small talk. *See also* time management
  - antipattern solutions, 48
  - consequences, 49
  - context, 48
  - ground rules, 49–50
  - online retrospectives, 51
  - personal anecdotes, 51–52
  - refactored solutions, 49–51
  - symptoms, 49
- spider web retrospectives, 76
- Start-Stop-Continue, 6
  - antipattern solutions, 7–8
  - consequences, 6
  - context, 7
  - Decide What to Do phase, 11–13
  - General Insights phase, 9
    - 5 Whys, 9–10
      - convergence, 11–12
      - divergence, 11–12
      - groan zones, 11
    - Generate Insights phase, 12
    - refactored solutions, 9–12
    - Set the Stage phase, 11
    - symptoms, 8–9
    - Wheel of Fortune, 8–9
- Storytellers, 159–161
- structural antipatterns
  - In The Soup, 58
    - antipattern solutions, 29
    - Circles and Soup activity, 30–34
    - consequences, 29
    - context, 28
    - defined, 28
    - general context, 28
    - influence, circles of, 30–34
    - online retrospectives, 34
    - personal anecdotes, 34–35
    - problems, 28
    - refactored solutions, 30–34
    - symptoms, 30
  - Nothing to Talk About
    - Agile Fluency, 70
    - antipattern solutions, 64
    - consequences, 65
    - context, 64
    - DevOps, 72–73
    - futurespectives, 66–67
    - online retrospectives, 70
    - personal anecdotes, 71–73
    - positivity retrospectives, 66
    - refactored solutions, 65–70
    - Ship retrospective, 67–68
    - symptoms, 65
    - team radar retrospectives, 66
- Overtime. *See also* small talk
  - anonymous voting, 41
  - antipattern solutions, 39
  - consequences, 39–40
  - context, 38
  - debriefing, 40
  - Decide What to Do phase, 40–41
  - general context, 39
  - Kahneman, Daniel, 42
  - online retrospectives, 43
  - parking lots, 41
  - personal anecdotes, 43–44
  - positivity retrospectives, 44
  - refactored solutions, 40–42
  - symptoms, 40
  - TCI, 44
  - time between retrospectives, 102–103
  - timeboxes, 60
- Political Voting, 76–77
  - 1-2-3 vote method, 79
  - antipattern solutions, 77
  - consequences, 78
  - Google Drawings, 76, 79
  - online retrospectives, 79
  - personal anecdotes, 79
  - refactored solutions, 78
  - symptoms, 78
- Prime Directive Ignorance
  - antipattern solutions, 21
  - context, 18–20
  - defined, 18
  - general context, 20–21
  - online retrospectives, 23
  - personal anecdotes, 23–24
  - refactored solutions, 22–23
  - symptoms, 22

- small talk. *See also* Overtime
    - antipattern solutions, 48
    - consequences, 49
    - context, 48
    - ground rules, 49–50
    - online retrospectives, 51
    - personal anecdotes, 51–52
    - refactored solutions, 49–51
    - symptoms, 49
  - Unfruitful Democracies
    - antipattern solutions, 57
    - consensus decision making, 58
    - consequences, 57–58
    - context, 56
    - dictatorships, 58
    - dot voting (dotmocracies), 58
    - minority vote, 59
    - online retrospectives, 60
    - parallel thinking, 57
    - personal anecdotes, 61
    - refactored solutions, 58–60
    - Six Thinking Hats, 56–57
    - symptoms, 58
    - timeboxes, 60
    - UX/UI experts, 61
    - Wheel of Fortune, 8–9, 19
  - Suffocating Slowly antipattern
    - antipattern solutions, 127
    - breaks, taking, 127
    - consequences, 127
    - context, 126–127
    - online retrospectives, 128
    - personal anecdotes, 129
    - refactored solutions, 128
    - symptoms, 127
  - Surprising Power of Liberating Structures* (2014), *The*, 171
- T**
- talking
    - Nothing to Talk About
      - Agile Fluency, 70
      - antipattern solutions, 64
      - consequences, 65
      - context, 64
      - DevOps, 72–73
      - futurespectives, 66–67
      - online retrospectives, 70
      - personal anecdotes, 71–73
      - positivity retrospectives, 66
      - refactored solutions, 65–70
      - Ship retrospective, 67–68
      - symptoms, 65
      - team radar retrospectives, 66
    - small talk. *See also* Overtime
      - antipattern solutions, 48
      - consequences, 49
      - context, 48
      - ground rules, 49–50
      - online retrospectives, 51
      - personal anecdotes, 51–52
      - refactored solutions, 49–51
      - symptoms, 49
  - TCI (Theme-Centered Interaction), 44
  - team performance, trust, 195–197
  - team radar retrospectives, 66, 76
  - temperature readings, 190
  - Terhorst-North, Daniel, 109
  - Think, Pair, Share technique, 170
  - time management. *See also* small talk
    - anonymous voting, 41
    - antipattern solutions, 39
    - consequences, 39–40
    - context, 38
    - debriefing, 40
    - Decide What to Do phase, 40–41
    - general context, 39
    - Kahneman, Daniel, 42
    - online retrospectives, 43
    - parking lots, 41
    - personal anecdotes, 43–44
    - positivity retrospectives, 44
    - refactored solutions, 40–42
    - symptoms, 40
    - TCI, 44
    - time between retrospectives, 102–103
    - timeboxes, 60
  - timelines, 19, 102–103
  - topology of trust, 194
  - trust
    - building, 206
    - Lack of Trust, 19–20
      - antipattern solutions, 191
      - body language, 192
      - consequences, 191
      - context, 190–191
      - culture of trust, 197
      - defining trust, 193
      - Different Cultures, 206
      - Hawthorne effect, 192
      - Measure Safety activity, 190
      - online retrospectives, 197
      - Performance versus Trust, 197
      - personal anecdotes, 197–200
      - QuizBreaker, 195–196
      - refactored solutions, 192–197
      - symptoms, 191–192



