

NORWEGIAN UNIVERSITY OF SCIENCE
AND TECHNOLOGY

TDT4501 - DATATEKNOLOGI,
FORDYPNINGSPROSJEKT

A Review of the Agile Retrospective

Authors:

Alf MAGNUS STÅLESEN

Bjørn DØLVIK

Supervisor:

Torgeir DINGSØYR

Autumn June 2, 2015

Abstract

The agile retrospective is an activity designed to learn from past experiences. In this article we performed a literature review in order to investigate the subject, and this resulted an overview over themes and exercises and their relation to the retrospective. We also discuss some of the common exercises performed during a retrospective, and their strength and weaknesses. We describe the work needed to be done before, during and after a retrospective session. In addition we inspect the impact the retrospective practice has on its participants, and discuss how to improve their enthusiasm for the practice. Lastly we conclude with our own thoughts on the state of retrospective practices today. We conclude that the practices today are substantial, however the follow up processes after the retrospective are not fully explored.

Contents

I	Introduction	5
1	Background	5
2	Theory	6
2.1	Agile development	6
2.2	Retrospectives	6
2.3	Retrospective activity	8
3	Terminology	8
4	Thesis	9
II	Method	11
5	Data Collection and searching strategy	11
5.1	Semantic Searching	11
5.2	Excluding irrelevant articles	12
	Step 1	13
	Step 2	13
	Step 3	14
	Step 4	14
5.3	Article selection	14
5.4	Article analysis	15
III	Results	19
6	Overview of Studies	19

7	Practices of the Agile Retrospective	20
8	Before the Retrospective	20
8.1	Wish to improve	21
8.2	Declaring a goal	22
8.3	Gather Data	22
8.3.1	Survey	23
8.3.2	Interview	23
8.3.3	Collect objective data	23
8.4	Preparing the Retrospective	25
8.5	Participants	27
9	During the Retrospective	27
9.1	Techniques Identified	28
9.1.1	KJ-sessions	28
9.1.2	Keep, Problem, Try	29
9.1.3	Estimation Retrospective	30
9.1.4	Experience Reports	31
9.1.5	Positive Strokes	31
9.1.6	Timelines	32
9.1.7	Evidence based Timelines	33
9.1.8	Fishbone diagram - Ishikawa diagram	34
9.1.9	Causal Map Analysis	35
9.2	Other Findings	36
9.2.1	Discussion on Decision Making	37
9.2.2	Globally dispersed teams holding retrospectives	38
9.2.3	Working Agreement	39
9.2.4	Put Improvement Actions in Backlog	39

10 After the Retrospective	40
10.1 Improvements Follow-up	40
10.2 Documentation	41
11 Enthusiasm for the Agile Retrospective	42
 IV Discussion	 44
12 Retrospective Practices	44
12.1 Planning the retrospective	44
12.1.1 Wish to Improve	44
12.1.2 Declaring a Goal	44
12.1.3 Participants	45
12.1.4 Preparing the Retrospective	46
Identify Issues	47
Improve Estimation	48
Feedback	48
Problem Understanding	49
12.1.5 Collect Data	50
12.2 During the Retrospective	53
12.3 After the Retrospective	54
12.3.1 Follow up	54
12.3.2 Documentation	55
13 Enthusiasm	55
Positive to Practices	55
Recurring Issues Kills the Joy	56
14 Conclusion	56

Part I

Introduction

1 Background

Agile development is a software development process that has become widely used during the last decade. It empowers the developers [14], makes the development process more adaptable to its environment and increases the communication between the customer and the developers. Agile development has also become a widely investigated area for the software process improvement research field.

To hear what the software development industry was missing from agile development we conducted a workshop. We set the theme for the workshop as “Knowledge management within agile development” and invited several agile development meetups located in Oslo. The workshop was conducted by us asking questions about different issues related to agile development. The participants would then participate in a discussion reflecting around these questions.

During the workshop a common complaint was the down-prioritization of the agile retrospectives in today’s industry. The agile retrospective had become a routine, and teams were just going through the motions without a clear benefit.

This statement interested us. The retrospective, a process in agile development that is responsible for software process improvement. That it is down-prioritized and is losing its benefits could be a hinder for development teams to improve. Based on this we decided to perform a literature review of the currently used retrospective practices from the existing research in retrospectives.

2 Theory

In this section we describe some of the fundamental concepts and theories that relate to an agile retrospective. This is by no means a comprehensive discussion on these concepts, but are meant to give a context to the rest of the article.

2.1 Agile development

Agile development is a movement within the software development industry [14]. Agile development has grown mainstream since it was first introduced, and include variants such as SCRUM and Extreme programming. Central to agile development is the idea of flexibility and close contact with customers and stakeholders. The agile team is empowered with self control and can in a high degree decide on it's schedule and tools independently. An agile team should through contact with customers strive to deliver the best solutions as fast as possible without going through an external process. During agile development the team should have a high level of social interaction and participation, while focusing on giving the individual developer what is needed in order to succeed. An example of an agile development is the daily meeting [12], the daily meeting is also called the "Stand up meeting". This meeting is often performed standing because it is intended to be a short meeting, usually lasting no more than 15 minutes. During this meeting the team identifies what has been accomplished, what is the future work plan, and identify any obstacles for progress.

2.2 Retrospectives

Retrospectives, sometimes also called postmortems, are a practical method intended to facilitate learning after a completed project or project phase. [6]

A retrospective should result in exploring the events and issues of the project. This information should be processed and evaluated by all the participants of the retrospective. The facilitation of the retrospective is often done by the project manager or an external facilitator. The retrospective is often centered on an activity intended to stimulate discussion. A retrospective can vary in length and scope, often depending on how large a part of the project it intends to cover. Dingsøyr [6] defines the retrospective, or postmortem, in the following manner.

By a postmortem, we mean a collective learning activity which can be organised for projects either when they end a phase or are terminated. The main motivation is to reflect on what happened in the project in order to improve future practise—for the individuals that have participated in the project and for the organisation as a whole. The physical outcome of a meeting is a postmortem report.

Another description of the retrospective is found in an article by Tiedeman [15] where he defines the retrospective thusly.

We may now define what a post-mortem is and what it is not. Contrary to popular conceptions of post-mortems, such as the 1986 space shuttle disaster, a post-mortem is not a “witch hunt” to find the guilty and assign the blame. Rather, it is a learning tool and should be administered much as coaches attempt to bring out the best in their players and mold highly effective, motivated teams. A post-mortem attempts to: provide information rather than just data, identify trends and their underlying causes, compare actual versus planned, confirm or deny gut feelings and intuitions, discuss significant events and decisions, and

identify where improvements can be made. Last, but not least, post-mortems should highlight what was done right. Identifying what was done right provides not only balance in the feedback and recognition of a job well done, but also an underlying understanding of success. To summarize, a post-mortem is a systematic, formalized review of a product’s quality and the associated quality of the processes that produce it. Its intent is to learn what worked and what can be improved upon as soon after the concerned process events took place so as to apply such knowledge to work which is underway or is soon to occur.

We define the agile retrospective in this article as the process of evaluating a software development process. Which one can perform in order to improve processes and practices within a team or a company.

2.3 Retrospective activity

A retrospective activity is intended engage the participants. A retrospective should be highly interactive and look to get input from all the participants. Some activities need very little preparation while others need work done beforehand, for example by gathering data to support the discussion in the activity “Evidence based Timelines” [2]. A retrospective activity is usually aimed at uncovering the root cause of a problem, or identifying patterns in the project. Once the activity is over the team should be left with a clearer picture over potential improvements.

3 Terminology

In this section we explain some of the more common terms used in this article. Table 1 provides a quick overview of these.

Agile retrospectives goes under several names in the literature. The more common ones are postmortem and project release review. We are instead of using several terms only going to use the word retrospective as we believe this will be more relatable to todays practitioners.

For the time-spans that occur in agile development there are several names. Iteration, sprint and release all can relate to a given time span in different agile methodologies. We are going to stick with iteration as we find this most generic.

There exists several methodologies within agile development. SCRUM, Extreme Programming and Adaptive Software Development is all examples. We are not focusing on any specific methodology and such will refer to the term agile development or simply agile.

Table 1: Terminology for this article

Term	Other terms	Description
Retrospective	Postmortem, Project Release Review	The process of evaluating a software development process.
Iteration	Sprint, Release	A time-span within a development project.
Agile Development	SCRUM, Extreme Programming, Adaptive Software Development	Software development methodology.

4 Thesis

Through this article we are going to review the agile retrospective practices employed today. Using a “grounded-theory” inspired process we will

go through existing literature finding strengths and weaknesses of the retrospective. Our goal is to provide a clear overview over the domain and identify, if any, gaps in the current research. We were also interested in how the agile retrospective is viewed by the participants of the meeting, and what factors influenced their enthusiasm. We aim to investigate what benefits the participants, and how to accomplish it. Additionally this article should be a reference guide for practitioners, as well as giving an overview of themes and exercises and their relation to retrospectives.

Part II

Method

5 Data Collection and searching strategy

Basing this article on systematic review our data collection was done through several stages: Searching for articles through well acknowledged sources, excluding irrelevant hits and assessing the remainder. Each stage will be described in detail in the following subsections.

5.1 Semantic Searching

Before searching, we decided on a set of keywords which were to be used as the searching word. The selection of words was aimed towards the agile retrospective, but as the name of this practice has changed over the years we also included Post Mortem. Trying to get a bigger set of data we also included general keywords from the field such as agile practices as articles can be poorly named or described, but still be relevant for this study. With these thoughts in mind we ended up with a set of relevant keywords as can be seen in Table 2.

