Oskari Aalto

MOBILE GAME PRODUCT DEVELOPMENT MODELS

Thesis
Kajaani University of Applied Sciences
School of Natural Sciences
Business Information Technology
15.11.2015

KAJAANIN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

THESIS
ABSTRACT

School	Degree Programme
School of Natural Sciences	Business Information Technology
Author(s) Oskari Aalto	
Title Mobile Game Product Development Models	
Optional Professional Studies Game Production and Business	Supervisor(s) Veli-Pekka Piirainen
Date 15.11.2015	Total Number of Pages and Appendices 51

The goal of every game development company is to continuously create successful game products, but this is not an easy task in the volatile game industry. Especially in the mobile game market competition has become fiercer and success rates are declining. To be able to survive, companies have had to find ways to develop new game products more rapidly and efficiently.

This thesis studies game development processes and product development models used in the game industry to develop successful games. The goal of the thesis is to identify a product development model that enables a mobile game company to efficiently build successful games and manage its product portfolio. The model should also help the company to evolve their success criteria and monitor progression of projects during development. Combining these elements should allow the company to make well-informed product decisions and manage their success rate.

The first half of the thesis focuses on the game development process and project stages used to manage game projects. The purpose is to identify which development process, waterfall or agile, is more suitable for managing mobile game development and to describe how a game project is structured with stages. The second half covers four product development models employed to create successful game products and provides a concrete example of how stage gate model has been adopted by the mobile game company Wooga.

As a conclusion for the thesis it is discovered that the combination of agile game development practices and stage gate model enables a mobile game company to build successful games faster and more efficiently. With better development process and product development model, the company can expect to able to experiment with more products and have a good chance at succeeding with the games they choose to launch to the market.

Language of Thesis	English
Keywords	agile, game production, mobile game, product development
Deposited at	☒ Electronic library Theseus ☐ Library of Kajaani University of Applied Sciences

KAJAANIN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ
TIIVISTELMÄ

Koulutusala Luonnontieteiden ala	Koulutusohjelma Tietojenkäsittelyn koulutusohjelma
Tekijä(t) Oskari Aalto	
Työn nimi Mobiilipelien tuotekehitysmallit	
Vaihtoehtoiset ammattiopinnot Pelituotanto- ja liiketoiminta	Ohjaaja(t) Veli-Pekka Piirainen
Aika 15.11.2015	Sivumäärä ja liitteet 51

Jokaisen peliyrityksen tavoite on jatkuvasti luoda menestyksekkäitä pelituotteita, mutta se on hyvin haastavaa peliteollisuuden epävakauden takia. Erityisesti mobiilipelimarkkinoilla kilpailu on lisääntynyt merkittävästi ja uusien pelijulkaisujen menestymismahdollisuudet ovat olleet laskussa. Peliyritysten onkin täytynyt löytää tapoja kehittää uusia pelituotteita nopeammin ja tehokkaammin.

Tämä opinnäytetyö käsittelee pelin kehitysprosesseja ja tuotekehitysmalleja, joita käytetään peliteollisuudessa uusien pelien luomiseen. Opinnäytetyön tavoite on tunnistaa tuotekehitysmalli, joka mahdollistaa mobiilipeliyrityksen kehittää menestyksekkäitä pelituotteita ja hallita sen tuoteportfoliota. Tuotekehitysmallin on myös tarkoitus auttaa yritystä parantamaan sen onnistumiskriteerien määritelmiä ja kehityksessä olevien peliprojektien hallintaa. Näiden pohjalta yrityksen pitäisi pystyä tekemään valveutuneita päätöksiä liittyen sen pelituotteiden hallintaan ja siten parantaa tuotteiden onnistumisprosenttia.

Opinnäytetyön kaksi ensimmäistä lukua keskittyvät kuvaamaan pelin kehitysprosessia ja vaiheita, joihin peliprojektit jakaantuvat. Tarkoituksena on tunnistaa, kumpi projektinhallintamenetelmä, vesiputousmalli vai ketterä kehitys, sopii paremmin mobiilipelin kehityksen projektinhallintaan. Lisäksi käsitellään eri vaiheet, joilla peliprojekteja jäsennellään. Kolmas luku käsittelee neljä tuotekehitysmallia, joita käytetään menestyksekkäiden pelituotteiden luomiseen ja viimeinen luku tarjoaa käytännön esimerkin siitä, miten mobiilipeliyritys Wooga on ottanut vaihe-portti-mallin käyttöönsä.

Opinnäytetyön keskeisenä havaintona todetaan, että ketterän kehitysmenetelmän ja vaihe-portti-mallin yhdistelmä mahdollistaa mobiilipeliyritysten kehittää menestyksekkäitä pelituotteita nopeammin ja tehokkaammin. Parantamalla pelin kehitysprosessiaan ja tuotehitysmalliaan peliyritykset pystyvät kehittämään useampia tuotteita ja voivat odottaa useamman julkaistun pelin menestyvän markkinoilla.

Kieli	Englanti
Asiasanat	ketterä, mobiilipeli, pelituotanto, tuotekehitys
Säilytyspaikka	☒ Verkkokirjasto Theseus ☐ Kajaanin ammattikorkeakoulun kirjasto

CONTENTS

SYMBOL LIST

KPI	Key performance indicator is a business metric that is used to evaluate factors crucial to the success of a company or a product.
Feature	Feature is a part of game product. It is usually an important game mechanic or functionality that adds value to the player.
Hit game	Term used to describe a game product that meets company's success criteria.
Scrum	Scrum is an agile project management framework that is commonly used in agile software- or game development.
Success criteria	Success criteria are the requirements used to evaluate whether a product has the potential to succeed or not.
Iteration	In agile software development iteration refers to a single development cycle. One iteration is usually one to four weeks in length.
Shippable	Shippable is a term used to describe the state of a game product that is in suitable condition for releasing to the market.

1 INTRODUCTION

The goal of every game development company is to continuously create successful game products, but it is a major challenge due to the game industry being very volatile. Hit games seem to appear out of nowhere, and at the same time success rates of companies are declining. The main reason for this is increased competition. Powerful game development tools are now available for everyone and digital distribution has made it possible to easily publish games. This has attracted more companies to game development, which has lead to more games being released than ever before. (Cook 2015; Kain 2015.)

Mobile game development has captivated many companies since the introduction of Apple's iTunes App Store in July 2008 and Google's Android market in October 2008, which was later branded to Google Play in 2012. These are the two biggest marketplaces for mobile games, combined they have over 3 million applications available for customers to download. According to Pocketgamer's statistics there were 44,678 games launched in Apple's iTunes App Store during its first month of operation. The number of launched games has been increasing, in October 2015 this went up to 440,510 games. These numbers display how the competition keeps increasing, the market being saturated with products and succeeding has become more difficult. (Sims 2015; Pocketgamer 2015; Statista 2015.)

This change in the mobile game market has forced companies to find ways to develop new game products more rapidly and efficiently. It is unlikely that every game product can become a success, so it is crucial to find ways to reduce development costs and to identify products with success potential as early as possible. Being able to cancel projects that do not meet a company's success criteria, allows them to utilize their resources more efficiently. Resources limit how many times a company can try to create a successful product, which is why mobile game developers like Supercell have openly discussed multiple canceled game projects, and Wooga implementing their hit-filter process. (Supercell n.d; Telfer 2014.)

Two of the most successful companies in the mobile game market, King and Supercell, have developed several extremely successful games, but those are not the only games the companies have developed. Both companies have rigorous evaluation processes to choose the best games to release to the global market. By rapidly producing new products and having a process to identify the games that meet the companies' success criteria, they have been able to keep creating new successful games. (Borison 2014; Street 2013; Supercell n.d..)

The purpose of this thesis is to research game development processes and product development models utilized in developing games in the mobile game market, and to identify their strengths and weaknesses. The goal is to describe a model that enables a company to, rapidly and efficiently, develop new game products and to evaluate each product's success potential. The model should also help the company to evolve their success criteria and monitor progression of projects during development. All this should allow the company to make well-informed product decisions and manage their success rate.

