

Game Development - Report



Game Development Disciplines and their Perspectives on the Software Engineering Process in Game Development A Case Study

Game Development - Field Studies
Spring term 2022

Fabian Fröding

Abstract

Previous research has discussed the potential differences between traditional software development and game development, and how the Agile development approach is used in the game industry. Other research reports that game development differs from traditional software development because game development includes various different disciplines.

This study was performed on-site at the office of a local game company to investigate the used development process and how the game industry might differ from the traditional software industry regarding Agile development, and how the development process in games are perceived by the involved disciplines.

By using in-depth interviews and direct observations, findings suggest that challenges in the game industry originate from management-related aspects and changing requirements from design departments, rather than from using similar development-approaches as in traditional software development.

Table of Contents

Introduction	1
Case Study	2
Method	3
Result	5
Discussion	9
Related Work	11
Limitations	12
Conclusion	13

1 Introduction

In game development, developers not only have deadlines to deal with, but also constantly changing desires of game designers (E. Murphy-Hill *et al.*, 2014). In addition, game developers regularly need to synchronise with said designers and also artists to get started on a task, since assets or decisions from those disciplines might be required.

In Agile development processes, an emphasis is put on fast releases of specific functionalities and often lessens the focus on the design of the system (L. Williams, A. Cockburn, 2003). As a consequence of focusing primarily on functionality, developers are often obligated to return to these items and rework their design when new features are demanded. This phenomenon is known as technical debt. Other sources define the process of reworking old program code as “refactoring” (M. Fowler *et al.*, 1999). When the initial architecture of systems fade as a consequence of excessive extensions, refactoring aids in maintaining the architectural design. However, due to intense deadlines, developers seldom have time to write maintainable code, which in the long run would arguably mitigate technical debt and allow developers to focus on the task at hand and reduce the amount of time spent on reworking or refactoring old code.

This study primarily focuses on providing insights on how the various disciplines involved in game development perceive the development process and how the disciplines affect each other. Consequently, the study may also bring new insights and findings for game research that studies the development process of games.

In this report, the term developer and programmer are used interchangeably, and refers to the company staff that writes program code. Designers and artists will be referred to as just designers and artists, but can also be referred to as developers in contexts that are more broad. Similarly, the term discipline is used interchangeably with department, and refers to the group of people responsible for a specific area such as programming or art.

The report was written in parallel with an internship at a local game studio in Skövde, Sweden, as part of a field-study course in the University of Skövde’s Game Development masters program. The internship presents an opportunity to conduct a study on the company’s development and management processes, and the different disciplines involved in the development process.

2 Case Study

The study was conducted as an ethnographic case-study at a local game company in Skövde, Sweden. The company consists of roughly 20 individuals. There are three main disciplines involved in the development process: 6 artists, 4 designers and 6 developers. Roughly, about half in each department are seniors while the other half are interns.

The company uses three week Agile-sprints with daily standups, where each employee writes about their current task on the platform Slack. Each monday there is a collective online-meeting to discuss the current sprint and to bring up eventual obstacles. These obstacles are referred to as “blockers”, which refers to an employee being dependent on the work of another employee from either the same or a different discipline and is prevented from making progress. In addition, each discipline has synchronisation meetings once each sprint (sometimes more often).

During the conduction of this study, most of the company staff was working on a rogue-like FPS (First Person Shooter) game funded by an external publisher. Some of the staff was working on other games as well.

3 Method

The methodology consists of three parts; *interviews*, *direct observations* and *analysis*, and will attempt to gain information about perspectives on the development process from the different disciplines involved in game development. In this study, the examined disciplines include; *programming*, *design*, and *art*. To gain information on the different development perspectives, the following research questions were formulated:

RQ1: How do game developers in different disciplines perceive the development process?

RQ2: How do the different disciplines involved in the development process affect each other?

In-depth interviews

In-depth, semi-structured interviews will be made with employees from the different disciplines within the company. The interview was designed to gain information about the employees' perspective on the development process and their challenges. In addition, the interviews will attempt to assess whether the employees are impeded by other disciplines involved in their tasks. Each interviewed employee will be categorised into a subgroup based on the disciplines defined earlier. The interview will primarily rely on open-ended questions to mitigate the chance of receiving non-elaborate questions.

