Agile Game Development: A Systematic Literature Review

Henna-Riikka Ruonala

Pro gradu
UNIVERSITY OF HELSINKI
Department of Computer Science

Helsinki, November 22, 2016

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Laitos — Institution — Department
Faculty of Science	Department of Computer Science

Tekijä — Författare — Author
Henna-Riikka Ruonala

Työn nimi — Arbetets titel — Title

Agile Game Development: A Systematic Literature Review

Oppiaine — Läroämne — Subject
Computer Science

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Pro gradu	November 22, 2016	50

Tiivistelmä — Referat — Abstract

A systematic literature review was conducted to examine the usage of agile methods in game development. A total of 23 articles were found which were analysed with the help of concept matrices. The results indicate that agile methods are used to varying degrees in game development. Agile methods lead to improved quality of games through a prototyping, playtesting, and feedback loop. Communication and ability of the team to take responsibility are also enhanced. Challenges arise from multidisciplinary teams, management issues, lack of training in agile methods, and quality of code.

ACM Computing Classification System (CCS):
Software and its engineering Interactive games *Software and its engineering Agile software development*

Avainsanat — Nyckelord — Keywords
game development, agile, development process, project management, multidisciplinary teams

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Contents

1 Introduction

As the games industry has grown there has been increasing interest in how games are actually developed. The processes have gone from ad-hoc through rather strict waterfall to more agile methods.

The industry itself has plenty of guidebooks available, but somewhat less academic research has tackled the issue. Studies have been done in different research fields, depending on whether game development is seen as software engineering or as a creative industry. Game development seems to be a somewhat inherently agile activity and many game studios have indeed adopted agile practices [A5].

The aim of this pro gradu is to examine agile practices within game development through a systematic literature review. How widely they are used, what benefits they offer, and what challenges there may be in adopting them within the games industry.

Agile methods are always iterative and embrace change, communication, and feedback [10]. Scrum and XP are popular agile methodologies and they seem to be commonly practiced at game studios as well [A5]. XP promotes sustainable pace of development meaning that workers should not be pushed over their limit with overwork. As crunch time is one of the problems in game development, XP could offer a philosophy to counter the current practice.

The games development process has traditionally been seen as a very waterfall-like process, and certainly many guidebooks and industry habits support this view. Nevertheless, when studying the differences between game development and other software engineering, it has often emerged that game development has rapidly changing requirements and iterative development methods [A6, A1]. Game development is also more of a multidisciplinary effort than general software engineering, with graphics and audio artists and game designers involved in the process.

Game development is thus a curious combination of many disciplines. The programming itself is a similar activity as in any software engineering enterprise, though it can be harder than e.g. developing office software, as games can need physics engines and artificial intelligence among other things. Games are commonly rather heavy on graphics, which means artists have to create concept art, 3D models, textures, etc. Many different stakeholders are involved, including producers and publishers. Many studios nowadays are able to self-publish [A11] but especially with bigger games and studios publishers are another source of demands, checks on the quality of the game and even the manner of development. These things all affect the process chosen, especially in the case of agile development, where managers have to trust the team to take on more responsibility.

Game development has been recognised as having characteristics not present in other software engineering [A1]. One difference is that game development is more prone to changing requirements. Thus agile methods would

seem to be a good fit for the games industry. Despite these differences some studies only approach game development from a purely software engineering perspective. This can perhaps cause problems in gathering a coherent and full view of games research.

It has been a commonly held belief that games are developed in a waterfall-like manner, or the development process is rather ad hoc. On the other hand especially modern online and mobile games are more like services and are developed in a more agile way. In any case, game development follows a certain succession of phases. These phases are usually thought to include a concept phase, preproduction phase, production phase, and postproduction. The traditional view of this process can indeed be rather waterfall-like [14].

The concept phase is where the game concepts and ideas are developed. Depending on the title, this initial concept may come from e.g. the publisher and be more of a business decision than an innovative gameplay concept.

Preproduction in this waterfall model is where the game design is refined and the game design document (GDD) is created. Often also the technical design document (TDD) is produced, requiring some knowledge already of the technical challenges the project will face. Preproduction is thus traditionally the planning phase of game development.

Production is where the game gets built, starting from the ground-laying mechanics enabling gameplay. Art assets and sound are also produced in the production phase. In this waterfall manner of development, it may take a while before a first playable demo is produced. Alternatively prototypes are produced to test ideas.

Postproduction is the time for testing and final fixing of problems. It is important to avoid too drastic changes near the end of the project.

Keith's agile model of game development [6] takes a somewhat different approach to the different phases. The phases remain the same but the importance of testing out ideas early on is emphasised. It is preferable to remain in preproduction until the fun of the game is found. Because even the definition of fun is evasive, it is difficult to predict how long preproduction will take. In agile game development, mechanics and coding is already done in preproduction, and prototypes are produced so that the gameplay can be tested to ensure fun. It is important to reduce uncertainty about mechanics and the technology and tools used. Production is then the time to add content and depth, using the mechanics produced in preproduction.

As complex projects, game development efforts are prone to face certain problems. The most common problems are well known in the industry and some studies have been conducted to gather knowledge of the frequency of these issues. Petrillo et al. went through game project postmortems and found that problems with project scope and feature creep were the most prevalent [13]. Many game development projects also suffer from what is called 'crunch time' at the end of the project partly because of feature creep and unrealistic demands as release gets nearer [4].

The question is whether agile methods and practices can help to alleviate these commonly experienced problems. Keith lists some benefits of using agile practices for game development, including finding the fun i.e. value early and gathering knowledge along the way [6].

Previously there has been a lack of research into game development processes but the pace of studies being published has picked up in the past years [11]. There has also been some research into agile practices within game development but there are no systematic literature reviews on that exact topic yet.

This thesis will use these previous studies to explore how far spread agile methods are within the games industry and how their use can impact the development process. As game development differs somewhat from general software engineering, possible problems with adopting agile practices are also explored. Do agile methods need to be specifically tailored for game development? Are certain phases of game development more suitable for agile development?

Chapter 2 will give some more background on agile methods, discuss the similarities and differences between game development and software engineering, and explain the game development process and different stakeholders involved. Previous work related to the topic of this thesis will be presented in chapter 3. Chapter 4 presents the current review questions and methods, and results are given in chapter 5. Discussion on the findings and their meaning can be found in chapter 6, and concluding remarks in chapter 7.

2 Background

This section will provide some background information and context needed to understand the analysis of the review articles.

2.1 Agile

Agile methods promote collaboration, self-organising, cross-functional teams, and frequent delivery of business value. The agile manifesto[1] sees more value in individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan.

Agile development is always iterative, incremental and thus evolutionary [10]. It uses iterations to incrementally build more value into the product. Popular frameworks and methods for agile development include Scrum and XP. These are often combined, as Scrum is more of a high-level management framework and XP consists of more hands-on practices.

In Scrum, the iterations are called sprints. Daily meetings called daily scrums are had, where the team updates each other about the progress they have made as well as any impediments to their work that others might help with. Sprint planning meetings, sprint reviews and retrospectives at the end of a sprint are other types of Scrum meetings for planning the next sprint and reflecting on the previous one. Burn-down charts are used during the sprint to track down tasks and time left before the end of the sprint. This gives a visual representation of whether the goals for the sprint are achievable or if some tasks perhaps need to be dropped. Tasks are picked up during the sprint planning meeting from a product backlog, which is a prioritised list of features or requirements, often in the form of user stories. The items in the backlog are evaluated in terms of how much effort and time they require, and if chosen for the sprint, are then added to a sprint backlog.

XP values communication, simplicity, feedback, courage, and respect. It promotes a sustainable pace of work and includes practices like pair programming, test-driven programming, and continuous integration.

Lean development is often seen in conjunction with agile. It emphasises eliminating waste, learning, fast delivery, and empowering the team among other things.

2.2 Game development vs. software engineering

The games industry is in a curious place in that it can be seen and has been studied from a purely software engineering perspective, or from a creative industry perspective, or as a combination of the two. There are indeed

[1] `http://agilemanifesto.org/`

studies trying to pinpoint the differences between software engineering in general and game development.

Murphy-Hill et al. confirmed some differences between game development and other software development [A6]. Their findings included a more agile process, less clear requirements, diverse teams, and creativity being valued more in game development. Their study included interviews and a survey among Microsoft employees to confirm interview findings. In another study, game developers, especially game designers, seem to mostly view making games as creative work but curiously managers are more likely to think of it as mainly software development [A4].

Kasurinen came to the same conclusion that game development indeed is somewhat different from other software engineering [A1]. Compared to traditional software engineering, the requirements of games can be rather abstract, with user enjoyment defining the quality of the game. With user enjoyment, or fun being the aim, there also comes the attempt to maximise rather than minimise the time used with the product.

Games as entertainment strive to make the players have fun. Fun is a non-functional requirement and as such can be a difficult to specify and keep track of. Game designers should attempt to find the fun of the game before the design is locked down, as it can be very costly to find that a game is almost finished but not fun. Flow is another concept often associated with games and fun. This flow state and fun are related to maximising the time used playing the game, which is again a difference to other software [A1]. Immersion is another term often used for the ideal state of the player when playing the game.

Game development is a multidisciplinary effort. It involves programming, art (both concept art and production graphics), audio, script writing and gameplay design. Big studios might have their own teams for these different disciplines, and smaller ones might outsource some of them, such as audio. Game studios are rather varied regarding size, the type of games they produce, if they are an independent studio or owned by a bigger corporation, if they self-publish or have a deal with a publisher, etc. All of these things may influence the process they use to develop a game.

There seems to be tension between the need to ensure business goals (leading to conservative decisions) and to innovate and be creative [9, 16]. At the beginning of the games industry, innovation was more highly valued and as games were more small-scale, individual developers were given more freedom to create. As the business has grown, big publishers have put more pressure on developers to make predictable hit games. This trend may be changing now with more possibilities for smaller studios to self-publish through digital publishing sites like Steam[2] or GOG.com[3].

[2] `http://www.steampowered.com/`

[3] `http://www.gog.com/`

2.3 The game development process

Game development has traditionally been project-based work. As explained in the Introduction, games are produced in four phases. These phases are usually thought to include a concept phase, preproduction phase, production phase, and postproduction. Additionally an aftermarket phase can be added. A lot of games these days also require regular maintenance after release, especially massively multiplayer online games and mobile games with constantly updated content. The traditional view of this process that has been presented in many guide books for the industry has been rather waterfall-like [14].

The concept of the game is developed in the concept phase. The initiative for a game title may come from a publisher, in which case it may be a very business oriented decision. The publisher may have a creative team to work on the concept or they may consult an external game designer. For a small start-up company, especially one that self-publishes, this process is of course very different, and the amount of independence during the whole development process is greater [A11].

In the waterfall model, preproduction is where the game design is refined and the game design document (GDD) is created. The GDD is written by the game designer and gives a detailed description of the gameplay, and depending on the type of game, characters etc. There is no one template for a game design document, as each game is of course different. Callele et al. warn of using the GDD as a requirements document for production [3]. The GDD is still meant to only convey the idea of the game and the requirements are presented in a separate document meant specifically for production purposes. Playable demos and prototypes are another option to test concepts and ideas, and for example Keith recommends that productions starts only once uncertainty about the mechanics and the technology have been reduced [6]. In this agile approach to game development it is preferable and important to remain in preproduction until the fun of the game is found. This cannot be done without actual playable prototypes, as has been found e.g. at Massive Entertainment as well [B10]. The different stakeholders, e.g. the publisher put their own demands on this process in the form of milestones and deadlines.

Winget and Sampson found in their study on game development documentation that playable prototypes are nowadays considered essential [19]. Their interviews also indicated that 'builds are fundamental to creating the "fun" element of the game'. This would seem to correspond nicely to agile development.

Preproduction has traditionally been the planning phase of game development. The technical design document (TDD) is produced in preproduction, containing a preliminary plan for the architecture and the technical aspects of the game [B10]. Production ideally begins when the ideas in preproduction

are sufficiently fleshed out. In a more traditional approach, where preproduction is purely a planning stage, production is where the game engine begins to be built, and for the first stages of production there may be no actual playable game to show [14]. Milestones are nevertheless agreed upon with the publisher, and progress is regularly checked. In Keith's model of agile game development, the core mechanics are preferably locked down already in preproduction, so production is the time to expand upon the gameplay and add content [6]. The phases of game development are less strictly separated in a more agile approach. Preproduction and production activities can for a while run in parallel.

