



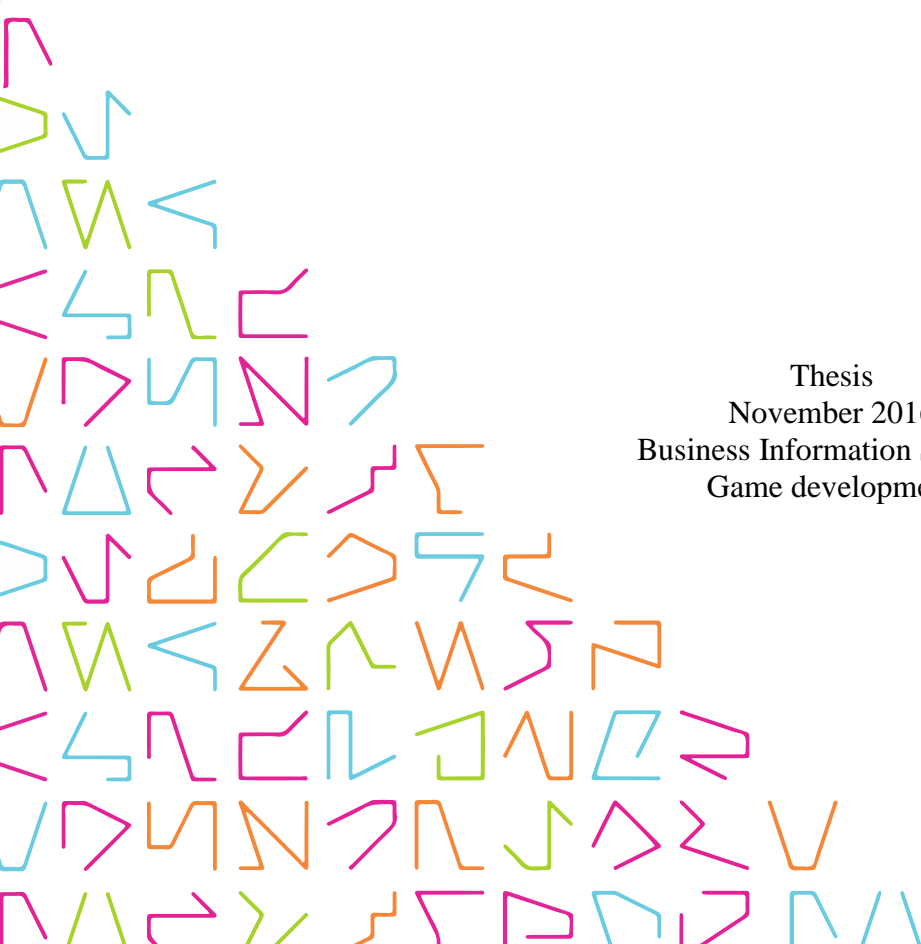
TAMPEREEN
AMMATTIKORKEAKOULU

THE ALMOST WELL DONE GAME PROJECT

Case Study of the Anti-Patterns in the Development
Process of Urban Empire

Juhana Mäntymaa

Thesis
November 2016
Business Information Systems
Game development



ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Business Information Systems
Game Development

MÄNTYMAA, JUHANA:
The Almost Well Done Game Project
Case Study of the Anti-Patterns in the Development Process of Urban Empire

Bachelor's thesis 55 pages, appendices 17 pages
November 2016

Urban Empire is a game that is currently in development by Reborn, formerly known as Fragment Production. The development has been plagued by issues which has led to the postponing of the project. This case study tried to map out if the problems with the development were systematic and if they could be grouped under known anti-patterns.

The study used both qualitative and quantitative methods to find evidence of different anti-patterns. Quantitative methods included a survey for the whole team whereas qualitative methods consisted of semi-constructed interviews of the team leads as well as workflow and document analyses. The documents analyzed were internal proprietary documents of Reborn and thus are not shown in the appendices.

The study identified five well-known anti-patterns in the development process as well as one completely new anti-pattern. The five anti-patterns were: Escalation of commitment, Groupthink, Dependency hell, Smoke and Mirrors, Ninety-ninety rule, with the sixth new one identified by the study being misallocation of resources.

The aim of the study was to create an internal document for Reborn to use in the future. The aforementioned anti-patterns were outlined, and a discussion was provided on how they affected the development and how the company could prevent them in the future.

In reviewing the results of the study, it is safe to say that identification of the anti-patterns plaguing the development was successful. The study was received well inside the company and will lead to changes in the following months.

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietojenkäsittely
Pelituotanto

MÄNTYMAA, JUHANA:

The Almost Well Done Game Project
Case Study of the Anti-Patterns in the Development Process of Urban Empire

Opinnäytetyö 55 sivua, joista liitteitä 17 sivua

Urban Empire on tammikuussa 2017 julkaistava peli, jota Reborn kehittää. Pelin kehityksessä on ollut monia ongelmia, jotka ovat johtaneet julkaisuajankohdan siirtämiseen syksyltä 2016 tammikuulle 2017. Tutkimuksen tavoitteena oli tunnistaa mahdolliset systemaattiset ongelmat, epämallit, jotka vaivasivat kehitystiimiä.

Tapaustutkimuksessa käytettiin sekä kvantitatiivisia että kvalitatiivisia menetelmiä. Kvantitatiivisena menetelmänä käytettiin koko kehitystiimille jaettua kyselyä, ja kvalitatiiviset menetelmät koostuivat työskentely- ja dokumenttianalyyseistä sekä puolikonstruktoituista haastatteluista. Tutkimuksessa analysoitiin Rebornin sisäisiä dokumentteja, jotka sisältävät yrityssalaisuuksia, tästä syystä ne eivät ole liitteissä. Kaikkiin opinnäytetyössä näytettyihin dokumentin osiin on saatu erillinen lupa.

Tutkimus tunnisti viisi jo ennalta tunnettua epämallia, ja tutkimuksen aikana luotiin myös yksi aivan uusi epämalli. Tutkimuksen pohjalta luotiin Rebornille sisäinen dokumentti, jossa sekä kerrottiin projektista löydetty epämallit että annettiin toimintaohjeita jatkoa varten näiden ongelmien ehkäisemiseksi jatkossa.

Tutkimus pääsi tavoitteisiinsa ja sai hyvän vastaanoton yrityksen sisältä. Tutkimus tunnisti epämallit onnistuneesti ja sisäisessä dokumentissa annetut käytännön ohjeet ohjaavat jatkossa yrityksen toimintaa tulevissa peliprojekteissa.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Reborn for giving me my first actual job and a paycheck in the games industry. Especially I would like to thank Juho Lyytikäinen for believing in me enough to hire me and Jussi Autio, for the many interesting conversations and nuggets of knowledge I have gained from him regarding game design. I would also want to thank Matej Komár, my partner in crime at reborn QA, for his friendship and great work ethic.

I'd also like to thank TAMK staff for their mentoring during these 3.5 years. Especially Jussi Ylänen, my mentor during this thesis project and Toni Pippola and Pasi Pekkanen for always putting students first. Also, thank you Gareth Noyce for all that I've learned from you.

Thanks to my family for their unyielding support and love. Thank you Säde for putting up with my stress and nerves during this fall. I love you all dearly.

ABSTRACT**ACKNOWLEDGEMENTS****TERMINOLOGY & ABBREVIATIONS****CONTENTS**

1	INTRODUCTION.....	8
1.1	Aim of the study	8
1.2	Review of the methods	9
1.3	What is an anti-pattern?	9
2	SUMMARY OF URBAN EMPIRE.....	10
2.1	Project	10
2.1.1	Reborn.....	10
2.1.2	The team.....	10
2.1.3	Pre-production.....	11
2.1.4	Production	11
2.2	Publisher	11
3	RESEARCH AND ANALYSIS.....	12
3.1	Quantitative research	12
3.2	Qualitative research	13
3.2.1	Lead interviews	13
3.2.2	Workflow analysis	14
3.2.3	Observational work	17
3.3	Analysing the results.....	18
3.3.1	Survey results	18
3.3.2	Interview analysis.....	30
3.3.3	Observations.....	31
4	ANTI-PATTERNS DURING THE DEVELOPMENT OF URBAN EMPIRE	32
4.1	Examination of anti-patterns during development of Urban Empire	32
4.1.1	Escalation of commitment	32
4.1.2	Groupthink	33
4.1.3	Dependency hell.....	34
4.1.4	Smoke and Mirrors.....	35
4.1.5	Ninety-ninety rule	35
4.1.6	Resource misallocation	36
5	CONCLUSIONS	37
	REFERENCES.....	38

APPENDICES 39

TERMINOLOGY & ABBREVIATIONS

Build	A version of a game
CSV	Comma separated values, a file extension which helps importing data to excel or other statistic softwares
GDC	Game developers conference
JIRA	A project management tool created by Atlassian
Pipeline	A workflow designed to do a certain task
PSPP	Open-source statistical analysis software
QA	Quality assurance
QA engineer	Quality assurance engineer
RPS	RockPaperShotgun, a games industry magazine
SPSS	IBM's proprietary statistical analysis software
UE	Short for Urban Empire
Vertical slice	Small slice of the game intended to showcase game mechanics to the publisher

1 INTRODUCTION

I started working at Reborn in April 2016 as a QA engineer and found myself in the middle of a large game project. I quickly realized that the game was behind schedule, that the team was tired and demotivated and that there was a lot of work to be done. The game, at that point, was almost unplayable; riddled with bugs and incomplete features and I remember thought crossing my mind:” Where am I getting myself into?” As the early shock started to dissipate I found that I was constantly asking myself ”What went wrong?” and ”How did we get here?”

An idea started to formulate in my head that the root cause for all these problems had been the lack of QA staff for most of the development. With that premise I started writing my thesis and creating surveys but as with most projects the deeper I got into it the more it felt like I was barking at the wrong tree. It seemed like there were systematic problems inside Reborn that had prevented a good quality game to be produced in the allotted time. And so I started again on a different path, this time looking at the workflows, surveys and interviews for patterns, *anti-patterns*.

1.1 Aim of the study

This case study aimed to analyze the workflows and practices of Urban Empire development team and to find out if there were any systematic problems in them. And if so, how to fix them. Towards that end I started creating an internal document for the entire Reborn company which was meant to highlight all the anti-patterns that plagued the development of Urban Empire. The research question for the study was:” What went wrong with the development of Urban Empire?” and” What kinds of anti-patterns plagued the development?”

1.2 Review of the methods

The study used both qualitative and quantitative methods. I did a survey for the whole Urban Empire team in summer of 2016 and analysed the results using an open source statistical analysis program PSPP. Based on the survey I interviewed all the team leads with a themed semi-constructed interview. I transcribed the interviews and analysed them. I also did a analysis on the internal Reborn documents, which helped me to pinpoint different anti-patterns.

1.3 What is an anti-pattern?

Anti-pattern is a form of practice, which tries to solve common problems, but ineffectively and is highly counterproductive. Anti-pattern is an antithesis of a software design pattern, a good habit or a practice of doing things (Koenig 1995, 46-48). Example of an everyday anti-pattern could be a dog owner not taking his dog for a walk daily. At first this seems to make the life of the dog owner simpler, but in the end, it leads to all sorts of problems in the dog's health.