- team performance, 195–197
- temperature readings, 190
- topology of trust, 194
- trusting stance, 194
- Two Truths and a Lie, 195–196

Tung, Portia, 150

Two Truths and a Lie, 195–196

## U

Unfruitful Democracies antipattern

- antipattern solutions, 57
- consensus decision making, 58
- consequences, 57–58
- context, 56
- dictatorships, 58
- dot voting (dotmocracies), 58
- minority vote, 59
- online retrospectives, 60
- parallel thinking, 57
- personal anecdotes, 61
- refactored solutions, 58–60
- Six Thinking Hats, 56–57
- symptoms, 58
- timeboxes, 60
- UX/UI experts, 61

UX/UI experts, Unfruitful Democracies, 61

## V

video conferencing

- avatars, 121
- Disregard for Preparation
  - antipattern solutions, 118
  - consequences, 118–119
  - context, 116–117
  - Google Drawings, 120
  - message boards, 120
  - online retrospectives, 122
  - personal anecdotes, 123
  - refactored solutions, 119–121

- reminders, 120
- round robins, 121–122
- shared documents, 118–119
- symptoms, 119

message boards, 120

Peek-A-Boo

- antipattern solutions, 139
- Ben Franklin effect, 142–143
- cognitive dissonance, 142
- consequences, 139–140
- context, 138
- personal anecdotes, 143–145
- refactored solutions, 140–143
- Six Thinking Hats, 145
- symptoms, 140

proxies, 121

reminders, 120

shared documents, 118–119

voting

- 1-2-3 vote method, 79
- anonymous voting, time management, 41
- antipattern solutions, 77
- dot voting (dotmocracies), 58
- Google Drawings, 76, 79
- minority vote, 59
- Political Voting, 76–77
  - 1-2-3 vote method, 79
  - consequences, 78
  - online retrospectives, 79
  - personal anecdotes, 79
  - refactored solutions, 78
  - symptoms, 78

## W

Wheel of Fortune, 8–9, 19

Wolpers, Stefan, 108

## X - Y - Z

Yoda, 22

## CREDITS

Cover: Illustration by Nikola Korać; courtesy of Aino Corry.

Pages xxxi, 16, 18: “Regardless of what we discover, . . . and the situation at hand.” Norman Kerth, *Project Retrospectives: A Handbook for Team Reviews* (Addison-Wesley, 2013).

Pages 3, 5, 17, 27, 37, 47, 55, 63, 75, 83, 91, 99, 107, 115, 125, 131, 137, 149, 157, 167, 175, 183, 189, 203: Illustrations by Nikola Korać; courtesy of Aino Corry.

Page 31: “God, grant me the serenity to accept the things I cannot change, courage to change the things I can, and wisdom to know the difference.” Source: Reinhold Niebuhr, *American Theologian* (1940s).

Page 126, Figure 14.1: Photo courtesy of Aino Corry.

Page 143: “Having heard that he had . . . and our friendship continued to his death.” Benjamin Franklin, *The Autobiography of Benjamin Franklin* (Houghton, Mifflin and Company, 1888).

Page 145: Photo courtesy of Aino Corry.

Page 206, Fig 23.1: Google Drawing Document, copyright Google.