The Addison Wesley Signature Series

A Mike Cohn Signature Book

Improving Agile Retrospectives

Helping Teams Become More Efficient

Marc Loeffler

Foreword by
Jutta Eckstein

Improving Agile Retrospectives

Pearson Addison-Wesley Signature Series

Visit informit.com/awss/cohn for a complete list of available publications.

The **Pearson Addison-Wesley Signature Series** provides readers with practical and authoritative information on the latest trends in modern technology for computer professionals. The series is based on one simple premise: great books come from great authors.

Books in the Mike Cohn Signature series are personally chosen by Cohn, a founder of the Agile Alliance and highly regarded Agile expert and author. Mike's signature ensures that he has worked closely with authors to define topic coverage, book scope, critical content, and overall uniqueness. The expert signatures also symbolize a promise to our readers: you are reading a future classic.

Improving Agile Retrospectives

Helping Teams Become More Efficient

Marc Loeffler

Translated from German by
Eamonn O'Leary

Addison-Wesley

Boston • Columbus • Indianapolis • 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

Editor-in-Chief: Mark Taub

Executive Editor: Chris Guzikowski

Development Editor: Chris Zahn

Managing Editor: Sandra Schroeder

Senior Project Editor: Lori Lyons

Copy Editor: Paula Lowell

Indexer: Erika Millen

Proofreader: H S Rupa

Editorial Assistant: Courtney Martin

Cover Designer: Chuti Prasertsith

Compositor: codeMantra

Visit us on the Web: informit.com/aw

Library of Congress Control Number: 2017959642

ISBN-13: 978-0-13-467834-4

ISBN-10: 0-13-467834-6

1 17

To mom and dad.

This page intentionally left blank

Contents at a Glance

This page intentionally left blank

Contents

Reader Services

Register your copy of *Improving Agile Retrospectives* on the InformIT site for convenient access to updates and corrections as they become available. To start the registration process, go to informit.com/register and log in or create an account. Enter the product ISBN **9780134678344** 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.

Please visit www.improvingagileretrospectives.com to download accompanying information to the book.

Foreword by Jutta Eckstein

I recently read the following story in a daily newspaper. In a hotel in Amman, Jordan, a businessman is waiting in front of an elevator. It is one of those big, lavish hotels, which has, in order to better meet the demands of its guests, placed six elevators next to one another. This businessman waits and waits, but the elevator doesn't come. The problem is that he is standing so close to the elevator that he fails to notice that some of the other elevators have been at his floor for a long time. Were he to take two paces backward, he would reach his goal more quickly.

This story illustrates how we humans tend to cling to an established decision or a previous experience (the elevator we have called will come and not another one). We then blindly follow the old, familiar path—"we've always done it this way" or "that's how it's always been"—instead of subjecting it to a critical assessment.

The fundamental idea of retrospectives is to pause, consider the chosen path and, in order to make better progress in the future, to correct that path by means of a (usually small) change. Actually, this approach is rooted in our DNA: the correct Latin term for the human race is not, as is commonly believed, Homo Sapiens, but Homo Sapiens Sapiens—that is, the human who thinks about thinking (or also, the human who thinks twice). It is exactly this reflection on our normal, everyday experiences that stands at the center of retrospectives.

It is often the case in projects or companies that individual team members are well aware of how things might be improved. However, it also often the case that there is insufficient time to examine the possible changes in detail. So nothing is changed, and the result is usually that the team has even less time. This situation is a vicious cycle and is aptly expressed with the old complaint: "We don't have time to sharpen the saws, we have to saw."

Retrospectives should thus also be considered part of risk management; the constant analysis of events and ensuing course corrections mean that risks can be more quickly recognized and managed. Despite the fact that retrospectives have been principally used in agile software development in order to ensure agility, the regular implementation of retrospectives can be valuable in other areas. The reason for this is partly that, as another old saying goes, you learn through mistakes. However, many companies consider making mistakes a mistake and demand instead that you "do it right the first time." But in our increasingly complex world, finding out what needs to change is not just the larger part of software development. In other areas, too, the first step is to explore which is the best path to the goal. In order to do that, you must also go down some "wrong" paths.; otherwise, you cannot know which are the right ones.

The right decisions can thus only be reached through the development of the system—and so you may well ask: why continue with this approach? Simply put, exploration is an inherent component of software development and, in today's world, many other areas.

The fostering or acceptance of a mistake-culture also demands deliberate and constant learning. Thus, through their focus on continuous development, retrospectives also contribute to the establishment of a learning organization.

Retrospectives need not only be used to find potential improvements. They also afford the opportunity to raise awareness of what already works well and what has thus far been achieved.

Team members can sometimes get to feeling that everything is going wrong, and the result is wide-spread frustration. Holding a retrospective to examine the work that is being done can help them to see that some things are actually working very well. This can increase the team's motivation.

In this book, Marc has succeeded in giving a truly comprehensive overview of retrospectives: he not only includes proven concrete

methods, but also picks up on the latest developments and assesses their usefulness. Marc tackles some far from simple topics—such as divided, systemic, or solution-oriented retrospectives—and makes them practicable.

All in all, Marc has created a work that stands alone—that is, a book that offers a solid and practical foundation for those who are new to retrospectives. Furthermore, he has made sure that experienced retrospective facilitators will also find extensive inspiration and guidance in structuring retrospectives more effectively, thus contributing to continuous learning and improvement in organizations.

Enjoy using this book to forge a new path or to correct an existing one!

Jutta Eckstein

Author of *Retrospectives for Organizational Change*, *Agile Software Development with Distributed Teams*, and *Agile Software Development in the Large: Diving Into the Deep*

This page intentionally left blank

Preface

When I started using agile frameworks and did my first retrospective, it was love at first sight. The use of retrospectives to establish a continuous improvement process made perfect sense right from the beginning. I liked the idea of having a dedicated workshop with a clear structure that happens at regular intervals: a place and time that can be used to reflect on what happened the last weeks and months together with your teammates and a place and time to think about potential improvements based on your discoveries. And I still love to do it.