Some anti-patterns are easier to notice than others. Anti-pattern such as vendor-lock in or dependency hell can be painfully obvious to all members of the development team, but are such in nature that they are extremely difficult to solve during development itself. Others, such as groupthink or resource allocation are more difficult to notice before it has negatively affected the project and usually can be responded to only in hindsight. With these problems, it's important to constantly review your work habits and try to respond as quickly as possible to the grassroots problems that the development team brings forth. Usually the game developers are aware of these issues long before the management hears about them. Using sprints and lean software development helps to also see these problems faster and gives tools to the management to do changes before it's too late.

Anti-patterns happen to even the best of teams and it doesn't necessarily tell anything about the competence of the development team. These problems must be considered even during project planning and have clear guidelines on how to avoid them. Hopefully this document will shed some light into these problems and help planning in the future.

2 SUMMARY OF URBAN EMPIRE

2.1 Project

Urban Empire is a strategy game dubbed city ruler (Internal Reborn Documents 2016). It has elements from different game genres. Technological advancement of "Civilization", the city building of "SimCity" and the society building of "Democracy". It has been shown in numerous publications, from RPS to Eurogamer. It was nominated as the "Best simulation game" in Gamescom 2016. It is slated for release 20th of January 2017.

In the game the player takes the place of a family member that is tasked with building a great city in a fictional state of Swarelia in the Austro-hungarian empire. The player then reigns the city through the different eras all the while trying to stay in power, by balancing politics with growing the city. The player also must balance the needs of different social groups and political parties.

2.1.1 Reborn

Reborn was formed in 2012 as Fragment production. It did a facelift in the Summer of 2016, rebranding itself as Reborn. Reborn is most well-known for its emergency service simulator series Rescue. They have developed 3 Rescue titles as of now. Reborn is quite a large company in the Finnish gaming industry with almost 30 employees.

2.1.2 The team

Team during the development of UE had many personnel changes which lead to over 20 people working in the project at some point or another. The development team consisted from an art team, code team and design team. Each lead by their respective leads. QA was part of design team. QA lead left the production fall 2015. First time QA joined the production after the departure of the QA lead was the spring of 2016.

2.1.3 Pre-production

Pre-production started in March 2015 and ended June 2015. The pre-production team consisted of 6 people, designer, two artists and two coders. (Reborn Internal Documents 2016) At the end of pre-production the lead artist left the production which left the art team without a direction and leadership. This led to problems down the line for the art team which affected the whole production. (Lead interviews)

According to the art director Niilo Altfan the pre-production was poorly done and at least art should have spent more time on pre-production and focused on different things during the pre-production phase. (Lead art interview)

2.1.4 Production

Production started in earnest right after the pre-production. The first big deadline was doing the vertical slice for Kalypso. During production team size varied from the high of 18 people to a low of 8. Most of the issues with the development happened during development, but some of them can be contributed to the mistakes done during pre-production. The game went public in March of 2016 at the GDC.

Other big timeline event for the production was line-up meeting, the first time where the game was played hands-on by the member of the press. The event was hosted by Kalypso and featured other up and coming games by the publisher as well.

2.2 Publisher

Kalypso is the publisher for the project. In addition to giving the budget for the game they have given support in the form of marketing, localization and external QA. Kalypso is a mid-sized German publisher based in Worms, Germany. Founded in 2006 by Simon Hellwig and Stefan Marcinek during the last 10 years it has risen to prominence in the German gaming scene with few game series, most notably Tropico. (Kalypso Media 2016)

3 RESEARCH AND ANALYSIS

3.1 Quantitative research

I used Google forms as a tool to create a survey for the development team during the summer of 2016, 14 people answered the study. I decided to use Google forms because it was well known tool for me and it had a lot of features that made analyzing the data easy. The ability to import CSV file directly into PSPP was also one of the bigger factors which lead me to using that instead.

Survey was conducted to the whole development team during the summer of 2016. Team members were contacted using the internal communication software Slack. The survey was conducted anonymously. Survey was done using Google Forms program. The main reason for using Google Forms was because I was already familiar with it and because of its integration with other Google programs, such as Drive. Important feature was also being able to export the survey results as CSV. This helped importing the results into PSPP, a open source statistical analytics program.

Main reason for using PSPP over SPSS, the IBM's industry standard version, was availability. SPSS license costs hundreds of euros and it offered nothing substantial over PSPP. During the analysis of the data PSPP felt good, but during visualization of the survey data it proved to be bit too restricted. The visualization of the graphs is pretty bad and the ability to change them is nonexistent. The overlapping of texts in the charts shows this for example. After importing the data to PSPP I cleaned the data and changed all the data points to their numeric values.

3.2 Qualitative research

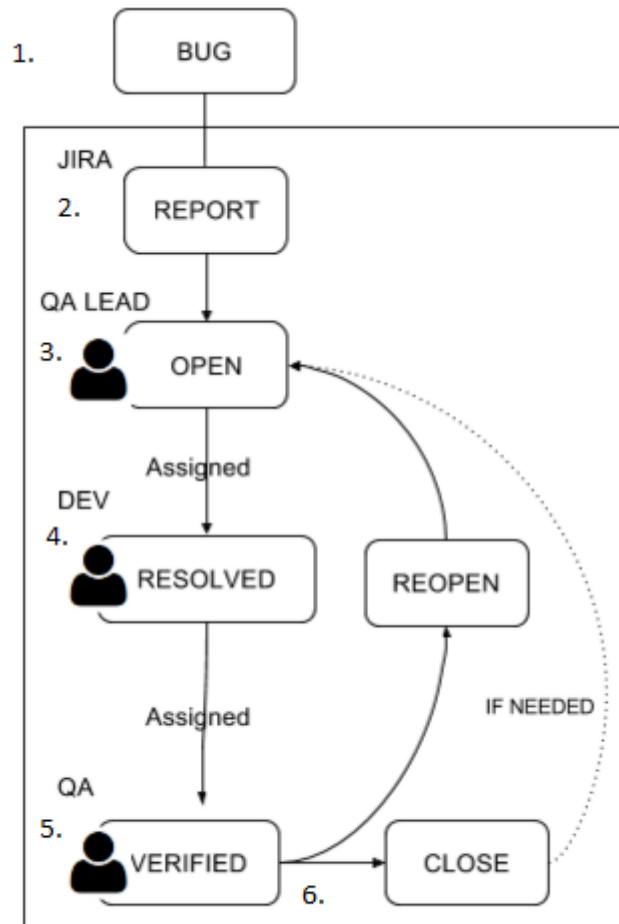
Qualitative research consisted of mostly interviews and observations but I also analyzed workflows of the team in regards to quality assurance. Being the senior member of quality assurance team helped me with the observations, because it intersected my work largely. Interviews were conducted after the survey had been analyzed, because I wanted to get more pinpointed information out of the leads.

3.2.1 Lead interviews

Lead interviews were conducted between 26.9. – 12.10. They were themed semi-constructed interviews. The questions were largely based on the data gained from the survey. I recorded the interviews and after that transcribed and analyzed them with focus on the issues and solutions found in the interviews for the projects many issues.

3.2.2 Workflow analysis

I looked at the workflows of the quality assurance pipeline as written in the reborn internal documents. These documents were created by the QA lead during beginning of



Example image: Bug Life Cycle

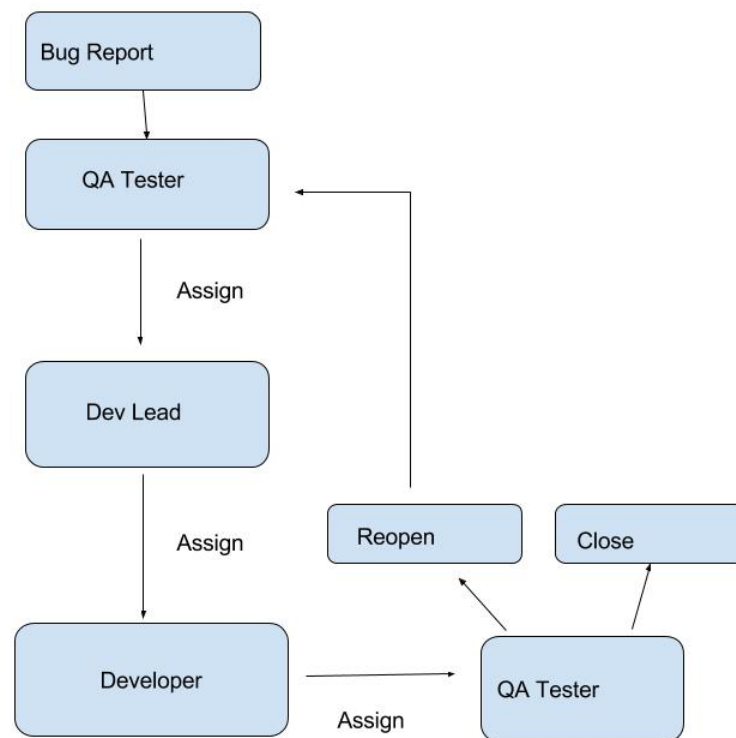
FLOWCHART 1. Proposed bug life cycle

production that was part of the project for about 5 months. Comparing these proposed qa pipelines to the actual ones gave a lot of information about the problem of not having a senior QA person.

The proposed bug life cycle was:

- 1) Bug is found
- 2) The person who found the bug reports it
- 3) QA assigns the bug to a developer whose realm of responsibility it belongs to
- 4) Developer resolves the issue
- 5) QA testers verify that the bug is fixed
- 6) If the bug is fixed QA testers close the issue, if not, they reassign it back to QA lead.

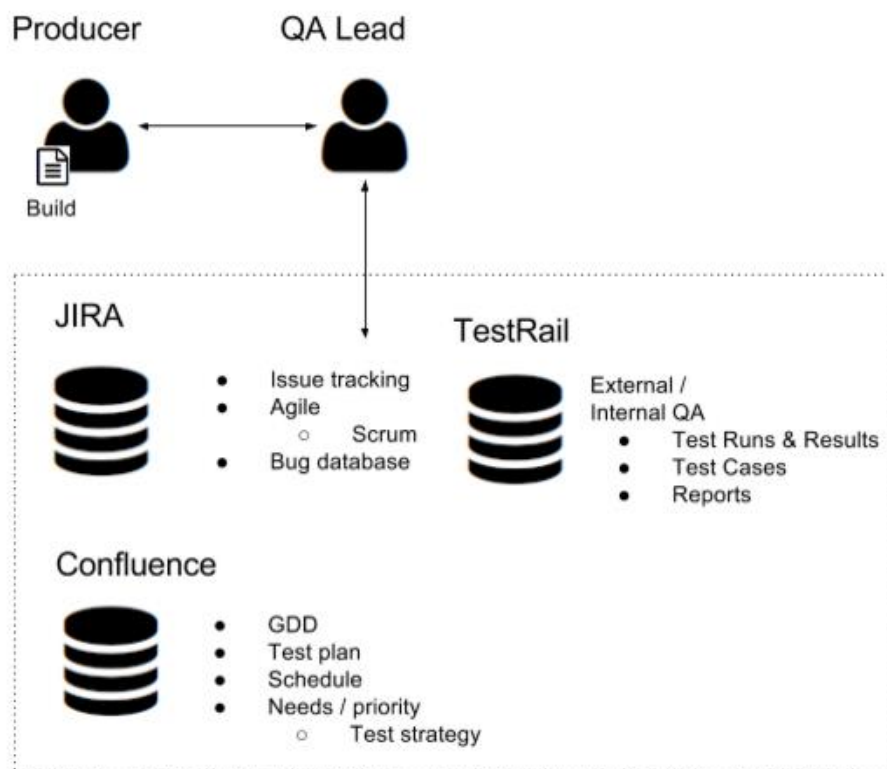
In actuality the development lacked a QA lead for most of the time and so the actual bug life cycle looked a bit different. The main difference was that instead of assigning task straight to developers, the QA's assigned tasks first to the department lead which in turn then assigned them to the right developer. So, the responsibility of balancing tasks between developers stays within the different departments. QA testers also have a larger role in this bug life cycle, where they need to actively keep track of the progress of different bugs.



