



NTNU – Trondheim
Norwegian University of
Science and Technology

Agile Retrospectives: An Empirical Study of Characteristics and Organizational Learning

Bjørn Dølvik
Alf Magnus Stålesen

Master of Science in Computer Science
Submission date: June 2015
Supervisor: Torgeir Dingsøy, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

NORWEGIAN UNIVERSITY OF SCIENCE
AND TECHNOLOGY

MASTER THESIS

**Agile Retrospectives: An
Empirical Study of
Characteristics and
Organizational Learning**

Authors:

Alf MAGNUS STÅLESEN

Bjørn DØLVIK

Supervisor:

Torgeir DINGSØYR

Spring June 18, 2015

Abstract

Agile retrospectives could help teams learn from the past and identify improvement opportunities. In this thesis we investigate the characteristics of the present retrospective practice and the organizational learning it provides for teams practicing them, through a multiple case-study.

The research is based on a depth-study of five years of retrospective practice from one team, and a breadth-study of seven interviews from other retrospective practicing teams.

Our results show that teams today are mostly satisfied with their retrospective practice and are able to identify improvement opportunities and implement them, which contradicts previous research. However we identify one barrier related to team commitment dependent on enthusiasm, and previous implementation of improvement opportunities that results in a feedback-loop that could both help implement future improvements and hinder them. We investigate the learning happening through today's retrospectives and find that the practice is approximating a learning system where teams are able to test their current work practices, learn from them and improve from them. Where most of the governing values and behavioral consequences for such a system are already present in today's practice. However we find that the primary learning type remains single-loop, even though double-loop should be expected. We identify a barrier that hinders double-loop learning, consisting of several factors and propose a method based on our findings. The method aims to facilitate triple-loop learning and adaption of the retrospective practice, which can lower the learning barrier.

Finally we conclude that today's retrospective practices provide agile development teams the ability to adapt their current work-practices and enable them to learn from past development iterations and thus provide the means for identifying improvement opportunities and improve from them.

Contents

1	Introduction	1
1.1	Background	1
1.2	Agile Development	2
1.3	Retrospective	2
1.3.1	Retrospective Definitions	2
1.3.2	Retrospectives: Earlier Academic Work	3
1.3.3	Retrospective Outcome	5
	Learning	5
	Improvements	5
	Enthusiasm	6
1.3.4	Retrospective Processes	6
	Dingsøy's Approach to Postmortems	6
	Derby and Larsen's Retrospective Structure	8
	Own Framework	9
1.3.5	Retrospective Impediments	11
	Zedtwitz Barriers	12
	Effective Decision Making	13
1.4	Organizational Learning	15
1.4.1	Single-loop	16
1.4.2	Double-loop	17
1.4.3	Model I	17
1.4.4	Model II	19
1.4.5	Triple-loop	20
1.5	Terminology	26
2	Method	28
2.1	Research Goal	28
2.2	Research Design	29
2.3	Depth Study	29
2.3.1	Content Analysis	30
	Data Material - Retrospective Reports	30

	Analysis Method	31
	Content Analysis Limitations	33
2.3.2	Feedback Sessions	36
	Feedback Session: Team Discussion	36
	Feedback Session: SCRUM Master Interview	37
	Feedback Session: Team Leader	37
	Feedback Session: Internal Team Evaluation	37
2.4	Breadth Study	37
2.4.1	Semi-structured interviews	38
2.4.2	Interview Subjects	39
2.5	Analysis Method	39
2.6	Participants	41
2.6.1	Depth Analysis Participant - Team Zulu	41
2.6.2	Breadth Study Participants	41
	Team Alfa	41
	Team Bravo	42
	Team Charlie	42
	Team Delta	42
	Team Echo	42
	Team Foxtrot	42
	Team Golf	42
3	Results	44
3.1	Retrospective Characteristics	44
3.1.1	Key Characteristics	44
3.1.2	Output	46
	Topics Discussed	46
	Improvements	62
	Enthusiasm	67
3.1.3	Processes	68
	Before the retrospective	68
	During the retrospective	70
	After the retrospective	73
3.1.4	Impediments	73
	Team factors	74
	Retrospective timespan	75
	Distributed Team	78
3.2	Organizational Learning in Retrospective Practice	79
3.2.1	Learning Through Retrospective Practice	79
3.2.2	Types of Learning	80
	Learning Types in Depth Study	80

	Learning Types in Breadth Study	82
3.2.3	Reflection About Learning	83
	Practices on Learning Reflection	83
	Reflection on Learning Through Feedback Sessions	84
3.2.4	Learning Obstacles	86
3.2.5	Learning Enhancing Factors and Potential Improvements	87
4	Discussion	88
4.1	Retrospective Characteristics	88
4.1.1	Output	88
	Work Areas	89
	Improvements	90
	Enthusiasm	91
4.1.2	Processes Characteristics	92
	Before	92
	During	93
	After	94
4.1.3	Impediments	94
4.2	Organizational Learning	97
4.2.1	Governing Values	97
	Model I	97
	Model II	99
4.2.2	Learning Types	103
4.2.3	Impediments for Learning	105
	Focus on Double-Loop Learning	105
	Reflection on Learning	106
	Generalizing Knowledge	106
	Internal Commitment	106
	Low Enthusiasm	107
	Little Action Implementation	107
	Tacitness of Process Knowledge	107
	Bureaucratic Overhead	107
	Impediments Reflection	107
4.3	Reflections on Retrospective Practice	109
4.3.1	Current State of the Retrospective Practice	109
	Before Retrospective	109
	During Retrospective	109
	After Retrospective	111
4.3.2	Method Proposal: Meta-Retrospective	111
4.3.3	Guidelines for Conducting Retrospectives	116
4.4	Limitations	118

5 Conclusion and Future Work	119
5.1 Conclusion	119
5.2 Future Work	120
Appendices	125
A Interview Guide	126
B Pilot Analysis	129
C Content Analysis Categories	131
C.1 Nature	131
C.2 Context	132
C.3 Decision Making	132
C.4 Organizational Learning	133
C.5 Development	134
C.6 Collaboration	135
D Shared Mental Models	138
D.1 The stages of Shared Mental Model generating	138
D.2 Shared Mental Models and retrospectives	139
D.3 Shared mental models discussion	139
E Graphs of Results from Content Analysis	142

List of Figures

1.1	Own Framework for Retrospectives	10
1.2	Own framework for retrospectives with existing recommendations from previous literature.	11
1.3	Single-loop learning visualization, based on Argyris and Schön's single-loop learning [6].	16
1.4	Double-loop learning visualization, based on Argyris and Schön's double-loop learning [6]	18
1.5	Triple-loop learning visualization, based on Argyris and Schön's deutero-learning as described by Tosey et. al. [29]	22
2.1	A picture of our whiteboard during analysis	40
3.1	A visual representation of some of the key numbers.	45
3.2	The distribution of negative, positive and undefined actions across the timespan	49
3.3	The distribution of technical, process and undefined related actions across the timespan.	51
3.4	Timeline showing the distribution of the different development phases over time.	53
3.5	Correlation between release and planning actions	56
3.6	Timeline showing the distribution of the different collaboration categories over time.	57
3.7	The total amount of bugfix actions over time.	59
3.8	The total amount of scenario template actions over time.	60
3.9	The total amount of developer-tester communication related actions shown over time.	61
3.10	A timeline showing the distribution of the different decision making decisions for all the actions.	64
3.11	Team changes and the amount of actions for the retrospectives across the 278 Weeks.	76
3.12	Secretary changes and the amount of actions for the retrospectives across the 278 Weeks.	77

3.13	Timeline showing the distribution of learning loops for the total actions.	81
4.1	Our perceived state of the retrospective.	110
4.2	Retrospective state after introduction meta-retrospective . . .	114
4.3	Visualization of meta-retrospective occurrence	115
B.1	An example of a slide from the pilot study presentation . . .	130
E.1	The negative, positive and undefined distribution of all the actions.	142
E.2	The negative, positive and undefined distribution of the active actions.	143
E.3	The distribution of negative, positive and undefined actions across the timespan	144
E.4	The distribution of technical, process and undefined related actions.	145
E.5	The distribution of technical, process and undefined related actions over.	145
E.6	The distribution of technical, process and undefined related actions across the timespan.	146
E.7	The distribution of different decision making decisions as they occurred over all the actions.	147
E.8	The distribution of different decision making decisions as they occurred over the active actions.	147
E.9	A timeline showing the distribution of the different decision making decisions for all the actions.	148
E.10	The distribution of single-loop, double-loop and undefined for all the actions.	149
E.11	The distribution of single-loop, double-loop and undefined for the active actions.	149
E.12	Timeline showing the distribution of learning loops for the total actions.	150
E.13	The distribution of the different development phases for all the actions.	151
E.14	The distribution of the different development phases for the active actions.	151
E.15	Timeline showing the distribution of the different development phases over time.	152
E.16	The distribution of different collaboration categories for all the actions.	153

E.17	The distribution of different collaboration categories for the active actions.	153
E.18	Timeline showing the distribution of the different collaboration categories over time.	154

List of Tables

1.1	Types of retrospectives	3
1.2	Dingsøy's Approach to Postmortems	7
1.3	Derby and Larsen's Retrospective Structure	8
1.4	Zedtwitz [31] Barriers	12
1.5	Decision making levels	13
1.6	Drury barriers	14
1.7	Theory-in-use of Organizational learning I borrowed from "Organizational learning II: Theory, Method and Practice" by Argyris and Schön [6]	23
1.8	Theory-in-use of Organizational learning II borrowed from "Organizational learning II: Theory, Method and Practice" by Argyris and Schön [6]	24
1.9	Social Virtues of Model I and Model II borrowed from "Organizational learning II: Theory, Method and Practice" by Argyris and Schön [6]	25
1.10	Thesis terminology	27
2.1	Research design for this multiple-case study.	30
2.2	The section of the retrospectives	31
2.3	An generic example of an action provided in the retrospectives.	31
2.4	The final set of themes and categories that the content analysis is based upon.	34
2.5	Description of the content analysis steps.	35
2.6	Interview guide overview	38
2.7	Table overview over interview participants	43
3.1	Some key numbers from the retrospectives	45
3.2	Discussion Areas	47
3.3	Analysis results from the content analysis for the nature of the action.	48
3.4	Analysis results from the content analysis for the context of the action.	50

3.5	Results from the content analysis in which development phase the action regards.	52
3.6	Results from the content analysis regarding the collaboration influences of an action.	55
3.7	Analysis results from the content analysis for the decision making perspective of the action.	63
3.8	Action Follow-Up Techniques Used	66
3.9	Usage of external facilitator	69
3.10	Retrospective Techniques Used	71
3.11	Frequency and duration for the different teams	72
3.12	Value Decreasing Hinders for the Retrospective	78
3.13	Results from the content analysis regarding the organizational learning nature of the action.	80
3.14	Root-Cause identifying techniques used.	83
3.15	Learning Reflection	84
3.16	Retrospective feedback report summary	85
3.17	Obstacles for learning	86
4.1	Retrospective characteristics	89
4.2	Governing values and consequences encountered in relation to retrospectives.	100
4.3	Governing values and consequences from Argyris and Schön's Model II encountered in relation to retrospectives.	104
4.4	Learning impediments	108
D.1	Four stages of mental model building	139

Foreword

This thesis is written by Alf Magnus Stålesen and Bjørn Dølvik. It is written as a completion to the Computer Science master at the Norwegian University of Science and Technology. The master focuses on agile retrospectives and their relation to organizational learning.

This thesis was written in the spring of 2015, under the guidance of our supervisor Torgeir Dingsøy. We also received help and advice from Nils Brede Moe. Both Dingsøy and Moe were invaluable in the writing of this thesis and provided constant guidance and advice that were of enormous benefit. We like would to thank both deeply!

Further, we wish to thank all participants of the studies we performed.

Also we want to thank the staff at SINTEF ICT, for providing us with offices. Also they proved to be invaluable discussion partners during countless coffee breaks.

Finally we wish to thank our study group who have met almost weekly during the whole semester, and provided us with countless ideas and inspirations.

Chapter 1

Introduction

In this section we will write a short overview over themes and terms relevant for this study. Also we will introduce theory that both serves as a motivation and basis for the work seen later in this report.

1.1 Background

With the increasing demand of expected delivery from customers, many software development teams adopted agile methodologies as a development practice. Agile methodologies provide shorter feedback-loops than traditional waterfall and they allow teams to adapt both their products and working processes in short intervals. With the introduction of these methodologies a practice called “Retrospective” was introduced. It’s roots based on reflective practices from organizational learning such as post-project reviews, experience reports and after action reviews. The purpose of the retrospective is allowing teams to look back at their last development phase and learn from it. Learning from the last period of time could provide useful knowledge that could be used to create improvement opportunities for future phases or projects.

There have been several studies, [4,5,8–10,15,23,31], that investigate how to introduce and perform retrospectives. Derby and Larsen [7] even wrote a book on the subject. Literature describing the outcome of the retrospective however is a little bit sparse. Zedtwitz [31] investigated organizational learning through post-project reviews in R&D. Drury, Conboy and Power [11] investigated the decision making in agile teams where the retrospective was one of the practices investigated. However we have not seen any studies that investigates the learning and outcome of retrospectives in an agile software development context.

As the intervals between retrospectives are shorter than that of post-projects reviews we believe that the outcome and learning effects of the retrospective may differ from that of the post-project reviews. Based on this assumption we will conduct an empirical study that investigates the characteristics and learning effects of the retrospective of mature teams. Answering Dingsøy and Dybå's call [12] for empirical studies into mature agile development teams.

1.2 Agile Development

Agile development is a set of working methodologies that is prevalent in the software development industry [28]. Agile development focuses on high adaptability and customer communication, leading to supposedly higher effectiveness and lower costs. Scrum, Kanban and Extreme Programming are examples of popular agile methodologies that are used in today's software development industry. An ideal agile team has a high degree of self control and develop project goals through frequent communication with the customer, without being impeded by external processes. Some of the typical concepts included in an agile project environment are the daily meeting, retrospectives and iterations. The daily meeting is also called a stand up meeting and is typically a short meeting at the start of the day performed standing where team members talk about issues, obstacles and the plan for the day [27]. This is by no means a full description of the agile development methodologies, but intended to give a context for the rest of the article.

1.3 Retrospective

Retrospectives, sometimes also called postmortems, are an activity that aim to improve learning within an organization. We will look at different retrospective definitions, some of the earlier academic work done on retrospectives and the characteristics of the retrospective in terms of retrospective outcome, processes and impediments in the following subsections.

1.3.1 Retrospective Definitions

There are several different definitions of retrospectives. Dingsøy [8] defines postmortem as:

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.

Derby and Larsen [7] defines retrospectives in another way:

A special meeting where the team gathers after completing an increment of work to inspect and adapt their methods and team-work. Retrospectives enable whole-team learning, act as a catalysts for change, and generate action.

We define retrospectives as the following:

Retrospectives is a process that aims to facilitate shared learning within a team or an organization after a learning event, and such create a focus to improve current work practices or teamwork.

We say learning event as different teams holds retrospectives after different kinds of events. We list the kinds of retrospectives known to us in Table 1.1. For this paper our main focus will be related to what we call iteration retrospectives, which is the practice commonly used in SCRUM and KANBAN.

Table 1.1: Types of retrospectives

Retrospective Type	Description
Iteration Retrospective	Held after project iteration
Project Retrospective	Held after a project ends
Feature Retrospective	Held after the release of a feature
Incidents Retrospective	Held after an incident
Back-on-track retrospective	Held when work practices is failing

1.3.2 Retrospectives: Earlier Academic Work

There have been several articles, conference papers, and experience reports on retrospectives or mentioning them as part of agile development.

Dingsøyr, Moe and Nytrø compared light-weight postmortem reviews against experience reports and found that the two provides different kind

of experience. light-weight postmortem focuses on implementation, administration, developers and maintenance, while experience reports yield experience related to contract issues, design and technology.

On the subject of retrospectives techniques several publications have been done. Stålhane, Dingsøy, Hanssen and Moe [15] compared two methods of semi-structured interviews and KJ-session and found that both may be applied to harvest knowledge. Bjarnason and Regnell [4] proposed the evidence-based timelines method as a way of conducting retrospectives. Bjørnson, Wang and Arisholm [5] compared the effectiveness of root-cause analysis against that of fishbone diagrams and found that root-cause analysis was more effective as the fishbone diagram limits the way issue were related to each other. Dingsøy [8] have written an article on the subject highlighting the purpose of the retrospective as well as discussing three possible approaches. Dølvik and Stålesen [13] did a literature review on agile retrospectives and found that different techniques would fit different goals, and they found a lack of research considering follow-up on the issues identified during the retrospective. There also several experience reports on the subject, two of them Maham [20], Kinoshita [17], which describes how to plan and facilitate retrospectives as well as describing different techniques. Finally Derby and Larsen [7] has as previously mentioned a book on the subject.

Other work returns some different findings. Drury, Conboy and Power [11] found in a study on obstacles to decision making in agile development teams that the SCRUM-master prioritizes other tasks than follow-up on actions from retrospectives. They proposed that discussion on decision-making should be a part of the retrospective. And they found that some regarded the retrospective as a waste of time while others found it useful. Comparing to this Stålhane, Dingsøy, Hanssen and Moe [15] found that the participants of their case-study regarded the postmortem as useful.

Zedtwitz [31] identified several barriers for learning in post-project reviews in R&D. Two psychological barriers were identified; inability to reflect and memory bias. Managerial barriers were time constraints and bureaucratic overhead. Epistemological barriers such as difficulty to generalize and tacitness of process knowledge were also identified by Zedtwitz. The last barriers were team-based and identified as reluctance to blame and poor internal communication.

There are several publications mentioning retrospectives or directly addressing them, however most of these studies are related to how one should conduct retrospectives or the purpose of it. However we have not seen any studies addressing the value of the retrospective, which is in part why we have focused to conduct this case study.

1.3.3 Retrospective Outcome

Conducting retrospectives should return some outcome for the teams doing them. Dingsøy [8] describes retrospectives as practice that could improve learning within an organization. Derby and Larsen [7] writes about the benefits of adapting and producing a better quality product and a better working life for team members. Drury, Conboy and Powers [11] however states that the retrospective does not provide any real change for the teams conducting them. Throughout this study we will look at the outcome of the retrospective. We will mainly focus on the learning benefits, but we will also investigate the improvements the retrospective provides and the enthusiasm it brings.

Learning

The learning benefits of conducting retrospectives can be viewed in several ways. Dingsøy [8] describes postmortems through learning as a process of conversion between tacit and explicit knowledge and learning through communities of practice. As a conversion process the participants employ learning through listening to others and thereby socialize and create their own tacit knowledge of the tacit knowledge told by others. Sharing your tacit knowledge externalizes the knowledge and makes it explicit. Through the communities of practice view Dingsøy describes postmortem as an arena for individuals to contribute knowledge towards the community and for the community to reflect on practices.

For this study we want to investigate which kind of learning occurs during the retrospective. To do that we will employ an organizational learning theory developed by Argyris and Schön [6], called Organizational learning II that is described further in section 1.4.

Improvements

Derby and Larsen [7] describes how retrospectives can improve team environments. It can help teams improve practices, handle issues and surface obstacles on a regular basis. Kinoshita [17] describes the importance of feedback that helps the team improve continuously through projects. Drury et. al. [11] states that no real change occurs and that the team do not implement the improvement opportunities as short term tasks are prioritized over the long term ones.

We have not seen any literature or studies on what improvements do occur in the retrospective. What topics are brought to the retrospective? What kind of decisions are made? And what improvements are actually

implemented, and how is it implemented? We will investigate questions such as these and describe our findings throughout this report.

Enthusiasm

Derby and Larsen [7] states that teams are invested in the success of improving their work as the improvements are chosen by the teams themselves and not from upper management. Our own findings [13] from previous literature review found that enthusiasm affect the retrospective. Recurring issues kills the enthusiasm and if the enthusiasm is low enough some might find that the retrospective is a waste of time, as also seen by Drury et. al. [11]. We want to investigate this further through this study and focus on how the retrospective outcome is influenced by the enthusiasm from the team.

1.3.4 Retrospective Processes

There are several approaches and methods on how to conduct a retrospective. We will look at two different approaches to conducting retrospectives, Dingsøy's [8] approach to postmortems and Derby and Larsen's Retrospective Structure [7]. After we have presented these we will present our own approach to retrospective practices and processes.

Dingsøy's Approach to Postmortems

Dingsøy [8] discussed several considerations that companies should consider when conducting postmortem. These eight considerations is shown in Table 1.2.

The first consideration is requirements for a good postmortem process. Dingsøy recommends having an open dialogue, where openness, patience, ability to listen, experimentation with new words and concepts, politeness, formation of a persuasive argument and courage are all present. Having a skilled process leader will help achieve this according to Dingsøy. Also maintaining a good atmosphere is important.

Who to invite is Dingsøy's second consideration. He recommends inviting as many as possible that can contribute with knowledge for future projects to the postmortem. However external stakeholders are not recommended as this will move the focus to stakeholder relations.

Dingsøy's third consideration is homework before the retrospective. Homework can stimulate individual reflection and externalization, but it require more time. If the individuals instead are allowed to reflect during the postmortem the most important issues should be dealt with if there is enough

Table 1.2: Dingsøy's Approach to Postmortems

<i>Approaches</i>	<i>Description</i>
Requirements for a good postmortem process	Open dialogue, preventing critique of individuals and dominating people absorbing the meeting and maintaining a good atmosphere.
Who to invite	Broad audience of project contributors, avoid external stakeholders.
With or without homework	Homework can stimulate individual reflection, externalization, but require more time. Allowing participants to reflect during the meeting should deal with most important issues.
Facilitator	External facilitator is recommended.
Open or structured discussion	Open discussion can take a lot of time and dominating participants can absorb the discussion. KJ-method equally allows the participants to influence the discussion, however can be time-consuming to find names for issue-groups.
With or without management	Not recommended, but with a strong facilitator project manager could be invited.
Output	Describing report if project if goal is to spread knowledge. List of recommendations if improvements are only goal.
Tacit or explicit knowledge	Small companies should share tacit knowledge, larger companies should codify it.

participants. Another possible solution from Dingsøy is to let the facilitator prepare before the retrospective.

The fourth consideration is using a facilitator. He recommends using an external facilitator as it will help get a more thorough picture of the project.

Open or structured discussion is the fifth consideration. Dingsøy describes how an open discussion easily can take a lot of time and be too limited in its focus in terms of number of issues. KJ-sessions are a structured discussion where every participant equally influence the discussion. There is also a risk in terms of time as finding names for groupings of issues may take some time.

Including management in the postmortem is not recommended by Dingsøy. The project manager should however be included if the facilitator is able to hinder him from taking a defensive position to the project. The reason he

should be included is that he can provide a broader overview of the project compared to the participants.

The output should be a list of recommendations or a report describing the project. If the goal is to spread knowledge in the company a short report on the project is appropriate. If the goal is only to improve a list of recommendations should suffice.

The last consideration of Dingsøy is whether the knowledge gained through retrospectives should remain tacit or made explicit. For small companies he recommends that they could easily share the tacit knowledge. For larger companies however he recommends using the resources needed to codify the knowledge and make it explicit.

Derby and Larsen’s Retrospective Structure

Derby and Larsen [7] have structured the retrospective into five steps. The five steps can be seen in Table 1.3 and will be described below.

Table 1.3: Derby and Larsen’s Retrospective Structure

<i>Step</i>	<i>Description</i>
Set the stage	Reiterates the goal for the retrospective and helps participants focus on the work at hand.
Gather data	Gathers a share picture of what happened during the last iteration.
Generate Insights	Allows the team to see the big picture, and delve into root-causes, finding experiments and improvements.
Decide what to do	Provides structure and guidance for the team to plan experiments and actions.
Close the retrospective	Helps the team retaining what they have learned from the retrospective.

Setting the stage is the first step described by Derby and Larsen. The purpose is to help the participants focus on the work at hand. The step include establishing the purpose, method and time frame for how the retrospective is planned. It also includes establishing a safe environment where the participants can have a challenging discussion. Creating a working agreement is recommended so the retrospective can remain a civil arena for collaboration.

The second step is gathering data. Gathering data helps create a shared picture of what happened, expanding the participants individual perspective. It is recommended starting with hard data such as events, metrics, completion rates and so on. Afterward feelings should be gathered so that one can

get an impression on which issues is most important. Gathering feelings also helps avoiding motivation loss or venting situations according to Derby and Larsen.

Generate insights is the third step recommended by Derby and Larsen. The purpose is to help the team generate solutions and experiments that will improve the team's current practices. It is stated that the first solution is most likely incorrect and that the team should maintain a root-cause focus.

The fourth step proposed by Derby and Larsen is deciding what to do. After the former steps a list of possible experiments and improvements should have been devised. The purpose of this step is to help guide the team plan to implement the changes to the next iteration. It is recommended that one assigns individuals to the actions as it won't get done if just the team is assigned to it.

Closing the retrospective is the final step in Derby and Larsen's structure for conducting retrospectives. They recommend ending the retrospective decisively to avoid the participants energy dribbling away. It is also recommend using just a couple of minutes giving feedback on the retrospective before ending it.

Own Framework

Derby and Larsen describes how to conduct a retrospective and Dingsøyir investigates different considerations one should employ before conducting a retrospective. None of them takes into considerations the outcome of the retrospective. Through our earlier studies [13] we found that the retrospective could be split into three parts: Before, During and After the retrospective, which is shown in Figure 1.1. By including the after retrospective part into the framework we are able to describe the outcome of the retrospective practice which is missing from Dingsøyir's and Derby and Larsen's processes. We will describe each of the three parts and how the framework aligns with Derby and Larsen's and Dingsøyir's processes below.

Before the retrospective, preparation is required no matter how one conducts the retrospective. At the very least one must have a wish to improve and invite the participants for the retrospective. Depending on how one want to conduct the retrospective one could also gather data or evidence beforehand and declare a focus for the retrospective. The participants are also required to have some experiences that could be used as input for the retrospective. During the preparations for the retrospective Dingsøyir's considerations could be a valuable thing for the facilitator to investigate. It can help improve the retrospective and maintain the focus the facilitator or team wishes to have for the retrospective.

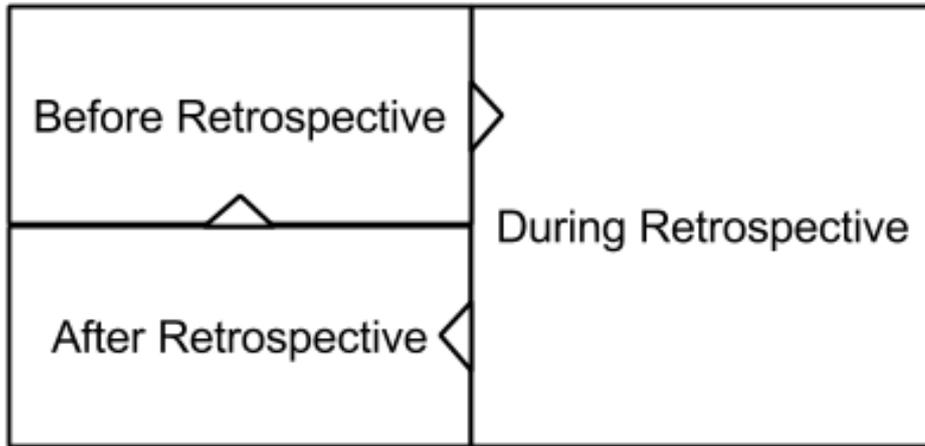


Figure 1.1: Own Framework for Retrospectives

a

During the retrospective one wish to reflect about the past to find improvement opportunities and learn. Several techniques are recommended by literature such as KJ-sessions, timelines, evidence-based timelines, root-cause analysis and open discussion. The focus is to learn from past work and find ways to improve in an open arena together in the team. Derby and Larsen's five steps is one way to arrange the retrospective meeting and will help maintain a focus for the retrospective.

Finally we have after the retrospective where the outcome, which is described in subsection 1.3.3, of the retrospective should be apparent. This period lasts from the retrospective meeting ends until the next one begins. During this time experiments and improvements should be implemented. This should result in some learning which can be brought for the next retrospective and some enthusiasm as the work-life of the team improves. We have not seen any framework that consider this phase of the retrospective in any literature previously and is one of the reasons this framework is devised.

Our framework is not exclusive with the existing frameworks found in previous literature. Dingsøy's list of recommendations and Derby and Larsen's process of conducting retrospective are both able to exist within this framework where one consider the retrospective cycle. In Figure 1.2 we have displayed the framework and its relation to previous processes and the expected outcome of the retrospective practice. We will use this framework to describe the results throughout this study.