Unfortunately, still, many teams ignore the potential of this practice or start using it when it is already too late. This reminds me of one of my favorite metaphors: the lumberjack. Imagine a lumberjack in the woods cutting down trees. Over the last days and weeks, it got harder and harder to cut down the trees. It already takes hours for one tree. But he still continues doing his work, as he has promised to deliver a certain amount of wood. To still be able to deliver on time, he skips breaks, works longer in the evenings and even starts to work on Saturdays and Sundays. But all of these activities do not solve his real problem: He is getting slower and slower. If he took the time to do a retrospective, he would find out that sharpening his ax would be a good idea, or even better, buying a chainsaw. The same concept often applies to our work life. Time and again, we become so busy trying to deliver that we forget to ask whether a better way to do our jobs exists.

That's what agile retrospectives are for. Instead of getting stuck in the current, potentially suboptimal way of working, these dedicated workshops help to find new ways that might improve your situation. From my point of view, agile retrospectives are the cornerstone of a successful, continuous improvement process. Additionally, they are one of the best tools to trigger a cultural change in organizations.

They can even help in traditional change initiatives. Of course, agile retrospectives can be used in private life, too. I use them at the end of the year to do a New Year's Eve retrospective with my family.

As agile retrospectives are not regular meetings, but workshops, you must take some things into account to benefit from this technique. At the same time, you always have to cope with resistance in your organization, if you apply the results of your retrospectives. If you were part of one or more change initiatives, you know what I'm talking about. But I guess you bought this book to get some answers, right? In this book, you'll find all the ingredients you need to facilitate successful agile retrospectives and establish a continuous improvement process.

A great agile retrospective is fun, energetic, diversified, has a clear goal and purpose, and takes the system you are currently in into account. I'll walk you through the steps you must take to get there. I hope you'll find this book useful and that you enjoy reading it.

Acknowledgments

I would like to thank all the people who have directly or indirectly helped to create this book—first and foremost, all the teams for which I had the pleasure to facilitate a retrospective in the past years.

I also want to thank all the experts who reviewed earlier versions of this book and helped to turn a good book into a great book. These are (in no particular order) Srinath Ramakrishnan, Susanne Albinger, Pierre Baum, Jon Eversett, Gemma Kuijpers, Mateusz Gajdzik, Dennis Wager, and Adi Bolboaca.

A big thank you goes out to Veronika Kotrba and Ralph Miarka, who wrote the chapter about solution-focused retrospectives. It adds an additional and valuable perspective on agile retrospectives.

Another big thank you goes to Eamonn O'Leary, who translated most of the German text. It saved me a lot of time that I used to add some additional information to this book.

I'd also like to thank Lisa Crispin, who connected me with Christopher Guzikowski, my editor at Pearson. Without Lisa, you wouldn't be able to hold this book in your hands.

Special thanks to my wife Andrea, who kept my back free while writing this book. Without her, this wouldn't be possible. And of course, a big thank you to my two boys Nico and Ben, who had less time with their dad during the last month.

This page intentionally left blank

About the Author

Marc Loeffler is a keynote speaker, author, and agile coach. Before encountering agile methods and principles in 2006, he was working as a traditional project manager for companies like Volkswagen AG and Siemens AG. His passion is to help teams implement agile frameworks like Scrum and XP and to transform our world of work. Marc has a passion for helping teams that are struggling with agile transitions and overcoming dysfunctional behavior. He loves to generate new insights by approaching common problems from the other side and trying to wreak havoc on the process deliberately.

This page intentionally left blank

1

Retrospectives 101

The primary purpose of this first chapter is to introduce you to retrospectives. I'll tell you how to use retrospectives in a family context, introduce you to the phase model, and give you some hints for how to fill these phases with life. After this chapter, you will have all the basics to start with your first retrospective, so let's get started.

1.1 What Is a Retrospective?

Generally speaking, a retrospective (lat. *retrospectare*, "look back") is a review, a look backward. When you lie in bed at night and let the events of the day cross your mind, that is a retrospective. When a family sits down to dinner and talks about the day—the children talking about school and the parents talking about their experiences—that is a retrospective. Looking back over the life's work of an artist, author, or director is also a retrospective. As part of a retrospective like this, various events take place at which a range of the artist's work is shown. All the important pieces are collected in a single place to provide a complete picture of the artist's work. This makes it possible to get a good overall impression and affords the opportunity of comparing and contrasting the different works of art. This would be impossible if we had access to only one example. Only by getting the overall impression is it possible to see the whole and have the opportunity to speculate about why the artist did one thing and not another.

Another kind of retrospective takes place on television, usually at the end of every year, in the form of a year-in-review program, where the different broadcasters compete to have the funniest, most beautiful, or most famous people on their programs. Entertainment is the priority here, and there's not much emphasis on getting a full picture. These year-in-review programs are therefore rather patchy and aren't really suitable for drawing conclusions or looking at the connections between different events.

When I speak of retrospectives in this book, I mean something else. The retrospectives I discuss also involve looking back, but that is just the first step. Much more important is to gain knowledge and insight from this activity. This knowledge and insight can help us learn from the past and adapt accordingly. We can learn from both successes as well as failures; good things can often be made even better. You could compare it to evolution: things that haven't worked become extinct, but everything that contributes to the preservation of the species is kept and developed further. In the end, each of these adaptations is nothing more than an experiment, because you never know for sure what the result will be. At best, these experiments lead to an improvement of the current situation. Sometimes they do just make things worse, which we then must analyze in the next retrospective.

Every retrospective is led by a facilitator, who ensures that the group achieves the goals it sets. He helps the groups to develop practical results that will be the foundation for future success. The facilitator is not a participant (although in small teams this is not always avoidable); he accompanies the process but is not actively involved in implementing solutions. A good facilitator is essential for a successful retrospective.

This kind of retrospective was first described by Norman Kerth in his book, *Project Retrospectives: A Handbook for Team Reviews* [1]:

A retrospective is a ritual gathering of a community at the end of a project to review the events and learn from the experience. No one knows the whole story of a project. Each person has a piece of the story.

The retrospective ritual is the collective telling of the story and mining the experience for wisdom.

