



TAMPEREEN TEKNILLINEN YLIOPISTO  
TAMPERE UNIVERSITY OF TECHNOLOGY

MIKKO AIRAKSINEN  
IMPROVING GAME DEVELOPMENT PROCESSES THROUGH  
RETROSPECTIVE MEETINGS

Master of Science thesis

Examiner: prof. Tommi Mikkonen  
The examiner and the topic were  
approved in the Faculty of Computing  
and Electrical Engineering  
Council meeting on 6.5.2015.

## ABSTRACT

### **MIKKO AIRAKSINEN: Improving game development processes through retrospective meetings**

Tampere University of Technology  
Master of Science Thesis, 58 pages

June 2015

Master's Degree Programme in Information Technology

Major: Software Engineering

Examiner: Professor Tommi Mikkonen

Keywords: game development, retrospective meeting, process improvement

Digital game industry is one of the fastest-growing industries, and game development professionals are in high demand all around the world. Despite this, studies on game development processes are relatively scarce on the scientific field. Game development is a subset of software development, but methodologies for traditional software production do not always fit with the unique challenges of making games.

This thesis evaluates using retrospective meetings from agile methodologies in a game development context. The ultimate purpose is to learn whether retrospective meetings alone, without other parts of an agile framework, can help game development teams improve their production methods and work processes.

Research was conducted as a case study in a game development company. The case study consisted of pre-study interviews, followed by several rounds of iteration retrospective meetings as well as two longer project retrospective meetings known as post-mortems. Data was collected from the retrospectives throughout the study. A post-study questionnaire with open and structured questions was also used to measure employees' attitudes towards retrospectives and postmortems at the end of the study.

Multiple data sources indicated that employees of the company felt that retrospectives were useful and that their use should be a permanent part of the company's processes. Retrospectives also demonstrably had concrete effects in the improvement of company level processes and conventions. Postmortem meetings were seen as potentially useful, but the structure used in them during the study was deemed subpar.

In conclusion, retrospectives had a positive effect on working processes and work environment. Their usage also does not have to be limited to software team context, as retrospectives with participants from many smaller teams produced concrete results. However, additional study is required to assess whether retrospectives are useful in a game development team context in the long term.

## TIIVISTELMÄ

**MIKKO AIRAKSINEN:** Pelikehityksen prosessien kehittäminen retrospektiivitapaamisten avulla  
Tampereen teknillinen yliopisto  
Diplomityö, 58 sivua  
Kesäkuu 2015  
Tietotekniikan diplomi-insinöörin tutkinto-ohjelma  
Pääaine: Ohjelmistotuotanto  
Tarkastaja: professori Tommi Mikkonen

Avainsanat: pelikehitys, retrospektiivi, prosessikehitys

Digitaalinen peliteollisuus on yksi nopeimmin kasvavia teollisuudenaloja ja pelikehityksen ammattilaisille on kysyntää kaikkialla maailmassa. Tämän huomioon ottaen pelikehityksen prosesseista tehty tieteellinen tutkimus on suhteellisen harvinaista. Pelikehitys on ohjelmistokehityksen osa-alue, mutta perinteiset ohjelmistotuotannon metodologiat eivät ole aina yhteensopivia siihen sen sisältämien erityishaasteiden vuoksi.

Tässä työssä arvioidaan agile-menetelmistä peräisin olevien retrospektiivitapaamisten käyttöä pelikehityksessä. Pääasiallinen tavoite on selvittää, voivatko retrospektiivit irrotettuna suuremmasta agile-kehiksestä auttaa pelinkehitystiimejä kehittämään omia tuotantometodejaan ja työprosessejaan.

Tutkimus toteutettiin case-tutkimuksena pelikehitysyrityksessä. Case-tutkimus koostui esihaastattelusta, useista kierroksista iteraatioretrospektiivitapaamisista sekä kahdesta laajemmasta projektitaso retrospektiivista eli postmortemista. Tietoa kerättiin tutkimuksen läpi retrospektiiveista tehtyinä muistiinpanoina. Tutkimuksen lopuksi yrityksen työntekijät myös täyttivät kyselyn, jonka tarkoituksena oli mitata työntekijöiden asenteita ja näkemyksiä retrospektiiveistä ja postmortemeista kohtaan.

Useammasta lähteestä kerätty tieto osoitti, että yrityksen työntekijät kokivat retrospektiivitapaamiset hyödyllisiksi ja niiden käytön tulisi heidän mielestään olla pysyvä osa yrityksen prosesseja. Retrospektiiveillä oli myös todistettavasti konkreettisia vaikutuksia yritystason käytäntöihin ja prosesseihin. Postmortem-tapaamiset koettiin potentiaalisesti hyödyllisiksi, mutta tutkimuksessa käytetyssä tapaamisen rakenteessa koettiin olevan paljon parantamisen varaa.

Johtopäätöksenä oli, että retrospektiiveillä oli positiivinen vaikutus työprosesseihin ja työympäristöön. Retrospektiivien käyttöä ei myöskään tutkimuksen perusteella tarvitse rajata ohjelmistotiimikontekstiin, sillä retrospektiivitapaamiset joihin osallistui työntekijöitä useista tiimeistä johtivat konkreettisiin tuloksiin. Lisätutkimus on kuitenkin tarpeellista ennen kuin voidaan tehdä vahvoja johtopäätöksiä siitä, ovatko retrospektiivit hyödyllisiä pitkällä tähtäimellä pelitiimikontekstissa.

## PREFACE

*“You know, sweetheart, if there's one thing I've learned, it's this: nobody knows what's gonna happen at the end of the line, so you might as well enjoy the trip.”*

- Manuel “Manny” Calavera, Grim Fandango

The work towards this thesis was done between January 2015 and May 2015. I wish to extend special thanks to my supervising Prof. Tommi Mikkonen for giving excellent advice and asking the tough questions necessary for me to see what needed to be seen.

As for people who have significantly influenced this work, I want to express my gratitude to my former supervisor Mikko Karvonen, who taught me a great deal about game production and development. Big thank you also to Jim Coplien for helping me understand what agile is actually about as well as giving valuable advice early in the work.

In addition, I want to thank Mika Rosendahl and Lasse Hirvonen for giving important feedback and ideas that enabled a crucial pivot early in the thesis. Thank you also to Heidi Jauni and Rasa Ozolina for the feedback on the quality of the text.

To Karoliina: The most loving thank you a man can give. You tirelessly read through a version after version of the thesis and helped me through my other studies. Quite frankly, I doubt I would have ever graduated if it was not for you.

To my parents Marja-Leena and Jukka and my brothers Matti and Jussi and the rest of my family: Thank you for shaping me to be the man I am today and being there for me always when I have needed you.

To my oldest and dearest friends Anssi, Valtteri, Hannu and Arttu: Thank you for your lasting friendship and support through these years. I wanted to take this opportunity to let you know how much your friendship means to me.

I started my trip seven years ago at the Tampere University of Technology, and the end of this particular line comes with the signing of this thesis. I have both learned a great deal as well as met and befriended many great people who have made it possible to enjoy the trip. The last thank you goes out to you all.

Tampere 19.5.2015

Mikko Airaksinen

## TABLE OF CONTENTS

1.	INTRODUCTION.....	1
2.	BACKGROUND.....	3
2.1	Retrospectives in agile methodology .....	3
2.1.1	History of agile methodologies .....	3
2.1.2	Definition of a retrospective.....	5
2.1.3	Structure of a retrospective .....	6
2.2	Game development .....	8
2.2.1	Roles in a game development team.....	8
2.2.2	High-level game development process .....	11
2.2.3	Agile game development.....	13
3.	STRUCTURE OF THE CASE STUDY .....	17
3.1	Research goal and questions .....	17
3.2	Research methods .....	17
3.2.1	Retrospective meetings .....	18
3.2.2	Postmortem meetings .....	23
3.2.3	Interviews and questionnaires .....	27
3.3	Execution and timeline of the study.....	28
3.4	Limitations .....	29
4.	GATHERED DATA .....	30
4.1	Participating personnel.....	30
4.2	Participating teams.....	32
4.3	Realized timeline of the case study.....	33
4.4	Retrospective journals.....	33
4.4.1	Team A.....	33
4.4.2	Team B.....	34
4.4.3	Team C.....	35
4.4.4	Team X.....	35
4.5	Postmortem meetings.....	36
4.5.1	Postmortem 1 .....	37
4.5.2	Postmortem 2 .....	40
5.	RESULTS.....	44
5.1	Findings from retrospectives.....	44
5.2	Findings from postmortems .....	45
5.3	Post-study questionnaire results.....	47
5.3.1	Open questions regarding work environment and processes .....	48
5.3.2	Questions regarding retrospectives .....	49
5.3.3	Questions regarding postmortems.....	51
5.4	Possible causes of error.....	52
6.	CONCLUSIONS.....	54
	REFERENCES .....	56

# 1. INTRODUCTION

Digital game industry is one of the fastest-growing industries and game development professionals are in high demand all around the world. Despite this, studies on game development processes are relatively scarce on the scientific field. Research on the area of traditional software development has been key for the emergence of many paradigms for software production, such as the increasingly popular agile methodologies. Game development is a subset of software development, but methodologies for traditional software production do not always fit with the unique challenges of making games.

This thesis evaluates using retrospective meetings from agile methodologies in a game development context. The ultimate purpose is to learn whether retrospective meetings alone, without other parts of an agile framework, can help game development teams improve their production methods and work processes.

The research was conducted in a game development company as a case study. The target company is a mid-sized company employing 18 game development professionals at the beginning of the study. The company requested a study to help with the unification of the used working processes and creation of best practices to enhance their productivity. The company had recently grown in size and what used to be a small startup-size group of people with loose or no processes was suddenly a multi-team mid-sized game production company. The processes or lack thereof that were adequate for a few people working together were no longer sufficient in the management of multiple game teams.

The goal of the study was to improve working processes and work environment in the target company iteratively using retrospective meetings and employee interviews. The purpose of these methods was to tap into the most valuable resource of a game company, its employees, in order to together create and implement processes and best practices that can benefit the entire company. Through the retrospectives, each team would enhance their own working methods and in doing so also generate knowledge for the company. In addition to team level processes, the company level processes would also be inspected to find ways to improve them.

Chapter 2 of this thesis covers the background theory for game development and agile methodologies, particularly that of the retrospective meetings. Based on that theory, the complete plan and schedule for the case study is presented in Chapter 3. The chapter also presents the templates for all methods of data collection, including interviews, questionnaires, retrospectives and postmortems. Chapter 4 reports all the data gathered through these methods. Summary of findings from pre-study interviews is presented,

followed by detailed journals from all retrospectives and postmortems. Chapter 5 summarizes the results from the retrospectives and presents the quantitative data from post-study survey and presents suggestions for best practices. The thesis concludes with Chapter 6 where conclusions from the results are presented along with recommendations for future study.

## 2. BACKGROUND

This chapter covers the necessary background theory for the thesis. Section 2.1 covers the history of agile and the theory behind the retrospectives often used as a part of agile frameworks. Section 2.2 describes the unique features, challenges and commonly used processes of game development.

### 2.1 Retrospectives in agile methodology

Agile methodologies are a very popular topic in modern software development. The industry as a whole has been gradually shifting away from older methodologies such as the waterfall model and towards the more modern agile counterparts. The old waterfall model relies on clear steps in the process and large amounts of detailed documentation (Royce, 1970). Conversely, as their name implies, agile methodologies are based on around frequent self-inspection and flexible adaptation (Coplien & Østergaard, 2009).

In agile methodologies, retrospective meetings serve a very important role in the "inspect and adapt" ideology. During a retrospective meeting, a software team works together to uncover issues in the work processes and work environment. Once they agree on the most important issues, the team forms and commits to a plan to resolve them. These meetings are repeated after every iteration, which usually translates into one or two weeks of calendar time. (Derby & Larsen, 2006)

#### 2.1.1 History of agile methodologies

The history of agile is surprisingly long and some of the principles predate the entire field of software engineering. The origins of agile are actually in Japanese car manufacturing. Honda and Toyota employed a production and management approach that the Western world knows as Lean. Lean focuses on reducing 'muda' (waste, unnecessary elements of work) and eliminating inconsistencies in systems. (Coplien & Østergaard, 2009)

The Lean principles are reflected in an article called "A New New Development Game" (Takeuchi & Nonaka, 1986), which emphasizes the importance of learning, flexibility and adaptation in production teams. The paper introduced the concept of Scrum in the context of a development team. The term Scrum originates from the sport of rugby, where it refers to the tightly packed formation where a team is huddled close together to work toward their common goal. Takeuchi & Nonaka used this metaphor to present a holistic approach where the team works as a unit, passing the metaphoric ball back and



forth. They wanted to make a case that the old, sequential production methods in which a project was clearly separated into distinct phases were simply not getting the job done any more. Teams had to work together, iterate, continuously learn and improve.

The timing of Takeuchi & Nonaka's article overlapped with a phase in software engineering where project sizes had started to increase dramatically. Managers were attempting to control the sizable projects by using management procedures designed for production lines (Coplien & Østergaard, 2009). The aforementioned sequential production was also used in software production, the most famous example being the waterfall model (Royce, 1970). As software development is a much more complex process than production lines, these management methods were seen as highly ineffective and wasteful by many. This led to various companies and researchers experimenting with applying the Lean principles to software in an attempt to replace the rigid production line based management principles (Coplien & Østergaard, 2009).

In 2001, 17 software professionals gathered in a ski resort for 3 days and together started a movement called Agile Alliance. In this meeting, the Agile Manifesto was also created (Fowler & Highsmith, 2001). It summarizes the core principles of agile software development as follows:

“We are uncovering better ways of developing software by doing it and helping others do it. We value:

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

That is, while there is value in the items on the right, we value the items on the left more.” (Fowler & Highsmith, 2001, p.2)

These principles are similar to many of those in the original article by Takeuchi & Nonaka. Currently, they are present in many agile methodologies, such as Scrum, Kanban and Extreme Programming.

While they have their differences, all agile methodologies share the same foundation. Agile is all about inspection and adaptation (Williams & Cockburn, 2003; Coplien & Østergaard, 2009). In other words, software development teams looking to adhere to agile should frequently inspect their processes and working methods in order to learn and adapt. Retrospectives are one of many possible tools that enable this and the following subsections will cover their usage.

## 2.1.2 Definition of a retrospective

Retrospectives are somewhat of an ambiguous topic in literature, and even the term itself has different meanings in different contexts. Some definitions for the word retrospective itself are:

*“directed to the past / looking or directed backward / retroactive, as a statute”*  
(Dictionary.com, 2015)

*“based on memory / relating to or being a study (as of a disease) that starts with the present condition of a population of individuals and collects data about their past history to explain their present condition”* (Merriam-Webster.com, 2015)

Retrospectives in agile context are quite similar to the second description, even though the “present condition” is not exactly a disease. Retrospectives are meetings where a team inspects many aspects of their work, such as processes, methods, teamwork and progress of the product being built. They are also a good moment to reflect on the work done, learn from mistakes and celebrate successes (Kerth, 2001). After a thorough self-inspection, the team adapts their working methods and processes to improve in one or more of these areas (Derby & Larsen, 2006; Keith, 2013). In a game development context, retrospectives are useful for regularly improving the methods in which the game development team adds value to the game and thus makes the game more attractive to the end user (Keith, 2013). The use of retrospectives as part of agile methodologies is a long-standing practice and their popularity among software developers is constantly growing (Coplien & Østergaard, 2009).