Table 2: Keywords used in searching databases

Keywords:
Agile Practices
Retrospective
Post Mortem
Software Development

When the list of keywords was complete the selection of databases were decided upon. The criteria for the databases were based upon trying to get a wide span of hits. This resulted in 2 databases used, Web of science and Scopus, These databases were selected as they had plentiful of search options and between them covered a wide set of journals, see Table 3.

Table 3: The databases used for data gathering and some well know

Databases:	Journals:
Web of science	Information and Software Technology Software Quality Journal International Journal of Information Technology & Decision making ++1400 more
Scopus	Information and Software Technology IEEE Software Journal of Systems and Software ++5000 others

Having selected the set of keywords and tools to use the searching began. Using the boolean operators of AND and OR one search was conducted using each tool. The search string given was the following:

"Retrospective AND software development OR post mortem AND software development OR agile practices AND software development"

This search resulted in 1,754 papers that could be relevant for our research.

5.2 Excluding irrelevant articles

Excluding articles not relevant for our research became the next step in obtaining data for this literature review. The criteria for exclusion can be found

in Table 4.

Table 4: Exclusion criteria for articles

Exclusion criteria
1. Exclude if the article have research domain other than computer science.
2. Exclude if the article isn't an empirical study. Lessons learned articles and books are in many cases not peer reviewed and we dismiss this as we wish this article to be based on high quality research.
3. Exclude if the article clearly isn't within the research area of this article. This paper has focus on the software development processes and all articles focusing on technical aspects of software engineering or is in other research area than software development are not relevant.
4. Exclude article if the focus isn't practices used in software development.

Step 1 Starting with 1,754 papers the first step was to apply automatic filtering on research area given by the databases. The area we selected was computer science as all other areas are not relevant for this paper. The result of this filtering left us with 801 articles too continue with.

Step 2 The next step in the filtering process was to exclude all papers not being peer-reviewed articles. This was also done by the databases automatic filtering. The article types we excluded were books and reviews. This resulted in 266 articles left.

Step 3 Having obtained 266 articles of potential relevance for this review, excluding articles not fitting our needs became the next step. First we excluded based on title. If the title clearly described the article as not focusing on the development practices it was excluded, i.e. if an article clearly focused on a technical aspect of some computer system it was excluded. This resulted in 119 articles left in our data collection.

Step 4 Left with 119 articles standing we started reading through the abstracts to exclude articles not focusing on the practices used in agile development. The focus of this paper is to bring a better overview of the agile retrospective and papers that focuses on the implementation of agile development or agile development as a part of a project become irrelevant. In the cases where the authors were uncertain about a papers focus, we discussed what potential parts could be relevant and if this didn't give us a result we quickly skimmed the article and looked for potential relevance. If no relevance was found we dismissed the article. This resulted in 48 articles that could be deemed relevant.

Finally the filtering resulted in 48 peer-reviewed articles that were relevant to our research area of agile retrospectives.

5.3 Article selection

These 48 peer-reviewed articles were of varying degrees of relevance for our study. We wanted to limit our final articles to the most relevant pieces. In order to do this final selection we constructed a grading scheme based on the focus of the articles, where a high grade meant a high focus on retrospectives or postmortems.

Once the grading scheme was agreed upon both group members read through all the articles separately, grading the articles. The grades were written into a document and the average was considered the final score. The

Table 5: Grading scheme for article selection

Grade	Description
1-3	Mentions retrospectives , but does not discuss in depth
4-7	Mentions and discusses retrospectives or discusses relevant practices or activities.
8-10	Focus on retrospectives or the practices relevant to a retrospective.

abstract of the article was read, in order to understand the whole content and intent of the article. The grading was very similar with few outliers. Any outliers were re-read and discussed together and if necessary. Once the final score was set any articles with a score of less than 7 were eliminated. The grade of 7 was chosen we deemed everything below not sufficiently relevant. Eliminating the low graded articles resulted in a set of articles who either focused on retrospectives or contained a high level of relevant discussion. This resulted in 19 articles that were considered relevant enough. This was considered a satisfactory amount of articles. Thus the article search ended. Out of the 19 articles chosen, only 15 gave the relevant findings for the writing for this study.

5.4 Article analysis

Once the article selection was complete we went on to perform our analysis. As earlier mentioned we approached the domain through a grounded approach inspired process. This mean that instead of using a conceptual framework and base our analysis on that we rather go through all the materials and develop our own theory. This way we find the theory grounded

in the field data rather than confirming an earlier stated theory through the field data. The reason for selecting this approach is that we want to perform a review on the whole domain of retrospective, evaluating the current practices and identify what is missing. If we were to use a deductive approach getting the full scope can become harder, and as such we ended on a grounded-inspired approach.

We conducted our analysis through several steps as can be in Table 6.

Table 6: The steps performed in analyzing the articles

Step	Description
1.	Divided the articles between the authors.
2.	Read through the articles highlighting important findings and writing them down on post-it notes
3.	Grouping the post-it notes to different views
4.	Performing the analysis through discussing each view, drawing assumptions and conclusions where feasible.

The first step in our analysis was to divide the articles between the authors. We numerated each article and decided the one author would take the even numbered articles and the other author would take odd numbered ones.

Reading through the articles was our second step. When reading each article we highlighted all of the important findings so we easily could find them again later when writing this report. We also wrote them down on post-it notes for future use. On each post-it note the finding was shortly described and the number of the article it was from was written in one of the corners. An example note is provided in Figure 1.

Our third step was grouping the post-it notes. We grouped the post-it notes so that findings that had the same theme were grouped together. If a finding could be grouped in several places we duplicated the post-it note.

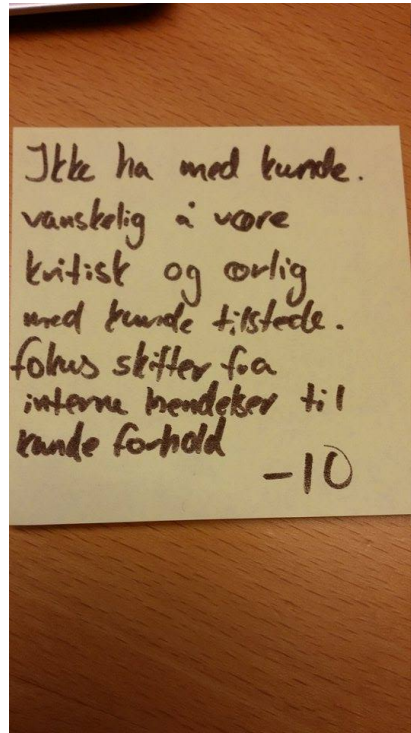


Figure 1: An example post-it describing the finding from an article with the article number written in the lower right corner

The final chart of all the post-it notes groupings can be seen in Figure 2.

The final step in analyzing the articles was performing the actual analysis through a discussion. For each group we had we discussed the theme and their findings. Assumptions or conclusions were drawn on each theme where it was feasible. The final set of the grouped findings, their themes, the assumptions and conclusions were then set together in a disposal making the skeleton for this article.

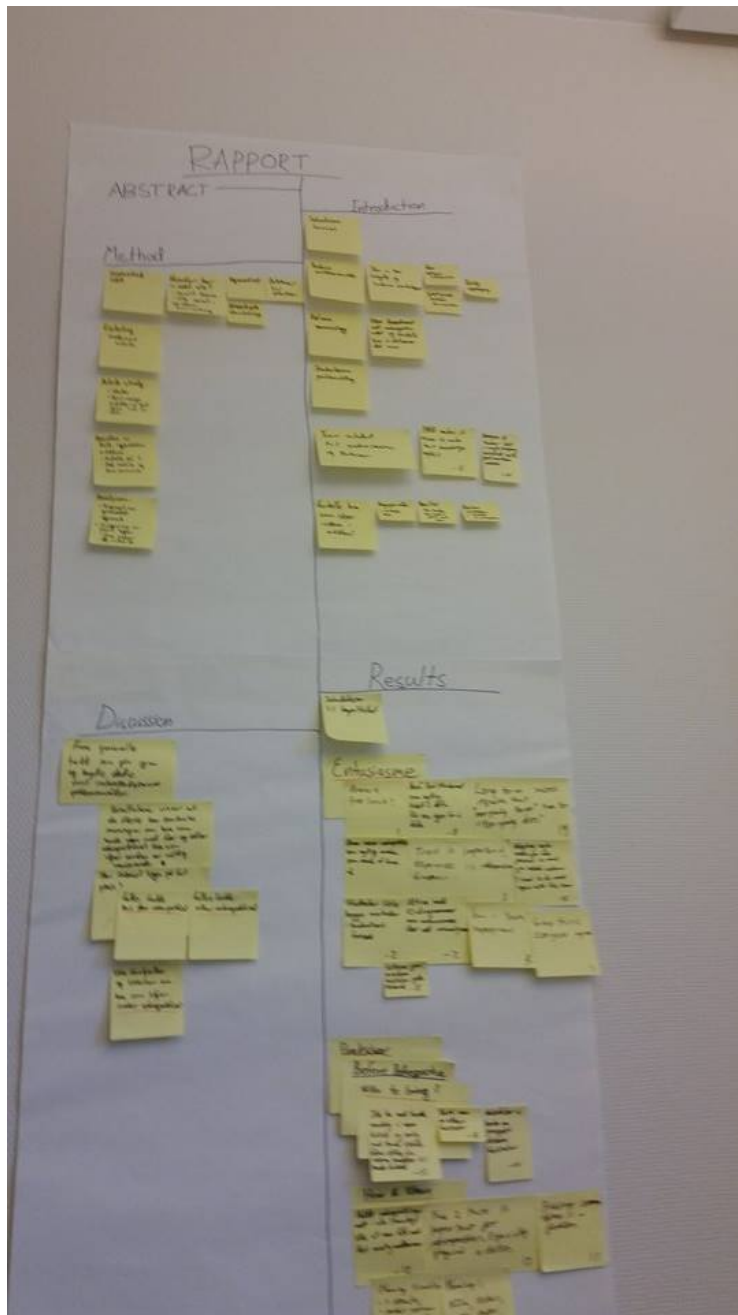


Figure 2: A picture showing the final grouping of the findings from the analysis

Part III

Results

The results from our literature review presented us with two primary views on how to evaluate the agile retrospective practices. The first one being the practices used when conducting an agile retrospective. The second one regards enthusiasm for the agile retrospective. Through this part we will present our findings on these two views, but first we will give an overview of the studies used in this literature review.