In his book, Kerth explains how retrospectives differ from so-called "postmortems" and "lessons learned." The main difference is that retrospectives focus on positive future actions and use them as a catalyst for change. They represent not the end of the project, but milestones in the process of continuous improvement.

In 2001, several people met in a ski lodge to write a manifesto for agile software development [2]. The foundation of the manifesto consists of four pairs of values and twelve principles. The last of these principles is an excellent description of what happens in a retrospective:

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

This manifesto is one of the main reasons that the agile community in particular enthusiastically incorporated retrospectives into its work process. These people realized that they did not have to wait until the end of a project to learn from what had happened and make appropriate changes. Instead, they organize a retrospective after each iteration; that is, after a certain period. This interval should be no longer than one month. Otherwise, you run the risk of stretching the feedback cycle too far.

What Is an Iteration?

The word *iteration* comes from the Latin *"iterare,"* which means "repeat." Iterations are applied in a wide range of areas where problems are solved step by step. In computer science, *iteration* is the name for the process of taking different steps until the desired condition is reached (as with a FOR loop, for example). In Scrum, an iteration is called a "sprint."

I use the term *iteration* to describe the process of running a project in clearly defined, short, repetitive steps. After each iteration, you stop to determine whether and to what extent the project objective has been realized and, if necessary, adapt the original plan. The goal is to keep the risk of uncertainty and surprises to a minimum. The same procedure can also be used in change management.

Holding retrospectives enables you to establish a process of continuous improvement, which constantly checks whether or not you are on the right path and also gives you the opportunity to intervene and make any necessary changes promptly. By scheduling a dedicated time for reflection, you give yourself the opportunity to solve problems immediately, instead of having to wait until the end of the project. If you do not hold the retrospective until the end of a project, you run the risk of forgetting what you have learned before the next project. You also gain the opportunity to implement improvements in every iteration.

What Exactly Is the Term "*Agile*" in This Context?

The word agile comes from the Latin *agilis*, "to do, make, or act." As described earlier, this agility is based on the 12 principles of the Agile Manifesto [2].

The Agile Manifesto is as follows: We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, although value exists in the items on the right, we value the items on the left more.

The corresponding 12 principles look like this:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity—the art of maximizing the amount of work not done—is essential.
11. The best architectures, requirements and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

As you can see, some of the principles directly target software development. However, most of the principles can also be applied easily in other areas. The agile manifesto is based on the fundamental idea that we live in a complex and unpredictable world. Therefore, creating a detailed project plan for several years or even months makes no sense. As most people who have ever planned a project know, after only a very short time, the plan bears little semblance to reality. Agile developers understand this situation and try to minimize its effect using short feedback cycles and to work closely with the customer.

Different frameworks and processes have been developed on the basis of the Agile Manifesto. Among these are XP, DSDM, Open UP, and, of course, Scrum, which is currently the most popular. At the same time, ideas from agile software development have also spread to other fields. For example, in his book *"The Leader's Guide to Radical Management"*[3], Stephen Denning describes the application of the ideas in the Agile Manifesto to the field of management.

1.2 New Year's Eve Retrospective

A few years ago, my family and I started a new Year's tradition. We call it the New Year's Eve Retrospective. Not only is it a lot of fun, but it also helps pass the time until midnight (especially helpful with children). The New Year's Eve Retrospective goes like this: To start, we all sit down together and look at some photos and watch some short videos that we took during the year. I've prepared a USB stick with the photos and videos beforehand. This phase of our retrospective is always loads of fun and results in a lot of laughter.

After this review, we have a look at our measures and hypotheses from the year. This is important because it is the only way we can determine whether or not the resolutions we made last year had the desired effect. If they didn't, we can decide whether the subject is still relevant and choose a new measure. After reviewing our hypotheses,

we start to recollect all the things about the last year that have been particularly memorable. We use three categories:

- What did I like this year?
- What I did not like at all this year (or what made me angry)?
- Thank you

The first category is for all of those things that were fun or made us happy; for example, our family holiday in a kyrgyz yurt. The second category includes all the negative events. These are things like "socks everywhere" or "annoying parents." The third category simply serves to say "thank you" to your wife or mom, to the children or siblings, and so on. Connecting your gratitude to a specific case is always important. For example, "Thanks for letting me play with your Skylander toys" or "Thank you for making me a snack every morning."

'It is then time to gain knowledge and insights. Each family member is allowed to choose a topic that he or she finds particularly important, and these topics are discussed in turn. The goal of these discussions is to find the underlying causes for the topic. At the moment, we're finding the 5-Why question method very valuable here

5-Why Method

This method starts with the question: "why is x happening?" or "why does x always happen?" The answer serves as the basis for the next "why" question. You then repeat the process, digging deeper and deeper until, hopefully, you've found the real cause. We make sure to write this cause on a piece of paper because it is the foundation of the next phase. The 5-Why method is around 100 years old and was created by Sakichi Toyota [5], the founder of Toyota, to get to the bottom of production problems and so prevent them from re-occurring.

The next step is to use the causes we've found to create concrete, measurable resolutions for next year. To this end, we have a short brainstorming session to collect ideas about our topics. You wouldn't

believe the ideas children can come up with, even for the topics closer to their parents' hearts. Everyone presents his or her ideas for each topic, and we choose the most promising idea. We make our choice by sticking colored dots up next to the ideas on the paper. This technique is called "Dot Voting." Each of us has three sticky dots, which we can put wherever we like. Once we've finished, we place the newly chosen measures in a prominent place: our family corkboard in the hall, which is our highly visual to-do list. There is nothing worse than results that are not visible after the retrospective. Our board helps us to keep an eye on our new measures and ensure that we actually implement them. Importantly, we also link each measure to a testable hypothesis that we can review in the next retrospective.

Of course, a retrospective also needs a worthy ending. 'In this case, the choice is easy: the New Year's Eve fireworks.

1.3 The Retrospective Phase Model

If you were paying close attention in the preceding section, you might have noticed that we went through six phases during the New Year's Eve retrospective, as illustrated in Figure 1-1.