The interviews were designed based on the guidelines presented by Cote and Raz (P. Lakonski, S. Björk, 2015). Their proposed design initially describes considerations for interviewing players, but also describes how the interview design is applicable to include interviews with game industry professionals. Therefore, their proposed interview design was deemed appropriate for the conduction of this study. Interviewees will be categorised into their appropriate discipline. For cases where interviewees have prior experience from disciplines other than their current one, they will be categorised into the one they are currently working in.

Despite the company consisting of roughly 20 individuals, this study focused on interviewing staff with employment, which meant excluding the interns. The primary reason for this was not because interns might have had less experience or a lack of things to contribute with, but rather because it would not be feasible to conduct that many in-depth interviews and provide a qualitative analysis on that much interview data for the scope of this study. Instead, it was decided to focus on one group of individuals within the company to interview and provide a more thorough analysis of their opinions, rather than mixing seniors and interns.

The interview structure consists of an *introductory script*, *warm-up questions* and *substantive questions*. The introductory script is aimed to inform the participant of the study and their confidentiality. The warm-up questions will focus on putting the participant at ease and gather basic data such as time in the game industry and in which discipline they would categorise themselves in. Finally, the substantive questions are formulated to provoke more elaborate answers that require more thought.

Since the game studio that this study is being conducted at only consists of Swedish individuals, the interviews were made in Swedish, and then the notes were translated into English afterwards. The interviewed participants were labelled as P1-Pn.

All interviews except one were conducted on-site in a meeting room at the company office, separated from the main room and undisturbed from the rest of the office. The interview that was not conducted at the office was done online through the platform Slack, since that employee was working remotely during the period when the interviews were made.

The exact formulations of the interview questions can be found in the Appendix, and the raw interview transcript can be found in a Github repository¹.

Direct Observation

The internship will provide hands-on exposure to the development process and will allow qualitative data collection based on observations made during meetings, conversations related to work-tasks, but also from less formal encounters with the company staff.

Analysis

The data from the interviews between the subgroups will be compared and discussed to find differences in how the various disciplines perceive the development process and how they handle challenges in the Discussion-section. In addition, the interview data will also be analysed to examine how the various disciplines might affect the quality of the work by each other.

¹ https://github.com/fabianfroding/it782a_interview_data

4 Result

Results from the interviews

In total, 9 interviews were conducted, consisting of interviews with 4 artists, 3 programmers and 2 designers. Only one participant (P4) had overlapping experience, in this case from both programming and art. Two of the participants (P1 and P7) had prior experience in traditional software development. P1 had worked as a software developer for 10 weeks as part of a summer job, while P7 had roughly 6 years of experience working in web and software development. The average working experience was 3 years for the interviewed programmers, 3.5 years for the interviewed artists, and 2 years for the interviewed designers.

The programmers

Based on the results from the interviews, the programmers reported communication being the most challenging aspect of working in game development. One of the programmers (P7) who had several years of experience in traditional software development mentioned that game development requires you to be much faster and be able to adjust things, compared to traditional software development where things are more rigid. Another programmer (P8) reported that requests from publishers can be very hard to work with since the publishers have no insight into the backend part of the project.

All programmers had similar opinions on the used development process. They felt that it fit the game industry very well to have an iterative sprint approach since they often have to deal with changing requirements from the design department and since it is also hard to know in advance what will make the game fun or not. None of the programmers had opinions on why an Agile approach would not fit the game industry. One programmer (P7), with experience in software development, also reported that he thought that development processes vary more from company to company than between traditional software development and game development.

Two of the programmers (P1 and P7) reported some sense of enjoyment collaborating with the other disciplines. One programmer (P1) reported that he experienced too little communication, while another (P7) reported that he usually ended up talking things over with the design department. Another programmer (P8) also expressed his difficulties with shared knowledge between the disciplines, and how getting too engaged in a discipline outside your own can negatively impact your own sprint performance.

The programmers varied greatly in their opinion about changing requirements. One (P1) responded with just accepting it and getting it done, while another (P7) explained his strategy to design flexible systems that can be reused later and how it's beneficial to have a mindset to design systems to be reusable. The last programmer (P8) responded with the drawbacks and his frustration with reworking systems and how its costs of a lot of time, which could have been saved were the requirements clearly defined from the beginning. All programmers reported that changing requirements and/or tight deadlines affected the quality of their work negatively, or at least had the potential to do so.