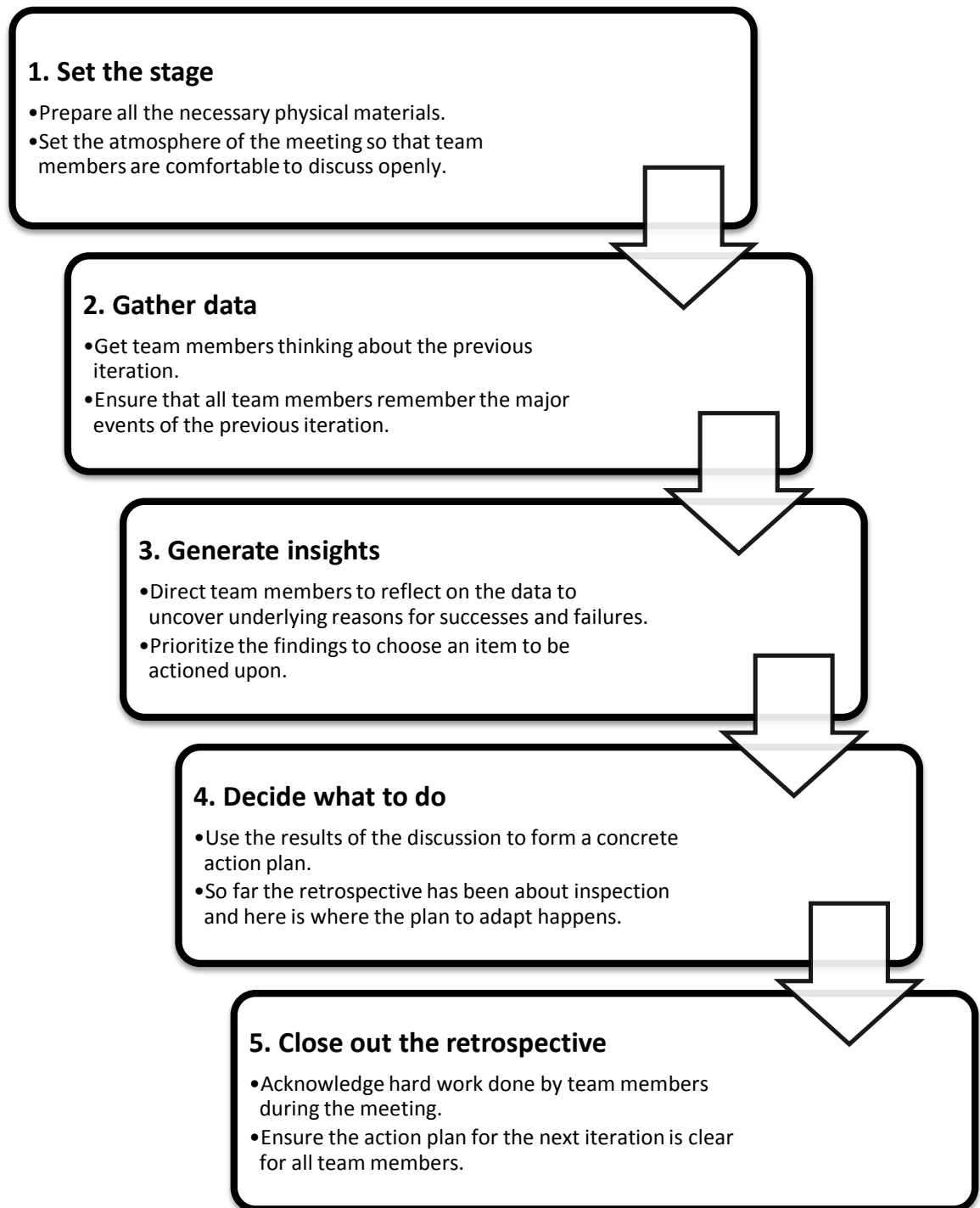
The timeframe that is being inspected in a retrospective can vary depending on the scope and purpose of the particular meeting. For example, a very famous book on the topic called *Project Retrospectives: A Handbook for Team Reviews* (Kerth, 2001) covers project retrospectives held at the end of a project. A project retrospective meeting is a very timely undertaking that the author suggests should last at least 3 days.

Retrospectives can also mean iteration retrospectives, which are approximately 1-2 hour meetings that take place at end of a sprint or an equivalent iteration (Derby & Larsen, 2006). Some authors make a distinction between the two, calling the longer interval sessions “Retrospectives” and iteration sessions “Check-Ups” (Coplien & Østergaard, 2009).

In this thesis, the term retrospective denotes an iteration retrospective. In the cases project retrospectives are discussed, they are called “postmortems”. This is done for clarity, even though the postmortem term is considered detrimental by some as it implies that the project is now dead and the lifeless body is being examined. (Kerth, 2001).

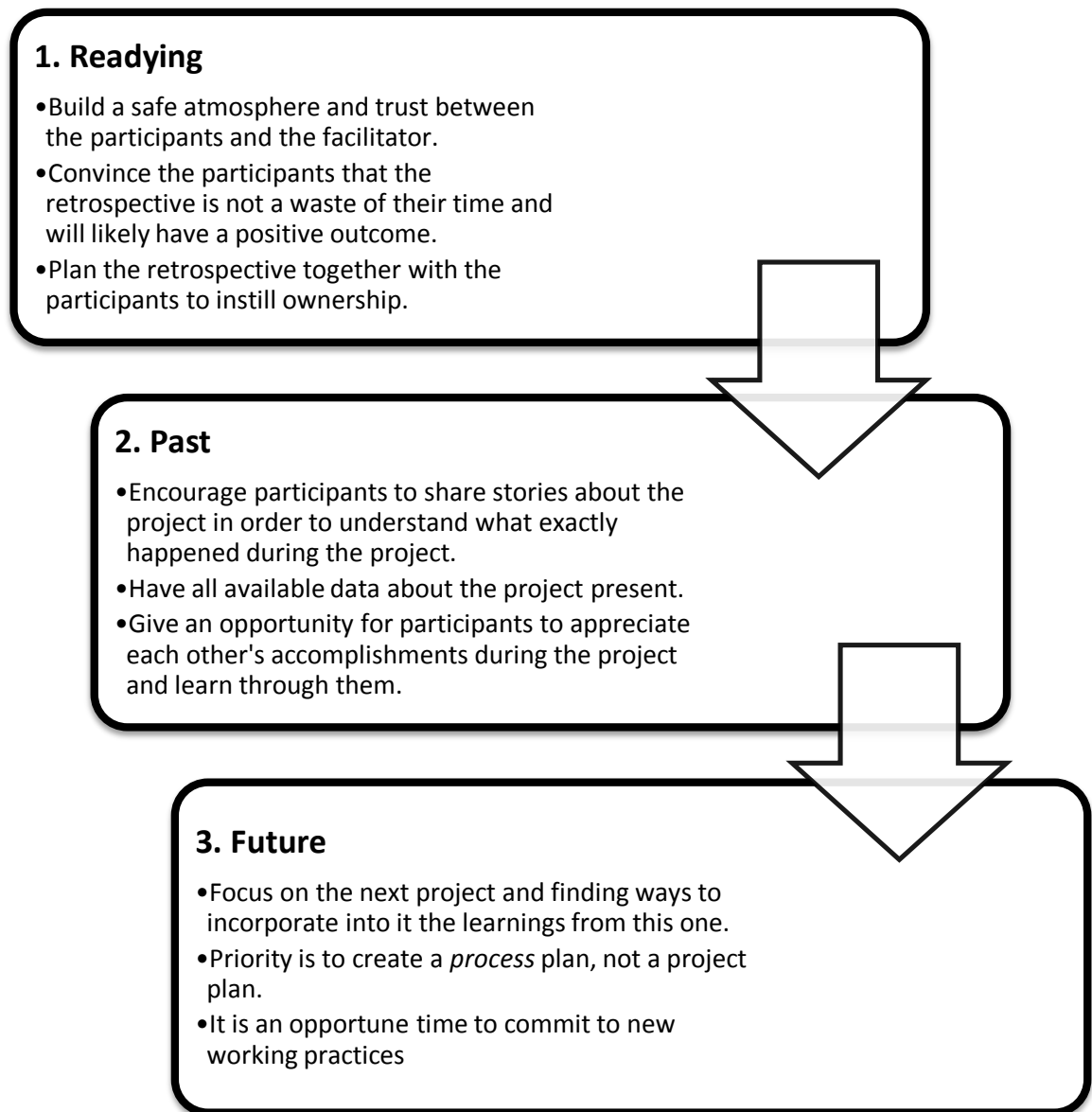
### 2.1.3 Structure of a retrospective

The structure of a retrospective meeting is not universal. However, there are some recommended best practices for it, such as the 5-phase model for an iteration retrospective (Derby & Larsen, 2006). The five phases the authors recommend and their descriptions are presented in Figure 2-1.



*Figure 2-1 Five phases of an iteration retrospective (adapted from (Derby & Larsen, 2006)).*

Comparable structure can be seen in a baseline retrospective called Everyday Retrospective (Agile Retrospective Resource Wiki, 2013). For a project retrospective, a three-phase structure is recommended (Kerth, 2001) and can be seen in Figure 2-2.



**Figure 2-2** Three phases of a project retrospective (adapted from (Kerth, 2001)).

Both figures show similar linear structure with phases for preparation, past and future. It is important to begin the session by setting participants in the correct state of mind for the exercise. After that is done, the participants will reflect on the past and try to gather all information and possible learnings. Towards the end of the session the focus should be on the future. Particularly important topics for future are ways to apply learnings from the past to the future work in order to either prevent earlier mistakes or repeat earlier successes. Concentrating only on mistakes is not necessarily an optimal approach, as many authors suggest that it is very important to also recognize and learn from successes (Kerth, 2001; Nolan, 1999).

A critical rule of a retrospective is that it must be a safe environment. Honesty is necessary to uncover problems in the processes and the work environment and thus honest discussion should be encouraged. Participants should be able to discuss their views without fear of repercussion. Furthermore, the atmosphere must in a retrospective must be that of learning, not blaming. Understanding and truly believing that every person did the best job they could during the iteration or project is of utmost importance (Kerth, 2001). Separate activities and exercises to achieve an atmosphere of trust, honesty and safety are usually a distinct part of a retrospective meeting.

## **2.2 Game development**

Game development is a branch of software development. This section gives an overview of game development and the unique aspects and challenges associated with it.

To understand game development, it is important to start by defining a game. A dictionary definition is:

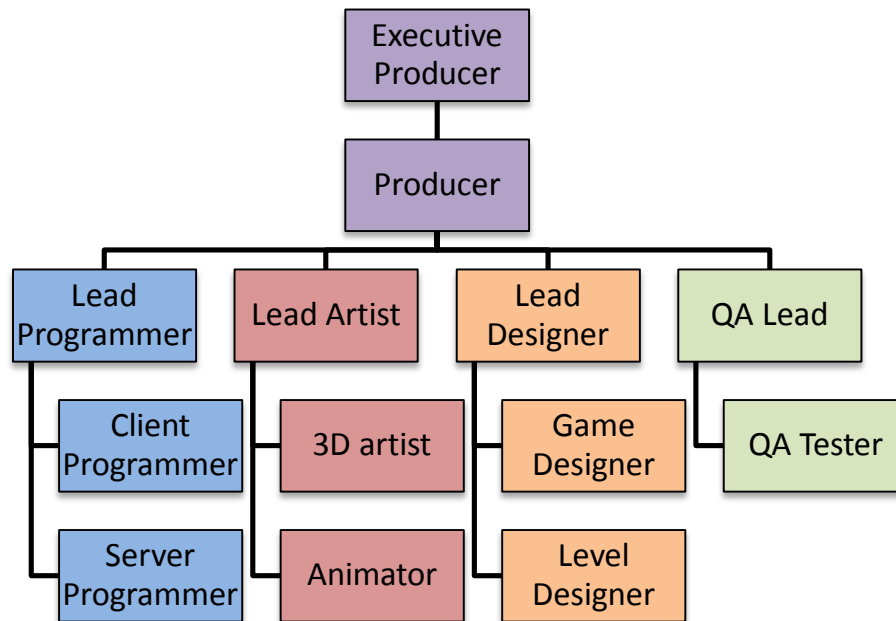
*“a physical or mental activity or contest that has rules and that people do for pleasure”* (Merriam-Webster.com, 2015)

Providing a more specific definition of a game is challenging because games are extremely diverse. The goals of a game can include, for example, providing entertainment, generating social interaction, educating the player, promoting health and fitness and training or recruiting people. Digital games can also be divided into a vast number of genres and sub-genres, such as action games, adventure games, role-playing games, strategy games, puzzle games and simulation games. (Novak, 2012)

Game development differs from traditional software development in many ways, not least of which is the heavier focus on creativity. However, this does not mean that game development is “art” to software development’s “science”. Ultimately, game development is just a different type of software development and there are production methods and processes that game developers should use if they want to consistently create games on budget and on time (Bethke, 2003).

### **2.2.1 Roles in a game development team**

One unique aspect of a game development team when compared to a traditional software team is that it consists of members with very different and possibly non-overlapping skillsets. A modern game development team usually includes some mix of producers, designers, artists, programmers and quality assurance testers (Bethke, 2003; Novak, 2012). Audio designers, who are responsible for producing all the sound effects, spoken dialog and music for the game are either members of the team or outsourced services (Novak, 2012). An example game team structure is presented in Figure 2-3.



*Figure 2-3 Example structure of a game team.*

Management of a game project is generally handled by producers. However, the definition of a producer is fairly ambiguous and the job description and duties can vary from company to company (Bethke, 2003). Production and management duties are also often distributed to different varieties of producers. For example, some game studios employ executive producers who are closest to the upper management or in some cases, act also as the studio heads (Novak, 2012). They are responsible for the business aspects of a project whereas a game producer is responsible for managing the day-to-day work of the project (Bethke, 2003). Duties of a producer can include, but are not limited to, building and maintaining schedules, managing the daily tasks of a game team, managing internal and external dependencies and collaborating with the publisher and other stakeholders (Keith, 2013). Ultimately, a producer's main responsibility is to ensure that the game release on time and on budget while managing and maintaining a productive and a happy development team (Novak, 2012).

Designers are responsible for the design of the game. This includes the mechanics of the game, design of the levels (if applicable) as well as writing of the story and the dialogue in the game. Designers have been described as being the advocate of the eventual player of the game, meaning that their responsibility is to assure that the game is enjoyable from a player's perspective (Keith, 2013). An often-used tool for game designers is the game design document (GDD), which is a detailed document about all aspects of the game, such as characters, levels, game mechanics and menu structures. It is meant to specify the game and solidify the vision or idea of the game that originally may only exist in the mind of the designer (Bethke, 2003). Through the GDD, designers communicate their vision to the different team members. For example, management should know the high concept as well as description of example gameplay, game programmers

need highly detailed descriptions of the game mechanics and artists require a section about the game graphics, where designers describe their vision for the appearance of the world and the characters (Moore, 2011).

Artists are responsible for creation of all art assets in the game. Early in game development history, a game project might have had just a single artist assigned to creating 2D sprites for the game. As computers have evolved, so have the requirements for the visual presentation of games. Nowadays, larger game projects can require an art team of tens of artists and game artists can specialize in a variety of areas, such as 2D, 3D, user interface or animation (Bethke, 2003). Game development has certain special challenges for artists, such as technical limitations on the art quality to maintain an acceptable performance. Also, in order to work efficiently, game artists require good tools and pipelines to deliver their assets into the build. The build itself must also be stable and working as artists need to be able to see their assets in a real game environment to properly iterate on them. (Keith, 2013).

Game programmers create the actual game based on the vision of the designers. Programmers in a game team, like artists, can have a multitude of specializations. Depending on the game being developed, there can be a need for client programmers, server programmers, tool developers, engine programmers or AI programmers among others (Bethke, 2003). In addition to implementing game mechanics and other design-related functionalities, the responsibilities of programmers also include integration testing, creation of unit tests and bug fixing (Rollings & Morris, 2004).

Quality assurance (QA) testers ensure that the game is properly tested and the critical problems in the game are found before release. In addition to bugs and stability issues, the testers look for inconsistencies, usability problems and overall deficiencies in the gameplay experience (Novak, 2012). The exact method for QA varies and is dependent on many factors, such as the size of the developer studio. Smaller studios might not have dedicated QA team whereas large publishers often employ a full QA staff (Bethke, 2003). Traditional approach to QA is often to emphasize testing towards the end of the game project where the game is close to a shippable product. This approach does not necessarily guarantee the highest quality games as some deep-rooted defects are impossible to fix late in the project. For this reason, some developer teams have moved to a more agile approach where testing is done actively throughout the entire project (Keith, 2013).