The thesis is divided into five supplementing chapters that describe different aspects of game product development. Chapters one and two explain game development processes and project stages used to manage game projects. Chapter three covers various product development models employed to create successful game products and to manage a company's game portfolio. Chapter four will look into how a mobile game company called Wooga manages their product development, and what processes and models they have in place to build successful games.

2 GAME DEVELOPMENT PROCESS

Game development process, also known as game development methodology, defines the way of organizing the work required to develop a game product. It determines what development stages are and how the work flows through them. To be able to understand game product development, it is first necessary to understand the game development processes and how individual projects carry out. Product development is intertwined with development process because the process dictates the state of the product at different stages of development. (Lotz 2013.)

Currently in the game industry there are two prominent development methodologies. The first one is waterfall, some times referred to as the traditional development methodology. It is a linear approach to game development that describes a sequential flow of development stages from concept to testing. In waterfall a functional product is only assembled at the end of the project. The other one is agile methodology, which is an iterative team based approach. It emphasizes continues improvement and evaluation of the product based on knowledge gained during the development process. (Bates 2004, 217 - 227)

2.1 Waterfall game development

Game development started with small teams, even a single person was able to create a successful game product. Small team size meant that having a development methodology was not utterly important. The development was mainly restricted by the capabilities of the video game hardware that was available. Over time the video game hardware evolved and became more capable of complex computing and processing. One person was not enough anymore to utilize the full power of the hardware. This led to bigger team sizes and budgets, which created a demand for a development methodology. To reduce the increasing risk, companies adopted waterfall-style methodologies copied from production line type of industries. (Keith 2010, 6.)

The waterfall methodology is a development approach where game projects are developed through series of stages. Each stage leads to a subsequent stage that is more complex and expensive than the previous. Figure 1 displays typical waterfall stages for a game project. The project starts from an idea from which a concept is created. Once the concept meets defined requirements for the idea stage, the project will move on to the next stage. The same pattern will repeat on each stage. (McGuire 2006.)

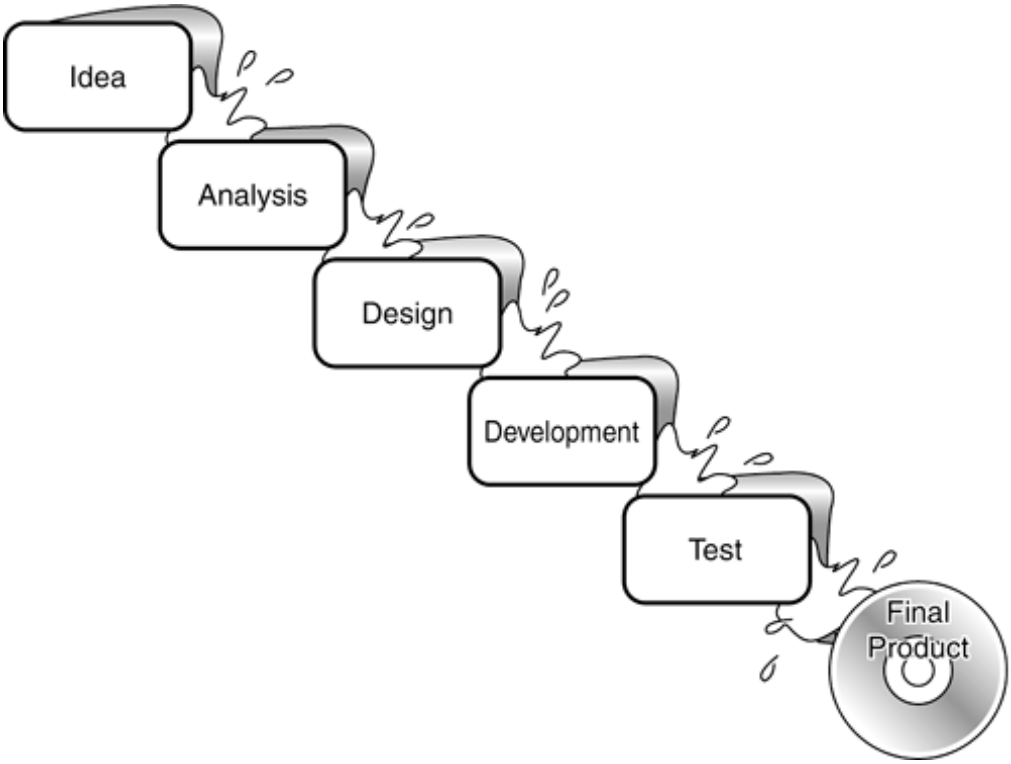

Figure 1. Waterfall game development. (@KaeTheDev 2014)

The intention of this type of approach is to reduce the risk before moving on to the next stage. This type of approach works if project requirements can be completely defined up front, for example series sequels. In these types of projects the basic gameplay mechanics are known, and other major subsystems, like the game engine and user-interface, can be reused from the previous project. Unfortunately in new product development this is rarely the case and, therefore, waterfall is not the best suited for that. (Keith 2010, 6 - 7; Bates 2004, 225.)

In waterfall development the product reaches functional for at the last stage of development and it is only then possible to evaluate it. When evaluation is left to the last stage of the project, it causes many issues. No matter how great the initial idea would have been or how thought–out the design was, it is possible that the final product does not work as well as the design suggested or the product does not satisfy the intended customer needs. Going through all the stages to be able to evaluate the product is not an efficient way to develop games. (Cook 2015, McGuire 2006.)

The main reason for this is that the project team often lacks sufficient knowledge of customer needs, especially in the early stages of development, which is usually gained throughout the development process and is critical to success. Therefore, the game development methodology used to create new products should encourage revisiting earlier stages and even adjusting the design during the process. (Cook 2015, McGuire 2006.)

However, waterfall methodology is not completely inflexible. It also includes some iterative elements where earlier stages are revisited. This is done by going back to design stage to redesign a feature if testing stage has shown a problem. The limitation of waterfall is that it is not feasible to iterate every feature or do major changes without offsetting the whole project and can causing a lot of work to be wasted. (Keith 2010, 7.)

Due to the issues mentioned, the waterfall methodology has lost its popularity and game companies have adapted new methodologies to develop games. Waterfall methodology can still suit certain types of projects and project teams, but in a fast-moving mobile game industry it is rarely used. Mobile game companies have moved to agile methodologies that emphasise fast and incremental improvement of the game product. This allows the teams to react to changing requirements, for example caused by changes in the market, and utilize the knowledge they gain during the development. (Bates 2004, 218 - 230; Telfer 2014.)

2.2 Agile development

Agile development is a project management approach where emphasis is on collaboration between people and continuous evolution of the project requirements and the product. The main point is that project team and stakeholders adjust the project planning based on reality. Agile development also focuses on delivering features iteratively. These features are prioritized based on the value they add to the customer. These methods enable reactive and predictable development. (Keith 2014.)

Agile methodologies started gaining a foothold in the eighties when large defence and IT projects were failing in increasing numbers. People started to look for better practices and writing about those. Some of these methodologies promoted developing products incrementally using iterations instead of using sequential staged approach. A single iteration contains all the stages of development and last from one to four weeks, which is a major difference to sequential waterfall where development stages are spread out over the whole project life cycle and each stage could take anywhere from a month to a year. (Keith 2010, 13.)

Early iterative and incremental methodologies were called lightweight methods but in 2001 a group of experts gathered in a summit and decided to name them agile methodologies. The summit also resulted in agile manifesto, which defines the values and principles of agile methodologies. (Keith 2010, 13.)

2.2.1 Agile Manifesto

The agile manifesto encourages actions that remove impediments in software projects. Each of the statements point to what project teams should value in comparison to what traditional project management methodologies teach. By valuing the items on the left side of the four agile statements, project teams can overcome some of the biggest issues that traditional project management methodologies have (figure 2). (Agile Methodology 2008.)