FLOWCHART 1. Actual bug life cycle

The issue with this bug life cycle is that it is taxing for the QA testers. When there isn't any clear lead or senior member to take care of the tasks and assign them to tester, the testers are left to regulate their own work. With experienced QA team this might work, but with inexperienced workers it might lead to issues. Especially in the realm of communication. Without lead to take care of any issues that might arise in the testing itself it takes attention from the testers away from the actual testing.

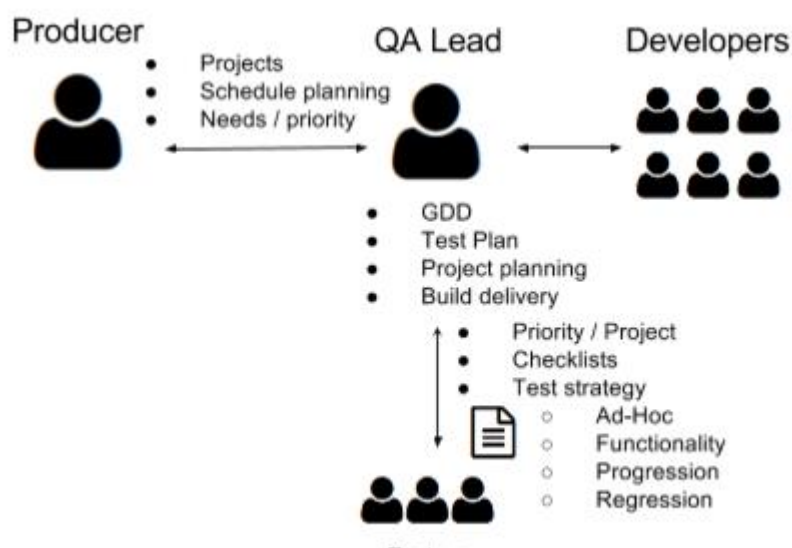
In the documents the QA lead also provided instructions for testing pipeline and chart about the role of a QA lead in regards to the rest of team. Without the QA lead the entire testing pipeline fell into the hands of the producer. This proved to be a bad decision, because it made it so that the producer couldn't focus on the actual producing completely. Slowly the testing pipeline and build pipeline was turned over to the QA testers completely, which freed up the producer and gave QA's better ownership of their work. This lead to increased activity in the project control system JIRA, which helped the development to react to issues with the game faster.



FLOWCHART 1. Testing pipeline

The picture above shows the proposed pipeline for testing. In the actual pipeline only JIRA was used for issue tracking and as a bug database. Google drive hosted GDD, schedule and other documents. External QA used JIRA as well and reported bugs there directly.

In the image below we can see the proposed QA lead role in the development. QA lead was supposed to work with producer about planning and prioritizing aspects of the game and with developers in regards to fixing different bugs. Without QA lead the job of the schedule planning in regards to testing timelines was left solely on the producer. This proved to work pretty well most of the time, sometimes the deadlines were too tight for the qa team to handle. All of this could have been prevented with a clear manager in the QA department.



FLOWCHART 1. Role of QA lead

3.2.3 Observational work

I observed the team during daily meetings and throughout the day. I documented these observations in my notebook. Mostly I wrote about the feels of the design team, as I was part of that. I also attended few daily meetings with the art department. My observational work is combined from these meetings.

3.3 Analysing the results

3.3.1 Survey results

14 people answered the survey during span of few weeks. There were few interesting results, that I want to highlight. They show that the team had experienced developers that were highly motivated, but that the development had issue.

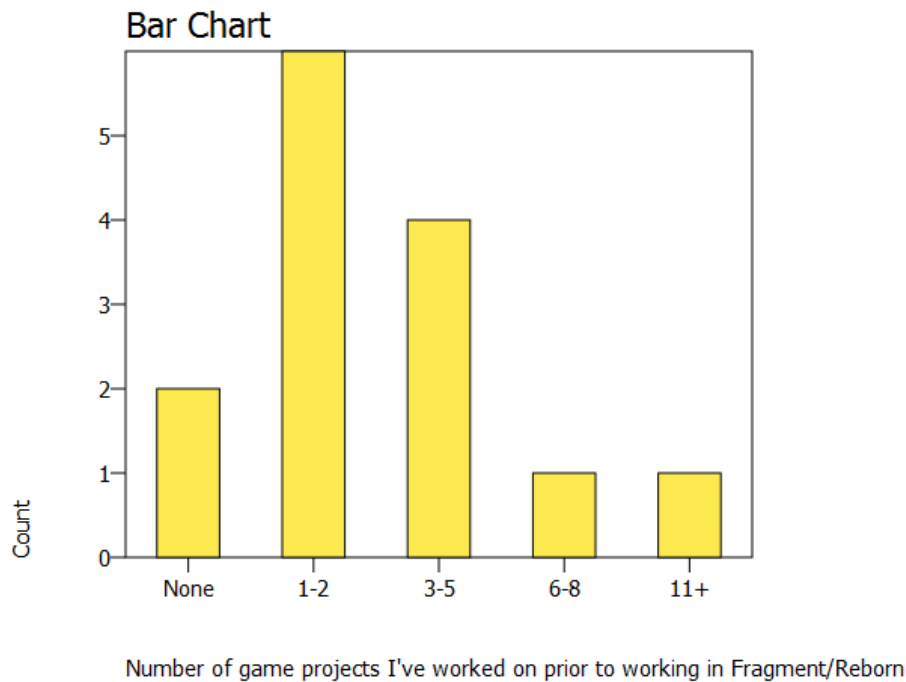


CHART 1. Number of game project worked on prior to working in Fragment/Reborn

In the chart above we can see, that only 2 people from the team are first timers for making games. 6 people have experience from developing 3+ games. This result was echoed during the lead interviews, with most of the lead saying that the teams experience wasn't an issue during development. Only the art lead said that the art team was too inexperienced for the game of this size.

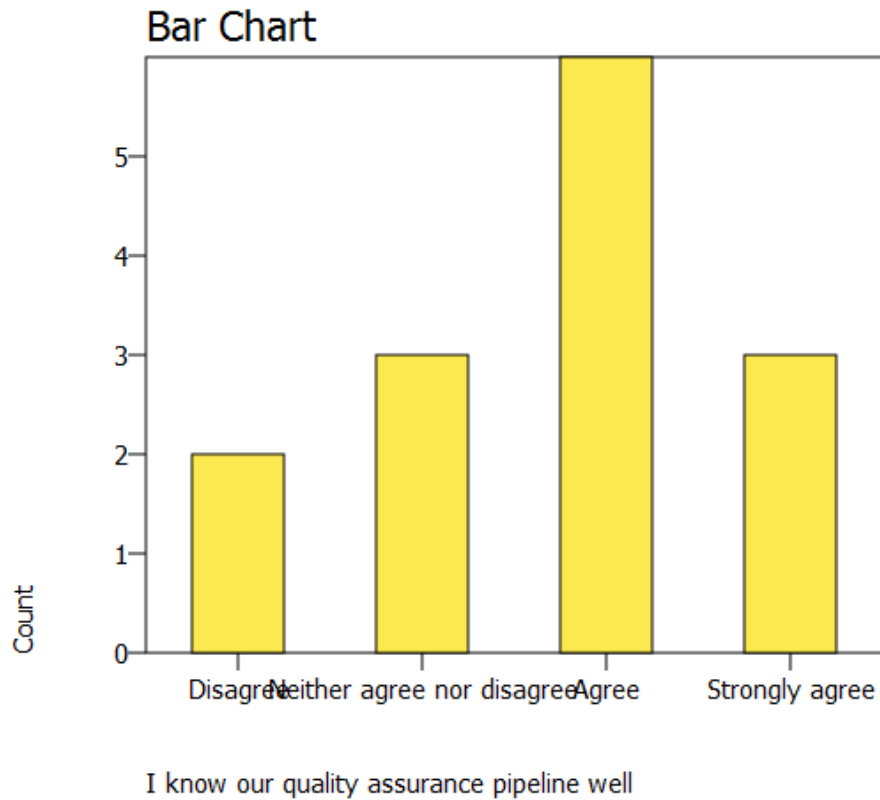


CHART 2. I know our quality assurance pipeline well

The team also evaluated their knowledge of the QA practices pretty highly, with only 2 team member disagreeing with the statement "I know our quality assurance pipeline well". The motivation for delivering high quality game was also high, which could be seen from the following question.