In addition to the development team, a game company often employs a range of non-development personnel such as community managers, marketing personnel, lawyers and sales representatives (Bethke, 2003). Larger companies tend to employ more non-development personnel whereas smaller companies usually focus on development personnel and work with publishers who handle the non-development side of the game's production.

## 2.2.2 High-level game development process

High-level or macro-level game development process has resemblance to the classical software development waterfall mode. The development process is commonly split into either phases and/or milestones. Van de Weerd et al. conducted a study in 2007 attempting to create a reference method for game development by comparing multitude of production methods. Their reference method proposes four stages of development: *concept phase*, *preproduction phase*, *production phase* and *post-production phase*. This structure is also present in various pieces of literature concerning game development and production (Keith, 2013; Novak, 2012). The overview of the reference method is presented in Figure 2-4.

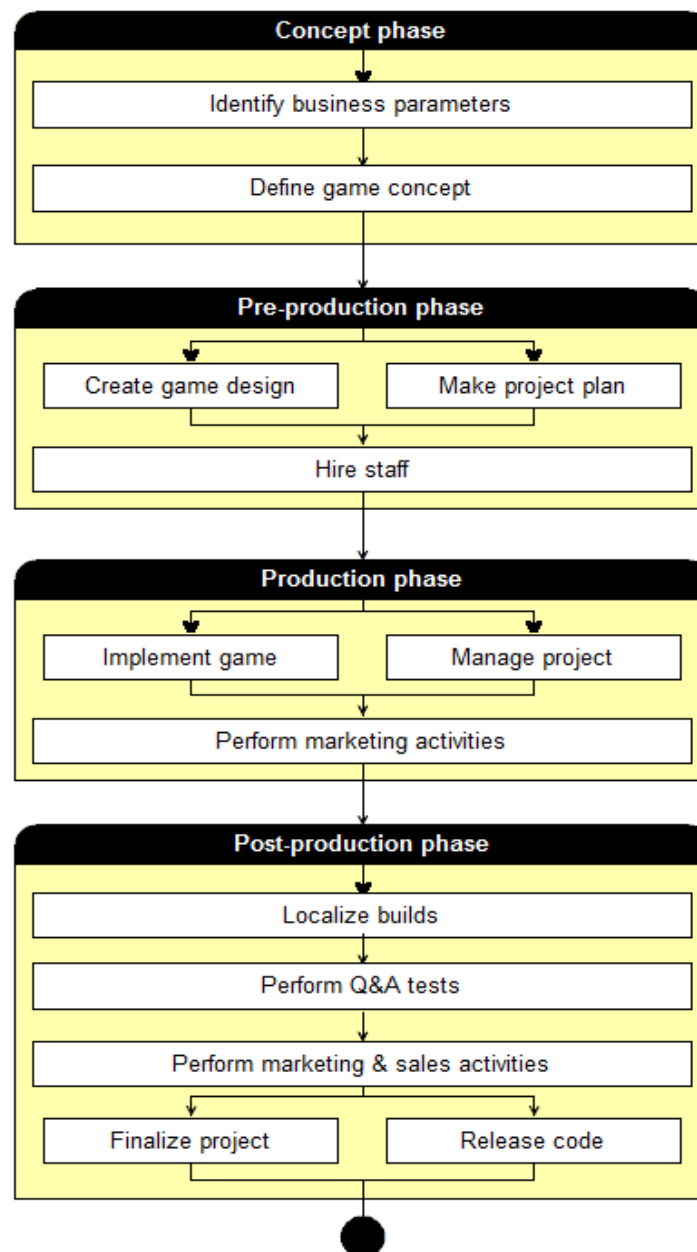


Figure 2-4 Game development process overview (van de Weerd, et al., 2007).



In the *concept phase*, the intention is to map out the business viability of the game and define what the core of the game is about. Main deliverables are the business parameters of the game and a game concept document accompanied by a prototype of the game. *Preproduction phase* prepares the production by creating detailed game design documents and a project plan as well as confirming that the personnel resources needed for the project exist in the company. Preproduction is also about mapping the unknowns and potential risks of the project and planning around them. It is considered to be the most important phase of game production by some (Bethke, 2003). *Production phase* covers the technical implementation of the game as well as preliminary marketing activities. Hypothetically, production should be about following a clear plan prepared in pre-production and contain relatively little surprises (Bethke, 2003). In *post-production phase*, the game goes through QA testing and the content is localized to all necessary languages. Marketing and sales activities are performed with possible help from the game team in creation of marketing assets. (van de Weerd et al., 2007)

Another method to formalize the game development process is the use of milestones. Milestones are frequently used in addition to the phases described above. The term milestone indicates a point in the project with specifically defined deliverables. The milestones usually have a date assigned to them before which the milestone deliverables must be ready. However, industry definitions for the names and amount of milestones are not standardized. Also, the definitions of deliverables for each milestone may differ from company to company. Table 2-1 contains the milestone definitions used by the target company to provide a clear reference to the milestone terms used in this thesis.

**Table 2-1** Milestone definition in the target company.

Milestone	Definition
Alpha	Feature complete build. No new features are allowed after Alpha.
Beta	Content complete build. All art assets, story content etc. must be ready and in the build.
Release Candidate (RC)	Build with no known critical bugs.
Gold Master (GM)	Final build of the game that is sent to the publisher or released in stores.

As mentioned above, the traditional development process ideally assumes that the plan for the game is set in pre-production and production is only about following that plan. As seen in Table 2-1, the alpha milestone is technically the point where all features are done and all future work focuses on polishing the content without adding new features.

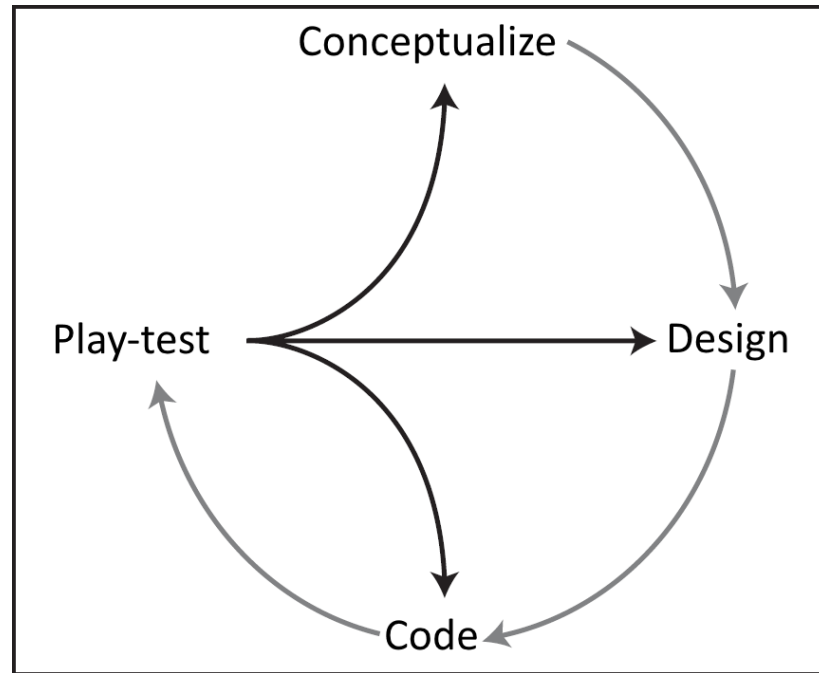
However, in reality, the design of the game will change from the original design in a vast majority of productions (Bethke, 2003). Two main drivers for change are either new ideas for features during development or old features not working as expected. This leads to new, emergent requirements added to the design document in the form of completely new features or re-designed old features (Keith, 2013). Often this leads to what is known in the industry as “feature creep”, a continuous increase in the amount of features. This extends the scope of the project beyond what was originally planned and is therefore a risk to the schedule of the project (Bethke, 2003; Petrillo et al., 2008). Feature creep is a particularly challenging aspect of the post-alpha development. The CEO of the target company mentioned in an interview that even though alpha milestone should lock the features of the project, he has not seen a single project in ten years in the industry where new features were not added after the alpha milestone. It is difficult to determine if this kind of breach of milestone rules is objectively detrimental to the development process. Game development is a highly iterative process and sometimes good post-alpha feature additions can be even a good business decision.

Another game-industry phenomenon related to the milestones is called “crunch”. Crunch often occurs in the time period towards the end of the project or a milestone when a deadline is approaching, but the project is behind in schedule. In many cases, this translates to increased intensity of work as well as significant amount of overtime, causing the team to constantly work long days and in some cases work during weekends. This is especially true in cases where the company developing the game is a subcontractor and has received advance payments for hitting their milestones from a publisher or other third party. In these situations, missing the deadline is not an option and overtime is often the only solution. Crunch is usually pronounced during the final stages of production just before the GM milestone. This is at least partially caused by most game projects failing to properly schedule. Furthermore, game development is an occupation of passion for many so even in a case where overtime is not required, they voluntarily work extra hours to make everything as perfect as they can. (Bethke, 2003; Petrillo et al., 2008)

### **2.2.3 Agile game development**

The traditional game development process assumes a complete design document during pre-production. In reality project requirements are usually changing throughout the development process as the design of the game changes due to the highly creative nature of the process. While the high level development process resembles the waterfall model, the day-to-day micro-level development processes are often much more agile. A 2010 study (Petrillo & Pimenta, 2010) analyzed 20 postmortems of various game studios and suggested that many game studios employ agile practices without necessarily consciously making the decision to adhere to agile. The authors believe that developers might have doubts towards official agile processes as overly formal and rigid while ac-

tually employing many agile practices in their work because they find them useful. Another study (Stacey & Nandhakumar, 2008) interviewed personnel from three game companies and likewise found many working practices similar to those in agile methodologies. A very key finding by the authors was the constant playtesting and a subsequent feedback loop created by it. They visualized the idea with a “Play-test boomerang” seen in Figure 2-5.

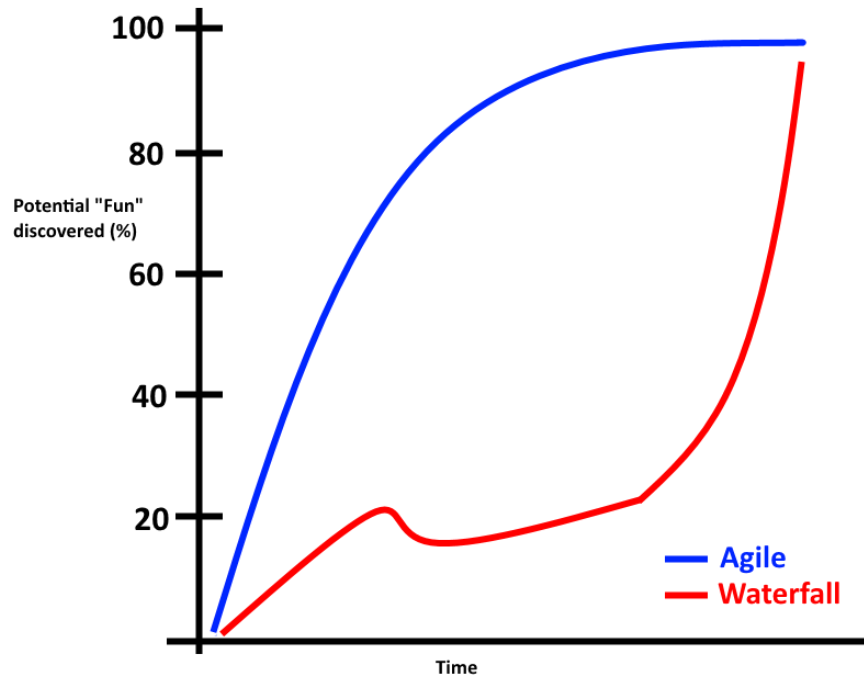


**Figure 2-5** The “Playtest boomerang” (Stacey & Nandhakumar, 2008).

The boomerang depicts a constant flow of the game development in the studied companies. Gameplay features start as a concept, move to more specific design, are then programmed into the game and finally play-tested. Depending on the playtesting evaluation and the completeness of the feature, it may be accepted as is or returned to one of the previous stages. For example, features that need a complete overhaul are returned to concept stage, and features that are almost there but have some programming defects are returned to the programming stage. This constant loop of evaluation is very similar to the iteration-based agile methodologies.

There are other proponents for consistent play-testing as well. Keith argues in his book *Agile Game Development with Scrum* (Keith, 2013) that no matter what business parameters or other goals for the project may exist, the ultimate goal in making a game is about making it a fun, compelling and engaging experience for the player. Predicting if a game will fulfill these requirements is very difficult without actually playing the game. Therefore the traditional sequential production process where the entire game design is determined before beginning production is risky. It often leads to a situation where the first two thirds or more of the project is spent on simply following a plan and hoping it will lead to a fun game. Only towards the end of the project is the entire

gameplay available and an evaluation of it can be done. The end of the project also happens to be the busiest where usually it is difficult to find time or resources to make changes that would be necessary based on the evaluation. The author therefore suggests that agile methodologies are much more suitable for game development needs and illustrates the difference with a graph similar to Figure 2-6.



**Figure 2-6** Realization of value over time in agile vs. waterfall (adapted from (Keith, 2013; McGuire, 2006)).

The graph equates value of a game project as the percentage of the discovered fun the game can potentially offer players. It highlights the distinct difference where the acceleration in the value gained happens in waterfall compared to agile. Very similar graphs are also present in (McGuire, 2006), where the functionality over time in waterfall approach is an exponential curve and in agile it is logarithmic.

The rapid acceleration at the end of the waterfall timeline coincides with the crunch period towards the end of the project. It is not coincidental, because this is the time period where in waterfall the entire gameplay is now available, and the only options in case of unwanted functionality or a lacking gameplay experience are either to cut the underperforming features or work significant overtime to correct them. (McGuire, 2006)

It is possible that by employing agile practices the need for extensive crunching would decline. The acceleration of value growth happens early on as the most important features are prioritized. The later stages of the project are at least in a theoretical case spent more on minor polishing and tweaking of the game. Agile methodologies in general are advocated by many as potentially beneficial and fitting for game development (Keith, 2013; McGuire, 2006; Al-azawi et al., 2014).

Despite the surplus of agile advocates, contradicting evidence exists. A recent quantitative study (Tozour et al., 2015) surveyed over 300 game project teams trying to establish statistical links between certain aspects of game development and the outcome of the game. The outcome was defined as an aggregate score of in-time delivery, critical success, return-on-investment and fulfillment of internal goals. There were many factors of game development that had a statistically significant correlation with positive game outcome but the choice of production methodology (waterfall, agile or even ad-hoc) was not one of them. Instead, the study revealed that training the team with the used production methodology, involving everyone to prioritize the work for each milestone, and making an effort to constantly improve the production methodologies were all significantly positively correlated with a better game outcome. An observation can be made from this that all the positively correlated activities are actually practices that are advocated by many agile methodologies. However, it might be best not to make strong claims about superiority of any given production methodology given this data. Important conclusion for the purpose of this thesis is that regardless of other findings, effort to constantly improve the production methodologies correlates with better games.

## 3. STRUCTURE OF THE CASE STUDY

This chapter introduces the structure of the case study executed in this thesis. Section 3.1 presents the research goal and the research questions of the study. Section 3.2 covers the research methods used. A template for the retrospective meeting, meeting log, pre-study interview and post-study questionnaire will be provided. Retrospectives and postmortems both have phases that contain activities. Phase names are underlined and activity names are **bolded** to improve readability. Section 3.3 covers the execution and timeline of the study. It showcases the expected run of the study as well as preparation for possible interfering developments during the study. In Section 3.4 the limitations of the study are discussed. This includes a list of potential risks and anticipated sources of error.