Figure 2. Agile Manifesto. (Agilemanifesto 2001)

Agile manifesto's principles enable all parties involved in the project to work with agility and embrace change. The principles can also be applied to game development even though its more creative nature compared to software development. The following paragraphs describe the principles from game development's perspective. (Keith 2010, 22-25.)

Individuals and interactions over processes and tools, in game development terms, means more collaboration between the widely different disciplines. An artist, a programmer and a designer should collaborate closely together instead of complex process and organization structure hindering their work. Agile encourages a bottom up approach where teams are enabled to solve problems on their own, which frees the leadership to focus on the big picture. (Keith 2010, 25-26.)

Working software over comprehensive documentation emphasises the importance of incrementally improving the game product with valuable features and keeping it in a testable state. A playable game has more value than any documentation that indicates what the game should do or how it should work. Games are interactive products that should be fun to play and the fun cannot be verified from a design document. Also designing every detail in advance causes unnecessary work, because it is based on assumptions and not knowledge that is gained during the development process. (Keith 2010, 27.)

Customer collaboration over contract negotiation points out that contract is an impediment to change. The typical game development contract has multiple defined milestones that are created before the project is started. Since payments are often tied to milestones, game developers avoid any changes that might make them miss a milestone. The publisher side cannot either make changes to the game product due to the binding milestone definitions in the contract. In both of these situations the contract does not allow improving the product and interferes collaboration. In agile environment payments are tied to time and materials. The publisher pays the developer for the cost of the last iteration. At the end of each iteration the developer displays a functional and improved game that enables both parties to evaluate the current state of the product and plan the next steps. (Keith 2010, 27-28.)

Responding to change over following a plan highlights the fact that focusing on what is known about the game product is more valuable than following a detailed plan. Projects rarely stick to a predetermined schedule, and detailed planning cannot solve all the unknowns that come up during a project. The project team is more likely to produce a successful product if they are able adjust the project plan based on the knowledge they gain throughout the process. (Keith 2010, 28.)

2.2.2 Agile project

An agile project consists of a series of iterations, which are short periods of time generally around two to four weeks. During each iteration the project team implements features that add value to the customer, enabling the game product to make incremental progress every iteration. A single iteration is a miniature version of the whole game project, containing all aspects of game development. This is opposite to waterfall where each aspect of game development is separated to its own stage and spread out over the project life cycle. (Keith 2010, 28)

The aspects contained into a single iteration are: concept, design, development, asset creation, testing, optimization, and polishing. Depending on the size of the project, the project team is usually divided into multiple smaller interdisciplinary teams. Each team focuses on creating a single feature of the game during an iteration that can be demonstrated and evaluated at the end the iteration. Interdisciplinary teams are required so that there is input from all disciplines relevant to the feature and collaboration is effortless. (McGuire 2006.)

Iteration contains all aspects of game development the game product is always close to a shippable state. Shippable means that it is in a suitable condition for releasing. The product may not be ready for release to the customers but the project team and any stakeholders can review a functional game. The results of the review influence the planning of future iterations. Bigger features, like online gameplay, may require several iterations to be fully functional. These are usually divided into smaller components so that the project team can evaluate the progress over each iteration before getting the feature into releasable state. (Keith 2010, 29.)

Evaluating the state of the project is part of the "inspect and adapt" principle that is the cornerstone of agile practises. The project team inspects the progress made after every iteration with customers and any other stakeholders. Based on the feedback the team can then adapt their plan to reach their goal (figure 3). The project team also evaluates their development process each iteration and changes it if necessary, to be able to work more efficiently. (Keith 2010, 29.)

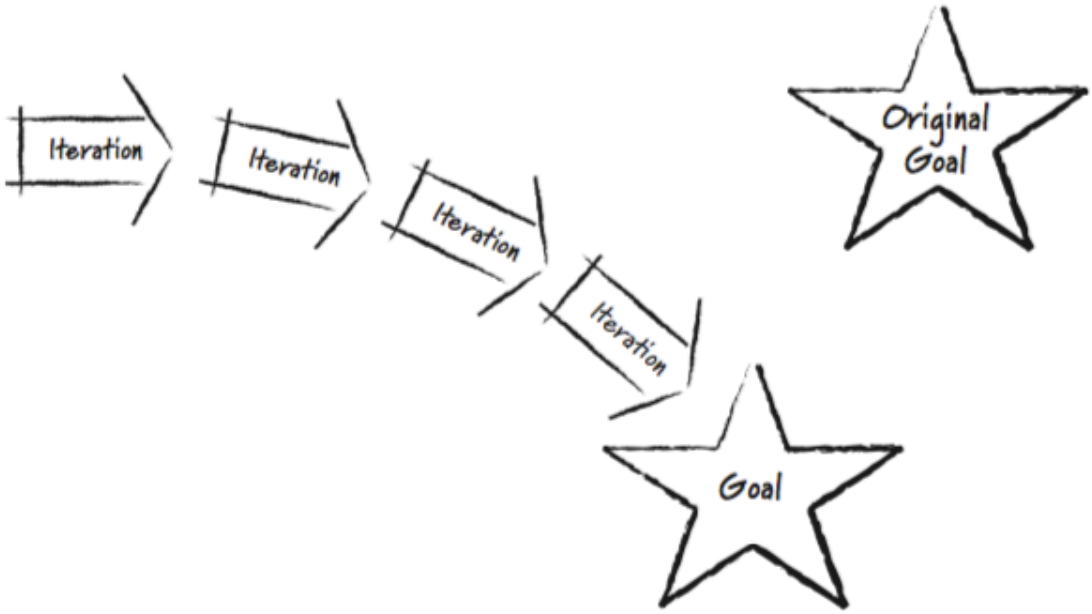

Figure 3. Iterations toward a goal. (Keith 2010, 30)

2.2.3 Agile game development

Agile game development combines the best practices of several other agile frameworks into one single agile approach. The values stated in agile manifesto have made it possible for agile frameworks such as Scrum, Lean, Extreme programming and many others to share common philosophy and principles. This also enables agile game development to choose the practices from other frameworks that are suitable for game development. Although agile game development may not follow all practices of any single framework, it does adhere to the agile mindset and values. (Keith 2014.)

The mobile game industry has been quick in adopting agile methodologies. This is due to the young age of the industry. Mobile gaming started gaining mainstream attraction and became a lucrative market around 2007, at the same time when first smartphones were released. This new platform provided a fresh start for game developers and companies, but also brought a lot of uncertainty with it. Agile methodologies have proven to manage that uncertainty well. Smartphones provided also a completely new gaming platform, which required

teams to innovate on the technology and gameplay side. Game companies had to figure out what kind of products they want to build and how they will do that. Experimenting is usually risky and better done with a smaller project team rather than a large one. This reduced the need for complex management structures and agile approach turned out to be a great fit. (Wright 2009.)

Since the release of the first smartphones, mobile devices and game development tools have continued to evolve rapidly along the side with the market becoming more mature and saturated. This has caused added pressure to develop and deliver games quickly. At the same time also business models have changed, most recent change being the transition from paid-model to free-to-play. Paid-model requires customers to buy the game before they can download it. In free-to-play model customers can download the game for free and the revenue is generated from small payments inside the game or by displaying advertisements. Utilizing agile methodology can be a major benefit for any company operating in mobile game industry where changes still happen constantly and the competition is ever fiercer. (Dryer 2013.)

Agile game development is a suitable methodology for any game project that contains a lot of uncertainty in the beginning, which is handled by embracing change and adjusting the project every iteration. In addition to agile game development embracing change, it puts emphasis on the customer being an important part of development process. The customer should be involved in the project through the whole process and not only in the beginning or at the end of it. This allows a project team to receive feedback from the customer during the development and adjust the project accordingly. (Bates 2004, 218.)