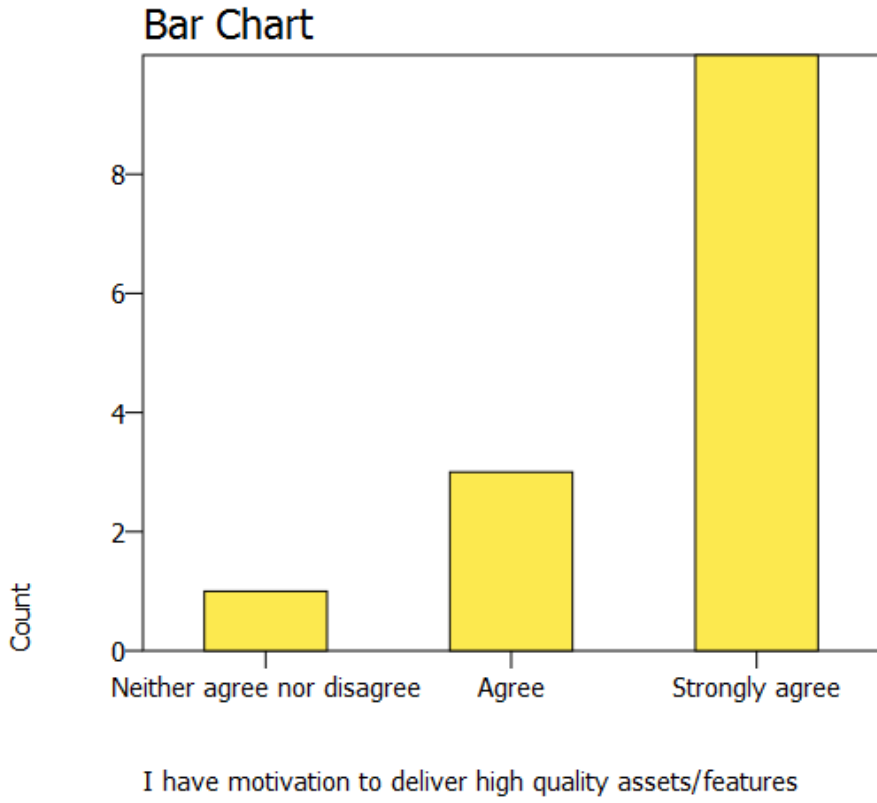


CHART 3. I have motivation to deliver high quality assets/features

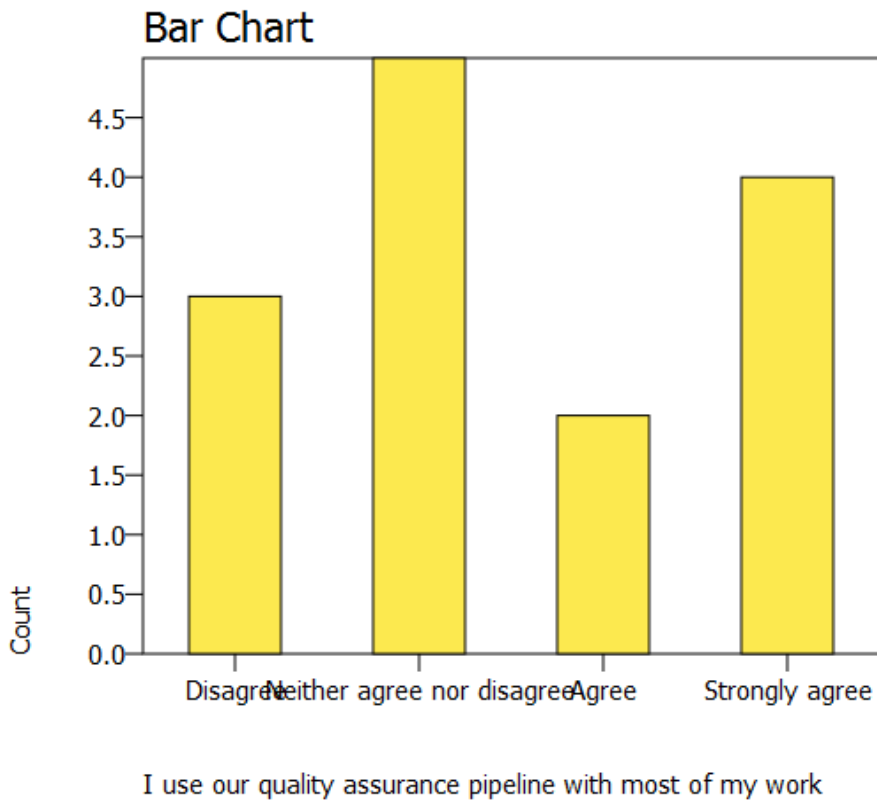
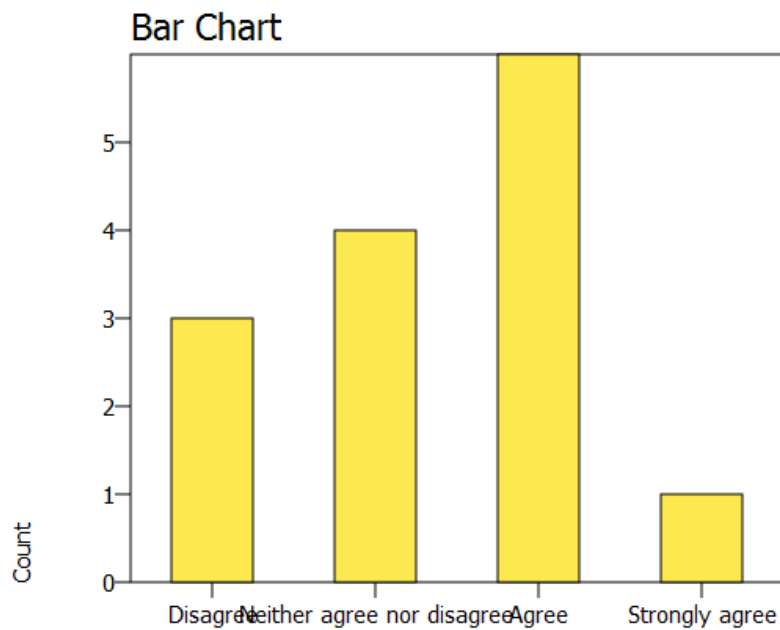


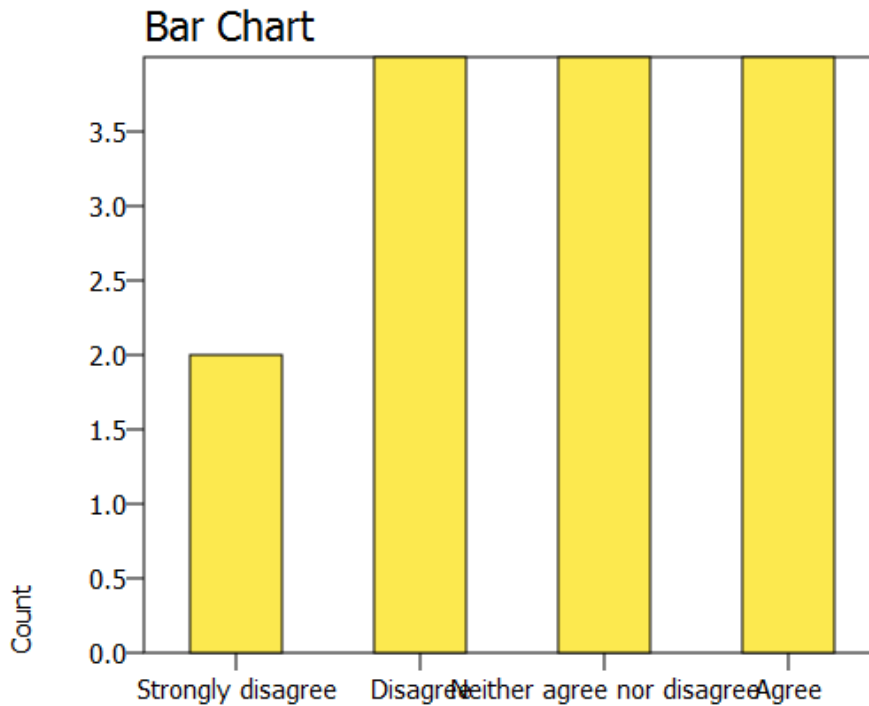
CHART 4. I use our quality assurance pipeline with most of my work

In the chart, above it is interesting to see that even though in the previous chart most of the team said that they understood quality assurance pipeline, we can observe that in this chart there were a lot of people who didn't use the QA pipeline consistently. In the chart below we can see one possible reason for this result, it shows that half of the participants didn't agree with the statement: "Our current quality assurance pipeline helps me deliver a high quality product". Other result that may be one of the reason behind the lackluster implementation of the quality assurance pipeline is the fact that people felt that they didn't have time to polish the features before adding them to the game. Only 4 participants agreed with that statement.



Our current quality assurance pipeline helps me deliver a high quality product

CHART 5. Our current quality assurance pipeline helps me deliver a high quality product



I have time to polish assets/features before adding them into the game

CHART 6. I have time to polish assets/features before adding them into the game

Questions were also asked about communication during pre-production (Ch and production. In the charts on the next page we can see that the team was dissatisfied with communication with only 1 participant answering that the communication worked great. It is also important to notice that communication started working better during production as we can see with the other chart. There were still people who felt that communication wasn't working during production, but there was also larger group of people who felt that the communication worked great. It is important to point out that the survey was conducted after changes had already happened to the communication practices and asking the same question few months prior might have yielded a different result.

Communication during pre-production(7 answers):

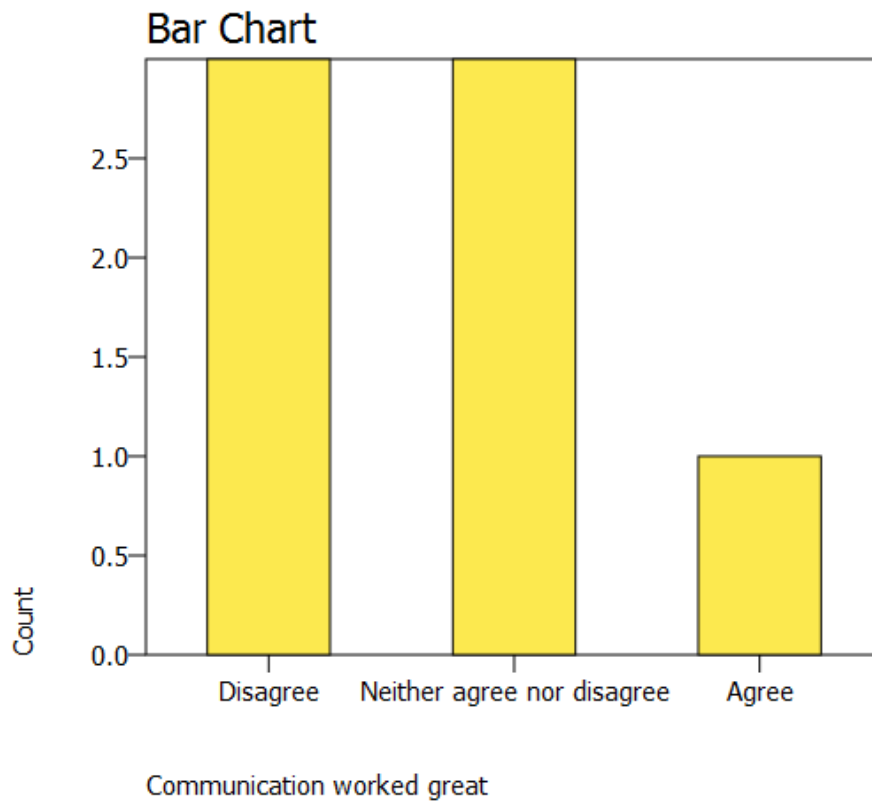


CHART 7. Communication worked great

Communication during production (14 answers):