6 Overview of Studies

In Table 7 we provide an overview of what kind of research methods is used in the different studies. 53% of the research material we have used is purely empirical research. This research consists of single-case studies, multiple-case studies, experiments and literature reviews. This is the material we will draw our conclusions upon.

The remaining 47% of the research material consists of experience reports and methodology suggestions. We recognize that experience reports are not empirical studies, but we believe they will help provide a broader understanding of the research area, as Wang et al. [16] states. We will however not draw any definite conclusions based on the experience reports will use them for supporting our own assumptions. The same goes for methodology suggestions.

Looking at Table 8 we see the publications channels of the different studies. The biggest distribution channels being Journal articles, with 58%, and conference papers, with 37%. The remaining 5% is consisting of one book that we treat as a methodology suggestion. IEEE and Information and Soft-

Table 7: Studies by research method

Research method	Number	Percent
Single-case	2	10.5
Multiple-case	6	31.6
Experiment	1	5.2
Experience reports	6	31.6
Literature review	1	5.2
Other*	3	15.7
Total	19	100

*“Other” refers to methodology suggestions

ware Technology is the two journals containing most studies and the different agile 20xx conferences having the most studies for conferences.

7 Practices of the Agile Retrospective

One of our main findings reading through the selected article was that there are many ways to conduct a retrospective. Categorizing our findings we ended up with three phases a retrospective contains. These three are: before, during and after the retrospective meeting. We will go through each of these phases presenting our findings.

8 Before the Retrospective

In this section we go through the work needed to be performed before the retrospective. This is arranged in both practical tasks as well as principles that should be kept in mind in order to hold a successful retrospective. The

Table 8: Studies by research method

Puplication channel	Type	Number	Percent
IEEE	Journal	4	21
Information and Software Tech- nology	Journal	3	16
Agile 20xx	Conference	3	16
The Journal of Systems and Soft- ware	Journal	2	11
XP 20xx	Conference	1	5
IEEE 20xx	Conference	1	5
Profes 20xx	Conference	1	5
Empirical Software Engineering	Journal	1	5
ESERNET	Journal	1	5
Euromicro	Conference	1	5
Agile Retrospectives Making Good Teams Great!	Book	1	5
Total		19	100

tasks discussed are “Wish to improve”, “Declaring a goal”, “Participants”, “Collect objective data” and “Which technique to use”.

8.1 Wish to improve

The retrospective is conducted out on an interest to improve the processes in an organization, and deciding to use the retrospective as a tool. Dingsøyr [6] describes the retrospective as a method for organizational learning. So the holding of a retrospective need to be based on a wish to improve through organizational learning. Especially in software engineering has this attention

to knowledge management increased, in the hope of improving processes through learning from the past. The wish to improve should not be limited to just the management, but a culture of knowledge management should also take place in the organization units themselves.

8.2 Declaring a goal

A retrospective is not a self contained activity, and there are multiple actions that can be taken before and after the actual retrospective in order to ensure a successful meeting. Dingsøy [6] lays out a set of steps for conducting a postmortem review. The first step is the project head deciding to hold the retrospective, followed by the declaring this intent to the project. Thereafter the participants for the exercise should be selected. Then work to prepare for the retrospective can then begin. It is important that the project participants start understanding and preparing for the intent of the retrospective. The participants need to understand that it is their input that forms the basis of the output.

8.3 Gather Data

The work done before the retrospective should be focused on gathering objective data, and the relevant context concerning the data. Dingsøy, Birk and Stålhane [11] describe how one should include both positive and negative data. A retrospective should not be wholly focused on the positive and negative aspects of a project, this should be reflected in the gathering of data. It is also important to be open to collection data and feedback from all possible sources, as this gives a more complete picture of the issues.

8.3.1 Survey

There are different ways of gathering information. Collier, DeMarco and Fearey [4] describes the need for a defined process, suggesting starting with a survey to gather objective information. The survey is chosen because it is possible to be a quick low investment option where the project team's input can be gathered in a wide range of projects. Especially large projects can benefit from electronic surveys, as it can allow the gathering of large volumes of timely feedback. Surveys also provide anonymity, giving participants more room to be honest. An example survey is provided by Collier et al. in Figure 3.

8.3.2 Interview

Another option is to interview a sample of the participants or stakeholders in the project. This is usually a more time intensive method and more resource intensive. Interviews are a very powerful information gathering tool, allowing the interviewee to share data with a personal context and connection. A benefit of interviewing is the possibility of gathering points of view and data from relevant sources who will not attend the actual retrospective. This might include customers, managers or team members with other commitments. Another potential data collection method is monitoring and observing the project as it develops taking time stamped pieces of data for later use, this might for example include copies of email or other communication.

8.3.3 Collect objective data

Collier, DeMarco and Feary [4] describes the need for objective data to support the decisions of the retrospective. The data collected should when seen in a context provide an outline of actual issues of the project. The data

SAMPLE SURVEY QUESTIONS

Our sample survey that is posted on the web contains a dozen or so questions in each of the eight categories, such as

Category One: Support and Goals

Sample question: Were interdivisional lines of responsibility clearly defined throughout the project?

☐ Always ☐ Sometimes ☐ Rarely ☐ Never

Category Two: Expectations and Communications

Sample question: Did project-related meetings make effective use of your time?

☐ Always ☐ Sometimes ☐ Rarely ☐ Never

Category Three: Issues Resolution:

Sample question: Were you empowered to participate in discussions regarding issues that affected your work?

☐ Always ☐ Sometimes ☐ Rarely ☐ Never

Category Four: Information Access

Sample question: Did schedule changes and related decisions involved the right people?

☐ Always ☐ Sometimes ☐ Rarely ☐ Never

Category Five: Product Specifications

Sample question: Was project definition done by the appropriate individuals?

☐ Always ☐ Sometimes ☐ Rarely ☐ Never

Category Six: Engineering Practices

Sample question: Was the build process for the component area you worked on effective?

☐ Always ☐ Sometimes ☐ Rarely ☐ Never

Category Seven: The Big Picture

Sample question: Considering the time-to-market constraint, were the right tradeoffs made between features, quality, resources, and schedule for this product?

☐ Always ☐ Sometimes ☐ Rarely ☐ Never

Category Eight: Demographics

Sample question: What was your primary function on this project?

☐ Quality Assurance ☐ Development ☐ Marketing

☐ Project Management ☐ Documentation

Figure 3: An example survey provided by Collier, DeMarco and Fearey [4]

should support the retrospective in identifying the biggest areas that can be improved. In order to maximize the efficiency of the retrospective the issues that can be improved should be ordered in a list by the highest return on time invested. The project can then use this as a work plan for its future improvements. The data collection also allows for comparison between previous projects, by having a measuring stick the project can compare itself to external projects. By comparing the current projects to other projects the retrospective can base its decisions on a wide range of data. The hard data that is collected will also help the discussion, and focus it on the issues, as a retrospective led entirely through memory can be biased as described in Bjarnason and Regnell [2] Memory bias is the selective nature of human memory, and can forget, ignore or misunderstand events. This memory bias can lead to potential improvements going lost. The objective data should aim to cover the metric types important to supporting retrospective decision making. These metrics are cost, schedule and quality.

Through logging observations a team can get a clear picture of the actual events of the project. By making the observations pieces of date-stamped data a project can prepare for an “Evidence based timeline” described by Bjarnason and Regnell [2]. These observations have to be clear pieces of data that can be interpreted at a meeting, such as reports, graphs, meeting logs and communication logs. If data collected can not be used as an input the future steps of the retrospective the data should be process it into a usable form.

8.4 Preparing the Retrospective

A retrospective should be planned in advance, and should be tailored to the needs and goals of the project. Derby and Larsen [5] lists a set of activities that can be used to generate insight and produce valuable evaluation of

Table 9: objective data metrics, with examples

Metric type:	Description
Cost	The actual costs for issues and events during the project phase. This can include but is not limited to development, testing, quality control, customer meetings and internal meetings.
Schedule	The original schedule compared to the actual performed schedule. Included should be an overview over any missed deadlines.
Quality	Defects, bugs and problems. As well as their detection rate and fix rate.

project. The activity chosen should be suited for the type of retrospective, depending on the goal and nature of the exercise. We elaborate on some of the more common activities in subsection 12.1.4. Other factors that should be accounted for is setting the date, providing the physical materials needed and often planning food. The date is especially important as one should accommodate as many possible participants as possible. The physical materials might depend on the activity selected, but typically include markers, post its and paper. The need for food might be different from retrospective to retrospective, but typically the length of a retrospective require some sort of lunch break. In some cases, especially after a release, the food eaten at a retrospective might be part of the reward for finishing the project phase.

