How Are Agile Methods and Practices Deployed in Video Game Development? A Survey into Finnish Game Studios

Jussi Koutonen and Mauri Leppänen

Department of Computer Science and Information Systems
P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland
jussi.koutonen@sysdrone.fi, mauri.leppanen@jyu.fi

Abstract. Agile methods and practices are largely deployed in software engineering. Game development shares many features that have given rise to the emergence of agility in software engineering. There is, however, a lack of understanding of the extent to which agile methods and practices are actually deployed in video game development and with which impacts. This paper reports on a survey into Finnish game studios. It shows that Scrum and, to a lesser degree, XP and Kanban are frequently used in the game studios. The most positive impacts of agility concern communication, quality of video games, and finding fun and implementable features earlier.

Keywords: video game development, Scrum, XP, Kanban**,** agile practices.

1 Introduction

The game industry is increasingly expanding. In 2007 the software portion of video game revenue was $9.5 billion, exceeding that of movies industry [38]. According to the forecast by PricewaterhouseCoopers [45], total global spending on video games will expand to $83.0 billion in 2016, growing at a 7.2 percent compound annual rate. The growth is expected to be rapid especially in the segment of online and wireless games with smartphones and tablets.

Game industry faces, however, a number of challenges. Players' expectations of getting "wow" and flow reactions in terms of visual appearance, script, sound world and technological novelty are growing. The games have to offer better and better player experience and co-experience [9]. Developing groundbreaking video games is very demanding [13]. The projects involve people with various expertise, making envisioning, communication, coordination and control most complicated [27, 36, 44]. Development budgets of high profile games are approaching the ones of Hollywood movies. Furthermore, game industry is a very competitive and risky field [42]. A publisher accepts a great risk in investing tens of millions for a development project without knowing whether the game is a success or not. It is estimated that only the top 5% of products make a profit. Industry employment is also fairly volatile, similar to other artistic industries [12].

H. Baumeister and B. Weber (Eds.): XP 2013, LNBIP 149, pp. 135–149, 2013.

© Springer-Verlag Berlin Heidelberg 2013

Game development has been traditionally based on the waterfall model or some of its variants [12, 50]. Due to its inspirational and unpredictable nature [56], many teams favor iterative processes including prototypes [44, 36]. An iterative process enables testing ideas in earlier phases and making rapid changes if necessary. This way a game's features emerge while the developers continually play-test to aspire a fun, entertaining and compelling game. An iterative and incremental way of working is inherent in agile methods [3], such as Scrum [54], XP (eXtreme Programming) [11], Lean [43] and Kanban [5, 32].

Although there is a large array of studies on games and their development, most of them discuss some specific issues, e.g. details of game mechanics or game experience. There are only some studies on game development methods and processes (e.g. [12, 57, 49, 50]). Yet fewer are those studies [27, 39, 40, 55, 56] that address agile game development. In particular, understanding of agile methods in use and their impacts on game development is yet to be achieved.

The research problem of our study is: To which extent game studios deploy agile methods and practices, and how they impact on game development? We accomplished the study as a survey targeted to Finnish game studios. The game industry has grown in Finland perhaps faster than in most countries. The total revenue of the Finnish game industry was estimated to be 335 million euro in 2011 [18]. Most of the studios are small, but there also are some large and successful studios, such as Rovio Entertainment (http://www.rovio.com/en/) whose product Angry Birds has become a worldwide phenomenon. The game has been downloaded more than 1 billion times (summer 2012). Another flourishing studio is Supercell with its Clash of Clans.

The remainder of the paper is organized into six sections. In Section 2 we shortly discuss video games and game development. In Section 3 the agile approach and agile methods and their use in software engineering are outlined. Section 4 provides a literature review of agile game development. In Section 5 we describe our research method and process, and Section 6 reports on the results of the survey. The paper ends with a summary and conclusion.

2 Video Games and Game Development

A *video game* is a game played by electronically manipulating images produced by a computer program on a monitor or other display (Oxford Dictionary). There is a large variety of game genres categorized by e.g. gameplay interaction, purpose, platform, and publisher [6, 63]. Every game has its rules. In addition to specific rules, game rules produce emergent properties such as player experience or playability that are quite difficult to predict or design. Player experience is different from that obtained from other home entertainments [52]. Looking at TV or films, reading books and listening music are passive entertainments which contain no interaction, whereas playing games a person can affect future events with his/her actions. Games can be fun in many respects [24]: sensation, fantasy, narrative, challenge, fellowship, discovery, expression, and submission.

Developing games is a complex endeavor [22, 13, 42] due to e.g. multiple disciplines, a large number of roles, divergent ambitions, conflicts of interests [21], and difficulties in anticipating what kind game will have a success in rapidly changing markets. These entail problems in schedules (cf. crunch time), coordination, team building, testing, and product family engineering [42, 19, 22, 39, 26]. Game development is particularly difficult for innovative, completely new kinds of games [33].

