



# **Getting value out of Agile Retrospectives: A Toolbox of Retrospective Exercises**

**Luis Gonçalves and Ben Linders**

Foreword by Esther Derby, co-author of  
*Agile Retrospectives: Making Good Teams Great*

**InfoQ**  
ENTERPRISE SOFTWARE  
DEVELOPMENT SERIES

# Getting Value out of Agile Retrospectives

A Toolbox of Retrospective Exercises

Luis Gonçalves and Ben Linders

This book is for sale at

<http://leanpub.com/gettingvalueoutofagileretrospectives>

This version was published on 2013-12-19



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

©2013 Luis Gonçalves and Ben Linders

# Tweet This Book!

Please help Luis Gonçalves and Ben Linders by spreading the word about this book on [Twitter](#)!

The suggested tweet for this book is:

I just got the book "Getting Value out of Agile Retrospectives" written by @lgoncalves1979 and @BenLinders. Be the next one! #RetroValue

The suggested hashtag for this book is [#RetroValue](#).

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

<https://twitter.com/search?q=#RetroValue>

# Contents

Foreword . . . . .	i
Preface . . . . .	iii
Introduction . . . . .	v
What is an Agile Retrospective? . . . . .	1
Business Value of Agile Retrospectives . . . . .	3
Retrospective Exercises . . . . .	5
Asking Questions . . . . .	8
Star Fish . . . . .	11
Bibliography . . . . .	14

# Foreword

I started designing and leading retrospectives nearly two decades ago. I've been teaching others to lead retrospectives for at least a decade. I've seen how the disciplined practice of retrospectives can help a team. I've seen teams improve their practices, boost collaboration, and make better products. Retrospectives can help teams grow into empowerment. They can catalyze the process of change for a whole organization.

I've also heard tales of retrospectives that fail to bring change. Sometimes, these failed retrospectives have fallen into a rut. The team repeats the same activities in the same order over and over again. Their habitual practice doesn't spark creativity or new thinking. Other retrospectives fail because they don't allow enough time for robust exploration.

Effective retrospectives help teams short-circuit ingrained patterns of thinking. They broaden each team member's perspective, and help teams think, learn, decide, and act together.

In this book, Luis and Ben share the potential for retrospectives. Their advice comes from practical knowledge. They've learned how to prepare an organization for retrospectives and how to introduce them into an organization. They've done the work of helping teams choose and make sustainable incremental improvements. They've grappled with the pitfalls of teams falling back into habitual thinking.

Ben and Luis offer guidance to help you and your teams make

the most of your retrospectives. They have collected activities that will help you and your teams think together and spark discussion.

In this pocket book, teams and retrospective leaders have a new and solid source to keep retrospectives fresh, focused, and full of learning.

Esther Derby

Co-author of *Agile Retrospectives: Making Good Teams Great*

Duluth, MN

November 2013

# Preface

We are both active bloggers at [www.benlinders.com](http://www.benlinders.com) and [lms-goncalves.com](http://lms-goncalves.com). On our blogs we share our experiences on several agile and lean topics including retrospectives.

Blogging is one way in which we share our knowledge, and it's a rewarding one. We like the comments we receive on our blogs when people share their own experiences, and we love hearing from people who have tried the things that we write about.

Many readers have told us that they value our blog posts. We started thinking about how we could make things easier for them, by providing a small book on a specific topic - a book that they can carry with them that has hands-on information that they can use in their daily work. These thoughts led us to produce this book about agile retrospectives.

We are aiming this book at agile coaches, scrum masters, project managers, product managers and facilitators who have at least some experience with doing retrospectives. They know about the purpose of retrospective, how they fit into agile and how to arrange and perform them.

During the years we have conducted many different kinds of retrospectives. We think that it helps when you develop your own personal toolbox with retrospective exercises. We have given this book a personal touch by including our own experiences which are marked with our initials (*BL*) or (*LG*).