### 3.1 Research goal and questions

The goal of the target company in the study was to unify and improve the working processes of the company. The method chosen to accomplish this are retrospective and postmortem meetings. In addition to this goal, the research will also look to answer the following questions:

**RQ1:** *Can retrospective meetings be used separate from other components of agile methodologies to improve working processes and environment?*

**RQ2:** *Are retrospective meetings suitable in a cross-discipline game development team context?*

**RQ3:** *What is the employee perspective towards continuous retrospective meetings?*

In order to accomplish the goal and answer these questions, a multitude of data will be collected through the study both in the retrospective meetings and outside of them. Next section presents a detailed plan for collection of the data.

### 3.2 Research methods

The primary research method of this work is case study. In the book *Case Study Research: Theory, Methods and Practice* (Woodside, 2010) it is recommended to compose data based on triangulation consisting of (1) directly observing the environment, (2) asking the participants for explanations and interpretations of operational data and (3) analyzing written documents and natural environments in the case. The recommenda-

tion for multiple data sources and triangulation is also repeated in other literary sources (Turner & Danks, 2014). In order to satisfy this recommendation, three main methods to collect data will be used. They are private interviews and questionnaires with employees, bi-weekly retrospective meetings and project postmortems. These methods cover the direct observation of the environment as well as asking the participants for explanations and interpretations. The company does not have written documents concerning the processes, so the third requirement will unfortunately have less emphasis. Additional methods may be added as research progresses (Turner & Danks, 2014). Metrics-based data collection was also considered, but it was ultimately decided that the short duration would mean that most teams are in only one, two or in rare cases three different milestones during the study. As the intensity of work varies between different milestones, the metrics-based data would likely be unreliable.

The study can also be considered as action research. Action research is a controversial topic as a fundamental aspect of it is challenging the traditional paradigm of research. Action research can be defined as a participative, democratic process where theory is generated through practice, as opposed to the traditional way of generating theory through research and only later applying it to practice (Brydon-Miller et al., 2003). However, action research does not have a unified methodology as it is more of an orientation to generate knowledge with people rather than examine them as subjects. Action research can be described as people working as a group to generate practical knowledge and address problems facing their communities or organizations (Reason & Bradbury, 2008). Interestingly, this description is similar to that of retrospectives.

### **3.2.1 Retrospective meetings**

The majority of the data used in the study will be collected through retrospective meetings. In order to make collection of data efficient and reliable, a template for the retrospective has been designed and is presented in this sub-chapter. Every retrospective that is held as a part of the study will follow this template to the extent that is feasible. An important note is that in general usage, retrospectives should not be considered as something with a static form but rather constantly adapt according to the feedback from the team (Derby & Larsen, 2006).

The template for the retrospective is presented in Table 3-1. It is based on the Everyday retrospective (Agile Retrospective Resource Wiki, 2013). The original Everyday retrospective has been designed to last two hours. In this study, the duration of the retrospective is limited to one hour and the timeline from the original source has been adjusted accordingly. The activities in the retrospective have also been slightly modified and grouped according to the five-phase structure presented in Subsection 2.1.3. Each activity is time-boxed and the facilitator will use a timer to make sure the activities do not exceed their time limits. Descriptions for the activities can be found in Table 3-2, Table 3-3, Table 3-4, Table 3-5 and Table 3-6.

*Table 3-1 Template for the retrospective meeting.*

<b>Phase</b>	<b>Activity</b>	<b>Time</b>
<u>Set the stage</u>	<b>Preparation</b>	(done beforehand)
	<b>Check-in</b>	2 min.
<u>Gather data</u>	<b>Happiness Histogram</b>	5 min.
	<b>Three words</b>	15 min.
<u>Generate insights</u>	<b>Open discussion &amp; Dot voting</b>	10 min
<u>Decide what to do</u>	<b>Identifying Goal &amp; Action</b>	20 min
<u>Close the retrospective</u>	<b>Check-out</b>	5 min.

*Table 3-2 Activities for Phase 1 – Set the Stage.*

<b>Activity</b>	<b>Description</b>
<b>Preparation</b>	Facilitator prepares the meeting by drawing a timeline of the previous 2 weeks on a whiteboard. They also ensure availability of post-it stickers and whiteboard marker pens. This activity is done before the meeting starts.
<b>Check-in</b>	Each team member takes turns to describe the past two weeks of work using exactly two words. After this, they estimate if they are happier than they were at the previous retrospective and give thumbs up if happier, thumb sideways if undecided and thumb down if they are less happy.



*Table 3-3 Activities for Phase 2 – Gather data.*

<b>Activity</b>	<b>Description</b>
<b>Happiness Histogram</b>	Facilitator asks the team for significant events that happened during the past two weeks. They are marked on the timeline on the whiteboard. After this, each team member takes turns to draw line from left to right across the whole timeline, with the height of the line corresponding to the emotional state of the team member at that point in time. Higher the line, the more positive the emotional state.
<b>Three words</b>	Facilitator writes 3 words on the whiteboard. The words are changed for every meeting, but the meanings will be similar. Examples are “Good / Bad / Ugly”, “Mad / Sad / Glad”, and “Keep Doing / Start Doing / Stop Doing”. Team members will have 5 minutes to write down thoughts on post-it stickers about the past 2 weeks that fall into one of these categories. The other 10 minutes is used in team members taking turns to explain their ideas and put the stickers onto the whiteboard. Stickers containing similar ideas are grouped together. After the grouping, the facilitator adds the item the team considered the most important in the last retrospective (if it is not already on the board).

*Table 3-4 Activities for Phase 3 – Generate insights.*

<b>Activity</b>	<b>Description</b>
<b>Dot voting</b>	Team members are given some time to think about which stickers (or groups) they think contain the most important thoughts/ideas/issues. Each team member has 5 “dots” they can distribute on the stickers. More dots signify more importance. Team can also vote the item from previous retrospective, signifying that the improvement actions taken have not been successful.

**Table 3-5** *Activities for Phase 4 – Decide what to do*

Activity	Description
<b>Identifying Goal &amp; Action</b>	This activity begins with free form discussion to confirm that the team agrees that the sticker with the most dots is the one that should be focused upon during the next two weeks. After this, from the sticker the team defines a Long Term Goal that is the ideal to reach for and a Short Term Action that can be actioned on immediately. An example of this is if the sticker with the most votes was “Too many broken builds”, then the Long Term Goal could be “Build is never broken” and the related Short Term Action “Create a commit checklist” ( <i>Agile Retrospective Resource Wiki, 2013</i> ).

**Table 3-6** *Activities for Phase 5 – Close the retrospective*.

Activity	Description
<b>Check-out</b>	The retrospective ends with a similar thumb vote as it began with. This time the team members vote if they felt that the retrospective was useful to them or not.

In order to ease the data collection effort of the study, a template for a retrospective log is presented in Table 3-7. Explanation of terms is given in Table 3-8. The facilitator of the retrospective will fill out the log for each retrospective meeting during the study. A completely filled log is the minimum limit for data collected from a retrospective. In addition, the facilitator will write down freeform notes.

Table 3-7 Template log for a retrospective.

# METADATA

Round#

Date

Retrospective#

# PARTICIPANTS

Team  Milestone

Team Members	Name	Check-in vote	Check-out vote
#1			
#2			
#3			
#4			
#5			
#6			
#7			

# ISSUES

# of stickers

Old Issue

Issue #1

# of votes

Long Term Goal

Short Term Action

Issue #2

# of votes

Issue #3

# of votes

Issue #4

# of votes

**Table 3-8** Terms used in the template retrospective log.

<b>Term</b>	<b>Description</b>
<i>Round#</i>	Running number representing the number of retrospectives each team has had.
<i>Retrospective#</i>	Running number of retrospective within a round. Each retrospective can be uniquely identified by combining the Round# and Retrospective# numbers.
<i>Team</i>	Identifying number for the team.
<i>Milestone</i>	Next milestone in the project.
<i>Team Members</i>	List of every team member's name and their vote in the check-in and check-out votes. The votes are recorded as + (thumb up), 0 (thumb sideways) or – (thumb down).
<i># of Stickers</i>	Total count of stickers in the Three words activity.
<i>Issue #Y</i>	The top 4 issues from the Dot voting activity are listed here. The # of votes each received is also recorded. For the #1 issue, the Long Term Goal and Short Term Action defined by the team are also recorded.
<i>Old Issue</i>	Checked if this issue is from the previous retrospective.

### 3.2.2 Postmortem meetings

In addition to iteration retrospective meetings, the study will also include two project postmortem meetings for projects that are scheduled to end during the study. For both projects, both the developer team and the customers will attend the meetings. The structure will follow the three-phase structure from (Kerth, 2001). A template for a 6 hour long postmortem is presented in Table 3-9. Descriptions for each of the activities are presented in the following paragraphs. The activities have been selected from (Kerth, 2001) and they have been modified where necessary due to space and schedule constraints. Kerth suggests postmortems to last for three days but in the context of the study it was not possible. Facilitator of the meeting will make adjustments to the durations of each activity based on the situation at hand. Potentially fruitful discussions will not be stopped even if the activity has went past its time limit.

*Table 3-9 Template for the postmortem meeting.*

<b>Phase</b>	<b>Activity</b>	<b>Time</b>
<u>Readying</u>	<b>Preparation</b>	(done beforehand)
	<b>Introductions &amp; Schedule</b>	20 min.
	<b>Create safety</b>	30 min.
<u>Past</u>	<b>Artifacts contest</b>	45 min.
	<b>Build the timeline</b>	60 min.
	COFFEE BREAK	10 min.
	<b>Mine the timeline</b>	60 min.
<u>Future</u>	<b>Cross-affinity teams</b>	90 min.
	<b>Change the paper</b>	45 min
(When necessary)	<b>Form natural affinity groups</b>	5 min

**Form natural affinity groups:** This activity may be performed as a part of different activities depending on results of other activities. It is performed once per postmortem and the same groups are used if they are needed in multiple activities. In the activity, facilitator prohibits all participants from speaking and asks them to stand up. Then they are instructed to locate themselves next to people who they worked with the most during the project. Once the facilitator sees that all significant movement has happened, he declares the end of the activity and assigns people to groups based on where they are standing. The purpose is to form groups where everyone is familiar with each other and discussion is as comfortable as possible.

**Preparation:** Facilitator prepares the meeting. They send an e-mail to the participants in advance, tells them the important details such as location and time. He also asks them to bring with them the artifacts needed in **Artifacts Contest**. Facilitator also prepares the necessary equipment and supplies. Minimum necessary equipment are large post-it notes, marker pens for every participant, roll of paper and tape for **Build the timeline**, A3 paper for **Change the paper** and prizes for the **Artifacts Contest**. Just before the meeting, the facilitator brings all equipment to the reserved meeting room and tapes a long line of the paper on the wall as the basis for the timeline.

**Introductions & Schedule:** All participants are asked the following three questions to get everyone acquainted and talking:

- What is your name?
- What was your role in the project?
- What is your favorite game of all time? Use exactly one sentence to explain.

After everyone has answered, a schedule for the postmortem is presented to all participants. Every activity and their purpose will be briefly described.

**Create safety:** First, facilitator writes two ground rules of the retrospective on a whiteboard as well as presents the group with “Kerth’s prime directive”.

1. All participation in all exercises is optional
2. We will not make jokes about anyone in the room

“Kerth’s Prime Directive: Regardless of what we discover, we must understand and truly believe that everyone did the best job he or she could, given what was known at the time, his or her skills and abilities, the resources available, and the situation at hand.” (Kerth, 2001, p.101)

Next, a closed ballot where every participant is asked to vote from 1-5 about how safe they feel discussing matters related to the project given that there are managers and customers in the room. 5 means they feel they can say anything and 1 means they will not say what they are thinking about as they fear something, for example repercussions from criticizing the management of the project. Facilitator counts the votes and draws a bar chart depicting the vote counts. If there are any votes below 3, additional rules are needed. The *Form natural affinity groups* activity is performed and the groups are given 10 minutes to think of any additional rules that should be added to increase the safety. After 10 minutes, all new rules are added to the whiteboard and another vote is conducted. This cycle is repeated until no votes are below 3.

**Artifacts contest:** Each participant has been asked beforehand to search for artifacts related to the project. Examples of artifacts are memos, meeting notes, old schedules, design wireframes, personal or project calendars, white papers, budgets, project plans and personnel loading charts. An artifact can be anything that helps the team remember everything that happened throughout the project. In the activity, every person presents all artifacts they brought and tells their story. Artifacts are laid out on a table and in the end a vote is held in two categories. Team votes for the most significant as well as the most unusual artifact and the members who brought them are rewarded with small rewards. This is done to incentivize people to look for artifacts.

**Build the timeline:** In this activity, the purpose is to build a complete timeline of the project with all the significant events that happened. First, starting and ending dates of

the project are written on the timeline. Then, each participant is asked to write their name on a large post-it note and put it onto the timeline where they entered the project. Then, if not yet done, **Form affinity groups** activity is performed. Participants are asked to work in the affinity groups to discuss everything significant that happened throughout the project. Artifacts from **Artifact contest** are left on the table in order to help participants memorize the entire project. Groups write significant events on large post-it notes and put them onto the wall above and below the paper that has been set up on the wall. When all events are posted, the facilitator asks the team members to draw a line similar than in **Happiness Histogram** (Subsection 3.2.1) through the project to depict their emotional state throughout the project. High means they were happy and low means they practically hated their job at that point in time.

**Mine the timeline:** Five questions are written onto the whiteboard:

1. What worked well that we don't want to forget?
2. What we learned?
3. What we should do differently next time?
4. What still puzzles us?
5. What we need to discuss in greater detail?

The facilitator then asks the whole group to take a look at the constructed timeline and discuss any thoughts they have regarding these questions. Thoughts for each category are written down on the whiteboard. At the end of the activity, the facilitator asks the group to identify the most important topics from the answers to questions 4 and 5.

**Cross-affinity teams:** In the penultimate activity, the affinity teams are lined up and new teams are formed by picking one member from each affinity team. The intention is to create teams with viewpoints from all across the project team hierarchy. These teams are tasked to discuss the last two questions of **Mine the timeline** in detail. Each team is assigned a topic from the previously decided most important topics. Every team is asked to produce a short presentation about their ideas for solving the issue. All presentations will be performed at the end and if the team feels some of them should be forwarded to someone outside the meeting the facilitator will arrange it.

**Change the paper:** The last activity is meant to capture the essential learnings of the postmortem. The group is asked to design any number of posters they feel is necessary to contain all important learnings. The group is informed that the posters will be framed and posted in a visible place so that the learnings are not forgotten. The end result of the exercise should be a design for the poster and an artist who was present can then properly illustrate it with art from the game.