The Artists

The art department had varying opinions on what they thought was most challenging. One (P2) reported that the technicality of things as most challenging, while another (P3) mentioned feature creep and estimations as most challenging, while another (P4) mentioned lack of communication, but also too much communication, as the most difficult.

All artists except for one (P3) had positive views of the used process structure, i.e. sprints, and thought that it went very well into the game industry because of the constantly changing requirements and/or opinions. P3 agreed partially that the sprints fit, but also had some minor complaints about difficulties seeing the end-goal or understanding the whole picture of the product. P3 also reported that “blockers” were an occurring problem.

One artist (P2) mentioned that communication for artists mostly occurs within the art department itself. This is also further confirmed by another artist (P9), who mentioned that during old projects, communication between disciplines was more common, and that it was boring that there was less communication for the current project and that it instead goes through the project manager, but that it’s more effective. Two artists (P2 and P3) both reported that they have to adapt to changing requirements from the design department.

The art department also had varying opinions on how to handle changing requests. Two of them (P3 and P9) responded with just getting it done, while the other two (P2 and P4) used an approach that involved saving work for later in case things are requested to be re-implemented.

Two artists (P2 and P9) had opinions that their work was impacted negatively by the changing requirements, while the other two artists (P3 and P4) said that changing requirements impacted the quality of their work positively, thanks to being granted feedback on their work in the process. One artist (P4), mentioned that he could be greatly affected by the art department on performance related tasks, since he was responsible for visual effects in the game. Both P2 and P9 also reported that tasks that are on a tight time-limit will affect their work negatively.

The Designers

One of the designers (P5) mentioned that the hardest part of the development process is the communication with the other disciplines and to tie everything together. The other designer (P6) said that the hardest part for him was to structure his work and determine what things are urgent and what's not.

P5 also thought that the development process worked well but there were some things that could be better in regards to communication. He also explained that sometimes it’s better to not show the long-term plan and the involved tasks since that can make the different departments confused or overwhelmed, and instead hand out smaller tasks. On the other hand, he believed that in doing so, presenting new tasks could come unexpectedly.

The two designers varied quite in their opinion on why an Agile approach fit the game industry or not. P5 believed that it fit well because it provided a clear pipeline, but made

communication suffer, while P6 believed that it fit well because of the vagueness of the end-product. Both P5 and P6 thought the cross-discipline collaboration worked well and was even fun, but P5 mentioned again that a lack of communication can cause problems.

Regarding how the design department handled changing requirements, P5 said that it rarely happens in the design department, while P6 said that it depends on how urgent things are.

One of the designers (P5) reported that flexibility is necessary and that non-iterative approaches and strong opinions can have a negative effect on the quality of the work in the design department, and that feedback from the other disciplines can affect the quality of the final product. P6 reported that open communication can contribute to improving the overall quality of his work, and that it can provide a much more detailed explanation of a problem.

Results from the direction observation

General observations

The following section describes observations and notes taken during the time spent in the office of the game studio where the study was conducted.

The disciplines in the company consist of 6 programmers, 6 artists, and 4 designers. Each discipline has a dedicated synchronisation meeting scheduled once a week. The company works in 3-week Agile sprints, but longer sprints sometimes occur. In the beginning of each sprint, there is also a collective meeting where all employees participate. The used management platform is Jira, where a QA engineer manages most of the work, but each individual employee is also free to manage their tasks on their own.

In the programming team, work is submitted directly into the main workspace in the used version control system. Submitted code is by standard not reviewed immediately either. Instead, there is a collective code review meeting for the programming team once each sprint, but there are no guidelines as to which code should be reviewed and not, and as such there is no guarantee that all code changes are reviewed. Furthermore, the programming team also has a profiling session once each sprint. This session consists of running an analysis software on the game project to identify sources that significantly impacts the performance of the game. Depending on what the identified sources are, the responsible discipline will be tasked to make adjustments. For example, if lightning appears to impact the performance, then the art team will have to be responsible to make changes related to lightning so that the performance improves. Changes will not be evaluated until the next profiling section.

An internal playtesting session is also done by the entire development team once each sprint to try out the current state of the game. At the end of each sprint, each employee has PDT (Personal development time) which consists of freely choosing an area to deepen their knowledge in to enhance their toolkit, suitable for the role in the company.