We want to thank the many reviewers of our book for investing their time and suggesting improvements: Robert Boyd, Paul van den Broek, Jens Broos, Gerard Chiva, Iñigo Contreras, Hans Dekkers, George Dinwiddie, Stuart Donaldson, Jos Duising, Doralin Duta, Jutta Eckstein, Murrae-Ann Erfmann, Earl Everet, Gerald Fiesser, Don Gray, Linda Halko, Shane Hastie, Joy Kelsey, Gert van de Krol, Cem Kulak, Diana Larsen, Kjell Lauren, Niels Malotaux, Claus Malten, Paul Marsh, Oluf Nissen, Lawrence Nyveen, Pierre Pauvel, Kim Payne, Sylvie R., Sebastian Radics. Whitney Rogers, Cherie Silas, Hubert Smits, Lene Søndergaard Nielsen, Ram Srinivasan, Johannes Thönes, Asheesh Vashisht, Matt Verhaegh, Patrick Verheij, Dan Verweij, Robert Weidinger, and Willy Wijnands. Your feedback has helped us to make this a better book.

Having a foreword by Esther Derby makes us feel very honored. With us, many have learned the why and how of retrospectives from the book *Agile Retrospectives: Making Good Teams Great* that Esther wrote together with Diana Larsen.

We thank [InfoQ](#) for publishing this book as mini-book. This helps us to reach a worldwide audience of passionate professionals involved in the adoption of agile.

Finally we would like to thank all the people who invest their time to read and comment on our blogs. Your feedback helps us increase our understanding of the subjects that we write about and makes it worthwhile for us to keep blogging!

Ben Linders & Luis Gonçalves



# Introduction

This book contains many exercises that you can use to facilitate retrospectives, supported with the “what” and “why” of retrospectives, the business value and benefits that they can bring you, and advice for introducing and improving retrospectives.

Agile retrospectives are a great way to continuously improve the way of working. Getting feasible actions out of a retrospective and getting them done helps teams to learn and improve. We hope that this book helps you and your teams to conduct retrospectives effectively and efficiently to reflect upon your ways of working, and continuously improve them!

This book starts with two chapters that provide answers to the questions [What is an Agile Retrospective?](#) and [Why Do We Do Retrospectives?](#) These answers help you to understand the purpose of retrospectives and to motivate people to do them.

The chapter [Business Value of Agile Retrospectives](#) explains why organizations should invest in retrospectives and what they can do to get more business value out of them.

The [Retrospective Pre-Requirements](#) chapter describes how you can prepare your organization for doing retrospectives and discusses the skills that retrospectives facilitators need to have.

The chapter [Designing a Retrospective](#) explains why you need different exercises for retrospectives, how you can design a retrospective that is valuable for a team given their situation, and what you can do to develop your own toolbox of exercises.

The main part of this book is the chapter with many practical [Retrospective Exercises](#) that you can use to lead retrospectives with your teams. Any time you are running a retrospective and you do not know what exercise to use you can pick one of the many exercises from this chapter.

The chapter [Benefits of Retrospectives](#) gives you ideas about what agile teams can expect to get out of doing them.

[Adopting Agile Retrospectives](#) describes what you can do to introduce retrospectives in your organization and how you can improve the way that you do them.

Getting Value out of Agile Retrospectives doesn't intend to teach you the theory behind retrospectives. For that purpose there are books like *Agile Retrospectives* from Esther Derby and Diana Larsen and *Project Retrospectives* from Norman Kerth (see the [Bibliography](#) for a full list of books and links).

With plenty of exercises for your personal retrospective toolbox, this book will help you to become more proficient in doing retrospectives and to [get more out of them](#).

# What is an Agile Retrospective?

The agile manifesto proposes that a “team reflects on how to become more effective”. Agile retrospectives can be used by teams to inspect and adapt their way of working.

At the end of an iteration typically two meetings are held: the sprint review (or demo) that focuses on getting product feedback and discussing how to proceed, and the retrospective that focuses on the team and the [processes](#) that are used to deliver software. The goal of retrospectives is help teams to continuously improve their way of working. This book is about performing and improving retrospectives.

An agile retrospective, or sprint retrospective as Scrum calls it, is a practice used by teams to reflect on their way of working and to become continuously better at what they do.

The 12th agile principle states that:

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

All team members attend the retrospective meeting where they “inspect” how the iteration has gone and decide what to improve

and how they want to “adapt” their way of working and behavior. Retrospectives are an effective way to move toward [short-cycled improvement](#).

The retrospective facilitator (often the scrum master) should have a [toolbox of possible retrospective exercises](#) and should be able to pick the most effective one given the situation at hand.

Typically a retrospective meeting starts by checking the status of the actions from the previous retrospective to see if they are finished, and to take action if they are not finished and still needed. The actions coming out of a retrospective are communicated and performed in the next iteration.