Postproduction in the waterfall model is the time for testing, final fixing of bugs, and polishing. In an agile model, the testing is continuous and done throughout preproduction and production but some specific hardware testing for example has to be left till the end of the project. Usability testing is the most widely used testing technique, as it is important to find out if the game mechanics work and if players are enjoying the game [A4]. User playtesting is also useful for finding out if the difficulty of the game is appropriate.

In any case, in both waterfall and agile development models, it is essential to protect the project from drastic changes at this stage to avoid running over schedule and budget. Game development has been more tolerant of even late changes than general software development though [A4].

2.4 Problems in game development

Petrillo et al. found that unrealistic scope, and both feature creep and cutting out features, were the most prevalent problems in the game development postmortems they studied [13]. Crunch time was reported in almost half the postmortems, in 45%, which was found to be fairly low compared to the games industry reputation of often demanding overwork. They mention communication between teams as a problem specific to the games industry (mentioned in 35% of the postmortems), arising from multidisciplinary teams with programmers and artists involved. More vague requirements like fun are also specific to game development. Overall they concluded that the problems in the games industry are managerial rather than technical in nature. All the postmortems they studied were of course of completed games, showing that problems arise even in successful projects, but offering no knowledge of the problems faced by unsuccessful ones.

The International Game Developers Association conducts surveys on developer satisfaction, gathering knowledge of e.g. crunch time. Even though most respondents of IGDA's 2014 Developer Satisfaction Survey agree that crunch is not inherently necessary in game development, it is nevertheless still common; only 19% reported not having had to do crunch time during the past two years [4]. The most common reasons for crunch time were an unrealistic schedule and feature creep. Inexperienced management was

mentioned as well.

Washburn et al. have also taken a look at game development problems and best practices through a postmortem analysis [17]. Iterations and prototypes were seen to help the development process, whereas inadequate preproduction caused problems in production as well. The most frequent problems observed in the postmortems were related to obstacles, schedule, and development process. On the other hand development process was often also one of the things that went right, along with game design and team.

Keith also lists feature creep, unrealistic scope, and challenges of production as the most pertinent problems in games industry [6]. Production attempts "to maximize efficiency, minimize waste, and create predictability". The challenges of production are mitigated by good preproduction.

Scheduling problems often arise from a need to set a hard deadline on the release. Christmas sales are the preferred release target, and advertising slots for a big title are bought well beforehand, so there is pressure for possible crunch time before release from the publisher and marketing.

The problems of a waterfall approach are well known nowadays. As game development differs somewhat from other software engineering, how easy is it to apply agile methods within the games industry?

3 Previous work

O'Hagan et al. conducted a systematic literature review of research on software development processes used in game development [11]. This was a wider topic not specifically focused on agile practices. They included quantitative and qualitative studies of both industry and non-industry. They were able to identify 356 processes which they grouped into 23 process models belonging either to an agile or hybrid (combining traditional and agile processes) approach. They found there were somewhat fewer industrial studies and only 9% of these studies mentioned purely agile processes, the rest were hybrid. They categorised industrial as research "performed in collaboration with or embedded in industry" which included not only game studios but also e.g. health care professionals and the US army.

The study covered software process improvement (SPI) initiatives in games development as well and also looked into what factors affect the adoption of software process models or SPI in the industrial studies. Functional prototypes are mentioned, as well as training, right people in the team, and quality, among others.

Another literature review by Ampatzoglou and Stamelos attempted to find out the extent of game development research up until 2009 [2]. The analysed articles were classified into categories according to the ACM Computing Classification System. This review did not include process as a category itself but did include studies involving the timeline of the development process in a category called Management. They also mention papers exploring agile programming methods in game development. The review concluded that requirements engineering was the most popular topic for games related software engineering research.

An article by Kanode and Haddad came to the conclusion that agile methods may be useful especially in preproduction for faster prototyping of game ideas [5]. An agile approach is seen to enhance creativity. In the production phase they suggest using a more traditional approach like the spiral model. Their suggestions are based on review of the literature, including the book on agile game development by Keith [6].

Koutonen and Leppänen have examined similar questions of agile practices and their impacts through an online survey [A5]. They found that agile practices are quite frequently used, with positive results. They asked companies about their use of agile methods in general and also which Scrum, XP, and Kanban practices and principles are used. Scrum and XP practices were both often used. Daily Scrum, sprints, cross-functional teams, informative work spaces, and continuous integration were the most used among Scrum and XP practices. The study revealed differences between the different phases of the game development process, with Scrum practices used more often in preproduction and production. XP however was more evenly used throughout the process. Kanban was rarely used. The findings of this

study are further examined in the Results section as this paper was included in among the articles for analysis in this thesis.

4 Review questions and methods

The aim of this thesis is to gather an understanding of how agile methods are used in the games industry, what benefits they may offer, and what challenges can arise when adopting them. This is done through a systematic literature review [8] and subsequent concept analysis [18] of the articles found. The research questions are:

1. How widespread is use of agile methods in game development?
2. To what degree are agile methods used within game studios?
3. What problems do agile methods solve in game development?
4. What challenges are there in using agile methods in game development?

The aim is to explore how agility in game development can help to solve certain problems, but also to see if there are occasions where it is not the best method to use, or if it needs to be adapted specifically to game development.

The starting point is a search for

```
TITLE-ABS-KEY((agile OR scrum OR kanban OR xp
OR "extreme programming")
AND ((game* OR gaming OR videogam*)
AND (develop* OR process* OR method*)))
```

in Scopus[4] and similar search string under `Topic` in Web of Science[5]. This searches within the title, abstract, and keywords of the documents. The search (done June 10th 2016) yields 341 results in Scopus and 186 results in Web of Science. Of those 332 in Scopus are in English, and 180 in Web of Science. These were filtered manually and then snowballing technique [20] for the English language articles was used to yield further relevant articles within Google Scholar[6]. The filtering was done based on the research questions and the following criteria:

1. Related to game industry
2. Describing current practices
3. Original research offering new results (survey, case study, postmortem analysis...)

[4] `https://www.scopus.com/`
[5] `https://webofknowledge.com/`
[6] `https://scholar.google.fi/`

Articles related to serious gaming were also excluded, as these often have different kinds of stakeholders compared to game development of games for entertainment. All platforms and genres for entertainment games were included however, as many studies themselves did not make a difference between them and excluding a certain genre for example would have unnecessarily narrowed down the selection of papers. This however does not mean that there is not a difference in the development process. A continually updatable mobile puzzle game presents different challenges of development compared to a big first person shooter title.

Table 1: Number of articles after each step.

	Scopus	Web of Science
before filtering (in English)	332	180
after filtering	8	7
after removing duplicates	9	
after 1st round of snowballing	15	
after 2nd round of snowballing	21	
after 3rd round of snowballing	22	
after checking other literature review [11]	23	
after new snowballing	23	

There were a couple of articles suggesting improvements on the game development process, and these were excluded as well as they had not been tested in practice. One improvement article was included though as it referred to previous data of an online survey done among game industry, and had consulted industry experts for opinions on the improvement suggestions [A8].

As can be seen from Table 1, this filtering yielded 9 papers when duplicates from the two databases were removed. After snowballing, the number rose to 22. As the results were compared to the previous systematic literature review on game development processes [11], it was found that the search had missed one paper which did include information on agile processes in the games industry. This article was added to the set, and snowballing was done for it but no further results were found. Finally, the number of papers for the analysis was 23.

	game development vs software engineering	artists vs engineers	agility in team vs agility in organisation	creativity	creative industry	multidisciplinary	team	management	organisation	stakeholders	publisher	autonomy	empowerment	control	leadership	power hierarchy	training/ knowledge of method	communication	collaboration/ cooperation	coordination	synchronisation	quality	code quality	feature creep	crunch time	over/underproduction/ workflow problems	flexibility	productivity	transparency	confidence	courage	responsibility	article sum
How Are Agile Methods & Practices Deployed in VGD?				1	1		1	1										1				1	1	1	1							1	10
A Survey on the State of the Practice in VGD	1				1	1																1		1	1	1							7
Improving VGD: Facilitating Heterogenous Team …				1		1	1												1					1	1								6
Cowboys, Ankle Sprains, and Keepers of Quality	1	1		1		1						1						1				1					1						8
Risk Management in VGD Projects				1	1			1			1									1							1						6
How Applicable is ISO/IEC 29110 in GD?				1				1		1	1																						4
Is Requirements Engineering Useless in GD?				1	1																												2
What Do Game Developers Test in Their Products?				1	1																	1	1										4
Games as Software – Similarities & Differences …	1				1																	1											3
Is Agility Out There? Agile Practices in GD						1																											1
Are the Old Days Gone?								1																									1
Towards an Understanding of Game Software Processes				1			1	1																						1		1	5
Agile & Kanban in Coordination							1													1	1						1	1					5
Bootstrapping Scrum & XP under Crisis							1		1	1							1	1		1	1	1	1		1								10
Organizational Enablers for Agile Adoption …			1			1		1		1		1	1		1	1	1					1	1										11
Virtual Reality Meets Scrum ...			1				1	1	1					1	1		1	1	1		1	1					1	1	1			1	15
Controlling the Uncontrollable …		1	1	1	1	1	1	1	1		1	1		1		1		1	1	1							1					1	17
Real-Time Strategy: Evolutionary Game Development			1	1				1			1							1	1			1		1									8
Management of Creativity in VGD			1	1			1	1			1				1			1												1	1	1	10
Creativity in the Video Game Industry			1	1			1		1		1			1					1	1		1											9
Managing Projects in a Games Factory: Temporality …		1				1		1				1	1			1	1	1			1			1		1		1					12
A Temporal Perspective of the Computer GD Process				1			1	1				1		1		1	1									1		1				1	10
Opening Up to Agile GD					1		1	1	1	1								1													1		7
sum	3	3	6	13	8	7	11	13	5	4	6	5	2	4	3	4	5	9	5	5	4	10	4	5	4	3	5	4	1	2	2	6	
sum by study	3	3	4	9	6	6	9	10	5	4	4	4	2	4	3	3	4	7	4	5	4	8	4	4	3	2	5	3	1	2	2	6	
sum of surveys	3	1	0	7	6	4	2	4	0	1	2	1	0	0	0	0	0	2	1	1	0	5	2	3	3	1	2	0	0	0	0	1	
sum of surveys by study	3	1	0	5	4	3	2	4	0	1	2	1	0	0	0	0	0	2	1	1	0	4	2	2	2	1	2	0	0	0	0	1	
sum of case studies	0	2	6	6	2	3	9	9	5	3	4	4	2	4	3	4	5	7	4	4	4	5	2	2	1	2	3	4	1	2	2	5	
sum of case studies by study	0	2	4	4	2	3	7	6	5	3	2	3	2	4	3	3	4	5	3	4	4	4	2	2	1	1	3	3	1	2	2	5	

Figure 1: An example of a concept matrix used for the analysis of the articles.

Once all the papers had been found, they were read more thoroughly and at the same time potential concepts were marked down. After all the articles had been examined, the concepts were gathered together and grouped according to themes. Concept matrices were constructed based on these groupings. An example of a concept matrix used can be seen in Figure 1. The matrix was examined for the frequency with which concepts appeared in the surveys and case studies, and the highest frequency concepts were taken under examination. It was noted that certain concepts were linked and formed groups, and these were examined at the same time, giving a fuller picture for answering the research questions.

5 Results

This chapter will detail the results of the search and the analysis of the papers found. The articles are first listed grouped into surveys and case studies. A brief description of the study is given, and the papers discussing the study are listed below. Then the research questions are examined. Concept matrices helped to focus the analysis on the relevant concepts. Some concepts such as quality had to be further broken down into quality of the games themselves on the one hand, and quality of code on the other, as the findings differed regarding them.