The strong suits of agile game development are: knowledge, iterative development, and efficiency. Knowledge is a key aspect of any project. When a project is started the company has the least amount of knowledge about it. During the development process the team will continuously gain more knowledge about the project, which can then be utilized to adjust the project plan and requirements between iterations. With iterative development the game product is developed in small increments, adding features that provide the customer the most value in fastest and most economical way. By focusing on features that bring the most value to the customer and developing working software iteratively, the project team is elimi-

nating waste. Prioritizing the most risky and crucial features first the project teams can find out early in the project cycle if the product is heading the right direction and possible change plans without actually having to go through the whole production cycle. (Keith 2010, 21 - 24; Segue Technologies 2015.)

Agile development approach also enables companies to predict costs and schedule because of the fixed iterations. Each iteration has a fixed-schedule that can only fit a certain amount of work. This makes the cost of each iteration predictable. Also by observing the work done in previous iterations a project team can better estimate how much resources are required to develop any new features. Once the project team knows how fast they work and how much work is done, they can estimate the cost and length of the whole project. Of course this is adjusted every iteration because some work is completed and the project plan might be adjusted. (Segue Technologies 2015.)

3 GAME PROJECT STAGES

One of the challenges of agile development is its stage-less approach. A single short iteration contains multiple elements of various traditional project stages without separating them in discrete stages. The lack of stages might cause the project to seem like homogenous string of iterations, but this does not really fit game development. Game projects usually have work that needs to be accomplished by certain stage of the project, regardless of the development methodology. This debt of work comes from the minimum required feature set that a game should have before it can be released. For example a first-person shooter game is expected to have single-player gameplay, weapons and targets to shoot. (Keith 2010, 127.)

Agile methodologies do not provide definitions for project stages so for agile game development they are taken from traditional project management. The traditional stages are: initiation, planning, execution, monitoring and controlling, and closing. Game development companies have modified these traditional stages and introduced new ones, to suit the industry and the agile approach better. This chapter will provide explanation for the stages commonly used in mobile game development. (Pathak 2014.)

The stages are high-level time boxes with gates that the project must pass through. A gate is a project milestone where project requirements are reviewed. A project cannot move to the next stage if all the requirements for the previous stage are not met. Passing through a gate also requires the approval from stakeholders out side the project team, like the executives of a company or the sponsor of the project. The commonly used stages in game industry are:

- Concept
- Prototyping
- Preproduction
- Production
- Post-production
- Servicing (Bonin 2014; Bates 2004, 203.)

In addition to the stages listed above, mobile game industry uses two extra stages, called soft-launch and global launch. Neither of them is really a separate stage, but they are intertwined with post-production and servicing. Nevertheless, they are both important parts of mobile game production and will be described later in this chapter. (Telfer 2014.)

These stages have become relevant due digital distribution and self-publishing model that enable game companies to release their own games. Companies are not anymore tied to single date, when they have to submit their game to a publisher that produces the boxed products and distributes them to stores. Now game companies can release their game when it best suits them and keep improving their game with updates even after the initial release. (Tran 2014.)

Clear stages are valuable to the creative and business aspects of game development. The process of iterative refinement is not an excuse to tweak the game endlessly without ever reaching milestones or declaring a product finished. The goal is to build a successful game product and release it to the market, or conclude that the product does not meet success criteria and cancel it during development. (Adams 2010, 48.)

3.1 Concept

Concept stage can be considered to be the first actual project stage for game development, even though there is an argument to be made that initiation is the first stage. This is highly dependant on the circumstances and the company. In a situation where a client orders a project from a game company, initiation is a separate stage. If a game company decides to develop its own a product for the games market, initiation is usually part of the concept stage. In mobile game industry the latter scenario is more common and therefore initiation will not be separated in the context of this thesis. (Bonin 2014.)

Concept stage starts of by identifying opportunities, whether it is a gap in the current marketplace or prospect emerging. This can be achieved with market research and competitor analysis, but a company must also consider its strengths and weaknesses. The goal is to find a competitive advantage that can put the company in a favourable business position, and to define the target audience for the product. (Fischer 2014.)

Concept development is almost purely iterative. A project team creates as many ideas as possible. Some of the ideas will be ignored and others will be iterated on, in an effort to identify the best ideas. From the best ideas the project team creates concept documents that detail a game product's goal in the context of the marketplace and the customer. (Dryer 2013.)

Concept stage usually involves a small project team and it has a time constraint to make sure the project can move on at certain point. The purpose of the concept stage is to gain knowledge about the market and create two to five concept documents. The project team and stakeholders involved in the project can then decide, which of these concepts can move to the next stage of development, prototyping. (Keith 2010, 133 - 133.)

The decision-making process involves comparing the concepts against the competitive advantage analysis. Once the concepts that support the company's strategic goals are identified the project will move to the prototyping stage. On the prototyping stage the team will focus on proving that the game is fun to play by building a prototype. On the concept stage it is hard to say if the product would actually satisfy any customer needs because the product cannot be tested. (Bates 2004, 203 - 207)

3.2 Prototyping

According to Bates (2004, 211) a prototype: "is a working piece of software that captures on target platform the essence of what sets your game apart from the crowd, and will make it successful". Prototyping stage's purpose is to prove that a concept can actually be developed into a successful game and convince stakeholders that the game is worth investing in. (Bates 2004, 211.)

Prototyping is the experimental part of game development. In prototyping stage there are not usually many limits and it is about exploring what makes a game fun and enjoyable. This leads to the stage being both exciting and time consuming. To enable the experimenting the project team is usually kept small but cross-functional. (Bonin 2014; Bestebroer 2015.)

The project team creates prototypes from each of the concepts, focus being on the most valuable and high-risk features. The prototypes are then used to point the strengths and weaknesses of each concept. This should aim to identify and answer open questions related to each product concept. (Bonin 2014; Bestebroer 2015.)

The only limit to prototyping is usually pre-defined time frame. It is supposed to restrict the experimenting from diverting too far and keep the project team focused. Prototyping needs to generate results in the form of playable prototypes that the project team and stakeholders can use to evaluate the different game products. Project teams also user-test the prototypes to find out how potential customers react to the product. At the end of the prototyping stage, one game concept is approved to move on the next development stage. (Bestebroer 2015; Telfer 2014.)

3.3 Preproduction

Preproduction is often said to be the make or break point for a game project. If a project enters production without a proper preproduction, it will most likely lead into delays later on in the project and work being wasted. In this stage the team needs to fully define what kind of game they are creating and how it will be created. Preproduction really is the most crucial stage of any project. (Keith 2010, 16.)

During preproduction the team utilizes agile practices by iteratively working on various aspects of the game. This allows them to incrementally add value, and also gain knowledge about the product and the production process. For example some assets are developed to represent production quality to test how the production pipeline works. (Keith 2010, 131 - 134.)

Usually during preproduction the project team's size is increased from prototyping stage because the project is starting to ramp up. It is important to involve people from all disciplines of game development to be able to cover all aspects of development. Technical prototyping is required to discover any major technical challenges, potential solutions for those and the feasibility of using existing technology needs to be investigated. Another valuable resource in

preproduction is the art bible, used to define visual style of the game and create consistency between artists working on the project and give insight to the external disciplines. The prototype of the game must also be further developed to nail down game design and gameplay mechanics. (Bates 2004, 207 - 212.)

The project team must also cover the business and production side of the project. The team is required to detail the game's business model. It is one of the major features and defines how the game will generate revenue. Additionally the team needs to create a project plan, which defines the budget, production pipeline, and schedule. There are also risks involved in any projects that the project team must identify and prepare backup plans for. (Bates 2004, 207 - 212; Bethke 2003, 26.)

After preproduction all the major features should be defined and there should be a solid development plan. The project team will then prepare milestone definitions for coming stages that define deliverables for each stage. The deliverables are groups of features that are prioritized based on their value to the project. Highest priority is given to the most valuable features and those will be developed first. The prioritized list provides an overview of the project's scope and will work as a clear task list for the next stage. (Bates 2004, 207 - 212.)