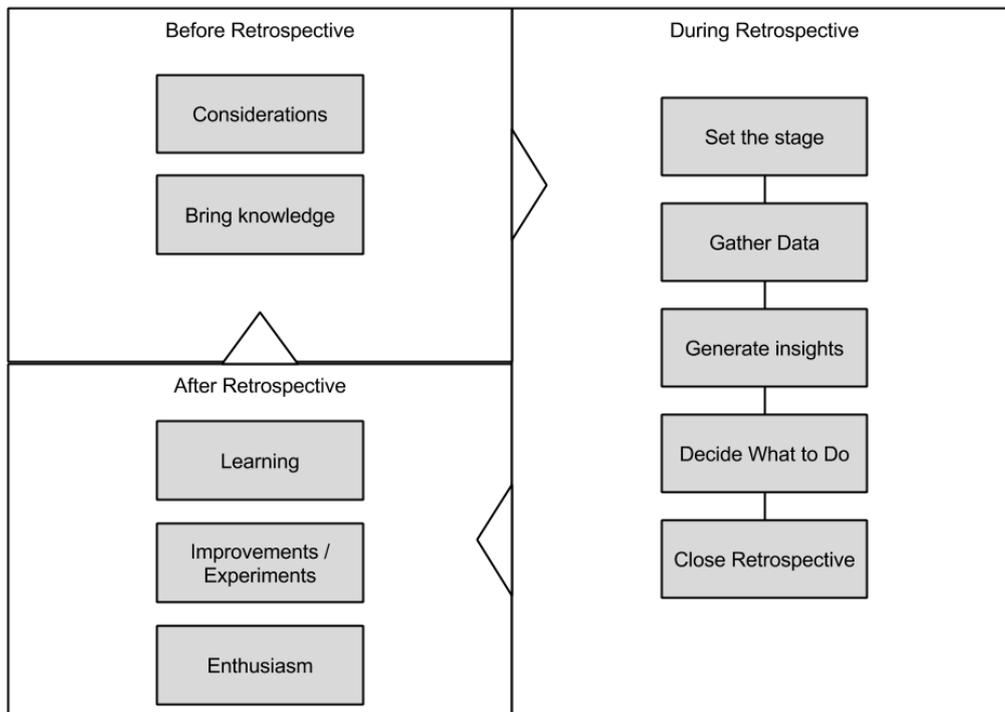


Figure 1.2: Own framework for retrospectives with existing recommendations from previous literature.

1.3.5 Retrospective Impediments

Retrospective practices is a simple process that aims to inspire learning and create improvement opportunities for teams conducting them, however it suffers from impediments. Not a single company was satisfied with their after project reviews was one of the results Keegan and Turner [16] found after conducting a survey. Zedtwitz investigates organizational learning through post-project reviews and finds eight barriers for learning in the practice. Drury, Conboy and Power [11] finds six obstacles to decision making that hinder improvements to be implemented from the retrospective. We will describe Zedtwitz barriers and Drury's et. al. obstacles below and through this study impediments facing the retrospective will be investigated and the results described.

!h

Table 1.4: Zedtwitz [31] Barriers

Area	Barriers
Psychological	Memory bias Inability to reflect
Managerial	Time constraints Bureaucratic overhead
Team-based	Reluctance to blame Poor internal communication
Epistemological	Difficult to generalize Tacitness of process knowledge

Zedtwitz Barriers

In an article Zedtwitz identifies four areas with a total of eight barriers that act as a resistance to learning through post-project reviews [31]. These four areas are psychological, team based, managerial and epistemological. An overview over the areas and their related barriers can be seen in summarized Table 1.4 based on a figure from Zedtwitz's article.

Psychological barriers The memory bias barrier is based on the human tendency to place a subjective importance on our experiences, for example Zedtwitz mentions an over emphasis on negative experiences, as these are easy to remember.

An inability to reflect is the second psychological barrier, this might be caused by a belief that past experiences are irrelevant as circumstances change. It was found that often managers are unaware of the past when it came to actions and rationale.

Team-based Shortcomings The first team based barrier is a reluctance within the team to blame, leading to deflection of blame from the real issues. This might be as a result of an aversion to accepting personal failure or fear of embarrassment.

Poor team internal communication is the second barrier, a team might intentionally hide information to gain an advantage over other teams when it comes to performance or promotions. Other factors might be physical distances making communication harder.

Epistemological The first barrier is a failure to generalize experiences from a specific context can lead to no experiences being transferable to future projects.

The second barrier is the tacitness of knowledge. The barrier addresses that knowledge cannot be shared easily. And reports or similar mediums might not reflect all the knowledge.

Managerial Constraints Time constraints are a barrier that might lead to retrospectives not being held. This is a result of time being considered critical and retrospection and reflection being down prioritized.

The last barrier is bureaucratic overhead, it might be hard to justify a retrospective session to bureaucracy because of a lack of immediately profitable output.

Effective Decision Making

The term decision making refers to the process of a team making a decision that impacts the team or its environment. In their article Moe et. al. [22] talk about three layers of decision making, operational, tactical and strategic. These levels are described in Table 1.5.

Table 1.5: Decision making levels

Decision level	Description
Operational	Day to day operations.
Tactical	Resource management.
Strategic	Goal and objectives, often long term.

Software development can be both routine and non-routine, and the decisions that relate to this is rational decision making and non-bounded decision making. [22] Where rational decision making is assuming that decision makers have complete information and possible solutions, and can therefore choose the decision most suitable. Rational decision making is often related to routine decisions. When goals are changing it makes more sense to talk about non-routine decisions, and these can be described within the bounded rationality model. In this model the decision maker is restricted, in information and problem types.

Drury et. al. [11] discusses obstacles to decision making present in agile software development teams. They specifically describes how an iteration retrospective can result in decision making that leads to improvement of future

performance. However this is not necessarily the case as some retrospectives can become a place to vent frustration instead of productive decision making. A summary on the barriers identified by Drury et. al. can be found in Table 1.6

Table 1.6: Drury barriers

Barrier	Description
Team members unwilling to commit and relying on scrum master	A lack of commitment in a team leading to the burden being put on the coach, scrum master or managers.
Conflicting priorities	A team might have multiple priorities or customers that lead to difficulty setting priorities
Decisions based on unstable staff availability	If people are moved to another project this can lead to problems with the strict estimations typical in an agile project
Not implementing decisions or relying on others for decisions	The failure to implement an agreed upon decision, or expecting someone else to do it leads to decisions falling flat
Not taking ownership of decisions	Teams failing to take ownership of their decisions leading to decision implementation being lackluster, often as a result of a decision not being designated to someone responsible
Collaboration stops experts from making decisions	The empowerment of the team can lead to individual experts not having the control over decisions they had previously

1.4 Organizational Learning

Organizational learning is in simple terms how an organization is able to acquire, store and utilize knowledge existing within an organization. In a more academic setting we can use Argyris and Schön's definition [6] for organizational learning:

“Organizational learning occurs when individuals within an organization experience a problematic situation and inquire into it on the organization's behalf. They experience a surprising mismatch between expected and actual results of action and respond to that mismatch through a process of thought and further action that leads them to modify their images of organization or their understandings of organizational phenomena and to restructure their activities so as to bring outcomes and expectation into line, thereby changing organizational theory-in-use. In order to become organizational, the learning that results from organizational inquiry must become embedded in the images of organization held in its members' minds and/or in the epistemological artifacts (the maps, memories, and programs) embedded in the organizational environment. ”

Organizational learning can be applied to groups of people over different sizes. One can use it to analyze huge organizations consisting of many actors in different roles, or small groups of people working closely together. For this research we are going to apply the organizational learning at an agile development team.

Different types of frameworks exist for organizational learning. Levitt and March uses a framework that sees learning in organizations as encoding inferences from history into routines that guide behavior [19]. The retrospective is a process of shared learning for an agile development team learning from the last iteration of development. Investigating learning using this framework won't yield much as the retrospective already is seen as a collective learning activity [8].

Argyris and Schön [6] uses two different models to determine how an organization learns through individuals, Model I and Model II. As agile teams often are limited to a close group of individuals we will apply Argyris and Schön's models. To understand these models we first need to investigate two types of learning: Single-loop learning and double-loop learning. Further in this section we will explain these types of learning as well as Model I and Model II. We have borrowed Table 1.7, Table 1.8, Table 1.9 from Argyris and

Schön's "Organizational learning II: Theory, Method and Practice" [6] and they provide a quick overview of the two models as well as the social virtues that differs between them. We will also look at Triple-loop learning as it is the concept of learning about learning.

1.4.1 Single-loop

Single-loop learning is a type of learning defined by Argyris and Schön. It consist of a single-feedback loop that finds an error and a way to fix this problem. Argyris and Schön [6] defines single-loop as:

"By single-loop learning we mean instrumental learning that changes strategies of action or assumptions underlying strategies in ways that leave the value of a theory of action unchanged.

... In such learning episodes, a single feed-back loop, mediated by organizational inquiry, connects detected error - that is, an outcome of action mismatched to expectations and, therefore, surprising - to organizational strategies of action and their underlying assumptions. These strategies of action and their underlying assumptions. These strategies or assumptions are modified, in turn, to keep organizational performance within the range set by existing organizational value and norms. The values and norms themselves ... remain unchanged. "

An example of single-loop learning from software development could be finding a bug, find a solution to fix it and then fix it. The single-loop learning would here be learning the solution and fixing the bug. We have provided a basic illustration of single-loop learning in Figure 1.3.

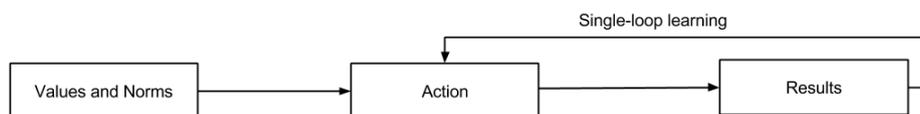


Figure 1.3: Single-loop learning visualization, based on Argyris and Schön's single-loop learning [6].

1.4.2 Double-loop

Double-loop learning is a type of learning where one asks whether the underlying factors that influence the actions and results is sufficient. One might say that one understands the root-cause for the issue. Argyris and Schön defines double-loop learning as:

“By double-loop learning, we mean learning that results in a change in the values of theory-in-use, as well as in its strategies and assumptions. The double loop refers to the two feedback loop that connect the observed effects of actions with strategies and values server by strategies. Strategies and assumptions may change concurrently with, or as a consequence of, change in values. Double-loop learning may be carried out by individuals, when their inquiry leads to change in the values of their theories-in-use or by organizations, when individuals inquire on behalf of an organization in such a way as to lead to change in the values of organization theory-in-use. ”

This type of learning can be shown in our bug example. If the developers who earlier found and fixed a bug did a root-cause analysis on the issue they could get several results indicating that underlying factors are not sufficient for the current state of development team. One underlying factor could be weak specification description requiring the team to rethink and restructure how they develop specifications. Another reason could be that the knowledge for the system is not good enough with the developers indicating a knowledge-gap in the team and considerations for the state of the team may be required.

For this research we use the term double-loop learning when the influences for the issue is understood and change occurs as a results of this. A simple figure of double-loop learning can be seen in Figure 1.4

1.4.3 Model I

Argyris and Schön [6] describes a learning system called Organizational learning I, which theory-in-use is described as Model I. As organizational learning I is an employed learning system that can be defined by different actions and implementations we will instead go into the theory-in-use, Model I, as these systems employ them.

Argyris and Schön [6] describes Model I through its governing variables. We will go through each of them along with the actions strategies, consequences for behavioral world and consequences for learning, effectiveness that follows of the different governing variables that actors try to satisfy.

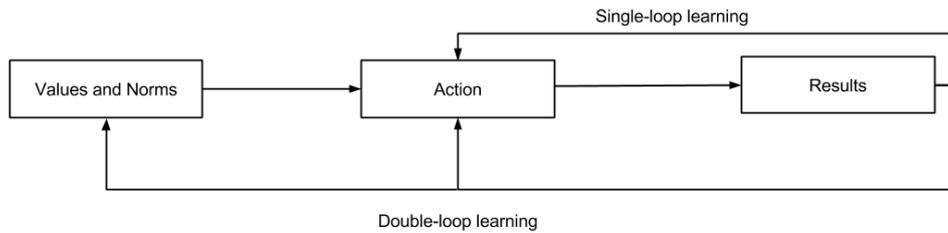


Figure 1.4: Double-loop learning visualization, based on Argyris and Schön’s double-loop learning [6]

The first governing value described is defining goals and trying to achieve them. This leads to the action strategies of designing and managing the environment unilaterally. The behavioral consequences are that the actor gets seen as defensive, inconsistent, incongruent, controlling, fearful of being vulnerable, withholding feelings, overly concerned about self and others, or under-concerned about others. For the effectiveness and learning the consequences are that the actor is self-sealing and the longterm effectiveness is decreased.

Maximize winning and minimize losing is the second governing variable that the actors in an organization try to satisfy. The actions strategies for this governing value is that the actor takes ownership and control of a task. This results in a defensive interpersonal and group relationship where the actor wants the rest of the group to see things his way and little help could be a result of this. For terms of learning this will provide single-loop learning as the output of the task is more important than the task itself.

The third governing value for the actors in this kind of learning model is minimizing generating or expressing negative feelings. The action strategy that follows is that the actors wishes to protect themselves using defensive actions such as blaming, stereotyping and suppressing feelings. The consequences for the behavioral world is defensive norms within the organizations.

Rivalry, lack of external commitment and mistrust are examples of such norms. For the learning consequences of this value little theories will be tested publicly prohibiting double-loop learning in the organization. However there will still be much testing of theories with the individuals, however this knowledge will not reach the organization.

Being rational is the fourth and final governing variable for Argyris and Schön's Model I theory-in-use for the learning organization. Being rational provides the action strategy of protecting others from being hurt resulting in actions like withholding information, creating censoring rules that can censor information and behavior. This returns the same consequences for the behavioral world, learning and effectiveness as described above.

Through the governing values of Model I the actors within the organization prohibits double-loop learning. The organization creates a win or loose situation between the actors leading to withholding of information, mistrust between the actors and little testing of current norms or values. This again provides competition instead of collaboration and thus have a negative impact on the learning and effectiveness of the organization. Model I is further described by Argyris and Schön as the different elements of the model interact in complex ways. This can be read about in "Organizational learning II: Theory, Method and Practice" [6].

1.4.4 Model II

The second model described by Argyris and Schön is the theory-in-use for organizational learning II systems called Model II. While Model I is creating a win-loose relationship between the actors resulting in actors trying to control the environment, Model II instead invites the actors to confront each others views and emotions. This is creating an environment for double-loop learning where the actors strive to get the most complete understanding of their environment and adapt this.

There are three governing values for Model II according to Argyris and Schön: Valid information, Free and informed choice and Internal commitment to the choice and constant monitoring of its implementation.

The three governing values give primarily four behavioral strategies. "Designing situations where participants can be origins of action and experience high personal causation" is one the strategies according Argyris and Schön. Another of these strategies is that the group of actors jointly control the tasks. This provides an environment where face-saving for the individual should not be needed and thus facilitates open and clear communication between the different actors. This again is connected to the third action strategy that is "Protection of self is a joint enterprise and oriented towards

growth". The three governing values also provide an environment of bilateral protection towards others.

There are several consequences for the behavioral world following the governing values and behavioral strategies for Model II. The first one is that actors within the organization act less defensively than in Model I systems and is experienced as minimally defensive. The second consequence is the same as the first one except that it relates to the relationship between the different individuals in the organization. The group dynamics and interpersonal relationships will be minimally defensive. As an organization approaches the practices that supports the three governing values learning-oriented norms will start appearing in the organization as a consequence. The final consequence Argyris and Schön describe is high freedom of choice, internal commitment and risk taking within the organization.

As for the consequences on learning Model II will support double-loop learning. Disconfirmable processes and frequent public testing of theories will both support double-loop learning. As the defensive stance of the actors and relationships are minimized Model II practices will advocate and allow the organization to test their current theories on how their organization is performing, allowing for double-loop learning.

As described by Argyris and Schön the Model II will in the long term increase effectiveness within an organization. Through facilitating collaboration rather than competition, allowing testing of the currently perceived world and norms and allowing experimentation the Model II allows double-loop learning. Being able to adapt the organization through double-loop learning will in the long term increase the effectiveness.

Finally the Modell II theory-in-use is not a goal one can achieve as described by Argyris and Schön. Instead it is a focus one can work against, as the nature of the Modell II and double-loop learning is always to challenge the current world for better one can never achieve the ideal world as this world will always be able to change and improve.

1.4.5 Triple-loop

The term triple-loop learning has been used in different ways as highlighted by Tosey, Visser and Saunders [29]. Triple-loop learning can be referring to both triple-loop learning and deutero-learning. For this work we are going to stick with triple-loop learning. Tosey, Visser and Saunders performed a critical review of the triple-loop learning concept and finds that three concepts have been widely used in academic literature.

The first concept they investigate is a form for learning above double-loop learning that addresses some learning in a level above the governing variable

of double-loop learning. Tosey et. al. finds that this concept seem to be poorly defined, imprecise and unfounded in the way it is used.

The second concept describes triple-loop learning as a form for meta-learning, a form of learning not above single-loop and double-loop, but rather beside. Learning about learning is a simple description. Tosey et. al. finds that this concept is a renaming of deutero-learning and raises the question if this renaming is needed.

The final concept connects triple-loop learning to Bateson's Learning III, which can be read about in "Gregory Bateson on deutero-learning and double bind: A brief conceptual history" by Visser [30]. This concept relates to a type of learning that is non-instrumental and exists beyond language. Tosey et. al. finds that the literature that uses this concept have not been conducting a comprehensive working-through of Bateson's Theory.

For our research we are going to use the second concept, where triple-loop learning is regarded as a kind of meta-learning. We define triple-loop learning as the learning about learning, where one organization is able to understand the learning processes occurring within the company and learn from these. A simple figure is provided in Figure 1.5.

As an example of triple-loop learning we again turn to the bug example used earlier. After understanding both how to solve the bug and the influences which made the bug occur, the development team can investigate their learning process, by for an example retracing their steps from noticing the issue until they have understood the influences which made the bug occur. Along the way they may be able to learn how to investigate future problems or change the way they already learn through their given processes.

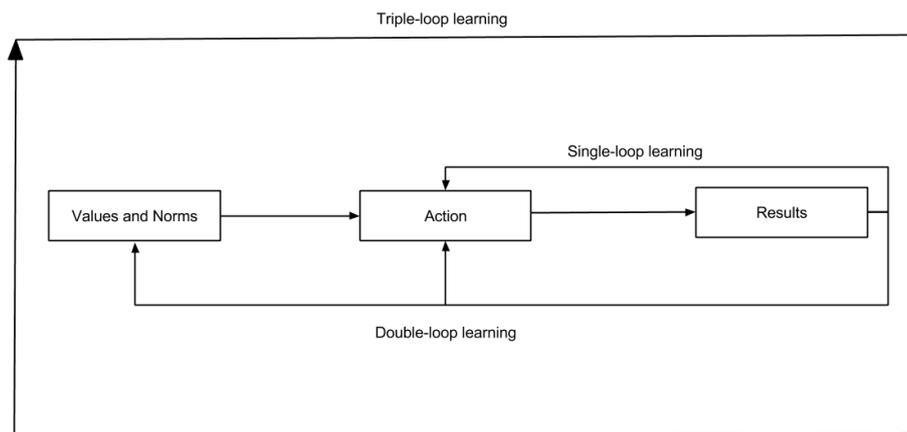


Figure 1.5: Triple-loop learning visualization, based on Argyris and Schön's deutero-learning as described by Tosey et. al. [29]

Table 1.7: Theory-in-use of Organizational learning I borrowed from “Organizational learning II: Theory, Method and Practice” by Argyris and Schön [6]

<i>Model I Theory-in-Use</i>	Action Strategies	Consequences for Behavioral World	Consequences for Learning, Effectiveness
Define goals and try to achieve them.	Design and manage the environment unilaterally (be persuasive, appeal to larger goals, etc.)	Actor seen as defensive, inconsistent, incongruent, controlling, fearful of being vulnerable, withholding of feelings, overly concerned about self and others, or unconcerned about others.	Self-sealing. Decreased Long-term effectiveness.
Maximize winning and minimize losing.	Own and control the task (claim ownership of the task, be guardian of the definition and execution of the task).	Defensive interpersonal and group relationship (depending on actor, little help to others).	Single-loop learning.
Minimize generating or expressing negative feelings.	Unilaterally protect yourself (speak in inferred categories accompanied by little or no directly observable data, be blind to impact on others and to incongruity; use defensive actions such as blaming, stereotyping, suppressing feelings, intellectualizing).	Defensive norms (mistrust, lack of risk taking, conformity, external commitment, emphasis on diplomacy, power-centered competition and rivalry).	Little testing of theories publicly. Much testing of theories privately.
Be rational.	Unilaterally protect others from being hurt (withhold information, create rules to censor information and behavior, hold private meetings).		

Table 1.8: Theory-in-use of Organizational learning II borrowed from “Organizational learning II: Theory, Method and Practice” by Argyris and Schön [6]

Model II Theory-In-Use

Governing Variables for Action	Action Strategies	Consequences of Behavioral World	Consequences on Learning	Consequences on effectiveness
Valid information	Design situations where participants can be origins of action and experience high personal causation	Actor experienced as minimally defensive	Disconfirmable processes	Increased long-term effectiveness
Free and informed choice		Minimally defensive interpersonal relations and group dynamics	Double-loop learning	
Internal commitment to the choice and constant monitoring of its implementation	Task is jointly controlled	Learning-oriented norms	Frequent public testing of theories	
	Protection of self is a joint enterprise and oriented toward growth	High freedom of choice, internal commitment, and risk taking		
	Bilateral protection of others			

Table 1.9: Social Virtues of Model I and Model II borrowed from “Organizational learning II: Theory, Method and Practice” by Argyris and Schön [6]

<i>Model I Social Virtues</i>	<i>Model II Social Virtues</i>
<p>Help and Support Give approval and praise to others. Tell others what you believe will make them feel good about themselves. Reduce their feelings of hurt by telling them how much you care and, if possible, agree with them that the others acted improperly.</p>	<p>Increase the other’s capacity to confront their own ideas, to create a window into their own mind, and to face the unsurfaced assumptions, biases, and fears that have informed their actions toward other people.</p>
<p>Respect for Others Defer to other people; do not confront their reasoning or actions.</p>	<p>Attribute to other people of high capacity for self-reflection and self-examination without becoming so upset that they lose their effectiveness and their sense of self-responsibility and choice. Keep testing this attribution.</p>
<p>Strength Advocate your position in order to win. Hold your own position in the face of advocacy. Feeling vulnerable is a sign of weakness.</p>	<p>Advocate your position and combine it with inquiry and self-reflection. Feeling vulnerable while encouraging inquiry is a sign of strength.</p>
<p>Honesty Tell other people no lies, or tell others all you think and feel.</p>	<p>Encourage yourself and other people to make public tests of their ability to say what they know yet fear to say. Minimize what would otherwise be subject to distortion and cover-up of the distortion.</p>
<p>Integrity Stick to your principles, values, and beliefs.</p>	<p>Advocate your principles, values, and beliefs in a way that invites inquiry into them and encourages other people to do the same.</p>

1.5 Terminology

In this section we have an oversight of the terminology used in this thesis, this is by no means a comprehensive discussion of the terms, but meant to be a reference of words and ideas that might not necessarily be self explanatory or common knowledge. Our terminology can be found in Table 1.10.

Table 1.10: Thesis terminology

<i>Term</i>	<i>Other terms</i>	<i>Description</i>
Retrospective	Post mortem, after action review, post-project review	A process that aims to facilitate shared learning within a team or an organization after a learning event, and such create a focus that aims to improve the current work practices or teamwork.
Sprint	Iteration, re-release cycle, project phase	A time period in a project, often ending at a set date or goal
Shared mental models		Group level thought processes and actions, commonality in understanding enabling consensus and attaining goals
Shared mental model accuracy		The degree with which a team's mental model matches objective measurements
Shared mental model similarity		The degree with which the team member's mental models are similar
Organizational Learning		Academic field of how organizations or teams achieve learning.
Single loop learning	Tuning, quick-fix	An action aimed at improving a process or fixing a problem, often immediately
Double loop learning	Root-cause analysis	An action designed to improve a problem or process by fixing an underlying influence
Theory-in-use		Value, action strategies and norms that act as a theory for some process.
Model I	Organizational learning I	The theory-in-use for a learning system of organizational learning I.
Model II	Organizational learning II	The theory-in-use for a learning system of organizational learning II.

Chapter 2

Method

In this chapter we will describe the environment and motivation for our research. We will also explain the process for this study.

2.1 Research Goal

After over a decade of agile development where retrospectives have been recommended as a practice that should introduce improvements in development practices, the outcome of the retrospective is still undetermined. We will investigate this, focusing on the organizational learning that happens through retrospectives and the characteristics of the retrospective. When we say characteristics of the retrospective we will investigate the output of conducting retrospectives, the processes used and the impediments that faces the retrospective. We will investigate these characteristics in the context of existing academic work. *Our goal for this study is to investigate the outcome returned from the retrospective in terms of organizational learning and retrospective characteristics.* Elaborating on this goal we get two sub-goals:

1. What are the main characteristics in current retrospective practices, in terms of outcome, processes and impediments?
2. How is learning achieved through current retrospective practices, in light of organizational learning theory?

The first sub-goal relates to the characteristics of the retrospective. We will throughout this study investigate and describe which characteristics are current in today's retrospective. We will focus on the output, the processes used and the impediments. When we say output we will investigate what

improvement opportunities are created, which improvements are actually implemented and how the enthusiasm evolves as a result of the retrospective. We will also investigate the processes used by practitioners of the retrospective and the impediments that hinder them from returning value.

The second sub-goal will investigate the learning achieved through the retrospective practice. We will employ the learning theory described by Argyris and Schön [6] and compare the governing values of Model I and Model II against the results we see from our case studies. We will also investigate which types of learning, single-, double-, or triple-loop, occur through the retrospective practice.

2.2 Research Design

For this study we will conduct a multiple-case study to investigate our research goal. It will consist of one depth study and one breadth study. We describe each of these in short in this section and following in the chapter we will elaborate on these. A short summary of the research design can be seen in Table 2.1.

Our first case study is a depth study investigating the long-term practice of one retrospective practicing agile development team. This will be done through analyzing a set of retrospective reports using tabulations and then reflect on the results together with the team.

The second case study is a breadth study investigating the retrospective practices of other teams. This will be done interviewing representatives from different retrospective practicing teams.

Our analysis method consists of two parts. The first part is a descriptive discussion on the results found during the two case studies that focuses on the characteristics of the retrospective practice. The second analysis method is comparing our results against the organizational learning framework created by Argyris and Schön [6] specifically the governing values of Model I and Model II.

2.3 Depth Study

Our depth study consisted of two steps. The first step was to perform a content analysis on earlier retrospective reports. The second step was to hold several feedback sessions with the team reflecting on the results of the content analysis. We will describe each of these steps further in the following subsections.

Table 2.1: Research design for this multiple-case study.

<i>Step</i>	<i>Description</i>
Case 1	Depth Study
Content Analysis	Tabulation Analysis of Retrospective Reports
Feedback Sessions	Reflection with team about the results of the content analysis.
Case 2	Breadth Study
Interview Sessions	Interview different teams
Analysis Method	
Characteristics	Descriptive Discussion on Results
Organizational learning	Compare results against Argyris and Schön's [6] governing values.

2.3.1 Content Analysis

Data Material - Retrospective Reports

Through our supervisor we were put in contact with an experienced team in Norway. The team, which we will refer to as team Zulu is described more closely in subsection 2.6.1, graciously allowed us access to their retrospective logs, for us to analyze.

The retrospectives consisted of five different sections being: Where and What, Actions, Comments, Signatures and Case Proceedings, as can be seen in Table 2.2. The “Where and What” section contained general data about the retrospective such as the date, iteration dates, iteration number, contact person and other general information. “Actions” described the improvement actions that had resulted from the retrospective. It contained a description for each action, who is responsible for that action, deadline, status and comments from the participants. The “Comments” section contained comments, if any, from the participants of the retrospective if they had any specific input for the retrospective in general. “Signatures” contained the signature for each participant in the retrospective. The last section, “Case Proceedings” contained information about the changes in the document and circulation of it.

While getting familiar with the retrospectives we found that the only section containing any value in the terms of organizational learning was the actions described in the “Action” section. In most of the retrospective reports multiple actions were described relating to different issues observed during

Table 2.2: The section of the retrospectives

Part	Description
Where and What	Containing general data about the retrospective such as date, iteration location, etc.
Actions	Describes the actions resulting from the retrospective it also includes data on responsible person, deadline etc.
Comments	General comments from the participants for the retrospectives.
Signatures	The signatures from each participant participating in the retrospective.
Case Proceedings	The changelog and circulation of the retrospective.

the iteration. The format of the actions can be seen in Table 2.3.

Table 2.3: An generic example of an action provided in the retrospectives.

Action x	
Deadline	01.01.2015
Action description	Always add a week to the iterations during holidays.
Comments	Resources are not reliable during holidays.
Responsible unit	Team X
User responsible for action	John Smith
Status	Completed or In Process
Completed	31.01.2014
Type	Preventive or Corrective

Analysis Method

To retrieve any research-worthy knowledge from the actions given by the retrospective reports we needed means compare them. We settled on tabulations as our analysis method for the actions. Tabulations provide easy means of rendering data comprehensible. Krippendorff [18] describes tabulations as:

Tabulation refers to collecting same or similar recording units in

categories and presenting counts of how many instances are found in each. Tabulations produce tables of absolute frequencies, such as the number of words in each category occurring in a body of text, or of relative frequencies, such as percentages expressed relative to the sample size, proportions of a total, or probabilities.

For our case we are going to use absolute frequencies to count the occurrences of different categories. Using relative frequencies would not suffice in our case where determining whether an action is twenty percent technical and eighty percent procedural or thirty percent technical and seventy percent procedural would be immensely difficult and not to mention impractical. Rather an action could be neither technical or procedural, be one of them, or be both. Thus resulting in us using absolute frequencies when we are counting occurrences of the different categories.

To determine what the categories should be we conducted a pilot analysis that can be read about in Appendix B. The final result of categories were agreed upon by the team and are shown in Table 2.4. It consisted of six main themes: Nature, Context, Decision Making, Organizational Learning, Development and Collaboration. Nature, Context, Decision Making, Development and Collaboration can be related to the characteristics and output of retrospective practices. The organizational learning theme relates directly to the learning output and the team organizational learning of the practice. Each of the six themes had several categories which an action would be put in. These categories are shown in Table 2.4 and are further described and defined in Appendix C.