The articles included both general survey type studies and case studies. The number of both was rather balanced, with 7 general studies presented in 11 articles, and 8 case studies presented in 12 articles. The same study material was sometimes used in several papers, sometimes focusing on different topics. As can be seen from Table 2, the general studies were either online surveys (4), postmortem analyses (2), or interview studies (3). Interviews were combined with online surveys in two studies. Table 3 gives an overview of the case studies, which included experience reports (2), questionnaires (1), interviews (1), and a combination of both interviews and observation (1). Among the genres of games in the case studies were real-time strategy, 1st person shooter, online poker, casino slot machines, and virtual worlds. Games were developed for PC, mobile, and hand-held consoles.

Table 2: Surveys.

articles	method/type	studios from
[A8], [A7]	online survey	Austria
[A5]	online survey	Finland
[A2], [A3], [A4]	interviews	Finland
[A1]	interviews, survey	Finland, multiple
[A6]	interviews, survey	?, Microsoft
[A11]	interviews	?
[A9]	postmortem analysis	multiple
[A10]	postmortem analysis	multiple

Some themes were recognised while going through the articles. Some of the general studies compared game development with software engineering, and some case studies mentioned tension between artists and engineers. The difficulties of working as an only agile team in a non-agile organisation were apparent, while success stories were told when the whole organisation was committed to agile.

The articles were written from different perspectives on a continuum of whether game development is a creative industry or only software engineering, or something in between. Most of the articles were done within the field

Table 3: Case studies.

articles	method/ type	platform	genre	employees	studio from
[B2]	interviews, observation	handheld console	sequel	200	Canada
[B1]	questionnaire	online, mobile	virtual world	300 (80 in Scrum teams)	Finland
[B4]	interviews	online, mobile	massively multiplayer online game	5 (startup)	Ireland
[B7], [B8]	interviews	PC, mobile	multiplayer 1st person shooter	20	Singapore
[B9]	interviews	PC, mobile	multiplayer 1st person shooter, location-based	20, 200	Singapore, United Kingdom
[B12], [B11], [B10]	observation, interviews	PC	real-time strategy	50	Sweden
[B6]	interviews	online	poker	? (22 interviewed)	Sweden
[B3]	experience report	online	poker	? (42 developers and testers)	Sweden
[B5]	experience report	casino slot machines		? (18 in team)	United States

of software engineering but there were some case studies from a business and social sciences point of view. These were leaning more towards game development as a creative industry. Not all studies recognised the creative side of game development at all. This was more common in case studies done from a software engineering perspective, while survey studies more often mentioned the creative aspects of game development even from a software

engineering perspective.

Four of the published peer reviewed papers were not found through the snowballing but formed single clusters by themselves. The others formed two clusters, one main cluster where most of studies were linked to each other, and a smaller cluster with only two separate studies and four articles in total. The references of the articles can be seen in Figure 2.

A full bibliography listing of the surveys can be found in Appendix A and of the case studies in Appendix B.

5.1 Surveys

An online survey done within the Austrian game industry. The companies were asked about which process methodologies and tools they were using. An improved flexible process based on Scrum is also suggested based on the findings. The articles offer some statistics of the usage of different methods but otherwise did not contribute much to the analysis.

[A7] Juergen Musil et al. *A Survey on a State of the Practice in Video Game Development*. Tech. rep. Vienna University of Technology, 2010

[A8] Juergen Musil et al. "Improving Video Game Development: Facilitating Heterogeneous Team Collaboration through Flexible Software Processes". In: *Systems, Software and Services Process Improvement: 17th European Conference, EuroSPI 2010, Grenoble, France, September 1-3, 2010. Proceedings*. 2010, pp. 83–94

Another online survey conducted in Finland. This one focuses specifically on agile methods and practices and their impacts on the game development process. Different practices were tracked for different stages of development, so this study offers some sense of the degree of agility in game studios as well.

[A5] Jussi Koutonen and Mauri Leppänen. "How Are Agile Methods and Practices Deployed in Video Game Development? A Survey into Finnish Game Studios". In: *Agile Processes in Software Engineering and Extreme Programming: 14th International Conference, XP 2013, Vienna, Austria, June 3-7, 2013. Proceedings*. 2013, pp. 135–149

Finnish studios have been examined in an interview study as well. Several articles have been published based on these interviews, presenting different viewpoints of the same data. An additional survey for game studios around the world was also conducted for one of the articles to back up the interviews [A1].

[A1] Jussi Kasurinen. "Games as Software - Similarities and Differences between the Implementation Projects". In: *CompSysTech '15: Proceedings of the 16th International Conference on Computer Systems and Technologies*. 2016

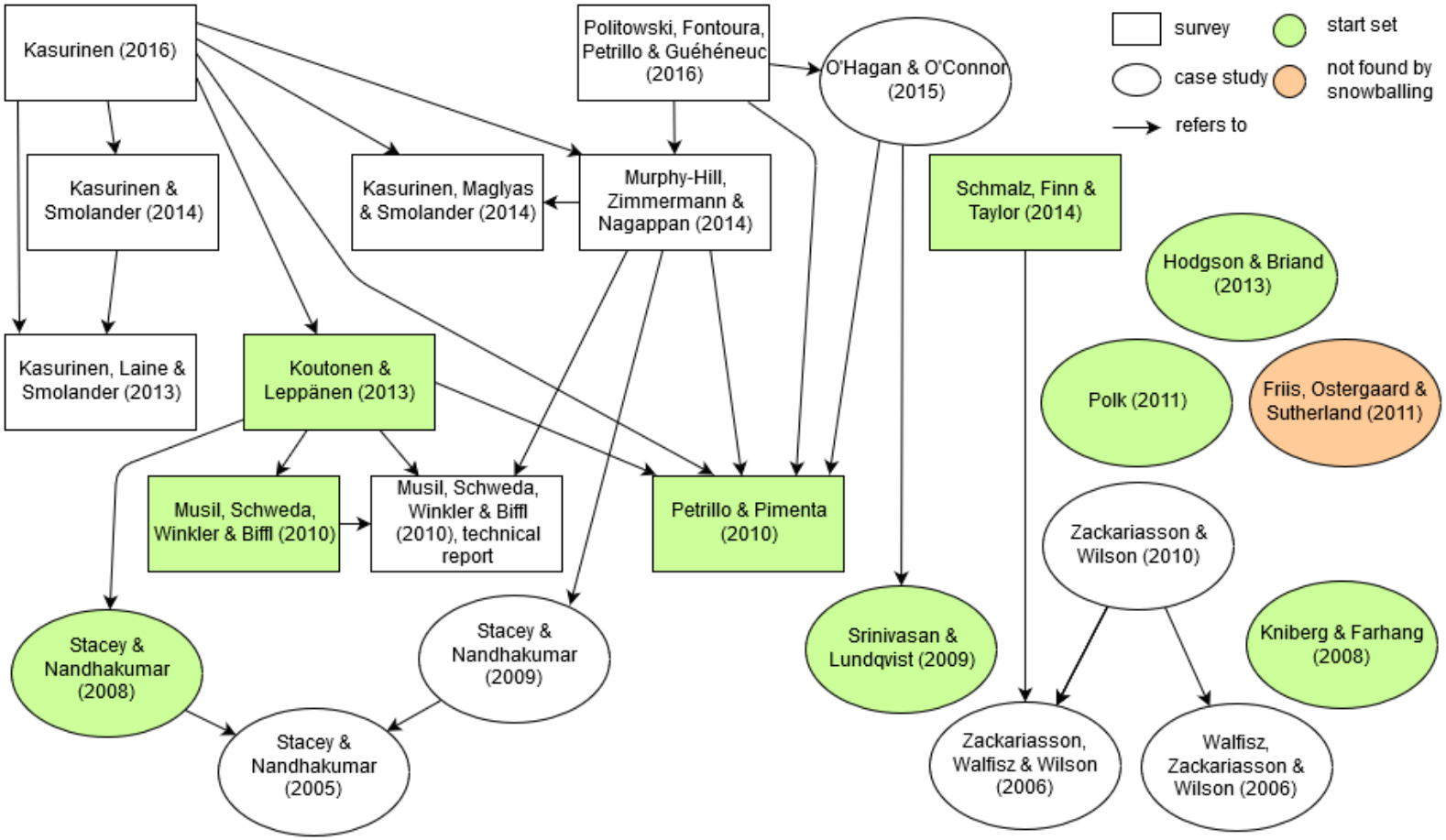

Figure 2: The analysis articles and their references to each other.

[**A2**] Jussi Kasurinen, Risto Laine, and Kari Smolander. "How Applicable Is ISO/IEC 29110 in Game Software Development?" In: *Product-Focused Software Process Improvement: 14th International Conference, PROFES 2013, Paphos, Cyprus, June 12-14, 2013. Proceedings*. 2013, pp. 5–19

[**A3**] Jussi Kasurinen, Andrey Maglyas, and Kari Smolander. "Is Requirements Engineering Useless in Game Development?" In: *Requirements Engineering: Foundation for Software Quality: 20th International Working Conference, REFSQ 2014, Essen, Germany, April 7-10, 2014. Proceedings*. 2014, pp. 1–16

[**A4**] Jussi Kasurinen and Kari Smolander. "What Do Game Developers Test in Their Products?" In: *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. ESEM '14. Torino, Italy, 2014, 1:1–1:10

Interviews were conducted among game development professionals, and a survey was done among Microsoft employees, comparing game development to other software engineering. The answers of Microsoft employees developing games, Office, and other products are compared to detect differences and to give validity to the interviews.

[**A6**] Emerson Murphy-Hill, Thomas Zimmermann, and Nachiappan Nagappan. "Cowboys, Ankle Sprains, and Keepers of Quality: How is Video Game Development Different from Software Development?" In: *Proceedings of the 36th International Conference on Software Engineering*. ICSE 2014. Hyderabad, India, 2014, pp. 1–11

A postmortem analysis of 20 game postmortems on Gamasutra[7]. The postmortems covered were mainly published in 2010-2013. The processes detailed in the postmortems were modeled and categorised as waterfall, iterative, hybrid, or ad-hoc, and whether or not they were agile.

[**A10**] Cristiano Politowski et al. "Are the Old Days Gone?: A Survey on Actual Software Engineering Processes in Video Game Industry". In: *Proceedings of the 5th International Workshop on Games and Software Engineering*. GAS '16. Austin, Texas, 2016, pp. 22–28

An earlier postmortem analysis covering 20 postmortems published from 1999 to 2002 and examining agile and good practices appearing in the postmortems. Other than percentages on the practices, this article did not include very much information or other statistics of agile game development, which could be seen from the concept matrix as well. This article had very few concepts besides the practices.

[7] `http://www.gamasutra.com/`

[**A9**] Fabio Petrillo and Marcelo Pimenta. "Is Agility out There?: Agile Practices in Game Development". In: *Proceedings of the 28th ACM International Conference on Design of Communication*. SIGDOC '10. São Carlos, São Paulo, Brazil, 2010, pp. 9–15

An interview study with eight producers from different types of game companies examining risk management practices. Game companies do not seem to have formal risk management but e.g. agile practices are used to mitigate risks.

[**A11**] M. Schmalz, A. Finn, and H. Taylor. "Risk Management in Video Game Development Projects". In: *2014 47th Hawaii International Conference on System Sciences*. 2014, pp. 4325–4334

5.2 Case studies

A social sciences study on a team in a medium-sized Canadian subsidiary of a publisher and game development company. The study focuses on modes of control and autonomy of the game team members. The team was using Scrum but the rest of the organisation was not.

[**B2**] Damian Hodgson and Louise Briand. "Controlling the uncontrollable: 'Agile' teams and illusions of autonomy in creative work". In: *Work, Employment & Society* 27.2 (2013), pp. 308–325

A questionnaire study among Sulake employees, both managers and non-managers. The study examines how adopting Scrum as the development methodology has impacted the organisation. Sulake upkeeps a virtual world, Habbo Hotel[8].

[**B1**] Dina Friis, Jens Ostergaard, and Jeff Sutherland. "Virtual Reality Meets Scrum: How a Senior Team Moved from Management to Leadership". In: *Proceedings of the 2011 44th Hawaii International Conference on System Sciences*. HICSS '11. 2011, pp. 1–7

Interviews done in a small Irish startup are analysed with grounded theory coding to examine the game development process and best practices. The study identifies Scrum as the methodology used but does not go into much detail in how the company applies it. The concepts recorded reflected this as this article had the lowest count of concepts of the case studies.