8.5 Participants

There are multiple concerns on who should be the participants of the retrospective. The participants are the most important part of the retrospective, as their input and reflections results in the output that makes the retrospective valuable. Therefore it is important that the participants feel safe in sharing their thoughts and experiences concerning the project. A retrospective should include representatives from all involved units of the development, included but not limited to planning, development and testing.

The units involved would naturally vary from project to project. Dingsøyr eliminates the managers [6] from participating in the retrospective, as the project members might be intimidated from sharing all their thoughts since the managers are responsible for evaluating employees. In addition stakeholders such as customers being present can shift the topics from the issues that the team can learn from over to customer relations.

The presence of the customer might make the team reluctant in bringing up bad experiences with the customer's representative, such as bad communication or long response time.

The facilitator can either be the project manager, or an external facilitator. An external facilitator has the advantage of seeing the project with a fresh set of eyes, and can potentially lead the team to viewing their problems from a different angle. The facilitator should be trained and prepared to hold the retrospective, it is also important that the facilitator inspires trust in the participants, in order to let them share as many viewpoints as possible.

9 During the Retrospective

The second phase of a complete agile retrospective is conducting the retrospective meeting. Here our literature review found that several different

techniques were used in practices. Following we will describe techniques used and suggested by the research literature.

9.1 Techniques Identified

9.1.1 KJ-sessions

KJ-sessions are often mentioned in the articles. Dingsøy, Moe and Nytrø [11] describes how they conducted a KJ-session:

We used a technique named after a Japanese ethnologist, Jiro Kawakita - called the KJ-method. For each of these sessions, we give the participants a set of post-it notes, and ask them to write one issue on each. We usually hand out between three and five notes per person, depending on the number of participants. After some minutes, we ask one of them to attach one note to a whiteboard and say why this issue was important. Then the next person would present a note and so on until all the notes are on the whiteboard. The notes are then grouped, and each group is given a new name.

This description is very much similar with description given by other researcher with only minor differences. Such as the number of post-it notes varying or instead of giving a fixed number of post-it notes one give the participants a fixed time to finish writing down the issues. An example of an output from a KJ-session is provided in Figure 4. Following a KJ-session there are cause analysis sessions, as the KJ-method only identifies the issues rather than analyze them. We describe different cause analysis techniques below.

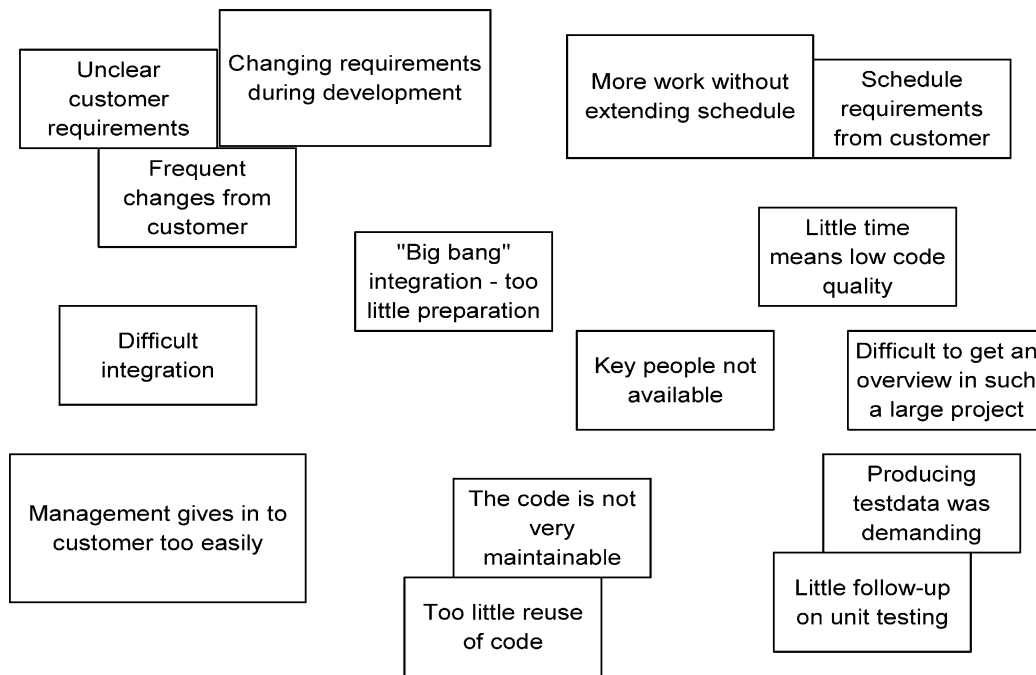


Figure 4: An graphic of the finished board after a KJ-session, provided by Dingsøyr [6]

9.1.2 Keep, Problem, Try

Keep, Problem, Try, also known as KPT, is described by Kinoshita [9] and works as follows: One splits a white board into three areas. One for Keep, one for Problem and one for Try. The session follows with the participants writing down positive feelings on the Keep area of the board, negative feelings on the Problem area. Finally the last area is filled with Try items that one tries to do in the next iteration of the project. This gives the opportunity that the team obtains feedback from oneself, according to Kinoshita. We have provided an example of a KPT board as described by Kinoshita in Figure 5.

<p>Keep</p> <p>Postive feeling 1</p> <p>Postive feeling 2</p>	<p>Try</p> <p>Try action 1</p> <p>Try action 2</p> <p>Try action 3</p>
<p>Problem</p> <p>Negative feeling 1</p> <p>Negative feeling 2</p>	

Figure 5: An example KPT board as described by Kinoshita [9]

9.1.3 Estimation Retrospective

Estimation Retrospective is another technique mentioned by Kinoshita [9]. This technique evaluates the time estimation for the current development iteration. The process is to go through each task and control whether the time estimate is close to actual time used. If there are any differences one discuss possible causes for this. The results are then used when estimating for the next retrospective. Kinoshita described how they used a whiteboard to visualize the estimations; All tasks using the estimated time would be placed in the middle, while those using shorter would be placed to the left side and those exceeding the estimation on the right.

9.1.4 Experience Reports

Experience reports are a way of collecting experience from a completed project. The process is described by Dingsøy, Moe and Nytrø [11] as follows:

This document is usually written by the project manager after the project is finished. The report follows a fixed template, which makes it possible to compare reports from different projects. In one of the companies we worked with, the report is divided into two parts: The first part gives an overview of all the facts and numbers from the project: Start and finish date, size of contract, labor used, deviation from estimated work size, the number of source lines-of-code developed, documents produced, and the activities that contributed most to the excess consumption. The second part of the report describes problems during project execution with proposal for improvement. For each phase of the project there is a Problem Description and Proposal for Improvements. This information is represented as text. In the other company they only have part two with problem definitions and proposed improvements.

9.1.5 Positive Strokes

Positive strokes are a technique used during special events such as releases or exchange of members, according to Kinoshita [9]. Its focus is to give each individual member a positive feedback from the rest of development team. The way it works is that every member gets a card which they write their name on. The card is then passed around and each member write a positive comment about the owner of the card. The card is then returned to the owner. An example card is provided by Kinoshita in Figure 6.

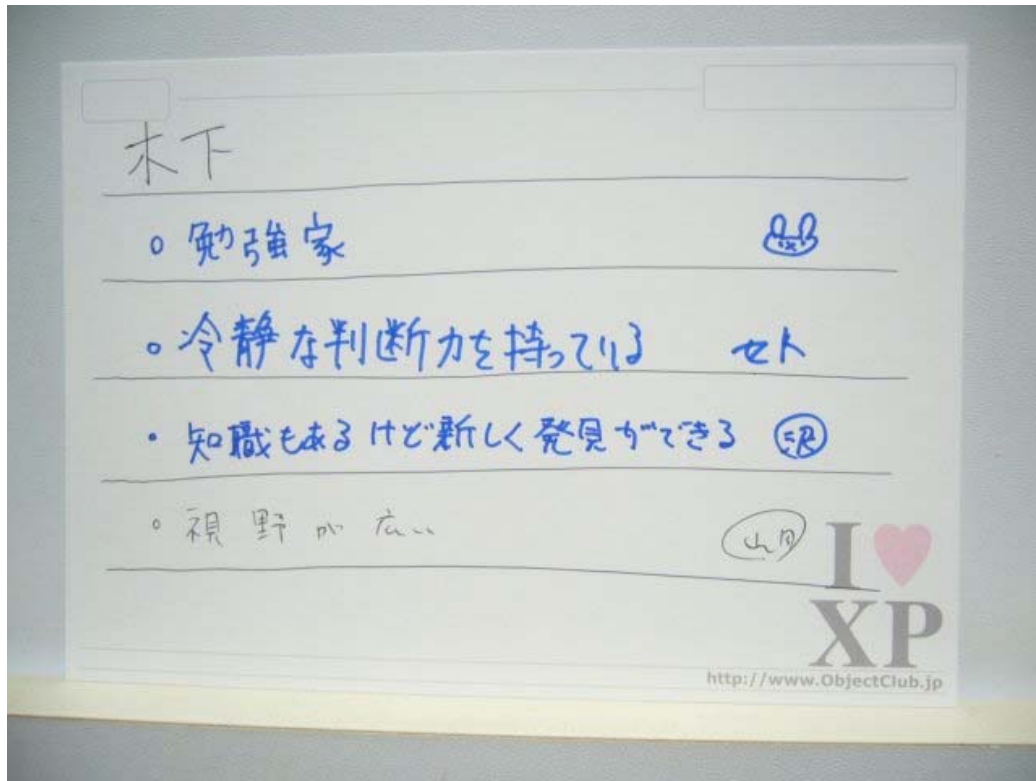


Figure 6: Positive stroke card provided by Kinoshita [9]

9.1.6 Timelines

Derby et al. [5] describes a process for a timeline activity, where the time spent is from thirty to ninety minutes. During the actual activity the retrospective leader should encourage all participants to engage in the discussion, with the goal of seeing the project from many perspectives. The participants of the retrospective should be divided into multiple groups, where every member gets markers, and post it notes. The participants should then write their own issues during the project. These should be written down on color coded post its, where the color relates to a feeling or emotional state. Alternatively the colors can represent events, functions or themes. When the groups are

done filling out their post its, place them on the timeline. Let all the participants look at the created timeline, before taking a break or lunch. After the break the group should analyze the timeline, looking for interesting patterns that can be used for future improvements.