Game development is often compared to software engineering [42, 39, 36], and indeed, its outcome is software and to produce it similar phases have to go through. However, video games also resemble films in terms of creativity and aethestic components. Crawford [17] argues that game design is an art, science, a craft, or any combination of the three. There is one unique aspect that seems to separate the video game from traditional software: the requirement to be fun [35]. This requirement, unlike many others in software engineering, has no metric that can be applied. What is fun for one audience may not be for another. However, fun must be supported by and validated at each stage of the development process. To do this, games must be developed in a highly iterative manner.

There exists no single game development process model, which could act as a standard for the industry [37, 15]. Studios have different semi-formal or formal procedures [8] and philosophies [37]. However some commonalities exist. Development of a commercial game is usually divided into multiple phases which are defined by milestones [48, 49, 50]. Contracts between publishers and developers are typically based on these milestones [21]. Earlier the development process was based on the waterfall model or some of its variants [50, 12]. Nowadays, many teams use iterative processes including prototypes [44, 36], and some of them have adopted agile methods and practices [27, 39, 40].

There are some generic models that synthesize features of multiple methods of game development. Van de Weerd [61] used a formal method comparison approach to construct a reference method to give an overview of the phases, activities, steps and deliverables in the game development process. Manninen et al. [36] propose game development to consist of six major phases: concept, pre-production, production, quality assurance, release & launch, and post-release. Typical for game development is that the process is iterative [34, 56].

3 Agile Development

Software engineering has radically changed since the new millennium. The *agile approach* emerged to provide new values, principles and practices [3], particularly for situations characterized by e.g., hard to predefine and volatile requirements, first-to-market thinking, release orientation, dependence on good people, and negotiable quality [7, 14, 16]. The values emphasize 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 [3]. These have led to incremental, iterative and adaptive development.

Agility is a highly multifaceted concept with different meanings [16, 1, 25]. Conboy [16] develops a definition and formative taxonomy of agility, based on a literature review of agility across a number of disciplines. The definition goes as follow: agility means "the continual readiness of an ISD method to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment." [16, p. 340).

Agility is believed to reduce time-to-market, help coping with rapidly changing requirements and priorities, lower defect rates, improve product quality and process productivity, increase customer-value, as well as reach sustainable pace and balanced workload, thus improving developers' motivation and morale.

There is a large array of agile methods and principles, such as Scrum, eXtreme Programming (XP), DSDM, FDD, Kanban and Lean. The most used methods are Scrum, XP, and their combination and variants [58]. In the following, we shortly outline Scrum, XP and Kanban.

Scrum is "a framework within which people can address complex adaptive problems" [54]. In the literature, it is often said to be an agile method. Scrum has been built on three "pillars" (transparency, inspection, and adaptation), three main roles (development team, product owner, Scrum master), five events (sprint, sprint planning meeting, daily scrum, sprint review, and sprint retrospective), and three main artifacts (product backlog, sprint backlog, increment) [54]. Although the Scrum Guide [54] does not explicitly define a process, it is commonly associated with some kind of process model (see e.g. [2]).

XP (eXtreme Programming) is "a lightweight methodology for small to medium-size teams developing software in the face of vague or rapidly changing requirements" [10]. It provides a large set of practices that are divided into 13 primary practices and 11 corollary practices [11], meaning that the latter ones should be implemented after the primary practices have been taken into use. The set of XP practices include e.g., sitting together, cross-functional team, informative work space, stories, pair programming, quarterly cycle, ten-minutes build, continuous integration, and test-first programming.

Kanban has been derived from Lean thinking [62,43, 5]. In the simplest form, it is based on three principles [32]: visualize the workflow, limit WIP (work in progress), and measure the lead time. The first principle guides to split the work into pieces, write each item on a card and put on a kanban board. The second principle means that explicit limits are assigned to how many items may be in progress at each workflow state. The third principle tells to optimize the process to make lead times as small and predictable as possible.

In software engineering, the adoption of agile methods has already bridged the "crossed chasm" [59, 20, 4]. According to the latest survey [58] more than 80% of respondents said their organizations have adopted agile development practices. Scrum and Scrum/XP variants continue to make up more than two-thirds of the methods being used. Kanban and Scrumban were used in 6 % of the organizations.

4 Agile Game Development

There is a myriad of academic publications on agile software development, but only a few of them address agile game development. Here, we first quote Keith [27] and Musil et al. [39] to describe how they see the usage of agile values, principles and practices in game development. After that, we describe some empirical studies [40, 41, 56, 55] on how agile methods and practices are used in game development.