[**B4**] Ann Osborne O'Hagan and Rory V. O'Connor. "Towards an Understanding of Game Software Development Processes: A Case Study". In: *Systems, Software and Services Process Improvement: 22nd European Conference, EuroSPI 2015, Ankara, Turkey, September 30 – October 2, 2015. Proceedings*. 2015, pp. 3–16

[8] `https://www.habbo.com/`

A temporal analysis of a Singapore game studio's development process. Employees were interviewed and they provided workflow diagrams to explain the process. Another article with two other studios points out some aspects of the process are similar to agile practices and can thus be seen as "triggers" for agile game development. The Singapore studio had not formally adopted any agile method. Nevertheless a fairly detailed description of the process is given, and it is iterative and uses a playtest-feedback loop. The length of the cycles varied according to circumstance.

[B7] P. Stacey and J. Nandhakumar. "Managing Projects in a Games Factory: Temporality and Practices". In: *Proceedings of the 38th Annual Hawaii International Conference on System Sciences.* 2005, 234a–234a

[B9] Patrick Stacey and Joe Nandhakumar. "Opening Up to Agile Games Development". In: *Commun. ACM* 51.12 (Dec. 2008), pp. 143–146

[B8] Patrick Stacey and Joe Nandhakumar. "A temporal perspective of the computer game development process". In: *Information Systems Journal* 19.5 (2009), pp. 479–497

A study on agile adoption and stakeholder alignment based on interviews at a Swedish game studio under the pseudonym of GameDevCo. Without the whole organisation on board, true agility is difficult to achieve.

[B6] Jayakanth Srinivasan and Kristina Lundqvist. "Organizational Enablers for Agile Adoption: Learning from GameDevCo". In: *Agile Processes in Software Engineering and Extreme Programming: 10th International Conference, XP 2009, Pula, Sardinia, Italy, May 25-29, 2009. Proceedings.* 2009, pp. 63–72

An experience report of a Swedish online poker company, Tain, that was in charge of hosting an online poker championship. They successfully adopted Scrum and XP to solve the problems in the development process and the demands on the system caused by the championship.

[B3] H. Kniberg and R. Farhang. "Bootstrapping Scrum and XP under Crisis A Story from the Trenches". In: *Agile, 2008. AGILE '08. Conference.* 2008, pp. 436–444

A study using observation and interviews in a Swedish game studio, Massive Entertainment, which developed its own agile and evolutionary game development method. The articles focus on the creative process, leadership, and decision making.

[B12] Peter Zackariasson and Timothy L. Wilson. "Creativity in the Video Game Industry". In: *Creativity: Fostering, Measuring and Contexts.* Ed. by Allessandra M. Corrigan. Nova Science Publishers, 2010. Chap. 6, pp. 109–120

[B11] Peter Zackariasson, Martin Walfisz, and Timothy L. Wilson. "Management of Creativity in Video Game Development". In: *Services Marketing Quarterly* 27.4 (2006), pp. 73–97

[B10] Martin Walfisz, Peter Zackariasson, and Timothy L. Wilson. "Real-time strategy: Evolutionary game development". In: *Business Horizons* 49.6 (2006), pp. 487 –498

Another experience report on a team working on a remote delivery system for casino slot machines. As the team is working in a supporting role regarding the actual game development effort, this paper will not be considered as strongly in the analysis.

[B5] R. Polk. "Agile and Kanban in Coordination". In: *Agile Conference (AGILE), 2011.* 2011, pp. 263–268

5.3 Use of agile methods

This chapter seeks answers to the first research question of *How widespread is use of agile methods in game development?* The quantitative survey studies offer some indication of this. Agile methods do seem to be used quite often, with Scrum being the most popular method mentioned in the studies.

In Austria, in a survey with 13 respondents, agile or lean methods were used in 77% of the companies [A8], and Scrum in 61.5% [A7]. The survey was conducted in 2009. In the Finnish survey done in 2011, almost all respondents used agile methods at least during some parts of the development process [A5]. Scrum was popular in preproduction, production, and postproduction, with more than 50% of studios using it in those phases. Thus the Austrian and Finnish surveys give fairly similar results.

The Finnish interview study conducted in 2012 was complemented in 2016 with a survey including game developers from around the world [A1]. This survey had somewhat different results from the survey with only Finnish studios [A5], with only 26% using Scrum as the development method, and 13% some other agile method. However nearly all companies used iterative approaches. Quite a high percentage, 61%, felt they did not have a systematic development method. This is in contrast with the Austrian survey, where only 23% did not use any specific process [A7]. The difference may be explained by the differences in the survey population: the number of respondents in the survey done by Kasurinen was higher, and not focused only on one country [A1].

In another article most of the Finnish companies interviewed reported using agile methods, only one self-identified as plan-driven [A4]. Three out of the seven companies used Scrum, one some other defined agile process, and two were using what was called an undefined agile development approach. It was nevertheless found that even the ones that reported using Scrum or other

agile approaches, were working in what the researchers called a "pipeline" process [A3]. In the part of the study on whether the games industry could make use of systems and software engineering standard ISO/IEC 29110 for very small entities, the interviewed companies felt that the standard was not iterative enough [A2]. Game development seems to often be at least a very iterative process, and often an agile one on top of that.

There were some differences perceived in the rate of adoption of agile methods in the Finnish survey [A5] and the Microsoft survey [A6]. The Finnish survey found that in comparison to Finnish software industry, agile methods were not as popular within the games industry [A5]. Yet, most companies did use agile practices. The interviews and survey done by Murphy-Hill et al. however indicated that agile was used more in game development than in other software development, but their interviews hinted that "agile" may sometimes be used as an euphemism for lack of process [A6]. On the other hand Kasurinen et al. point out that companies may report using an agile method such as Scrum but have a rather plan-driven pipeline for game development in the end [A3]. Thus it is uncertain whether self-reporting on the development process is actually reliable. It is nevertheless clear that many companies have at least an interest in agile development.

In an analysis of postmortems, 11 out of 20 projects were iterative, 6 were waterfall, and 2 hybrid [A10]. Both hybrid processes were also agile, as well as 7 of the iterative ones, whereas 4 iterative processes were not identified as agile. Thus 9 out of 20 processes were thought to be agile. There may of course be some problems with interpreting the process from a postmortem, as not all postmortems contain similar amounts of data. An interesting point about the hybrid process model was that early production was done in an agile manner, but then later production changed into a waterfall process as the product neared completion. An earlier analysis of 20 postmortems did not look at agile methods per se, but tried to find evidence of agile practices [A9]. They concluded that game developers did use some agile practices even if informally.

5.4 Degree of agility

The second research question of *To what degree are agile methods used within game studios?* explores how deeply agile game studios are.

The Finnish interview study found that most companies identified as agile and even formally used an agile method such as Scrum [A3]. This did not necessarily mean however that all companies were equally agile. They found that there were differences in the overall flow of the process, with some companies starting out with a more traditional plan-based pipeline approach, and others seeking out features and requirements through constant feedback and iteration. The differences can be seen in Figure 3.

The online survey for Finnish game studios found that agile methods are

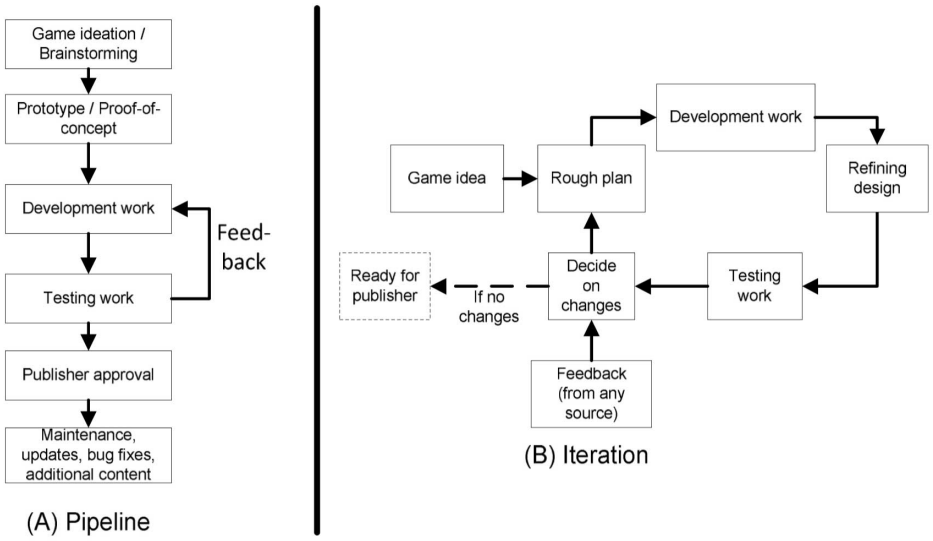

Figure 3: The differences between "pipeline" agile and continuously iterative agile processes, taken from Kasurinen et al. [A3].

used to a different degree depending on the phase of development [A5]. Scrum practices for example are mostly used in preproduction and production. Not all practices are used to the same degree, sprints and backlogs are popular in preproduction and production, and along with daily Scrum are used by companies in all phases of development as well, while burn down charts are used mainly in production if used at all. Cross-functional teams, informative work spaces and continuous integration are the most used of the included XP practices.

Something similar can be seen in some of the process models in the postmortem analyses of Politowski et al. [A10]. The hybrid process uses Scrum in the production phase but then at a certain point switches to a more waterfall process. The model can be seen in Figure 4.

An example of an agile, evolutionary process used at Massive Entertainment [B12] can be seen from Figure 5, where the process spirals towards completion, and there are points for reflection and potential changing of direction even within a cycle.

Thus the degree of agility seems to vary across companies and across the development process within a company. From the case studies it was also apparent that there were differences on the organisational level. Some studios have only one agile team while the rest of the large organisation is more plan-based [B2, B6]. In others, the whole organisation is successfully using agile methodology [B1, B10].

Information about specific agile practices in the case studies is fairly scarce. Sprints or cycles are used in all of them, as all of them are iterative.

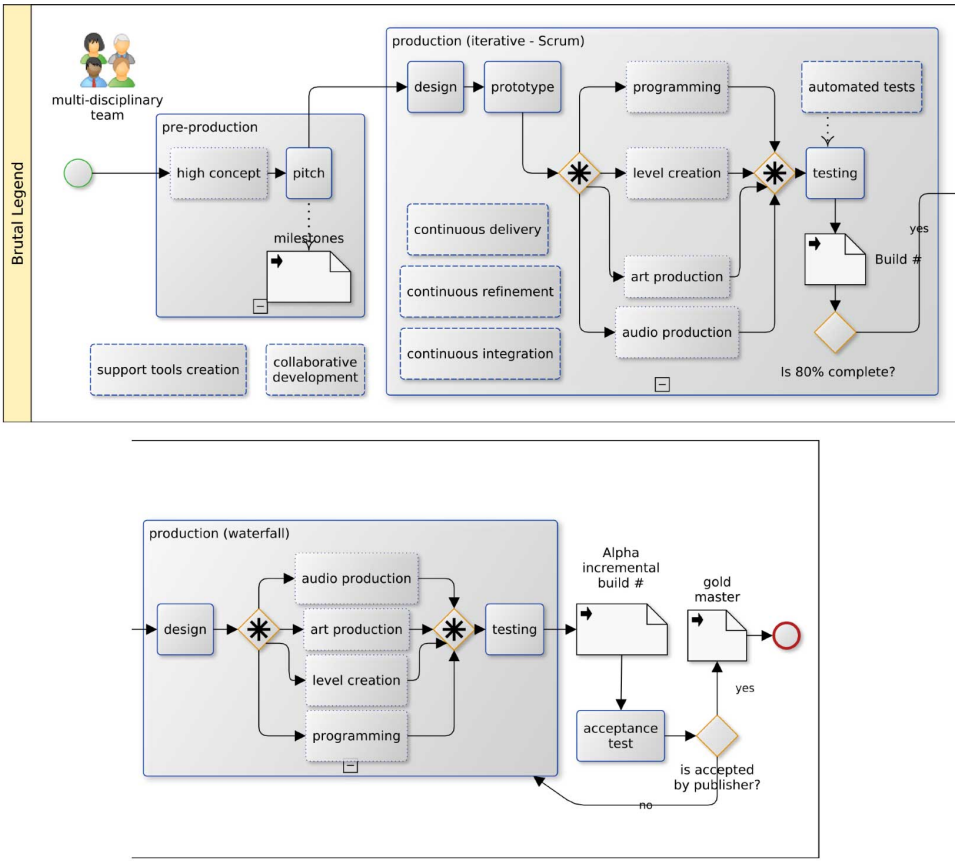

Figure 4: The hybrid process of making Brutal Legend, adapted from Politowski et al. [A10].