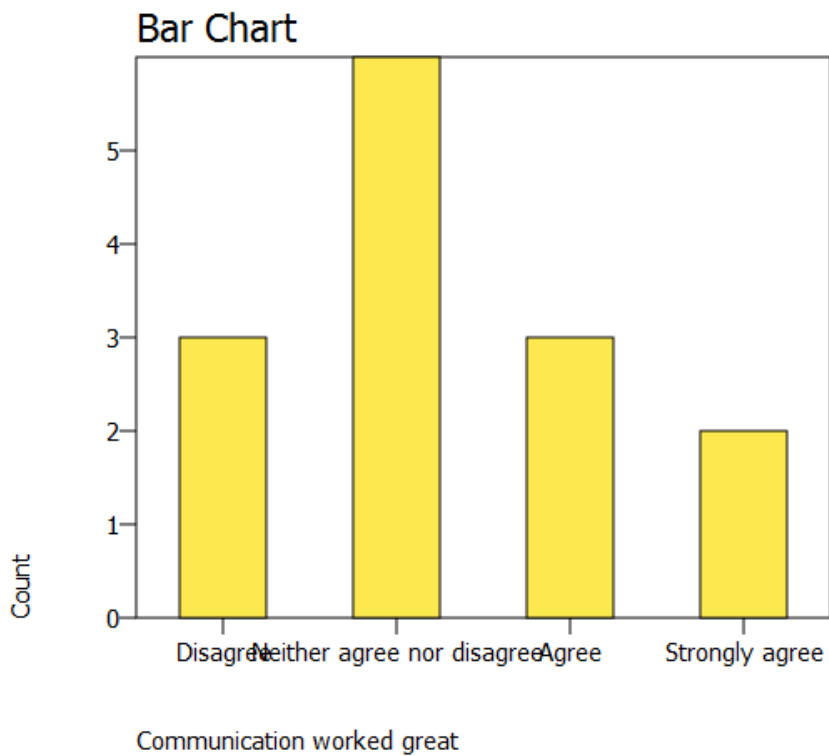
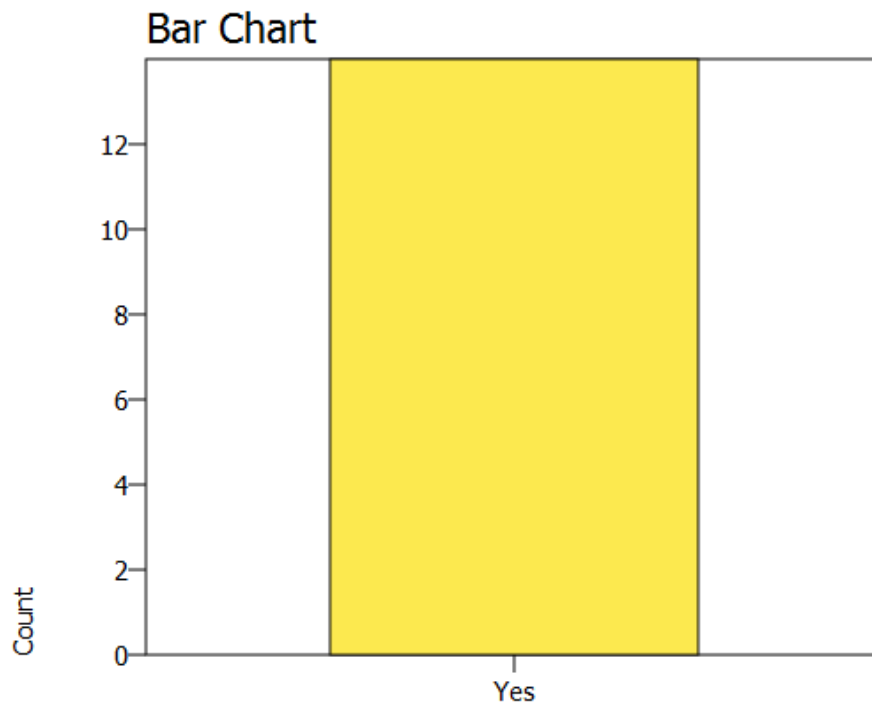


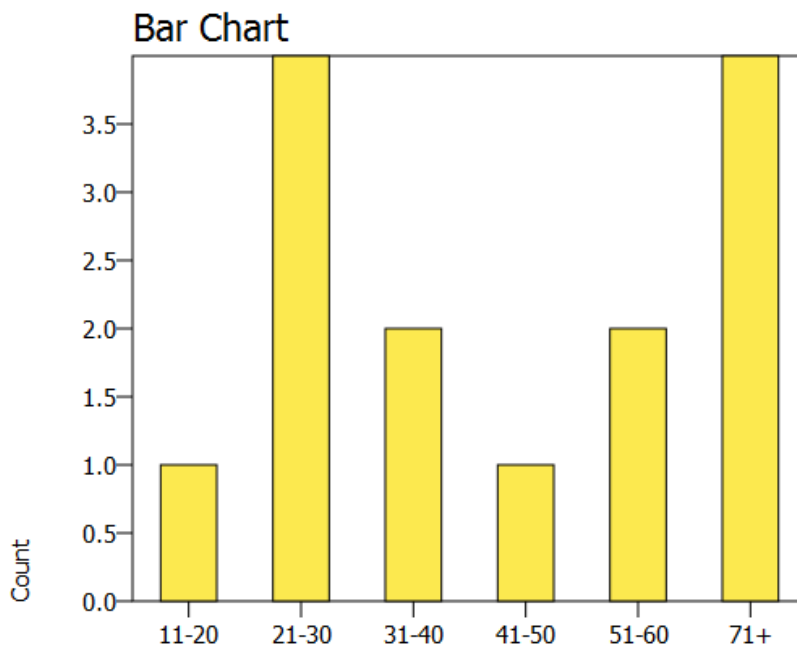
CHART 8. Communication worked great

Survey also highlighted the problems with the production. As we can see, 100% of the participants said that they did overtime during production. This is a clear sign that there were issues with the development that the team tried to compensate by overworking.



Did you do overtime during production?

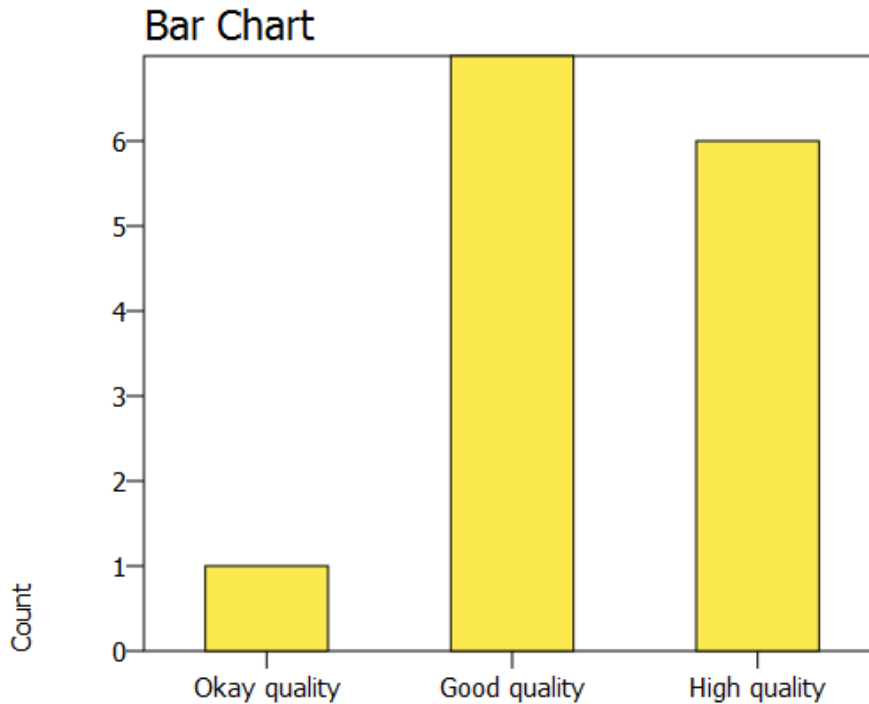
CHART 9. Did you do overtime during production?



How many hours did you do overtime during production?

CHART 10. How many hours did you do overtime during production?

If we look at the hours worked overtime, we can see that for some people the overtimes went as high as over 70 hours. Notice that no-one did under 10 hours of overtime. People still felt that the quality of the work is of atleast good quality (7 participants), only 1 participant estimated his/her work to be of "Okay quality" during overtime.



I feel that the quality of the work I did during overtime was...

CHART 11. I feel that the quality of the work I did during overtime was...

So what went wrong? Survey tried map out the reasons that affected the quality of the work of the development team. The participants answered the question: "Reasons preventing me from doing high quality work during production were..." and they evaluated that in 7 different categories from "No at all" to "Affected greatly". These results shed some light as to the problems inside the development team. As we can see from the first chart, deadlines prevented a lot of people from doing high quality work. This can be attributed to having to rush an asset out or not having enough time to properly test it before the deadline. For all the participants communication prevented high quality work at least a little. Technical difficulties also proved to be a significant obstacle for obtaining high quality of work, with over 9 participants answering that it affected their work. Personal chemistry of the team was not an issue for most of the development team, neither was bad specifications, latter contradicting data gained from the lead programmer interview.

