



# The Retrospective Handbook

A guide for agile teams

Patrick Kua  
Foreword by Diana Larsen

# **The Retrospective Handbook**

A guide for agile teams

Patrick Kua

©2012 - 2013 Patrick Kua

# 1 Retrospective Fundamentals

## 1.1 A Short History of Retrospectives

Norm Kerth first published his ideas on the retrospective practice in his 2001 book, *Project Retrospectives: A Handbook for Team Reviews* [KERTH]. The book describes retrospectives as:

A ritual gathering of a community at the end of the 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.

Kerth's book describes how retrospectives differ from 'Project Post Mortems' and 'Lessons Learnt' sessions with a particular focus on taking positive action and acting as a catalyst for change. At around the same time, a number of authors published the *Agile Manifesto*<sup>1</sup> as a way of rallying people to adopt lightweight methodologies. These new

---

<sup>1</sup><http://agilemanifesto.org/>

agile methodologies and approaches made learning and communication primary concerns. One of the principles of the Agile Manifesto aligns very well with the practice of retrospectives:

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

The agile community started to embrace the key idea of the retrospective practice. Agile teams saw that they didn't need to wait until the end of a project to run a retrospective and began running them at the end of each iteration. Instead of running retrospectives every three, six or twelve months, teams ran retrospectives as frequently as every week, or at least every month. By stepping back and reflecting on current situations, teams realised value immediately by applying recommendations whilst the project was in flight, rather than at the end.

The following table highlights practices and principles that align well with the spirit of retrospectives.

<b>Methodology</b>	<b>Practice or Principle</b>
Extreme Programming	Fix XP When It Breaks
Scrum	Sprint Retrospective practice
Crystal Clear	Reflective Improvement
Adaptive SW Development	Learn
Lean SW Development	Amplify Learning
Kanban	Improve Collaboratively
DSDM Atern	Timebox Close-Out
RUP	Adapt the Process

Reflecting over a shorter period of time, one to four weeks instead of three to twelve months changed the nature of these meetings. Kerth wrote about taking several days to fully reflect over a project, but looking back over a significantly shorter period, at the end of an iteration rather than an entire project, reduces the time needed to conduct an agile retrospective. Instead of project retrospectives lasting several days, agile retrospectives only take an hour or two to run. In response to these shorter meetings, sometimes called ‘Heartbeat Retrospectives’, Esther Derby and Diana Larsen published the *Agile Retrospectives: Making Good Teams Great* [DERBY]. This book provided a significant contribution to the retrospective community, providing a more relevant framework for agile teams and an even wider set of exercises for people to try out.

## 1.2 The Key Questions

At the heart of the agile retrospective is the idea of looking for lessons learnt and methods of improvement. It is with this agile mindset of constantly learning, inspecting and adapting, that we ask: What did we do well? What did we do less well? What still puzzles us?

We ask ‘**What went well?**’ so that we can explore and celebrate the good practices the team are doing. Constant celebration of success imparts energy to the team, driving them to experiment further. Recognising success also provides important positive feedback on any changes the team trialled, further cementing those changes into place.

Focusing on success helps team members to explore other ways to amplify already successful changes.

We ask ‘**What went less well?**’ to jointly agree the pain points the team may have experienced. Only by exploring the background to problems that have arisen, rather than the symptoms each individual may have seen, can teams develop a shared understanding of the issues at hand. Only with a shared understanding can teams move forward productively to choose solutions that the team trials.

We ask ‘**What still puzzles us?**’ to help team members address issues that don’t fall under the previous two categories. Sometimes agile iterations or sprints move so fast that people don’t have chance to ask about something that just puzzles them. There may be a revelation that no one else knows the answer either. If we skip this question, we lose the opportunity to learn surprising insights people may have.

## 1.3 A Simple Framework

While the key questions appear straightforward enough, you may consider using the simple framework outlined in *Agile Retrospectives* [DERBY] for more impact. It is effective for both planning and running retrospectives. The retrospective framework consists of five phases:

### 1. Set the Stage

Introduce the purpose of the retrospective and help establish the focus for this retrospective (eg. the last week, or the

last two weeks). If there have been previous retrospectives related to the same project, this is a good point to review actions taken as a result of those retrospectives, connecting them with this retrospective. This phase is also useful for testing the comfort level, or safety and engagement level of the group. It also presents the ideal opportunity to introduce the [Prime Directive](#)<sup>2</sup>.

## 2. Gather Data

This stage is for collecting facts or key memories that stand out in the minds of the team members. It is important for the facilitator to help collect sufficient information from all participants to generate a common understanding of what happened during the focus period.

## 3. Generate Insights

This phase is where the team focuses on interpreting the data gathered in the previous phase, perhaps exploring why certain events happened or the impact that they had on the team and the project. Facilitators work to ensure that blame is not apportioned to any one individual. The team strives to discover the possible causes and effects before identifying possible solutions or improvements.

---

<sup>2</sup><http://www.retrospectives.com/pages/retroPrimeDirective.html>

## 4. Decide What to Do

The team have now brainstormed a large possible list of things to try. Attempting to tackle all of them often results in only a handful of them being done. During this phase, participants narrow down the list of viable alternatives and agree as a group which actions they will try.

## 5. Close the Retrospective

The possibility of change and a plan of action feels exciting. This phase helps teams recognise a successful outcome, leaving them more enthused to make change happen.

You may be tempted to skip the five phases. Avoid skipping any phase if you can. Teams that skip a phase tend to suffer with poor-quality discussions and less effective changes, or sometimes no change whatsoever! I think this framework works because it caters to people's tendency to have conversations at a different pace. Using the framework ensures that everyone progresses at the same time. Any changes are more likely to succeed because everyone is part of the process.

Without the framework, there is a danger of people jumping to conclusions and recommending an action before everyone has a shared understanding of the issues. This action will almost always be suboptimal or address the wrong problem. Chris Argyris, one of the founders of organisational psychology, introduced the concept of the **Ladder of Inference** to explain how people process information. You can find a more detailed explanation of



this in Peter Senge's *The Fifth Discipline: The Art & Practice of the Learning Organization* [SENGE].

## **The Ladder of Inference**

More on this in the full version.

### **1.4 Where to Find Activities**

More on this in the full version.

### **1.5 The Importance of the Prime Directive**

More on this in the full version.

### **1.6 The Right Context for Retrospectives**

More on this in the full version.

### **1.7 Complementary Improvement Practices**

More on this in the full version.