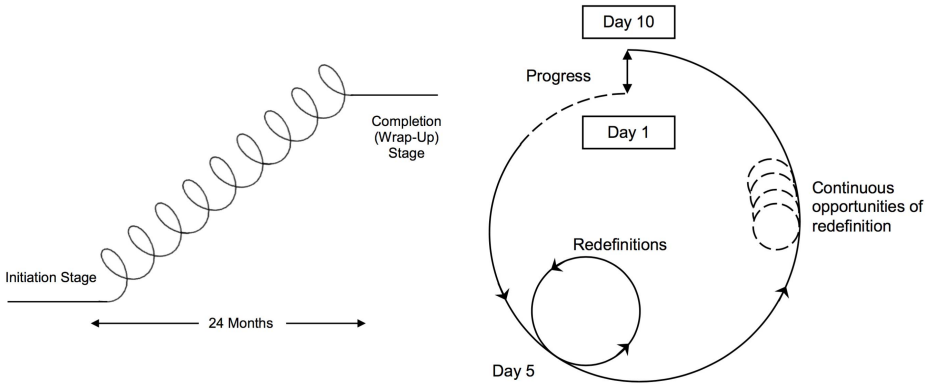

Figure 5: The agile, evolutionary process used at Massive Entertainment, with a spiral representing the overall process which is made up of cycles. Adapted from Zackariasson and Wilson [B12].

Daily Scrums or stand-up meetings occur in at least five of the cases [B5, B3, B2, B6, B1], though one of them is of a team not directly working on the game [B5]. Other meetings for planning, reviewing, and retrospection are also held in the case study companies, but they may be called by different names and be structured somewhat differently.

Using a backlog is mentioned in four of the case studies [B5, B3, B2, B1]. Something similar under the name of "feature plan" is used in one additional study [B10]. Burn-down charts are similarly used in four of the studied companies [B5, B3, B2, B1].

Continuous integration is mentioned in two case studies [B3, B5]. The two quantitative studies that measured continuous integration had conflicting results. The postmortem analysis of Petrillo and Pimenta found continuous integration mentioned in only two postmortems out of 20 [A9], but in the Finnish survey continuous integration was one of the most used XP practices [A5].

5.5 Problems solved by agile methods

The answers to the third research question of *What problems do agile methods solve in game development?* are based on the concept analysis and grouped according to themes.

Benefits of agile game development included more emphasis on testing ideas early with prototypes and a playtest-feedback loop, which enhanced the quality of the games produced.

5.5.1 Communication

Communication is an important part of agile development. Communication (especially with non-engineers) and conflict resolution skills were found to be especially important for game developers in both the interviews and the survey of Microsoft employees [A6].

In the Finnish survey, communication between professionals was seen as improved by adopting agile methods [A5]. Communication between stakeholders was not improved to the same extent however. Nevertheless some managers reported communication between departments within Sulake as having improved after adopting Scrum [B1]. Transparency was better as well. The managers also kept themselves up-to-date about progress by attending daily Scrum meetings. At Tain, communication problems were solved when adopting Scrum by cross-functional teams and seating arrangements in an open office [B3]. At the Irish start-up, the small team size enabled co-location and thus a "fluid process" [B4].

At Massive Entertainment, constant communication was seen as "the foundation for producing fun games" [B11]. The company CEO was instrumental in this, as he participated in briefings and meetings with the team leaders,

and had a habit of walking around the office to keep up-to-date with the production efforts. Something similar was hinted at happening also at Goo in London where the executive producer commented that fluid communication on the production floor lead to the best quality games [B9]. The project manager at Massive Entertainment also made sure that miscommunications were handled promptly during the cycles [B10].

5.5.2 Responsibility and autonomy

In the best cases, team responsibility and autonomy increases, but hiring the right kind of team is important.

At Sulake, one of the most reported changes after adopting Scrum was that the team took on previous responsibilities of the project manager [B1]. This had affected the recruitment processes as well. Managers were looking for people that were "flexible, with an open mind and good social skills". At Massive Entertainment, it was seen as important that the developers also be gamers, as that enables them to make more informed choices as to game design, and thus to be ready to take responsibility for decisions [B10].

Some concerns over the confidence of the development team were nevertheless expressed by the CEO at Massive Entertainment [B11]. Even though the fact that the team were also gamers contributed to their ability to make game design suggestions and decisions, it was still felt that the team could have had more courage to make decisions and take responsibility.

One of the respondents' comments in the Finnish survey mentions the importance of finding the right team, as with the wrong kind of team, the shifting of responsibility to the team may result in negative impacts of agility: "Agile methods transfer responsibility to teams and rules of working get an essential role. This does not necessarily suit all the teams." [A5]

Sometimes the shift in responsibility happens by circumstance, as in the Singapore game studio, where the same person working as both managing director and game designer was unavailable at the studio, and the team had to start taking responsibility for some game design choices [B9]. This was seen as similar to the courage valued in agile methods.

The other side of the team taking on responsibility is that managers have to let go of it, and to change their interaction with the team. As the managers in Sulake were well trained in Scrum, this mostly worked quite well for them [B1]. There was however a change from handing out orders to motivating the team to take responsibility. This was also seen as a good thing, as it freed up the managers' time for other things.

5.5.3 Prototypes, playtesting and feedback

Although prototyping is not necessarily a core agile practice per se, prototyping and early demos are recognised as an important part of the game

development process especially in early phases. Prototyping is used to test out game ideas in preproduction before committing to them in the production phase. Prototypes can also be marketable demo versions of the product for possible publishers or financial sources.

In the Finnish interview study, a developer is quoted as saying "Why should we bother with strict decisions when we can make a prototype in two hours to test things out?" [A2]. The authors point out that usability, playability and 'fun factor' are driving the decisions made, and these are found by testing with prototypes. Prototypes are an important part in ensuring these more abstract non-functional requirements.

Prototypes, sufficient preproduction for different parts of the project, and agile practices (even if not followed strictly) were found to be risk mitigating factors in a study by Schmalz et al. [A11].

A playtest feedback loop is also a core feature of the workflow in the Singapore studio [B8]. The feedback from playtesting has the potential to influence large changes in the product, even reverting the development back to the concept phase. These kind of changes do not necessarily mean however that the whole product reverts back to the beginning. The interview study on risk management had one interviewee commenting that level design for example may already be in production, while some gameplay mechanics may still be in preproduction to make them more fun [A11].

Demos at the end of a sprint are also used to keep the organisation and stakeholders up-to-date on the progress of the project [B1]. Playable versions of the game were also produced at the end of each cycle at Massive Entertainment [B10]. The feedback gained from within the company on these builds from the very beginning of the development process helped to make the quality of the game better. Because of the importance of this feedback, Massive Entertainment made sure to employ gamers, so that the developers had the viewpoint of the customer as well. In a somewhat similar manner, at each milestone, playable versions of the game were distributed among all staff at a studio in London [B9]. In a company with 200 employees, this generated a large amount of feedback. In this case the feedback came from all staff, not all of whom were necessarily gamers. The company however was producing mobile games, which are often more casual games appealing to a larger audience, compared to Massive Entertainment's real-time strategy titles.

Sometimes playtesting or user testing inspires new features, as is described in the case study of the Singapore studio where a bug turned into an interesting feature [B8]. This was also found in the Finnish interview study, where one company mentioned changing the product after having accidentally stumbled upon good new features through user testing [A3]. The feedback gathered from testing was often directly implemented into a new version of the game and not documented systematically. This type of feedback loop was not used by all the companies that claimed to be agile.

This loop is seen in some studies as preventing feature creep and schedule overruns at the end of the project [B10], but on the other hand the interviews among Finnish studios revealed that even large changes were acceptable even at the end of production [A3, A4]. This was apparently not seen as problematic by the studios, but the articles did not mention if these late changes lead to crunch time or extension of the schedule.

5.5.4 Quality of games

The quality of games was mostly seen to improve with agile methods in most of the studies that mentioned quality aspects. Quality can be seen as a result of the playtesting feedback loop.

The respondents of the Finnish survey indicate that agile methods improve the quality of the games produced, even though the effect on the quality of the code is less noticeable [A5].

When the Swedish company Tain adopted Scrum and XP, they started doing test-driven development, and devoted 20% of time to refactoring old code [B3]. This resulted in a clear improvement in the quality of their code.

At the Finnish company Sulake, where Scrum had been adopted across the organisation, managers thought that Scrum had both improved productivity and the quality of the product [B1].

Game quality was closely associated with playable versions of the game at Massive Entertainment in Sweden [B12]. These builds were produced after each cycle and feedback was gathered within the company. Quality arose from this feedback loop, focusing the efforts of the team towards quality instead of just meeting milestones [B10]. This user testing was started from the very beginning, building up the quality and user experience as the game grew. In postproduction the quality focus changed to fixing bugs and optimising code, and good quality user experience was expected to have been reached already. Ensuring gameplay quality early avoided feature creep in post-production.

5.6 Challenges in using agile methods

The fourth research question was *What challenges are there in using agile methods in game development?* Based on the concept analysis, the challenges encountered in the studies were related to the degree of agility in the whole organisation, and the creative nature of game development.

5.6.1 Artists vs. engineers

Half of the case studies mention creativity [B4, B2, B10, B8] (as can be seen from the concept matrix 1) and the other half consider games to be a software engineering effort [B5, B3, B6, B1]. In some of the ones that consider the creative aspects of game development and also one survey, it is apparent that artists seem to be more resistant to adopting agile methods or

enforcing process control in general than more technically oriented employees. This can be a challenge for adopting and using agile methods in the games industry. The flow of work for artists and for programmers can be different, and requires more careful synchronisation.

Even though Scrum meetings were held to enhance communication in the Canadian game studio, the artists effectively did not participate, as they remained silent during the meetings the researchers observed [B2]. This is in contrast to the improvement in communication seen in other studies [A5, B1], and may be reflective of the general atmosphere and problems of a single team adopting agile in a non-agile organisation. The artists also sometimes did not update the backlog, which further complicated communication, as meetings relied on the information in the backlog.

Although not explicitly or formally agile, the Singapore studio also had problems with artists, with artists often resigning, and over- or underproduction of graphics assets [B8]. The workflow problems arose partly because of poor comunication and non-permanent arts staff.

Due to having multiple disciplines working on a game, one interviewed game developer pointed out the importance of good communication skills [A6]. Good conflict resolution skills are also of importance. The multidisciplinary and creative nature of the team was also hypothesised to be one reason for a perceived lack of process in game development. One interviewee was quoted as saying:

> "We've got so many specialists on the team, so the kind of planning that you usually do in Agile doesn't work quite so well... You know [specialists] are more concerned about the creative process than an engineering process."

5.6.2 Management and other stakeholders

While in some studies management was up-to-date about agile methods and practices, in some studies it was clear that management was holding the agile team back. Adopting an agile method requires changes in management style as well [B1]. Further complicating the situation for many game development companies is the fact that many stakeholders are involved in publishing a game. The publisher and marketing department may have a set release schedule in mind, and if the publisher is not agile, it may complicate things for the agile team. In the Canadian studio, external stakeholders determined milestones and who was working on the team, with the team size varying across the development timeline [B2]. This reduced the ability of the team to properly take responsibility for decision-making.