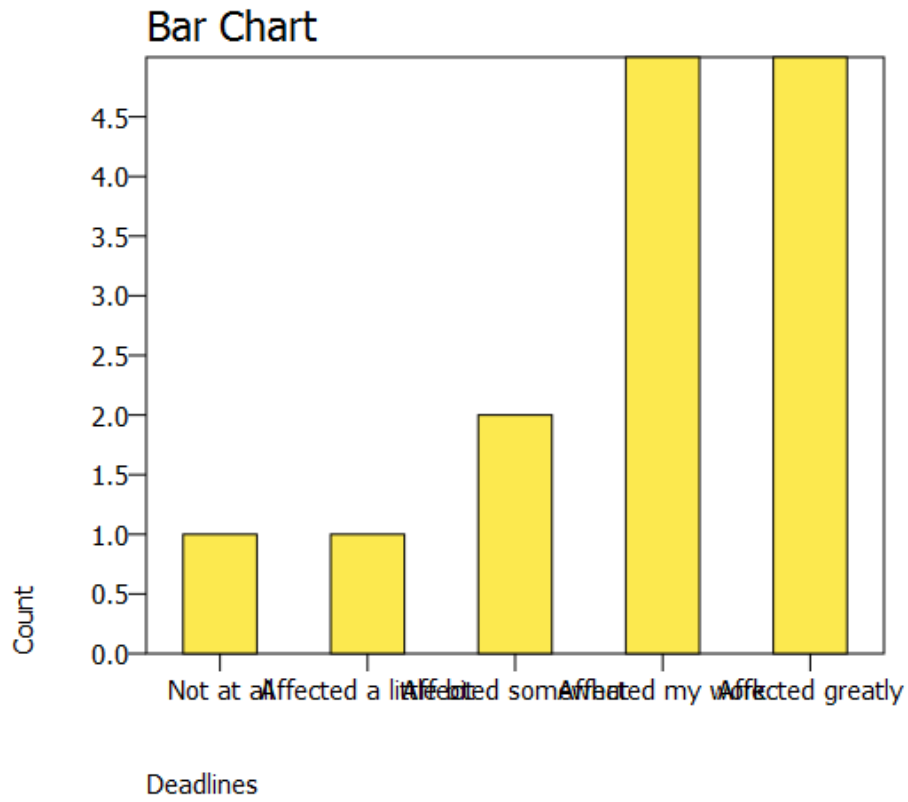


CHART 12. Deadlines

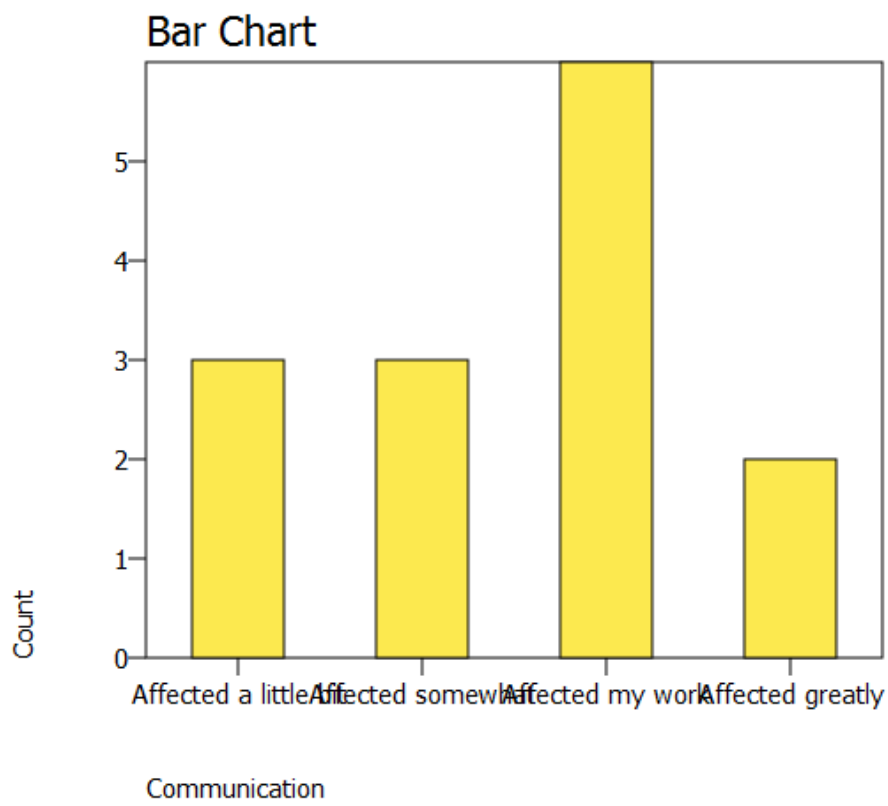


CHART 13. Communication

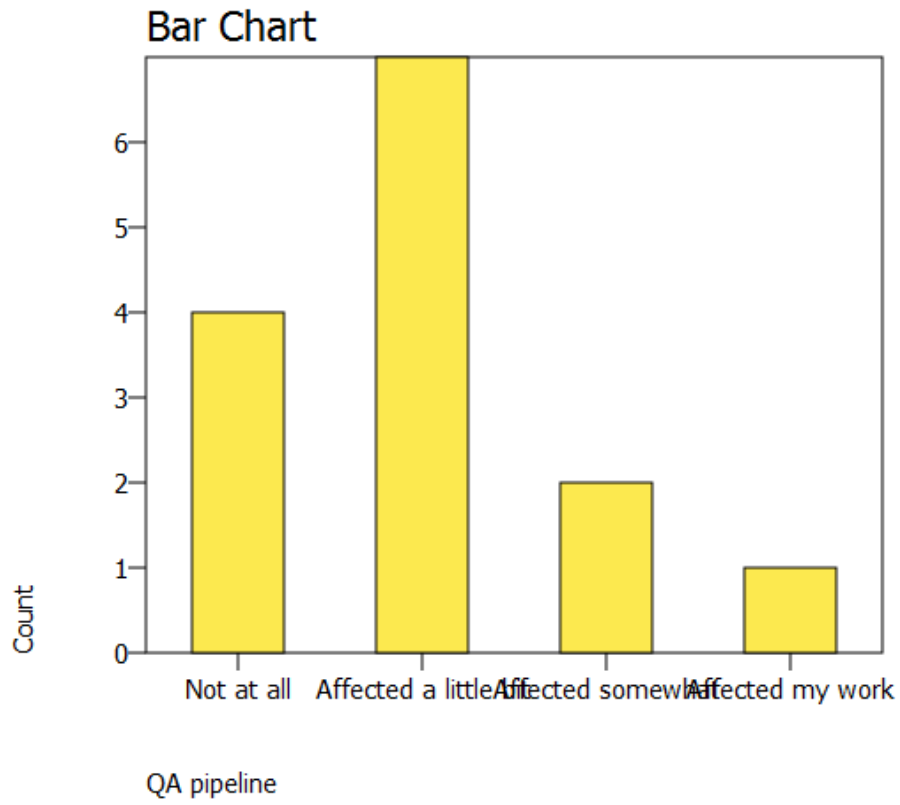


CHART 14. QA pipeline

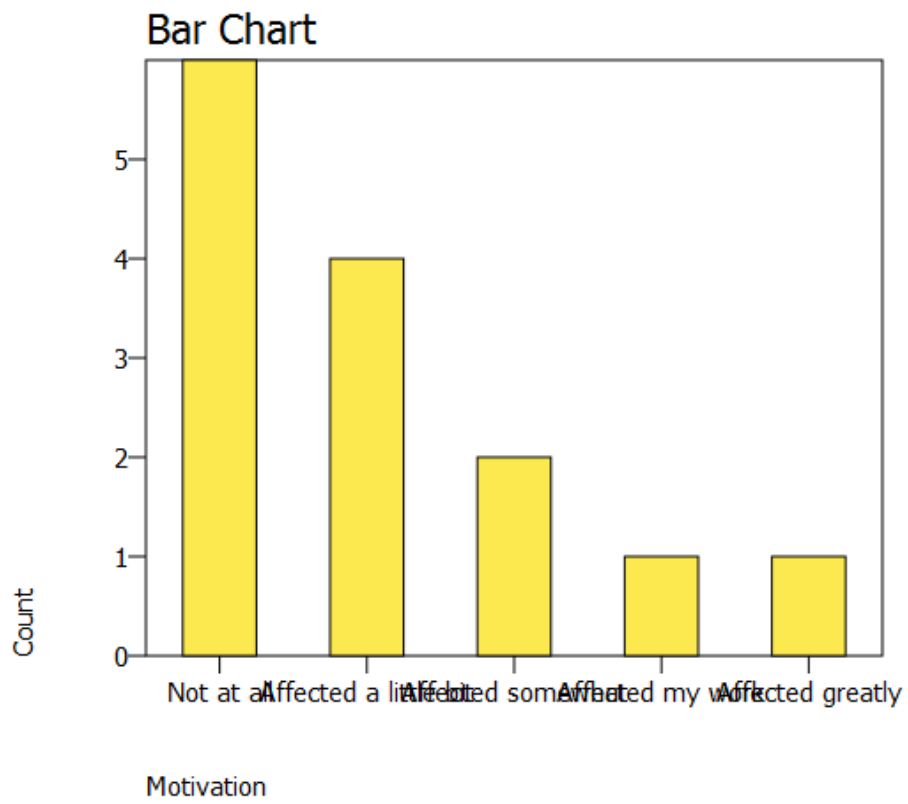


CHART 15. Motivation

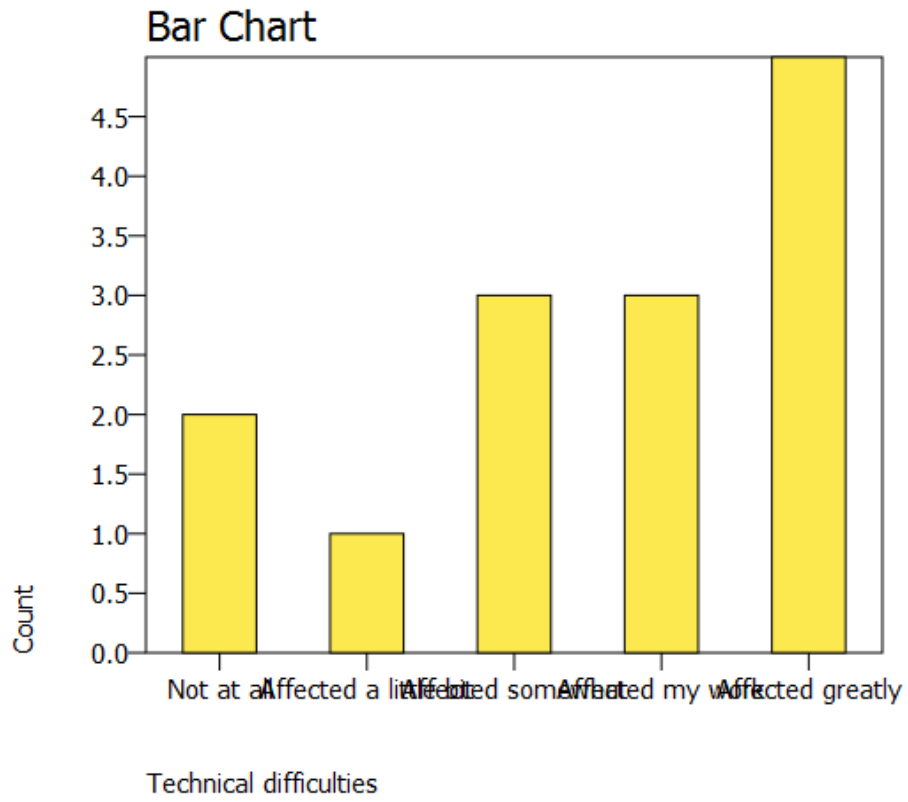


CHART 16. Technical difficulties

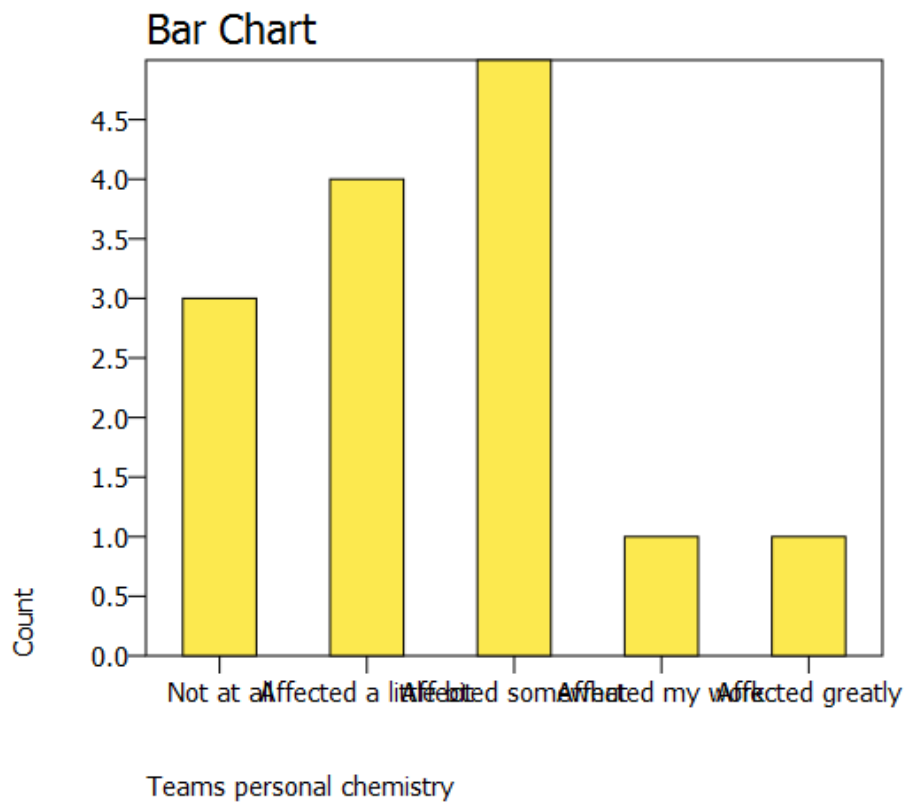


CHART 17. Teams personal chemistry

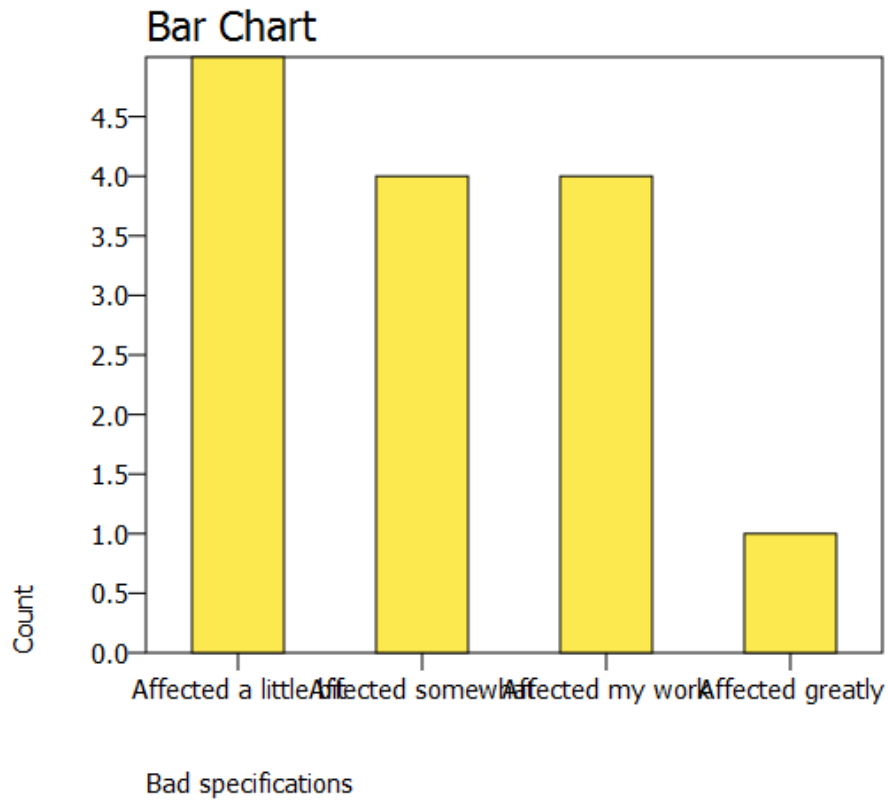


CHART 18. Bad specification

3.3.2 Interview analysis

The interviews were conducted during the span of two weeks. After I that I transcribed the recordings and analyzed them, trying to find clues for few key questions of the study. These questions were: Was the team too inexperienced? What were the major issues of the development? And also, more info about the different issues regarding to specific departments.

Where the survey gave me and insight into the grassroots feelings of the team, the lead interviews gave me an insight into the management side of things. There was a common thread among all the lead interviews which was the fact that they all seemed to have noticed the problems with the development in advance, but for some reason nothing was done to them. The only exception for this was the art lead interview because the art lead had joined the development in a later stage after the pre-production. (Art lead interview)

The interviews showed that the issue with development wasn't that the problems were not known, but more or less in the responses to those issues. Lead designer Jussi Autio said in the interview in response to a question about the biggest issue in UE's development that: "Overwhelmingly it has been the misallocation of resources." adding that he had already tried to talk about the issue 6 months before any changes happened.

Lead programmer Matti Pulkkinen attributed the biggest issue during development to "Lazy specifications" and that the team "Started to implement stuff before its design had been finished." Another thing that Matti highlighted was problems with Unity, confirming that using Unity lead to a large overhead in coding.

Project lead Juho Lyytikäinen also highlighted the resource allocation and scope of the game saying: "[Biggest problem in the development of urban empire] has been understanding the games scope in regards to available resources" also saying that the "QA was hired a bit too late [into the project]".

3.3.3 Observations

I observed the behaviors of the whole team during my entire time with the project. Most of the observations happened during daily meetings where, at first, the whole team met and everyone answered three questions: "What they worked on yesterday? What they are going to work on today? And is something preventing you from doing your work?" During the Summer of 2016 the team transitioned away from these huge 10+ member dailies to department specific dailies, a move that was greatly lauded by the teams and the leads. (Survey / Lead interview)

Mostly the issues that rose during dailies were problems with communication. Usual issue was for example a member of another team not understanding specifications to a task. Also sometimes people just forgot about a particular task and needed to be reminded. Dailies seemed to help communication a lot after the change was made from whole team daily meetings to a department specific dailies. We managed to get a lot deeper into the issues and thus the communication got better. We could also bring up problems that we felt that the leads should answer immediately, knowing that the leads would have their own meeting later in the day discussing the project.

4 ANTI-PATTERNS DURING THE DEVELOPMENT OF URBAN EMPIRE

4.1 Examination of anti-patterns during development of Urban Empire

From my research, I wrote an internal document for the entire company. In the document, I explained the anti-patterns that I had observed during my time in the development of UE and all the things I found out while analysing the surveys, workflows and interviews. I showed the results of the case study in a presentation for the leads & management of the company.

4.1.1 Escalation of commitment

Escalation of commitment is closely related to the sunk cost fallacy, a term used to describe a phenomenon where people justify increased investment of money, time or other resources when ending investment now would be better decision overall (Staw 1997, 191-215). This can be understood with a simple gambling example. Imagine a player who has already lost 400 euros in the blackjack table, it is hard for him to stop playing now and usually continues playing in hopes of “winning the money back”. This is psychological fallacy that leads to an increased loss of money.

This anti-pattern affected development in few different ways. The UE development team spent time and energy on assets that they knew were going to be changed dramatically. Especially the decision to focus vertical slice on the middle part of the game. This meant that instead on focusing how the player is introduced to the game mechanics at the start of the game, the team focused on badly specified assets and features and month to two months of work was lost. This lead to increasing stress inside the team and also increased overtime as the team tried to redo all things that were not actually ready.

4.1.2 Groupthink

Groupthink refers to a group of people, like a game's development team, starting to think "inside the box". This impairs their ability to solve issues or see issues with the product when no new ideas or "fresh eyes" look at it (Turner, M & Pratkanis, A 1998). One of the most well-known examples of that is the bankruptcy of Swissair. Swissair was once known as the "Flying bank" because of their financial stability. Over the years the management, which now believed the bank to be impermeable to financial trouble, started to do bad financial and business decision. Leading to the most reported airline decline of the 20th century and leading to the bankruptcy at the beginning of 21st century (Kilhefner 2016).

Groupthink is extremely common in the games industry. Usually this is because a game starts with a vision and to get every team member on board, they have to be excited about the vision and truly embrace it. This leads to "like-mindedness" where it gets progressively harder for the developers to critique their own game or look at it from fresh perspective. Although the UE team could criticize their work, when asked about it privately, the concept of groupthink could be seen from the actions of the team. They wanted to keep the status quo. Usually in regards to assets the team could give good constructive criticism and people responded to it, but in terms of workflows and practices change was slow. It took the team 6 months of discussions to change from the entire team daily meetings to department specific dailies, which helped the development largely.