As an intern, observations were made while working in the used game engine, Unreal Engine, where several chunks of visual code snippets were saved with notes that could be formulated as *“Save in case design changes their mind”*.

Individual observations

During a conversation with a senior programmer, an observation was made where the programmer had several research papers related to Pathfinding open on his office computer. When asked about the papers, the programmer said that they were tabs that he had forgotten to close, and that the findings in the research papers were discarded due to them being too theoretical and inapplicable to the actual game that was being worked on.

Another conversation with another senior programmer that was currently working as a consultant on an external project revealed additional findings. The external project was being developed in the Unity engine and C#, where the programmer explained the terrible state of the project's internals (referring to extremely poor code architecture, also known as “*spaghetti code*”) and the challenges working with such an architecture. He also mentioned that he believed that it would probably benefit everyone involved to simply start from scratch with the codebase, but that there just wasn't enough time for it due to short deadlines.

5 Discussion

Compared to programming, there did not seem to be any sort of recurring trends of challenges in the art or design departments; their opinions varied greatly and seemed to be more individual rather than department-oriented. The opinions of the programmers were more consistent with each other than between those from artists and designers.

It also seems that it is more individual how changing design requirements are handled, rather than how they are handled between departments. Both programmers and artists vary in their approaches, such as just getting it done, or designing their work to be adaptable to changes. Therefore, it seems more individual-based on how people handle these requirements. However, the interview data suggests that those who develop their works to be adaptable to changes were also less frustrated. Furthermore, communication seems to be more consistent between design and the other departments, than between, for example, programming and art. And for the most part, the lack of communication between non-design departments does not seem to necessarily have a negative impact on the development. However, the communication between the design and the other departments seems to be challenging for various participants.

Based on the results from the interviews, it appears that most of the challenges in the company originate from changing design decisions from the design department. Communication between the disciplines also seems to be a recurring theme related to issues within the development process. The results from the interviews do not seem to indicate that an Agile approach for game development is problematic. On the contrary, according to the interview data, it seems to be very suitable since an iterative approach works as a solution to the changing requirements.

Furthermore, there were few mentions of any negative impact of tight deadlines on the development side. However, this might have been because of the specific publisher for the current project. In one of the other external projects that one of the programmers worked as a consultant on, tight deadlines were without question a root cause for bad code quality and “*spaghetti code*”.

Based on these findings, it seems that the programming-discipline in game development shares more consistencies on how they perceive the development process, while the art and design disciplines vary more. Generally, all disciplines seemed to have a positive perception on the used Agile development approach and its sprint-structure and there were not any major differences in their perception of it (**RQ1**). There were no reports of it having a negative impact on any aspect of the development process either. This is not surprising, since Agile uses an iterative approach that allows developers to be adaptable to changes. Instead the interviews suggest that communication with the design department is a more pressing issue and that their changing requirements in themselves are demanding and challenging, while not necessarily being related to the development process. The design department did not suffer from design decisions problems since, not surprisingly, they are the ones who make them.

Based on the information gathered from the interview, it seems that the design department is the department that affects the work of other disciplines the most, and generally in a less

positive way, while the effect of the non-design departments in some cases affect each other by communication problems (**RQ2**). In some rare cases, the communication between departments also had a positive impact on each other, for example through feedback.

Some findings from the direct observations included visual coding-snippets with comments formulated to be saved in case the design department wants to reuse it. This agrees with what some of the participants said about saving their solutions when the design department changes their mind and wants to reimplement an old solution or feature, so that they don't have to rewrite things from scratch.

Based on the individual conversations with the programmer that was working as a consultant on an external project, we can also see that publisher relations play a greater role on the quality of the work of the developers. In this case, the external project that the programmer was working on did not have the same publisher as the project that the rest of the team was working on. So it seems then, that the publisher for this external project had unrealistic demands and deadlines which caused the quality of the codebase to suffer, since the programmer referred to the codebase as "*spaghetti code*". This could indicate that the type of publisher for a game development project has a greater impact on the product quality rather than the used development process.

6 Related Work

The argument that game development is different from traditional software development because game developers need to constantly adapt to the changing desires of game designers (E. Murphy-Hill *et al.*, 2014) does not necessarily make it distinct from software development. In software development, developers need to adapt and make changes to clients' changing preferences as well (M. Bano *et al.*, 2012). So in that sense, the two are quite similar.