While agile methods may increase the responsibility and autonomy of team members, this can be undermined by non-agile management and orders from higher up the organisation. In the Canadian studio, even though

the team leaders tried to emphasise the responsibility and autonomy of the team members, it was nevertheless the team leader whose viewpoint was the most respected [B2]. These internal power hierarchies can hamper communication as well, when junior team members yield to the views of more senior members. As a subsidiary of a larger company, the team at GameStudio was also interrupted in their work by the parent company's demands. Eventually even the team leaders felt the need to make the internal hierarchy clearer, and thus gave out conflicting messages about the autonomy of the team.

Despite using Scrum on the team level, the Canadian studio was also perhaps more traditional in their prototyping, producing a prototype at the end of the concept phase, and sequences of the game after preproduction [B2]. As the studio was a subsidiary of a larger publisher, progress had to be approved by the parent company that also set the targets to be met.

In the Singapore studio, the project manager had previously worked on developing banking systems in a rather waterfall-like and structured process [B8]. He tried to bring this style of working into the game studio as well, mainly by introducing more documentation. Nevertheless the process in the studio was very iterative and made use of a playtest-feedback loop. The authors saw a need to balance between routine and improvisation in game development due to the creative nature of the effort.

Stakeholder alignment, employee empowerment, group and organisational learning, and governance mechanisms were recognised as an important enablers of agile adoption in a case study of a Swedish online poker game company GameDevCo [B6]. Among the problems in stakeholder alignment at GameDevCo was that GameDevCo had been acquired by a much larger company, which did not have a good understanding of GameDevCo's product. This also created a more hierarchical organisation, which broke down the cross-disciplinary nature of their teams. The project management tool used was required by the corporation but was insufficient to the needs of rapid agile development.

5.6.3 Quality of code

The Finnish survey did not find the quality of code similarly improved as overall game quality after adoption of agile methods [A5]. Though no respondent fully disagreed with the statement that code quality had improved, there were actually quite a few who partly disagreed. The quality of the code was also deemed difficult to assess, with respondents answering "do not know". This was suspected by the authors to be due to improper implementation of continuous integration, lacking automated testing. Indeed the survey found that test-first programming was mainly used in the preproduction stage, and less in production. Overall test-first programming was not a very popular practice.

Another study found that automated testing was indeed seen as more difficult to implement in game development due to difficulties of separating the user interface from the gameplay mechanics, and also due to the large search space of possible game states [A6]. Automated testing was even seen as contrary to agility, preventing the fast pace of changes often needed in the game design process. One of the interviewees stated that "Games are tested, but at the game play level." Similar findings were apparent in the Finnish interview study where user testing was found to be the most popular type of testing done [A4].

One of the case studies, of the Swedish GameDevCo, found that adoption of Scrum lead to "poor fidelity artifacts", most likely meaning that code quality suffered from too tightly packed sprints [B6]. One of the problems in adopting agile practices in the company however was that not all staff was well trained in the Scrum process.

5.6.4 Training

Training, or lack there of can make it difficult to successfully adopt agile methods, as there can be a lack of consensus of what the agile method in use actually is, and how it is to be implemented. This can be seen in the case of GameDevCo [B6], where one interviewee even commented that "[w]e spend more time talking about what SCRUM is, and what it should be, rather than focusing on what it should be doing for us." Some employees had never received training in Scrum.

The opposite can be seen in the cases of Tain [B3] and Sulake [B1]. Tain brought in a Scrum consultant to help them transition to using the method effectively and also offered training for ScrumMasters in all teams. GameDevCo brought in a consultant as well, but in their case the process was interrupted and the consultant left before the knowledge was solidified in the studio [B6]. At Sulake, most of the managers were trained in Scrum, and this was seen to make a difference in how they were able to work with the teams [B1].

6 Discussion

First there will be some reflection on the thesis process; of the challenges encountered during the search for the articles and the concept analysis. This will be followed by discussion of the research questions and findings, comparing them to Keith's model of agile game development [6].

6.1 Reflection on the work done

During the search for papers, it was found that the search string could have included `management` or `manag*` in addition to `develop*`, `process*` and `method*`. This also became apparent during the analysis of the articles. The articles often presented a management viewpoint, as management has a role in establishing and enabling different process methodologies. Including management in the search string would have found the article [B1] discovered through the previous literature review on game development processes [11].

Another article found through snowballing did not explicitly state it was describing an agile process [B11]. This was only mentioned in a footnote in a related article [B12]. The development process was nevertheless well described and seemed rather agile, so the snowballing trail was followed after reading the article.

It is also apparent that some studies on game development do not make a difference between game development and other software engineering. This purely software engineering perspective can cause problems in identifying studies that are actually on the games industry. These may mention games only implicitly, for example one case study was done in a company with the pseudonym GameDevCo [B6].

While writing up the results of the concept analysis it was found that since the research questions had been divided into finding the benefits and challenges of agile game development, it would have been useful to have had further division of some of the concepts. Responsibility and autonomy of the teams for example was something that some companies struggled with, while some companies saw improvements in it after having adopted agile practices.

6.2 Use of agile methods

The quantitative survey studies offered some insight into the first research question of *How widespread is use of agile methods in game development?* There are however some difficulties in drawing conclusions of the usage of agile methods within the game industry. Some of the surveys had similar results [A7, A5] but others contradicted them [A1]. The self-reporting by companies can be problematic, as even if officially an agile process such as Scrum is adopted, the process may not really reflect it [A3, A6]. Some companies may rush to adopt agile methods as a "trendy" thing to do,

without proper training, and subsequently run into problems [7].

Nevertheless, game development is recognised as a very iterative process, and as knowledge of agile methods becomes more widespread in the games industry, more companies may opt to incorporate agile practices into their process, as anticipated by Stacey and Nandhakumar [B9].

6.3 Degree of agility

The second research question of *To what degree are agile methods used within game studios?* attempted to gather an understanding of how deeply agile the processes used by game studios are. Many of the survey studies point out that the agile process used by game studios is in many cases not purely agile. This may be due to lack of training [B6], or consciously adopting and improving a custom process [B12].

It is also worth noting that the perceptions of those involved in the process itself may not be objective. Hodgson and Briand observed that the development team actually had less autonomy and decision power than they thought they did [B2]. Similarly Kasurinen et al. observed that the process model in use by companies calling themselves agile was not really agile in a grander scale [A3]. It seems that there may be different levels on which the game development process can be agile. For some projects the whole process is equally agile but for others agility is present within the phases of development in varying degrees.

The Finnish survey [A5] seems to confirm the suggestion by Keith [6] that Scrum be used in preproduction especially. Keith then goes on to suggest adopting lean concepts and kanban for the production phase. A kanban board makes it easier to visualise the different workflows, which helps to keep the production pipeline moving with all specialists having a consistent amount of work to perform. The estimations for the time taken for e.g. a level's audio design are made based on the experience of preproduction. Kanban is clearly less well known within the game development community, as only one company of the respondents of the Finnish survey of agile methods used work visualisation in all phases [A5]. Other kanban practices are even more uncommon.

While the Singapore studio [B8] had not formally adopted agile methods, it seems that they could benefit from consciously deciding to adopt agile practices. The authors do suggest that some agile "triggers" are present at the company [B9], and they have a very iterative process with a playtesting-feedback loop.

6.4 Problems solved by agile methods

The third research question of *What problems do agile methods solve in game development?* explored the benefits of using agile methods and practices.

Most benefits were found in communication, team responsibility, and game quality through a playtesting-feedback loop using prototypes.

Communication problems, especially between artists and more technically oriented members of the team are sometimes a problem in game development [13]. It was also a theme that clearly showed up in the concept analysis. Even though some companies still struggled with communication issues [B2], there were others that had seen improvements in communication after the adoption of agile methods [B1].

Prototypes are equally emphasised in both the surveys and case studies. The concept analysis seems to confirm the findings of Winget and Sampson that playable prototypes are considered essential in current game development [19]. There is also a clear link between prototypes (and other playable versions of the game), playtesting, feedback, and game quality, creating a loop throughout the development process. This agile loop thus helps to counter development process problems [17], as well as design problems [13].

The prototyping at the beginning of a project can create a lot of waste, with a sizable portion of ideas rejected in early stages [A11]. This prototyping is nevertheless part of an important feedback loop for the process, where design ideas are built into prototypes, tested by users for usability and fun (users may be the developers themselves, other company employees, professional testers, and in later stages actual players as well), and feedback from the testing is used to create another prototype or version of the game.

The type of prototyping in game development is more incremental in nature and thus the term 'prototype' may be somewhat misleading. While there are large amounts of waste at the very beginning of a project when early ideas are tried out, often the later prototypes are actually playable builds of the game and work towards enhancing the game experience incrementally. At this stage, the "prototyping" resembles agile practices more closely.

The benefits of agile usage were most clearly seen in project management, programming practices, and game quality in an informal online survey for game developers conducted by Keith in 2010 [7]. Game quality was clearly seen to improve in many of the articles studied for this literature review as well, but there were contradictory results for project management and programming practices. In the companies where there was sufficient knowledge of agile methods, management was able to change their ways of working, shifting responsibility to the team and freeing up time for inter-departmental cooperation, as happened at Sulake for example [B1]. Programming practices were consciously improved at Tain, resulting also in better quality of code [B3]. Nevertheless, this was not a simple result of adopting agile methods, but a conscious choice and need to improve a tangled codebase to be able to work more effectively.

The shifting of responsibility to the team is not easy. To address this, Keith emphasises the need to mentor the team so that they can grow to self-manage and self-organise [6]. An issue apparent in one of the case studies

was that managers can perhaps give out contradictory messages regarding the autonomy of the team. On the one hand, they seemed to wish for the team to take on more responsibility and be more self-managing, but on the other hand the power hierarchy is reinforced [B2]. Managers may need help in letting go of hierarchical power structures, to be able to reap the benefits of agile creative work.

Interestingly, even though crunch time is a widely occurring and recognised problem in game development [4], it was not discussed much in the case studies, except for one case at Tain, where working overtime was a problem and was subsequently specifically discouraged as they adopted Scrum as their methodology [B3]. Crunch time is perhaps more of a managerial choice in a competitive project-based industry [12], than a problem arising only from the process used. Nevertheless, it is possible that the shifting of games projects towards games as services and agile methods that encourage a healthy pace of working may eventually contribute to changing this trend.

In addition to crunch time, the studies analysed give no specific evidence of agile methods preventing feature creep either, but at Massive Entertainment it was consciously avoided at the end of a project [B10]. It could be deduced that the iterative process which helps to find fun and implementable ideas sooner [A5], also prevents feature creep later on, as features have been implemented already and proven to be good by prototyping, playtesting, and feedback.

6.5 Challenges in using agile methods

The fourth research question of *What challenges are there in using agile methods in game development?* found challenges in the degree of agile adoption and commitment across the organisation, and managing the creative aspects of game development.

Challenges mentioned in the comments gathered from game developers in the informal survey by Keith [7] are rather similar to the findings in this thesis. Management needs to change their attitude towards managing the team, but in some cases adopting agile methods caused resistance to change, despite the fact that benefits were also seen. External stakeholders are also seen in the comments as potential sources of problems with demands that could cause feature creep. Other possible challenges are communication with other non-agile teams, as well as lack of knowledge and training in agile methods. The successful companies had adopted agile across the company, and employees had received training in the method adopted. There was a division in the case studies into success stories and to those that focused more on the problems of adopting agile methods. Three studios were described as successful [B3, B1, B10], and two studios were struggling somewhat [B2, B6].

One clear difference emerged between the successful companies and the struggling ones. When the whole organisation is committed to agility, it is

also easier for the teams to work in an agile manner. When the rest of the organisation is hierarchical and not supportive of agility, it creates problems for the agile team, as there is poor stakeholder alignment and the team cannot take full responsibility for their decisions.

Another complication is the difference in the working styles between artists and engineers. Artists can be more resistant to attempts to control the process, preferring to have artistic freedom and more creative control. Communication skills thus emerge as an important asset in game development, as especially managers need to be able to broker between artists and engineers. The different, more enabling than controlling, approach to management in a successfully agile organisation enables better communication as well.

Keith also mentions artists being more sceptical of Scrum and agile at first [6]. Cross-functional teams are important in ensuring the programming and arts stay on the same track and problems are solved quickly. Cross-functional teams can be a concern at first though, as traditionally game development is done in separate arts teams and programming teams. Even though cross-functional teams was found to be one of the most used XP practices in the Finnish survey [A5], there was little evidence of the practice in the analysed articles.