One interesting thing that the surveys and interviews told was the fact that most of the issues with the development were well known with the leads, but for some reason the "Collective reasoning" didn't know them. This could be seen for example with resource misallocation, where the lead coder and lead designer knew that there were too few coders for the project but still no new resources were hired. Anti-pattern of Groupthink slowed the collective from making changes into their practices and workflows.

4.1.3 Dependency hell

Dependency hell refers to the problem caused by depending on a system that development team cannot fully influence (Jang 2006). This system can be anything from a game engine to a third-party application used during games development. For example, if a game engine has a bug that crashes every time you do a certain action in a game. This problem might be something that you cannot fix.

For the UE team this meant depending on the game engine Unity, which lead to enormous amounts of overhead in the development. In some cases, such as the fix for the button colliders not working properly in non-native resolutions, it took a year for Unity to find a fix for it. This forced the UE team to work on finding workarounds which greatly attributed to the overtime in the project.

Limiting effects of dependency hell is extremely hard. In some cases, counting on only proprietary software just isn't viable and developing your own game engine is time consuming and expensive. UE team spent some time on thinking about the different engines, but decided to go with Unity because it had worked in the past projects. The problems started to become apparent only in the project of this size. (Lead programmer interview)

4.1.4 Smoke and Mirrors

Smoke and Mirrors is a problem where a program or functionality that doesn't exist, but is shown like it is (Source-making.com 2016). It is very prevalent in the game industry with flashy trailers showing gameplay that isn't indicative of the actual finished product. Notable examples are Watch dogs or No Man's Sky, both of which garnered universal acclaim when first revealed, but which fell from grace soon after release when the public found out that the games were not as amazing as the trailers led the customer to believe.

UE team decided to show the middle of the game in the vertical slice. This meant that they had to emulate a lot of the core functionality of the game without focusing on stuff that they knew was important, such as the start of the game. This meant that screenshots taken from the vertical slice would look almost nothing like the game that is going to be delivered.

The issue with this was the lost development time while doing the vertical slice and the lost development time for having systems that were not fit for the final game. A lot of time was spent in the later development refactoring vertical slice code and a lot of bugs were introduced into the game because of that.

4.1.5 Ninety-ninety rule

Ninety-ninety rule refers to a fallacy where the development team estimates that there is only "10%" left to do while in reality the remaining tasks will take a longer time. This is prevalent problem with software development, when the development team underestimates the time it takes to polish features. (Bentley 1985, 896-901) This problem is usually attributed to an inexperienced team although it can happen to others as well. Mostly it's just an issue with the project management. Management should see these problems ahead of time and have a large enough buffer in the deadlines to make sure that no one has to work crazy overtimes to meet them.

During UE's development, there were a lot of too optimistic deadlines, which lead to failed deliveries and stressful overtimes to the whole team. Sometimes the problem was focusing on wrong things to work on and that lead to missed deadlines. (Project lead interview) Especially the time thought to be needed for testing was way too optimistic, in some cases leaving only half a day for testing a build that would need to have 10 hours of playable working content. This lead to the team missing deadlines or sending builds not thoroughly tested.

4.1.6 Resource misallocation

During the study, I noticed that there was still one systematic problem that was the major source of problems for the development. I researched many different sources, but found none which talked about this specifically. I call it the anti-pattern of resource misallocation.

This is an anti-pattern where the development team underestimates the amount of manpower, time or other resources needed to complete a functionality or a program. This can happen because of the inexperience of the team or because another anti-pattern adds more resource demand to the project.

This was extremely influential in making the project miss its deadlines and for other issues with the project. Both the lead designer and lead programmer confirmed during interviews that mishandling and misallocation of resources was one of the greatest issues with the development. There were too few coders for such a big project (lead designer interview) and too few designers writing specs (lead programmer interview). Having too few coders led to bad code and thus slowed down production in other areas. There should have been at least 4 coders right from the start in the project. When new coders did join the team their skills were underused, which lead to them not being able to debug and fix internal problems with the game. (Lead designer interview)

5 CONCLUSIONS

Urban Empire was close to being a well done game project. There were a lot of things done right and considering the scope of the game and the size of the development team it is a miracle that the development has gone as smoothly as it has. There is no doubt in my mind that the game will launch successfully on January 2017. I collected the findings of my study to a presentation which I presented to the management November 2016. I also created a document for the internal use for the Reborn titled "Anti-patterns during the development of Urban Empire and how to avoid them", it listed the same anti-patterns that I had observed in this thesis, but also gave concrete advices for the future. My reasoning behind not sharing those advices in this thesis was that I believe that every game development team is different. There are no absolute answers and all advice is highly subjective, thus I didn't think that sharing those suggestions in this study would have helped anyone. Instead I decided to highlight issues in a broader manner and tell the effects of anti-patterns with the hope that anyone who is reading this might be able to notice those in their own game project and react before it is too late.