In addition using tabulations for analyzing the actions, we investigated recurring issues. A retrospective is intended to highlight issues and potential actions that can be taken to correct or improve the issues. However it is not necessarily the case that every action is implemented as intended, or in the time frame originally intended. As such issues might come up again and again. Being able to identify these long term issues and address them before they become reoccurring is a clear potential improvement for a team or an organization. Therefore it is of great interest to identify these issues in our analysis. We will be looking through the available information and noting if an issue arises multiple times over time. Also of interest are unresolved issues, which are issues that are raised and a corrective action is agreed upon, but the action is either not implemented or implemented fully.

Processing Steps To perform our content analysis we first defined a set of processing steps that both authors were to follow. All the processing steps can be seen in Table 2.5.

The first step was to specify all the tabulation categories that were going to be included in the content analysis. Once all the categories were found and specified we documented them to ensure both authors had the same understanding of what each category meant.

When the categories had been documented we sent them to the case participants to get feedback on the analysis measurements and verify that it was not anything that we had overlooked that the participants would like answers to.

Once the feedback was incorporated into the content analysis we created a spreadsheet containing all the actions along the top row and all the categories along the first column. We created one spreadsheet for each author and one extra that would later be used for comparison.

When the spreadsheets were done the next step was analyzing all the actions. Each author, separately, read through all the retrospectives and for each action marked which categories it belonged to. Each action could belong to several categories for an example an action could belong to both Testing and Documentation. For each of the six themes at least one mark had to be put down, and the maximum marks would be the number of categories in that theme.

After all the retrospectives were read through and every action was tabulated the authors compared their results. For each action the authors would compare all the marks set for that action and if they both agreed upon all the marks the action would be copied to a new spreadsheet. If the authors disagreed they would go back to the action read it once more and try to find a classification that both could agree to. If this turned impossible we would seek advice from external researchers to gain a suitable solution.

Once all the actions were compared and all the actions were in the new spreadsheet the content analysis was finished and data could then later be extracted from the spreadsheets.

Content Analysis Limitations

In this subsection we will describe the challenges and limitations of our content analysis from the 77 retrospectives. Mainly there were two challenges that occurred during our content analysis, missing contexts and borderline actions. Each is described in detail below.

Missing Contexts As the retrospectives we were given mostly contained actions that resulted from the retrospective meetings, context of how the actions were brought to light could sometimes be missing. In those cases the actions either could have come from a problem happening during the

Table 2.4: The final set of themes and categories that the content analysis is based upon.

Theme	Category	Short Description
Nature	Positive	If an action is a result of a continuation of a process
	Negative	Action is a result of a arisen problem
	Undefined	The nature of the action is unclear or undefined
Context	Technical	If the action is related to some technical context.
	Process	Action is related to a process context
	Undefined	The action could not be related to either technical or process
Decision Making	Strategic	Action is suggesting long-term change
	Tactical	The action is related to identification and use of resources.
	Operational	If the action is ensuring effectiveness and day-to-day operations
	Undefined	If the action is unclear and doesn't fit any of the other decision making categories.
Organizational learning	Single-loop	If the action do a change that only influence the effects
	Double-loop	If one understand the factors that influence effects, and the nature of this influence.
	Undefined	If the action unclear in terms of organizational learning
Development Phase	Development	Action is related to the development phase
	Testing	The action is related to testing
	Documentation	Action is related to Documentation
	Builds	The action is related to building of software systems
	Release	The action is related to releasing of software
	Business	The action is related to business development
	Undefined	The action is not related to any of the development phases described above
Collaboration	Communication	Related to communication within a team
	Leadership	Action is related to leadership
	Competence	Action is a result of lacking knowledge or experience
	External relations	The action is related to customer relations or other external stakeholders
	Planning	The action is a result of bad or good planning
	Undefined	Action is not related to any of the collaboration issues

Table 2.5: Description of the content analysis steps.

Step	Description
1	Specify and document all the tabulation categories.
2	Send analysis documentation to the case participants to get feedback on the analysis measurements.
3	Create a spreadsheet for each author containing data on all the categories and all the actions.
4	Each author read through all the retrospectives separately, setting marks for each category that an action fits in.
5	When both authors were finished reading through all the retrospective they compared the results and put everything into a combined spreadsheet. If the authors disagreed on their markings a short discussion would commence and one would try to find an agreement. If an agreement could not be found the authors would turn to other external researchers to help identify the correct solution.
6	The content analysis was finished when all the actions were in the combined spreadsheet.

iteration, a new idea that was put forth during the iteration or something similar. As some of the actions lacked context we included the undefined category to each theme so that in the cases were we could not tell whether an action was in a concrete development phase, or another theme, we could put it in undefined.

Borderline Actions Borderline actions were the second challenge while performing our content analysis. An example is the action:

To improve the understanding of the processes, the workflow diagrams and the other relevant documentation should be copied to the WIKI.

This action could either be understood as “The workflow diagrams and other relevant documentation is so good that we should put it on the WIKI” or “People do not have enough understanding of the processes and we should give them more documentation”. The first option could be classified as a positive action as the working documentation is good and should be used more, while the second option would be classified as a negative action as it is a problem that the understanding of the processes is not good enough. This action would be classified as undefined in our analysis as it both requires more context and as it lacks the context can be a borderline action between positive and negative.

To deal with the borderline actions we added the steps of each author doing a separate analysis before comparing with the other author to identify the borderlines and find a adequate classification for them.

2.3.2 Feedback Sessions

As the final part of the first case-study several feedback sessions were held, to gather valuable input from the team and their reflections on the analysis. Each of these feedback session will be described in the following subsections.

Feedback Session: Team Discussion

For the first feedback session we visited the team in person and held a presentation followed by a group discussion. The presentation was held in a meeting room in front of the team. The concept of the analysis was first explained and discussed with the team members, in order to ensure that there existed a common understanding of the work being presented. The presentation was also simultaneously a dialog between the presenters and the team. This was facilitated by constantly asking questions and encouraging discussion. One of

the ways of encouraging discussion was to ask the team “What expectations do you have for this category?” before showing the results of the analysis. This ensured that we got their speculation as well as their reaction to the analysis.

Feedback Session: SCRUM Master Interview

The second feedback session we had was an interview with the SCRUM master of the team. We spent a short while preparing and reviewing the questions we had prepared. The interview would be held in a semi-structured manner with the goal of having an open discussion of the results of the team discussion as well as gaining insight on the themes of the analysis. During the interview notes were taken, as well as recorded, and the SCRUM master was encouraged to speak his mind on relevant subjects.

Feedback Session: Team Leader

The third feedback session was an interview where we presented additional findings, based on the results from the former feedback sessions and discussed the impact on the team. The interview was conducted with voice chat with the leader of the team. The session also served as a brainstorming session for the team leader in order to prepare a retrospective evaluation session he would lead the following week. This evaluation session would be a discussion between all members of the team where they discussed the current state of the retrospective after the feedback session and analysis.

Feedback Session: Internal Team Evaluation

The last session was conducted by the team internally without the authors present. The team had a “Meta-Retrospective” where they discussed the current state of the retrospective sessions, as well as their thoughts on the first feedback session. The team held the session as a normal retrospective, but with a semi-structured agenda that the team leader had developed in cooperation with the authors. The team leader then sent the meeting log to us, and had a short interview explaining the content.

2.4 Breadth Study

Our second case-study aimed to get a broader picture of the practices used for retrospectives today. We decided on a semi-structured interview, in order

Table 2.6: Interview guide overview

Step	Description
General overview	Questions relating to the holding of the retrospective and learning
Team dynamics	Questions on how team dynamics are handled and how they are experienced in the team
Organization learning	Questions on how the team approach learning
Anything else	Summary questions intended to cover potentially missed topics

to be free to improvise as well as being prepared to ask stimulating questions to the interview subject.

2.4.1 Semi-structured interviews

We decided to use a semi-structured interview as our model for the interviews. An unstructured, or semi-structured does not have a complete script, and leaves room for improvisation, unlike a structured interview where there is a complete script. [24] This approach was chosen to leave the interview subject free the elaborate on subjects they found interesting.

A list of questions was prepared in order to serve as a guideline for the interview. This interview guide was divided into four sections, relevant for our research, being: “General overview”, “Team dynamics”, “Organizational learning” and “Anything else”. A short description of each of these are shown in Table 2.6. The complete interview guide including all the questions can be found in appendix A. These four sections were chosen due to experiences working with team Zulu, as well as through a discussion with our supervisor.

In order to conduct a successful semi-structured interview Myers and Newman mentions several factors that need to be acknowledged [24]. Among these are the lack of trust, elite bias and ambiguity of language. In order to avoid these potential pitfalls we spent time ensuring a mutual understanding and respect with the interview subject, for example by explaining our intentions and maintaining a friendly tone during the interview.

The interviews were if possible held in person or over voice chat. This was done to lower the barrier for dialog as much as possible and let the interviewee speak as easily as possible. However at one point one interviewee had to resort to mail due to time constraints, we decided to keep the interview and the answers are included in the appendix.

2.4.2 Interview Subjects

In order to select interview subjects within the industry we contacted potential practitioners through e-mail or phone. As well as consulting our advisors. We wanted subjects with experience both holding and participating in retrospectives in order to gain multiple views of the retrospective experience. The base criteria was that participants had to be a part of a team that conducts retrospectives on a regular basis. We wanted a varied spread for the interview subjects and selected subjects based on their project and team compositions. Consultants, distributed teams, product development teams, small teams and big teams were some of the considerations for teams we wanted for this study.

2.5 Analysis Method

Our analysis method consisted of several steps. After gathering all the data from content analysis, feedback sessions and interviews each researcher went through the material separately. For the feedback sessions and interviews we listened through the recordings taking notes ensuring nothing was missed during the sessions. After all the material was worked through we compared the different data against each other and against earlier academic work. The academic fields used is: Retrospective literature and organizational learning. The fields was chosen as retrospectives is a shared learning activity and thus organizational learning is useful to see how the team as a whole learns and retrospective literature as it can be used to compare the characteristics for retrospective practices.

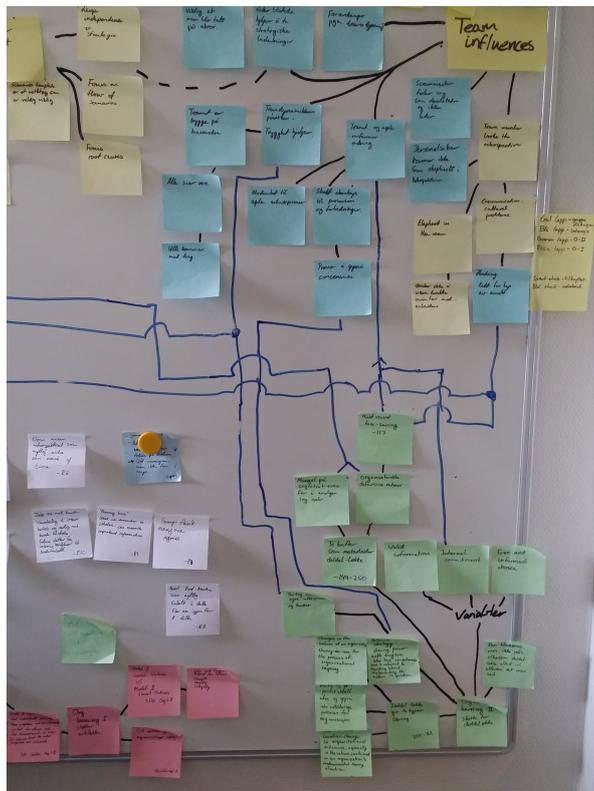


Figure 2.1: A picture of our whiteboard during analysis

2.6 Participants

Our research bases itself on the primary data gathering methods: Retrospective depth analysis of one team and interview input from other teams. In this section we will present the participants using the phonetic alphabet as identifiers. Many of the participants were recruited from existing research projects being “Agile 2.0” and “SMIGLO”.

2.6.1 Depth Analysis Participant - Team Zulu

Team Zulu is situated in Norway and develop a human resource system. The team has been working with agile principles for over 10 years. An overview of the team can be seen in subsection 2.6.1

1. 1 HoD – Product Owner
2. 1 SCRUM Master
3. 1 Architect
4. 4 Developers
5. 1 Content Responsible
6. 2 Testers
7. 1 Build and DB responsible
8. 1 Third line support engineer

2.6.2 Breadth Study Participants

In this section it is a short description of each interview subject. All participants in the interview research were familiar with retrospectives, and agile development practices. The selection process is described in subsection 2.4.2 and an overview of the team and representatives can be found in Table 2.7

Team Alfa

Team Alfa consisted of 22 members and we spoke to two of them. The project leader and one developer. The team consisted of designers, testers, front-end developers and back-end developers who all worked with media solutions for a customer.

Team Bravo

Team Bravo consisted of 6 members, 2 in Norway, 3 developers in Poland and 1 tester in Shanghai. The project leader and Scrum master were in Norway.

Team Charlie

From team Charlie we spoke to one representative. The participant was project coordinator, project leader and SCRUM master for a development team consisting of four persons. The participant also had responsibility for agile practices in his company which develops HR solutions.

Team Delta

The representative for Team Delta was a developer and the teams SCRUM master. The team developed HR solutions.

Team Echo

The participant from team Echo was a consultant acting as a developer and semi-project leader for a multi-platform development project. The team consisted of 10 members and developed four different products for one customer.

Team Foxtrot

Team Foxtrot consisted of about 20 members in the roles of design, ux, front-end, back-end and project leaders. The representative from team Foxtrot was a system developer and he chose to respond to our interview through email. We acknowledge that this might have resulted in input getting lost. However the input that has been gathered will still be able to give us some data which can be used for this study.

Team Golf

Team Golf consisted of 2 members working in Norway, with the rest of the team situated in Shanghai. The representative from Team Golf was the SCRUM master for the team.

Table 2.7: Table overview over interview participants

Team	Description	eRrepresentatives
Alfa	Project leader Developer	2
Bravo	SCRUM Master	1
Charlie	SCRUM master, project leader and coordinator	1
Delta	Developer and SCRUM master	1
Echo	Developer and project leader	1
Foxtrot	System Developer	1
Golf	SCRUM Master	1

Chapter 3

Results

In this section we will report the results of our research. We will describe our results from the depth and breadth study, and the chapter is divided into two sections. The first section is the results that relate to the characteristics of the retrospective. This includes the output, the processes used in practice and the impediments that face the retrospective. The second section is all the results related to organizational learning.

3.1 Retrospective Characteristics

In this section we will first describe some key characteristics found during our depth study. We will then continue describing the output, processes and impediments found in both of our studies.

3.1.1 Key Characteristics

From our depth study of team Zulu we found some key numbers and these numbers can also be seen in Table 3.1. The retrospective reports spanned over a period of five years from August 2009 to November 2014. This amounts to 278 weeks and we are going to refer to week numbers from the first retrospective for the remainder of this report. During the 278 weeks 77 retrospectives were held and within these 343 actions were created, where 65 of these actions were still unresolved. This yields an average of 4.45 actions per retrospective and 0.84 unresolved per retrospective. This is equal to 1.23 actions per week, where one has 0.23 unresolved actions per week.

In Figure 3.1 we can see the development of the numbers of actions. We can see that the team has had a pretty steady amount of actions with no abnormal spikes or changes until week 180. The total amount of actions

Table 3.1: Some key numbers from the retrospectives

Key-value	Value
Retrospective report period	278 Weeks
Number of total actions	343
Number of unresolved actions	65
Average actions per week	1.23
Average unresolved action per week	0.23

follow the average expected actions quite close and reveals the steady number of actions. At week 180 a slight increase in the number of actions begin. This lasts until until 237 when the amount of actions per retrospective starts to even out. The amount of active (or unresolved) actions has a steady amount of total actions increasing at about the same rate as the total number of actions.

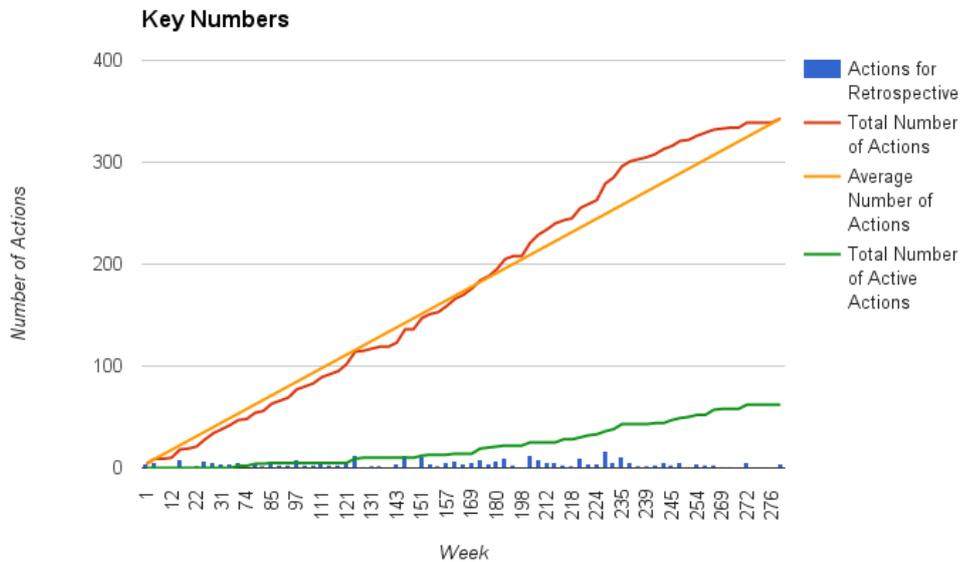


Figure 3.1: A visual representation of some of the key numbers.

When presenting the key-numbers, during the first feedback session, team Zulu mostly found our results agreeable with their own thoughts. The only surprise to the team was the amount of active actions. Their surprise came as they believed the number of active actions to be higher. The reason for

this belief was that they thought they were worse at closing actions as the list of active actions seemed so long, but compared to the amount of total actions it seemed more reasonable. However it was pointed out that existing actions hinder new actions relating to the same problem to be created and thus the amount of active actions might not be accurate with how the team works with the problems, as an action might not be documented, but worked on never the less.

When we raised the question on why the decrease of actions after week 237 the team gave their thoughts. During this period, they had acquired a new foreign developer within the team. Unfortunately the new developer had not lived up to the task, creating what the team called an “Elephant in the room”. We will discuss this further in section 3.1.4 as this is regarded as an impediment.

3.1.2 Output

The output of the retrospective practice can be divided into several themes. We divide it into topics discussed, improvements and enthusiasm. The results in “Topics Discussed” describes our findings on which topics are discussed during the retrospective. The “Improvements” results describe how decisions are made and how the teams implement them. “Enthusiasm” describes how enthusiasm inflicts the retrospective practice.

Topics Discussed

From both our case-studies we got an indication of what issues were brought to the retrospective. From our breadth study we learned which areas topics originated from. From our depth study we found which topics from our categories, described in section 2.3.1, were mostly influencing the retrospective. Below we will first describe the general results and then follow with the results from the depth study describing each category-set and some of the trends we identified.

General In general we found three main areas discussed in the teams participating in our breadth study. The three areas were work environment, process improvement and technical issues.

Work environment issues were described mainly by team Echo. They used the retrospective to improve or fix their work environment discussing issues such as noise in the working areas and bad wifi connection. We also saw some similar issues in team Zulu, however it was not their main focus.

Process improvement was the second area we found. This was the main focus of almost all of the teams. Team Alfa, Charlie, Delta, Golf and Zulu had this as their main focus. Team Bravo had it as a main focus along with technical and team Echo discussed such issues, but rarely. Process improvement issues were regarded by most of the teams as the most valuable output from the retrospectives. Issues that were discussed included how the team communicated and how development should be done in a process perspective.

The third area discussed during the retrospective was technical issues. Team Bravo had this as their second focus along with process improvement. It occurred in the other teams as well. In some teams, Alfa and Delta the issues were actively censored during the retrospective as they wanted to have a focus only on process improvement. Technical issues were usually related to the product and ways to either fix faults or improve the quality.

Table 3.2: Discussion Areas

<i>Discussion Area</i>	<i>Team Focus</i>
Work Environment	Team Echo
Process Improvement	Team Alfa, Team Bravo, Team Charlie, Team Delta, Team Golf, Team Zulu
Technical	Team Bravo

Nature The retrospective analysis of team Zulu revealed that most actions are created as a result of negative problems that has occurred during the development. 89.3% of the actions were negative, while 5.5% of the actions were positive and acknowledged good working practices that would be continued. 5.2% of the actions we lacked the context to determine whether they were positive or negative. As for the distribution of the actions over each retrospective there were no abnormalities except week 97 where there was an unusual amount of positive actions. However while looking into this week we found nothing in particular that could be identified as cause for this spike. As can be seen in Table 3.3 the classification of the active actions pretty much mirrored the results from the total actions.

The results from the nature context did not surprise the team at all. We asked if they would like to guess what the ratio between positive and negative would be and they said one-to-nine, which is quite close. Our results revealed five percent undefined and five percent positive with the remain-

Table 3.3: Analysis results from the content analysis for the nature of the action.

Category	All Actions		Active Actions	
	Number	Percentage	Number	Percentage
Positive	19	5.5%	1	1.6%
Negative	310	89.3%	57	90.5%
Undefined	18	5.2%	5	7.9%

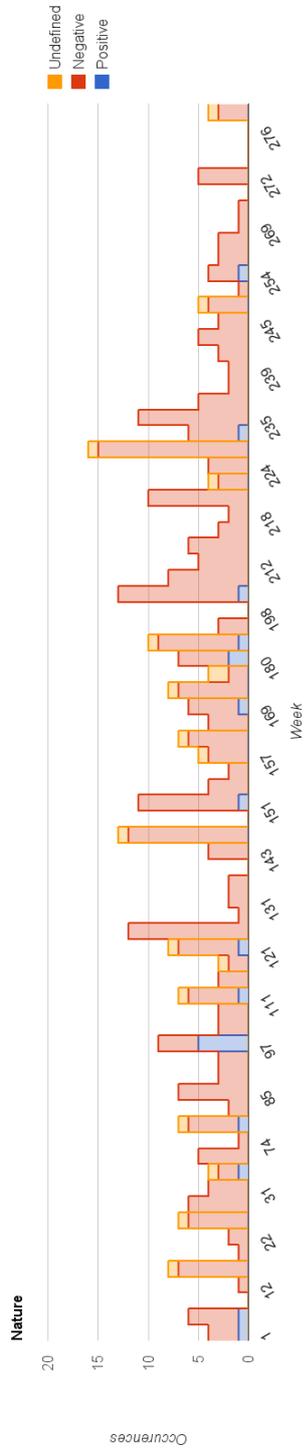


Figure 3.2: The distribution of negative, positive and undefined actions across the timespan

ing 90 percent negative. The team described themselves as problem-oriented and reasoned that this was the cause for the high amount of negative actions. They also said that they do, during the retrospectives, talk about good things that have happened, but as they are problem oriented it rarely gets documented.

After reflection of the results through an internal retrospective the team decided to do a concerted effort to increase the positive attitude and number of positive issues during a retrospective meeting. One step taken was an informal attitude where team members tried to focus on positive events, and bringing them up during the retrospective meeting. Another step was the decision to add a numerator to track positives. Lastly they decided to add a “positive/negative” value to each action point. This in combination with the plan to implement a dashboard for data tracking let the team plot the positive/negative actions over time. During the fourth feedback session the team leader presented that they had already felt changes in the atmosphere in the retrospective meeting and described it as “less of a funeral”.

Context For the context of the actions analyzed, in the depth study, the majority were process related ones. The process actions numbered in 228, which is equal to 58.6% of all the actions. The Technical ones numbered as 157 which is 40.4%, while only 4 actions were undefined which results in 1% of the total actions. As for the distribution over the timespan analyzed there were no abnormalities as can be seen in Figure 3.3. For the active actions the results become more equal as seen in Table 3.4. However it is worth mentioning that the active actions are a sub-group of the total and thus this result is probably a skewed grouping.

Table 3.4: Analysis results from the content analysis for the context of the action.

Category	All Actions		Active Actions	
	Number	Percentage	Number	Percentage
Technical	157	40.4%	37	52.1%
Process	228	58.6%	34	47.9%
Undefined	4	1%	0	0%

Development Phase Planning, testing, development, and documentation were the four dominant phases in which an action was related according to our content analysis of the retrospective reports. As can be seen from Table 3.5.

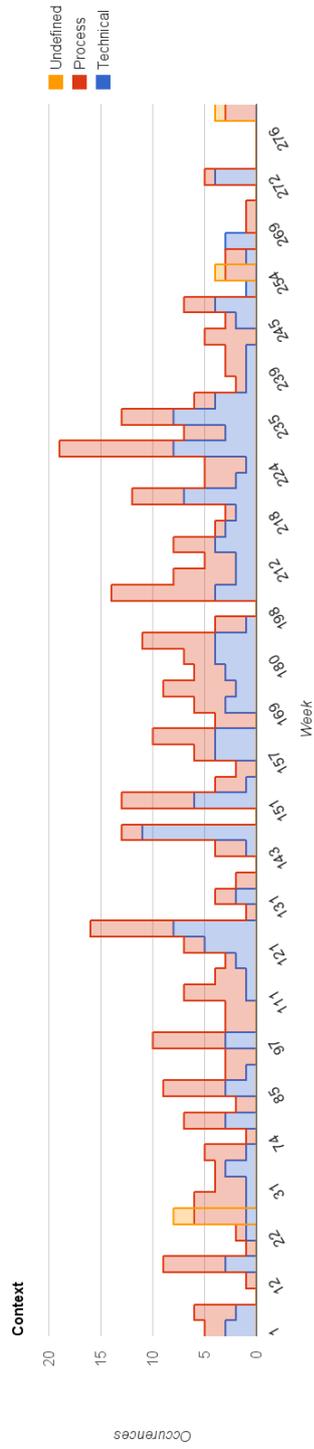


Figure 3.3: The distribution of technical, process and undefined related actions across the timespan.

Planning being the biggest has a distribution value at 24.6%. Second is the testing which 21.1% of all the actions are related to. Development is related to 18.4% and documentation is 13.2%. Finally we have the remaining five categories Release, Build, Business Development, Bugfix and undefined which varies between 3.7-6.4 percent as can be seen in Table 3.5. For the distribution of the different categories over time all of the categories are evenly distributed, in other words; No category is clustered to a specific period in time, but rather occurs evenly through the whole timespan. This can be seen in Figure 3.4. As have been the cases with the other themes the sub-group of the active actions mirrors the total actions with only minor variances as can be seen in Table 3.5.

Table 3.5: Results from the content analysis in which development phase the action regards.

Category	All Actions		Active Actions	
	Number	Percentage	Number	Percentage
Development	89	18.4%	11	13.1%
Testing	102	21.1%	18	21.4%
Documentation	64	13.2%	16	19%
Release	18	3.7%	4	4.8%
Build	23	4.8%	6	7.1%
Business Development	18	3.7%	5	6%
Planning	119	24.6%	19	22.6%
Bugfix	20	4.1%	2	2.4%
Undefined	31	6.4%	3	3.6%

The results from the retrospective analysis were quite different from the team expectation in the context of which development phase the actions were related. The team expected that build and documentation would be the two biggest groups. The content analysis revealed that planning was the biggest followed by testing, development and documentation. The team thought this made sense as most of the planning is directed towards process improvement and they do focus on process during the retrospective. The team speculated if the releases might have a correlation with planning actions coming right after. It was also again mentioned that actions might not be created even though they were discussed as actions already existed and that they had not been completed.

The team expected build to be the phase that occurred most. It was a surprise that it was not better represented within the actions as it felt that it

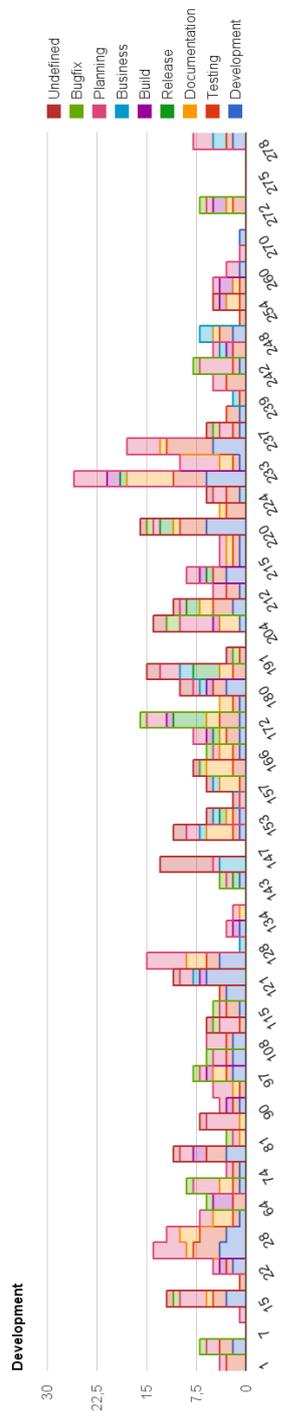


Figure 3.4: Timeline showing the distribution of the different development phases over time.

was discussed often in retrospectives. One of the members said that the lack of actions specific for builds might be the reason for why it was a problem. Even though it was discussed during the retrospectives.

We asked the team if there were any categories they missed from the development phase analysis and two were mentioned. The first one was hotfix which the team often mentioned during the feedback session. The second one was flow as to how the development of the product progressed. For future similar analyses this should be taken into consideration.

After the first feedback session with team Zulu we took the time to create a chart showing releases against the number of actions and planning actions as displayed in Figure 3.5. These results provided interesting data. As can be seen in Figure 3.5 the releases does not have any impact on either the amount of planning actions or number of actions in total. This is interesting as the team themselves expected there to be a correlation. During the third feedback session the team leader also found this interesting. However he did not have any arguments for why this could be the case.