Clinton Keith [27] states that the values of the Agile Manifesto, with minor changes, are applicable for game development. He applies Scrum practices and presents four game development stages: concept, pre-production, production and post-production**.** In concept stage, ideas are generated, possibly prototyped, on a regular basis in time-boxed sprints. In pre-production stage, teams explore what is fun and how they are going to build assets to support it during production. They also create levels and other assets that represent production quality. In production stage, teams focus on creating an eight- to twelve-hour experience using the core mechanics and processes discovered during pre-production. This stage focuses on efficiency and incremental improvements. In post-production stage, teams polish the game experience, with the content brought to shippable quality. After that, the game is submitted to hardware testing.

Keith [27] criticizes some Scrum practices. Especially, the use of the sprint backlog in production stage causes problems in practice. That is why, he suggests Lean principles, in particular kanban, for production stage. From XP, suitable practices are informative working space, pair programming, continuous integration, test-driven development, user stories and short releases [27].

Musil et al. [39] propose a game development process, which is composed of three phases: pre-production, production and project closure. The main tasks of pre-production are to identify possible software project candidates, as well as to carry out requirement analysis, risk assessment and general project requirements like financing. Production receives the complete project package from pre-production and creates a sellable product with the given time, money and quality. The overall production workflow is based on Scrum, whereby it is separated into the three process time lines: vision loop, sprint loop, and validation loop. Project closure covers the distribution of the final game as well as retrospective analysis (post mortems), processing of created tools and integration of lessons-learned into the company's knowledge base.

Musil et al. [40] conducted a web-survey in the Austrian game industry (20 game studios) to identify the state of the practice and possible future trends regarding process and method support. Nine process methods were provided for the selection grouped into flexible (Scrum, XP, Agile/Lean), traditional (RUP, Crystal Clear, PSP/TSP) and unstructured methods (others). 23% of the respondents did not use any software process, but developed games ad-hoc. 77% of the studios applied flexible methods, and 61,5 % Scrum.

Petrillo and Pimenta [41] investigated how Agile principles and practices were adopted in game development, by gathering evidences through a postmortem analysis of 20 game development projects. 13 agile practices of Scrum, XP and Agile modeling methods were identified, including qualified team, belief in the success of the project, creativity stimulus, focus on the product, version control, using simple tools, and programming good practices. As can be seen, the list also contains general practices, not only agile practices.

Stacey and Nandhakumar [56] studied three computer game studios and recognized similarities between game development and agile development: getting feedback is equally important value although feedback in game development comes mostly from in-house, not from customers, and a fluid communication is an important value in agile development, as well as in game development. They noticed that the studios did not deploy agile methods as such but rather some of agile practices. Schofield [55] discusses the use of five XP practices (test-driven development, pair programming, continuous design, real customer involvement and energized work) in game development. He states that XP encourages the designer to steer the game during development and make more changes in the game design. XP practices focus development energy into delivering results quickly and keeping the project flexible.

5 Research Method and Process

Our research objective was to find out to which extent Finnish game studios deploy agile methods and practices and how their usage impacts on game development. From alternative research methods (e.g., case study, action research, and postmortem analysis) we selected survey [29] because data collected from a large population enables better generalization. To make the threshold of answering lower, we used an unsupervised survey [28] in which participants completed and submitted an online questionnaire through web browser and answers were recorded anonymously.

We took several steps to ensure that enough people return the survey with meaningful information [29]. First, we wanted to select respondents that are knowledgeable, willing and motivated to answer the questions. We used the language that is close to the one the practitioners use in their work. We presented questions in simple and unambiguous sentences to avoid misunderstandings and in a well-structured form to increase the clarity. The persons were also promised a copy of the research report for their reflection and benchmarking.

The population of the survey contained all the Finnish game studios with five or more employees. The size limit was based on an assumption that work in very small studios is not well organized and may apply more or less ad hoc ways of working. To find respondents, we contacted two professional associations, the Finnish Game Developers (http://www.pelinkehittajat.fi/) and Neogames (http://www.hermia.fi/neogames/). Using their lists of the member studios we asked each studio to name a knowledgeable person. This way we found 45 suitable studios, from which 37 gave direct contact information. We sent the invitation letter to them in July 2011, and got answers from 20 companies.

The questionnaire addresses four themes: background information, game development process, deployment of agile methods and practices, and experiences. *Background theme* concerns the general information about the game studios (no. of employees, age), their products (no. of game platforms and game genres), and development projects (size, no. of concurrent projects). These are relevant for analyzing impacts on the ways of developing games and applying agile methods and practices. *In Game development process theme* we were keen to learn which development tasks are accomplished and in which phases. These questions are based on a general phase structure derived from Keith [27] and Manninen et al. [36]. Unfortunately, we are not able here to report on the answers to these questions due to the space available.

Agile methods and practices theme was defined first to reveal which agile methods (Scrum, XP, Kanban, other) are used in each of the phases. Second, we wanted to find out which of nine Scrum practices, nine XP practices and three Kanban principles are deployed in the studios. Finally, in *Experience theme* we examined experiences the game studios had got from applying agile methods and practices. The questions were presented in the form of statements derived from Petrillo et al. [42], Musil et al. [40], and Keith [27].