3.4 Production

Production is the execution stage of game development. It is also the most expensive stage of game development because it requires the longest development time and biggest project team size. The goal of the stage is to develop the game product, including plenty of content for the players to consume, that the company intends to launch the market. (Keith 2010, 131.)

The focus is on creating the final product using the core mechanics and development processes discovered during pre-production. Production is about efficiency and incremental improvements. The project team iterates less on major features defined during preproduction, because it is also creating content based on those features. Making any major changes during production can be very costly and cause a lot of content to be reworked. (Keith 2010, 131.)

Production should be divided into multiple releases, because the long stretch of time it takes. The releases are smaller milestones with that include a subsection of the whole stages deliverables. With the incremental progress of the releases, the project team and any stakeholders can evaluate the progress of the project even during long production stage. (Keith 2010, 131.)

Even if the project was fully defined and planned in pre-production, the project team should keep questioning and assessing the product. Through the iterative releases, the project team keeps creating knowledge about the product and can adjust their plan accordingly. This type of approach follows the agile principles. The project team focuses on what is important for the product and incrementally adds value, it does not blindly follow the a predefined plan. The team should embrace changes and be prepared to adjust the plan, as agile methodology encourages. (Keith 2010, 131.)

Prioritizing the most important and valuable features first, allows the team to achieve two things during production. First of all it provides the team with a clear starting point and something to rally around. Secondly, getting the most important features done first lets the team know if the game should continue or be cancelled, which is critical for the project's success and for efficient management of a company's resources. (Dryer 2013.)

The end result of production is a minimum viable product (MVP), which according to York (2012) is: "the bare minimum game experience necessary to prove that people find your core game mechanic engaging". It is more beneficial to create the minimum viable product and quickly move on with the project. This enables the team to start testing the game as early as possible and collect valuable feedback internally and from the customers. The minimum viable product works as the basis for the decision, if the game should continue to the next development stage or be cancelled. (York 2012.)

3.5 Post-production

Post-production starts when the minimum viable product is done and majority of the content work is done. The team focuses on polishing the whole game experience. This stage also keeps improving the game incrementally but mainly by focusing on smaller details. Post-production also includes a lot of rigorous testing to find out any errors or issues in the game. (Keith 2010, 131.)

The goals of post-production are mainly defined by day-to-day basis, even though release reviews can still occur. The team aims to get the game into a shippable state. Usually post-production is restricted by upcoming key dates related to the game's launch to the customers. (Keith 2010, 134.)

Reaching the post-production stage means that the project team and other stakeholders have faith in the product. All parties involved agree that the product meets all the project requirements and they are convinced that the product can become successful when launched. But before committing to actual launching the game to all of the customers, game companies should verify the product on a test market by soft-launching it. (Keith 2010, 134; Telfer 2014.)

3.6 Soft-launch

Digital distribution has enabled mobile game companies to self-publish their games, and also quickly provide updates and new content to the customers. This has also made it possible for the companies to verify their game product's performance on a test market before investing in launch to the global market. Launching a game to a small section of a market that is usually restricted by geographic region is called soft-launch. (Mobiledevmemo 2013.)

The reasons for doing a soft-launch are numerous but most important ones are: ratings and quality, the cost of user acquisition, and understanding the customer behaviour. Ratings and quality have a significant impact on a game's success and they work hand-in-hand. A high quality game will get higher ratings from the customers. Ratings are important because they impact discoverability and perception. Customers are more likely to download a game with good ratings. Therefore a game needs to be tested by a notable number of users before it is

made available to the whole market. By limiting the launch of the game to a test market, a project team can receive feedback and address any issues without hurting the game's long-term discoverability in the markets that hold the most revenue potential. (Mobiledevmemo 2013.)

The second reason for doing a soft-launch is the cost of user acquisition. User-acquisition is a marketing term that refers to buying advertisement that aims to get people to download a game. By iteratively improving the game product during soft-launch, a project team can make sure that they will launch the best possible product to the whole market. This increases the likelihood of recouping development and marketing spend. (Mobiledevmemo 2013.)

Understanding the customer behaviour is the third reason for soft-launch. The game product has almost reached the state in which it will be launched to the market. But before a large number of users have played the game, it is difficult to forecast how the game will be received. Soft launch can point out interesting user behaviour that the project team did not predict during development and adjust the product accordingly. (Mobiledevmemo 2013.)

Soft-launch enables the project team to iterate on the game product based on user data they collect. Data collection is the act of tracking the users, who are playing the game and their behaviour. This has become an integral part of mobile game development and is done by most of the mobile game companies. The project team uses the data to measure and evaluate key performance indicators also known as KPIs. The metrics that are the most integral for a mobile game's success are those that indicate how well players understand the game's core gameplay mechanics, how engaging the game is, and how willing the players are to introduce the game to other people. (Mobiledevmemo 2013.)

The most important KPI is retention. It is very important to retain the users who are playing the game and turn them to dedicated players. Dedicated players are more likely to generate revenue and therefore the most valuable customers. Retention is highly influenced by how fun and interesting the players find the core gameplay. The core of the game cannot be changed this late into the project, subpar retention numbers during soft-launch often indicate that the game product will not be a major success. (Kriese 2014.)

If the project team sees that the KPIs meet the company standards and the game is good enough to become a successful, the project can continue. Often at the end of the soft-launch stage a project teams does small tweaks and prepares for the next stage, which is launch to the global market. (Wooga n.d..)

3.7 Global launch

Global launch is the stage when the game product is finally made available to the whole market. Getting to the launch stage requires tremendous effort from the project team and meeting all the milestone requirements of the previous stages. It is in no way an easy task and a lot of projects do not survive to this stage. Depending on the company's requirements for a project to pass the various gates on the different development stages, the amount of projects making from prototyping to launch can be less than 10 %. (Kriese 2014.)

Launch stage rarely involves any actual game development. Launch is usually major effort from the whole company involving marketing, public relations, executives and community management. The project team ensures that the product is in flawless shape and works with supporting teams to make sure the launch will be successful. In smaller companies, where there are no separate supporting teams, those task fall to the project team. (Wooga n.d..)

Launching a game is an expensive endeavour where lots of capital is invested into acquiring users, so it is important that the product is the best possible and the success potential is thoroughly evaluated during the development process. Once the launch has been done the project will enter servicing stage and there is no turning back. In the servicing stage the company will finally become a success or not. (Cook 2015; Shapina & Shpilber 2013.)

3.8 Servicing

Servicing is the final stage of game development. It starts when the game product has been launched to the market and will continue as long as the company feels it is necessary to keep improving the game and providing more content to the players. During the earlier development stages the project team has been focusing on the most important features, to be able to verify the minimum viable product on the soft-launch stage. Between soft-launch and launch there is rarely enough time to expand the product, since the team is occupied by tweaking the game based on the feedback from soft-launch and preparing for the launch stage. So the team is left with a backlog of ideas and features they would like to add to the game to make it a better experience.

The purpose of the servicing stage is to keep incrementally improving the game product and provide a service for the customers. Especially in the mobile game industry it is crucial to provide players with updates, new content and address potential issues. The goal is to keep customers happy and engaged with the product. (Sivak 2013.)

New content also allows the team to generate further revenue from the customer. It is easier to develop more content for a successful game, with an existing customer base, than starting a new game project from the beginning. Games that succeed in the servicing stage are some of the most profitable ones. Also providing a long-lasting service means a steady income flow for a company, which provides stability and enables taking risks with new projects. Developing games is an expensive process, and not every game will make back its development costs. (Chernyak 2014.)

3.9 Using releases to manage game project stages

Depending on the size of a game project, some project stages take from one month to several years. To be able to follow agile principles and keep incrementally improving the game, a project team should divide stages into releases. Release is a set of iterations and works as a milestone with defined deliverables, similar to a project stage. Figure 4 displays how the project is divided into multiple releases over its whole life cycle. (Keith 2010, 132.)