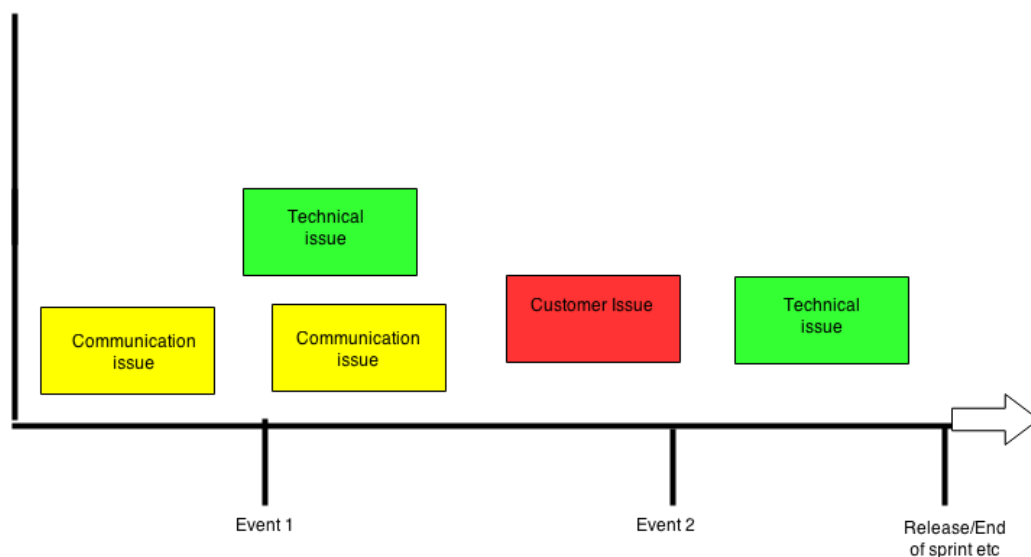


Figure 7: A very simple timeline showing a set of different issues over time, where the issues are color coded

9.1.7 Evidence based Timelines

In an agile retrospective the Timeline exercise is an exercise aimed at supporting the evaluation of the issues and changes within a project. Bjarnson and Regnell [2] describes how timelines should be based on evidence, and that this evidence should be time-stamped data that is collected through available systems in the environment of the project. The need for evidence is because the memory of a participant can be biased or incorrect. In preparation for the timeline activity the timeline should be sent out to the participants of the retrospective, along with a set of reports that will

be discussed during the actual retrospective. As the retrospective begins the history of the project is visualized, typically by posting the timeline created on a wall or blackboard, this timeline usually goes from the start of the project, to the end. The timeline could also be tailored to just contain a sprint, iteration or release within the project. As issues and events are placed on the timeline they are discussed by the participants of the retrospective. The participants themselves should correct incorrect events, as well as add their own if needed. As the timeline is corrected and clarification added, the meeting produces a timeline of events and issues that the relevant participants agree upon. The patterns and tendencies that emerge from the project timeline are then considered and evaluated in order to facilitate potential improvements. As multiple timelines are produced using the same template, over time this makes comparison analysis simpler.

9.1.8 Fishbone diagram - Ishikawa diagram

One of the cause analysis methods used is the Ishikawa diagram, commonly known as fishbone diagram. Dingsøyr [6] describes the process of fishbone diagrams as follows:

The process leader draws an arrow on a whiteboard indicating the issue being discussed, and attach other arrows to this one like in a fishbone with issues the participants think are causing the first issue. Sometimes, also underlying reasons for some of the main causes are attached as well.

In Figure 9 we give an example on how the issue late for work could be represented through a fishbone diagram.

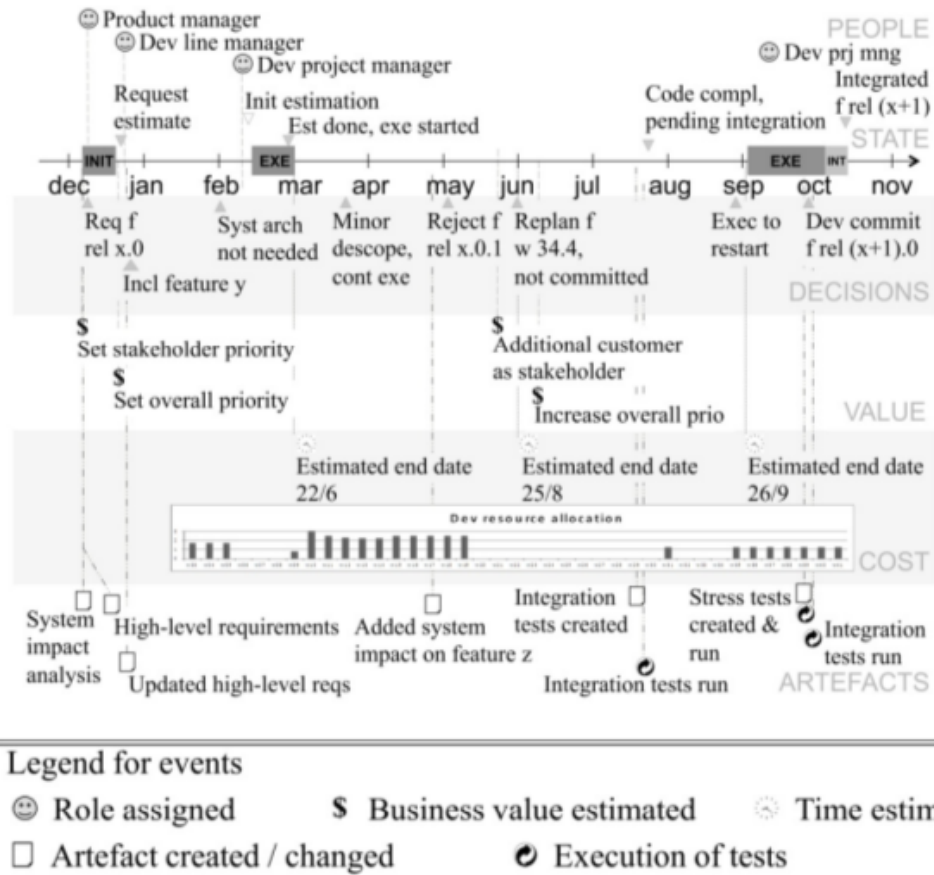


Figure 8: An example of an evidence based timeline as presented by Bjarnson and Regnells [2]

9.1.9 Causal Map Analysis

Bjørnson, Wang and Arisholm [16] proposes an alternative to fishbone diagrams in the form of causal map analysis. This is conducted in a similar way to the KJ-sessions, described above. The participants are given post-it notes on which they separately write down the causes of the issue the group is analyzing. They each then present their causes for the rest. When all causes are up on the whiteboard the participants together group the causes where able

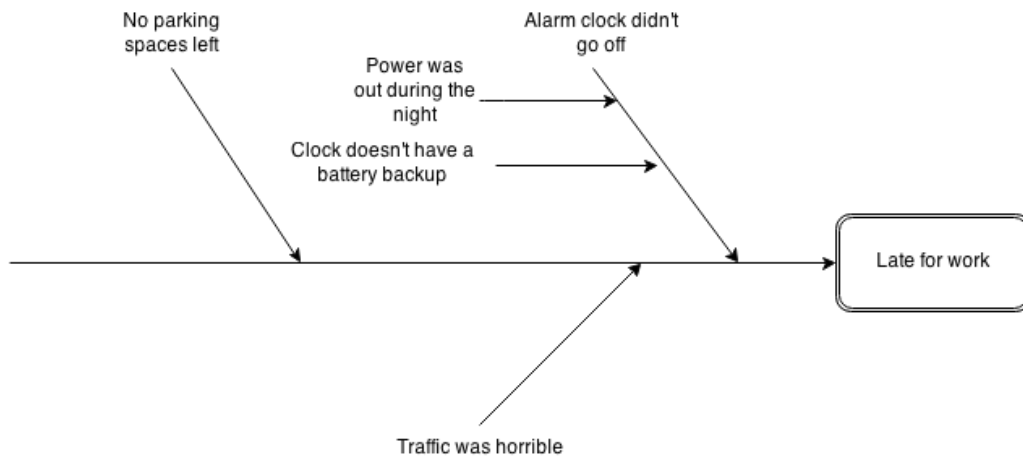


Figure 9: Fishbone diagram showing the cause analysis of the issue of arriving late for work.

before they draw cause-effects relationships between the causes indicated by arrows. The group then are allowed to write new notes finding deeper or missing causes and place them in the map. This process continues until the participants are satisfied with the analysis. Figure 10 gives an example of a causal map identifying the causes of the issue of arriving late for work.