The questionnaire was edited through several iterations, including pre-testing by four persons. After receiving the answers we followed the recommendations by Kitchenham [31]: the number of the answers for each question was checked (four respondents did not answer the questions about the use of agile methods and practices), for closed questions the distributions were calculated, and the answers to open questions were used to clarify the interpretation of the structured data.

The quality of a research study should be assessed in terms of reliability and validity. *Reliability* "is concerned with how well we can reproduce the survey data" [30]. "If another researcher later on conducted the same study, the results should be the same" [51]. We enhanced the repeatability of the survey by using the structured, web-based and pre-tested questionnaire, thus minimizing a researcher's effect on respondents.

Validity is concerned with "how well the instrument measures what it is supposed to measure" [30]. *External validity* is concerned with the extent to which it is possible to generalize the research results [51]. External validity is dependent on the size of the sample in relation to the population and its representativeness. We got a rather reliable estimate of the number of the Finnish game studios suitable to our study (N=45), and received answers from 20 studios. The response rate (44%) can be considered to be fairly good for making generalizations with regard to this population. In other countries, the sizes, funding principles and labor markets of the game studios, as well as the diffusion stage of agile methods in general can differ from those in Finland. Without knowing the contextual factors, generalization should be considered with care. On the other hand, the Austrian survey [40] shows that in corresponding circumstances the use of agile methods can be similar. *Internal validity* can be assessed in terms of several types of validity. Here, we consider *content validity* that is a subjective assessment of how appropriate the instrument seems to persons with the knowledge of the subject matter [30]. In order to address the subject matter in a proper way, the themes and questions were strongly based on relevant literature on game development and agile approach. The questionnaire was pre-tested by four persons.

6 Results

6.1 Background Information of the Game Studios

Based on the answers, the game studios were rather young: seven (35 %) studios had been on the markets for 0-2 years, eight (40 %) studios for 3-5 years and three studios for 5-10 years. Only one studio was over 10 year old. They were also rather small; half of the studios had 15 or less employees, five (25%) studios had16-50 employees, and only two (10%) studios had more than 50 employees. The most common game platforms were PC (70%) and mobile devices (65%). Nine (45 %) companies concentrated on one platform, while the others made games for 2-6 platforms. The most common game genres were Casual (70%), Action-adventure (30%), Platformer (30%), Adventure (25%), and Strategy (20%). Other genres included Simulation, Music, Racing, Serious and Role-playing games. Eight (40%) companies developed games of three or more genres.

Game development projects normally took less than one year. In seven (35%) companies, projects took less than half a year in average, in six (30%) companies ½-1 year, and in six companies for 1-2 years. In one company the projects took in average more than 3 years. The size of the project team was commonly 1-5 persons (35 %) or 6-15 (60%) persons. In one company, the size of the project team was 16-30 persons.

The last question of this theme concerned the number of concurrent projects. Five (25 %) companies had only one project at a time, six (30%) companies two projects, three (15%) companies three projects, four (20%) companies four projects and two (10%) companies five or more projects.

6.2 Agile Methods Used in the Game Studios

In the third part of the questionnaire, the respondents were asked about agile methods (Scrum, XP, Lean, Kanban, other) they are using in the game development phases (concept definition, pre-production, production, post-development, other phase). Scrum was the most common method: more than 50% of the companies deployed Scrum in production, pre-production and post-production phases, and more than 30% also in concept definition phase. The second common method (25%) was “Other method”. Lean was deployed in about 10 % of the studio in all the phases except post-production. Kanban was used only by one studio (in concept definition phase). Surprisingly, XP was not mentioned in this context. A likely reason for this is that XP was not recognized as a method but a set of practices. This is line in the finding of Stacey et al (2008) that game studios did not deploy agile methods as such but rather some of agile practices. Another explanation is that the studios using a customized mix of Scrum and XP answered “Other method”. Young studios did not use agile methods so largely as older ones, perhaps due to their more ad-hoc like processes. Three respondents explained their “Other method” answer saying that they use customized Scrum or other agile method. One answer to the open question elaborates that for concept definition they select a method case-by-case; sometimes Scrum, sometimes a “one-man innovation process”.

6.3 Agile Practices Used in Game Studios

The questions of this part addressed the use of nine Scrum, nine XP and three Kanban practices in the game development phases, with the following options: ”in all the phases”, ”in concept definition”, “in pre-production”, ”in production”, ”in post-production”, ”not used” and ”do not know”. Four respondents did not answer to the questions about the Scrum practices, and eight respondents did not answer to the questions about the XP and Kanban practices.

Scrum practices were most commonly used in the game studios (see Table 1). Yet, for each of the Scrum practices 25 - 44 % of the respondents said they do not use it. Daily Scrum and Sprints were the most deployed practices in ”All phases”. Most of the Scrum practices were more often used in pre-production and production phases than in concept definition and post-production phases.