### 3.2.3 Interviews and questionnaires

Private pre-study interviews with every employee will be conducted at the beginning of the study. In the beginning, the interviews focus on collecting the necessary metadata about every employee as well as their perceptions of the state of the work processes and environments. Metadata will contain the age, gender and name of the employee (name is redacted in the thesis). Interviews will be conducted privately between the employee and the researcher. The answers will be translated from Finnish to English. The translated transcripts will be approved by each interviewee to prevent distorting errors caused by translation. The following questions will be presented:

- *What is your position at the company?*
- *How many years of industry experience do you have?*
- *How long have you been with the company?*
- *What is your perception of the work environment at the company?*
- *Describe your daily work routine*
- *Describe the current working processes your project is using.*
- *Describe the company wide conventions and processes.*

At the end of study, a post-study questionnaire will be sent to all employees. The questionnaire will contain the last 4 questions from beginning of study interviews to evaluate the change in employees' perception of the environment and working processes. In addition, it will also contain a survey asking about the employees' feelings towards the retrospectives and postmortems they participated in. The questionnaire has optional paths that are described as indentations in the presentation below. Post-study questionnaire questions are as follows:

- *What is your perception of the work environment at the company?*
- *Describe your daily work routine*
- *Describe the current working processes your project is using.*
- *Describe the company wide conventions and processes.*
  
- *How many postmortems did you participate in? (0-2)*
  - *Which postmortems did you participate in? (Check all teams that apply)*
  - *Describe your thoughts about the usefulness of the postmortem meetings you attended (1-5, 1 = I did not find them useful at all, 5 = I found them extremely useful)*
  - *How do you feel about the duration of the postmortem meetings (6-8 hours)? (1-5, 1= too short, 3 = perfect duration, 5 = too long)*
- *What is your opinion on the future of postmortem meetings in the company?*
  - We should not continue having postmortems
    - *Why do you think postmortems should not continue? Please be as specific as possible.*
  - We could keep having postmortems, but they should be different
    - *How, in your opinion, should the postmortems change? Please be as specific as possible.*
  - We should keep having postmortems regularly



- *How many iteration retrospectives did you participate in? (0-4)*
  - *Which retrospectives did you participate in? (Check all teams that apply)*
  - *Describe your thoughts about the retrospective meetings held during the study (1-5, 1 = I did not find them useful at all, 5 = I found them extremely useful)*
  - *How do you feel about the duration of the retrospective meetings (1 hour)? (1-5, 1= too short, 3 = perfect duration, 5 = too long)*
- *What is your opinion on the future of iteration retrospective meetings in the company?*
  - We should not continue having iteration retrospectives
    - *Why do you think iteration retrospective meetings should not continue? Please be as specific as possible.*
  - We could keep having iteration retrospectives, but they should be different
    - *How, in your opinion, should the iteration retrospectives change? Please be as specific as possible. (After this question also ask the next question about frequency of retrospectives)*
  - We should keep having iteration retrospectives regularly
    - *How often should iteration retrospectives be held in your opinion? (Weekly/Every two weeks/Every three weeks/Once a month/Other: )*
- *Give feedback, ideas and recommendations about retrospectives and postmortems*

The questions have been formatted to be as neutral as possible in order to avoid influence the attitudes of the respondent. Quantitative data will be collected through questions using a Likert scale from 1-5 (Likert, 1932). The most important questions for the study's purposes are the ones measuring how useful employees found the retrospectives and postmortems as well as the questions regarding if these meeting should continue after the study.

### **3.3 Execution and timeline of the study**

Before any retrospective meetings are held, every employee participating will undergo the pre-study interviews. The retrospective meetings will be conducted over a time period of three months. During these three months, each team in the company will participate in a retrospective meeting bi-weekly. In total, each team is expected to have 6 retrospective meetings during the study. It is possible that project teams disband or are formed during the study. All such occurrences will be properly documented and taken into account. All new teams formed during the study will also participate for the remainder of the study. At the end of the three month period, the post-study questionnaires are sent to all employees in the company. Two projects are scheduled to finish towards the end of the study and a postmortem meeting will be held for each once they are completed.

### 3.4 Limitations

A known limitation in using case studies is that it is dangerous to make general assumptions from data collected from a very specific context (Woodside, 2010). This risk is pronounced by the small sample size and short duration of the study. The findings and possible improvements might not apply in any other context than that of the target company.

Another potential problem is that the data sources are not very diverse. All of the data comes from the employees through either the retrospectives or the personal interviews. This is potentially problematic due to the highly subjective nature of the data. However, it is important to note one of the goals of the research is to enhance the work environment. Perception of the work environment is a highly subjective matter so the subjective data collected can be useful or even necessary to achieve this goal.

One further risk in the data collection is the researcher's conflict of interest. As a retrospective facilitator, they must be participating in the discussion and suggesting possible improvements for the team, but as a researcher they should stay as impartial a possible and simply observe. There is a risk that the outcomes of the retrospectives are too similar to each other if the teams have a tendency to give the facilitator's opinions and suggestions too much weight. It is very important for the researcher to try to assist the teams in generating ideas without overly polluting the discussion with their own ideas.

## 4. GATHERED DATA

This chapter presents the data collected through pre-study interviews, retrospectives and postmortems. Section 4.1 introduces the personnel participating in the study as well as a summary of their pre-study interviews. Section 4.2 covers the different teams that participated in the retrospectives over the course of the study. Section 4.3 presents the realized timeline of the study as well as changes to the planned structure of the study. Section 4.4 lists retrospective journals from each individual retrospective. Section 4.5 presents detailed journals from the postmortem meetings.

### 4.1 Participating personnel

A total of 16 employees of the company were interviewed in the beginning of the study. The company does not have a strict job title policy and most of the interviewees mentioned that they do not even know if they have a particular title or not. Regardless, a rough split of personnel per title basis can be made from the answers. The results are presented in Table 4-1.

*Table 4-1 Participating personnel by (approximate) job title.*

<b>Title</b>	<b>Number of personnel</b>
Senior Programmer	2
Programmer	4
Junior Programmer	2
Lead Artist	1
Artist	2
Designer	1
Producer	2
Community Manager	1
QA Tester	1

In addition to the interviewed personnel, the company also has a CEO and some sub-contracted employees who work remotely. Many of the employees also have split capabilities and responsibilities. For example, one employee has education in both design and graphical design and splits his work between both duties.

As mentioned in Subsection 3.2.2, the pre-study interviews consisted of the following questions:

- What is your position at the company?
- How many years of industry experience do you have?
- How long have you been with the company?
- What is your perception of the work environment at the company?
- Describe your daily work routine
- Describe the current working processes your project is using.
- Describe the company wide conventions and processes.

The average industry experience in the company was 5.2 years. The average company experience was 1.8 years. There were 6 people whose entire industry experience consisted from the time in the company.

The perception of the work environment was very much aligned. Most people considered the environment to be open, free of hierarchy and trusting of the employees. Micromanagement and watching over people's shoulders did not happen as people felt they were generally trusted to do their job. Skype (Skype/Microsoft, 2015) was used for most of the communication and the environment was considered by many to be very silent. This was seen as a negative by some and as a positive by some. Interviewees who considered it to be a negative wished there was more face to face communication. There was very little if any official hierarchy. This was generally considered to be a positive factor. However, many interviewees mentioned that communication both on the project and company level is sometimes lacking and some attributed this to the lack of hierarchy. Some interviewees mentioned that it is very beneficial for the environment that people share similar backgrounds, such as roleplaying, board gaming and/or card gaming.

Processes and conventions on project level were seen as extremely lightweight. All projects used Skype as communication tool and Trello (Trello, Inc., 2015) as bug tracking tool. Some projects used Trello also for task management, while others maintain personal TODO-lists and/or distributed tasks over Skype. One project had a very recent convention of weekly builds. Many interviewees mentioned that projects used to have recurring meetings, but they were consistently abandoned at some point in projects. Reasons given were timetable pressures. Both interviewed producers still had a weekly conference call with the project customers. In one of the projects the entire team used to take part in the conference call but over time it was reduced to just the producer. No reason for this was identified.

Company level conventions and processes were non-existent. All interviewees unanimously answered that there are no conventions or processes. There was an attempt at having weekly company meetings but it stopped after one time when the facilitator changed companies.

## 4.2 Participating teams

This section lists all teams that participated in retrospectives or postmortems.

**Team A** was working on a game that was being developed for an external publisher. The team reached their Beta milestone one day before the first retrospective. This project was a very important one for the company and had recently added personnel as it was approaching the GM and launch dates. The timespan of the retrospectives in the study covered the end of the project and the beginning of the update and expansion work for the game.

**Team B** was working on an update for the company's own collectable card game. The update consisted of new cards and new functionalities such as chat, friend list and a quest system. The team consisted of 3 programmers and one remotely working artist. The update was completed by the time of the first retrospective so this team only had one retrospective, which had broader scope and looked at the development of the entire update.

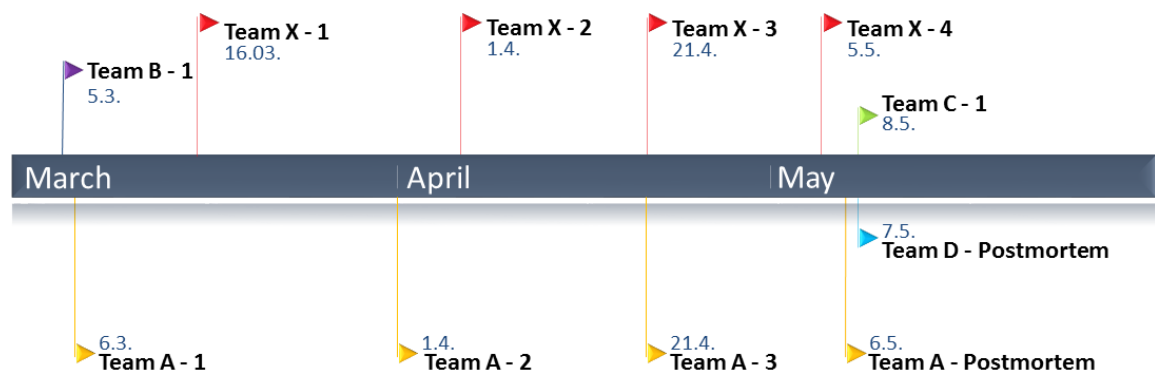
**Team C** was assembled towards the end of the study as a new customer project began. The product was a simple puzzle game with short production time and a small team. This team had only one retrospective during the study.

**Team D** had a project that was developed for an external customer, who had commissioned a game for an intellectual property they owned. The project was mostly completed by the time the study began. One programmer was still working on fixing bugs. No iteration retrospectives were held for this team, but they did have a project postmortem.

**Team X** was a name for the group of people who were not on a team at the time of the retrospectives. This consisted of some single person projects as well as a Community Manager and QA Tester who were considered as shared resources between projects. Participating personnel changed each week through the study as teams finished projects and some people started new projects. The retrospectives were modified to accommodate for this group of people. The *Happiness Histogram* activity was removed. Discussion was also steered towards generic best practices and company-wide conventions as the participants did not share a project that they could focus on. Individual issues were not ignored however, as they also provide learning opportunities. These retrospectives were experimental, as generally retrospectives look at a time period where the participating people worked together on a project.

### 4.3 Realized timeline of the case study

Contrary to the initial plan, only 4 rounds of iteration retrospectives were conducted during the 3 months period. This was due to many factors such as inconsistent availability of personnel as well as schedule conflicts. The company moved to a much larger office just as the study was about to be conducted, so a short time period was reserved for letting people settle in before starting the retrospectives. There was also a two week period where no project was in a state where it would be reasonable to conduct an iteration retrospective, as the projects had either just started or were just about to finish. Figure 4-1 presents the timeline for all retrospectives and postmortems that were conducted. Retrospectives are marked with the syntax of “Team A - 3” meaning the 3<sup>rd</sup> retrospective for Team A.



*Figure 4-1 Timeline of all retrospectives and postmortems.*

### 4.4 Retrospective journals

This section contains short summaries of each retrospective meeting held during the study. Observations from the facilitator, notes from the discussion as well as the agreed long term goal and short term action are presented for each retrospective. Teams had a different amount of retrospective meetings as some of the projects started and others ended during the study.

#### 4.4.1 Team A

**Team A - 1:** The retrospective had 8 participants and none of them had ever participated in any retrospectives. Despite this, the meeting proceeded relatively rapidly and finished after 1 hour and 15 minutes. The general feelings towards the project were positive even though the team admitted there had been a period of intensive overtime on the same week. Almost all team members expressed the wish for a regular meeting to check on the status of the project to keep the entire team aware of all relevant developments in the project.

- **Long term goal:** Everyone in the team is aware what is going on in the project.
- **Short term action:** Start having weekly status meetings and prioritizing Trello tasks.

**Team A - 2:** The retrospective had 8 participants. GM milestone of the team's project was scheduled to be in 3 days and visible stress and tiredness was observable from all participants. The word choices in two word check-ins reflected this. In addition to stress and tiredness, many team members had been physically ill and had to have some sick leaves during the 2 weeks. No exact figures were discussed, but a non-trivial amount of overtime had occurred in the previous two weeks. Many team members also expressed discontent over new feature requests very late in the project. It was apparent that the project was affected by at least a slight amount of crunch and feature creep. The team members expressed their sincere worry that the game would not have the overall quality that they had hoped for because there seemed to be not enough time to polish the game.

- **Long term goal:** Quality of the game does not suffer from schedule pressure.
- **Short term action:** Present concrete expansion plan and schedule to customer before starting work and do not allow scope changes after locking plan down. Have a postmortem meeting about the project with the customer.

**Team A - 3:** The retrospective had 5 participants. One member of the team had left the company between milestones and an additional QA tester had been added to the project. The project was nearing its end as the last 3 weeks had been spent in continuously fixing and submitting various GM builds. Some team members had been on a one-week vacation during this period. Discussion proceeded smoothly and the main topics were communication with the publisher and handover of the project to the expansion team. Communication with the publisher was deemed to be a topic that should be discussed in the project postmortem where representatives from the publisher would also be present.

- **Long term goal:** Expansion team knows necessary things about the project and its situation when they start.
- **Short term action:** Producer prepares the necessary materials and information and arranges a hand-over meeting after the game is shipped.

#### 4.4.2 Team B

**Team B - 1:** This retrospective had 4 participants. Even though it was their first retrospective, the team picked up on the retrospective format quickly and there was active discussion almost from the beginning until the end. Notable topics in the discussions were the poor shape of the code base and the project's uncertain future and ownership.

- **Long term goal:** Project has a clear plan for the future.
- **Short term action:** Arrange a "Project future" workshop with stakeholders.

### 4.4.3 Team C

**Team C - 1:** The retrospective had 3 participants in addition to the facilitator. Two participants were new employees who had not participated in the beginning interviews or any other retrospectives. The third participant had participated in one postmortem and one retrospective by this time. The project had started 4 weeks ago and was a subcontracting project for another company. The project had a very tight schedule and a great deal of the discussion was hovering around that topic. The team generally considered internal teamwork to have been very good. Schedule pressure and confusing internal and client communication were seen as problems. The team directed the discussion quickly towards ways of solving both the schedule problem and the communication problem. All agreed that even though it is likely that nothing can be done about this project's schedule, some effort should be put to finding ways to be maximally efficient with the little time available. Interesting note was that team suggested daily stand-up meetings even though they did not have experience in using agile methodologies. The suggestion might have stemmed from theoretical knowledge some of the members had about agile.

- **Long term goal:** Team is focused and in the know all the time so precious time does not get wasted on doing the wrong things.
- **Short term action:** Start having daily stand-up meetings.

### 4.4.4 Team X

**Team X - 1:** All four persons not assigned to a particular project currently participated. The focus of the discussion was on company level and general best practices. The discussion was noticeably less fluid than in the game team retrospectives. After a slow start the flow of the discussion improved. Again, the most common topic was communication. People felt that information does not pass properly between projects and there are even some people in the company who do not know what projects are going on in the company at the time.

- **Long term goal:** Everyone in the company is aware what is happening and when.
- **Short term action:** Start having weekly kick-offs and programmer meetings.

**Team X - 2:** The retrospective included the former Team B members as their update project was completed and the team had been dispersed to work on other projects. A small experiment in the retrospective structure was conducted. The grouping exercise was omitted and instead only exact duplicates were grouped together. The participants were asked to vote on single post-it stickers instead of groups of them. As the group was made of participants from two different previous retrospectives, the previous issues were not added to the board. The group composed solely of programmers and perhaps



for this reason a big theme in the discussion was a worry towards inconsistent technical conventions between projects that make it difficult for programmers to move from a project to another.

- **Long term goal:** All projects use unified technical conventions.
- **Short term action:** Create a proposal for a solution and present to CEO.

**Team X - 3:** The retrospective had 6 participants. Many new participants were present as people had been moved off projects as they were completed. All but two members had participated in a project retrospectives earlier during the study. The retrospective proceeded swiftly as everyone was already familiar with the structure and purpose of the meeting. Many smaller issues were discovered and instead of committing to a singular long term goal, the team wanted to take action on multiple smaller issues. The team agreed to commit to a checklist of smaller improvements instead of a single short term action.

- **Short term actions:**
  - Start production meetings with project leads in addition to weekly meetings.
  - Create an Enter and Exit checklist for employees.
  - Label and list all QA equipment.