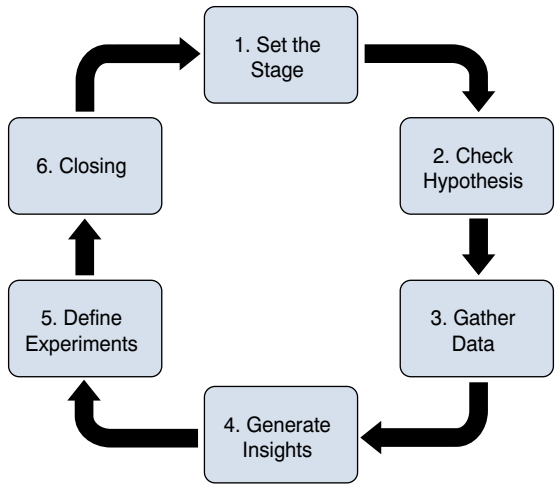

Figure 1-1 The six phases of a retrospective

These form the structure of a retrospective and are based on the original phase model in Esther Derby and Diana Larsen's book [5]. The model I describe here is an expanded form of Derby and Larsen's, the big differences being that I introduced the "Check Hypothesis" step and extended the "Define Experiments" step to include hypotheses. I explain the reasons for this later in the book. In the following sections, I explain the six phases in more detail.

1.3.1 Phase 1: Set the Stage

The first phase of a retrospective should set the stage. This phase is very important because every participant has to be mentally "picked up" from somewhere else. If you leave out this phase, you run the risk of one or more participants being mentally absent from the retrospective as they are still thinking about the last piece of work they were doing before walking in. Preparing the ground serves to get all the participants' attention and get them involved. Starting with a few words of welcome and thanking everyone for taking part is best. Then you as a facilitator briefly explain the reason for and the aim of the retrospective as well as the timeframe and the agenda. The agenda is important because, after all, we all want to know what we're spending our time on.

Practical Tip

Make sure that everyone in the room says something (brief). Someone who is silent at this stage is likely to remain so for the rest of the retrospective. However, it is very important that every voice be heard, because only then you will be able to get a complete picture. The participants don't all have to tell long stories; a few words per person is enough. For example, you might have people say their name or describe their expectations of the retrospective in a single word. Interestingly, this simple technique works so well in most cases that the quieter and silent team members will also participate in the discussions.

The last step of the first phase is also very important. The aim is to create an atmosphere in which difficult topics can be addressed. Only in an atmosphere where even unpleasant things can be discussed is it possible to get to the bottom of things and to address the real causes of problems. Moreover, that is the basis for a successful retrospective. What happens in Vegas, stays in Vegas.

You create this atmosphere by establishing the rules for cooperation, or the "working agreement." Some teams have already defined the values they have for their daily work, and in that case, you should use those values and simply remind the team of them. You might need to adjust a few values to the retrospective. The same applies, of course, if the team has already defined rules for collaboration. Many agile teams create a team charter at the beginning of their collaboration.

What Is a Team Charter?

A team charter defines all the rules for teamwork, including the rules for communication and conduct as well as the timing and length of regular meetings. Software development teams also have a list of the development tools they use and possibly links to further information. The team charter is, among other things, a good starting point for new team members. It should be a living document that is iteratively developed. If any team member feels and expresses that the charter should be adjusted, then the team discusses that request and, upon agreement, adjusts.

If there are no rules for cooperation yet, now is the time to define them. However, why are these rules so important? The following is a brief example.

Let's say your colleague James has the habit of taking his laptop with him into every meeting. He uses the time in these meetings

to answer his e-mails or surf the web. If you start the retrospective without clearly pre-established rules, he will probably do that same thing. It will annoy everyone, but no one will have the rules to point to and ask him to close the laptop. However, if the rules have been defined in advance, they can be pointed out at any time. Another advantage of having common rules for cooperation is that all the participants are responsible for observing them. This makes it easier for the facilitator to concentrate on the actual work of the retrospective.

Practical Tip

If the team does not yet have a team charter, invite members to a workshop immediately after the retrospective in order to create one.

Unfortunately, this is the phase of the retrospective that is most frequently skipped because people want to save time and get started right away. In my experience, taking a team through this phase has never been a waste of time. If the team has been working together for a long time, it often takes no more than five minutes. Five minutes

- that minimize the risk of someone not speaking
- that make sure that everyone feels they are in a safe environment in which to work
- to get everyone present and let them clear their heads for this important meeting

Sometimes it can also be five minutes of fun. For example, you might ask the team: "If the last iteration were a car, what kind of car would it be?" All it takes is one or two words, and you get everybody mentally present.

Check-In

This check-in technique is described in Derby and Larsen's book [5, p. 42] and is implemented after you have welcomed the participants and presented the goal of the retrospective. The facilitator asks a short question, which each participant answers in turn as quickly as possible. Here are a few example questions:

- In one or two words, what do you hope for from this retrospective?
- If the last iteration were a country, which country would it be?
- What kind of weather word (sunny, cloudy, rainy, thunderstorm) would you use to describe your present mood?

It is okay for a participant to say "pass." Even that one word is enough for his voice to be heard.

As a reminder, in our New Year's Eve retrospective, we set the stage by looking at the photos and videos from the past year. Believe me—it's a lot of fun!

1.3.2 Phase 2: Check Hypothesis

The purpose of the Check Hypothesis phase is to review the hypotheses created at the last retrospective. Ideally, these hypotheses are created from the experiments chosen (see section 1.3.5). However, why is this step so important?

Let's say that during the last retrospective you discussed the problem of very poor communication with the product management team. The product manager is hard to reach, and questions are only answered after major delays. At the end of the last retrospective, you decided on a measure to be taken: The product manager would now be available to the team for a specific time slot every day. This time would be for discussing current questions and getting answers, thus reducing delays to a minimum. The hypothesis that you connected to

this experiment might have been as follows: "Current questions will now be answered in less than 24 hours." This would be a real improvement on the recent situation, in which the team sometimes had to wait several weeks for a response.

After the stage has been set, the team checks the hypotheses. It turns out that the experiment was wrong. It seems that although the response time is getting a little better, it is still far from the 24-hour mark. So, the problem remains. In the further course of this retrospective, the team will, therefore, try to pinpoint the causes of this problem and then either adapt the current experiment or define a new one. During this process, it might discover, for example, that the product manager was never consulted about the new change and was simply told to implement it. Rather than motivating him to work more closely with the team, this just made him angry. Using hypotheses enables the team to work on a topic until the problem is either solved or reduced to a tolerable level.