Table 1. The use of Scrum practices in game development phases ([1] = all phases, [2] = concept definition, [3] = pre-production, [4] = production, [5] = post-production, [6] = not used, [7] = do not know) (n = 16)

Scrum practices	**[1]**	**[2]**	**[3]**	**[4]**	**[5]**	**[6]**	**[7]**
Daily Scrum	5	0	3	4	3	6	0
Sprint	5	2	7	7	4	4	0
Sprint planning meeting 1	2	3	6	7	3	6	0
Sprint planning meeting 2	1	3	5	7	3	6	0
Sprint review meeting	3	1	4	6	4	6	1
Sprint retrospective	1	1	4	5	2	7	1
Sprint burn down chart	0	1	3	6	2	7	1
Product backlog	4	2	7	8	5	4	0
Sprint backlog	4	1	7	8	5	4	0

From the large set of the XP practices [11] we selected those nine that were mentioned in existing literature on game development. The answers showed that XP practices are largely used (see Table 2). The most used XP practices were cross-functional teams, informative work spaces and continuous integration. The most uncommon XP practices were Quarterly cycle and Ten-minutes build. The latter result seems a bit surprising because Continuous integration and Ten-minutes build are commonly used together. Pair programming and Test-first programming were not largely used although they belong to the set of the primary XP practices [11].

Kanban practices were quite slightly used in the game studios. ”Limit Work in Progress (WIP)” was deployed in all the phases only by two game studios. Work Visualization was used in all the phases by one company, and another company deployed it in concept definition. No game studio ”measures the lead time”.

Table 2. The use of XP practices in game development phases ([1] = all phases, [2] = concept definition, [3] = pre-production, [4] = production, [5] = post-production, [6] = not used, [7] = do not know) (n = 12)

XP practices	[1]	[2]	[3]	[4]	[5]	[6]	[7]
Sitting together	4	3	2	1	1	6	0
Cross-functional team	5	2	3	3	2	4	0
Informative work space	4	2	3	3	2	4	2
User stories	2	1	2	2	0	7	1
Pair programming	1	1	2	1	1	7	2
Quarterly cycle	0	0	0	1	0	11	1
Ten-minutes build	0	0	0	0	1	9	2
Continuous integration	3	0	4	4	4	4	0
Test-first programming	2	0	4	1	0	6	3

6.4 Impacts of Agile Methods and Practices to Game Development

Finally, we asked the respondents' opinions about the impacts of agile methods and practices on game development. The questions about potential positive impacts were presented in the form of statements (e.g. Quality of code has improved), to which the respondents could answer: "Agree", "Partly agree", "Partly disagree", "Disagree", or "Do not know". The statements were based on assumptions of, and findings from empirical studies on, impacts of the agile approach on software engineering [53, 11] and game development [27, 40, 42]. They were divided into two groups: those concerning development work, and those involving project management. Four respondents did not answer to these questions. The summary of the answers is presented in Figures 1 and 2.

From Figure 1 we can conclude that all the statements of the impacts got more positive (agree, partly agree) than negative (partly disagree, disagree) answers. The most positive impacts was perceived as regards to communication between the professionals (60%), quality of games (more than 60%), and finding fun (60%) and implementable (60%) features more quickly. Other positive impacts involve such issues as team awareness and problems in game design. Interestingly, improvements in quality of code (30%) and communication between the stakeholders (35%) were not experienced so largely. Some of the issues were considered difficult to assess (cf. Testing games, Quality of code).

From Figure 2 we can see that the opinions are divided more strongly than above. The most positive impact was seen to occur in the easiness of project management (65%), scope management (55%), and sticking to the dead line (50%). Instead, despite the use of agile practices there still existed problems in feature creep and overwork, especially in the later part of the project. Some of the issues on project management were difficult to assess (cf. replacements in the personnel).

In the final question, the respondents were asked to describe potential negative impacts of using agile practices on game development. Examples of the negative impacts reported are:

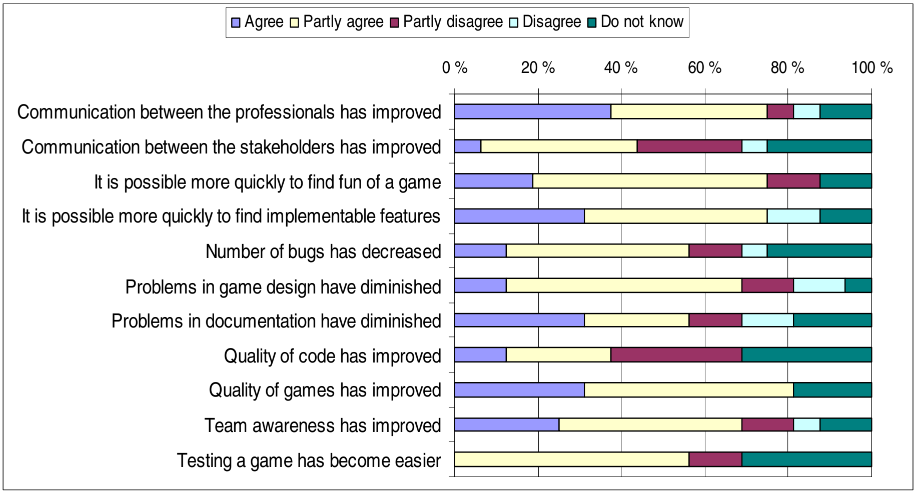

Fig. 1. Positive impacts of agile methods and practices to game development work (n = 16)