Collaboration The depth study of team Zulu showed that 45.3% of the actions were undefinable in terms of the collaboration and the categories we had created for it. From the actions that were definable Communication was the biggest with 35.2%. The second was external relations at 11.5% and third competence at 6.9%. Finally leadership was the smallest at 1.1% of the total actions. The statistics can be seen in Table 3.6. For the distribution of the different categories over time, Figure 3.6, most of the categories was evenly spread across the whole timespan. the only exception to this is week 146 where there is a clear spike of external relations. This spike was a result of the team attending a networking meeting in which they did a retrospective to better prepare them for the next networking meeting. This anomaly will be disregarded further in the report. The active actions shows that the three categories communication, competence and external relations evens out while leadership and undefined remains nearly the same with only some small variances. However it is worth mentioning that the active actions are a sub-group of the total and thus this result is probably a skewed grouping.

During the first feedback session the team expected communication to be the category with the most actions. If you disregard the undefined actions which was the biggest in terms of collaboration the team's expectation was correct. We asked if they could explain what kind of communication that were discussed most during the retrospectives. They said that communication between the different stages of the development and more oral communication rather than written were the kind of communications that

Table 3.6: Results from the content analysis regarding the collaboration influences of an action.

Category	All Actions		Active Actions	
	Number	Percentage	Number	Percentage
Communication	128	35.2%	14	20.9%
Leadership	4	1.1%	1	1.5%
Competence	25	6.9%	1	11.9%
External relations	42	11.5%	11	16.4%
Undefined	185	45.3%	33	48.3%

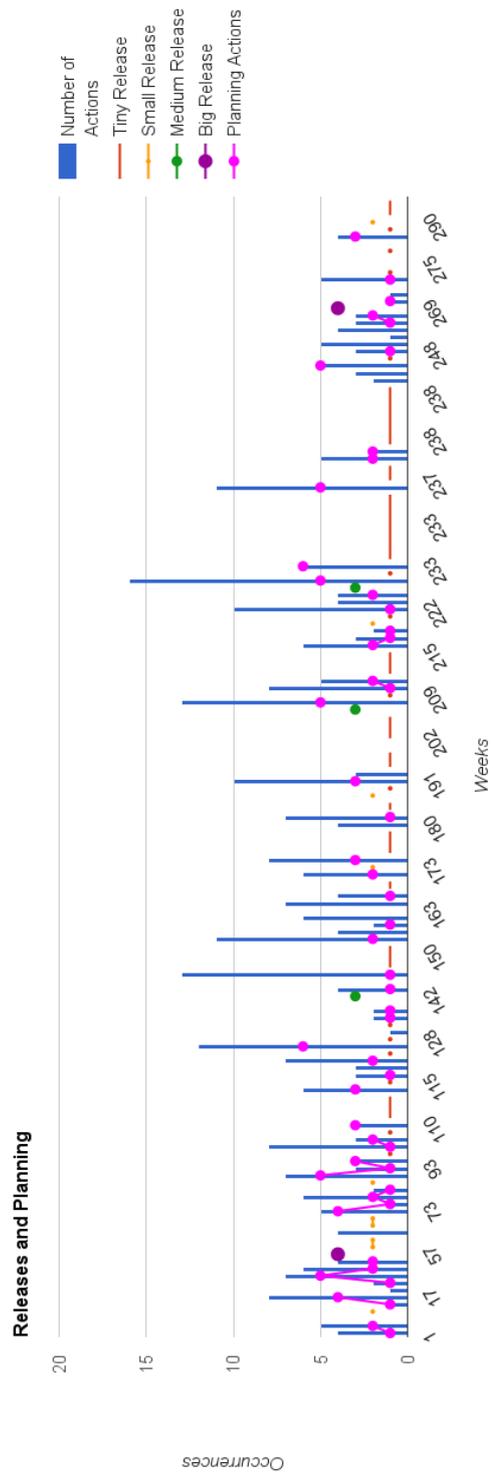


Figure 3.5: Correlation between release and planning actions

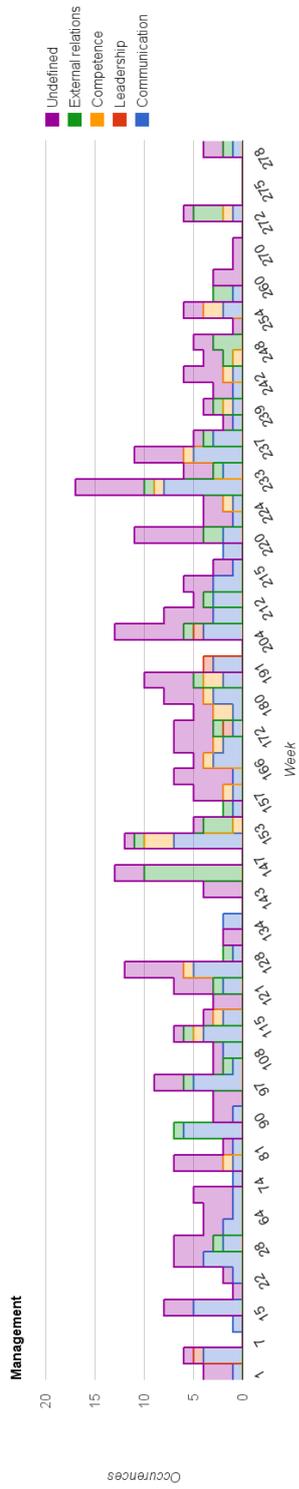


Figure 3.6: Timeline showing the distribution of the different collaboration categories over time.

were discussed the most. As for the other categories the team had no special feedback.

Trends While conducting our retrospective analysis of team Zulu, we uncovered some trends. By trend we mean actions that are related to the same issue/theme. The trends were identified by comparing actions towards each other noting which actions were similar to each other. Also noted were actions that still were not implemented. Interestingly the three trends we found turned out to have very few unimplemented actions. Which can indicate that the team has a focus on trends. We identified three trends. Bugfix, Scenario Template and Developer-Tester Communication. We'll go through each of these in the following sub-sections.

Bugfix The first trend we recognized performing our content analysis was bugfixing. Developing computer systems is sure to create bugs and fixing them then becomes a natural part of developing software. In total we found 20 actions that were related to bugfixing. Of these, two were purely technical actions, five were technical and process related and the remaining 13 action were purely process related. Of the 20 actions nine were related to communication between team members. In Figure 3.7 one can see that the total amount of bugfixing actions have increased steadily throughout the timespan.

When we presented the bugfix trend to the team, during the first feedback session, they were surprised that we had identified it as a trend. The surprise came as a result that they had found a way to work with the bugs. Having dedicated bug-days was a practice that had been implemented and used for the last year which the team found very helpful. As can be seen from Figure 3.7 the amount of actions still continue to increase after the implementation of the bug-day. When inquired by this the team told us that even though the bug-day worked very well, they still wanted to improve and that was why the actions still were increasing.

Scenario Template The second trend we discovered were in relation to a worktool called scenario template that team used to help specify requirements, create user stories and etc. In total we found 25 actions that were related to the scenario template. Of these 25 actions four of the actions were technical and process related, six were purely technical adjustments of the tool and 16 of the actions were process oriented on how the scenario template should be used. 18 of the actions were single-loop, only changing the effects which the scenario template provided. There were also six double-loop ac-

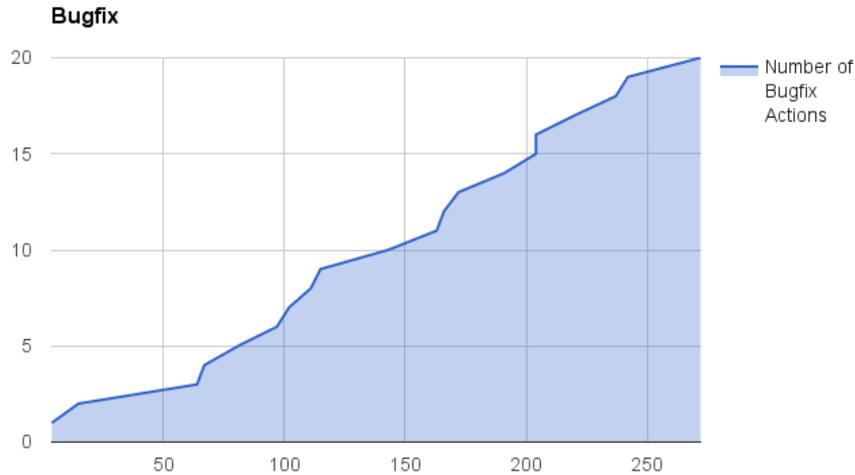


Figure 3.7: The total amount of bugfix actions over time.

tions acknowledging root-cause issues with using the scenario-template and changes to reflect them. In Figure 3.8 the total amount of scenario template actions are shown over the 272 week long timespan. One can see that until week 163 the team has a slow increase in the number of scenario template actions. After week 163 however, a huge increase in number of scenario template actions occur. This continues until week 235, with a little slow period between week 180 and 205. At week 235 the team planned a meeting to go through the complete scenario template and after week 235 there are no more actions related to the scenario template.

The trend of scenario template changes provided surprises both for the team and the researchers during the first feedback session. As described above the retrospective reports indicated that during week 235 a meeting were scheduled to go through the whole template. After this action no more actions occurred during the timespan. The researchers expected it to be a classic case of taking the root-cause, double-loop learning and the problem disappeared. However the scheduled meeting was never held. The team discussed several reasons for why this could be the case. One were that they still might be waiting to hold that meeting. Another was that a cleanup had been done, and changes to the scenario template still occurred, but that actions on it were not created. A third possibility was that the same period they had a team member that didn't work out as described in section 3.1.4.

Unfortunately the discussion ended only with possibilities and no conclusion.

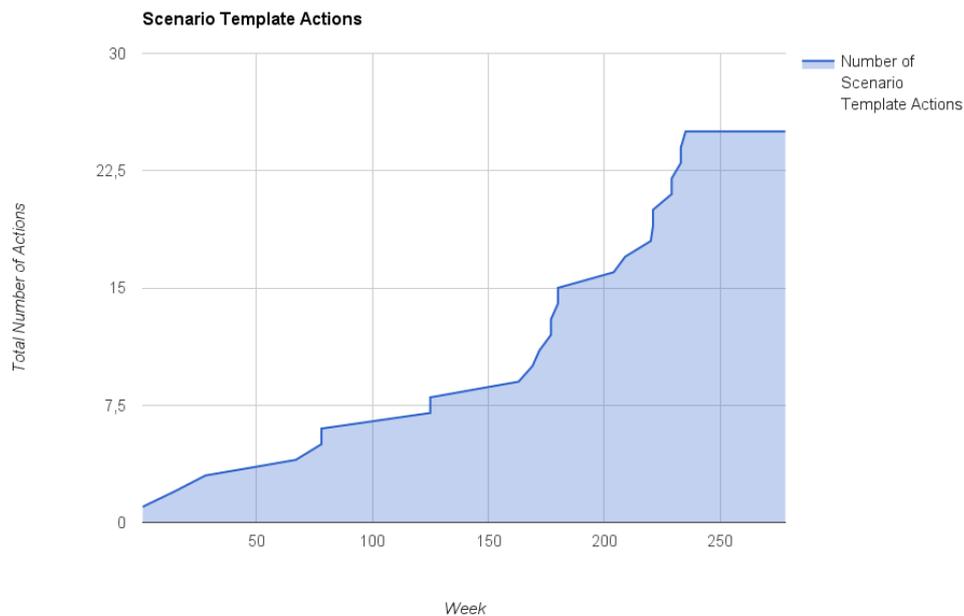


Figure 3.8: The total amount of scenario template actions over time.

Developer-Tester Communication The final trend we observed during our content analysis was the communication between developers and testers. In total 25 actions were related to this. Of these 24 were process oriented and all the actions occurred from issues with a negative nature. Figure 3.9 shows the distribution of the 25 actions over the 272 week long timespan. It can be seen that for the first 209 weeks the amount of actions increase slowly with only two to seven actions every 50th week. After week 209 however we see a dramatic increase in the amount of actions, before it completely stops in week 238. We were not able to find any possible reasons for this sudden stop from reading through the reports. However as can be seen from Figure 3.9 there has been periods between actions as long as 45 weeks so it is possible that this stop can be such a break.

The final trend, developer and tester communication returned similar feedback as the scenario template trend. The pattern identified in the content analysis where after week 238 the actions related to developer tester

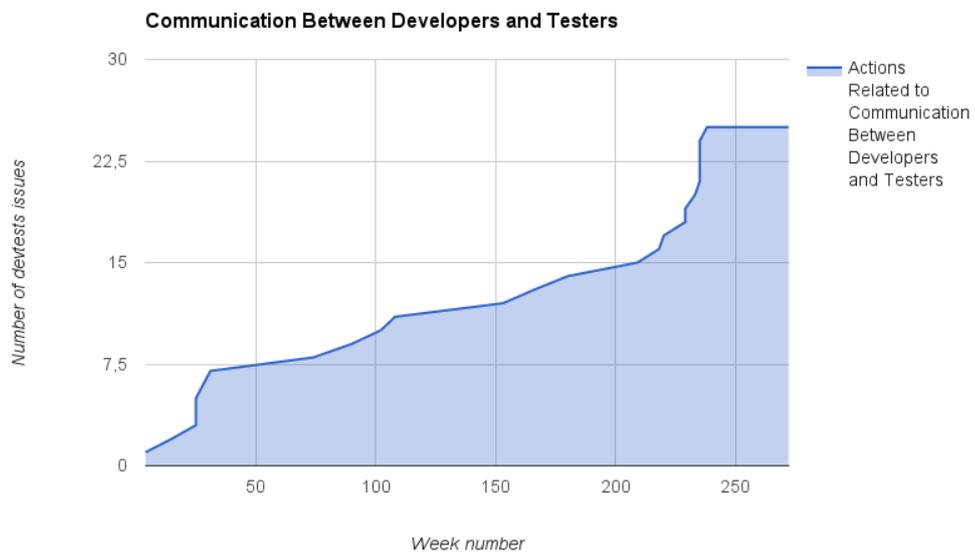


Figure 3.9: The total amount of developer-tester communication related actions shown over time.

communication stopped. The team again found several possible solutions. The first one was, as described in section 3.1.4, the team member which did not fit the team. Another possible reason was secretary changes as well as scrum master's leave of absence. The team also said that most of the communication had been mostly oral and handover meetings had been introduced and that this worked pretty well. The team seemed to believe that all the mentioned reasons would explain the stop of developer-tester communication actions.

Improvements

Which improvements are returned by the retrospective practice and how they are implemented were some of research questions we asked. From both of our case-studies we got some results relating to this. Our depth study gave us an insight into which kind of decisions are made during the retrospective. From both studies we got insight into how improvements were implemented. From the fourth feedback session with team Zulu we also learned that the team wished to develop a tool help them support implementation. We will describe each these results below.

Decision Making in Team Zulu From the depth study of team Zulu we got insights into what kind of decisions were made during the retrospective. From the retrospective analysis the decision making results showed that the operational decisions occurred most in the actions as can be seen in Table 3.7. Operational decisions occurred in 53.2% of the actions, while tactical was at 25.9% and strategic was at 16.1% of the actions. There were only four cases where we were not able to determine which kinds of decision making type it belonged to. For the distribution over time, as shown in Figure 3.10, there was no emerging patterns and all the decision making types were evenly distributed. The active actions mirrored the total actions almost equal as can be seen in Table 3.7.

The team thought that it was a good sign that the team was able to do strategic decision making. When we presented the results for the decision making analysis, the team was pleased to see that they had all of the three categories represented. They were especially pleased to see that there was a substantial portion of strategic actions. We asked what could be the reason for this and the team responded that they felt autonomous. As they described it, the company allowed the different development teams to have a fairly large amount of independence allowing room for strategic choices within the team.

Table 3.7: Analysis results from the content analysis for the decision making perspective of the action.

Category	All Actions		Active Actions	
	Number	Percentage	Number	Percentage
Strategic	55	16%	10	16.1%
Tactical	89	25.9%	18	29%
Operational	195	56.9%	33	53.2%
Undefined	4	1.2%	1	1.6%

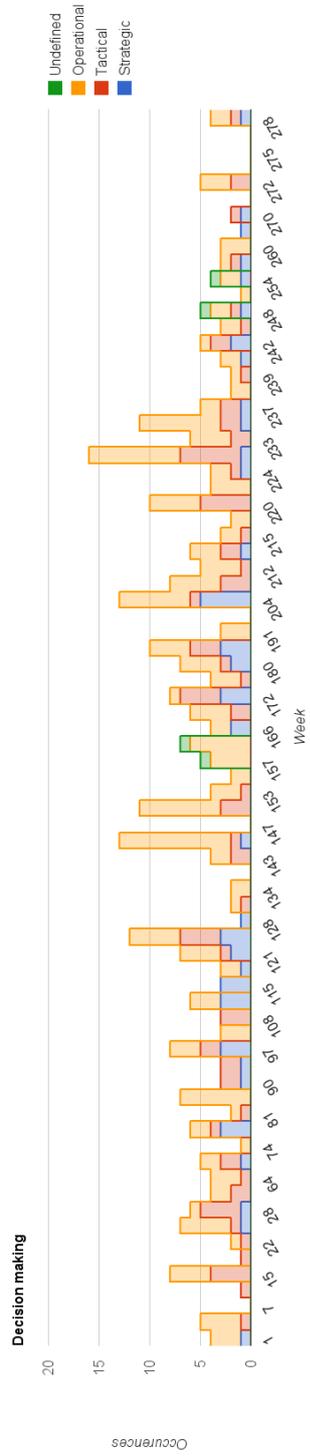


Figure 3.10: A timeline showing the distribution of the different decision making decisions for all the actions.

Implementation The steps taken to implement actions created during the retrospective and whether unresolved actions were a problem varied between the interviewed teams.

Team Echo used several techniques to enforce actions. Some actions were added to the sprint backlog as it helped reminding the team that they needed to be done. The actions that were related to the work environment was usually handled by the SCRUM master. Reaching a common consensus was something that the team regarded as a beneficial way of getting things done. Finally the team, along with other interviewed teams, always assigned a name to the actions found during the retrospective. This ensured that a person would have a responsibility to the action and at every retrospective, the team would take a quick round to see if the action had been completed. We asked the developer what happened if someone forgot to do it and he replied that it was noticed by the other members of the team and it had never occurred that an action had been unresolved over two retrospectives. This resulted in that most actions became resolved as was the case with other teams that assigned names to the actions.

The SCRUM master of team Zulu described the current action assignment as “push” based where an issue was delegated to a team member, often the one with expertise relevant to the issue. Also mentioned was a desire to introduce a more “pull” based system where tasks could be chosen by team members at their leisure. A major issue was the problem of enforcing process related actions, as there existed no formal tool or process for ensuring that the process actions were actually employed. A lot of the actions were quick fixes that were handled rapidly by whoever were delegated the task.

Team Delta said that most actions were resolved. However new routines that were part of an action that was not working would not be done regardless of names assigned. These new routines however was later deemed bad routines as it did not work when it was practiced.

Team Alfa managed to resolve most of the actions created. They assigned names to the different actions and the project leader said this was more effective rather than not doing it, which they had done earlier. Some actions however the team wasn’t able to resolve. There were several reasons for this. Some of the actions became really creative and thus were difficult to implement. Other actions were huge and required resources the team didn’t have.

Follow-up by the SCRUM master of the team was generally regarded as an important measure to help resolve actions from the retrospective by several teams. Also adding the actions to a SCRUM board or a separate board was regarded as a good measure to get actions resolved.

The SCRUM master of team Charlie admitted that very few of the actions

they created were resolved. He told us that actions were not assigned to individuals, but rather the group as a whole. As part of the leadership for the whole department he had participated in a retrospective for the department as a whole, where individuals had been assigned to different actions. This had not worked in that retrospective and thus he was hesitant to do this in his own team. He hoped that having a white board to put the actions up on would help remind the team to resolve some of the actions.

Table 3.8: Action Follow-Up Techniques Used

Team	Follow-up technique	Satisfied
Echo, Delta, Alfa, Foxtrot, Zulu	Assigning Name to action	Yes
Charlie, Bravo	Assigning to group	No
Foxtrot	Adding to backlog	No
Delta	Visualizes Action	Yes
Echo, Golf	Handled by SCRUM-master	Yes

Major breakthroughs that stemmed from retrospective all came from process related improvements. Team Delta described how one team member had brought up a personal issue, the team member felt he did not get enough help from the other team members. The team was then forced to examine how this had occurred, and how what they could do to prevent it. This issue was resolved. This was described as a “tough retrospective”, but the team was very happy with this issue being raised, and it was regarded as one of the best retrospectives they ever had, because it worked directly on team dynamics. The scrum master for team Delta said:

We were all very pleased with this issue being raised, it was one of the best retrospectives we ever had, because it directly addressed team dynamics, we handled it pretty well, everyone were eager to discover the problems and how they could help.

The team Delta SCRUM master described this issue as a “heavy” issue to raise, especially for the person who was not getting help. The SCRUM master considered this a proof that the team trusted each other, and said that the issue would never have been raised in a team without trust.

Decision to make a dashboard Team Zulu held a retrospective session where the impressions from the analysis and earlier feedback sessions were discussed. The authors of this report were not present, but received the

results through a talk with the team leader and the team SCRUM master, as well as a written report that is further described in section 3.2.3. One of the actions they wished to implement, as a result of the depth study, were to develop a dashboard to help coordinate and focus the implementation of actions. This dashboard would be based on the method and form of the analysis done by the authors. The team decided that the team should at least contain the data seen in section 3.1.2. This list was based on a similar list provided in the written feedback from the team.

1. Number of actions over time.
2. Number of positives over time.
3. Distribution of positive and negative actions over time.
4. Distribution of single- and double-loop actions over time.
5. Open and closed actions over time.

We earlier described how team Delta visualized their actions and felt that it helped implement actions. As the wish to visualize is apparent in both team Zulu and team Charlie suggests that visualization could be a good strategy to help follow-up the implementation of actions.

Enthusiasm

Our case-studies revealed that enthusiasm both inflicted the retrospective and was affected by it. The results revealed that enthusiasm could create both a positive loop and a negative loop. They also revealed some factors that affected the enthusiasm, being oversight and ownership and trust. We will describe each below.

Positive Loop Described by several teams, Alfa, Delta, Echo and Zulu, enthusiasm was able to be a part of a positive loop related to the retrospective. The loop could be described as the following:

If the retrospective produced any implementable actions and those actions were implemented it would produce more enthusiasm for the retrospective practice and therefore increase the chance of future actions actually being implemented.

It was emphasized that change was important for the retrospective practice by all the teams where the subject came up. It would produce enthusiasm and help improve the working practices.

Negative Loop As two sides of a coin enthusiasm could, in addition to create a positive loop, create a negative loop. The SCRUM master of team Charlie told us that unresolved actions could create a negative loop, where enthusiasm for the retrospective went down as improvements never came and the low enthusiasm made sure fewer actions were implemented. And that it could be a challenge.

Oversight From the second feedback session with team Zulu during the depth analysis we learned that oversight over implementation rate of actions had affected the retrospective. The team had not previously appreciated just how much had been accomplished, and that this had led to a lower enthusiasm around the retrospective. When the team was confronted with the completion rate they were surprised and pleased with the amount they had accomplished. In order to continue this more objective oversight the team decided to create a dashboard in order to have better view of the statistics concerning implemented versus unimplemented actions as described in section 3.1.2.

Ownership and Trust The SCRUM master of team Zulu also had some reflection on how the team influenced the enthusiasm. During the second feedback session the SCRUM master spoke about how he considered his team mature and willing to learn. Over time the team had developed an ownership feeling regarding the retrospective, and that this had been developed through working with it over time. Another important element was that a high degree of trust between team members led to productive sessions. An important part of this is that every team member feels that they are taken seriously. This helped create positive enthusiasm for the retrospective practice. All of the other teams we spoke to about the subject confirmed that ownership toward the development process helped create enthusiasm for the retrospective practice.

3.1.3 Processes

In this section we will detail the results we discovered during our studies. The results will be divided into before, during and after the retrospective meeting.

Before the retrospective

This subsection will contain our observations on work done before the retrospective. Specifically the considerations “Preparations”, “External Facilita-

tor” and “Encouraging through external things” will be described.

Preparations When asked about how the team members prepared for a retrospective the team Zulu SCRUM master commented that there was a variety of approaches, some team members made lists beforehand and arrived prepared, while others were more impulsive and decided on their issues during the retrospective.

Of all the teams we talked with no one except one team used any external or specific tools to gather information and prepare the retrospective. The one team that used some other information than what was gathered at the retrospective used lead times as a source of information.

External Facilitator There were different views on facilitating retrospectives. Several teams used an external facilitator and said that they would encourage others to do the same. The benefits was that the external facilitator was able to see things that existed within the team, that the team themselves were not aware of. The external facilitator was not hired as a facilitator, but rather a SCRUM master from another development team.

The use of external facilitators was an interesting concept to team Zulu, and the one experience they had with using one had been a positive experience. When the team Zulu SCRUM master was asked about if he felt like a leader, or if the team viewed him as a leader, the SCRUM master said he felt more like a facilitator, and that he didn’t think the team considered him a leader. Though he was mindful of this possibility. When asked about the inclusion of the project leader he considered as long as the team was not afraid to speak their minds this could make it easier to make strategic decisions.

Other teams had not been using external facilitators. They used their regular SCRUM master. Common for the SCRUM masters were that they all felt like they were a facilitator rather than a leader during the retrospectives. Those we spoke to about external facilitators were positive to the idea and mentioned they might want to try it out in the future.

Table 3.9: Usage of external facilitator

External facilitator	Teams
Used external facilitator	Alfa, Echo
No external facilitator	Charlie, Delta, Fox-trot, Golf, Bravo

Encouraging Through External Benefits Encouraging learning with external benefits like bonuses and such was not used by any teams. The closest was some teams that had sometimes brought some pastries to the meeting. One developer said that encouraging through bonuses would be a destructive force in the retrospective and the focus would be removed from the process.

During the retrospective

From both our case-studies we got insight into how retrospective meetings were conducted in today's practice. We will describe these results below and it is split into three parts. The first part is which practices are used by the teams today. The second part is the frequency and duration for the retrospective meeting for all the teams in the case study. The final part is reflecting on the team factors that affect the retrospective meeting.

Retrospective Practices Used When we asked the different developers and SCRUM masters how they conducted the retrospectives we got a wide range of answers.

The current state of the retrospective in team Zulu was described as a simple meeting where team members could speak their mind and discuss any issues they themselves wanted to bring up. However they had experimented with some different retrospective techniques and found them interesting, and considered using them occasionally. An important part of the retrospective was an attempt to reach a consensus. The tone of the retrospective was described as light, but sometimes could get more heated during discussions. The retrospectives were held every 3rd week, but could sometimes be moved due to pressure related to releases or other pressing issues.

Team Echo started their retrospective with ranging on the range one-to-nine the three categories; team flow, team moral and technical quality. This allowed them to see trends on the three categories. When we asked what happened if something was graded really low the response was that it had not happened yet. We also asked if the opposite, what happened if something was graded really high and the response was the same that it had not happened yet. After the grading the team used KJ-sessions, where one write positive and negative issues on post-it notes and then discuss them in the group. Which Dingsøyr et al. describes as a structured brainstorming technique [9]. According to the developer the things that came up during the KJ-session were really specific things related to the work environment. The reason for this was that the team worked on different pieces of software and that each sub-team had their own process.

Team Alfa described how KJ-sessions functioned as a good method for engaging team members. Members that were passive or silent during discussions could be an obstacle to learning if their feedback weren't gathered during the retrospective. The KJ-session helped hinder this as all the participants were forced to provide some feedback to the meeting.

Two of the teams, Charlie and Delta, varied each time how they conducted the retrospectives. They followed the five steps devised by Derby and Larsen [7], which is described in section 1.3.4 and varied the techniques for gathering data and generating insights. The reasons for this was to challenge the comfort zone of the members as well as counter group thinking and monotone activities. Group thinking is the desire for cohesion in a team and McAvoy and Butler [21] refers to it being considered as way of ineffective decision-making.

We asked the SCRUM master in team Delta if it always was new techniques or if some were repeated. The response was that the same technique never came twice in a row, but techniques that worked well would be repeated on several retrospectives. As to how they found the new techniques they used several sources. Blogs, websites like retromat [3] and agile podcasts were all mentioned as a good way of obtaining new techniques.

During our discussion we asked if there were any downsides to using varying techniques. Team Charlie's SCRUM master said that that for some participants it could become a little bit too much. This could result in lower enthusiasm for the retrospective. Another downside was that for some techniques the focus could move to the technique instead of the issues which could lower the results of the retrospective. However he said that he still found that varying the retrospective techniques was beneficial to the retrospective.

Table 3.10: Retrospective Techniques Used

Team	Technique
Echo, Alfa, Foxtrot	KJ-session
Zulu Golf, Bravo	Team Discussion
Alfa, Charlie, Delta	Varying Techniques
Echo	Team Barometer
Foxtrot	Weather Forecast

Frequency and Duration of Retrospectives The duration and frequency varied between the different interview participants. An overview can be seen in Table 3.11 Team Delta held retrospectives every week, while

team Charlie held every second week and yet team Echo held every third week. The SCRUM master for the team Delta said that doing it every week gave continuous follow-up. The team regarded the retrospective as the most important meeting during the development and all the team members saw the value that the retrospective provided.

Team Alfa performed retrospectives at irregular frequency. Instead of conducting retrospective after each sprint or at a given time they performed retrospectives after they had finished each major feature in their project. They also conducted retrospectives whenever one person in the team felt it was necessary. Usually a retrospective was held once a month. This resulted in two kinds of retrospectives being conducted. One feature retrospective where persons actively working on that feature participated and the complete development process of that feature was discussed. And one work process retrospective which handled general work processes for the team. This latter retrospective was conducted by an external facilitator. Of the two retrospectives the feature retrospective was the most common.