Practical Tip

If any of your hypotheses are not confirmed as you expected, use the next phases of the retrospective to find out why.

This example shows that hypotheses are an important tool. Some teams merely check whether or not the measures chosen in the previous retrospective were actually implemented. Only a few bother to check whether those measures also had the desired effect. However, only by checking for the desired effect can you actually create improvement. This is certainly not a panacea, but it is effective in most cases. Hypotheses also help to make retrospectives meaningful and help you to stay focused on a topic instead of letting the discussion wander.

1.3.3 Phase 3: Gather Data

Now we come to the actual looking back invoked in the word *"retrospective."* The aim of the Gather Data phase is to collect data on a clearly defined period from the past. This could be the last iteration

(or sprint in Scrum), the period of an entire project, or even the last working day. The time between an event considered and the retrospective should be kept as short as possible. Your main goal in this phase is to create a common understanding of the period you are considering. Without this common picture, the participants might not understand one another's perspectives and opinions and will tend to project their feelings onto others. To create a common picture, everyone gets the opportunity to present his or her view of things.

You start by collecting the hard facts. These facts can be anything that took place during this period, from meetings and decisions to personal experiences. Include everything that had and has a meaning for anybody on the team here. Numbers (measures) might also feature in this step; for example, the number of completed requirements, or the number of closed, open, and new errors. The more memorable the result, the better.

You could simply talk about all of these things, but including a visualization is much better. A visualization simplifies the recording of information and is indispensable, especially in the case of longer retrospectives. One example of a visualization is a timeline laid out on a wall, which allows you to see the temporal relationships between events (see Figure 1-2).

Figure 1-2 Gather data using a timeline

Although the hard facts are important, they are still only part of the story. Just as important are the personal perspectives that people have on the time being considered because these tell us which events are more important and which are less so. Collecting both facts and personal perspectives helps to focus on the issues that have most affected the team. At the same time, the emotional quality of these perspectives also reveals the situations in which people felt good. Knowing when people felt good gives you the opportunity to re-create this situation more often. A further reason to discuss emotional issues is that, though they have the potential to become a drain on energy and motivation in daily working life, they are often overlooked.

Only by talking to your team can you find out what is going on and put yourself in a position to address concerns, eliminate negatives, and strengthen positives.

Definition of the Term *"Team"*

When 'I use the term *team* in this book, I'm talking about any form of team in the professional context. This could be a software development team, a team of HR people, or any other kind of team. It could even be the team of your sport club. In other words, a team is a group of people working together to achieve a common goal.

Before moving on to the next step, take the time with the team to get an overall picture of the period you are reviewing. You can do this by having each team member present his or her insights, or by giving the whole team some time to reflect on the information you have collected (using the timeline, for example).

Reminder: In the New Year's retrospective, we collected the data by *sorting* events into three categories:

- What did I like this year?
- What did I not like at all this year (or what made me angry)?
- Thank you

Each of us then briefly presents the topic we've chosen. By using emotional words in the question, we set ourselves up to get a combination of hard facts and feelings. Experience has shown me that this phase of the retrospective should be varied very often. I will talk about possible variations in the chapters throughout this book. If you can't wait, have a look at the later section 1.4.

1.3.4 Phase 4: Generate Insights

You use the Generate Insights phase to understand the context of the situation as well as possible causes and connections. You analyze the events collected in the previous phase and then ask, "Why did these things happen?" What you are looking for are insights into the fundamental causes of the events that took place.

After the first phase, this phase is the next most frequently omitted. Many teams skip this phase and immediately try to define future experiments without considering the possible causes of the current situation. This means that they only ever scratch the surface and that their measures will only treat the symptoms instead of dealing with the root causes. It's like using pain killers, when you actually broke your leg. The pain will vanish for a short period of time, but because the root cause wasn't addressed, the pain will come back. This is not a good idea because what might seem a promising path out of your problem often leads you straight back into it. On the other hand, carrying this phase out well provides you with a solid foundation for the next phase: define experiments and their hypotheses. Do not try to tackle all of your problems at once. Instead, choose the issues that the group feels are the most important. You won't be able to solve all of your problems in a single retrospective. This phase is designed to help the team step back, see the whole picture, and start looking for the root causes. It doesn't make sense to work on more than three topics during the next iteration, as these topics won't be the only thing you have to work on, right? You need the insights gained in this phase to define reasonable and effective measures.

Remember, during our New Year's Eve retrospective, every family member is allowed to choose the topic that is most important to him or her and which he or she would like to discuss at this stage. We currently use the "5 Whys" to look for causes. When our children get older, we will vary the technique we use.

1.3.5 Phase 5: Define Experiments

The first four phases have set up a strong foundation for the Define Experiments phase. You've created an overall picture and common understanding of the period under consideration and have also gained some insight into the various events that took place. At this point, most of the team will already have some ideas about what to improve or try out next. So the team's next task is to choose one or two actions and to agree on how to implement them. This also ensures that the team will have the time to implement its decisions. After all, the daily workload still must be done. Trying to implement too many changes at once can lead to problems. It also makes it more difficult to tell later which experiments actually had an effect.

I use the word *'experiments'* deliberately here. Nobody knows what will happen if you try something out. Although we may have an idea of what might happen (our hypothesis), no one can actually be sure. An analogy for this is a lab researcher who creates an experiment to test his hypothesis. Only at the end of the experiment will he be able to tell whether or not it actually worked. The most effective way to work with these experiments is to repeat your retrospectives at regular intervals that are as short as possible. This creates a safe space: An experiment that is going wrong will make less mess if you cut it off quickly rather than let it run rampant.

Just as important as the definition of the experiment itself is the definition of the corresponding *hypothesis*. You carry out experiments not (just) for fun, but because you think it will create an effect.