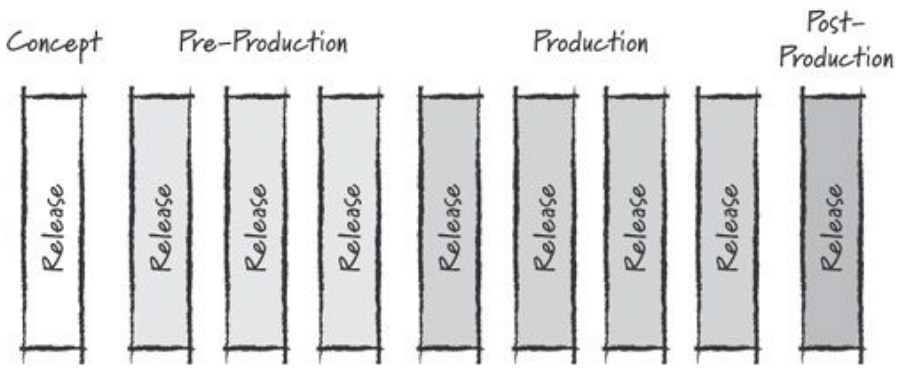

Figure 4. Project divided into multiple releases. (Keith 2014, 32)

Some features require multiple iterations to be completed and there is work that must be completed before the game project can move forward within a project stage. To manage the dependencies and bigger features during development, project teams use releases. At the end of each release the project team creates a vertical slice that demonstrates progress across all aspects of the project. This can be then used to evaluate the current state of the project. (Keith 2010, 31 - 32.)

4 PRODUCT DEVELOPMENT MODELS

Mobile game industry is a hit-driven business where every game product will not turn out to be successful, even if the development process and its stages were executed flawlessly. Successful games need to generate enough revenue to cover all those projects that were cancelled during various development stages or failed after launch. To be able to survive and excel in this environment companies need to have proper product development model. (Cook 2015; Shapina & Shpilber 2013.)

According to Rouse (2014) product development is: "process of designing, creating and marketing new products or services to benefit customers". Its purpose is to define and improve methods that a company uses to get a new product to market. Having a proper product development strategy is especially important for companies that want to develop successful products efficiently and react to changes in the marketplace. (Rouse 2014.)

Understanding how game development process works and knowing the stages each game project should go through is not enough to build successful products. A company needs a product development model to manage its products and portfolio. A product development model combines development process principles and project stages into a framework that enables a company to manage the costs and risks of creating new products. (Cook 2007.)

This chapter will describe four product development models: waterfall model, standard portfolio model, iterative model and stage-gate model. Waterfall model utilizes the waterfall development process, as the name suggest. Standard portfolio model combines multiple waterfall projects into one product development model. Iterative model applies agile practices to product development. The last model, stage gate, combines portfolio management with agile development practices and project stages into a single product development model. (Cook 2007.)

4.1 Waterfall product development model

Waterfall product development model and waterfall game development process go hand-in-hand, therefore, much of the same things apply as described in chapter 2.1 "Waterfall game development". The model is still commonly used, even though it has fallen out of fashion in the fast-paced industries where time-to-market and flexibility are important for succeeding. (Cagan 2005.)

Waterfall model is a very simple product development model. It starts by collecting the product requirements, which define what the end product should be. Based on the requirements the project team creates a project plan and schedule. Following the project plan the team then goes to develop the game through series of well-defined project stages. Each stage has its own specific set of work and deliverables that must be completed before the project can move to the next stage. This is because each stage is intended to add more complexity to the project. Only at the last stage the different parts of the product are assembled together to represent a complete product. (Cagan 2005.)

The strong suits of waterfall approach are its predictability and deliverables. The project team can create a relatively accurate schedule based on the initial requirements, assuming they completely understand the product requirements and the technology they are using. The deliverables of each stage enable the team to measure progress towards the end of the project and estimate the resources needed to complete each stage. (Cagan 2007.)

The downfall of waterfall model is that the project team's goal is to develop a product that matches the initial requirements. The team and stakeholders rely on being able to define requirements in the beginning of a project that would turn into a successful product. Only at the end of the project, they are able to verify if the requirements actually produced a good product. During the process they do not have any means of verifying if the product meets any customer needs. This also means that they cannot use knowledge about customer needs to adjust where the project is heading. Figure 5 displays how a project in waterfall product development progresses linearly from start to the target. (Cook 2007.)

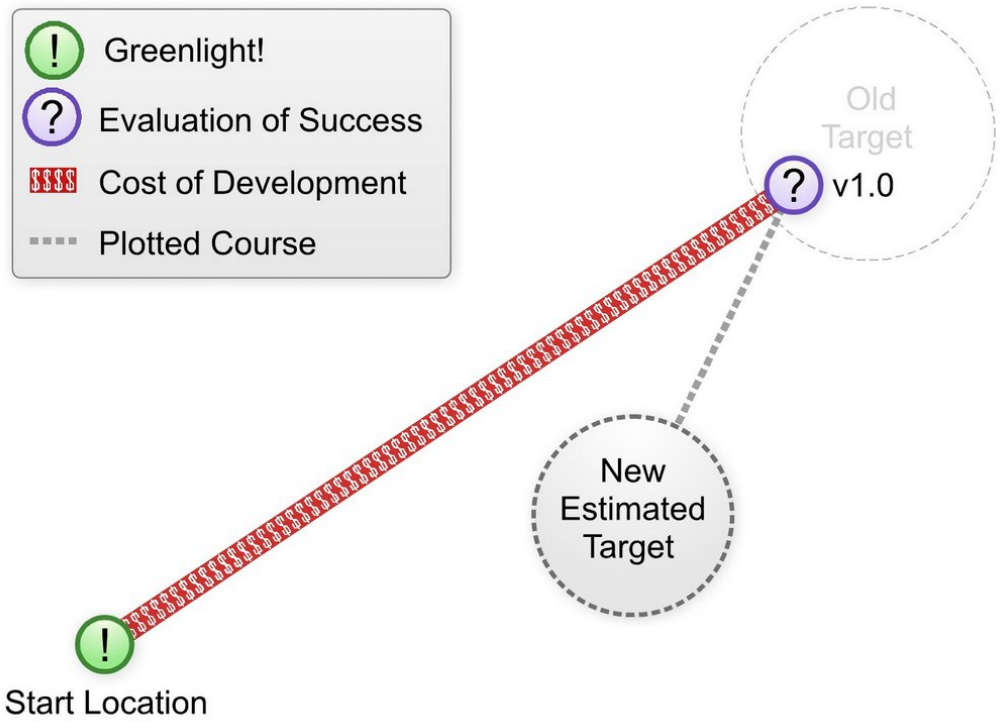

Figure 5. Product development progressed linearly in waterfall model. (Cook 2007)

Waterfall product development is a very costly model due the linear approach and the incapability of making adjustments during the process. Once a project is started it will consume company's resources until it reaches the target. Basically this means that when the company is finally able to evaluate the product for the first time, it has already spent most of the project budget. If the product does not end up meeting the intended customer needs, the company has wasted a lot of resources. (Cagan 2005.)

If the first product ends up failing it does not necessarily mean the end of the project. If the company has enough resources, they can decide to adjust the initial requirement and start the waterfall process again. After readjusting a few times the project team is likely to finally create a product that meets customer needs, as displayed in figure 6. (Cook 2007.)