Both the developer and project leader in team Alfa found that feature driven retrospectives worked very well. However the developer admitted that he sometimes missed having all members of the team participating in the retrospective. Currently only the people actively participating in the development of the feature was invited to the retrospective. The developer said that there was a risk that some members of the team might participate in a retrospective rarely as their assignments didn't necessarily result in a feature being created.

There were varying degrees of participation to the retrospective, some teams like team Golf had every team member attend, while team Bravo only included testers in the release retrospectives.

The duration of the retrospectives varied between the teams, from a fixed amount of hours to until "we are done". For those having fixed time one team had 1 hour, while another had 2 hours long retrospectives.

Table 3.11: Frequency and duration for the different teams

Team	Frequency	Duration
Delta	Every week	1 hour
Charlie, Echo, Bravo, Golf	End of every sprint	1-2 hours
Foxtrot	Every second week up to six months	0.5-4 hours
Alfa	End of every released feature	1-2 hours
Alfa	On request by team member	1-2 hours

Team Factors When asked about the impact team dynamics has on retrospectives most teams emphasized the need for a positive culture and an eager attitude.

Team Echo described how one team member was a very “negative” person, constantly bringing up problems that needed to be fixed, however in the context of the retrospective this was seen as a positive, since it brought up necessary issues.

Team Charlie was a small team of four developers, where all were highly experienced, with especially two of the developers holding very senior positions within the company. The two senior developers were described as especially strong personalities. The team was described as very eager to learn and always searching for ways and ideas that could be used to improve the team. The strong personalities influenced the retrospectives in that sometimes the junior developers would agree without arguing strongly for their own ideas or suggestions. The SCRUM master told us that this could be a challenge during the retrospective.

In team Alfa they described how some team members were very outspoken and others team members had a tendency to take issues being raised very personally, when these type of team members collided it resulted in friction during the discussion.

After the retrospective

Our discoveries concerning work done after the retrospective mostly concerns the steps taken to implement the decisions that were made as a result of the retrospective session. Mostly teams with a clear protocol for designating responsibility for the action implementation were satisfied with their implementation process. Specifically assigning a name to an action instead of to the group as a whole seemed to lead more satisfaction in relation to the action implementation. When asked about reflection of the retrospective process on a team level no teams practiced this, however the SCRUM masters at team Delta and team Charlie practiced this at SCRUM master meetings. A further description on the implementation of retrospective actions can be found in section 3.1.2.

3.1.4 Impediments

In this section we discuss the impediments we discovered during our work. These are divided into team factors and retrospective timespan.

Team factors

In this section we elaborate on our findings of impediments relating to team factors, these are divided into “Cultural differences”, “Team Changes” and “Team personalities”.

Cultural differences As uncovered by our depth study of team Zulu they had a decrease in actions for a period of several months. They reasoned that having a foreign developer that created an “Elephant in the room” during the retrospectives was a possible reason for this. The situation absorbed the other problems within the team. No one would be rude and tell that the new developer that he was the problem. As one of the team members described during the feedback session:

“We actually discussed it once at the coffee after lunch, the retrospectives at the moment were just a waste of time”.

When asked what was the reason for the developer not living up to his expectations the team told us that face-saving and cultural differences made it difficult to communicate properly and that tasks that were assigned to him weren’t satisfiable.

When this came up during our interview with the team Zulu SCRUM master he agreed with the observations done by the team and elaborated that the communication issues had caused multiple problems. These communication problems became so prevalent that that they directly influenced retrospectives. Team members would consider the communication problems so major that they would consider more minor, and fixable issues irrelevant, thus leading to little or no issues being solved at all. This was not handled in the retrospective since personnel issues were not considered within the domain of the retrospective.

On the topic of norms or cultural differences having an impact the participants in the breadth-study reported low or no impact. team Golf reported some issues, but most of the work done on this area was done outside the retrospective. This work consisted of talking with team members ahead of the retrospective as well as building a culture ahead of issues arising.

Team Changes For team Zulu the foreign developer quit the team after a period of 34 weeks in week 263. This still meant that there was a low action period of 15 weeks between week 263 and 278 and this period remained unexplained. When inquired about this the team had several possible reasons. One were that there were already to many active actions on the plate resulting in fewer getting made. Another were that the communication within the team

improved after team had lost the developer creating the problems. A final possible reason was that the secretary for the retrospectives changed a lot within that period of time.

After discussing the decrease in actions period the team mentioned that it would be interesting to see if there were any correlations between retrospective actions and team changes other than the one already explained. For the third feedback session we created such charts as can be seen in Figure 3.11 and Figure 3.12.

In Figure 3.11 we can see that the period of low actions starting at week 237 and lasting out week 278. In that period the foreign developer, T7, joins and leaves the team. In the same period the SCRUM master of the team takes a leave of absence for a period of 20 weeks which also might have influenced the period.

For the secretary changes, shown in Figure 3.12 we can see that the changes in who writes the reports doesn't seem to influence the number of actions that comes out of the retrospectives. The only exception to this could be the period after week 263 where there are a lot of changes, as the team described. However we believe this to be unlikely as between week 270 and 275 the secretary stays the same and the amount of actions follows the trend of low number of actions and as changes earlier in the timespan haven't revealed any effects.

Team personalities The SCRUM-master of team Charlie said that personality might hinder the function of the retrospective. In his department SCRUM masters had been assigned to developers in teams where there only was developers. This had for some of the teams become a problem. In those teams that the SCRUM master had low enthusiasm for the retrospective practice, as they rather would just do regular developing, this low enthusiasm could spread to other team members. Another incident where team personalities impacted the retrospective was in team Charlie, where two senior developers would dominate conversations, leading to the junior developers becoming timid and lowering their input during the retrospective.

Retrospective timespan

In team Foxtrot, a developer told us that the one hinder for getting value from the retrospective was too long timespan between the retrospectives. In that team retrospectives could be held on irregular times ranging from two weeks up to six months. If the timespan was too big, the retrospective returned no value as there simply was too much to discuss according to the developer.

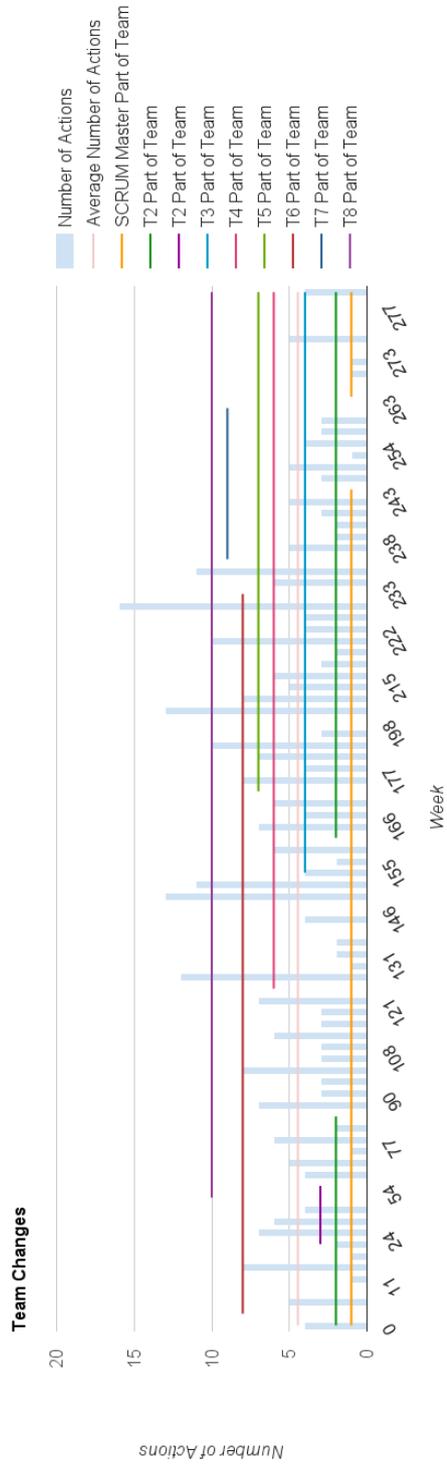


Figure 3.11: Team changes and the amount of actions for the retrospectives across the 278 Weeks.

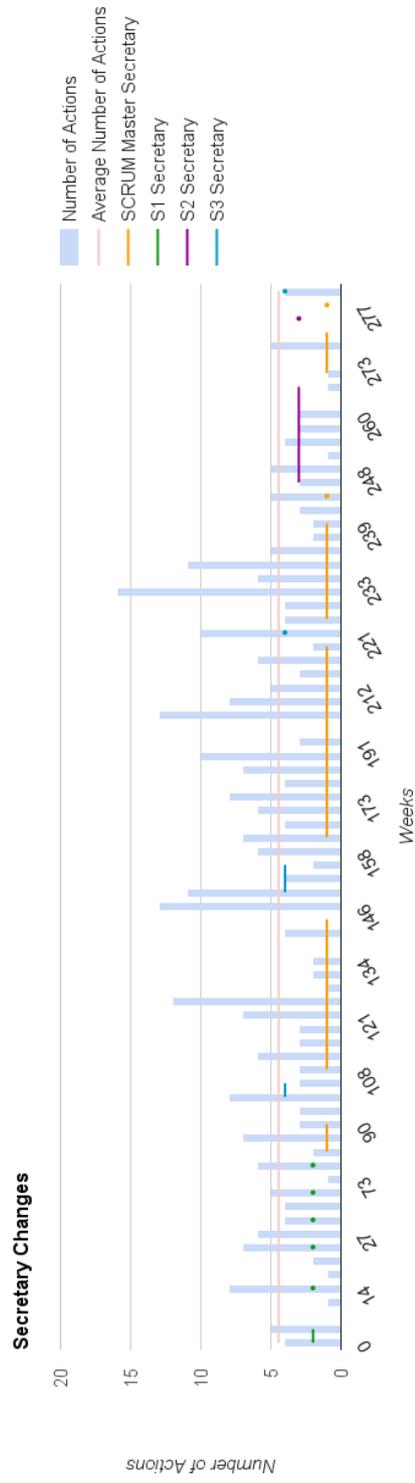


Figure 3.12: Secretary changes and the amount of actions for the retrospectives across the 278 Weeks.

Team Alfa’s feature driven retrospectives could lead to some developers being excluded for long periods of time as described in Table 3.1.3.

Distributed Team

Team Golf had parts of the team in Norway and parts of the team in China. This resulted in problems both at conducting the retrospective, due to time-zone differences, and cultural differences. In order to ensure that everyone spoke up they had members submit written issues before the retrospective meetings and had a strong focus on the discussion and order of issues being resolved democratically.

Table 3.12: Value Decreasing Hinders for the Retrospective

Team	Hinder	Description
Alfa	Rare participation from some members	As the feature driven retrospectives only invited people who had worked on a specific feature, team members working on legacy code could have a long timespan between participation
Charlie	Personality	SCRUM-masters who is not motivated to do retrospective creates low enthusiasm in the group.
Foxtrot	Timespan between retrospectives	Having too long time-spans between retrospectives creates more topics which takes longer to discuss and thus is unproductive.
Golf	Distributed team and cultural differences	The team was distributed and had large cultural differences, this was countered by using written preparation and having a strong focus on democratic decision making.

3.2 Organizational Learning in Retrospective Practice

In this section we will look at the results related to learning and organizational learning. We will first describe some examples of learning that have occurred within some of the teams. Then we will describe which types of learning occurs. Then some results on how the teams reflect about their own learning. After that we will describe the learning obstacles identified. Finally we will give an overview of the learning enhancing factors and potential improvements that the teams identified.

3.2.1 Learning Through Retrospective Practice

To create a common ground for discussing organizational learning we asked if there was any specific things that had changed in how the team worked or thought, through the retrospective. We will described three of these episodes here.

One team, Delta, had for a while been using much time on estimating the time needed to complete a user story. The estimations turned out to be generally wrong and during a retrospective this had become a discussion. This resulted in the team changing the practice completely. Instead of using much time on estimating time they used little time on giving story points for the stories which gave them a better estimate on the workload required to complete a story.

Team Echo had switched from SCRUM to KANBAN and back again over the course of half a year. This had been a result of having very time consuming process with a lot of steps from one begun planning until one user story was completed. The developer told us that at one point they had used more time on doing processes than developing. The team brought it up on the retrospective and they decided to try KANBAN instead. After a period of KANBAN the team found out that they needed some more structure than what KANBAN provided. This resulted in them reevaluating SCRUM and restructured it so it fitted what the team wanted better.

An example from Team Zulu was the arrangement of a “bug-crunch” day that was created during the retrospective lead by a external facilitator. The “bug-crunch” day is a day set aside solely to the elimination of bugs and similar issues, the day is held at regular intervals. This was described as a very positive change that had led to a clear decline in bugs.

3.2.2 Types of Learning

When we say types of learning we refer to the learning types single-, and double-loop learning. From our studies we gained some insights into which learning types occurred the most through the retrospective practice. We will first describe our results from the depth study and then the results from the breadth study.

Learning Types in Depth Study

In terms of organizational learning each action in team Zulu’s retrospective reports could be defined as single-loop, double-loop or undefined. The results yielded from the retrospective analysis showed that single-loop was the most occurring type of organizational learning with 66.4% of the actions. Double-loop had 27.2% of actions, and the rest was undefined at 6.4%. The distribution over the timespan, Figure 3.13 of the analysis showed that the three categories were evenly distributed. The active actions were very similar to the total amount of actions and only had some small negligible variances as can be seen in Table 3.13.

Table 3.13: Results from the content analysis regarding the organizational learning nature of the action.

Category	All Actions		Active Actions	
	Number	Percentage	Number	Percentage
Single-loop	227	66.4%	41	66.1%
Double-loop	93	27.2%	16	25.8%
Undefined	22	6.4%	5	8.1%

During the first feedback session with team Zulu the assumptions they had were as expected. They used the term direct causes as single-loop and root causes as double-loop, but we will continue using the terms of single-loop and double-loop. They expected that single-loop would have the most occurrences, even though they maintained a double-loop focus during the retrospectives. This expectation is reflected in our results quite well as 66 percent were single-loop and 25 percent double-loop. The reason they expected it to be more single-loop rather than double-loop was that it was easier to do single-loop and as one team member said: “...sometimes things just need to be fixed”.

The team were satisfied that they had what they considered a good ratio of double loop to single loop actions. The team made a formal decision

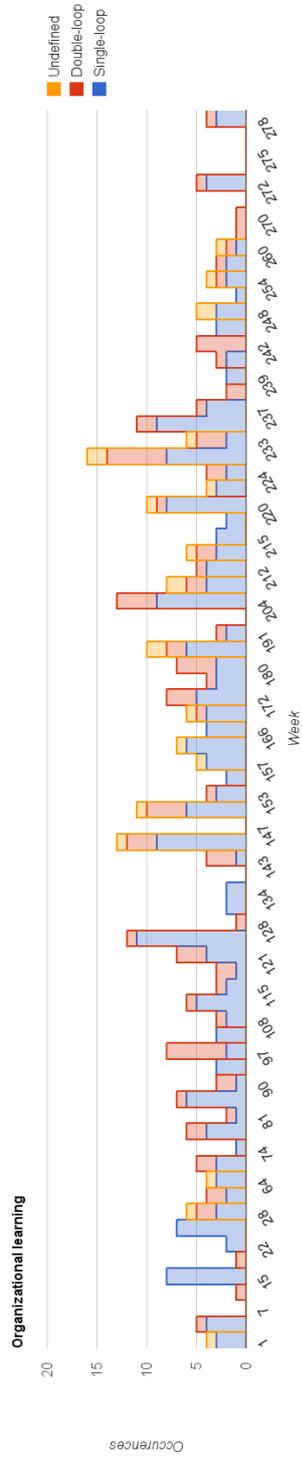


Figure 3.13: Timeline showing the distribution of learning loops for the total actions.

to work on identifying opportunities for root cause and double loop actions instead of fixing symptoms. Another action step taken was to formally log actions with a single loop or double loop attribute, as seen in section 3.1.2.

Learning Types in Breadth Study

The teams we interviewed had varied focus on root-causes. Several teams admitted that they rarely tried to find the root causes and only fixed the effects and not the underlying problems, thereby only conducting single-loop learning. However they told us that it really was something they should do or that they wished to do.

Team Charlie and Delta used the technique “five times why” to dig down to the root causes of the issues. They tried to find the root causes for every issue whether it was technical, procedural or personal. The SCRUM master presented us with a case where one of the developers had not been able to finish up their assignment. After digging into the problem they found the root cause which turned out to be that the person was not able to get help from the rest of the team. This was then addressed and actions were taken to solve it. The SCRUM master also told us that when technical issues did come up they dug down to see if there actually was a technical problem or a procedural problem. In most cases it turned out to be a procedural problem that was the root cause of the technical issue.

In team Alfa they simply discussed the issues until they found a concrete action that would hinder the issue to easily return at a later time. They started the retrospective with identifying issues. Then everyone would use three dots and place them on the issues which they found most important. Then they would start at the highest ranking issue and discuss it until they found the root cause and a way to counter it. Then they would continue on with the next issue. The developer in the team provided an example. During a retrospective a issue about long merging times had been brought up. After discussing the issue they found the root-causes for the long merging times were that they had too few releases as well as too big branches. Their solution to this was to start using another version control program making branches smaller in the transfer process as well as for the future, creating smaller tasks.

One developer in team Foxtrot told us that some issues were able to be solved immediately within the meeting. Other more difficult issues could require more investigation and thus would be followed up in the next retrospective.

Table 3.14: Root-Cause identifying techniques used.

Team	Technique	Description
Charlie, Delta	Five times Why	Through working in groups team members ask “Why?” five times in order to challenge their habitual thinking [7]
Alfa	Team Discussion	A discussion in the team where one discusses the issue until one find an action such that the issue won’t resurface.

3.2.3 Reflection About Learning

Through our studies we investigated reflection on learning. The results we have split into two parts. The first one is “Practices on Learning Reflection” which describes the practices that the teams from our breadth study use to reflect on their own learning. “Reflection on Learning Through Feedback Sessions” is the second part which describes the reflections and learning that team Zulu did through the feedback sessions which we held.

Practices on Learning Reflection

We asked whether any of the teams ever reflected on how they learned from the retrospective.

The SCRUM master of team Charlie said he was a part of the firm’s community of practice on development process. In this community they reflected quite a lot about both how to conduct the retrospectives and the results of them. However the results from these reflections never made it outside the community and to the teams. When we pointed this out to the SCRUM master he realized that this should not be the case. Some teams suffered from low enthusiasm about the retrospective and if the reflections reached these members that problem could possibly be countered. Team Delta was also a part of this community of practice.

In team Alfa, during the work process retrospectives the project leader found that they reflected over learning as part of the discussion. A developer in the same team however found that the last year the reflection had decreased a little. They had earlier used reports from earlier retrospectives and used time on reflecting on how things from that retrospective were handled in terms of resource management as well as things that did not get handled.

This had helped in the long run.

Most teams however did not use any time on reflecting on how they used or learned from the retrospective. However when we asked the question, most seemed to realize that this would be a good way of increasing the value from the retrospective.

Table 3.15: Learning Reflection

Technique	Team
Reflection in community of practice	Charlie, Delta
Reflecting in discussion	Alfa
Using Earlier Retrospective Reports	Alfa
No reflection	Echo, Foxtrot, Bravo, Golf

Reflection on Learning Through Feedback Sessions

From the first feedback session team Zulu reflected on the analysis conducted. The team found the presentation and analysis very useful. It was described as a refresh on retrospectives and they speculated if it was possible to develop a tool that could codify the actions from a retrospective and visualize them on a dashboard. They found the presentation provided a good awareness on the retrospectives, which earlier had not been that good. They also decided that they should focus more on finding root-causes as the presentation gave a short introduction to single-loop and double-loop learning.

During the second feedback session team Zulu's SCRUM master reflected on the learning results of the retrospective analysis. An interest was expressed in doing a more comprehensive analysis of the retrospectives within the team, but a major concern was a possible waste of time, with little return on investment. When discussing the process of spreading the lessons learned within the team to the outside there was no clear process for this, but creating a "notable efforts" documentation where positive efforts were documented was considered.

The fourth feedback session consisted of a presentation from the team leader, a discussion along with the team leader and SCRUM master and a report from an internal retrospective held about the depth study by the team.

"What have we learned from 77 retrospectives" was the title of the presentation the team leader held. The results of the analysis and the impact it had on the team were presented.

He talked about how retrospectives could be sad and described some of them as “more depressing than a funeral”. He also described how it could be hard to handle personal issues and the tendency of issues to pile up after retrospectives.

He presented how the external view made them realize how much they had actually accomplished in terms of implementing actions and it was described as “Positive to have someone from the outside as they are felt more independent”. How it had helped them realize they had a negative focus during the retrospective. He felt that already people were more conscious of being positive and there was less a “funeral” atmosphere.

It was also presented how single loop and double loop had been analyzed and that they for the future would focus more on double-loop learning.

As a final remark he described that they would make a new KIP dashboard for displaying positive actions, double loop and closed actions. To help keep a focus for the retrospectives and improvement opportunities.

The presentation reflected most of the results from the internal retrospective held and a summary of the report can be seen in Table 3.16.

Table 3.16: Retrospective feedback report summary

Issue	Description
Double loop	Aim to continue finding root causes instead of fixing symptoms
Track more key data	The team decided to add fields in the report describing the nature and depth of the action decided on. Where nature could be positive or negative, and depth could be single or double loop.
Numerator to record positives	Aiming to improve the tracking of positives the team decided to add a numerator to record positives
Dashboard creation	The team decided to create and use a dashboard with information similar to the information presented in this thesis.
Completed actions	In order to track what actions are completed they decided to keep them as open in their systems, as well as add a new field to these actions for “improvement management”

3.2.4 Learning Obstacles

According to the breadth study participants there were several obstacles that could prohibit learning.

Low enthusiasm for the retrospective within the team was seen as an obstacle by several teams. The reasoning was that the persons with the low enthusiasm would rather do something else.

Another obstacle was described as change. The retrospectives had to return some value in terms of procedural change. If nothing got done it would create low enthusiasm as there was no value to the retrospective. This could create a negative feedback loop as described above. Obstacles that supported this was mentioned by other teams. A long backlog for a sprint made it harder to take decisions, as a lot of energy was required to accomplish the actions that would be put there.

Another obstacle to learning were the external parties. The project leader in team Alfa told us that some decisions and actions were agreed upon by the team, but customers, leadership or other parties sometimes could hinder the team from following through the action.

Team Golf, who was a distributed team, added large physical distances as an obstacle. It made communication harder and thus could be a hinder to learning as information and feedback could be lost.

Table 3.17: Obstacles for learning

Obstacle	Description	Team
Low enthusiasm	Participants finds the retrospective a waste of time and would rather do something else	Charlie
Change	Without process improvement and visible changes the enthusiasm for the retrospective would decrease between the participants.	Echo, Alfa, Charlie, Foxtrot
External parties	Leadership, customers or other third parties could reject the teams from implementing actions	Alfa
Large physical distance	The team is distributed over large distances, making communication complicated	Golf

3.2.5 Learning Enhancing Factors and Potential Improvements

When asked about what attributes, that existed within the teams, contributed to learning the feeling of ownership to the process was described as a very strong factor by multiple interviewees. It was even described as critical to the success of a retrospective by the SCRUM master of team Delta.

The developer from team Echo expressed the high impact a SCRUM master had on the team's appreciation of the retrospective by ensuring that retrospective tasks were completed. The attitude of wanting to improve was also described as very important.

On potential improvements that would increase the potential for learning one interviewee said he would like to improve process of ensuring tasks set during a retrospective were completed.

Also expressed was a desire to be able to visualize issues during the retrospective. Team Charlie's SCRUM master intended to buy a white board that could be placed visibly to the team with a list of intended retrospective improvements. Team Zulu would do this as well, through implementation of a dashboard.

The project leader of team Alfa only held retrospective when the need arose, and thought that more frequent retrospectives might be beneficial. As previously mentioned team Alfa held retrospectives both after releases and when requested by one of the team members. The project leader was satisfied with the feature driven retrospective, but told us that he thought having request retrospectives more frequent could be beneficial as these retrospectives were more thorough.

Another potential improvement discussed was the inclusion of external parties, in order gain an outsider perspective of why they did the things they did.

Chapter 4

Discussion

Our goal for conducting these studies was the following: *Our goal for this study is to investigate the outcome returned from the retrospective in terms of organizational learning and retrospective characteristics.* Throughout this chapter we will first provide a descriptive discussion on the retrospective characteristics we have observed and then discuss the results in terms of organizational learning. Finally we will provide some reflections on the retrospective practice, the current state, a method proposal and some guidelines for conducting retrospectives.

4.1 Retrospective Characteristics

One of our research objectives was: *What are the main characteristics in current retrospective practices, in terms of outcome, processes and impediments?* Throughout this section we will provide a descriptive discussion on the characteristics we have identified throughout our studies. The section is split into three parts: Output, Process Characteristics and Impediments. An overview of all the characteristics can be seen in Table 4.1

4.1.1 Output

In this section we discuss the observations regarding the output of a retrospective process. The section is divided into “Work areas”, “Improvements” and “Enthusiasm”. Some of these sections are divided into subsections, where we discuss relevant observations.

Table 4.1: Retrospective characteristics

<i>Retrospective Characteristics</i>
Output
Reflection on work processes, technical issues, work environment
Creates improvement opportunities
Improvement implementation
Provides organizational learning
Can improve team enthusiasm
Can decrease team enthusiasm
Can improve efficiency
Facilitates empowerment
Process Characteristics
Little considerations taken
Wish to improve
Varying techniques
Occurs regularly
Collects opinions from participants
Arena for open discussion
Allows for experiments in work environment
Shared learning event
Impediments
Team commitment
Enforcing of process improvement actions

Work Areas

Several work areas are covered by the retrospective. Mainly we have seen three areas that are covered by the retrospective. These areas are work processes, technical issues and issues related to the work environments. As we saw in team Zulu about 40% of the actions created were related to technical issues and 59% to process issues. Some of these issues might have been work environment related, however our content analysis did not include this. For future work this should be considered. From the interview with team Echo we learned that the team mostly discussed work environment issues as the team consisted of several sub-teams that worked differently from each other.

We have seen very little of personnel issues brought up during the retrospective. The only case was with team Delta where one of the members never got help from the others. In team Zulu we saw that not being able to take up personnel issues hindered the retrospective. When asked, several of the

study participants voiced an opinion that personnel issues were something that should be taken outside of the retrospective to hinder blaming, and a negative mood during the meeting. We have not seen any earlier literature discussing the topic. We do believe however that the consideration of including personnel issues is something that could be appended to Dingsøy's [8] set of considerations, which should occur before the retrospective.

Improvements

Here we discuss the improvement outputs from a retrospective process. The outputs discussed are “Creates Improvement Opportunities” and “Improvement Implementation”.

Creates Improvement Opportunities Through our studies we have seen that teams are able to improve based on decisions created during the retrospective. From the depth study of Team Zulu we saw that through 77 retrospectives the team had created 343 actions which reflect improvement opportunities. Also all the teams in our breadth study created actions to improve some aspect of their work-life. It is clearly evidence that improvement opportunities are created through the retrospective. This seems to fit well with previous research [7, 8, 11] that the retrospective help identify improvement opportunities.

Improvement Implementation The question if the improvement opportunities are actually implemented can also be seen through our studies. The results of our studies revealed that most of the teams were satisfied with their implementation rate. From team Zulu we learned that only 65 of the 343 actions were not yet implemented. It was also revealed that implementation could be a challenge, as was the case with team Charlie. We identified two methods that seemed to help overcome this challenge. The first was assigning responsible team members to each action. The second was SCRUM master follow-up of the actions. We have seen that retrospective practicing agile development teams are able to implement improvement opportunities confirming Derby and Larsen's [7] statement of retrospective helping team adapt. This is contradicting the research of Drury et. al. [11] which finds that no real changes occur as a result of the retrospective.

Can Improve Efficiency The retrospective practice can, as we have seen in multiple examples, improve the efficiency of teams conducting them. Team Echo's practice changes from SCRUM to KANBAN and then to modified

SCRUM and Team Zulu’s “Bug-crunch day” are just two of the examples we have seen that the teams have been able to improve their efficiency through the retrospective practice. This again confirms the previous literature [7,8,17] that retrospectives are able to improve practices and contradicts the finding of Drury et. al. [11] that retrospectives provide no real changes.

Enthusiasm

Team enthusiasm is both affected by the retrospective practice and inflicted by it. It can be increased through a positive feedback loop or decreased by a negative feedback loop. As a result of the retrospective practice individual empowerment is facilitated and this also increases the enthusiasm. We will describe and discuss each of these statements below through examples from our studies and earlier literature.

Can Improve Team Enthusiasm We have seen that the enthusiasm of the participants of retrospective practice can be both affected by the retrospective. We uncovered a positive loop that helps increase the enthusiasm of the team conducting retrospectives. If changes occurred as a result of the retrospective the participants would become enthusiastic and thus the chance of more changes would occur. Ownership towards the development process was one factor that could help increase the enthusiasm and feed the positive loop. This positive loop confirms what Derby and Larsen [7] states that teams are invested in the success of improving their work as the improvements are chosen by the teams themselves and not from upper management.

Can Decrease Team Enthusiasm The retrospective practice has also the ability to decrease the enthusiasm of the practicing team. As the opposite of the positive loop a negative are able to decrease the enthusiasm. If no changes occur enthusiasm will decrease and as the enthusiasm decreases the chance of new changes occurring decreases. Some even might see the retrospective as a waste of time as described by team Charlie’s SCRUM master which had happened with some of the teams in his department. This confirms our previous literature review [13] that recurring issues kill the joy and Drury et. al. [11] research that some may see the retrospective as a waste of time.