The teams in the larger studios described in the case studies were divided by disciplines, with programmers in one team and artists in another. This is in contrast to what is described by Keith [6]. He suggests that informal "guilds" for the different disciplines can be used to enhance communication within a particular discipline.

Keith lists the following problems with publishers not well acquainted with agile methods [6]:

- Not playing sprint builds
- Not attending reviews or planning sessions
- Ignoring the product backlog
- Demanding detailed schedules and documents up front and ignoring the need to revisit them based on actual progress
- Making urgent requests in the middle of a sprint

Especially the last point was seen in the case studies of the current analysis. Keith comments that publishers are more willing to accept agile practices in preproduction than in later phases of development. Using a stage-gate model is also mentioned as a compromise for working with a publisher concerned about the feasibility of a new game idea. This involves setting specific decision points along the way, with the project receiving funding and moving onwards to the next stage of development only if certain conditions are met. This model was used in a couple of the case studies as

well, at GameDevCo in Sweden and at GameStudio in Canada [B6, B2]. In contrast, it was pointed out that at Massive Entertainment the stage-gate conditions were impossible to determine due to the creative and uncertain nature of the process, so a stage-gate model was seen as impractical and thus was not in use [B11]. The two cases where the model was used listed more problems regarding the process and its management than the ones where it was not in use. It may be that the stage-gate model is reflective of the trust between the studio and the publisher.

Quality of code was mostly seen as decreasing after the adoption of agile methods, for example at GameDevCo where it was felt that the speed of development during the sprints was too fast to write good quality code [B6]. In contrast at Tain the quality of code was seen to improve along with velocity [B3]. The difference at Tain was that they consciously took time to refactor and improve the quality, and also started using test-driven development to improve the quality of new code [B3].

Both GameDevCo and Tain were making an online poker game system, so the two cases are rather comparable and the genre does not explain the difference between the experiences. It may in fact be easier for a game like poker, even though test-driven development and automated testing have been found difficult to implement for games in general [A6]. Test-driven development is also recommended by Keith in his book on agile game development [6]. The fact that it is found difficult for games in general, may indicate that game developers are simply not used to test-driven development and need to find ways to implement it according to the requirements of each game.

Game teams often work as part of a larger organisation or are dependent on outside sources for finances and publishing, and these external stakeholders may not be aware of agile processes. The question is, how to best make agile methodology work on the team level despite outside pressures? Perhaps with time the game industry as a whole will become more knowledgeable of agile methods, but further research on these issues may be useful as well.

6.6 Other remarks

The surveys focused more on requirements than the case studies, and often pointed out the difficulties of figuring out non-functional requirements like the 'fun factor'. The creative aspects of game development were also more likely to be recognised in the surveys, while half of the case studies failed to mention creativity at all, considering game development from a purely software engineering viewpoint. Most of the case studies that did mention creativity were not software engineering studies but from the field of social sciences. Thus there may be a certain bias in the case studies covering agile game development from a software engineering perspective. Some aspects of the process are not recognised, and thus game studios cannot fully benefit

from research and e.g. standardisation efforts, as was discovered in one of the surveys [A2].

The case studies, partly due to the social and system sciences viewpoints, focused more on the team and organisational dynamics. New case studies from a software engineering point of view, taking into account the creative multidisciplinary nature of game development, would shed more light on the details of the agile game development process.

Interestingly, while the surveys often found the 'fun factor' to be important for requirements and risk management, this only showed explicitly in one case study, at Massive Entertainment [B11, B12, B10]. There it was clearly the target and goal to strive for, and was mentioned in all of the related articles.

Answers to the statement "Impact of replacements in the personnel have diminished" in the survey of Finnish game studios were mostly "do not know" or "partly disagree" [A5], so even though according to Stacey and Nandhakumar the event of people leaving the company might trigger agile practices like job rotation [B9], it is not clear that it would get easier to handle with agile practices.

A point that did not appear that often in the analysed studies is that there is a growing trend towards games as services [15]. With digital distribution and modern game consoles being able to connect to the internet, updates can be offered regularly and automatically. Constantly updatable online or mobile games are more naturally amenable to an iterative agile process than traditional boxed copy PC or console games. One of the companies that had successfully adopted agile on the full organisational level, was Sulake which upkeeps a virtual world, Habbo Hotel[9]. On the other hand, more traditional titles can be produced in a successfully agile manner, as exemplified by Massive Entertainment [B10].

When considering the success of companies adopting agile practices, the lens through which the game studio is viewed may affect the interpretation. The Canadian studio [B2] and Massive Entertainment [B10, B12, B11] are both successful game studios, and there are some similarities reported in that the management feels the team should take on more responsibility. The Canadian study was focusing on control and autonomy, whereas the Swedish study was focusing on leadership and management of creativity. The picture of the Canadian studio emerges more as a struggling one, but it may partly be because of the specific focus of the study.

As well as studies on how games are actually developed, there have been studies attempting to find an improved process model for game development. Musil et al. [A8] base their improvement suggestion on Scrum as it was found to be the most popular method in their survey. The process emphasises the ability to go back from testing and even operation to earlier stages including

[9] `https://www.habbo.com/`

specification, design, implementation and content creation. Comments from Austrian game developers indicated potential for the process to reduce crunch time, focus on preproduction and prototypes, and view game design as a continuous process. With the operation phase taken into account, the model would seem to take into account games as services as well. It also resembles the feedback loop used at the Singapore studio [B8].

Team configuration and management, game design document management, game engine development, game test management, and programming practices were confirmed to be important factors for success of the game development process in a survey among game developers across the world [1]. The results of the analysis of this literature review would seem to indicate that agile methods and practices can positively influence at least some of these factors like team configuration and management, and game test management. Nevertheless, team management presents some challenges as well. Interestingly game engine architecture was not as important a success factor [1]. This could be due to usage of third-party game engines by game developers.

7 Conclusion

Game development is a varied field with studios of different sizes and affiliations, in which multidisciplinary teams make games of numerous genres. Even though software engineering makes up an important part of the technical side of game development, it is also very much a creative effort, with often elusive non-functional requirements like the 'fun factor'.

Games are produced in phases or stages, beginning with a concept phase where rough game ideas are elaborated. The exploration of ideas is continued in preproduction, testing out gameplay with prototypes to ensure and refine the 'fun factor' before production begins. Preproduction is often very iterative and a lot of agile practices are well suited to it. Production begins ideally with a solid base for a fun game. Content is added and gameplay expanded. Practices may change towards lean production, or even a more waterfall model towards the end of the phase. Testing is done continuously throughout the process in agile game development. The most common type of testing is gameplay or usability testing, to ensure non-functional requirements are met. Postproduction fixes final bugs and issues and may involve testing with the actual target group.

This literature review inspected the evidence of agile methods and practices in game development in both surveys and case studies. Agile practices are indeed applied by game studios, in varying degrees. It seems most companies adapt the methods to suit their needs and may not use purely agile methodologies.

It was found that agile practices help to mitigate risks, enhance team communication and responsibility, and improve game quality through a fast prototyping, playtesting, and feedback loop. They may not however be sufficient in overcoming all problems of game development. Crunch time is a prevalent issue among game companies, affected by unrealistic schedules and feature creep. Good preproduction and prototyping may nevertheless alleviate these problems. Crunch time may be an issue of work culture in a maturing and highly competitive field.

Despite the benefits, there are also challenges in adopting and using agile methods in game development. Multidisciplinary teams need good communication skills for the specialists to understand each other. One of the ways to achieve this is gathering co-located cross-functional teams. Artists may be more sceptical of agile methods at first as their workflow may differ from that of programmers. Management willing to relinquish control and let go of hierarchical power structures is needed to fully enable agility. Training for both management and the developers is essential in ensuring that agile methods can be a benefit instead of a burden. Quality of code can easily reduce with too fast a pace for sprints, but it can be improved by finding ways to adapt test-driven programming to game development and consciously taking time to refactor the codebase.

Based on the challenges found for adopting and using agile methods in game development, new directions for research could include finding ways to enable organisations to change power hierarchies standing in the way of agile creative work. In addition to internal power structures, game companies often have to interface with publishers which may be large hierarchical corporations. The game industry is constantly changing however, and digital distribution gives opportunities for even smaller studios to self-publish. Games as services can ensure more financial freedom with earlier releases and gradually building up content. Companies offering games as services can find adopting agile practices more natural.

Future research could find new insights regarding the applicability and need for adaptation of agile software engineering practices within the game industry, through greater focus on game development as a creative multidisciplinary field of software engineering.

Appendix A: Survey articles

[A1] Jussi Kasurinen. "Games as Software - Similarities and Differences between the Implementation Projects". In: *CompSysTech '15: Proceedings of the 16th International Conference on Computer Systems and Technologies*. 2016.

[A2] Jussi Kasurinen, Risto Laine, and Kari Smolander. "How Applicable Is ISO/IEC 29110 in Game Software Development?" In: *Product-Focused Software Process Improvement: 14th International Conference, PROFES 2013, Paphos, Cyprus, June 12-14, 2013. Proceedings*. 2013, pp. 5–19.

[A3] Jussi Kasurinen, Andrey Maglyas, and Kari Smolander. "Is Requirements Engineering Useless in Game Development?" In: *Requirements Engineering: Foundation for Software Quality: 20th International Working Conference, REFSQ 2014, Essen, Germany, April 7-10, 2014. Proceedings*. 2014, pp. 1–16.

[A4] Jussi Kasurinen and Kari Smolander. "What Do Game Developers Test in Their Products?" In: *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. ESEM '14. Torino, Italy, 2014, 1:1–1:10.

[A5] Jussi Koutonen and Mauri Leppänen. "How Are Agile Methods and Practices Deployed in Video Game Development? A Survey into Finnish Game Studios". In: *Agile Processes in Software Engineering and Extreme Programming: 14th International Conference, XP 2013, Vienna, Austria, June 3-7, 2013. Proceedings*. 2013, pp. 135–149.

[A6] Emerson Murphy-Hill, Thomas Zimmermann, and Nachiappan Nagappan. "Cowboys, Ankle Sprains, and Keepers of Quality: How is Video Game Development Different from Software Development?" In: *Proceedings of the 36th International Conference on Software Engineering*. ICSE 2014. Hyderabad, India, 2014, pp. 1–11.

[A7] Juergen Musil et al. *A Survey on a State of the Practice in Video Game Development*. Tech. rep. Vienna University of Technology, 2010.

[A8] Juergen Musil et al. "Improving Video Game Development: Facilitating Heterogeneous Team Collaboration through Flexible Software Processes". In: *Systems, Software and Services Process Improvement: 17th European Conference, EuroSPI 2010, Grenoble, France, September 1-3, 2010. Proceedings*. 2010, pp. 83–94.

[A9] Fabio Petrillo and Marcelo Pimenta. "Is Agility out There?: Agile Practices in Game Development". In: *Proceedings of the 28th ACM International Conference on Design of Communication*. SIGDOC '10. São Carlos, São Paulo, Brazil, 2010, pp. 9–15.

[A10] Cristiano Politowski et al. “Are the Old Days Gone?: A Survey on Actual Software Engineering Processes in Video Game Industry”. In: *Proceedings of the 5th International Workshop on Games and Software Engineering*. GAS ’16. Austin, Texas, 2016, pp. 22–28.

[A11] M. Schmalz, A. Finn, and H. Taylor. “Risk Management in Video Game Development Projects”. In: *2014 47th Hawaii International Conference on System Sciences*. 2014, pp. 4325–4334.

Appendix B: Case study articles

[B1] Dina Friis, Jens Ostergaard, and Jeff Sutherland. “Virtual Reality Meets Scrum: How a Senior Team Moved from Management to Leadership”. In: *Proceedings of the 2011 44th Hawaii International Conference on System Sciences*. HICSS ’11. 2011, pp. 1–7.

[B2] Damian Hodgson and Louise Briand. “Controlling the uncontrollable: ‘Agile’ teams and illusions of autonomy in creative work”. In: *Work, Employment & Society* 27.2 (2013), pp. 308–325.