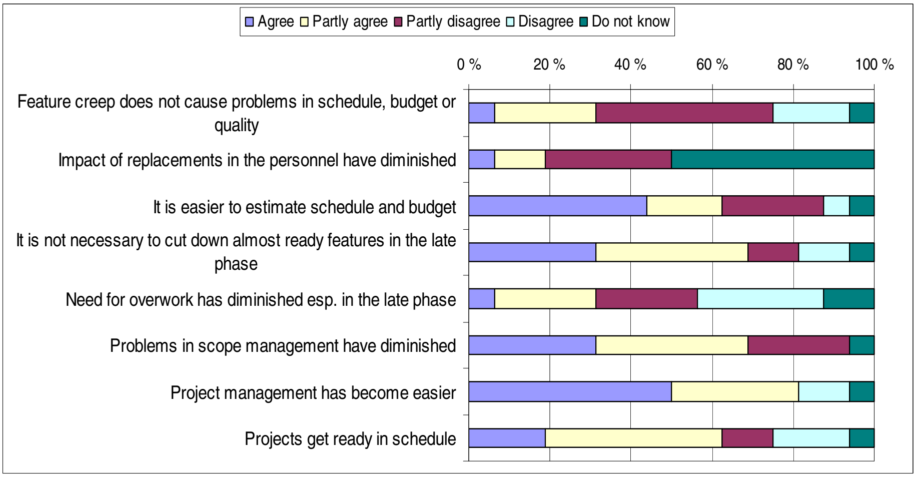

Fig. 2. Positive impacts of agile methods and practices to project management of game development (n = 16)

> *"Agile methods transfer responsibility to teams and rules of working get an essential role. This does not necessarily suit all the teams. "*

> *"Taking the inevitability of changes a bit too much for granted creates a window of opportunity for feature creep"*

> *"At the worst, there exists continuous crunch time if [rules for] sprints are taken too seriously."*

7 Discussion and Conclusion

This paper explores the use of agile methods and practices in Finnish game studios. Based on the survey, the Finnish game studios are rather young and small, yet including some fairly large ones (e.g. Rovio Entertaiment, 224 employees in 2011, 500+ in 2012). The game platforms are mostly PC and mobile devices. Development projects in average are small in terms of development time and number of employees. Reasons for that presumably are the industry's rapidly changing and risky nature, funding problems, and small job market. In this regard, the situation resembles the one in Austria [40].

All the studios, except one, deployed agile methods at least in some of the development phases. The most commonly mentioned method was Scrum, as was also the case in the Austrian survey [40]. XP, Lean and Kanban were used in the smaller scale. Some studios applied a mix of several agile methods, but not in the way Keith [27] suggested, i.e. the use of Lean and Kanban in production phase. Instead, agile processes resembled more the one suggested by Musil et al. [39]. No dependences between the use of agile methods, on the one hand, and the size of the studios or the number of the concurrent projects, on the other hand, were found.

At the level of agile practices, the survey showed that Sprint and its related events (Daily Scrum, Sprint planning, Sprint review) and artifacts (Product, Sprint backlog) from Scrum, as well as cross-functional teams, informative work space, and continuous integration from XP were in large use. These findings were as expected when taking into account special features of game development [33, 42, 44]. Compared to a large survey on Agile and Lean usage in Finnish software industry in 2011 [47], agile methods and practices were less frequently used in game development.

The survey indicated that agile methods and practices benefit game development in many ways. An iterative and incremental process enables inventing, designing and testing ideas of a playable game and betters the quality of the game (cf. [55]). Benefits were also perceived as faster recognition of fun and implementable features, and as better communication. The finding that quality code was not improved makes to suspect that continuous integration was not always applied in a proper manner (cf. together with automated testing). Although agility was seen to help scope management, estimation of schedule and budget, and sticking to the schedule, there were still problems as regards overwork and feature creep (cf. [42]).

As the survey included quite a large sample of the game studios in Finland, it provides a good descriptive view to the state of agile game development. However, the study has some limitations. First, the findings can only be generalized into the contexts with situational features similar to Finland. Second, to obtain a deeper insight into the use of agile methods and practices in the game studios, how this use evolves [60], and how it affects the productivity and quality of game development, we need a series of case studies. We should also pay more attention to ways the game studios customize and deploy agile methods and practices to match them to their needs. Despite of these limitations, the study provides interesting information about the current state of agile adoption in game development, which is of value for those who are considering how to improve the productivity and quality of their game development.