To ensure that actions from a retrospective are done they can for instance be added to the product backlog as user stories, brought into the planning game and put on the planning board so that they [remain visible](#) to the team.

# Business Value of Agile Retrospectives

Agile retrospectives help your teams learn and improve, and in effect [increase their business value to their customers and the company](#). They can make your organization faster, more efficient and innovative.

A few things that you can do in retrospectives to raise business value are:

- Make the team aware that we look for actions that they can do [empower your teams](#). A benefit of retrospectives is that [actions are defined and done by the team](#).
- Focus on learning and understanding instead of blame. You can use the prime directive to set a positive culture for improvement.
- Limit the number of issues and the action items that you investigate in retrospectives. It's better to have a few high-quality actions, than many actions with a risk that they won't be done. Try to [change only one thing at a time](#).
- Use the [golden rules for agile process improvement](#) to help teams work together in a smooth, efficient and positive way while improving the way they work.
- Focus on clearly defined problems and help teams to find improvement actions that matter to them and enable them

to do their work better. Use retrospectives to [give power to your teams](#) and [to empower your professionals](#).

- Use [root cause analysis](#) to find the causes (not symptoms) of problems. Then define actions to prevent them from reoccurring. When people understand the problems and their causes they are often more motivated to work on them.
- Follow up on and evaluate the progress of actions to help the team to understand why some actions worked and some didn't (double-loop learning) and [make the progress visible](#).
- Use [different exercises](#) in retrospectives depending on the issues at hand, the mindset of the team, etc. Make sure that you have a [toolbox of retrospective techniques](#). When in doubt over what to do, try something new!

If retrospectives are done frequently, where each one analyzes what happened in the iteration and defines actions to improve, then they will lead to continuous improvement with considerable business value in the long run

# Retrospective Exercises

Using different kinds of exercises helps you to get the most out of retrospectives. The following sections describe the exercises that you can use to do retrospectives.

The retrospective exercises described in this chapter are:

- An easy but powerful exercise is [Asking Questions](#). There are [many different questions that you can ask](#). The trick is to pick the ones that help the team gain insight into the main and urgent issues and identify improvement potential. Then, by asking more detailed questions, it allows the team to dive even deeper into the retrospective.
- The [Star Fish](#) is a variant on the “What went well?, What did not go so well?, What can be improved?” exercise. It uses a [circle with 5 areas](#) to reflect on what activities the team should stop right away, what activities the team should continue with in a reduced role, what activities should be kept, what activities should play a bigger role in the future and what activities the team should start.
- The [Sail Boat](#) is an exercise to remind the team of their goal the product they need to deliver, the risks they might face, what is slowing them down and most importantly, what helps them deliver great software. It uses a [metaphor of a boat, rocks, clouds and islands](#).
- When there are issues within a team that need to be discussed, you can do a [One-Word Retrospective](#). You start

by asking each team member to state [how they feel about the past iteration in one word](#). These words are used to discuss topics that might otherwise not reach the surface.

- Team member can rate their performance as a team, by stating the [Car Brand](#) that they associate with the iteration. It allows everyone to [share their opinions about the iteration](#) and come up with topics that can be improved.
- The moods of team members are often affected by problems encountered while working together. Having team members state their feelings in a retrospective using the [Happiness Index](#) helps to identify possible improvements. This exercise uses a [graphic representation of team members' emotions](#).
- If there are significant problems that a team wants to avoid in the future, you can use [Five Times Why](#). This exercise uses [root cause analysis](#) to [get to the deeper causes of problems](#) and to define actions that address them.
- The [Constellation](#) exercise can be used to visualize if team members are in [agreement or disagreement about relevant topics](#). It's an opener that can be used to help teams to feel comfortable and speak freely about any topic.
- The [Team Assessment Survey](#) allows teams to do introspection in different areas, for example: performance of the product owner, management of activities within the iteration, team spirit within the team, and implementation of technical best practices. This exercise is based on the [team assessment survey from the Scaled Agile Framework](#).
- A [Strengths-Based Retrospective](#) visualizes the strengths that your team members and teams have using a [solution-focused approach](#). It helps to explore ways to use strengths as a solution to the problems that teams are facing.