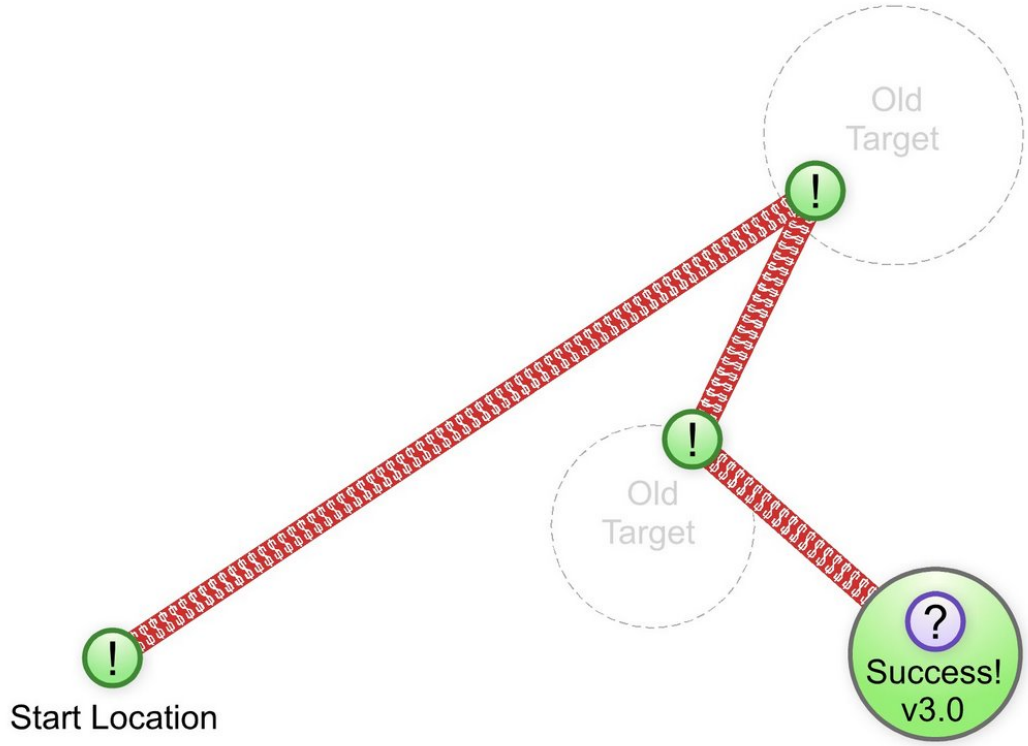

Figure 6. Readjusting product requirements in waterfall model to reach success. (Cook 2007)

Through multiple failures a company is more likely to learn what the customers actually want. Each time the project is readjusted the chances of creating a successful product increase. But it is expensive to go through the whole waterfall process to gain insights about the customer needs and then having to change direction. (Cook 2007.)

Achieving success through several failures has been proven to be effective model by several new industries. This shows that having knowledge about customer needs is necessary to building a successful product. But in waterfall model the company only gains insights about the customer needs when they reach their target. There are alternative methods that speed up the process of building successful products. (Cook 2007.)

4.2 Standard portfolio model

Portfolio model combines multiple waterfall projects into one product development model. It has similar downfalls as waterfall model where a single project is costly to produce and it cannot be adjusted during the development process. On the other hand, standard portfolio model enables a company to try several options at the same time, which speeds up the process of possibly producing a successful product. (Cook 2007.)

Figure 7 shows a company developing multiple projects with the expectation that at least one of them will hit a target and become a success. This increases the chances that the company will create a successful product faster than by developing a single product using waterfall model. (Cook 2007.)

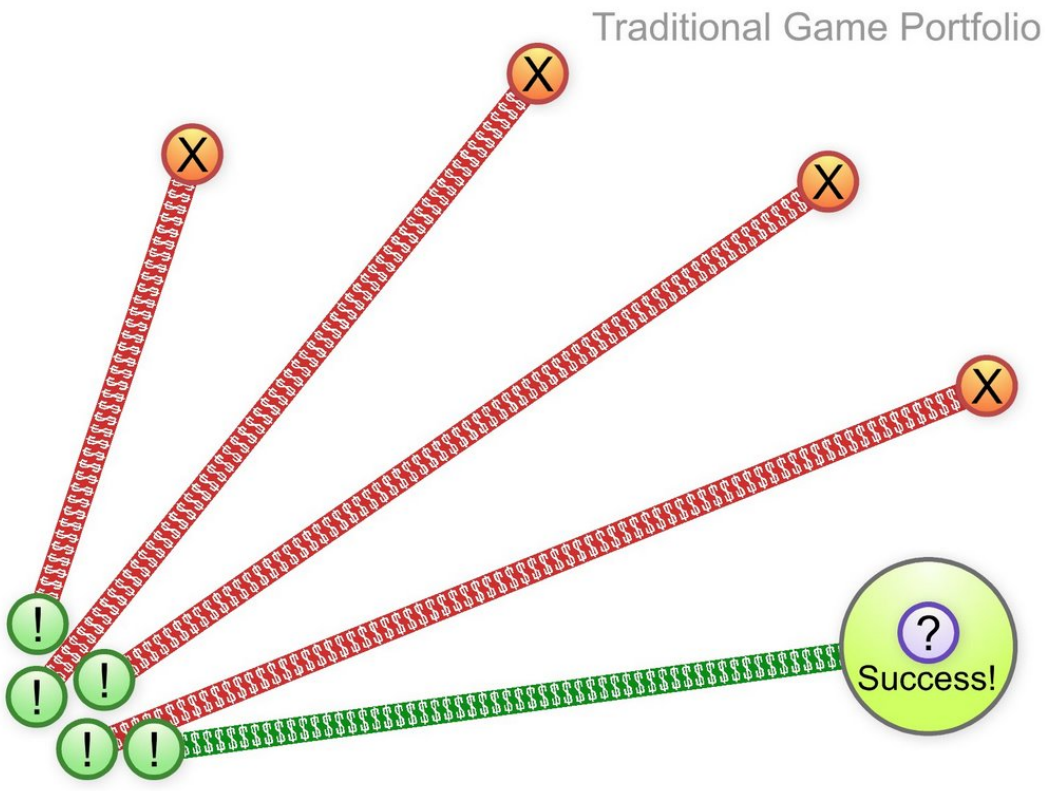

Figure 7. Standard portfolio model. (Cook 2007)

Standard portfolio model is suitable for companies that have limited understanding of the target market. The model does not force a company to choose a single product to focus on without proper understanding of customer needs. It enables the company to spread the risk and increase the likelihood of succeeding by trying multiple things. The downfall of this is the cost. Many smaller companies cannot simply afford trying their luck with multiple products simultaneously and they have to continue using waterfall product development model or look into other alternatives. (Cook 2007.)

4.3 Iterative model

Iterative model utilizes agile development practices, and is the opposite of waterfall model. The idea of iterative model is to balance planning with reality and to focus on actual customer needs during the process, unlike in waterfall model where most of the planning is done in the beginning of the process. Iterative model allows a responsive and predictable approach to development by balancing different aspects of game development within small iterations. (Keith 2014.)

A project starts by defining and creating a simple version of the product during the first iteration, which is then reviewed to see if it meets customer needs and used to identify further requirements. Based on the feedback the project requirements are redefined and the project's target adjusted accordingly (figure 8). The following iteration will then guide the project towards the new target. (ISTQB Exam certification 2012; Cook 2007.)