A similar study was conducted that investigated how the software architecture of a game project adapts to changes (A. I. Wang, N. Nordmark, 2015). The study included developers from varying game companies of varying sizes. They found that most changes occur due to changing ideas in the creative teams, and that management always has the final word when it comes to decisions. If these findings are true on a more broad scale than in just that study, it is not strange that several major game titles suffer from game breaking bugs and quality issues, since last-minute decisions can require big changes.

In his book, Engström discusses the gaps between game development research and the game industry, and that most research related to game development are not done on industry cases and the ones that are, are scattered across multiple academic communities with different perspectives on game development (H. Engström, 2020). As such, this study contributes to lessening the gap by conducting research in direct collaboration with a game studio to document their development process, and may also serve as an example to be used in other studies looking to examine the development process of games.

7 Limitations

The results of this study were from the same single game studio and are not representative of the entire game industry. In-depth interviews are not useful for making inferences about a broader group, but provides a detailed level of personal depth and describes smaller groups (P. Lakonski, S. Björk, 2015). As such, the findings in this report may still serve as an important part for other research looking to combine documentation from several studios.

Even though most of the company staff was working on the same game, some individuals were also working on other projects. This means that if the feedback they reported during the interviews referred to the development process of one game, it might have been different from those that worked on another game. However, during the duration of the internship at the company and through direct observations and interaction with the involved people, it was ensured that all the interviewees at least partially had some involvement in the same project.

Some interviewees might have had experience from multiple disciplines. For example, an employee currently working as an artist might have worked as a programmer before. If such cases existed, it could mean that the results from those interviews are skewed, since the past experience of an employee might affect the perception in his current discipline.

8 Conclusion

This study investigated the development process in a local game company by using in-depth interviews on how the various involved game development disciplines perceived the process and were affected by each other. Based on the findings from the interviews, it seems that communication, changing design decisions, and publisher relations affect the work quality more than the type of development process. In this case, the company used an Agile approach, which was perceived as very suitable for the production of the game that they worked on, thanks to its adaptable approach. The involved disciplines in the development process perceived communication and changing design decisions to be more problematic than the development process in itself.

Future research related to the development processes in the game industry could benefit from the findings in this study. For example, the study could serve as a datapoint in research looking at the development processes and the involved disciplines in multiple studios.

References

- A. I. Wang, N. Nordmark, (2015), *“Software Architectures and the Creative Processes in Game Development”*, Norwegian University of Science and Technology.
- E. Murphy-Hill, T. Zimmermann, N. Nagappan, (2014), *“Cowboys, Ankle Sprains, and Keepers of Quality: How is Video Game Development Different from Software Development?”*, ICSE 2014, Proceedings of the 36th International Conference on Software Engineering.
- G. C. Ullman, C. Politowski, Y-G. Guéhéneuc, F. Petrillo, J. E. Montandon, (2022), *“Video-Game Project Management Anti-Patterns”*, GAS 2022, May 2022, Pittsburgh, PA, USA.
- H. Engström (2020), *“Game Development Research”*, University of Skövde 2020.
- L. Williams and A. Cockburn (2003), *“Agile Software Development: It's About Feedback and Change”*, Computer, vol. 36, no. 6.
- M. Bano, S. Imtiaz, N. Ikram, M. Niazi, (2012), *“Causes of requirement change - a systematic literature review”*, January 2012.
- M. Fowler, K. Beck, J. Brant, W. Opdyke, D. Roberts, E. Gamma, (1999), *“Refactoring: Improving the Design of Existing Code”*, Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- P. Lakonski, S. Björk, (2015), *“Game Research Methods”*, ETC Press 2015, pp. 93-116.

Appendix

Interview Questions
<i>Warm-up Questions</i>
Q1: How long have you been working in the game industry?
Q2: What's your favourite part of working in the game industry?
Q3: What's the most challenging part of working in the game industry?
Q4: What's your work experience prior to working in the game industry?
Q5: Which game development discipline would categorise yourself working in? Design, art and/or programming?
<i>Substantive Questions</i>
Q6: What is your personal experience/opinion on the used development process?
Q7: Why do you think the used development process is suitable or not suitable for the game development process?
Q8: What's your experience collaborating with the other disciplines involved in the development process?
Q9: How do you handle changing requests/suggestions/demands from other disciplines?
Q10: In what ways does collaboration with other disciplines affect the quality of your work?