References

1. Abbas, N., Gravell, A.M., Wills, G.B.: Historical roots of agile methods: Where did ‘Agile Thinking’ come from? In: Abrahamsson, P., Baskerville, R., Conboy, K., Fitzgerald, B., Morgan, L., Wang, X. (eds.) XP 2008. LNBIP, vol. 9, pp. 94–103. Springer, Heidelberg (2008)
2. Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J.: Agile software development methods – Review and analysis, vol. 478. VTT Publications (2002)
3. Agile Alliance: Agile Manifesto (2001), http://agilemanifesto.org/
4. Ambler, S.W.: Surveys Exploring the Current State of Information Technology Practices (2011), http://www.ambysoft.com/surveys (accessed April 2011)
5. Anderson, D.: Kanban – Successful evolutionary change for your technology business. Blue Hole Press (2010)
6. Apperley, T.: Genre and game studies: towards a critical approach to video game genres. Simulation Gaming 37(6), 6–23 (2006)
7. Baskerville, R., Pries-Heje, J.: Short cycle time systems development. Information Systems Journal 14, 237–264 (2004)
8. Bates, B.: Game design, 2nd edn. Thomson Course Technology (2004)
9. Battarbee, K.: Co-experience. Understanding user experience in social interaction. Dissertation thesis, University of Art and Design, Helsinki, Finland (2003)
10. Beck, K.: Extreme programming explained: embrace change. Addison-Wesley (1999)
11. Beck, K., Anders, C.: Extreme programming explained: embrace change, 2nd edn. Addison-Wesley (2004)
12. Bethke, E.: Game development and production. Wordware Publishing, Inc. (2003)
13. Blow, J.: Game Development: Harder Than You Think. Queue 1(10), 28–37 (2004)
14. Boehm, B., Turner, R.: Rebalancing your organization’s agility and discipline. In: Maurer, F., Wells, D. (eds.) XP/Agile Universe 2003. LNCS, vol. 2753, pp. 1–8. Springer, Heidelberg (2003)
15. Chandler, H.: The game production handbook, 2nd edn. Infinity Science Press (2009)
16. Conboy, K.: Agility from first principles: reconstructing the concept of agility in information systems development. Information Systems Research 20(3), 329–354 (2009)
17. Crawford, C.: The art of computer game design. McGraw-Hill (1984)
18. FIGMA (2012), http://www.figma.fi/index.php/tiedotteet/113-suomen-pelimyynti-laski-viime-vuonna-pelikehittajat-huikeassa-kasvussa
19. Flynt, J., Salem, O.: Software Engineering for Game Developers, 1st edn. Software Engineering Series. Course Technology PTR (2004)
20. Forrester: From agile development to agile engagement. Forrester Research (May 2009), http://www.forrester.com/research
21. Fullerton, T., Swain, C., Hoffman, S.: Game design workshop: designing, prototyping and playtesting games. CMP Books, San Francisco (2004)
22. Gershenfeld, A., Loparco, M., Barajas, C.: Game plan: the insider’s guide to breaking in and succeeding in the computer and video game business. St. Martin’s Griffin Press, New York (2003)
23. Gibson, A.: Agile game development and fun. Technical Report, University of Colorado, Department of Computer Science (2007) http://www.cs.colorado.edu/department/publications/theses/docs/bs/andrea_gibson.pdf