**Team X - 4:** The retrospective had 4 participants. Everyone was present who would not participate in the other retrospective that week or in either of the big postmortems scheduled for that week. Recently, an update for a game had been released and an emergency update had to be submitted due to critical errors in the released update build. This was brought up in the discussion and people felt QA processes needed improving. Interesting observation was that this was brought up even though a QA tester was present. It is possible that participants were getting familiar with retrospectives and their purpose as participants proactively assured the QA tester that they did not mean to blame anyone and instead wanted to find ways to improve.

- **Long term goal:** Company has a consistent QA process that is clear to both QA and developers.
- **Short term action:** QA testers of the company will prepare a presentation for management about QA process problems from their perspective.

## 4.5 Postmortem meetings

This section presents a report from both postmortem meetings containing a list of participated personnel, realized schedule as well as written notes taken by the facilitator during the postmortems.

### 4.5.1 Postmortem 1

This project had finished very recently and the game was not yet released to the public. The release date was scheduled to be one week from the postmortem. All except one invited people were present. The one absent team member had an unexpected personal event and was unable attend. In addition to the development team, representatives from the publisher of the game were also present.

Team, date, scheduled duration and participant information of the postmortem can be found from Table 4-2. Realized schedule for the postmortem is shown in Table 4-3. Separate notes for each activity are presented in the following paragraphs.

*Table 4-2 Postmortem 1 metadata*

<b>Team</b>	Team A
<b>Date</b>	6.5.2015
<b>Scheduled duration</b>	6 hours
<b>Participants</b>	<ul style="list-style-type: none"> <li>➤ Publisher <ul style="list-style-type: none"> <li>○ Producer#1</li> <li>○ Producer#2</li> </ul> </li> <li>➤ Developer <ul style="list-style-type: none"> <li>○ CEO</li> <li>○ Producer</li> <li>○ Designer</li> <li>○ Artist</li> <li>○ Lead Programmer</li> <li>○ Programmer</li> <li>○ QA Tester</li> </ul> </li> </ul>

**Introductions & Schedule:** The mood was quickly set to be quite positive when people readily applauded and commented on each other's favorite game selections. The introduction round completed quickly. The activity well went under the scheduled amount of time.

**Create safety:** The participants were presented the rules and the Prime Directive. The safety vote was conducted. Lowest score was 3 and the average was 3,7. This was deemed as good enough to proceed. Later in the postmortem the CEO made a comment regarding the Prime Directive that he never has had any doubts that any employee would not do their best. All participants agreed to this notion. In general, the

*Table 4-3 Postmortem 1 planned vs. realized activity schedule.*

<b>Phase</b>	<b>Activity</b>	<b>Time (planned)</b>	<b>Time (realized)</b>
<u>Readying</u>	<b>Introductions &amp; Schedule</b>	<i>20 min.</i>	10 min.
	<b>Create safety</b>	<i>30 min.</i>	10 min.
<u>Past</u>	<b>Artifacts contest</b>	<i>45 min.</i>	80 min.
	<b>Build the timeline</b>	<i>60 min.</i>	75 min.
	COFFEE BREAK	<i>10 min.</i>	10 min.
	<b>Mine the timeline</b>	<i>60 min.</i>	105 min.
<u>Future</u>	<b>Cross-affinity teams</b>	<i>90 min.</i>	90 min.
	<b>Change the paper</b>	<i>45 min.</i>	postponed
(When necessary)	<b>Form natural affinity groups</b>	<i>5 min.</i>	5 min.

trust level inside the company appears very high and a possible reason for some of the average safety votes might have been the presence of publisher representatives that some people had never met before. This activity took much less time than anticipated because an additional voting round was not needed.

**Artifacts contest:** Participants brought a total of 24 artifacts. Examples were early sketches by artists, early e-mail communication between the publisher and the CEO, schedules, design documents and on the more unusual side a Dungeons & Dragons Adventure manual. Many artifacts raised long discussions about their history and significance. The exercise took nearly double the planned time to go through all the artifacts. The publisher representatives asked questions about artifacts and asked the developer members to reply honestly if they felt so. The discussion became very open at that point and it is likely that the prompt and permission to be honest from publisher side opened the developers to talk about difficult issues that they might not have dared to say out loud in the past. The most significant artifact was deemed to be a quest design document that some of the team members had never seen. The most unusual artifact was a Dungeons & Dragons adventure manual that had been used as inspiration during development.

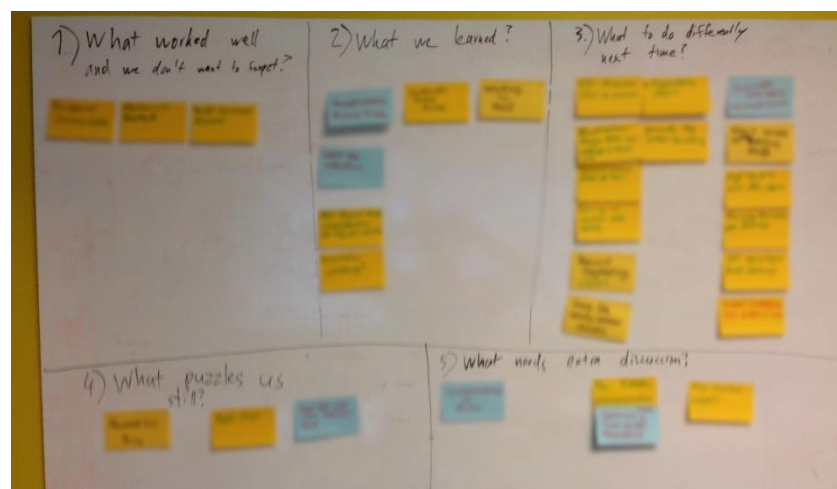
**Build the timeline:** In this activity, much less notes about significant events were posted than expected. The **Happiness Histogram** exercise that was planned to be a brief

sub-activity took much longer than anticipated. Physical space was limited and people struggled recalling their emotional states during the project. A picture of the finished timeline is seen in Figure 4-2. The logo of the game has been redacted and all notes blurred as many of them contained names of the people or identifying components of the game. After building the timeline, there was a 10 minute sandwich and coffee break.



**Figure 4-2** Timeline from postmortem 1 (Game name redacted).

**Mine the timeline:** This activity started out rather slow. People were coming up with topics one at a time and each topic was discussed rather thoroughly before the note was posted on the wall. The timeline was not used as actively as it could have. Occasionally, people referred to the notes and/or the emotional lines as they brought a topic up. Up to this point, some team members had remained fairly silent but by this activity everyone was participating in the discussion. The overall amount of notes was relatively low and most of the notes were in the “What to do differently next time” category. A picture of the results is presented in Figure 4-3.



**Figure 4-3** Results of Mine the timeline from postmortem 1.

**Cross-affinity teams:** By this time, time was running low and the **Change the paper** activity had already been agreed to be done after the retrospective. The two topics that had caused the most discussion so far were publisher-developer communication and schedule risk management regarding milestones. The group was split into two and the two sub-groups both discussed one of these topics to create a presentation about their idea how to solve the problems. The selection of the topics should have been done more carefully, as one of the groups had their presentation ready in 10 minutes whereas the other spent nearly 60 minutes on their presentation.

**Change the paper:** This exercise was agreed to be done after the postmortem by an artist taking all the notes from **Mine the timeline** exercise and creating a poster with art from the game accompanied by the learnings.

In general, the postmortem was relatively successful. Communication was open and honest and there were many findings that positively surprised either the developer, publisher or both. Such findings were for example the fact that when discussing tight schedules the publisher representatives told the developers that they are more likely to consider a longer, more expensive schedule that has proper amounts of buffer time rather than a shorter, less realistic schedule with no room for error.

Low amount of time was clearly a problem, as some topics brought up during the postmortem were left undiscussed and it is possible many more topics could have been brought up if time had allowed it. All of the activities past the Readying phase lasted significantly longer than planned. Activities from the arguably most important phase, Future, were subsequently either shorter than planned or had to be omitted altogether.

## 4.5.2 Postmortem 2

This project had finished some time ago and the game had been available for purchase for a week before the postmortem. Some team members had been working on other projects at least partially for some time already at this point, whereas others were still working on a post-release update that would fix some critical bugs in the game. One team member was not able to attend due to personal reasons, but everyone else invited were present.

As in postmortem 1, team, date, scheduled duration and participant information of the postmortem can be found from Table 4-4. Realized schedule for the postmortem is shown in Table 4-5. Separate notes for each activity are presented in the following paragraphs.

**Introductions & Schedule:** This round went remarkably similarly as it did in Postmortem 1. Favorite game question was again successful in invoking jokes and laughter as people presented their favorite game and bent the one sentence rule in creative ways.

*Table 4-4 Postmortem 2 metadata.*

<b>Team</b>	Team D
<b>Date</b>	7.5.2015
<b>Scheduled duration</b>	8 hours
<b>Participants</b>	<ul style="list-style-type: none"> <li>➤ Customer <ul style="list-style-type: none"> <li>○ IP Owner</li> </ul> </li> <li>➤ Developer <ul style="list-style-type: none"> <li>○ CEO</li> <li>○ Producer</li> <li>○ Artist</li> <li>○ Programmer</li> <li>○ QA Tester</li> </ul> </li> </ul>

*Table 4-5 Postmortem 2 planned vs. realized activity schedule.*

<b>Phase</b>	<b>Activity</b>	<b>Time (planned)</b>	<b>Time (realized)</b>
<u>Readying</u>	<b>Introductions &amp; Schedule</b>	<i>20 min.</i>	5 min.
	<b>Create safety</b>	<i>30 min.</i>	5 min.
<u>Past</u>	<b>Artifacts contest</b>	<i>45 min.</i>	75 min.
	LUNCH BREAK	<i>60 min.</i>	70 min.
	<b>Build the timeline</b>	<i>60 min.</i>	75 min.
	<b>Mine the timeline</b>	<i>60 min.</i>	215 min.
<u>Future</u>	<b>Cross-affinity teams</b>	<i>90 min.</i>	Removed from schedule
	<b>Change the paper</b>	<i>45 min.</i>	Postponed
(When necessary)	<b>Form natural affinity groups</b>	<i>5 min.</i>	Skipped.

**Create safety:** The participants were presented the rules. The Prime Directive was on the wall but the facilitator accidentally omitted reading it aloud. Regardless, the safety level turned out to be very high as the average was 4.56 and the lowest vote was 4. Participants clearly knew each other well and the atmosphere was very relaxed.

**Artifacts contest:** 17 artifacts were presented in total. Only half of the participants brought artifacts. However, as a learning from Postmortem 1, facilitator encouraged the people who did not bring artifacts to think of artifacts they should have brought, explain what they would have been and writing their name on a paper and presenting the paper. This caused two extra artifacts to be presented, one of them being a debug tool developed by a programmer that was extremely useful to the QA tester presenting it. The most significant artifact was voted to be the entire printed list of player reviews from the first week of the game's release. The most unusual artifact was a pile of concept art books from a game with a similar visual style that had been used as reference by an artist.

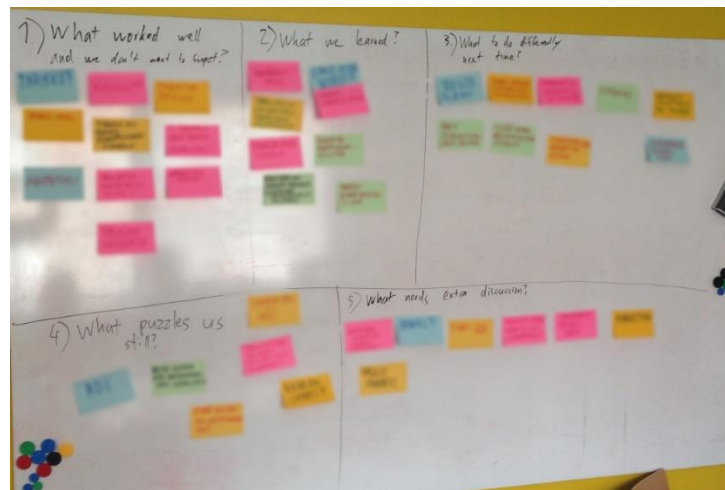
After the artifact contest there was a lunch break. During the lunch there was some additional discussion about the postmortem and the project's past. Interesting observation that came up during lunch was that the customer representative had arrived already the night before and a majority of the participants had spent the previous night together in a restaurant already discussing the project and the future of the collaboration. This was a very likely reason for the relaxed atmosphere and very high safety votes of the postmortem.

**Build the timeline:** Due to the small amount of participants, the affinity team creation was skipped and the team agreed to work as a whole in the same room to build the timeline. Contrary to the early part of the postmortem, this exercise was very silent. People were coming up with events but discussion was relatively sparse and the facilitator had to actively ask questions and point the participants towards using the artifacts. It might have been problematic that there was a break between the artifacts and the timeline building. It is also possible that work in affinity teams could have been more active than the entire team working in one space. Figure 4-4 shows a picture of the finished timeline.



**Figure 4-4** Timeline from postmortem 2 (Game name redacted).

**Mine the timeline:** Contrary to postmortem 1, this activity started out extremely fast. People were writing and posting notes on the wall at a very rapid pace. It was agreed that people would write and post their notes first and then the team would discuss each note. The activity ended up taking the rest of the time allotted for the postmortem as there was a relatively large amount of notes and each note was carefully dissected and even the most trivial ones caused some amount of discussion. Some of the notes caused emotional exchanges between team members, as there had been some large adversities along the project. However, apart from few exceptions where voice levels raised slightly, the exchanges stayed rather neutral. A coffee break was called when the voice levels raised, and afterwards the discussion was steered towards finding ways to prevent the same adversities happening in the future. A picture of the finished exercise can be seen in Figure 4-5.



**Figure 4-5** Results of *Mine the timeline* from postmortem 2.

**Cross-affinity teams & Change the paper:** These activities were cancelled due to a lack of time. The customer was in a hurry to leave and the development team agreed to continue the discussion on another day to ensure that the learnings are documented in some way or another.

Postmortem 2 was clearly more emotionally charged as postmortem 1. It is difficult to say if the more emotional discussion was because the participants knew each other really well or if the project had a more troubled past than the other project. Nonetheless, many concrete findings and learnings were had from this postmortem as well. However, as in postmortem 1, all activities beyond the Readying phase lasted longer than planned and it is possible that even more could have been learned had the overall time reserved for the postmortem been longer. Again, the Future phase suffered from the lack of time and this time both activities had to be completely omitted.



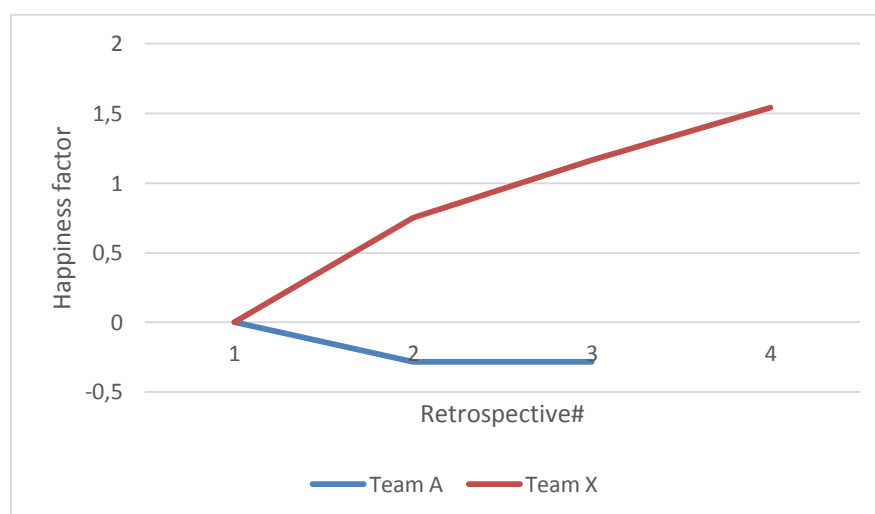
## 5. RESULTS

This chapter summarizes the results from the data. Section 5.1 covers all data gathered from retrospective meetings. This includes some amount of quantitative data as well as facilitator observations. Section 5.2 presents findings from postmortem meetings. Post-mortems did not generate quantitative data, but there were some significant observations. Section 5.3 presents the results from the post-study questionnaire. Both quantitative data and analysis of the open questions are presented. Section 5.4 closes the chapter by presenting possible sources of error in the study.