- A [High-Performance Tree](#) is a metaphor that is used to help teams draw a vision and define a destination where they want to go. At the same time it points out what needs to be in place for the teams to reach their vision. This exercise is based on the [high-performance-tree retrospective originally created by Lyssa Adkins](#).
- When teams have become more mature, [Value-Stream Mapping](#) is an exercise that can help them understand their pitfalls and find ways to improve. The exercise [visualizes the way that the team develops software](#). It reveals dependencies and shows waste within the software-development process.
- When you have an agile project with multiple teams, you can do a [Retrospective of Retrospectives](#) to [improve collaboration between teams](#). This is an effective way to share learning's across a project and to solve problems that a project is facing.

### How are the exercises described?

All retrospective exercises are described in the following format:

- *What you can expect to get out of this exercise:* The potential results that this kind of retrospective can give you and the benefits of using this retrospective exercise.
- *When you would use this exercise:* Situations where this retrospective exercise can be most useful.
- *How to do it:* A detailed description of the exercise and how to apply it.

## Asking Questions

One exercise often used in agile retrospectives is to ask questions to the team and collect and cluster the answers. The results can be used to define improvement actions that the team can do in the next iteration.

### What you can expect to get out of this exercise

Asking questions helps teams that have just started to reflect and improve their way of working to [become agile and lean](#). Realizing that they can get retrospective actions done motivates teams to learn and improve continuously.

You can help mature teams by asking more detailed and focused questions to help them to fine-tune their way of working.

### When you would use this exercise

If you have never facilitated a retrospective before then asking questions is an easy way to start. Since questions can vary, it's also flexible which makes it suitable in many situations.

### How to do it

With a team that is new to retrospectives you can use [the four key questions](#) which have been defined by [Norman Kerth](#):

- What did we do well, that if we don't discuss we might forget?
- What did we learn?
- What should we do differently next time?
- What still puzzles us?

The four retrospective questions are usually very effective. Asking “What should we do differently next time” urges team members to look for things that they want to change. It often helps to facilitate a discussion, to find out why a process needs to be changed and to build a shared understanding and commitment for the actions that the team will do.

“What did we do well” is a [solution-focused approach that can be used in a strength-based retrospective](#). The addition of “if we don’t discuss we might forget” makes this question even stronger; if something good happened by accident, that’s great, but what can you do to ensure that you will keep doing it?

The question “What still puzzles us” can provide useful insights by revealing things that had remained unspoken previously. If things come up, a [One-Word Retrospective](#) can be used to [deal with the team’s emotions](#). Asking “What did we learn” makes people aware that in order to become better, they will need to learn. If this question doesn’t lead to answers in several consecutive retrospectives, it can be a signal that the team is not trying enough new things. That’s something you can dig into using [root cause analysis](#).

Asking questions is an exercise that is easy to learn, but the effectiveness depends on the questions that you ask to the team. (BL) Working with agile and non-agile teams, I have been doing project evaluations, audits, assessments using the [CMMI](#) and the [People-CMM](#), retrospectives and many other kinds of feedback sessions. The questions below are a mix based on these frameworks, but worded in such a way that you can ask them in agile retrospective to help teams to find things that they can improve.

Examples of questions:

- What helps you to be successful as a team?
- How did you do it?
- Where and when did it go wrong in this iteration?
- What do you expect, from whom?
- Which tools or techniques proved to be useful? Which did not?
- What is your biggest impediment?
- If you could change one thing, what would it be?
- What causes the problems that you had in this iteration?
- Are there things that you can do to these causes?
- What do you need from people outside the team to solve the problems?

The trick is to pick the questions that help the team to gain insight into the primary issues that they are having, and questions that help them to visualize their improvement potential.

Use open questions to elicit answers that provide more information, and use follow up questions to help teams get insight into what happened. Ask for examples to make situations concrete, summarize answers to build a shared understanding in the team and come to actions that the team will do.

## Star Fish

Star fish exercise is an evolution of the typical three questions that are used for retrospectives: What went well? What did not go so well? What should be improved?

### What can you expect to get out of this exercise

This exercise helps to identify problems of and opportunities for the team. Instead of the typical three questions, we have a circle with five words:

- *Stop* – These are activities that do not bring value to a team or customer; activities that bring waste into the process.
- *Less* – These are activities that require a high level of effort and produce little benefit. They may also be activities that were brought into the team from the past but did not lead to any overall improvements to a process.
- *Keep* – Usually these are good activities or practices that team members want to keep. These activities are already being applied.
- *More* – Activities on which a team should focus and perform more often. For example, many teams tell me how pair programming is helpful yet they do not need to do it every time.
- *Start* – Activities or ideas that a team wants to bring into the game.