There are few things I wish that I had changed in the study. I started the study with the idea of analyzing the quality assurance pipelines and practices of the team, hoping to find evidences that would explain the state of the project. The survey was conducted with the hopes of answering the research questions back then, but as the aim of the study changed more towards project management some of the questions asked became unimportant. I didn't have time to redo the whole survey and analysis so I tried to use the data I had as efficiently as possible. Another thing I know is missing from the study is the post-production, which I think might pose another set of anti-patterns. This is also mostly a timing issue as if the game had launched in its original launch date I would have had time to study post-production also, but because the release date was pushed 3 months later it was impossible for me to get the data on time into this thesis.

I believe that the study managed to answer its research questions. I think if more data would be needed out of any project a more longitudinal approach into the study itself would be beneficial. Gathering data at many points during development would yield more accurate results overall when analyzing the data.

REFERENCES

- Bentley, J. 1985. Programming pearls: Bumper-Sticker Computer Science.
- Internal Reborn Documents. 2014-2016.
- Lead Interviews. 2016.
- Jang, M. 2006. Linux annoyances for geeks.
- Kalypso Media. [website]. Luettu 14.9.2016. www.kalypsomedia.com.
- Kilhefner, J. Read 2016. Groupthink Examples in Business. Houston Chronicle.
- Koenig, A. 1995. Patterns and Anti-patterns. Journal of Object-Oriented Programming.
- Schwartz, B. 2005. The Sunk-Cost Fallacy. Los Angeles Times.
- Staw, B. 1997. The escalation of commitment: An update and appraisal. Organizational decision Making.
- Turner, M & Pratkanis, A. 1998. Twenty-five years of groupthink theory and research: Lesson from the evaluation of a theory. Organization behavior and Human Decision Processes.
- Sourcemaking.com. Read 23.09.2016. www.sourcemaking.com/antipatterns/smoke-and-mirrors

APPENDICES

APPENDIX 1. (1)

Thesis questionnaire

This questionnaire is a part of my thesis work, a case-study about Urban Empire's QA. In this form I will ask you to evaluate your experiences of quality assurance during the pre-production and production phase. The results will be anonymized and individual answers won't be discernible from the overall answers. Answering will take approximately 30 minutes. If you have any questions, please direct them to me at: juhana.mantymaa@fragmentproduction.fi

***Required**

1. Age *

Mark only one oval.

- Under 18
- 18-24
- 25-31
- 32-38
- 39-45
- 46 or Older

2. Position during Urban Empire's development * Mark only one oval.

- Managerial
- Lead
- Senior position
- Mid
- Junior position
- Trainee

Other:

APPENDIX 1. (2)

3. Time worked in the company * Mark only one oval.

- Under 1 year
 1-2 years
 2-3 years
 Over 3 years

4 Projects I've worked on while working in Fragment/Reborn (In whatever capacity) * Tick all that apply.

- Rescue: Everyday heroes
 Rescue 2
 Rescue: Heroes in Action
 Rescue Heroes in Action: Rising Storm
 ADAC Die Simulation (Roadside Assistance Simulator)
 Urban Empire
 Transit King

Other:

5. Number of game projects I've worked on prior to working in Fragment/Reborn * Mark only one oval.

- None
 1-2
 3-5
 6-8
 9-10
 11+

6. I've worked with Urban Empire during *

Check all that apply Tick all that apply.

- Prototyping/Pre-production
 Production

APPENDIX 1. (3)

General Questions regarding quality assurance in Urban Empire

In this part, please mark along the scale how much you agree/disagree with the claims.

7. Quality assurance is an important part of game development * Mark only one oval.

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

8. I can affect the quality of the final product (game) *
Mark only one oval.

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

- 9 I know our quality assurance pipeline well * Mark only one oval.

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

10. Our current quality assurance pipeline helps me deliver a high quality product * Mark only one oval.

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

11. I use our quality assurance pipeline with most of my work *
Mark only one oval.

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

12. I have ownership for the work that I have created *

Ownership = feeling that you are responsible of your work and can make decisions regarding it. Mark only one oval.

Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree
-------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	----------------

APPENDIX 1. (4)

13. I have motivation to deliver high quality assets/features *

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

14. The level of quality that is expected of me is high * Mark only one oval.

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

15. I have time to polish assets/features before adding them into the game * Mark only one oval.

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

16 It's my job to make sure my work is of high quality * Mark only one oval.

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

17. Using my work time for quality assurance is important * Mark only one oval.

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

Pre-production

This section of the form will be about prototyping/pre-production phase of the Urban Empire development.

18. I worked on Urban Empire during pre-production * Mark only one oval.

- Yes
- No Skip to question 41.

APPENDIX 1. (5)

Pre-production

In this section you will evaluate your experience with QA during pre-production phase

19. My team during pre-production * Mark only one oval.

- Art
- Code
- Design
- Manager
- Other:

During pre-production...

Please mark along the scale how much you agree/disagree with the claims.

20. Communication worked great * Mark only one oval.

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

21 I had ownership of my own work *

Ownership = feeling that you are responsible of your work and can make decisions regarding it. Mark only one oval.

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

22. We had a clear quality assurance pipeline *
Mark only one oval.

APPENDIX 1. (7)

Reasons preventing me from doing high quality work during preproduction were...

1 = not at all, meaning that the reason stated didn't affect your work

5 = affected greatly, meaning that the reason stated affected your work greatly

29. Communi-
cation *

Mark only
one oval.

	1	2	3	4	5	
Not at all	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Affected greatly

30. Deadlines

* Mark
only one
oval.

	1	2	3	4	5	
Not at all	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Affected greatly

31. QA pipe-
line *

Mark only
one oval.

	1	2	3	4	5	
Not at all	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Affected greatly

32. Motivation

* Mark
only one
oval.

	1	2	3	4	5	
Not at all	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Affected greatly

APPENDIX 1. (8)

33. Technical difficulties

* Mark only one oval.

	1	2	3	4	5	
Not at all	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Affected greatly

34 Teams personal chemistry *

Mark only one oval.

	1	2	3	4	5	
Not at all	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>			Affected greatly

35. Bad specifications * Mark only one oval.

	1	2	3	4	5	
Not at all	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Affected greatly

36. Other reasons that prevented me from doing high quality work during pre-production...

.....

.....

.....

.....

37. Did you do overtime during pre-production? *

Mark only one oval.

Yes Skip to question 38.

No Skip to question 41.

Can't remember Skip to question 41.

APPENDIX 1. (9)

Pre-production

38. How many hours did you do overtime during pre-production? *

Mark only one oval.

- 1-5
 6-10
 11-15
 16-20
 21-25
 26-30
 31-35
 36+

39 I did overtime during pre-production because... *

Check all that apply

Tick all that apply.

- I needed to meet a deadline
 I had technical difficulties
 I needed to do revisions to my work
 I needed to create specifications for others
 I didn't have time to plan my hours
 My work pipeline was ineffective
 My work came back from quality assurance
 My work didn't meet specifications

Other:

40. I feel that the quality of the work I did during overtime was of...

* Mark only one oval.

- High quality
- Good quality
- Okay quality
- Bad quality

APPENDIX 1. (10)

Production

This section will ask about the production phase of Urban Empire.

41. I worked on Urban Empire during production *

Mark only one oval.

- Yes
- No Skip to question 64.

Production

In this section you will evaluate your experience with QA during production phase

42. My team during production *

Mark only one oval.

- Art
- Code
- Design
- Manager
- Other:

During production...

Please mark along the scale how much you agree/disagree with the claims.

43 Communication worked great *

Mark only one oval.

1 2 3 4 5

 Strongly disagree

 Strongly agree

APPENDIX 1. (11)

44. I had ownership of my own work *

Ownership = feeling that you are responsible of your work and can make decisions regarding it. Mark only one oval.

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

45. We had a clear quality assurance pipeline * Mark only one oval.

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

46. Our quality assurance pipeline helped me in my work * Mark only one oval.

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

47. I followed our quality assurance pipeline * Mark only one oval.

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

48. I managed to keep the quality of the product at high level * Mark only one oval.

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

49. My work wasn't hindered by technical difficulties * Mark only one oval.

54. QA pipeline *

Mark only one oval.

	1	2	3	4	5	
Not at all	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Affected greatly

APPENDIX 1. (13)

55. Motivation *

Mark only one oval.

	1	2	3	4	5	
Not at all	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Affected greatly

56. Technical difficulties *

Mark only one oval.

	1	2	3	4	5	
Not at all	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Affected greatly

57. Teams personal chemistry * Mark only one oval.

	1	2	3	4	5	
Not at all	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Affected greatly

58. Bad specifications * Mark only one oval.

	1	2	3	4	5	
Not at all	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Affected greatly

59. Other reasons that prevented me from doing high quality work during production...

.....

APPENDIX 1. (14)

60. Did you do overtime during production? * Mark only one oval.

- Yes Skip to question 61.
- No Skip to question 64.
- Can't remember Skip to question 64.

Production

61 How many hours did you do overtime during production? *

Mark only one oval.

- 1-10
- 11-20
- 21-30
- 31-40
- 41-50
- 51-60
- 61-70
- 71+

APPENDIX 1. (15)

62. I did overtime during production because... *

Check all that apply

Tick all that apply.

- I needed to meet a deadline
- I had technical difficulties
- I needed to do revisions to my work
- I needed to create specifications for others
- I didn't have time to plan my hours
- My work pipeline was ineffective
- My work came back from quality assurance
- My work didn't meet specifications

Other:

63. I feel that the quality of the work I did during overtime was... * Mark only one oval.

- High quality
- Good quality
- Okay quality
- Bad quality

Urban Empire QA pipeline

This section contain some question about urban empire QA pipeline and your familiarity with it.

64 After I complete an asset I... *

Asset = any piece of work associated with the game. This includes all the scripts, 3D models, 2D art etc.

Mark only one oval.

- Send it to QA for approval
- Send it to my department lead for approval
- Add it to the project
- Test/evaluate it myself
-
- Other:

APPENDIX 1. (16)

65. If a task I'm supposed to do doesn't have good enough specifications... * Mark only one oval.

- I ask for better specifications
- I do the task the best I can
- I wait for further instructions
-
- Other:

66. If I find a bug in the game I must... * Mark only one oval.

- Report it immediately in JIRA
- Forward the problem to QA
- Ask about the problem in slack
- Tell about it to my team lead
-
- Other:

67. If I find a low quality asset in the game I must... *

Asset = any piece of work associated with the game. This includes all the scripts, 3D models, 2D art etc.

Mark only one oval.

- Report it immediately in JIRA
- Forward the problem to QA
- Ask about the problem in slack
- Tell about it to my team lead
-
- Other:

68. After I have fixed problems with my asset I must... *

Mark only one oval.

- Close the task in JIRA
 - Assign the task back to QA for re-evaluation
 - Assign the task back to my team lead
 - Inform it in slack
 -
- Other:

APPENDIX 1. (17)

Thank you

Thank you for answering all the questions! By pressing the submit button below you give me a permission to use your answers in my thesis work with working title: "Pitfalls of quality assurance in AAsized games, case study: Urban Empire" . Your answers will be analyzed, anonymized and the results will be published in my thesis in December. Your answers will help me make recommendations to Reborn about the future of QA for this company and hopefully save time and effort in the future.