9.2 Other Findings

So far we have showed what practices and techniques could be employed during an agile retrospective and compared these against one another. However we also came across some other findings related to what to do during the retrospective, but not any specific practice or techniques. These findings will be presented in the following sections.

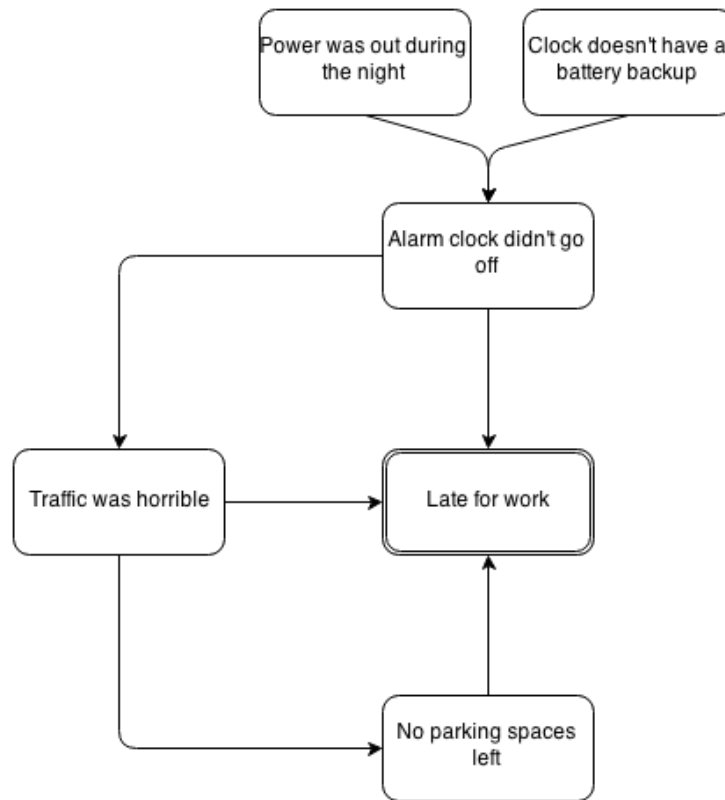


Figure 10: Causal map showing the cause analysis of the issue of arriving late for work.

9.2.1 Discussion on Decision Making

M. Drury, K. Conboy and K. Power [7] did a focus group study on obstacles to decision making in agile development and found that one obstacle was uncertainty on who was responsible for acting upon decisions made. They make the conclusion that as a part of the retrospective one should do a discussion on decision making. The reasoning behind this is that the retrospective is the only opportunity to evaluate the short-term tactical decision in relation to the bigger long-term strategic ones. They also predicts that doing discussion on decision making during a retrospective will help follow up upon the deci-

sions made. By clarifying owners for acting on decisions made one removes the uncertainty on who is supposed to act on it, and thus the decisions will not be forgotten.

9.2.2 Globally dispersed teams holding retrospectives

The rise of globally dispersed teams has led to challenges in holding retrospectives. The first major challenge is actually arranging a face to face meeting, due to the large geographical distances. This leads to potentially high travel cost both in time spent traveling as well as transportation cost. An option is to hold a virtual retrospective instead of face to face meetings. Holding virtual retrospectives is discussed by Terzakis [13] in his article. Terzakis describes the process of holding a virtual retrospectives, and the challenges needed to be overcome. A major challenge in very dispersed teams is dealing with time zones. In the example in the article the teams are 10 hours apart, resulting in careful planning in order to make the impact on the team members as low as possible. Another challenge is to overcome differences in language and culture, this can be helped by agreeing on a glossary of terms as well as recognizing cultural differences. The exercise in the example is the timeline exercise, and it is completed with the help of digital tools. The tools used is a combination of email, spreadsheets and an audio bridge. Due to the usually highly interactive nature of retrospectives the tools chosen should be chosen carefully in order to engage all participants. The facilitator gains a lot of extra responsibility in planning the retrospective and understanding the potential variations between teams. Extra facilitator training might be necessary in some cases. The main challenges in holding a virtual retrospective is seen in the following table.

Table 10: Issues in virtual retrospectives

Issue	Example of issues
Logistics	Time zones differences, choosing tools
Cultural differences	Importance of family, social norms difficulty speaking English (or another common language)
Safety	Openness between team members
Fairness	Some teams might dominate the discussion

9.2.3 Working Agreement

M. Maham [10] explains how to facilitate and conduct agile retrospectives. He suggest that starting with creating a working agreement so one can maintain a clear focus and get as much out of the time invested in the retrospective as possible.

9.2.4 Put Improvement Actions in Backlog

One of the main benefits of the agile retrospective is to be able to improve on ones processes by selecting improvements actions and work on those through the next phase of the development. M. Maham [10] suggests to not only select the improvements actions, but put them in the backlog for the next development iteration. This would make the actions easy to track and remember through the next iteration.

10 After the Retrospective

The final phase of the retrospective is after the retrospective meeting is conducted. Our study revealed two topics of things to consider after the retrospective meeting is finished. These topics are: Improvements follow-up and documentation. Our findings will be described below.

10.1 Improvements Follow-up

As one of the main benefits of the retrospective allows one to improve on ones development practices. The retrospective meeting helps identify improvement opportunities and such, one can create actions to implement these improvements. Creating the actions however does not by themselves implement the improvements one has to perform and follow-up upon them as well. This however can be a challenge. We only found two articles [4, 7], mentioning this topic. They identified several obstacles that hindered follow up and some concrete actions to increase organizational knowledge.

One of the obstacles identified was uncertainty of who was responsible to act on the decisions made at the retrospective. One of the case studies under observation revealed that issues were never addressed and improvements never came, thus issues repeated over several retrospectives. This can be hazardous and teams may lose their will to perform retrospectives, only seeing them as a waste of time.

Another obstacle identified was that tactical decisions took priority over strategic ones. Tactical decisions, things to meet short term goals, was considered by agile teams to be more demanding and important than strategic ones. This resulted in the people responsible for follow-up upon the actions from the retrospective rather would consider daily tactical tasks rather than the strategic ones from the retrospective. This again can result in improvements never occurring and people losing the will to perform retrospectives.

Collier et al. [4] suggests four actions which would increase the organizational knowledge. The first action should be storing all the outputs from the retrospective. The second action recommended is to categorize all lessons learned by their functional area. Then assign each item to a specific person for the next project and make him report back to leadership on its progress. The third action is for one of the team members to present the results from the postmortem for the upper management at their regularly organizational reviews. The final action recommended by Collier et al. is to assign each lesson learned to a person in the organization who will be held responsible for investigating and implementing a solution.

10.2 Documentation

After the retrospective several articles [4,6,8,11] suggests writing a report from the retrospective meeting. The articles varies in the terms of focus while documenting, but it is not however mentioned that the reports should vary depending on the development methods used.

Dingsøyr [6] elaborates that if one of the goals of the retrospective is to transfer knowledge to people outside the project one should write a report that explains what went well, what went wrong, the causes and what was said during the meeting. If the intention is to only find improvement actions these should be documented. Dingsøyr also recommends to consider the size of the company. If the company is small less time should be spent on codifying the knowledge into a report. But if it is a big company one should use the time needed for codifying the knowledge.

Collier et al. [4] recommends another format for the documentation of the report. First the report should contain a brief project description including the unique aspects of the project. The next should be a summary of the positive findings of the postmortem. Then a summary of the three worst

factors that hindered the team from reaching their goals. And finally the report should include one improvement action, describing the problem and the actions to counter it.

11 Enthusiasm for the Agile Retrospective

The second main view of our literature review is about enthusiasm. Throughout most of the articles we read the responses from participants was mentioned, if only briefly.

In a case study [8] assessing the retrospective using KJ-sessions and Fishbone diagrams feedback from the participants was given. They believed that it was useful, and it was easy to grasp so everybody could participate. They also said they felt like they got something back for participating. New insights and ways to improve their job were some examples given by the researchers.

In a controlled experiment performed by Bjørnson, Wang and Arisholm [3] on students performing postmortem reviews, the researchers got feedback on the impression of the methods used. All the students participating responded with positive impressions regardless if they performed fishbone diagram or root cause analysis. The authors did however observe that the enthusiasm was lower when performing cause analysis than when doing KJ-sessions. This seems to fit well with the “lesson learned” by Stålhane et al. [8] that some developers did not participate much in the root cause analysis.

However not all are satisfied with the retrospectives. Drury et al. [7] reflects on their findings:

To some, the Iteration Retrospective appears a waste of time because people just talk about issues but no one takes them on board to make changes in the process in future iterations.

This shows that even though many are happy with performing retrospectives

there are some that just sees it as a waste of time.

Part IV

Discussion

Using a “grounded approach” inspired process we identified several views on the agile retrospective. We will now discuss our results and provide an overview over the research domain of agile retrospectives.

12 Retrospective Practices

12.1 Planning the retrospective

The first thing we want to discuss is the planning of a retrospective. Dingsøyr [6] presented several steps to planning a retrospective. They included deciding to hold a retrospective, declaring the intent, find the participants and selecting the exercises, before collecting objective data on the project. We believe that this is a healthy process of conducting a retrospective. A retrospective should not be held solely on the reason of holding a retrospective.