With this exercise, teams can get an overall picture of what's going on within the team, what is working and what is not. They can get an overview about failed as well as successful ones in the

past. (LG) In my personal opinion, I think this is a great evolution of the typical three retrospective questions.

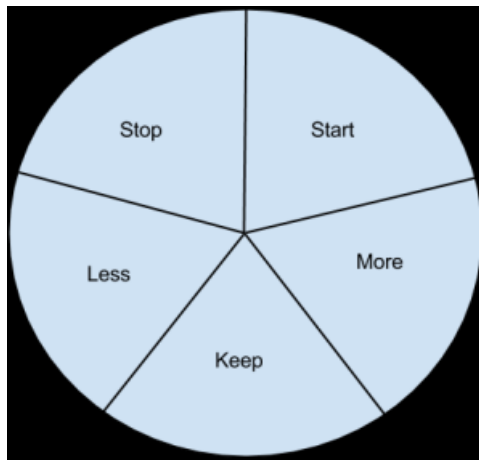
### **When you would use this exercise**

I believe this simple technique does not require any special occasion. It might be interesting for situations when a team goes through several ups and downs during the iteration. This technique reveals good actions as well as less positive observations the team has performed and therefore might be a good tool for summarizing the iteration.

Star fish is suitable for any team. It does not require any specific level of maturity.

### **How to do it**

This retrospective is simple to do. First, draw this picture:



Star Fish

One of the beauties of this exercise is the fact that a team need not be collocated. You can use tools like [Lino](#), for example, to apply the exercise on non-collocated teams. [Lino](#) allows users to do everything needed in order to run this exercise.

After drawing the picture on a flip chart, it's good to start a brainstorming session by allowing the team to dump their ideas in the Stop area. After that, give two to three minutes to each person to read aloud the Stop ideas and spend 10 minutes on a short discussion to see if everyone is aligned.

Repeat the exercise for each of Less, Keep, and More.

For the Start part, add one extra step. Use Toyota approach, choosing a single topic to discuss. You can hold a vote to see what the team considers the most important topic to start with. After selecting the topic, design a small strategy to make sure a topic is well implemented. This strategy might include responsible persons, due dates, and, most importantly, success criteria. In order to know if the implementation was successful, we must have success criterion.

A theme that is chosen in the Start part does not need to be new to a team. It can be an improvement of something that is not working well.

The order of topics within the circle is important. *(LG)* I like to order them as Stop, Less, Keep, More, and finish with Start. I think this has a big impact. Starting with negative topics and progressing little by little towards the positive ones will help the team to end the retrospective with a much more positive feeling than if they did it in a random order.

# Bibliography

## Our Blogs

[Luis Gonçalves - A blog where you come to search for ideas...](#)

[Ben Linders - Sharing my Experience](#)

## Books

Lyssa Adkins. *Coaching Agile Teams: A Companion for Scrum-Masters, Agile Coaches, and Project Managers in Transition*. Addison-Wesley, 2010.

Rachel Davies and Liz Sedley. *Agile Coaching*. The Pragmatic Programmers, LLC, 2009.

Esther Derby and Diana Larsen. *Agile Retrospectives: Making Good Teams Great*. The Pragmatic Programmers, LLC, 2006.

Jutta Eckstein. *Agile Software Development with Distributed Teams*. Dorset House, 2010.

Norman Kerth. *Project Retrospectives: A Handbook for Team Reviews*. Dorset House, 2001.

Henrik Kniberg. *Scrum and XP from the Trenches*. InfoQ, 2007.

Patrick Kua. *The Retrospective Handbook: A guide for agile teams*. Leanpub, 2013.

Dean Leffingwell. *Scaling Software Agility: Best Practices for Large Enterprises*. Addison-Wesley, 2007.



Mary Poppendieck and Tom Poppendieck. *Lean Software Development: An Agile Toolkit*. Addison-Wesley, 2003.

Mike Rother. *Toyota Kata*. McGraw-Hill, 2009.

Jean Tabaka. *Collaboration Explained: Facilitation Skills for Software Project Leaders*. Addison-Wesley, 2006.

### **Links**

[Manifesto for Agile Software Development](#)

[retrospectives.eu](#)

[retrospectives.com](#)

[retrospectivewiki.org](#)