24. Hunicke, R., LeBlanc, M., Zubek, R.: MDA: A formal approach to game design and game research. In: Proc. of the Challenges in Game AI Workshop (2004)
25. Iivari, J., Iivari, N.: The relationship between organizational culture and the deployment of agile methods. Information and Software Technology 53(5), 509–520 (2011)
26. Kanode, C., Haddad, H.: Software engineering challenges in game development. In: 6th Intern. Conf. on Information Technology: New Generations, pp. 260–265 (2009)
27. Keith, C.: Agile game development with Scrum. Addison-Wesley (2010)
28. Kitchenham, B., Pfleeger, S.: Principles of survey research. Part 1: Turning lemons in lemonade. SIGSOFT Software Engineering Notes 26(6), 16–18 (2001)
29. Kitchenham, B., Pfleeger, S.: Principles of survey research. Part 2: Designing a survey. SIGSOFT Software Engineering Notes 27(1), 18–20 (2002)
30. Kitchenham, B., Pfleeger, S.: Principles of survey research. Part 4: questionnaire evaluation. SIGSOFT Software Engineering Notes 27(3), 20–23 (2002)
31. Kitchenham, B., Pfleeger, S.: Principles of survey research. Part 6: Data analysis. SIGSOFT Software Engineering Notes 28(2), 24–27 (2003)
32. Kniberg, H., Skarin, M.: Kanban and Scrum – making the most of both, Enterprise software Development Series, InfoQ (2009)
33. Koivisto, E., Suomela, R.: Using prototypes in early pervasive game development. In: Sandbox Symposium, San Diego (2007)
34. Kreimeier, B.: Game design methods: A 2003 survey (2003), http://www.gamasutra.com/view/feature/2892/game_design_methods_a_2003_survey.php
35. Lewis, C., Whitehead, J.: The Whats and the Whys of games and software engineering. In: Proc. of Workshop on Games and Software Engineering, pp. 1–4 (2011)
36. Manninen, T., Kujanpää, T., Vallius, L., Korva, T., Koskinen, P.: Game production process: A preliminary study. University of Oulu, Finland (2006)
37. McGuire, M., Jenkins, O.: Creating Games: Mechanics, Content, and Technology. A K Peters (2009)
38. Moore, M., Novak, J.: Game industry career guide. Cengage Learning, Delmar (2010)
39. Musil, J., Schweda, A., Winkler, D., Biffl, S.: Improving video game development: Facilitating heterogeneous team collaboration through flexible software processes. In: Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R. (eds.) EuroSPI 2010. CCIS, vol. 99, pp. 83–94. Springer, Heidelberg (2010)
40. Musil, J., Schweda, A., Winkler, D., Biffl, S.: A survey on a state of the practice in video game development, Report IFS-QSE 10/04. Institute of Software Technology and Interactive Systems, Vienna (2010)
41. Petrillo, F., Pimenta, M.: Is agility out there? Agile practices in game development. In: Proc. of the ACM Int. Conf. on Design of Communication, pp. 9–15 (2010)
42. Petrillo, F., Pimenta, M., Trindade, F., Dietrich, C.: What went wrong? A survey of problems in game development. ACM Computer in Entertainment 7(1) (2009)
43. Poppendieck, M., Poppendieck, T.: Lean software development – An Agile toolkit. Addison & Wesley (2003)
44. Potanin, R.: Forces in play: the business and culture of videogame production. In: Proc. of the 3rd International Conf. on Fun and Games, pp. 135–143 (2010)
45. PricewaterhouseCoopers: Global entertainment and media outlook 2012-2016; Video games (2012), http://www.pwc.com/gx/en/global-entertainment-media-outlook/segment-insights/video-games.jhtml
46. Rabin, S.: Introduction to game development. Charles River Media (2005)

47. Rodriguez, P., Markkula, J., Oivo, M., Turula, K.: Survey on agile and lean usage in Finnish software industry. In: Proc. of ESEM 2012 Conference, pp. 139–148 (2012)
48. Rollings, A., Morris, D.: Game Architecture and Design. The Coriolis Group (2000)
49. Rouse, R.: Game Design: Theory & Practice. Wordware, Inc. (2000)
50. Rucker, R.: Software engineering and computer games. Addison Wesley (2002)
51. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering 14(2), 131–164 (2009)
52. Sanders, E.: Virtuosos of the experience domain. In: IDSA Education Conf. (2001)
53. Schwaber, K., Beedle, M.: Agile software development with Scrum. Prentice-Hall (2002)
54. Schwaber, K., Sutherland, J.: The Scrum guide – The definitive guide to Scrum: The rules of the game (2011), http://www.scrum.org/storage/scrumguides/Scrum_Guide.pdf
55. Schofield, B.: Embracing fun: Why extreme programming is great for game development. Gamasutra: The Art & Business of Making Games (March 2007)
56. Stacey, P., Nandhakumar, S.: Opening up to agile games development. Comm. of the ACM 51(12), 143–146 (2008)
57. Tran, M., Biddle, R.: Collaboration in serious game development: a case study. In: Proc. of the 2008 Conf. on Future Play, pp. 49–56 (2008)
58. VersionOne: State of Agile Survey – The state of Agile Development (2011), http://www.versionone.com/pdf/2011_State_of_Agile_Development_Survey_Results.pdf
59. Vijayasarathy, L., Turk, D.: Agile software development: a survey of early adopters. Journal of Information Technology Management 19(2), 1–8 (2008)
60. Wang, X., Coboy, K., Pikkarainen, M.: Assimilation of agile practices in use. Information Systems Journal 22(6), 435–455 (2012)
61. van de Weerd, I., de Weerd, S., Brinkkemper, S.: Developing a reference method for game production by method comparison. In: Ralyté, J., Brinkkemper, S., Henderson-Sellers, B. (eds.) Situational Method Engineering: Fundamentals and Experiences. IFIP, vol. 244, pp. 313–327. Springer, Boston (2007)
62. Womack, J., Jones, D.: Lean thinking: Banish waste and create wealth in your corporation. Simon & Schuster (1996)
63. Ye, Z.: Genres as a tool for understanding and analyzing user experience in games. In: Proc. of Conf. on Human Factors in Computing Systems, pp. 773–774 (2004)