12.1.1 Wish to Improve

To achieve the most out of a retrospective one should first have a wish to improve some aspect of one's development process, thus deciding to hold a retrospective meeting.

12.1.2 Declaring a Goal

When one has decided to hold a retrospective one should further plan the intent of the retrospective. You cannot plan without a goal, and without a plan most things tend to not reach their full potential. We believe retrospectives are no exception to this. The fact that several articles [1, 3, 6, 8, 10, 11]

describes concrete planning steps or suggest using external facilitators seems to fit well that one should plan for the retrospective.

12.1.3 Participants

Who to include in the retrospective meeting provides multiple concerns to consider. The current literature differentiates between four types of participants: Team member, managers, customer and external facilitator. The four types have different roles in the retrospective and should be included based on what the goal of the retrospective should be.

The team member takes part as a participant contributing knowledge and insight during the meeting. According to Dingsøyr [6] one wants to include as many as possible to get different kinds of input. Team members should always participate during retrospective.

Managers can take part as either a participant contributing knowledge or as an internal facilitator leading the meeting. However it is not recommended by Dingsøyr [6] to include managers as the other team members can be intimidated to not honestly share issues as the manager is also the one evaluating them. If one want as much knowledge as possible one should not include managers during retrospective meetings.

The customer can participate as a knowledge contributor during retrospectives. It is however not recommended [6] to include the customer as the focus then can change from process improvement to customer relations. If the goal however is to improve customer relations we assume bringing the customer along would be a good idea, even though we haven't found any studies bringing up the subject. But in any other case the customer shouldn't participate.

The external facilitator participates in a facilitating role. Through encouraging discussion, leading activities and providing a fresh set of eyes the facilitator can improve retrospectives by providing neutral ground and pro-

vide new angles to look at problems. If available one should include an external facilitator to improve the value of the retrospective.

The four types of participants in a retrospective provides an overview of whom one should include as can be seen in Table 11. It is recommended to include as many team members as possible and use an external facilitator. It is however not recommended to include either managers or customers into the meetings, as this can hinder an open arena for discussion.

Table 11: Participants, their roles and whether they should be included in a retrospective meeting.

Participant Type:	Role:	Should Participate?
Team member	Knowledge contributor	Yes, as many as possible.
Manager	Knowledge contributor	No, can hinder open discussion
	Facilitator	
Customer	Knowledge contributor	No, changes focus from process improvement to customer relations
External facilitator	Facilitator	Yes, if possible

12.1.4 Preparing the Retrospective

When all goals are set, participants decided and objective data collected one need to decide which techniques one should use during the actual meeting. The results of our literature review revealed several techniques to use. The choice of which technique to use should reflect the goal one has set for the retrospective. We have through our analytical approach grouped the different techniques listed in subsection 9.1 into four types of goals that represents

the process. These goals are “Identify issues”, “Improve estimations”, “Feedback” and “Problem understanding”. We will elaborate on these goals and the techniques defined within them in the following sections. In Table 12 we have provided an overview of the different techniques, with their goals, costs and outputs attached.

Identify Issues By identifying issues we mean that the technique helps discover issues currently present within a development process. These issues can both be positive and negative. The techniques that we’ve categorized to have the goal of identifying issues are: KJ-sessions, Keep Problem Try (KPT), Timelines and Evidence based Timelines. Each of the four techniques provide a way of identifying issues by different means.

The three techniques: KJ-session, KPT and Timelines, lets the participants identify the issues occurring in a current project. This is done by letting the participants write down their own issues and presenting them to the group. The thing that differs between these three techniques is how the issues are grouped after being identified. In KJ-session the issues are grouped by means of whether they are related to each other resulting in issue groups that can be analyzed. In KPT the issues are grouped by whether they are positive or negative. Positive issues are grouped as Keep while negative are grouped as Problem. Using these one performs an informal analysis that results in improvement actions, grouped as Try. In Timeline the issues are placed in the order of where they occur chronologically in the timeline. This can help identify interesting patterns that can be analyzed. These three techniques we identify as low cost as no to little preparation is needed and one can perform the practice quickly if wanted.

The fourth issue identifying technique is evidence based timelines. This technique differs from the others as it require the facilitator to present the issues for the participants. This is done by the facilitator placing issues along

a time line based upon hard data acquired through the project/iteration. For each issue presented by the facilitator a short discussion commence to assure that all agrees. If some issues are wrong or some issues are missing the participants are required to provide corrections to the issues or add the issues. When the timeline is complete it can help identify interesting patterns that can be analyzed in similar way to the timeline technique described above. As this practice require a substantial amount of preparation from the facilitator, but the same amount of time as the techniques described above, we classify this as medium cost process.

Through our literature review we did not find any comparisons between these four method of identifying issues, neither empirical or experience based. Studies reveals only the feedback received while using one of the methods. Interestingly all techniques have positive feedback, which in turn leads us to believe that regardless of which technique one uses it will give some valid output. This implicates that one can choose which technique oneself prefers.

Improve Estimation The goal improve estimations is about being able to more correctly estimate the time needed for completing a type of task. The only techniques we have identified having this goal is the estimation retrospective. By walking through all tasks completed and comparing the actual timed used against the estimate one gains insights how one should estimate for the future. We believe that improving the estimates can be highly valuable as it can improve efficiency through making more efficient and correct plans for the development process.

Feedback Even though most issues can be seen as feedback given to the team or the process we came across one method solely providing feedback to each individual member of the group. This technique, described by Kinoshita [9] is “Positive Strokes”. Positive strokes allows each team member to write

positive feedback down on a card which will be given to the member obtaining the feedback. Kinoshita conducted this practice only in cases where the team changed, by exchanging members or delivering a release. The reason for this, we believe is that the feedback would become more valuable for each member. If feedback was to be given each week the same positive strokes would occur again and again which we believe would reduce the enthusiasm for it.

Problem Understanding The final goal we've identified for the techniques is "Problem understanding". We defined problem understanding here as the means of analyzing an issue and obtain new insight about the issue before one derives improvements actions from these new insights. The techniques we have identified as having these goals are: Experience Reports, Timelines, Evidence based timelines, Fishbone Analysis and Causal Map Analysis. Looking at the different techniques we found two types of problem understanding: pattern analysis and issue analysis.

Pattern analysis is done by both timelines exercises. By identifying patterns and find causes for how the patterns emerge an analysis is performed. The causes found during this analysis can further be used to find improvement actions, by countering/improving the actions.

Issue analysis is performed by experience reports, fishbone analysis and causal map analysis. In experience reports issue analysis is performed when one derives improvement actions from the issues presented. This is performed solely by the project leader while filling out a template as described in subsubsection 9.1.4. In the fishbone analysis the issues are derived into causes and subcauses which results in the issue analysis. These causes and subcauses can then be used to find improvement actions. The same analysis is used during causal map analysis, but with another method. The participants are separated, in similar way to the KJ-session, writing down causes and subcauses before presenting them to the others while creating a map of

the issue and all the causes related to it. These issues can then be used create improvements actions in similar way of timelines and fishbone analysis.

During our studies we found some studies that compared some of these techniques. Dingsøyr et al. [11] compared Experience reports against what they called lightweight postmortem review, which in turn consists of KJ-sessions and Fishbone analysis. They found that the methods found very little overlapping experience. The experience report revealed more information related to contract issues, design and technology, while the postmortem revealed experience related to implementation, administration, maintenance and developers. They continue however to mention that both methods is usable and the experience given from either is valuable for problem understanding.

Another study compared the fishbone analysis against the causal map analysis through an experiment performed on students. Bjørnson et al. [3] found through this experiment that the causal map analysis provided more room for freedom in the diagram allowing a deeper analysis of the issues. It also included more input from the participants as they all was forced to participate during the analysis. It is mentioned that if an experienced facilitator would have been present these results might differ. They finally conclude that causal map analysis is a more effective way of performing analysis than fishbone analysis especially for small teams without any experienced facilitator. No other comparisons of the methods were found.

12.1.5 Collect Data

After determining the goal, deciding on techniques and inviting the participants, data has to be prepared for the meeting. This data can be related to the project in general, needed for techniques or collected through surveys to prepare the facilitator for leading the retrospective meeting. Regardless of what the data is in relation to it has to include both positive aspects

Table 12: An overview table of different retrospective techniques

Technique	Goal	Cost	Output
KJ-Sessions	Identify issues	Low	A set of grouped issues
Keep, Problem, Try	Identify issues	Low	Set of positive issues Set of negative issues Set of improvement actions.
Estimation Retrospective	Improve estimations	Low	Insights for estimations
Experience Reports	Problem understanding	Low	Experience related to contract issues, design and technology
Positive strokes	Feedback	Low	Individual feedback to each team member
Timelines	Identify issues	Low	Chronological outline of issues
	Problem understanding		Improvement actions
Evidence based timelines	Evidence based issue identification	Medium	Chronological outline of evidence based issues
	Problem understanding		Improvement actions
Fishbone	Problem understanding	Low	Issue analysis Improvements actions
Causal map analysis	Problem understanding	Low	Issue analysis Improvements actions

and negative aspects. This according to Stålhane et al. [1] will help avoid forgetting working practices.

Several methods are described for collection the objective data before the

retrospective. Survey is an efficient, quick and low cost process for obtaining data according to Collier et al. [4]. It helps establish the scope for the retrospective and severity of different issues. It also returns quantitative data for organizational use.