[B3] H. Kniberg and R. Farhang. “Bootstrapping Scrum and XP under Crisis A Story from the Trenches”. In: *Agile, 2008. AGILE ’08. Conference*. 2008, pp. 436–444.

[B4] Ann Osborne O’Hagan and Rory V. O’Connor. “Towards an Understanding of Game Software Development Processes: A Case Study”. In: *Systems, Software and Services Process Improvement: 22nd European Conference, EuroSPI 2015, Ankara, Turkey, September 30 – October 2, 2015. Proceedings*. 2015, pp. 3–16.

[B5] R. Polk. “Agile and Kanban in Coordination”. In: *Agile Conference (AGILE), 2011*. 2011, pp. 263–268.

[B6] Jayakanth Srinivasan and Kristina Lundqvist. “Organizational Enablers for Agile Adoption: Learning from GameDevCo”. In: *Agile Processes in Software Engineering and Extreme Programming: 10th International Conference, XP 2009, Pula, Sardinia, Italy, May 25-29, 2009. Proceedings*. 2009, pp. 63–72.

[B7] P. Stacey and J. Nandhakumar. “Managing Projects in a Games Factory: Temporality and Practices”. In: *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*. 2005, 234a–234a.

[B8] Patrick Stacey and Joe Nandhakumar. “A temporal perspective of the computer game development process”. In: *Information Systems Journal* 19.5 (2009), pp. 479–497.

[B9] Patrick Stacey and Joe Nandhakumar. “Opening Up to Agile Games Development”. In: *Commun. ACM* 51.12 (Dec. 2008), pp. 143–146.

[B10] Martin Walfisz, Peter Zackariasson, and Timothy L. Wilson. “Real-time strategy: Evolutionary game development”. In: *Business Horizons* 49.6 (2006), pp. 487 –498.

[B11] Peter Zackariasson, Martin Walfisz, and Timothy L. Wilson. “Management of Creativity in Video Game Development”. In: *Services Marketing Quarterly* 27.4 (2006), pp. 73–97.

[B12] Peter Zackariasson and Timothy L. Wilson. “Creativity in the Video Game Industry”. In: *Creativity: Fostering, Measuring and Contexts*. Ed. by Allessandra M. Corrigan. Nova Science Publishers, 2010. Chap. 6, pp. 109–120.

Bibliography

[1] Saiqa Aleem, Luiz Fernando Capretz, and Faheem Ahmed. "Critical Success Factors to Improve the Game Development Process from a Developer's Perspective". In: *Journal of Computer Science and Technology* 31.5 (2016), pp. 925–950.

[2] Apostolos Ampatzoglou and Ioannis Stamelos. "Software Engineering Research for Computer Games: A Systematic Review". In: *Inf. Softw. Technol.* 52.9 (2010), pp. 888–901.

[3] D. Callele, E. Neufeld, and K. Schneider. "Requirements engineering and the creative process in the video game industry". In: *13th IEEE International Conference on Requirements Engineering (RE'05)*. 2005, pp. 240–250.

[4] Kate Edwards et al. *Developer Satisfaction Survey*. Tech. rep. International Game Developers Association, 2014.

[B1] Dina Friis, Jens Ostergaard, and Jeff Sutherland. "Virtual Reality Meets Scrum: How a Senior Team Moved from Management to Leadership". In: *Proceedings of the 2011 44th Hawaii International Conference on System Sciences*. HICSS '11. 2011, pp. 1–7.

[B2] Damian Hodgson and Louise Briand. "Controlling the uncontrollable: 'Agile' teams and illusions of autonomy in creative work". In: *Work, Employment & Society* 27.2 (2013), pp. 308–325.

[5] Christopher M. Kanode and Hisham M. Haddad. "Software Engineering Challenges in Game Development". In: *Proceedings of the 2009 Sixth International Conference on Information Technology: New Generations*. ITNG '09. 2009, pp. 260–265.

[A1] Jussi Kasurinen. "Games as Software - Similarities and Differences between the Implementation Projects". In: *CompSysTech '15: Proceedings of the 16th International Conference on Computer Systems and Technologies*. 2016.

[A2] Jussi Kasurinen, Risto Laine, and Kari Smolander. "How Applicable Is ISO/IEC 29110 in Game Software Development?" In: *Product-Focused Software Process Improvement: 14th International Conference, PROFES 2013, Paphos, Cyprus, June 12-14, 2013. Proceedings*. 2013, pp. 5–19.

[A3] Jussi Kasurinen, Andrey Maglyas, and Kari Smolander. "Is Requirements Engineering Useless in Game Development?" In: *Requirements Engineering: Foundation for Software Quality: 20th International Working Conference, REFSQ 2014, Essen, Germany, April 7-10, 2014. Proceedings*. 2014, pp. 1–16.

[A4] Jussi Kasurinen and Kari Smolander. “What Do Game Developers Test in Their Products?” In: *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. ESEM ’14. Torino, Italy, 2014, 1:1–1:10.

[6] Clinton Keith. *Agile Game Development with Scrum*. Addison-Wesley, 2010. ISBN: 978-0321618528.

[7] Clinton Keith. *The State of Agile in the Game Industry*. Accessed October 18th 2016. 2010. URL: `http://www.gamasutra.com/view/feature/4295/the_state_of_agile_in_the_game_.php`.

[8] Barbara Kitchenham and Stuart Charters. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Tech. rep. EBSE-2007-01. School of Computer Science, Mathematics, Keele University, and Department of Computer Science, University of Durham, 2007.

[B3] H. Kniberg and R. Farhang. “Bootstrapping Scrum and XP under Crisis A Story from the Trenches”. In: *Agile, 2008. AGILE ’08. Conference*. 2008, pp. 436–444.

[A5] Jussi Koutonen and Mauri Leppänen. “How Are Agile Methods and Practices Deployed in Video Game Development? A Survey into Finnish Game Studios”. In: *Agile Processes in Software Engineering and Extreme Programming: 14th International Conference, XP 2013, Vienna, Austria, June 3-7, 2013. Proceedings*. 2013, pp. 135–149.

[9] A. Kultima and K. Alha. “Hopefully everything I’m doing has to do with innovation: Games industry professionals on innovation in 2009”. In: *2010 2nd International IEEE Consumer Electronics Society’s Games Innovations Conference*. 2010, pp. 1–8.

[10] Craig Larman. *Agile & Iterative Development: A Manager’s Guide*. Addison-Wesley, 2004. ISBN: 978-0-13-111155-4.

[A6] Emerson Murphy-Hill, Thomas Zimmermann, and Nachiappan Nagappan. “Cowboys, Ankle Sprains, and Keepers of Quality: How is Video Game Development Different from Software Development?” In: *Proceedings of the 36th International Conference on Software Engineering*. ICSE 2014. Hyderabad, India, 2014, pp. 1–11.

[A7] Juergen Musil et al. *A Survey on a State of the Practice in Video Game Development*. Tech. rep. Vienna University of Technology, 2010.

[A8] Juergen Musil et al. “Improving Video Game Development: Facilitating Heterogeneous Team Collaboration through Flexible Software Processes”. In: *Systems, Software and Services Process Improvement: 17th European Conference, EuroSPI 2010, Grenoble, France, September 1-3, 2010. Proceedings*. 2010, pp. 83–94.

[B4] Ann Osborne O'Hagan and Rory V. O'Connor. "Towards an Understanding of Game Software Development Processes: A Case Study". In: *Systems, Software and Services Process Improvement: 22nd European Conference, EuroSPI 2015, Ankara, Turkey, September 30 – October 2, 2015. Proceedings*. 2015, pp. 3–16.

[11] Ann Osborne O'Hagan, Gerry Coleman, and Rory V. O'Connor. "Software Development Processes for Games: A Systematic Literature Review". In: *Systems, Software and Services Process Improvement: 21st European Conference, EuroSPI 2014, Luxembourg, June 25-27, 2014. Proceedings*. 2014, pp. 182–193.

[12] Amanda Peticca-Harris, Johanna Weststar, and Steve McKenna. "The perils of project-based work: Attempting resistance to extreme work practices in video game development". In: *Organization* 22.4 (2015), pp. 570–587.

[A9] Fabio Petrillo and Marcelo Pimenta. "Is Agility out There?: Agile Practices in Game Development". In: *Proceedings of the 28th ACM International Conference on Design of Communication*. SIGDOC '10. São Carlos, São Paulo, Brazil, 2010, pp. 9–15.

[13] Fábio Petrillo et al. "What Went Wrong? A Survey of Problems in Game Development". In: *Comput. Entertain.* 7.1 (Feb. 2009), 13:1–13:22. ISSN: 1544-3574.

[A10] Cristiano Politowski et al. "Are the Old Days Gone?: A Survey on Actual Software Engineering Processes in Video Game Industry". In: *Proceedings of the 5th International Workshop on Games and Software Engineering*. GAS '16. Austin, Texas, 2016, pp. 22–28.

[B5] R. Polk. "Agile and Kanban in Coordination". In: *Agile Conference (AGILE), 2011*. 2011, pp. 263–268.

[A11] M. Schmalz, A. Finn, and H. Taylor. "Risk Management in Video Game Development Projects". In: *2014 47th Hawaii International Conference on System Sciences*. 2014, pp. 4325–4334.

[14] Tom Sloper. "Game Production and Project Management". In: *Introduction to Game Development*. Ed. by Steve Rabin. Charles River Media, 2005. Chap. 7.1, pp. 1–2.

[15] Olli Sotamaa and Tero Karppi (ed.) *Games as Services: Final Report*. Tech. rep. Department of Information Studies and Interactive Media, University of Tampere, 2010.

[B6] Jayakanth Srinivasan and Kristina Lundqvist. "Organizational Enablers for Agile Adoption: Learning from GameDevCo". In: *Agile Processes in Software Engineering and Extreme Programming: 10th International Conference, XP 2009, Pula, Sardinia, Italy, May 25-29, 2009. Proceedings*. 2009, pp. 63–72.

[B7] P. Stacey and J. Nandhakumar. “Managing Projects in a Games Factory: Temporality and Practices”. In: *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*. 2005, 234a–234a.

[B8] Patrick Stacey and Joe Nandhakumar. “A temporal perspective of the computer game development process”. In: *Information Systems Journal* 19.5 (2009), pp. 479–497.

[B9] Patrick Stacey and Joe Nandhakumar. “Opening Up to Agile Games Development”. In: *Commun. ACM* 51.12 (Dec. 2008), pp. 143–146.

[16] F. Ted Tschang. “Balancing the Tensions Between Rationalization and Creativity in the Video Games Industry”. In: *Organization Science* 18.6 (2007), pp. 989–1005.

[B10] Martin Walfisz, Peter Zackariasson, and Timothy L. Wilson. “Real-time strategy: Evolutionary game development”. In: *Business Horizons* 49.6 (2006), pp. 487 –498.

[17] Michael Washburn Jr. et al. “What Went Right and What Went Wrong: An Analysis of 155 Postmortems from Game Development”. In: *Proceedings of the 38th International Conference on Software Engineering Companion*. ICSE ’16. Austin, Texas, 2016, pp. 280–289. ISBN: 978-1-4503-4205-6.

[18] Jane Webster and Richard T. Watson. “Analyzing the Past to Prepare for the Future: Writing a Literature Review”. In: *MIS Q.* 26.2 (June 2002), pp. xiii–xxiii. ISSN: 0276-7783.

[19] Megan A. Winget and Wiliam Walker Sampson. “Game Development Documentation and Institutional Collection Development Policy”. In: *Proceedings of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries*. JCDL ’11. Ottawa, Ontario, Canada, 2011, pp. 29–38.

[20] Claes Wohlin. “Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering”. In: *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. EASE ’14. London, England, United Kingdom, 2014, 38:1–38:10. ISBN: 978-1-4503-2476-2.

[B11] Peter Zackariasson, Martin Walfisz, and Timothy L. Wilson. “Management of Creativity in Video Game Development”. In: *Services Marketing Quarterly* 27.4 (2006), pp. 73–97.

[B12] Peter Zackariasson and Timothy L. Wilson. “Creativity in the Video Game Industry”. In: *Creativity: Fostering, Measuring and Contexts*. Ed. by Allessandra M. Corrigan. Nova Science Publishers, 2010. Chap. 6, pp. 109–120.