### 5.1 Findings from retrospectives

This section presents the data collected from the retrospectives. The quantitative data available are the results of the check-in and check-out votes as well as the defined long term goals and short term actions.

Check-in vote data over time are presented in Figure 5-1. Participants estimated in the beginning of each retrospective if they were happier than they were at the previous retrospective. The vote was an informal thumb vote. Thumb up was calculated as vote of +1, sideways was 0 and thumb down was -1. The happiness factor was calculated using a starting point of 0 and adding the average of the votes to the previous week's total. From the figure, it is clear that the data is insufficient to make any strong conclusions. Measurements should be done over a longer time period in a more stable setting. Team X had an upward trend in the happiness factor, but it has to be taken into consideration that Team X was a different group of people in each retrospective.



*Figure 5-1 Check-in vote development over time.*

It is possible that the upward trend was a result of overall improvement in company working environment but no definitive statement should be made. Team A's second retrospective overlapped with a particularly difficult time in the project where the end of the project was approaching and many people were working overtime. The downward swing could be indicative of that. It is also possible that the retrospectives for Team A were not particularly effective in improving the work processes of the team.

The participants were also asked to thumb-vote at the end of the retrospectives if they considered that particular retrospective to have been useful and worth their time invested into it. The results were very clear: out of 45 check-out votes, 44 were thumbs up and only one vote was a thumb sideways. The participant who voted sideways said they felt that so far the biggest advantage of the retrospectives had been that they were a good opportunity for people to express their frustrations but they were unsure if any permanent changes were resulting from the retrospectives. The sole neutral vote happened in the third retrospective of Team A.

A summary of all the long term goals and short term actions in chronological order is presented in Table 5-1. Clear trends in the long term goals were communication issues. Four separate short term actions (2, 3, 7a, 9) were about establishing regular meetings on company or team level. Also, a vast majority of the long-term goals (1, 2, 3, 6, 8 and 9) were related to ensuring that employees have sufficient amounts of information about matters related to the projects or to the company.

A very interesting observation is the results from Team X retrospectives. Team X was the most experimental section of the study and it may have been the most efficient. All Team X retrospectives produced clear goals to improve the company processes as well as tangible short term actions that were mostly taken forward. This has potential implications for the usefulness of retrospectives outside the traditional context.

## **5.2 Findings from postmortems**

Postmortems held during the study provided many valuable learnings even though no quantitative data was collected. Each postmortem produced a long list of learnings categorized into separate subcategories. The full lists of learnings will not be presented in as many of them contain identifying and/or project-specific information. Instead, a summary of findings that were similar in both postmortems is presented.

The first and most important finding was the need for buffer time in scheduling. Both studied projects were subcontracting projects. They can have a problematic dynamic where the developing company, in order to win the bid from the customer, feels like they have to present a condensed schedule and project plan in order to keep the price low. This is very risky as any underestimation in the schedule can lead to either overtime for the developers, reduction in scope mid-project or in the worst case a contract breach when the developer is unable to deliver the promised content in schedule.

**Table 5-1** Summary of all long term goals and short term actions from retrospectives.

#	Team	Long term Goal	Short term action
1	B	The project has a clear plan for the future.	Arrange a "Project future" workshop with stakeholders.
2	A	Everyone in the team is aware what is going on in the project.	Start having weekly status meetings and prioritizing Trello tasks.
3	X	Everyone in the company is aware what is happening and when.	Start having weekly kick-offs and programmer meetings.
4	X	All projects use unified technical conventions.	Create a proposal for a solution and present to CEO.
5	A	Quality of the game does not suffer from schedule pressure.	Present concrete expansion plan and schedule to customer before starting work and do not allow scope changes after locking plan down. Have a postmortem meeting about the project with the customer.
6	A	Expansion team knows necessary things about the project and its situation when they start.	Producer prepares the necessary materials and information and arranges a hand-over meeting after the game is shipped.
7	X	<i>N/A, team decided to take forward three separate short term actions.</i>	<ul style="list-style-type: none"> <li>a) Start production meetings with project leads in addition to weekly meetings.</li> <li>b) Create Exit and Enter checklists for employees.</li> <li>c) Label and list all QA equipment.</li> </ul>
8	X	Company has a consistent QA process that is clear to both QA and developers.	QA testers of the company will prepare a presentation for management about QA process problems from their perspective.
9	C	Team is focused and in the know all the time so precious time does not get wasted on doing the wrong things.	Start having daily stand-up meetings.

The worst possible risks did not realize in either project, but nevertheless they both suffered from overly optimistic schedules. This led to some amount of overtime and reductions in scope. In both postmortems, the developers and customers agreed that there should have been buffer in the schedule throughout the projects. Software projects and game projects in particular are complex undertakings that are extremely difficult to estimate correctly. If there is no buffer in the schedule for unexpected additional work, the unfortunate consequence is often that the overall quality of the game will be lower. Features have to be dropped from the scope and the time originally reserved for polishing the game will have to be instead spent in making the features functional.

Another key finding was that neither project had a dedicated full-time producer. Each project had a producer who was also either a designer, programmer or both. This was not a constant problem in the projects, but there were times when the producers became overloaded and both project management duties as well as their secondary duties suffered. For example, one of the projects had a time period where the story of the game required heavy attention and the entire team was extremely busy. The producer, who was also the main story writer for the game, suddenly had to both write large amounts of story as well as spend more time managing the team and the project. In the postmortem, it was agreed by all that the quality of both responsibilities suffered in the time of overload. The learning was that a producer's job is vital to a game project and while it may not be directly problematic that a producer also has other duties, an effort should be made that it does not come at the cost of their primary work functions.

### **5.3 Post-study questionnaire results**

This section presents the data gathered through the questionnaire at the end of the study. Both open questions and structured questions will be assessed. Open questions will be evaluated mostly in comparison to the responses to the same questions in the pre-study interview. The structured questions that used a Likert scale from 1 to 5 will be presented using a diverging stack bar chart (Robbins & Heiberger, 2011). Multiple choice questions results will be presented using pie charts.

In total, there were 15 employee responses and one publisher representative response to the questionnaire. An average employee respondent had participated in 2,7 iteration retrospective meetings and 0,7 postmortem meetings. Some respondents had not participated in the pre-study interviews as they had started working in the company during the study.

### 5.3.1 Open questions regarding work environment and processes

The questionnaire began by repeating four questions from the pre-study interview. These questions were:

- *What is your perception of the work environment at the company?*
- *Describe your daily work routine*
- *Describe the current working processes your project is using.*
- *Describe the company wide conventions and processes.*

The answers regarding work environment had some interesting trends. In the pre-study interviews words that were used frequently were *silent*, *quiet* or *introverted*. A few responses still repeated this, but overall there was a significant reduction in describing the environment as silent. Some programmers explicitly mentioned that they have a peace to work and there are few disruptions, implying that the environment still is not very noisy. A possible interpretation is that people talk more to each other and thus do not consider the environment that silent any more.

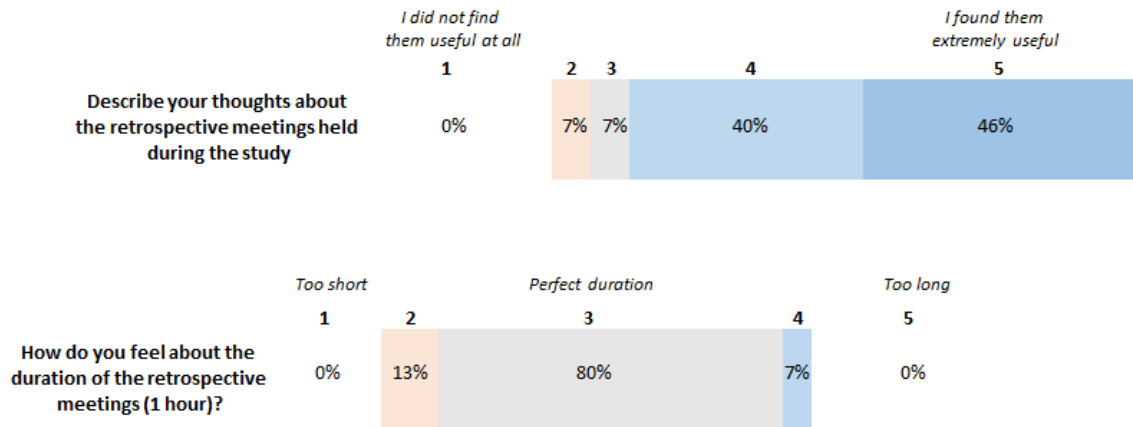
In general, many respondents said that the work environment has improved. Some reasons mentioned were more and improved communication, more decisions and actions and people taking more initiative. Further evidence indicating an improvement in the environment were the responses from new employees. In pre-study interviews, recently hired employees seemed the most reserved towards the environment. Two new employees had started during the study and their responses in the questionnaire were much more positive, one of them having arguably the most positive tone of all the responses.

Several themes from pre-study interviews were still present. Openness, employer freedom and trust were again pronounced. Many respondents mentioned that there are still communication problems. Granted, some of them also said that communications have improved, but clearly there is still need for further improvement.

In Section 5.1 an observation was made regarding the Team X retrospectives that seemingly produced strong goals and actions to improve company-level processes. This observation was confirmed by the questionnaire. In the pre-study interview, employees were unable to name a single company level process or convention used in the company. In the post-study questionnaire however, it was clear that at least part of the short term actions from Team X were taken into regular use as all respondents named weekly company meetings and all programmers named the weekly programmer meeting as company level processes.

### 5.3.2 Questions regarding retrospectives

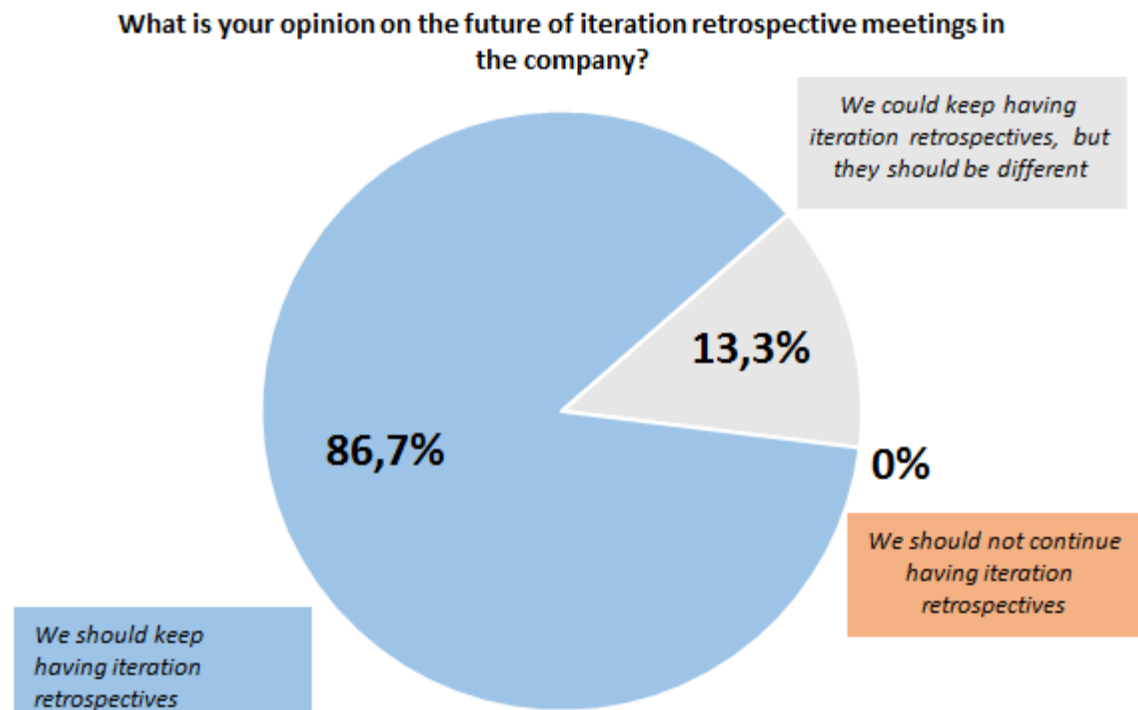
Attitudes towards retrospectives were generally very positive. Figure 5-2 shows that 86% of respondents considered that retrospectives were either extremely useful or somewhat useful. No respondent replied that retrospectives were not useful at all.



**Figure 5-2** Scaled question responses regarding retrospectives.

As for the duration of the retrospective meetings, 80% felt that the 1 hour duration was good for an iteration retrospective. Interestingly, of the people who felt that 1 hour was not the perfect duration, more people (13%) regarded it too short than too long (7%).

As seen in Figure 5-3, a vast majority of employees (86.7%) also considered that retrospectives should be continued in the future. No respondent said that retrospectives should not be continued.



**Figure 5-3** Responses about the future of retrospective meetings.

The questionnaire allowed the respondents to explain why they were uncertain about the continuation of the retrospectives. The main concern was that they felt retrospectives did not yield concrete enough results. The same finding was done earlier in Section 5.1. This is an important finding that must be taken into consideration in the future. Short term actions should be both actionable and have a long term goal that they are meant to advance. One respondent made a good recommendation that the agreed short term actions should be posted in a visible location near the team's working area.

A question about preferred time interval between retrospectives was also presented to all respondents who replied that retrospectives should continue or that they could continue if they were different. Results are seen in Table 5-2. The first row of the table shows the results for all respondents and the second row has been filtered to contain only respondents who participated in three or more iteration retrospectives. This was done to see if attending more retrospectives would have an effect towards the perceived ideal time interval.

*Table 5-2 Preferred time interval between iteration retrospectives.*

**How often should iteration retrospectives be held in your opinion?**

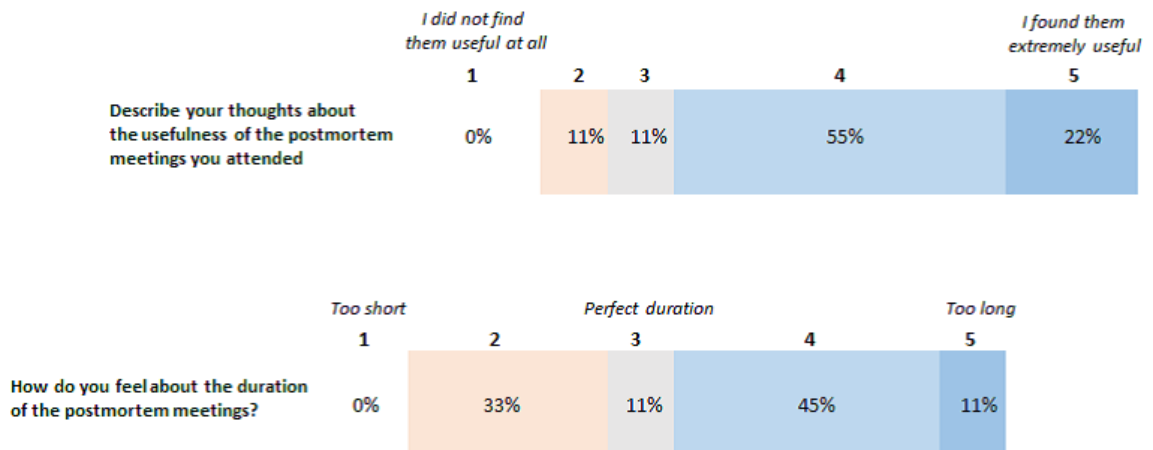
	<i>Weekly</i>	<i>Every 2 weeks</i>	<i>Every 3 weeks</i>	<i>Once a month</i>	<i>Other</i>	<b>Average Interval</b>
<b>All respondents</b>	0%	26,7%	33,3%	26,7%	13,3%	3 weeks
<b>3+ retrospectives</b>	0%	25%	37,5%	37,5%	0%	3,1 weeks

The only clearly significant observation that can be made from the data is that no respondent considered that retrospectives should be held every week. Taking the small sample size into account, the rest of the responses were spread relatively evenly between two, three and four week intervals. Both responses in the "Other" category were from participants who only participated in one retrospective, and they both stated that the interval should depend on project needs. The average intervals in both groups were almost identical.