Interviewing is another method described by Stålhane et al. [1]. It is a powerful way of gathering information, and set this information in a personal context that can be beneficial in cases where personal feelings have to be included. It is however more costly as it requires more time from both facilitator and participants.

The last method mentioned by Collier et al. [4] is collecting objective information. Objective information is valuable for the retrospective for four reasons according to Collier et al. It will help identify the biggest improvement opportunities. It gives easier comparison between different projects. It gives feedback on scheduling efforts. And it helps focusing discussions.

By using various techniques one can collect data to help prepare and improve the retrospective. In Table 13 we provide an overview over the techniques described above.

Table 13: Techniques for collecting data before a retrospective

Technique	Cost	Main Benefit
Survey	Low	Helps establish scope of retrospective as well as severity of issues.
Interviews	High	Obtains data in a personal context
Objective Data Collection	Low	Will help identify larger improvement opportunities. Helps focusing discussions.

12.2 During the Retrospective

During a retrospective meeting there are several issues to be aware of. As a facilitator maintaining an open arena for discussion can be difficult. Or remembering to assign decision owners for improvement actions may be easy to forget. We will discuss considerations like these in the following sections. We have also provided a table over possible considerations to take notice of, Table 14

Table 14: Different considerations during the retrospective

Consideration	Reason
Open discussion	One want to identify as many issues as possible and to do that it requires that the participants are allowed to freely raise their issues.
Not make taboo	The participants must be able to freely raise their issues without the risk of being ridiculed or criticized.
Positive and negative issues	If one tend to only focus on the one part the other will be forgotten and can result in either good practices being forgotten or improvement opportunities not identified.
Assign decision owners	Makes the follow-up on improvements actions more substantial.

Maintaining an open discussion is important to get the most value out of a retrospective. According to Dingsøyr [6] one want hinder critique of individuals and dominating participants using up all the time. He also suggests that for longer retrospectives using activities to create safety might be beneficial. Kinoshita [9] also raises the same issue by recommending not making any taboo within the group. If the members are afraid of being ridiculed or

criticized they won't raise their issue. By creating safety one will be able to let the participants freely raise their issues and thus identify more issues.

A. Birk et al. [1] suggests that while performing a retrospective one shouldn't limit experience gathering to the project's negative aspects. One should also identify the positive aspects as well or they might go unnoticed. If only positive issues were raised no improvement actions would be identified. If only negative issues were raised it would lower enthusiasm, and practices working great might be forgotten in the process of improving.

Assigning decision owners can increase the value of the retrospective as decision owners are responsible for follow-up on the decisions made during the retrospective. According to Conboy et al. [7] agile development teams will not act on decisions if there is no owner as every member believes the rest of the group will do it and thus it doesn't get done. By allowing decisions to get owners one improve the chances of following up on the improvement actions as they are decided during the retrospective meeting.

12.3 After the Retrospective

Even though the retrospective meeting is completed the work is not completely over.

12.3.1 Follow up

Through the retrospective meeting one has identified different issues and actions to resolve these. Until the next retrospective the job of resolving these issues has to be done.

Through our literature review we found only one clearly defined action to help improve resolving identified issues. The one action clearly defined was to assign owners to the improvement actions. By doing this several articles [4, 7] assumes that it will help resolve these issues, as when an owner of an

issue is held responsible for the issue being worked on. If no owner would be assigned other tasks would get prioritized. Thus the issue would never be resolved. By declaring an owner for an action resolving an issue, it is assumed it will be resolved. However we have not seen any research been done in this part of agile development and we believe the lack of defined actions may be the cause for issues recurring throughout several retrospectives.

12.3.2 Documentation

The second activity that is due after the retrospective is documenting it. Different research suggests different kinds of documentation. Some suggests [4] writing lightweight reports containing project description, positive issues, negative issues and actions to resolve the negative ones.

Dingsøyr [6] recommends considering the size of the company as well as the distribution of the report. If the company is big then one should use time to codify the knowledge, but if the company is small one could do with only write a short report as the knowledge will be spread anyways.

13 Enthusiasm

In this section we discuss some of the aspects of enthusiasm in the team, and how retrospective planning can improve the attitude of the team members. Since the input of the participants is the main source of knowledge during the retrospective engaged and enthusiastic participants are preferable.

Positive to Practices One potential action for the management is to reinforce the positive environment surrounding the retrospective meeting. Tessem [14] mentions rewards, but does not find a large amount of data in the category. A reward mentioned is a free lunch. By relating the holding of

the retrospective with good free food it might be possible to make participants look forward to the retrospective, and relate it with something positive. Some teams consider working in agile teams enjoyable, and reinforcing retrospectives as a part of the agile process might lead to team members feeling like they are in control of their own project, and therefore more content. Hanssen [8] describes the need for a well planned and focused retrospective in order to be productive. If a retrospective is not productive it might lead to team members thinking it is a waste of time as seen in the Drury et al. article [7]. In order to stop a retrospective becoming a waste of time it is important to follow through with the proposed changes and make changes to future iterations.

Recurring Issues Kills the Joy The lack of commitment to following through on implementing improvements to deal with issues can lead to recurring issues, as described in a case study by Drury et al. [7]. In the case study the team became focused wholly on delivering content instead of implementing improvements that would address issues. Due to time pressure no team member felt they had the resources or time to take ownership of the problem and its solution. This lack of ownership was reduced by being clear on who were the owners of the decisions made during the retrospective. This lack of clarity had led to the team missing out of potential improvements to their work process.

14 Conclusion

Throughout this article we have evaluated agile retrospective practices and found their weaknesses and strengths based on existing literature. Our findings revealed that practices today provides a big variety of techniques each fitting for different kinds of goals. There is also a difference in how teams

appreciate retrospectives, we believe that careful planning and clear management can go a long way in increasing the appreciation for a retrospective. We do however believe there is a gap in current research considering the follow-up of issues identified in agile retrospectives. There seem to be few clearly defined processes/actions that handles this area. And we believe that this may be a reason for the falling enthusiasm of the retrospective.

Our contribution has been to create an overview of the existing literature on the research on agile retrospectives, where we have evaluated the strengths and weaknesses of practices and identified a possible gap in current literature. For practitioners this article can serve as a reference guide, as well as an overview of retrospective themes and exercises.

Acknowledgment

We would like to thank SINTEF and Teknologihuset for providing us with offices and a place to conduct a workshop. We also want to thank the researchers at SINTEF ICT for answering our questions whenever we had any, and also the studygroup that have met every Monday for the past six months providing valuable feedback and new insights. Especially we would like to thank Torgeir Dingsøy for all his help as a supervisor, providing valuable feedback, answering questions and supporting us with advice when it has been needed.

References

- [1] A. Birk, T. Dingsøyr, and T. Stålhane. Postmortem: never leave a project without it. *IEEE Software*, 19(3):43–45, May 2002.
- [2] Elizabeth Bjarnason and Björn Regnell. Evidence-Based Timelines for Agile Project Retrospectives – A Method Proposal. *XP 2012*, pages 177–184, 2012.
- [3] Finn Olav Bjørnson, Alf Inge Wang, and Erik Arisholm. Improving the effectiveness of root cause analysis in post mortem analysis: A controlled experiment. *Information and Software Technology*, 51(1):150–161, January 2009.
- [4] Bonnie Collier, Tom DeMarco, and Peter Fearey. A Defined Process For Project Postmortem Review. *IEEE SOFTWARE*, 1996.
- [5] Esther Derby, Diana Larsen, and Gather Data. Agile Retrospectives: Making Good Teams Great! *Agile Retrospectives*, (2006), 2005.
- [6] Torgeir Dingsøyr. Postmortem reviews: purpose and approaches in software engineering. *Information and Software Technology*, 47(5):293–303, March 2005.
- [7] Meghann Drury, Kieran Conboy, and Ken Power. Obstacles to decision making in Agile software development teams. *Journal of Systems and Software*, 85(6):1239–1254, June 2012.
- [8] Geir Kjetil Hanssen, Nils Brede Moe, Torgeir Dingsøyr, and Tor Stålhane. Post Mortem – An Assessment of Two Approaches. *ES-ERNET 2001-2003*, pages 129–141, 2003.
- [9] Fumihiko Kinoshita. Practices of an Agile Team. *Agile 2008 Conference*, pages 373–377, 2008.

- [10] Michael Maham. Planning and Facilitating Release Retrospectives. *Agile 2008 Conference*, pages 176–180, 2008.
- [11] Nils Brede Moe, Torgeir Dingsøy, and øystein Nytrø. Augmenting Experience Reports with Lightweight Postmortem Reviews. *PROFES 2001*, pages 167–181, 2001.
- [12] Viktoria Gulliksen Stray, Nils Brede Moe, and Aybuke Aurum. Investigating Daily Team Meetings in Agile Software Projects. *2012 38th Euromicro Conference on Software Engineering and Advanced Applications*, pages 274–281, September 2012.
- [13] John Terzakis. Virtual Retrospectives for Geographically Dispersed Software Teams. *IEEE Software*, 28(3):12–15, May 2011.
- [14] Bjørnar Tessem. Individual empowerment of agile and non-agile software developers in small teams. *Information and Software Technology*, 56(8):873–889, August 2014.
- [15] M.J. Tiedeman. Post-mortems-methodology and experiences. *IEEE Journal on Selected Areas in Communications*, 8(2):176–180, 1990.
- [16] Xiaofeng Wang, Kieran Conboy, and Oisín Cawley. “Leagile” software development: An experience report analysis of the application of lean approaches in agile software development. *Journal of Systems and Software*, 85(6):1287–1299, June 2012.