The hypothesis allows you to determine the extent of an experiment's effect in the next retrospective. So, hypotheses must be testable. A hypothesis such as, "This will lead to fewer errors in the software" is vague and harder to assess meaningfully. A better version of this hypothesis is, "The number of known errors in the software will be reduced to ten or less." You must always consider how your hypothesis is to be tested. This is the only way to make hypotheses meaningful and to use them to define new experiments if the first proves unsuccessful.

Practical Tip

Explicitly explain to the team that any action defined in this phase is nothing more than an experiment. No one can be certain beforehand of what the actual outcome will be.

Making the results of the retrospective visible to everyone is good practice. Agile teams, like a Scrum team, always include the defined experiments in the next planning session. The experiments chosen are considered part of the normal workload and are not extra tasks. That's exactly how it should be. It is also important that the team is willing to carry out these tasks. Having a single person take responsibility for each experiment is best. This person does not have to carry out the experiment alone but is responsible for ensuring that action is taken. If nobody is assigned responsibility now, you're likely to find that no one feels responsible for carrying out the experiment.

We used sticky dots (like those shown in Figure 1-3) to choose the experiments during the New Year's Eve retrospective. We then displayed these experiments on our corkboard to keep their status in mind. There's nothing worse than task lists that get lost in some document, wiki, or email.

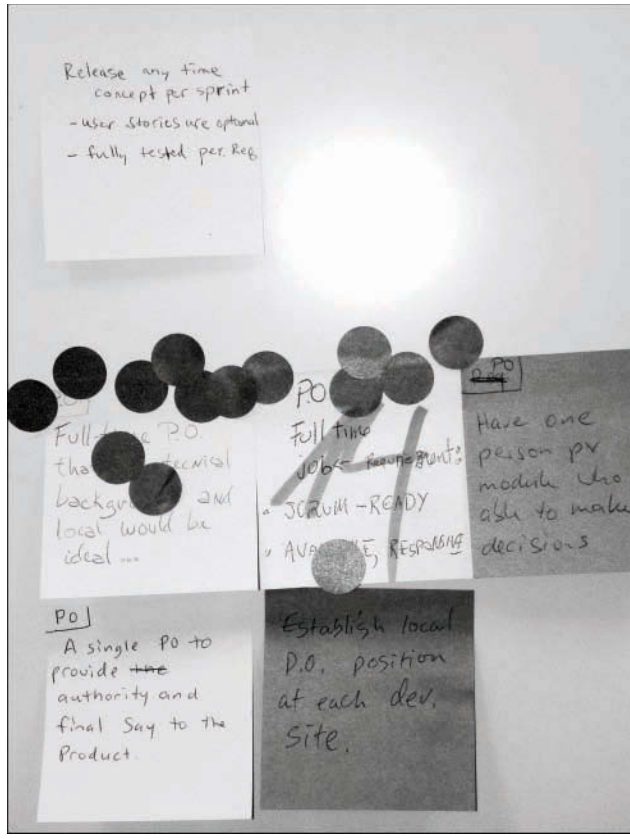

Figure 1-3 Dot voting

1.3.6 Phase 6: Closing

To conclude, spend a few minutes on a short review and celebrate the results of your retrospective. This honors the time and energy that the team has put into both the retrospective and the preceding time span or iteration. You should also document your results appropriately. There are many ways to do this, including taking photographs of the whiteboard and keeping the flipchart paper the team used to

develop their ideas. As described earlier, display these things very visibly in the team's workroom. Finally, the facilitator summarizes on how to proceed. This is to check that everyone understands the plan.

As a very last step, having a brief retrospective on the retrospective itself is always a good idea. After all, you want continuous improvement to extend to your retrospectives, too. One tool for this is a ROTI (Return on Time Invested) graph, as shown in Figure 1-4.

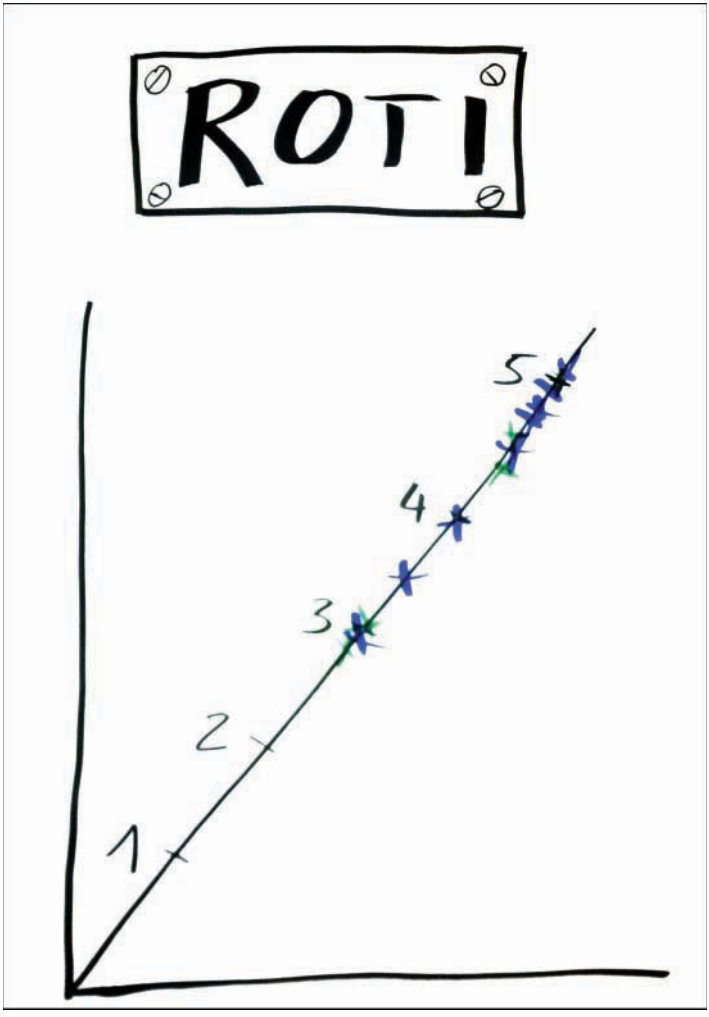

Figure 1-4 ROTI (Return on Time Invested)

What Is a Return on Time Invested (ROTI) Graph?