Facilitates Empowerment Ownership towards the development process was seen as crucial towards getting improvement out of the retrospective by the interviewed teams. Each team member has the possibility to participate in shaping their working process through the retrospective. As seen in team

Alfa and Echo even the shy are required to participate in returning feedback and contribute solutions for current work processes. Tessem [28] identifies participation in process improvements as an empowering practice. This directly relates to the retrospective practice, which is also a parallel drawn by Tessem and our work supports this. Enthusiasm is increased as members are empowered [28] and thus the retrospective increase enthusiasm through empowerment.

4.1.2 Processes Characteristics

In this subsection we will discuss the process characteristics observations we made in relation to our results and established theory. The subsection will be divided into before, during and after.

Before

As seen in section 1.3.4 we will compare our observation results of the work done before a retrospective primarily to the theory from Dingsøy's [8] work. This section is divided into "Little Considerations Taken", "Wish to Improve" and "Facilitator".

Little Considerations Taken When we consider our results we see that few of the teams interviewed do a thorough consideration of their practices in relation to the approaches seen in Table 1.2. Most teams did not do an informed decision on several of Dingsøy's considerations. For example on who to invite or sharing tacit and explicit knowledge. For example team Alpha's interviewee said that it could go long periods of time where a developer was not invited to a retrospective. When it comes to sharing the knowledge generated none of the teams made a concerted effort to share the knowledge further in their organization that came as a result of learning through the retrospective.

Wish to Improve Many teams had a great wish to improve. As seen throughout our results the need to build a culture that allows for learning is absolutely essential for a productive environment. This is in accord with Dingsøy's work. Especially trust between the team members emerged as critical for reaching the maximum potential of a retrospective session. An example of the possible improvements is the team dynamic improvements experienced by team Delta when one of their team members brought up the problem that he was not getting help from other team members, as described in Table 3.1.2. Also after our depth analysis with team Zulu their eagerness

to improve let them turn the results from our analysis into a basis for multiple actions intended to improve the team’s learning capabilities.

Facilitator Deciding on the facilitator can be considered both part of before the retrospective, and during, as work needs to be done in advance. Few teams used an external facilitator as recommended and the experiences we observed were uniformly positive. This is in accord with Dingsøy’s work. However only two teams used external facilitators consistently, as described in section 3.1.3. Team Zulu had only one experience with using an external facilitator but it was described as a very productive and positive experience. This indicates that many teams would benefit from making a more concerted effort to utilize external facilitators.

During

In this section we will discuss our observations relating to what happens during the retrospective process. This is divided into “Occurs Regularly”, “Collects Opinions from Participants” and “Open Arena for Discussion”.

Occurs Regularly Retrospectives happens at a regular basis varying from team to team from every week to every six months. Usually it occurs after an ending of a development stage either a project iteration or a released feature.

Collects Opinions from Participants During the retrospective data is gathered from the participants. We have seen that feelings and opinions are two primary data types gathered. None of the teams in this study have used other sources of data for their retrospective, with one exception of lead times which were used by one of the teams. The participants in the retrospective consist of team members contributing to a project iteration or feature that the retrospective focuses on. These consists of developers, testers, designers, architects, consultants, SCRUM masters and in some cases project leaders as seen in Team Zulu.

Open Arena for Discussion The retrospective provides an open arena for discussion. Each participant are allowed to bring their own issues and the issues are then analyzed through discussion or root-cause analysis. A facilitator, either internal from team or external, facilitates the discussion. From the facilitators we have spoken to, some censor the discussion on some subjects. In team Alfa personnel cases where not allowed and would have to be discussed outside the retrospective. In team Delta the SCRUM master

tried to hinder technical discussions as their focus is on work processes. Other than this most topics are allowed during the retrospective as we have seen in Team Zulu.

After

In this section is our discussion on factors that take place after the retrospective process. This is divided into “Allows for Experiments in the Work Place” and “Shared Learning Event”.

Allows for Experiments in the Work Place As seen in section 1.3.4 a retrospective can be an area that facilitates experimentation in a team. We have observed this, for example as described with team Echo in subsection 3.2.1. Here team Echo tried to move to KANBAN from SCRUM, the experiment was not an immediate success, but allowed them to return to a SCRUM methodology that they could tailor after their experiences from the experiment. This resulted in their work methodology fitting their team better. Our work with team Zulu also led to experimentation, for example their decision to create a dashboard to log their retrospective actions. Another experiment by team Zulu was the inclusion of a “bug-crunch” day seen in subsection 3.2.1. This experiment was a success and led to a noticeable decrease in bugs.

Shared Learning Event The theory of the retrospective as a shared learning event was described in subsection 1.3.3. An example of a shared learning event from the same section was performed by team Delta, as they changed their time estimation practices to great success after discussing the process in a retrospective. However none of the teams interviewed made an organized effort to make the result of the retrospective a learning event for personnel outside the team, as recommended by Dingsøy [8].

4.1.3 Impediments

In this section we discuss our observations on impediments in context of the retrospective process. The sections is divided into “Personalities”, “Team Commitment”, “Enforcing of Process Improvement Actions” and “Availability”.

Personalities Some personalities could provide obstacles for the retrospective. We saw three examples of this. The first one was in team Zulu where cultural differences provided miscommunication and difficulty providing an

open discussion in the retrospective. The second was in the department to team Charlie where some SCRUM masters had low enthusiasm for the retrospective practice and this could influence the rest of the teams as well. The third example were two senior developers with strong personalities, in team Charlie, that could hinder the other developers from voicing their opinions. These were the only three cases we saw in our studies, but further personalities may be investigated. Derby and Larsen [7] talks about personalities that take a lot of time and hinder others from taking part in the retrospective and this is reflected pretty well in the last example. They suggest that talking to them privately and directly asking them to hold back a little could help the situation. Team Alfa recommended using KJ-sessions to help everyone participate.

Team Commitment Even though most of the teams in our study was satisfied with the commitment from their teams, implementation of actions still could provide a challenge. As mentioned by several of the interviewees if the actions were not assigned to a specific person the action would not be implemented. This indicates that the team as a whole don't have the commitment to implementation of retrospective actions. Drury et. al. [11] described several obstacles that fit with these results. Team members unwilling to commit and relying on SCRUM master, not implementing decisions or relying on others for decisions, and not taking ownership of decisions are all obstacles that could be identified through our research. However as mentioned this was only the case when the group as whole was assigned to a decision, when individuals of the group were assigned the teams were satisfied with the implementation rate. This still indicates however a lack of team commitment even though the consequences are dealt with.

Enforcing of Process Improvement Actions Enforcing the implementation of process improvement actions was seen as a challenge. The SCRUM master of team Zulu described how enforcing and monitoring process improvement actions was a challenge. Some process changes required the whole team to implement the action and both enforcing the implementation and monitor it could be hard. Conflicting priorities and not taking ownership for decisions are two of Drury's et. al. [11] obstacles to effective decision making, and these results reflects the two obstacles.

Availability The last impediment we have seen for conducting retrospective is availability. In team Golf we saw how having a distributed team could make it harder to conduct the retrospective. In team Alfa and Fox-

trot we learned how the unavailability of retrospective due to long timespan or participants not being able to participate could inhibit the retrospective. Zedtwitz's [31] barrier of memory bias and Drury et. al. [11] obstacle of unavailable staff is reflected in this.

4.2 Organizational Learning

One of our research objectives were: *How is learning achieved through current retrospective practices, in light of organizational learning theory?* Throughout this section we will first discuss the results of our case-studies in terms of the governing values of Argyris and Schön's [6] organizational learning Model I and Model II. Secondly we will discuss our results in terms of learning types. Finally we will discuss the impediments for learning that we have seen throughout our studies.

4.2.1 Governing Values

Argyris and Schön [6] described several governing values for learning organizations as we described in section 1.4. Throughout this section we will reflect on our results using these governing values, investigating how retrospectives is performing as a learning practice.

Model I

In Model I, described in subsection 1.4.3, four governing values set the focus for the learning organization. In our investigation of the retrospective practice we have seen very little to any of these values. We will discuss this below for each governing value and then the consequences before we summarize the findings related to Model I.

Setting and Achieving Goals One could argue that the team setting goals and achieving them could be compared to creating actions and fulfilling them, however we do not support this as the fulfillment of actions is part of a collective efficiency improvement and learning practice. The retrospective and its participants are instead of trying to design and manage the environment unilaterally, investigating all the different angles and perspectives the team can present and finding solutions to the problems existing within that environment. The joint team discussion and retrospective practices are evidence of this.

Maximize Winning and Minimize Losing Maximizing winning and minimizing losing is not visual in the current retrospective practice. The teams are not afraid to use the retrospective as an arena for creating experiments on new practices where some might work and some might not. Team Delta and Echo, gave examples where the teams had tried to implement new practices and instead of going down with the ship when the practice had not

worked, leave ship and try something else. In Team Delta things that did not work for the members of the team simply did not get done and it was a joint understanding that actions that did not get any attention were bad practices. In Team Echo the team was not satisfied with current work practices and changed them drastically. After some months time they found that these new practices only made things worse and instead of claiming ownership for the task and try to force it through, the team decided to try something new.

Minimize Generating or Expressing Negative Feelings Through retrospectives we have seen that the participants are allowed to express their feelings regardless if they are good or bad. This effectively counter the third governing value which is minimizing generating or expressing negative feelings. Team Echo and Delta both described events that showed participants raising negative issues and feelings towards the team. From the retrospective analysis for Team Zulu we saw that 89.3% of the actions created came from negative issues. Even though allowing negative feelings can be a good thing some of the interview subjects said that the atmosphere could become a little bit too negative sometimes, and that the retrospective should not be an arena to vent. As was the case with Team Zulu the third and fourth feedback session revealed that the team had become more aware of raising also good feelings and issues during the retrospective and they felt the retrospective had improved as a result. This indicates that allowing negative feelings is important as one can learn and improve from them. However one should also ensure that good feelings are raised as they also provide the same opportunities and creates a more enthusiastic feeling about the retrospective.

Be Rational As feelings are encouraged to be shared during the retrospective, to some moderation, the fourth governing value being rational also seems to not be apparent in the retrospective. Being rational implies censoring feelings and as we described above retrospectives encourages sharing of feelings towards the group.

Consequences The consequences resulting from Model I governing values also seems to be rarely encountered in retrospectives in terms of the behavior. We have witnessed one case where an actor acted defensibly and such created an atmosphere that suppressed the other participants feelings. Creating what was described as an elephant in the room. It was also confirmed by the interviews that personnel issues were not addressed during the retrospective and such events could decrease the value gained from the retrospective. Other than this we have neither seen nor heard about teams having defensive norms,

or having defensive interpersonal relationships.

Most of the learning consequences seems to be opposite of what is expected by Model I except single-loop learning and possibly decreased long-term effectiveness. It might not be surprising, that as the governing values are rarely encountered the consequences of them is not either. Neither self-sealing, lack of public testing of theory or too much testing of theory in private seems to be appearing in teams conducting retrospectives. What is more surprising is that single-loop is quite occurring. We will dwell deeper into this in subsection 4.2.2. In terms of decreased effectiveness we have no way to compare these to any of our results as we lack control group applying most of the governing values from Model I.

Model I Summary The governing values from Argyris and Schön's Model I seems to not occur or be the system employed by agile development teams performing retrospective systems today. We have only seen two cases where the governing values of Model I have had any implications on the teams. The first one was that of face-saving from one of the team members resulting in an atmosphere suppressing the other member's feelings. This member later left the team. The other implication of Model I governing values is that of single-loop learning which most teams experience and will be discussed further in subsection 4.2.2

Model II

The governing values of Argyris and Schön's Model II are more apparent in the teams that practice retrospectives. We will discuss our findings related to each of the governing values and the consequences observed below. A summary of this discussion is at the end of the subsection.

Valid Information Valid information is the first of the governing values of Model II and the retrospective practice require information gathering from the team members. We have seen several ways that retrospective teams gathered information. Nominal brainstorming through KJ-session, around the table discussion and others techniques have all been used to gather information and find issues with the development process. Zedtwitz [31] found memory bias as one of the barriers to learning and Bjarnason and Regnell [4] proposed evidence based timelines as a technique to counter this. None of the teams we investigated used this technique and none of them mentioned memory bias being a problem for the retrospective. Neither did we see this through our content analysis of team Zulu. However during the feedback sessions with team Zulu looking back at specific events happening a year

Table 4.2: Governing values and consequences encountered in relation to retrospectives.

<i>Model I</i>	<i>Encountered</i>
Governing Values	
Defined goals and try to achieve them.	No
Maximize winning and minimize losing.	No
Minimize generating or expressing negative feelings	No
Be rational	No
Behavioral World Consequences	
Defensive actors	Once observed
Defensive interpersonal and group relationship	No
Defensive norms	No
Learning Consequences	
Self-sealing	No
Decreased long-term effectiveness	Not observed
Single-loop learning	Yes
Little testing of theories publicly	No
Much testing of theories privately	No

past the team was uncertain. This gives us reason to believe that in terms of iteration retrospectives which happen regularly with a timespan of weeks don't suffer from memory bias. However feature driven retrospectives and project retrospectives could suffer from this.

To identify the issues we have seen that the practitioners of retrospectives uses only information gathered from the participants with one exception of a team using lead times as a measurement tool. However we have not seen many examples on using other information tools to evaluate solutions for issues than the participants of the meeting. We have seen team Zulu and team Foxtrot postponing issues until they can investigate it further. This have not come up in our discussions with other teams, but it can seem that the teams during retrospectives acquire knowledge when their own is lacking.

Free and Informed Choice The second governing value of Model II is free and informed choice, and through our study we have seen that teams are mostly free to make their own improvements through the retrospective. Many firms have adopted agile methodologies and an important part of it is having short feedback loops and thus the retrospective can be a valuable practice. None of the teams spoke of problems with management having

too little time or being allowed to conduct retrospectives. This effectively eliminates Zedtwitz [31] barrier of managerial time-constraints. The second managerial barrier of bureaucratic overhead is also for most cases absent from the retrospective practice. Teams are free to conduct the retrospectives in any matter they themselves chooses. The only cases where the barrier provides any impediments are cases where implementing an action has a high resource cost which we learned from team Alfa. Also issues that required change with external parties could hinder implementation as we learned from Team Zulu. Other than that we have seen that all the teams are free to conduct and manage their own retrospective practice, providing an environment of free and informed choice for the team.

Internal Commitment to the Choice and Constant Monitoring of its Implementation The third and final governing value of Argyris and Schön's Model II for organizational learning is internal commitment to the choice and constant monitoring of its implementation. This value is one of the challenges facing retrospectives today. All the teams in the study emphasized that getting things done and actually see the actions followed through was crucial to having a valuable retrospective. If the team could not see any choices become implemented this would create a negative feedback loop where participant's enthusiasm would lower and the chance of new actions being implemented would decrease even further. However as we have seen in our content analysis only 19% of the actions created were still left unresolved. Most of the other teams we interviewed seemed pleased with the action implementation. That most teams acknowledged the risk of not implementing actions reveals that teams have a focus towards maintaining the issue.

Enforcing process improvements was admitted as a challenge by some of the teams and thus reveals that implementation of actions could prove difficult. Considering all the input we got on the subject push-tactics, assigning responsible individual, worked better than pull-tactic where it was expected that some would handle the implementation. This reveals a lack of commitment and aligns well with Drury et. al. [11] findings that daily operational tasks trumps that of strategic/tactical tasks made during the retrospective. The interviews revealed that enabling the team-members to acquire an ownership towards the work process improved the commitment to the retrospective and implementation of tasks. Some teams also added retrospective actions as a part of the backlog to increase the implementation rate.

Consequences The consequences of the three governing values for Model II are both apparent and absent in the teams participating in the study.

According to Model II, organizations that focuses toward an organizational learning II system will have actors that are experienced as minimally defensive. As earlier mentioned we have seen only one case were an actor has behaved defensively, and where the actor later left. The retrospective practice would suffer during such actors as seen in case of team Zulu, censoring the rest of the team as they would not openly give blame to the actor. Reluctance to blame is one of the team base barriers identified by Zedtwitz [31] and the team suffered for this. However as this was a single case it indicates that such types of actors are not welcomed into teams approaching such organizational learning II systems. Thus we can say that actors are minimally defensive during participation in retrospective. Zedtwitz barrier of blaming will also seem to be removed. However we have not seen any blaming during our studies and as earlier mentioned, subsection 1.4.3, this is an action strategy occurring in organizational learning system applying Model I which one would seek to avoid. Instead replacing it with confronting current views.

Another consequence of the behavioral world, described by Argyris and Schön, is: “Minimally defensive interpersonal relations and group dynamics”. In general our study has seen little of actors, that are participating in retrospectives, acting defensively towards the other participants in the group. We have only seen one example that an actor act defensively. In this case the problem was solved. Team Delta provided an excellent example on one of the actors acting minimally defensive. The actor had acknowledged for the rest of the group that he wasn’t able to get help from anyone. The SCRUM master said that trust had made it possible to bring up the issue. One can deduce that acting minimally defensive towards others is closely related to trusting one another. Team Alfa and Echo also said that trust was important for the retrospective.

The retrospective in itself is a learning-oriented norm, which is the third consequence described by Model II. The participants of a retrospective performs it to learn from the last phase of a development process, and find new opportunities to improve.

The fourth consequence of the behavioral world is related to freedom of choice, internal commitment and risk taking, which the retrospective all provide an opportunity for. As mentioned above the freedom of choice and risk taking are granted by the retrospective as long as it is not too costly in terms of resources or requires change from an external party to the retrospective team. Internal commitment can be, as described above, a challenge to agile teams performing retrospectives, and creating ownership towards the development process as well as implementing actions is important to overcome

this barrier.

The consequences of learning and effectiveness for approaching a Model II learning system is frequent public testing of theories, double-loop learning and disconfirmable processes.

Frequent public testing and disconfirmable processes are both seen throughout our studies. Conducting retrospectives enforces participants inquiry into their current work processes and adapt, discard or improve them. An example of this is team Echo who changed practices from SCRUM to KANBAN to improve, but found that this was less effective and instead went back to an adapted version of SCRUM that suited the team better.

Double-loop learning on the other hand is not as apparent as the rest of the consequences and we will dwell more into this in subsection 4.2.2.

For the increased effectiveness the retrospective seems to yield better work practices. Most teams in our studies revealed that the retrospective helped increasing effectiveness for the work practices.

Model II Summary The governing values and it's consequences for Argyris and Schön's Model II of organizational learning systems are both apparent and absent from agile teams and retrospectives. Valid information and free and informed choice are both seen in retrospectives. Internal commitment and implementation is also seen, but regarded as a challenge by the teams conducting retrospectives. The behavioral consequences this yield are relationships between actors and the actors themselves are less defensive, learning oriented norms and high freedom of choice and risk taking. The learning consequences from retrospectives are frequent public testing of theories and disconfirmable processes. Double-loop learning is seen in some teams, but not in others. In general the retrospective practice and teams that are conducting them are approaching an Organizational learning II system with some impediments still apparent in the practice.

4.2.2 Learning Types

Of the three learning loops described in section 1.4 some were more occurring than others in our studies. From team Zulu we learned that 66.4% of the actions were a single-loop learning result. 27.2% found the influences of the issues and fixed them resulting in double-loop learning. In total only four of the teams studied had any focus on root-cause and double-loop learning. Only two of the teams had some reflection on how they learned from the retrospective, which is triple-loop learning.

We find these learning results surprising as Model II is supposed to facilitate double-loop learning. As most of the governing values are in use

Table 4.3: Governing values and consequences from Argyris and Schön’s Model II encountered in relation to retrospectives.

<i>Model II</i>	<i>Encountered</i>
Governing Values	
Valid information	Yes
Free and informed choice	Yes
Internal commitment	Challenge for some teams
Monitoring of choice implementation	Challenge
Behavioral World Consequences	
Actors minimally defensive	Yes
Minimally defensive relations and group dynamics	Yes
Learning-oriented norms	Yes
High freedom of choice, internal commitment, and risk taking	Yes, but internal commitment is a challenge for some teams
Learning Consequences	
Frequent public testing of theories	Yes
Disconfirmable processes	Yes
Double-loop learning	Appearing in some teams, absent from others

during the retrospective one could assume that double-loop learning would occur more. Especially as most of the teams had a wish to do so. Of all the teams only Alfa seemed to perform double-loop learning on the issues they discussed, however they only discussed the most pressing issues. Team Zulu performed double-loop on about a third of the issues and team Delta found the root-causes on issues they found important. Drury et. al. [11] found that operational daily tasks are prioritized above tactical and strategic ones and this can seem to be one possible reason for not doing double-loop learning. Teams may simply find solutions to current problems and not investigate if these problems can occur again and if some extra measures taken should be created to avoid them.

Ideally every issue should result in some double-loop learning. However in a realistic world, where time is a valuable resource, taking the time to investigate every issue to its root-cause and implementing a solution can be difficult. Especially if external pressure to perform is present. This can skew

the focus of the retrospective and result in only single-loop learning being the result from it. It would be interesting to see which types of issues is most important and should be given the time to conduct double-loop learning. In team Alfa and Bravo they voted on which issues they found most important and this could be a good indicator on which issues to dig deep into. However we have not been able to get data on this, but it could make an interesting topic for future studies.

We have seen that triple-loop learning was not apparent in any of the teams except Alfa and to some degree Charlie. Team Alfa was in general very satisfied with their practice of retrospective and also indicated that they perform learning so that issues will not recur, thus double-loop learning. We believe that this provides an example of triple-loop learning and reflection on the retrospective helps teams focus the retrospective and improve the learning value from it. Through our feedback-sessions with Team Zulu we have reflected together with the team and provided an arena for triple-loop learning, reflecting on how they conduct their retrospectives. The final feedback-session revealed that the team had decided to keep a better focus on doing double-loop learning and include more positive issues, strengthening our assumption that triple-loop learning helps focus the retrospective. That our interview subjects also responded that it was a good idea and should be done, strengthens this as well.

The three learning types single-, double-, and triple-loop learning can all be a part of the retrospective practice. In our study we have seen that most issues discussed during the retrospective results in single-loop learning, even though they are approaching a Model II learning system. Some teams are able to do double-loop learning on issues they find important, like Alfa, Delta, Echo and Zulu. Triple-loop are not seen much during the retrospective. In team Zulu it helped focus the retrospective, and as team Alfa was doing reflection and managed to do double-loop learning we assume that this triple-loop learning helps facilitate double-loop learning and focus the retrospective.

4.2.3 Impediments for Learning

Through our studies we identified several impediments for learning.

Focus on Double-Loop Learning

The first impediment is the lack of focus for double-loop learning. Even though the teams practicing retrospectives have the properties that should facilitate double-loop learning, very few teams are able to do it. We believe

this is a lack of focus where teams rather find solutions to problems instead of solving the influence that made the problem occur in the first place.

Reflection on Learning

We see the lack of reflection on learning as the second impediment for learning for retrospective performing teams. We performed feedback sessions together with team Zulu, reflecting upon learning, and it resulted in the team to regain a focus on double-loop learning as well as focus on positive issues as well as bad ones. Performing this kind of reflection clearly gave some increased value and changes for team Zulu's retrospective and we believe this can be the case for other teams as well.

Generalizing Knowledge

The third impediment is the difficulty of generalizing knowledge from specific events. This was originally found by Zedtwitz [31] in 2002, and we believe still is an impediment today. Zedtwitz described the impediment as:

“The human mind is not made to abstract experiences to a general level so that they can be applied to a wide range of future projects. Furthermore, project results (no matter if they are positive or negative) are often naively extrapolated in a simple linear fashion. The reality, however, is much more complex, so that the outcome of a project depends on a whole variety of interlinked variables which again are very difficult to generalize.”

We believe that this challenge is closely connected with single-, and double-loop learning where the teams only find solutions for the specific event instead of finding the general cause of such events occurring.

Internal Commitment

Internal commitment is as previously mentioned a challenge for some retrospective conducting teams and is the fourth impediment for retrospectives. When the commitment of participants is low, valuable feedback from either opinions or implementation of actions may be lost. This can inhibit the team from learning from those opinions or action implementations and result in the team not improving work practices.

Low Enthusiasm

Low enthusiasm is the fifth impediment and is closely related to the internal commitment. Low enthusiasm feeds a negative loop where changes and improvements don't occur and this lowers enthusiasm and such creates even lower enthusiasm. This results in lower internal commitment.

Little Action Implementation

Little action implementation is the sixth impediment. Almost all of the teams we talked with confirmed that if actions were not done it would result in lower enthusiasm for the team. This feeds into the negative feedback loop described above.

Tacitness of Process Knowledge

Tacitness of process knowledge is another impediment originally described by Zedtwitz [31]. Team Echo employed an external facilitator and listed one of the reasons for doing so as the facilitator forcing the team to explain their process explicitly so the facilitator could understand what was tacit in the team. By acknowledging that it could be a problem without an external facilitator the impediment could still be a problem among other teams.

Bureaucratic Overhead

The final impediment is bureaucratic overhead. For the cases where investigating an issue or implement a solution for it is resource costly, the upper management might reject the teams attempt to do what is needed to resolve the issue. However this seems to be pretty rare according to team Alfa and our analysis of team Zulu.

Impediments Reflection

From our studies we found several impediments to learning. These impediments results in low quality feedback from either participants or implementation of actions. If these impediments were overcome the feedback would be better and the team could be able to learn even more from it. A quick overview of the impediments are shown in Table 4.4.

Table 4.4: Learning impediments

<i>Impediment</i>	<i>Description</i>
Lack on focus for double-loop learning	Tries to find a solution to issue, rather than prevent it to occur again.
Difficult to generalize	Create general rules for specific events.
No reflection on learning	The lack of triple-loop learning in teams inhibits the team from improving their learning practices.
Internal commitment	Participants that don't contribute or help implement actions prohibit learning as the team will miss feedback.
Low enthusiasm	Learning decreases if the participants are not motivated to do retrospective. Provides negative feedback loop. Low enthusiasm gives no changes which gives lower enthusiasm. Creates low internal commitment.
Little action implementation	Creates lower enthusiasm.
Tacitness of process knowledge	Makes it difficult to get an objective view on current state of working practice.
Bureaucratic overhead	Issues that require a lot of resources to investigate or implement solutions for, could be stopped by management.
External Factors	Issues that relate to some external factors like other teams on same project, support or the like can be a challenge to implement solutions for and thus miss feedback from implementation.

4.3 Reflections on Retrospective Practice

In this section we will reflect about the retrospective practice and its characteristics, value and challenges. We will also present a proposal for a new method supplementing the retrospective and a set of guidelines that could help the practitioners facilitating valuable retrospectives.

4.3.1 Current State of the Retrospective Practice

Throughout this study we observed several characteristics and have seen the organizational learning through the retrospective practice. We will now summarize all we have learned through our own framework, where we divided the retrospective into three parts, before, during and after. The framework is described in section 1.3.4. A visualization of this is provided in Figure 4.1.

Before Retrospective

The first part of the retrospective practice consists of three elements that we have been able to see through this study: Considerations, bring knowledge and bring enthusiasm.

Before the retrospective the facilitator or team is required to consider some considerations. Dingsøyrr [8] proposes several things, however we have seen few of this employed in practice. The considerations we have observed employed by teams today are external facilitator and open or structured discussion. Two of the teams, Echo and Alfa had made the decision to employ an external facilitator. Several of the teams had chosen KJ-Sessions as discussion form, while others just held an open discussion. We have not seen any evidence that any of the teams has made any other informed decision on Dingsøyrr's considerations.

Bringing knowledge and enthusiasm is required by the participants before the retrospective. If no one has any feedback from the last development phase then no learning or improvement opportunities could be identified. Enthusiasm could be brought either as negative or positive as we have seen multiple examples of, in among others teams Zulu, Echo and Charlie. If positive enthusiasm is brought, it will feed into the positive loop, described in section 4.1.1. If negative enthusiasm is brought it will feed into the negative loop described in section 4.1.1.

During Retrospective

During the retrospective we have seen that Derby and Larsen's [7] structure for conducting retrospectives, described in section 1.3.4, is mostly followed

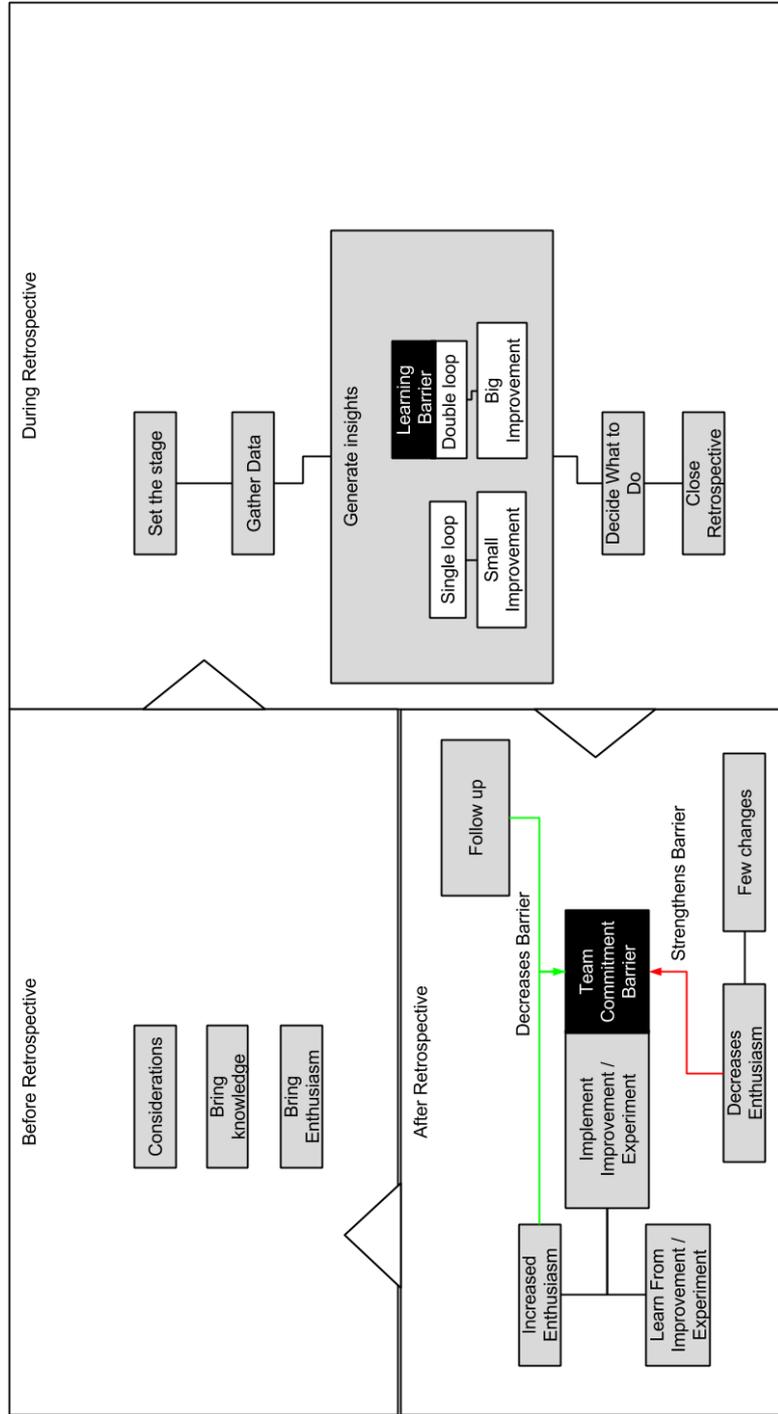


Figure 4.1: Our perceived state of the retrospective.

by all the teams in this study.