To summarize, it is safe to say that the vast majority of employees considered retrospectives as useful with 98% approval in the check-out votes and 86% approval in the questionnaire. Effort needs to be made to ensure the issues discovered will be actioned upon and the agreed action points are followed through. Based on the responses on preferred time interval between retrospectives, a recommendation can be made to increase the interval between the retrospectives to three weeks in the future.

### 5.3.3 Questions regarding postmortems

Attitudes towards postmortems were neither as positive nor as aligned as they were in the questions regarding retrospectives. Nevertheless, as can be seen from Figure 5-4, a vast majority (77%) still considered postmortems as useful. As was the case regarding retrospectives, no respondents considered postmortems as not useful at all.

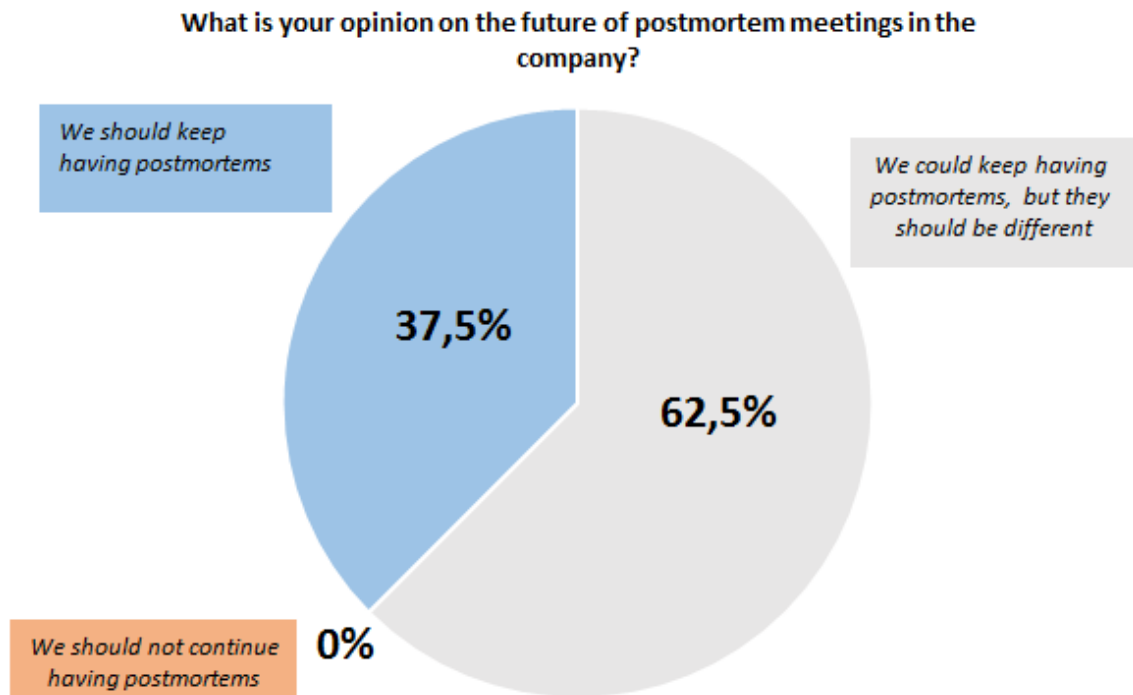


**Figure 5-4** Scaled question responses regarding postmortems.

Opinions regarding the duration were much more diverse than they were in the case of retrospectives. Interestingly, only a small minority (11%) considered the postmortem duration to be perfect. However, adjusting the duration based on this data would be difficult due to the fact that while 56% considered the duration to be too long, 33% considered it to be too short. Taking a deeper look at the data, an interesting observation can be found. All of the respondents who considered the postmortems too short were producers. A possible explanation is that producers are the members in the team who are responsible for maintaining the proverbial big picture in both product and team matters. Therefore, it is likely that producers can participate in all of the conversation in postmortems whereas other team members might feel that some of the discussions are not related to their work. For example, this can lead to the postmortem feeling too long for programmers if most topics discussed are related to art and production issues.

Similar uncertainty could be observed in the responses regarding the future of postmortems in the company as presented in Figure 5-5. This time, only 37.5% of respondents said that postmortems should continue. Even though no respondent said that postmortems should not continue, 62.5% said that they could continue but only if they were different. In the open question asking how the postmortems should be different, there were some noticeable trends. Many people suggested the removal of the Happiness Histogram part as they felt it is hard to remember the emotional states properly. Respondents also felt that the lines drawn were not used enough to justify the exercise. The cross-affinity team exercise received some criticism as well. Some respondents felt it would have been better to discuss what went wrong and what went right with the entire





*Figure 5-5 Responses about the future of postmortem meetings.*

team instead. The preparation that participants were asked to do in the form of artifacts was also considered by some as insufficient individual preparation.

Overall, a conclusion can be made that participants considered postmortems in general to be at least somewhat useful, but the specific structure of the meeting should have been different. The use of postmortems should continue but additional work is required to properly design and conduct postmortems fitting for the company's needs

## 5.4 Possible causes of error

Some potential causes have already been discussed, such as the relatively small amount of both retrospectives and interviewees. This section presents additional observations and conjectures about possible errors in the study and its methods.

First, there was a noticeable difference in the respondents' views regarding postmortems and retrospectives. Retrospectives received a much more positive response with less criticism towards the structure of the meeting. The facilitator of the study had a significant amount of experience in facilitating iteration retrospective meetings but had not designed or facilitated any postmortem meetings in the past. It is very likely that this inexperience led to subpar design of the postmortems which was in turn a major factor in the more critical response to the postmortems. Therefore, making conclusions that postmortems in general would be less appreciated by employees should not be made.

Another potential problem in the data is that none of the data was gathered anonymously. The thumb votes were public in the meetings and both the interviews and question-

naire were not anonymous. It is possible that overly negative criticism was left unsaid as the facilitator was a fellow co-worker in the company. Therefore, it is possible or even likely that the data was skewed towards the positive end of the spectrum. However, as the data about the perceived usefulness of the retrospective meetings was overwhelmingly positive (86% for questionnaire and 98% for check-out votes), it is unlikely that this bias would be large enough to significantly affect the conclusions made from the data.

One observation was that employees considered the work environment to have changed for the better. It should be considered the company had moved to a new office just prior to the beginning of the study. It is possible that the perceived improvement in work environment was influenced by the new office space, as the difference to the old one was very significant. The old office was located in a basement and was very crowded with small amount of personal working space per employee. The new office was very spacious, in a good location and had much better office equipment. In the beginning of the study, there was an effort made to reduce the bias the new office might have caused by reserving a month for people to get accommodated to the new office before the study began. It is still possible that the new office was at least a moderate factor in the observed improvement in work environment.

## 6. CONCLUSIONS

The goal of the study set by the target company was to improve and unify the working processes of the company. This goal was accomplished at least partially as many new conventions were taken into use after they had been discussed in the retrospectives. It should also be noted that some actions that were decided in the retrospectives had been taken forward, but not enough time had yet passed for them to have been noticeable enough in the post-study questionnaire. Retrospectives themselves were generally seen as very useful and nearly all employees felt they should be a permanent company process in the future. Postmortem meetings were considered as potentially useful, but a continued use in the company would require the structure of the postmortem meeting to be redesigned.

The research questions and their evaluations are presented in the following paragraphs.

**RQ1:** *Can retrospective meetings be used separate from other components of agile methodologies to improve working processes and environment?*

Retrospectives served as a consistent method of communication about potential issues within the projects and the company. Employees found them highly useful and they prompted many tangible changes. However, an important observation was that some employees felt that retrospectives did not produce concrete enough results. This should be taken into account if a company looks to add retrospectives into their processes. It is very important to agree to clear action points, ensure that the team remembers them and that they are actually followed through. As long as this is taken into account, it can be said that retrospectives are useful on their own for improving the working processes. As for the working environment, in this company it was clearly improved as a result of the retrospectives. However, it should be noted that the data is insufficient to make a generalized assumption that this would be the case in all companies and contexts.

**RQ2:** *Are retrospective meetings suitable in a cross-discipline game development team context?*

Team A was the only game team that had multiple retrospectives throughout the study and they showed the worst results. However, they did also have the retrospectives during a difficult timeframe where the project was nearing its end. Taking all into account, no conclusion can be made as to whether retrospectives are beneficial in a game team context in the long term. On the other hand, the results from Team X retrospectives which were both cross-discipline as well as cross-team do support the conclusion that

retrospectives can be beneficial outside a traditional software team setting. It also has interesting implications that retrospectives, when properly adjusted, could have applications even in other industries than software.

**RQ3:** *What is the employee perspective towards continuous retrospective meetings?*

This question was the only one where a strong conclusion can be made. Employee reception of the retrospectives was overwhelmingly positive. Even the few employees who were not sure of the overall usefulness of retrospectives did note that retrospectives would have some value even if they were only occasions to express frustrations in a safe way. A consideration for future application in the company should be to include the section at the end of the retrospective where the team briefly discusses their ideas on how to improve the retrospectives themselves.

In order to draw stronger conclusions, future research into the topic is recommended. A long-term study throughout a course of an entire game project using retrospective meetings could help solidify the aspects of the study that were left inconclusive.

Based on this study, a recommendation can be made for game development companies in similar situations to consider adding retrospective meetings as a part of their company processes. Employees are the most valuable asset a game company has and retrospectives demonstrably have a positive effect on the employees and their perspective on the company work environment.

## REFERENCES

- Agile Retrospective Resource Wiki, 2013. *Everyday Retrospective*. [Online] Available at: [http://retrospectivewiki.org/index.php?title=Everyday\\_Retrospective](http://retrospectivewiki.org/index.php?title=Everyday_Retrospective) [Accessed 20 Feb 2015].
- Al-azawi, R., Ayeshe, A. & Al. Obaidy, M., 2014. Towards Agent-based Agile Approach for Game Development Methodology. Hammamet, Tunisia, 2014. IEEE.
- Bethke, E., 2003. *Game Development and Production*. Plano, TX: Wordware Publishing, Inc.
- Brydon-Miller, M., Greenwood, D. & Maguire, P., 2003. Why action research? *Action Research*, 1(1), pp.9-28.
- Coplien, J. & Østergaard, J., 2009. *Scrum: It's all about common sense*. [Online] Available at: [http://silab.fon.rs/index2.php?option=com\\_docman&task=doc\\_view&gid=238&Itemid=5](http://silab.fon.rs/index2.php?option=com_docman&task=doc_view&gid=238&Itemid=5).
- Derby, E. & Larsen, D., 2006. *Agile Retrospectives: Making Good Teams Great*. Dallas, TX: The Pragmatic Bookshelf.
- Dictionary.com, 2015. "*retrospective*" in *Dictionary.com Unabridged*. [Online] Available at: <http://dictionary.reference.com/browse/retrospective> [Accessed 25 April 2015].
- Fowler, M. & Highsmith, J., 2001. *The Agile Manifesto*. [Online] Available at: [http://andrey.hristov.com/fht-stuttgart/The\\_Agile\\_Manifesto\\_SDMagazine.pdf](http://andrey.hristov.com/fht-stuttgart/The_Agile_Manifesto_SDMagazine.pdf).
- Keith, C., 2013. *Agile Game Development with Scrum*. New Jersey, NJ: Addison-Wesley.
- Kerth, N.L., 2001. *Project Retrospectives: A Handbook for Team Reviews*. New York, NY: Dorset House Publishing.
- Likert, R., 1932. A technique for the measurement of attitudes. *Archive of Psychology*, 22(140), pp.5-55.
- McGuire, R., 2006. *Gamasutra - Paper Burns: Game Design With Agile Methodologies*. [Online] Available at: [http://www.gamasutra.com/view/feature/131151/paper\\_burns\\_game\\_design\\_with\\_.php](http://www.gamasutra.com/view/feature/131151/paper_burns_game_design_with_.php) [Accessed 6 April 2015].

Merriam-Webster.com, 2015. "game" in *Merriam-Webster: Dictionary and Thesaurus*. [Online] Available at: <http://www.merriam-webster.com/dictionary/game> [Accessed 13 May 2015].

Merriam-Webster.com, 2015. "retrospective" in *Merriam-Webster: Dictionary and Thesaurus*. [Online] Available at: <http://www.merriam-webster.com/dictionary/retrospective> [Accessed 25 April 2015].

Moore, M.E., 2011. *Basics of Game Design*. Boca Raton, FL: Taylor & Francis.

Nolan, A.J., 1999. Learning from Success. *IEEE Software*, pp.97-105.

Novak, J., 2012. *Game Development Essentials*. 3rd ed. Clifton Park, NY: Cengage Learning.

Petrillo, F. & Pimenta, M., 2010. Is agility out there?: Agile practices in game development. New York, 2010. ACM.

Petrillo, F., Pimenta, M., Trindade, F. & Dietrich, C., 2008. Houston, we have a problem.: a survey of actual problems in computer games development. Fortaleza, Ceara, Brazil, 2008. ACM.

Reason, P. & Bradbury, H., 2008. *Handbook of Action Research*. 2nd ed. London: SAGE Publications.

Robbins, N. & Heiberger, R., 2011. Plotting Likert and Other Rating Scales. Philadelphia, 2011. Joint Statistical Meetings.

Rollings, A. & Morris, D., 2004. *Game Architecture and Design*. Indianapolis, IN: New Riders Publishing.

Royce, W.W., 1970. Managing the development of large software systems: concepts and techniques. In *Proceedings of the 9th international conference on Software Engineering (ICSE '87)*. Los Alamitos, CA, 1970. IEEE Computer Society Press.

Skype/Microsoft, 2015. *Skype*. [Online] Available at: <http://www.skype.com/> [Accessed 15 May 2015].

Stacey, P. & Nandhakumar, J., 2008. Opening Up to Agile Games Development. *Communications of the ACM*, 51(12), pp.143-46.

Takeuchi, H. & Nonaka, I., 1986. The new new development game. *Harvard Business Review* 68, pp.137-46.

Tozour, P., Wegbreit, D. & Parsons, L., 2015. *The Game Outcomes Project, Part 1: The Best and the Rest*. [Online] Available at: <http://www.gamasutra.com/blogs/PaulTozour/>

20150113/233922/The\_Game\_Outcomes\_Project\_Part\_3\_Game\_Development\_Factors.php [Accessed 4 May 2015].

Trello, Inc., 2015. *Trello*. [Online] Available at: <https://trello.com/> [Accessed 16 May 2015].

Turner, J. & Danks, S., 2014. Case Study Research: A Valuable Learning Tool for Performance Improvement Professionals. *Performance Improvement*, 53(4), pp.24-31. Available at: <http://onlinelibrary.wiley.com/doi/10.1002/pfi.21406/pdf>.

van de Weerd, I., de Weerd, S. & Brinkkemper, S., 2007. Developing a Reference Method for Game Production by Method Comparison. In Ralyte, J., ed. *Situational Method Engineering: Fundamentals and Experiences*. Geneva, Switzerland, 2007. Springer Science.

Williams, L. & Cockburn, A., 2003. Agile Software Development: It's About Feedback and Change. *Computer*, pp.39-43.

Woodside, A.G., 2010. *Case Study Research: Theory, Methods and Practices*. Boston, MA: Emerald Publishing Ltd. Available at: <http://site.ebrary.com/lib/helsinki/reader.action?docID=10400678>.