A ROTI graph is often used after a meeting to get some quick feedback from a team. It is a good way to determine whether your retrospectives are working well or whether they need to be improved. To create a ROTI graph, simply draw x and y axes and then a diagonal line numbered from one to five. One means, "This meeting was a total waste of time." Three means, "This meeting was just about worth the time I invested in attending." Five means, "This meeting was absolutely fantastic; the time I invested in attending paid off incredibly well." Each participant adds a cross to the graph to show his or her opinion, and the result is the completed graph. As you can see in Figure 1-4, this team was quite happy with their retrospective.

My family and I were able to celebrate our New Year's Eve retrospective with a beautiful fireworks display. Unfortunately, you can't do this every day, but a delicious slice of cake at the end of a retrospective can also provide a great ending.

Practical Tip

The time you'll need to spend on each phase of a retrospective depends, of course, on the activities you select for each phase, as well as on the total amount of time at your disposal. In general, however, the time you'll spend on each phase can be reliably calculated as a percentage of the total time. By way of example, here are the phase timings for a 60-minute retrospective:

1. Set the stage (5 minutes = 1/12 time)
2. Check hypotheses (5 minutes = 1/12 time)
3. Gather data (10 minutes = 1/6 time)
4. Generate insights (20 minutes = 1/3 of the time)

5. Define experiments (15 minutes = 1/4 time)
6. Conclusion (5 minutes = 1/12 time)

These timings are only a general rule, but they are a reliable place to start when planning your retrospectives.

The phase model provides you with a simple framework that will help you to plan and carry out retrospectives effectively. Keep to this framework, and you'll have an ideal foundation. Remember, though, that each and every retrospective is unique. These six phases have been tested many times, and they work. In the rest of the book, you will learn how to bring this model to life as well as how to deal with the typical difficulties that can arise.

1.4 Finding Activities for Each of the Phases

The six phases are just a framework which helps you to structure retrospectives. Like many frameworks, it tells you what to do, but does not specify how. Your task then, is to bring these phases to life and you do that by finding a range of activities to carry out in each of the phases. The activities you choose should be appropriate to the goal of each phase and, when you're still new to retrospectives, it to finding something suitable can be difficult.

Practical Tip

As you're starting out, avoid the temptation to find a new activity for each phase every time you do a retrospective. Just try out a few activities at first.

Many experienced retrospective facilitators have written about their ideas and made them available in books and on the Internet. In the following sections, I present some of the sources I have used.

Later in the book, you will also learn a few techniques for generating your own activities, but the following sources are an excellent place to start.

Practical Tip

When choosing activities, make sure that they dovetail. You need to be able to use the results you get from an activity in one phase in the following phase. You can't choose activities at random. Just as you'll only be able to cook a good meal if your ingredients work well together, you'll only have an effective retrospective if your activities work well together.

1.4.1 Agile Retrospectives Book

"Agile Retrospectives: Making Good Teams Great" by Esther Derby and Diana Larsen [5] was the first book to discuss retrospectives in the context of agile software development and is one of the key texts on retrospectives in general. After a brief introduction to the topic and the description of the phase model, the writers swiftly move on to the practical component. Eighty percent of the book consists of descriptions of activities that can be carried out in the different phases. The description of each activity includes the goal, the time required, the individual steps, the materials required, and possible variations.

Derby and Larsen describe a total of 38 activities, which provides enough material for quite a few retrospectives. Combining these activities in different ways means you can keep a sense of variety and novelty in your retrospectives over a long period.

1.4.2 Retromat

I came across the Retromat [6] website by chance and have recommended it as often as possible ever since. No other source enables you to find activities for your retrospectives as easily as it does. It was created by Corinna Baldauf [7].

The first time you visit the site, you immediately get a suggested retrospective plan with different activities proposed for each phase. If you don't like those activities, you can either generate a completely new plan, or click through different activities per phase until you find what you want. The activities on the site come from various sources, including Derby and Larsen's book. Each plan has a reference number that allows you to find it again or share it with other people. As of the writing of this book, Retromat offers 131 activities, and more are always being added. Retromat also allows you to enter your own activities.

1.4.3 Retrospective Wiki

Another great source for ideas on designing your retrospective is the Retrospective Wiki [8], which contains a list of possible activities and complete plans. This wiki also features some tips and tricks, descriptions of typical problems and potential solutions, and links to further sources. Many of the activities included will be familiar from the other sources I've described, but you will also find some new ideas. This wiki is constantly expanded and maintained.

1.4.4 Tasty Cupcakes

Tasty Cupcakes isn't really dedicated to retrospectives but features a wide range of games and simulations that can be used in all areas of life. For example, you might find a workshop on product innovation or a simulation to make it easier to understand a particular topic. This website was created by Michael McCullough and Don McGreal after they presented a variety of games at the Agile2008 conference. They were assisted by Michael Sahota.

Several of the ideas on the site can be used in retrospectives. Just click on the words "retrospective" or "retrospectives" in the tag cloud' to get a list of possible activities. This site is constantly being expanded and maintained, so 'having a look from time to time is worthwhile [9].

1.4.5 Gamestorming

Gamestorming [10] is a wonderful collection of creative games that support innovation and implementing change. Some people might be put off by the word *'game,'* but the creative techniques presented in the book are more like playful approaches to work than games.

This book is a practical reference with a total of 88 different activities, most of which can be used very easily in retrospectives. After all, a retrospective is nothing if not a catalyst for change. The activities are divided into four categories:

- Core Games
- Games for Opening
- Games for Exploring
- Games for Closing

The names of these categories have some overlap with the six phases of a retrospective. "Games for Opening," for example, are likely to work well in the "Set the Stage" phase. The activities listed under "Games for Exploring" are suited to both the "Gather Data" as well as "Generate Insights" phases. "Games for Closing" can be used in "Define Experiments" and to conclude the retrospective.

Here is a possible plan for a retrospective using activities from *"Gamestorming"*:

- Set the Stage: Draw the Problem (p. 90)
- Gather Data: Pain-Gain Map (p. 190)
- Generate Insights: Understanding Chain (p. 218)
- Define Experiments: Prune the Future (p. 247)
- Closing: Plus/Delta (p. 246)

The book provides key information for each activity, including the goal, a detailed description of the process, and an approximate

runtime, which helps with planning. Also included is a piece of information that is important if you want to carry out the activity effectively: the number of participants.