“Set the stage” and “Close Retrospective” is not explicitly mentioned by any of the teams except Charlie and Delta, but it is safe to say that they do start and end the retrospective. The three other steps, “Gather Data”, “Generate Insights” and “Deciding What to Do” is done by all of the teams in our study.

While investigating the learning effects, of the retrospective practice, we found that during the “Generate Insight” step a learning barrier was present. The learning barrier consists of several learning impediments, which is described in subsection 4.2.3, inhibits the team’s ability to perform double-loop learning and thus big improvement opportunities are lost. Several of the teams, Zulu, Echo, and Alfa, were able to penetrate this barrier however not without extra effort. For the cases where the teams were not able to penetrate the barrier single-loop learning would be the outcome of the “Generating insight” step.

After Retrospective

The final part of the retrospective practice is after the retrospective. In this part several new findings revealed a more complex picture than the previously assumed elements of “Learning”, “Improvement” and “Enthusiasm”.

We identified a barrier to implementation of improvement opportunities that was strengthened and weakened by several other steps. The barrier consisted of the impediments related to team commitment, described in section 4.1.3. If few changes were the result of the retrospective it could decrease the enthusiasm which would feed into the negative loop, described in section 4.1.1, and strengthen the barrier. If the team was able to overcome the barrier and see clearly improvements they would be able to learn from these improvements and enthusiasm would be increased. When the enthusiasm increases it would feed into the positive loop, described in section 4.1.1 and decrease the barrier. Another measure to decrease the team commitment barrier were follow-up done by the SCRUM master as this would help implementation of improvement opportunities.

4.3.2 Method Proposal: Meta-Retrospective

In this section we aim to discuss the potential of improving the current state of the retrospective in a team. We will propose a method based mainly on our depth study work with team Zulu. This method we decided to call a “Meta-retrospective”. The intention is to provide a framework for evaluating the current state of a retrospective in a project, and potentially improving

the retrospective process as a result. The aim of this process is to decrease the learning barriers affecting the team. An overview figure can be seen in Figure 4.2. This section will describe a process on how a “Meta-retrospective” potentially could be conducted.

Motivation for Meta-Retrospective After observing that no teams practiced team level reflection of own learning and the retrospective process, as seen in section 3.1.3, we developed the meta-retrospective as a method-proposal. The motivation is based in triple-loop learning, described in subsection 1.4.5 where one reflects on ones own learning process and learns from it. Seeing the results from the feedback sessions with team Zulu and the results of their own internal retrospective on retrospectives we observed that they had created a better focus on how they should benefit most from the retrospectives including more focus on double-loop learning, better monitoring of improvement implementation and a more positive attitude during the meetings. All actions and improvements that will approximate towards an organizational learning II system as proposed by Argyris and Schön [6].

Set the Stage The very first step necessary is to set the stage for the retrospective. This includes informing the team of the intent of the meta-retrospective, making sure that they understand and the focus is long term learning, for example a team member might think that the meta-retrospective will just cover the most recent sprint. A second consideration is who should facilitate the meta-retrospective, in our work with team Zulu we acted as an external facilitator. This was considered a positive aspect by the team leader, as seen in section 3.2.3. However this responsibility should be delegated after consideration by suitable personnel, for example the team leader or SCRUM master.

Gather Data The second step for the meta-retrospective is to gather data that will serve as a basis for discussion. This is in order to contribute an objective assessment of the current state of the retrospective. We propose using the last meta-retrospective report and all iteration retrospective reports since then as some of the data for this meeting. Also the evidence-based data proposed by Bjarnason and Regnell [4] could be used. During the meeting we also suggest gathering issues and feelings related to learning or the retrospective practice from the participants.

Reflect on Learning During this step the team should be presented with the data, and discuss trends or other notable observations made. The facil-

erator should aim to let the team discuss as much as possible. One step we performed with success was to have the team guess some trends from the data gathered. This allowed for discussion on the impression team members had, as well as a discussion on the objective results. This session lasted a little under two hours when we performed it. The team should be allowed to come to their own conclusions when presented with the data in order to increase their sense of ownership to eventual decisions. Therefore it is important to have an open discussion when presenting for example trends, and be open to input from team members. The team should if needed be motivated to suggest improvements if any are needed.

Decide what to do When the reflection on learning session is finished, decisions have to be made. We suggest looking at the bigger picture and focus on making decisions, based on effective decision making. The decisions should be either tactical or preferably strategic.

Closing the Meta-Retrospective The meta-retrospective should ideally be closed by formally converting any improvements or knowledge into explicit knowledge, for example by writing a report as done by team Zulu. The closing of the meta-retrospective should be when there are no more factors to discuss or consider.

When to hold a Meta-Retrospective The optimal interval between each meta-retrospective is uncertain, however we believe that holding it after each regular retrospective is not necessary. Several times a year seems reasonable, but not too often, and it should strive to fit the schedule of the team in question.

Team Zulu Meta-Retrospective Example In the case of team Zulu they had a meeting separate to the data analysis discussion where they decided formally on what actions they wanted to implement. This allowed them to let the impressions from the data analysis presentation and improvements suggested to mature. The team held a semi-structured retrospective as described in section 3.2.3, where the agenda had been developed through a cooperation between the team leader and the external facilitators. The team then made a written report as usual after a retrospective.

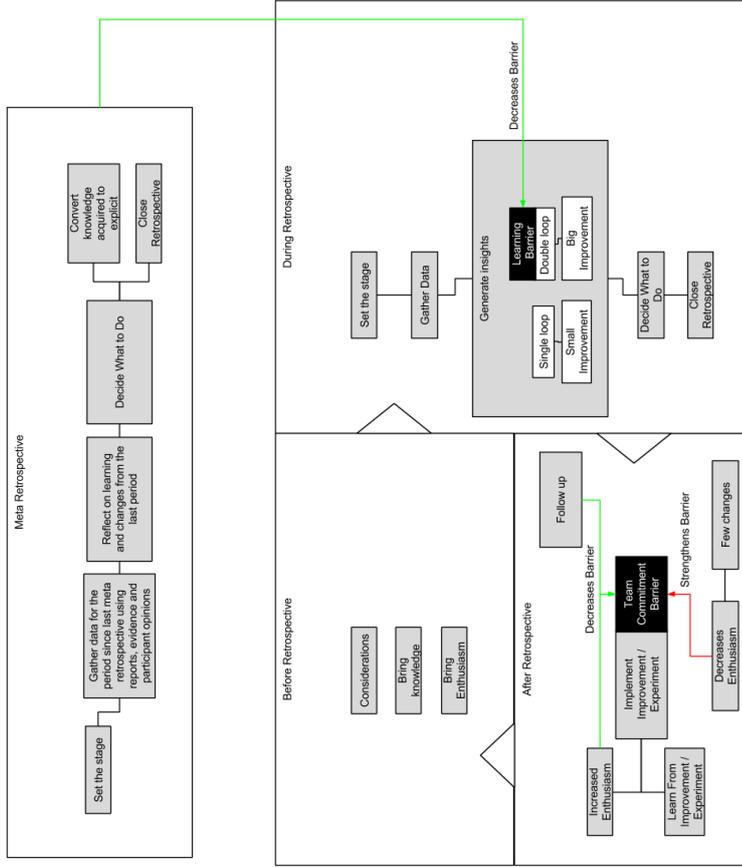


Figure 4.2: Retrospective state after introduction meta-retrospective

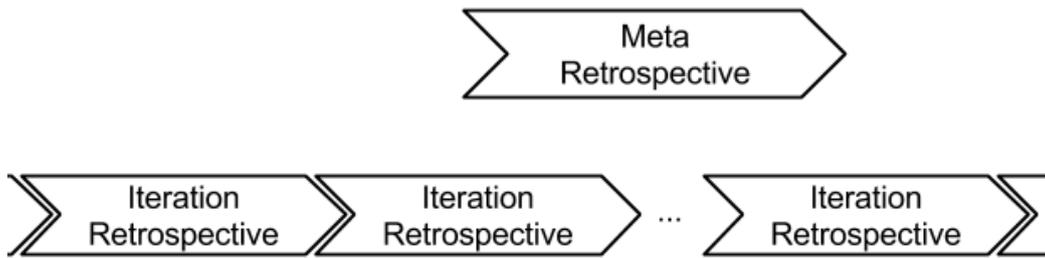


Figure 4.3: Visualization of meta-retrospective occurrence

4.3.3 Guidelines for Conducting Retrospectives

Through this study we have identified several characteristics that we believe will have positive impact on the conduction of the retrospective process in a modern team. These will be presented in a set of guidelines, followed by a short explanation of each point.

- External facilitating
- Implementation of actions
- Address actions to name and visualize
- Enthusiasm
- Trust
- Ownership
- Experimentation
- Reflection on own learning
- Regular retrospectives
- Measured varying of technique
- Find root causes
- Follow up implementations
- Reflect on how to conduct own retrospective
- Move beyond day-to-day decisions

External Facilitating As described in section 4.1.2 we observed multiple instances in which the use of an external facilitator was positive to the team. There are multiple approaches that could be used to arrange an external facilitator, an example would be switching SCRUM masters between teams in the company, another possibility would be using personnel from outside the company.

Implementation of Actions The ensuring implementation of actions appears critical for a team to have confidence and trust in the retrospective process. Therefore attempts to conduct a good retrospective should attempt to ensure that actions are completed, and that this success is acknowledged by the team. As seen in our work with team Zulu this is very possible and can lead to a positive effect in the team. Implementation of actions feeds the positive loop described in section 4.1.1.

Address Actions to Name and Visualize One step we observed that had a major impact on the implementation of actions was the follow-up protocol. In Table 3.8 we observe that the teams assigning name to an action are satisfied with their implementation of actions. Another positive step is the visualization of the actions.

Follow up Implementations A team should ensure that there is some sort of follow up that ensures that actions are implemented. For example this responsibility can fall to SCRUM master as seen in team Echo in section 3.1.2.

Trust The building of trust seems absolutely central to unlocking the potential of the retrospective session. Thus a team looking to improve should investigate possibilities relating to increasing the sense of trust between team members. One example of the positive impact this can have on a team can be seen in Table 3.1.2, where team Delta improves team dynamic through a discussion enabled by the high level of trust in the team.

Ownership The generation of ownership has a very positive effect on the team's attitude towards retrospective. This is discussed in section 4.1.1. In order to facilitate this ownership a team should be empowered in their decision making, leading to them shaping their own work processes into something they own.

Experimentation A retrospective should be open to allow experimentation in the following development iteration. We have observed this leads to several benefits for the teams in our analysis, this is described in more detail in section 4.1.2.

Reflection on Own Learning Through our work we discovered that team level reflection on learning processes has little or no presence. The work done with team Zulu indicates that this reflection can provide benefits, and we elaborate our ideas for this characteristic in subsection 4.3.2

Regular Retrospective Teams without regular retrospectives can be negatively affected in several ways, as described in section 3.1.4. Making sure there is not too long between retrospectives decreases the effect of memory bias as described in section 4.2.1.

Measured Varying of Technique We got a wide range of answers when we investigated the use of techniques in a retrospective, as seen in section 3.1.3. Thus it seems advisable for a team to experiment with using different techniques in order to discover what works for them. One should be mindful of too much variation, as too much variation could steer focus over to the technique instead of the retrospective session

Reflect on How to Conduct Own Retrospective A team can improve by reflecting how they can benefit from established theory, for example considering how they compare to Dingsøy's approach elaborated on in section 1.3.4.

Find Root Causes As described in subsection 4.2.3 some teams can benefit from increasing their focus on double-loop learning. This can lead to more effective decision making as underlying issues are targeted instead of superficial ones. Some techniques observed to find root causes are described in section 3.2.2.

Move Beyond Day-to-Day Decisions Our work has seen that a retrospective have a potential to move beyond operational decisions, this is for example shown in section 3.1.2. A team mindful of this potential is better suited to improve from their retrospective.

4.4 Limitations

There are several limitations to this study. First and foremost is that the teams participating in the study is only recruited from Scandinavia, which could mean that the results of this study might not reflect for teams other places in the world.

Also only teams that were willing to participate could be studied. This could be a selection bias that potentially might impact the results.

Another limitation of this study could be the small sample size as we only have had one depth study and only seven interviews in our breadth study. This could mean that the results of this study do not necessarily reflect the general mass of practitioners.

The final limitation is the researchers own inexperience with this kind of research. This is the first time both researchers have conducted such a study, which could mean that some bias might have occurred.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Throughout this thesis we have conducted an empirical study, of mature agile development teams. Investigating the outcome returned from the retrospective in terms of organizational learning and retrospective characteristics. Thereby answering Dingsøy and Dybå's call [12] for empirical studies into mature agile development teams. For the practitioners we have investigated the outcome of the characteristics and proposed a set of guidelines which could help improve their practice.

For the current characteristics of today's retrospective we have seen that the outcome of the practice is improvement opportunities and learning which could result in improved efficiency, increased enthusiasm and adaptation of work-processes, work environment, and product quality. For the processes of the practice we have identified a feedback loop with a barrier for implementation of improvement opportunities, depending on team commitment for implementation and enthusiasm for the retrospective.

- Retrospectives allow teams to improve their current work-practices through learning, team commitment and investigation of past development phases.

The studies revealed that today's retrospective practices is a learning practice where teams are able to test their current work-practices, learn from them and improve from them. Which means that agile development teams that practice retrospectives is approximating an organizational learning II system as described by Argyris and Schön [6]. Where most of the governing values are already introduced to current practice. We have seen that single-loop learning is the primary learning occurring, even though double-loop learning

is the expected outcome from organizational learning II systems. We have identified one barrier to double-loop learning which consists of several factors. Through our depth study we have seen that reflection on ones own learning helps give focus to the retrospective and lower this barrier and we have proposed a method to help achieve this.

- Learning in the present retrospective practices are primary single-loop learning, however the learning environment could facilitate double-loop learning.

Previous critique [11] have stated that the retrospective does not provide any changes to the work environment and we have observed that this could be the case if the team are not able to overcome the team commitment barrier for implementation of improvement opportunities. Further we would say that the main problem for current retrospective practice, along with the team commitment barrier, is the development team's inability to reflect on their own learning and learning processes.

Finally todays retrospective practices provide agile development teams the ability to adapt their current work-practices and enables them to learn from past development iterations and thus provide the means for identifying improvement opportunities and improve from them.

5.2 Future Work

For future work we would recommend conducting similar studies in other parts of the world to increase sample size and further verify the results of this thesis. Research investigating the method of meta-retrospectives and its value would also help verify this research and hopefully provide some value for practitioners. Other future work would be to determine the value of double-loop learning to help identify which issues should be identified down to their underlying influences.

If someone chooses to perform similar studies they should have the following criteria in mind. First of all it should be an empirical study [12]. If a breadth study is to be conducted we would recommend having many teams from different types of projects and other parts of the world than Scandinavia. Teams participating in large scale agile projects, teams that are distributed across both small and large distances, teams who do continuous delivery, teams who do cyclic delivery, teams with consistent availability of members and the opposite are some examples of teams and projects. In general to get a wide understanding of how different types of teams are practicing retrospectives. For a depth study we would recommend working with teams

that are open in both communication and information sharing. As this will form the cornerstone of the study. Further both the team and the researchers should both agree to the analysis method used. If a similar analysis like our content analysis should be conducted we would recommend the researchers to add those categories we have missed, like flow, work-environment and hotfix which in hindsight were missed by the authors.

Finally we did some work that examined the relationship between shared mental models and the agile retrospective, but due to time constraints we were not able to perform this examination to what we think is its full potential. However what we did manage to investigate some of it and this is described in appendix . This remains an area for potential future work. Such a study could potentially lead to a deeper understanding of how tacit knowledge exists within a retrospective team and could possibly be used to improve the retrospective practice even further.

Bibliography

- [1] Agile Manifesto. <http://www.agilemanifesto.org/principles.html>. Accessed: 2015-02-04.
- [2] Merriam Webster Dictionary Website. <http://www.merriam-webster.com/dictionary>. Accessed: 2015-02-04.
- [3] Retromat. <http://www.plans-for-retrospectives.com>. Accessed: 2015-05-06.
- [4] Elizabeth Bjarnason and Björn Regnell. Evidence-Based Timelines for Agile Project Retrospectives – A Method Proposal. In *XP 2012*, 2012.
- [5] 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, 2009.
- [6] Argyris C. and D.A. Schön. *Organizational Learning II: Theory, Method and Practice*. Addison-Wesley Publishing Company, 1996.
- [7] E. Derby and D. Larsen. *Agile Retrospectives: Making Good Teams Great!* Pragmatic Bookshelf, 2006.
- [8] Torgeir Dingsøy. Postmortem reviews: purpose and approaches in software engineering. *Information and software technology*, 47, 2004.
- [9] Torgeir Dingsøy, N.B. Moe, Tor. Stålhane, and Geir Kjetil. Hanssen. Post mortem - an assessment of two approaches. In R. Conradi and A.I. Wang, editors, *ESERNET 2001-2003*. Springer-Verlag Berling Heidelberg, Berlin, 2003.
- [10] Torgeir Dingsøy, N.B. Moe, Tor. Stålhane, and Joost. Schalken. Organizational learning through project postmortem reviews - an explorative study. In *EuroSPI2007*, 2007.

- [11] M. Drury, K. Conboy, and K. Power. Obstacles to decision making in agile software development teams. *The journal of systems and software*, 85, 2012.
- [12] T. Dybå and T. Dingsøy. Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50, 2008.
- [13] B. Dølvik and A.M. Stålesen. A Review of the Agile Retrospective. <https://github.com/AgileMasters/Report/blob/master/Report.pdf>.
- [14] T.E. Fægri. *Collaborative learning in software development*. PhD thesis, NTNU, 2012.
- [15] Geir Kjetil Hanssen, Nils Brede Moe, Torgeir Dingsøy, and Tor Stålhane. Post Mortem – An Assessment of Two Approaches. *ES-ERNET 2001-2003*, pages 129–141, 2003.
- [16] A. Keegan and J.R. Turner. Quantity versus quality in project based learning practices. *Management Learning*, 32, 2000.
- [17] Fumihiko Kinoshita. Practices of an Agile Team. *Agile 2008 Conference*, pages 373–377, 2008.
- [18] K. Krippendorff. *Content Analysis An Introduction to Its Methodology*. Sage Publications, 2004.
- [19] Barbara Levitt and James G. March. Organizational learning. *Annual Review of Sociology*, 14, 1988.
- [20] Michael Maham. Planning and Facilitating Release Retrospectives. *Agile 2008 Conference*, pages 176–180, 2008.
- [21] J. McAvoy and T. Butler. The impact of the abilene paradox on double-loop learning in an agile team. *Information and Software technology*, 47, 2007.
- [22] N.B. Moe, A. Aurum, and T. Dybå. Challenges of shared decision making: A multiple case study of agile software development. *Information and software technology*, 54, 2011.
- [23] Nils Brede Moe, Torgeir Dingsøy, and øystein Nytrø. Augmenting Experience Reports with Lightweight Postmortem Reviews. *PROFES 2001*, pages 167–181, 2001.

- [24] Michael D Myers and Michael Newman. The qualitative interview in is research: Examning the craft. *Information and Organization*, 26, 2007.
- [25] K. Nakakoji, Y. Ye, and Yamamoto Y. Supporting expertise communication in developer-centered collaborative software development environments. In I Mistrík, J. Grundy, A. Van der Hoek, and J. Whitehead, editors, *Collaborative Software Engineering*. Springer-Verlag Berlin Heidelberg, Berlin, 2010.
- [26] Stacie Petter and Xiaodan Yu. Understanding agile software development practices using shared mental models theory. *Information and Software Technology*, 56, 2013.
- [27] Viktoria G Stray, Nils B Moe, and Aybüke Aurum. In *Euromicro Conference on Software Engineering and Advanced Applications*, pages 274–281, 2012.
- [28] Bjørnar Tessem. Individual empovement of agile and non-agile software developers in small teams. *Information and software technology*, 56, 2014.
- [29] Paul Tosey, Max Visser, and Mark NK Saunders. The origins and conceptualizations of 'triple-loop' learning: A critical review. *Management Learning*, 43, 2011.
- [30] Max Visser. Gregory bateson on deuterio-learning and double bind: A brief conceptual history. 39, 2003.
- [31] M. von Zedwitz. Organizational learning through post-project reviews in r&d. *R&D Management*, 32, 2002.

Appendices

Appendix A

Interview Guide

Purpose: This interview is designed to be conducted in 45 minutes, it will focus on organization learning and retrospectives in a project or business. The interview will be recorded and transcribed.

Participants: 2 interviewers (Alf Magnus Stålesen and Bjørn Dølvik) as well as a relevant representative from the team in question. For example the SCRUM master of the team, or the person in charge of holding the retrospectives.

General overview

1. What is your role?
2. Could you describe your team?
3. How does the team conduct its retrospective meetings?
 - (a) Frequency
 - (b) Duration
4. What are the positive aspects that the retrospectives bring to your team?
5. What are the negative aspects that the retrospectives bring to your team?
6. What steps are taken to enforce or follow up on decisions made during the retrospective?
7. Do you have any examples on issues or actions that can hinder the function of the retrospective?

8. What kind of methods are you utilizing, and how would you evaluate them?
9. Do you take any steps to encourage learning? Bonuses etc?
10. As a SCRUM master do you feel like a facilitator or a leader? How does this impact the retrospective?

Organization learning questions

11. Has something that has come up during a retrospective that has changed how you work or think?
12. Do you use any special information to assist in decisions? What kind of information is this and what value does it have? Tools etc.
13. Do you solve problems during retrospectives or do you take steps to investigate the root cause of the problem? Do you have any examples of root cause identifying?
14. Are there any obstacles which makes it difficult to take decision and prohibits learning?
 - (a) If so which?
15. Does the team reflect on how you learn?
 - (a) If so how is this done? And is this spread further into the organization.
16. How do you learn from retrospectives?

Team dynamics questions

17. What is it about your team, that makes you able to learn through the retrospective?
18. What in your team could be improved to further enhance the learning through retrospectives?
19. Do you have any experiences where norms and cultural differences have an impact?

20. Which properties in the team do you see as positive or negative? How do they influence the retrospective?
- (a) Are there someone in the team who uses a lot of the time during the retrospective? Why and how does it influence the retrospective?
 - (b) Are the someone in team who rarely contributes during the retrospective? Why and how does it influence the retrospective?

Anything else

21. Do you have any examples of major breakthroughs or development that has happened through team learning?
22. Is there anything we haven't covered, that you feel is important for us to know that is related to how your team work with retrospectives and the knowledge learned from it?

Appendix B

Pilot Analysis

Settling on tabulations as our means of content analysis the different categories had to be determined. We performed a pilot study. The pilot study was conducted in order to investigate the potential of analyzing a set of the 77 retrospective reports. The pilot study was limited to 11 reports, where we picked every 7th retrospective chronologically. This distribution was chosen in order to get an even spread to represent the whole set, as well as keep the size manageable for the short preliminary study.

The pilot study analysis lasted for one week, and included agreeing on the parameters and methods for the study, as well as a short workshop session where the results were presented in front of a group of fellow researchers. This workshop session consisted of a short presentation of the findings of the study. After the presentation we had a brainstorming session where we received feedback on potential improvements, as well as general impressions.

Single-loop vs double-loop

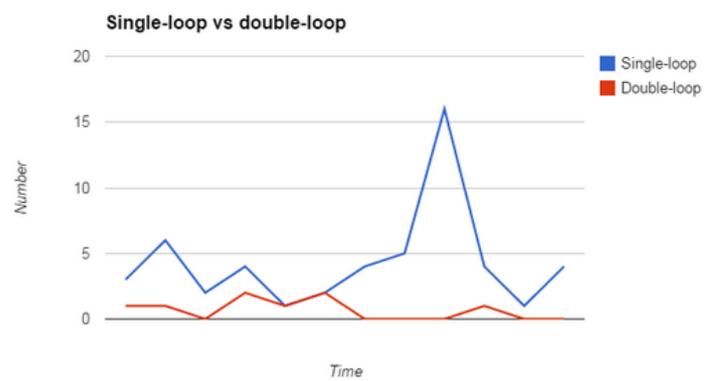


Figure B.1: An example of a slide from the pilot study presentation

Appendix C

Content Analysis Categories

The results of our pilot analysis gave us an extended set of categories that we will now describe further. Mainly we found six main themes that we could derive categories from. The six themes were: Nature, Context, Decision Making, Organizational Learning, Development and Management. We will describe each of these themes and their set of categories further in the sections below.

C.1 Nature

The nature of the action is the first theme that the content analysis is going to inspect. We define the nature of the action as how the origins of the action began. Did they come from a problem that occurred during the iteration or is it a continuation of something that has been working well in the past. Through our analysis we will try to understand the origins of the actions and classify them either as positive, negative or undefined. We define our classifications below.

Positive Positive actions is those actions where the origins of the action is in a positive context. If the action represents a current good working practice being continued, or something uncommon happened that gave positive results, it would be classified as positive.

Negative The negative actions are those actions that has its origins from a problem or abnormal issue resulting in negative results. If an action is a result of a problem or abnormal issue it is classified as an negative action.

Undefined In the cases where it is unclear whether the origins of the action is positive or negative we classify the action as undefined. Such occurrences can be a result of missing context or actions that seem to have neither positive issues or negative issues as its origin.

C.2 Context

The context surrounding the action will be analyzed. The context off the action is based on the underlying issue that leads to the needed action. We divide the issues into three main categories, technical, process and undefined.

Technical A technical issue can be a issue relating to technical competence, bugs or problems.

Process A process issue stems from a problem with a process, or potential for improvements in the existing processes. This can for example relate to communication between colleagues or work scheduling.

Undefined An undefined issue might not have any clear origin, or might be too loosely described to be classified easily.

C.3 Decision Making

Rational decision making is how decision makers should think and should act based on coherence and rationality, according to M. Drury et al. [11]. N.B. Moe et al. [22] appends bounded rational decision-making as the means of understanding how decision making is made in non-routine activities. N.B. Moe et al. reasons that both are needed when analyzing decision-making in an agile context:

Software development involves both routine and non-routine activities. Hence, it makes sense to use both rational and bounded rational decision making theories when explaining decisions in software development processes.

Drury et al. distinguishes between two types of decisions being made in an organization, the strategic and the tactical. Moe et al. distinguishes between three types: Strategic, tactical and operational. We will use the three-type model. Each decision type will be described in the following paragraphs.

Strategic A strategic decision is a wide ranging decision dealing with multiple or sizable issues, often causing major changes and have a long term impact. Moe et al. describes the strategic decisions as the following:

Strategic decisions are related to organizational goals and objectives. The information concerning such decisions is usually incomplete and the decision-making process may extend over a considerable period of time.

The actions that are categorized as strategic is the ones that proposes changes that have a long term impact or proposes changes that are related to the organizational goals.

Tactical A tactical decision is smaller than a strategic decision it seeks to deal with the distribution and use of resources available to the team. Moe et al. described it as:

Tactical decisions are related to identification and use of resources.

All actions that specifically proposes to identification of resources or proposes changes to how resources are spent will be classified as tactical.

Operational Moe et al. describes an operational decision as:

Operational decisions deal with ensuring effectiveness of day-to-day operations within the organization.

Every action that are restricted to only day-to-day operations will be classified as operational decisions. They might be quick fixes that solve a single problem.

Undefined An undefined decision might be difficult to categorize because of a lack of context or an unclear description.

C.4 Organizational Learning

Organizational learning is a process where an organization takes steps to improve its current work environments by reacting to issues that arise. These steps can be varied, and we divide them into single-loop, double-loop and undefined. Retrospectives have a central role in organizational learning, as described by Dingsøyr [10]

Single-loop A single-loop action is an action designed to change or tune a process in order to improve it. The action does not seek to address underlying problems, and are a single-feedback loop from observing an issue to making a change. A retrospective can facilitate single-loop learning where the project team uses the input during the retrospective to make adjustments to their current work [8]

Double-loop A double-loop action is designed to solve an issue, as well as address the underlying cause of the issue. This requires an understanding of the underlying issue and the nature of its influence. Double-loop decisions can be facilitated through a retrospective, often these are as a result of a more drastic need for change and an understanding of the underlying problem. [8]

Undefined An action might not be clearly described, or the nature of the action can not be interpreted. We will classify these actions as undefined.

C.5 Development

An issue that is related to the development of the product is included in this category. The development process is the actual work performed to create the product, as well as processes directly related to this work. The different categories included in development were chosen after discussing with advisors, as well as personal experience from the writers

Development Development includes the writing of code, specifying requirements, construction the system architecture and other aspects of designing of the system.

Planning An action that is categorized as planning is when the action is suggesting changes to planning in future iterations or is a result of an issue occurring in a former iteration. Estimations, task-prioritization, scheduling and etc. all goes under the planning category.

Testing Issues related to the testing of a product, this includes unit, module and system testing. Issues related to testing can also be communication between testers and developers.

Documentation Action that are categorized as documentation is those that are related to the documentation part of developing a system. Typical actions are those that describe or propose changes to documentation practices. This can also include tutorials that explains the product and improves usability.

Builds During development of a software system building the software system can be a tiresome task. The actions that are categorized as builds are those who proposes practices that changes the current build practices.

Release When a system feature or a part of a system is finished, usually at the end of an iteration, one deploy the new part of the system into production, available for the users. For the actions that categorized as release, the action describe some aspect of the release practice and either proposes changes or suggest new practices.

Business Business development is a critical part for an organization to create profit. Software system often can save costs and create new business opportunities and as a part of a software development team one also has to consider business perspectives. We categorize all actions that are related to business development or proposes some business related changes as business.

Undefined If an action is neither of the development phases described above we classify it as undefined.

C.6 Collaboration

One of the aspects during a software developing process is collaboration. Fægri [14] describes collaboration as the following:

Collaboration is a key aspect of software development. Collaboration allows groups of software practitioners to deal with uncertainty, complexity and interdependence. And in dealing with these challenges, the group demonstrates its collective problem-solving ability.

Through our pilot analysis we registered several activities that are related to collaboration. Communication, leadership, competence, external relations and planning all belongs beneath the collaboration banner, and we describe in detail how we classify each of them below.

Communication Communication is a widely used word and concept, but it rarely is defined. By using Merriam-Webster dictionary [2] we found a definition that can serve as a clarification:

The act or process of using words, sounds, signs, or behaviors to express or exchange information or to express your ideas, thoughts, feelings, etc., to someone else.

Nakakoji et al. [25] distinguishes between two types of communication related to software development, coordination communication and expertise communication. The first one being the process of coordinating the development activities and the last one being when a developer obtain some information regarding a software artifact, either through code comments, wikis or other means. We are however not going to distinguish between these two communication types in our content analysis as we believe the two differentiations is covered by the context of the action as described in section C.2. For our content analysis we are simply going to count every instance of communication for every action that is related to communication between team-members regardless if it is through text, speech or other means of communication.

Leadership As is the case with communication, leadership is also widely used concept, that is rarely specified. Again we turn to Merriam-Webster [2] for a definition:

The power or ability to lead other people.

Agile development teams is often self-organized as it is one of the principles in the agile manifesto [1]. This can result in no clear leadership. For our content analysis we are going to consider decisions and guidelines set by the group itself as leadership activities. We categorize actions as leadership if they somehow suggests changes to leadership or if the actions is a result of some leadership related issue.

Competence We define competence as the ability to perform a certain task in a adequate quality. Each member within an agile team has their own set of knowledge that they use to solve different tasks. If any issues arises and the group is lacking the knowledge to counter it we categorize the action created to resolve it as an competence action. An example could be issue of lacking the knowledge resolving a database error and the action is to send one developer at a database course. This action would then be categorized as an competence action.

External Relations By the category external relations we mean if the team has any actions that is a result of issues arising from external factors or the actions that team is creating to inflict some external factors. Example of external relations can be customer relations, communication with server maintenance team or other development teams.

Undefined For those actions that the origin is unclear and the goal is not related to any of the collaboration categories we categorize them as undefined.

Appendix D

Shared Mental Models

The theory of shared mental models is from cognitive psychology, and can be used as a lens to evaluate agile development and methodology as described by Petter et al [26]. The concept is that a team has a shared mental model that is central to the mutual understanding between team members, and thus essential to project success. Without a good shared mental model a team is left with a poor understanding of the task at hand, as well as barriers for cooperation. Two metrics used to measure a team's shared mental model are "similarity" and "accuracy". Where "similarity" is the degree which the shared mental model is similar between team members, and "accuracy" is the degree which the shared mental model matches objective measures.

D.1 The stages of Shared Mental Model generating

Four different stages of building shared mental models are identified [26]. These are knowing, learning, understanding and executing. An overview of these stages is seen in Table D.1. Knowing is the stage where a team gets exposed to information relating to their project and project goals, at this stage team members are encouraged to share information between each other. The second stage is the learning stage, this stage consists processing the information gained in the knowing stage. The understanding stage is defined by reaching consensus and understanding the team member's individual views. Executing is the last stage, with a developed shared understanding the team is able to reach goal, at this stage a team responds to a situation based on the work done in the previous stages.

Table D.1: Four stages of mental model building

Stage	Description
Knowing	Information exposure and sharing
Learning	Information processing
Understanding	Consensus and common ground
Executing	Shared understanding and

D.2 Shared Mental Models and retrospectives

Petter et. al. [26] describe multiple agile development practices, but do not focus a lot on the agile retrospective. In the appendix of the article they give the following link between retrospectives and shared mental models.

Enhances the development of learning and understanding stages by facilitating the information sharing and integration. This practice also improves the teams' executing capability in the next sprint

The shared mental model practices that are involved in the retrospective are identified as self corrective, training and reflectivity. We discuss the relationship between our results, retrospectives and shared mental models in section D.3.

D.3 Shared mental models discussion

In D we describe the basics of the shared mental models theory and its relation to agile development. In this section we will see how agile retrospectives can impact the four stages of shared mental model generation described in section D.1. Our thoughts are a result of the analysis and interviews. It should be noted that the agile retrospective described in Petter et al. [26] confines the retrospective to being in the end of a sprint, while our work does not put the same confines on the retrospective.

Improves Shared Mental Model The knowing stage is not included in the domain of the sprint retrospective described by Petter et al. [26], in our work we found that the facet of team members sharing their individual knowledge, thereby updating the shared information. One example of this was found in team Echo, where the different sub teams would use the retrospective to share their discoveries and knowledge development, thus updating the meta-knowledge of the team as a whole.

The learning phase of the shared mental model is impacted by the agile retrospective by using techniques that facilitate the integration of the knowledge from the knowing stage. One example could be the use of evidence based time lines as described by Bjarnasson et al. [4], where the time line would work as an outline of the information received by the team. Another example is seen in the weekly retrospectives held by team Delta, where team members would continually update their colleagues on their work day, allowing for reflexivity. Another example would be the use of the “five times why”-technique as used by team Charlie and Echo, where both the possibility for reflexivity and self correction exists.

The understanding phase described by Petter et al. [26] is far reaching and includes many facets of team cooperation and some of them are described in section D.1. One part of the understanding phase we did not expect to see explicitly in our study was conflict resolution on a personnel level, and can be considered part of the conflict reconciliation and consensus building that is part of the understanding phase. Also part of the understanding phase is the practice of refining team communication and team processes which is a central component of the retrospective purpose. We observed every team discussing these topics during our interviews and analysis. Lastly, not included in the work of Petter et al. is the use of the retrospective as a tool for planing, and 24.6 percent of the actions analyzed in our work with team Zulu were deemed to have a planning component, as described in section 3.1.2. This planning component could be said to be increasing the similarity of the information between team members.

The execution is perhaps the phase most impacted by the retrospective, as the explicit actions decided during a retrospective almost always is intended to improve or refine the processes that in one way or another help them reach their goals. For example the information similarity generated in the understanding phase can lead to a quicker response time to new tasks. One typical example is the refining of the communication processes as seen in team Zulu, or the introduction of the bug-fix days described in section 3.1.2.

Shared Mental Models Practice In this section we discuss observations on the relationship between retrospectives and shared mental model practices.

Reflectivity is a central part of the learning process in the team. We have observed that many teams use the retrospective as a review tool of the last work period. We have observed very different practices when it comes to frequency as seen in Table 3.11, and thus the definition of a work period is different from the sprint retrospective definition from Petter et al. One

observation done by us is that very few teams practice reflectivity on the retrospective itself, or if they do they do not formally do it on the team level, as seen in team Charlie. In team Charlie the scrum master would discuss and reflect on the retrospective with other scrum masters and team leaders within the company, but it would not be brought back to the team. In some ways the analysis done with team Zulu together with the feedback sessions with them could potentially be considered an example of this kind of reflectivity. The interviews done with team Zulu's leader and SCRUM master suggests that both the team's mental model similarity and accuracy was improved through the analysis and reflection done.

The sprint retrospective defined by Petter et al. Does not include planning, but our work with team Zulu indicates that planning is an integral part of retrospective actions in some teams. This is discussed in section 3.1.2, this high degree of presence of planning was unexpected and more thought on the potential of mental model improvement through planning in retrospective seems interesting. For example the use consensus based approach used by team Golf in planning could potentially increase the similarity of the team's mental model.

Appendix E

Graphs of Results from Content Analysis

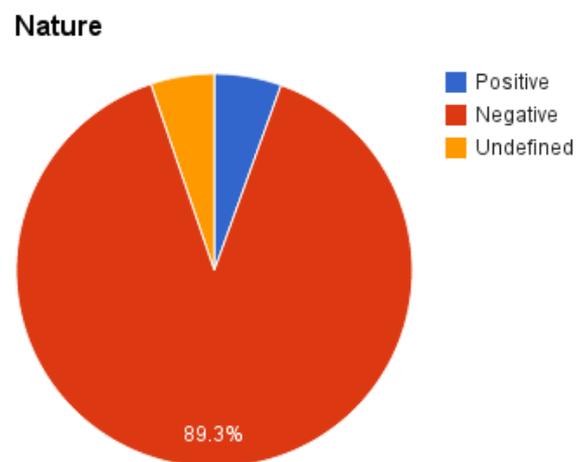


Figure E.1: The negative, positive and undefined distribution of all the actions.

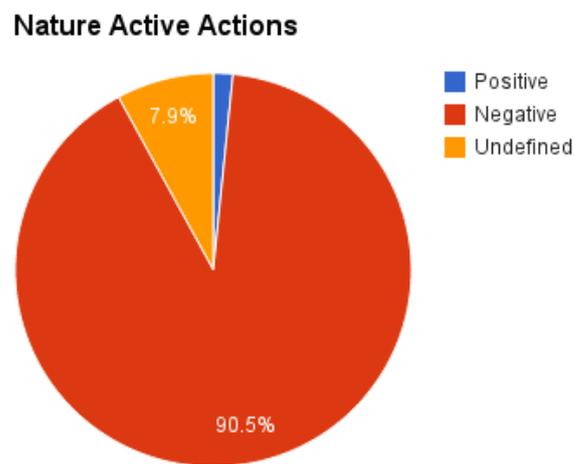


Figure E.2: The negative, positive and undefined distribution of the active actions.

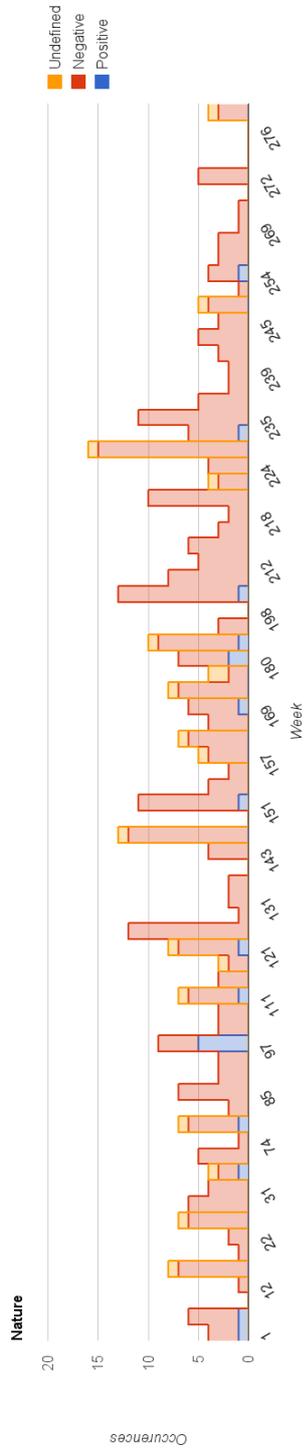


Figure E.3: The distribution of negative, positive and undefined actions across the timespan

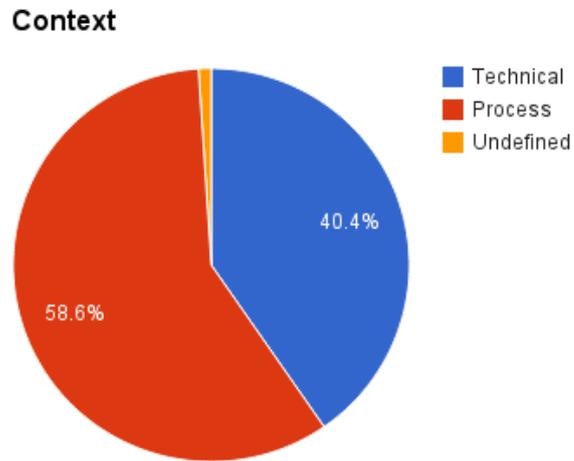


Figure E.4: The distribution of technical, process and undefined related actions.

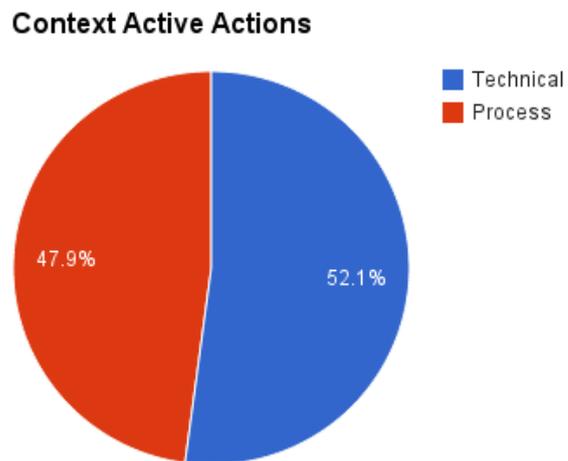


Figure E.5: The distribution of technical, process and undefined related actions over.

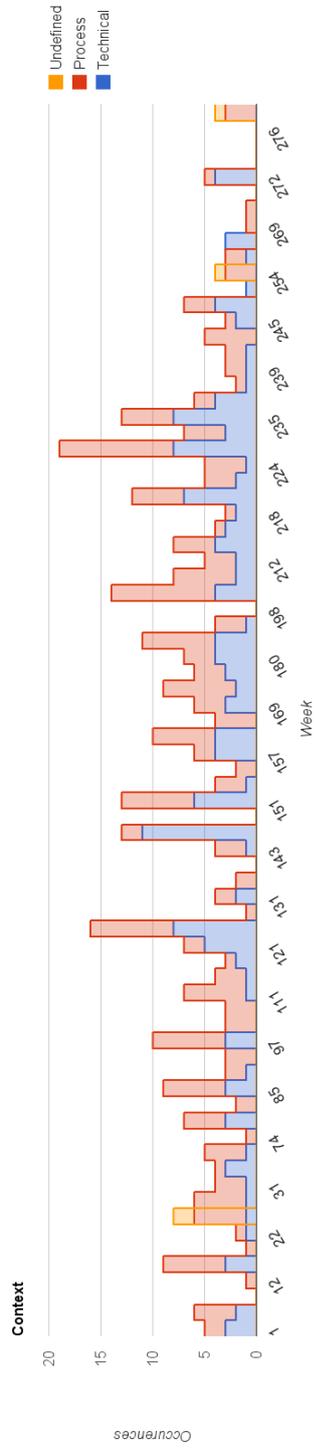


Figure E.6: The distribution of technical, process and undefined related actions across the timespan.

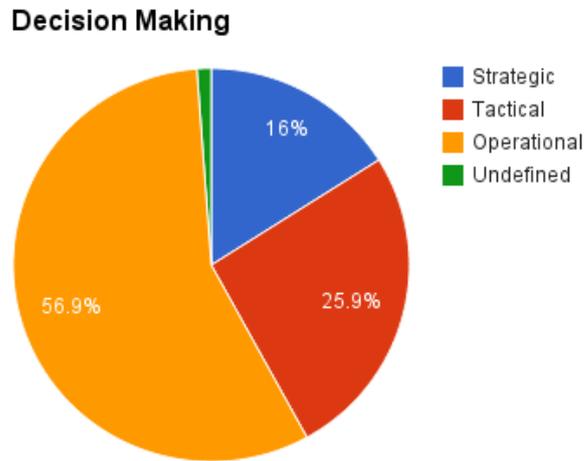


Figure E.7: The distribution of different decision making decisions as they occurred over all the actions.

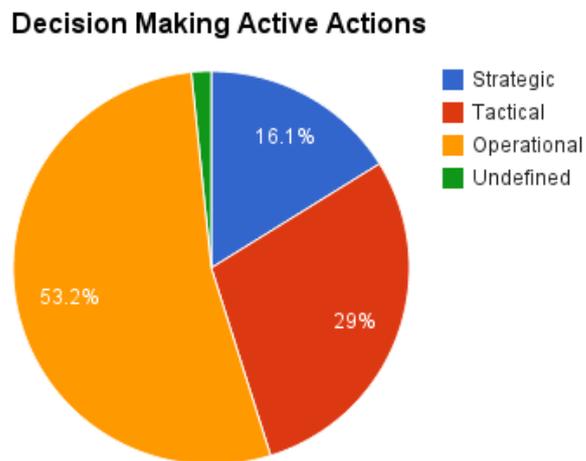


Figure E.8: The distribution of different decision making decisions as they occurred over the active actions.

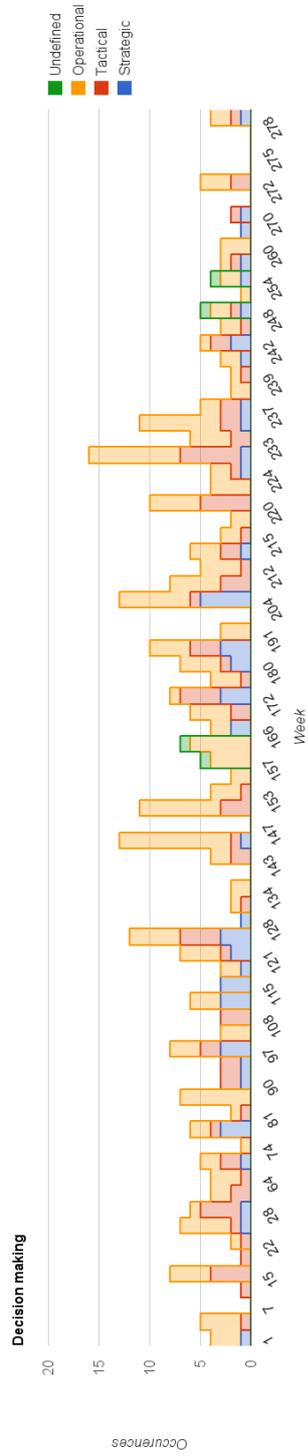


Figure E-9: A timeline showing the distribution of the different decision making decisions for all the actions.

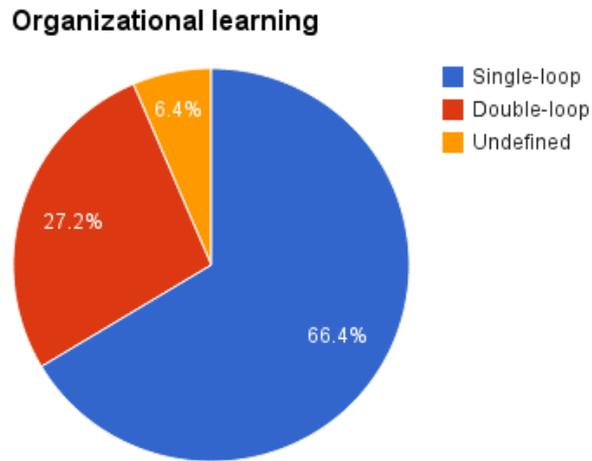


Figure E.10: The distribution of single-loop, double-loop and undefined for all the actions.

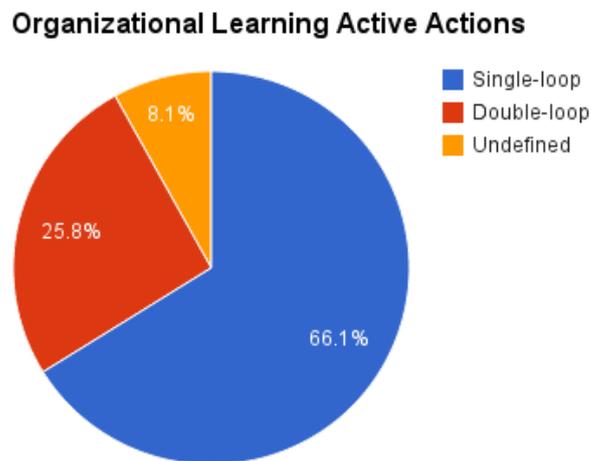


Figure E.11: The distribution of single-loop, double-loop and undefined for the active actions.

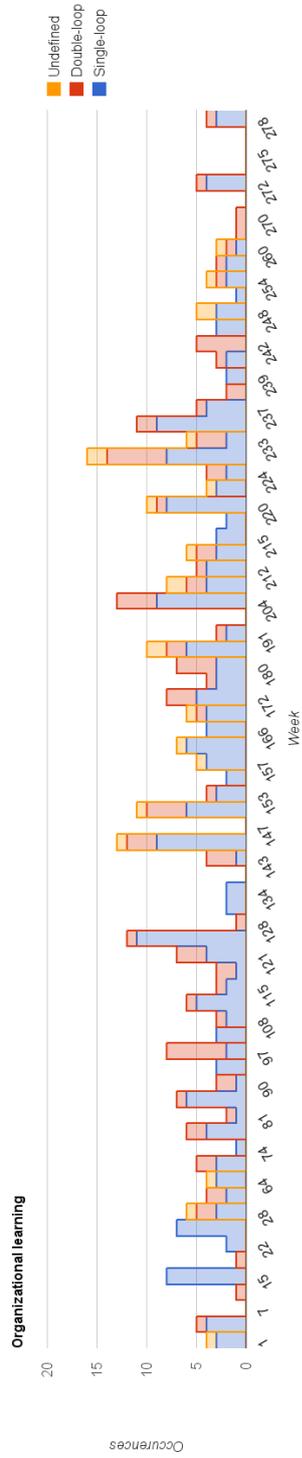


Figure E.12: Timeline showing the distribution of learning loops for the total actions.

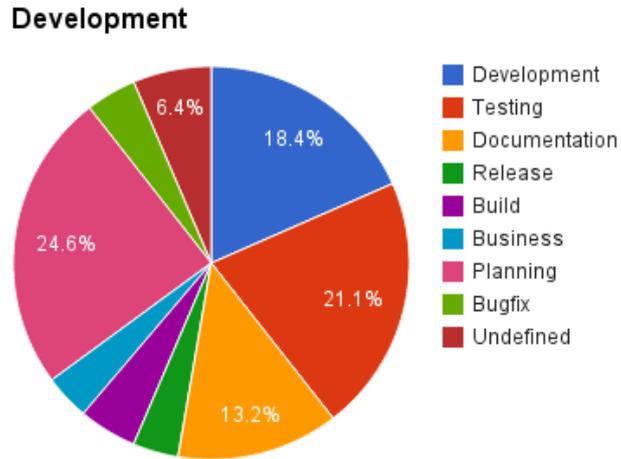


Figure E.13: The distribution of the different development phases for all the actions.

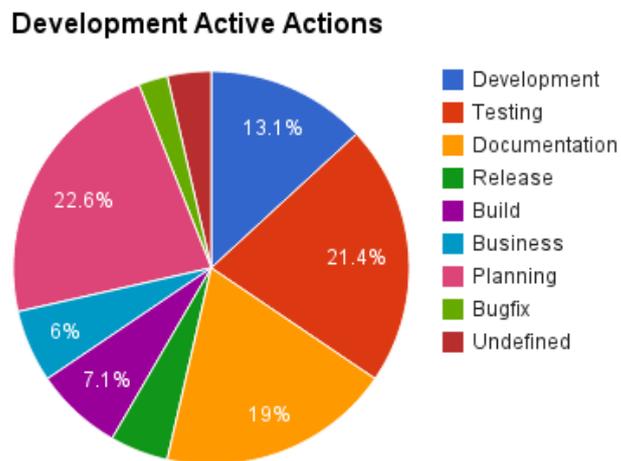


Figure E.14: The distribution of the different development phases for the active actions.

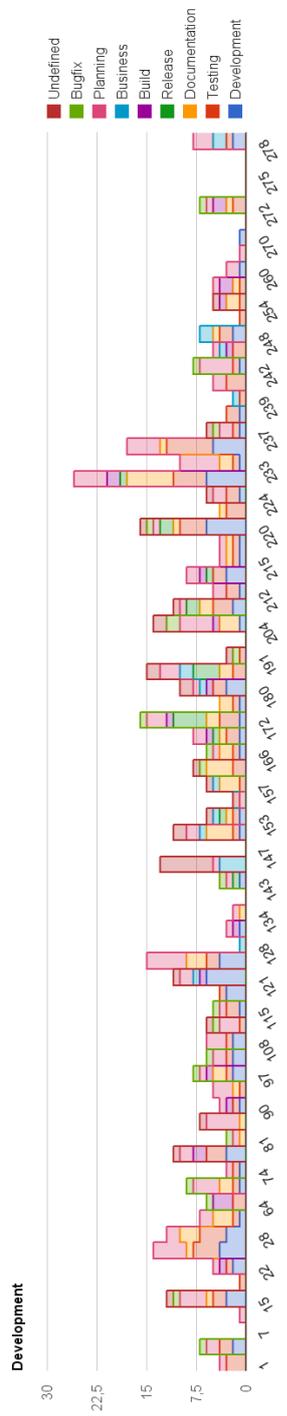


Figure E.15: Timeline showing the distribution of the different development phases over time.

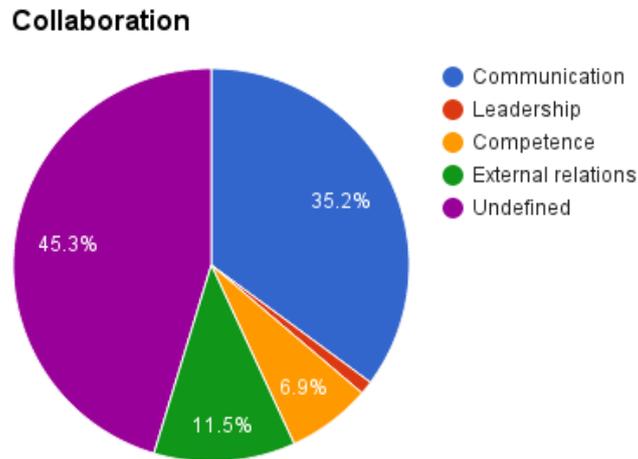


Figure E.16: The distribution of different collaboration categories for all the actions.

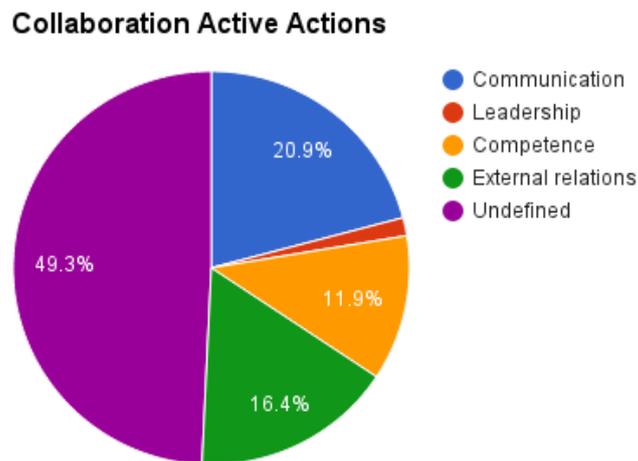


Figure E.17: The distribution of different collaboration categories for the active actions.

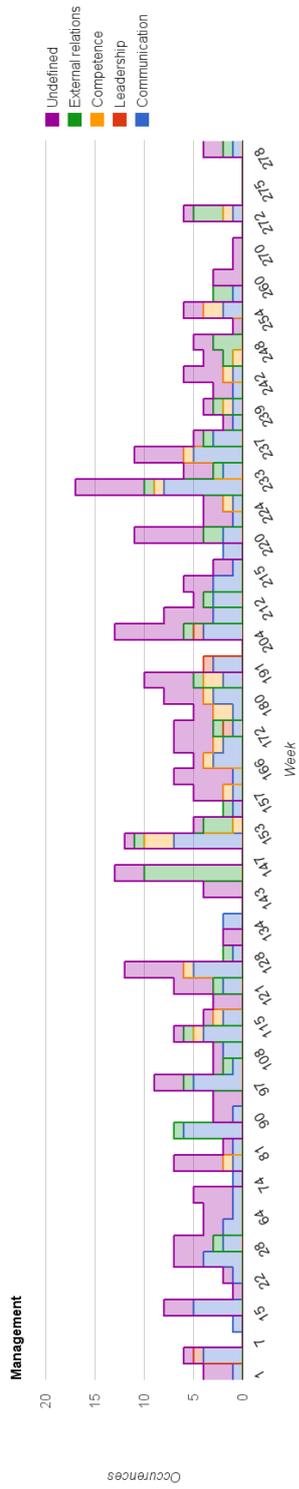


Figure E.18: Timeline showing the distribution of the different collaboration categories over time.