In addition to the activities, the book features a good introduction to the idea of game storming as well as provides you with the information you need to start creating your own activities. This book is a must for anyone serious about retrospectives and implementing change.

1.5 The Prime Directive

Some facilitators begin their retrospectives by reading out the fundamental principle, the Prime Directive. First articulated by Norman Kerth in his book, *Project Retrospectives: A Handbook for Team Reviews* [1], the Prime Directive is designed to set the stage for the retrospective:

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.

This principle is read aloud at the beginning of a retrospective, precisely in this wording.

The idea is to make it clear to everyone that we are all human and make mistakes. The principle also points out that we shouldn't assume that things have been done badly deliberately.

Practical Tip

You don't need to read out the Prime Directive at every retrospective. In later retrospectives, simply reminding people of it is enough.

Many retrospective facilitators swear by the Prime Directive. They feel that retrospectives that don't start with this fundamental principle are less effective and therefore less useful. Pat Kua writes

[Kua 2012] that this is related to the *Pygmalion* [11] or *Rosenthal* effect, or what is commonly known as "'a self-fulfilling prophecy.'"

The effect of a teacher's preconceptions about his students might be an example of the Rosenthal effect. The idea is that a teacher's positive preconception about a student ('that student is a high achiever') will affect the teacher's behavior in such a way as to create confirmation of his expectations. What happens is that the teacher subtly transmits his preconception to the student through, for example, more one-to-one attention, more time given for response, frequency and strength of praise or blame, or high-performance requirements. This is an unconscious rather than deliberate course of action.

In essence, the theory is that someone who is treated as having certain characteristics will manifest them. In fact, Rosenthal's results were repeatedly called into question and could only be reproduced in 40 percent of cases [11].

I personally believe that the success of a retrospective depends not on the careful reading out of the Prime Directive, but rather upon the values that it describes. I have carried out many successful retrospectives during which I did not explicitly mention the Prime Directive. I'm not saying that reading the principle isn't a good thing; in new teams or established teams that are about to experience their first retrospective, this ritual can have a very positive, if not measurable, effect. In my experience, however, you lose that positive effect if you read out the directive at every retrospective. Repetition does to the directive what frequent flying does to pre-flight safety briefings. The first time you fly, you pay close attention. However, with prolonged exposure, you pay less and less attention until, in the end, you hardly notice it's happening.

A positive attitude is essential for a successful retrospective, but I believe there are many ways to achieve that attitude and the Prime Directive is only one (and one that is certainly no guarantee of success).

There is also an alternative prime directive that is somewhat longer but may work better for some teams [12]. I personally like the fact that it is written in the first person and is thus more appealing:

Some days are better than others. Some days I'm in the "flow" state, doing awesome work. Some days I come to the end of a day and realized I've wasted a lot of time, made mistakes that I should have foreseen, or wish I could have done something differently.

Regardless, those days have happened and our purpose here is to find out:

What can we learn from our past actions and thinking that will inform and guide our future actions and thinking so that we can do a little better?

How can we change our environment ("the system") so that it's easier for us to do awesome work and less likely for us for us to waste time and make mistakes?

Like the original Prime Directive, this version describes the goal of a retrospective and articulates the underlying principles. Also like the original, this alternative is just a tool and does not guarantee a successful retrospective. My advice is that you experiment with both versions and see what kind of an impact it has on your retrospectives. When properly used, the Prime Directive can be a valuable tool.

Summary

In this book, I describe what retrospectives are and how to use them to establish a process of continuous improvement. Looking back into the past is only a part of a retrospective, and not even the most important. Retrospectives should be used to help you gain insights and try new things, to create and carry out experiments and to question them, too. That is the best way to support a goal-oriented and meaningful process of continuous improvement and constant learning.

Although retrospectives are still most commonly used in working life, as at the end of projects or in the form of "heartbeat" retrospectives in agile teams, they can be usefully applied to any area of life, as in our New Year's Eve retrospective.

A six-phase process that defines the framework for retrospectives will help you to make retrospectives as effective as possible:

- Set the Stage
- Check Hypotheses
- Gather Data
- Generate Insights
- Define Experiments
- Closing

Each phase can be brought to life with a range of activities, which, when regularly changed, will bring fresh energy and ideas into the process. You can either design these activities yourself or turn to one of the many books or websites available.

Starting retrospectives by reading out the Prime Directive can help to prepare the ground for a successful retrospective, but you should remember that doing so does not guarantee a successful outcome.

Ultimately, the success of a retrospective lies with the facilitator and the participating team. In the chapters that follow, I describe the keys to success and the common pitfalls to avoid.

References

[1] Norman Kerth. 2001. *Project Retrospectives: A Handbook for Team Reviews*. New York: Dorset House Publishing Co Inc.

[2] Manifesto for Agile Software Development. http://agilemanifesto.org.

[3] Steve Denning. 2010. *The Leader's Guide to Radical Management: Reinventing the Workplace for the 21st Century*. San Francisco, CA: Jossey-Bass.

[4] Sakichi Toyoda. https://en.wikipedia.org/wiki/Sakichi_Toyoda.

[5] Esther Derby and Diana Larsen. 2006. *Agile Retrospectives: Making Good Teams Great.* Farnham, Surrey UK/: O'Reilly Ltd.

[6] Retromat. https://plans-for-retrospectives.com.

[7] Website of Corinna Baldauf. http://finding-marbles.com/

[8] Retrospective Wiki. http://retrospectivewiki.org.

[9] Tasty Cupcakes. http://tastycupcakes.org/

[10] Gray et al. 2010. *Gamestorming: A Playbook for Innovators, Rulebreakers, and Changemakers.* Sebastopol, CA: O'Reilly.

[11] Pygmalion Effect. https://en.wikipedia.org/wiki/Pygmalion_effect.

[12] Ted M. Young. *The Alternative Prime Directive.* http://jitterted.com/2013/02/11/another-alternative-to-the-retrospective-prime-directive/