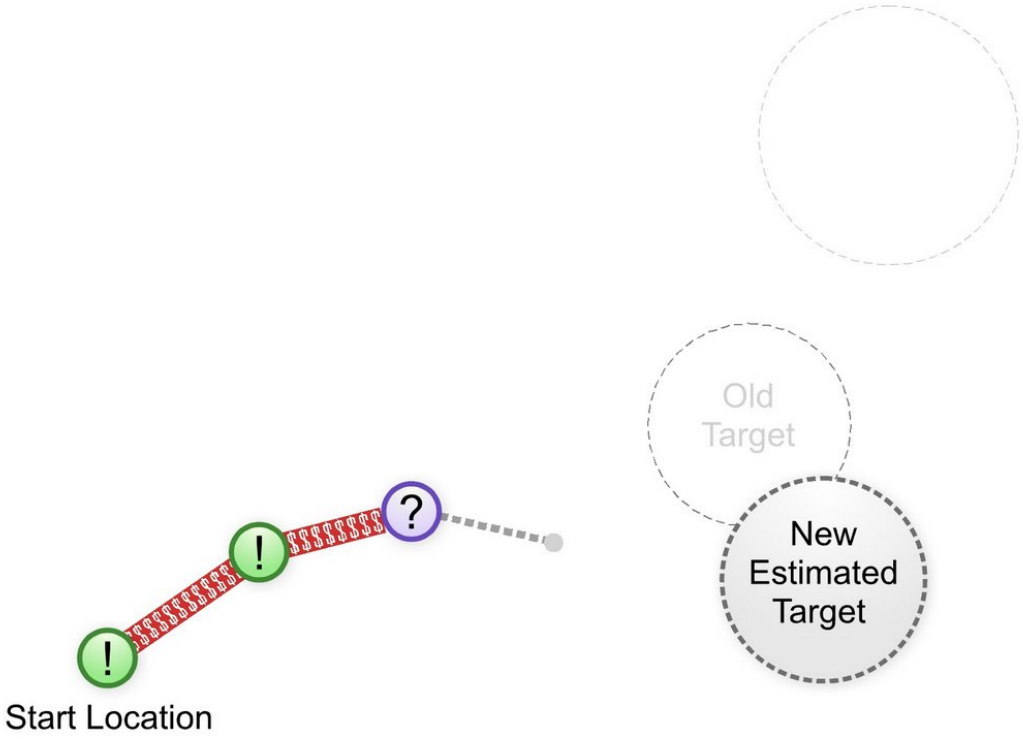

Figure 8. Each iteration leads the project closer to its target. (Cook 2007)

Each iteration incrementally improves the product and enables the project team to gain more knowledge about it. The rapid iterations enable short feedback cycles, which provide opportunities for learning and making adjustments needed for building a successful product. All the small changes will cumulate over the project life cycle and allow the project team to reach their target efficiently. Any mistakes made during the project will not offset the whole project. The team will most likely end up only wasting a few iterations but can then correct their efforts due to the short feedback loops (figure 9). (Cook 2007.)

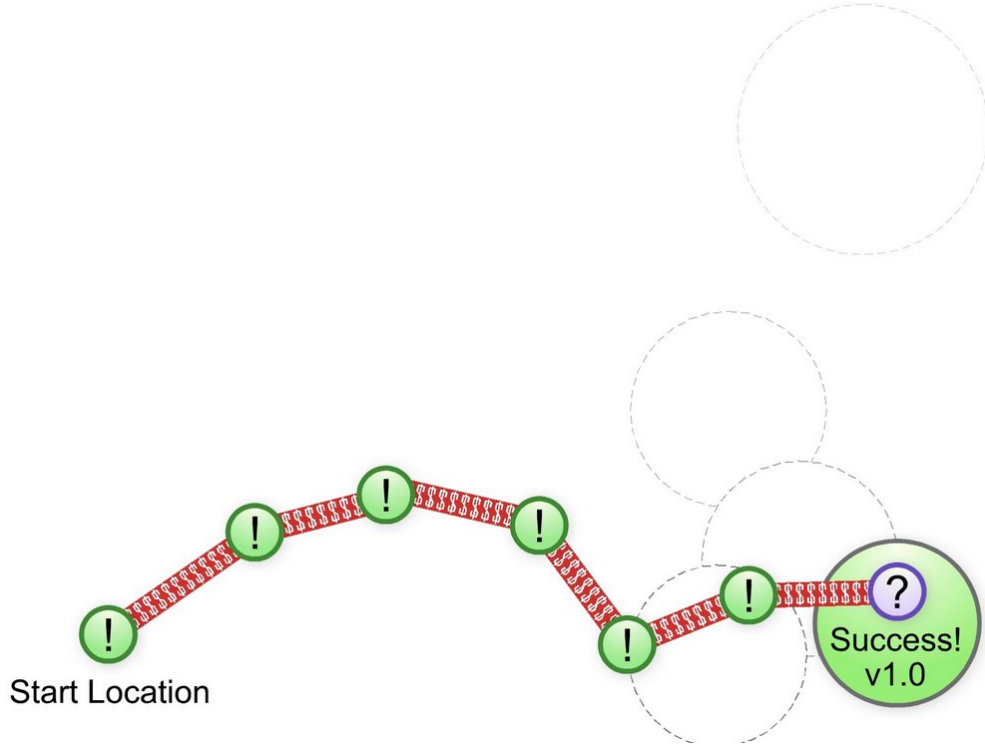

Figure 9. Short feedback cycles correct project's direction. (Cook 2007)

The advantages of iterative model come from the ability to react to changes and being focused on customer needs. In iterative model the project starts by only having a high-level design of the product and the design is then evolved during the process. The project team is focused on building working software and incrementally adding features that bring the most value to the customer. Instead of spending the whole project life cycle developing a product they can evaluate, they build a new version every two to four weeks. With iterative model the project team is more likely to build a product that serves actual customer needs. (Cook 2007; ISTQB Exam certification 2012.)

Iterative model also has disadvantages. First of all, it requires the customer to be actively involved. This will eventually lead to building a better product but can be time consuming for the project team to manage. Secondly, iterative model requires good communication and coordination skills from everyone involved in the project. Cross-functional project teams work closely together developing features and ideas in few weeks rather than months. It is essential that management structures will not hinder the process and everyone is able to communicate with each other. Thirdly, when an iteration ends and the product is reviewed

there is plenty of feedback available. This feedback can also come in the form of informal requests or increased customer demands. This can lead to unplanned work being done or increase in the project scope, if not managed properly. (Cook 2007; DiCaterino & Green 1998, 6; ISTQB Exam certification 2012.)

Other disadvantages that iterative model has are related to the constant adjustment of requirements and rapid iterations. Making major design or system changes, especially late in the project life cycle, may require significant overhauls to the already created parts of the product. The short iterations also push the project team to focus on developing the most valuable features, which removes the chance to explore many alternative ideas. The project will adjust to the closest target that meets the project requirements but it might miss potentially bigger opportunities (figure 10). For companies whose goal is to deliver a product this is beneficial, but the approach lacks strategic portfolio management. (Cook 2007; DiCaterino & Green 1998, 6; ISTQB Exam certification 2012.)

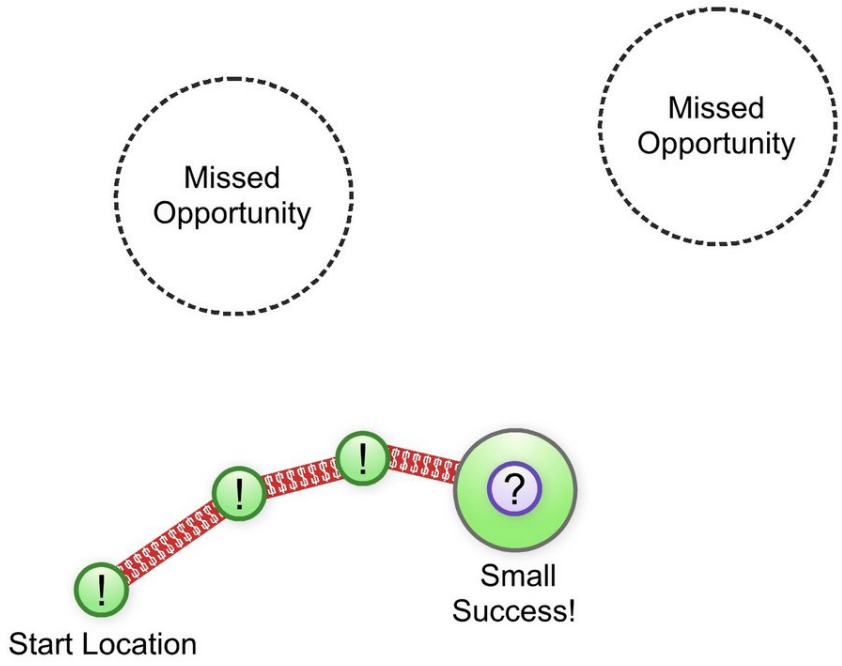

Figure 10. Iterative approach may lead to missing other opportunities. (Cook 2007)

By iterating and adjusting project requirements the project is likely to end up reaching its target at some point, but without project stages and management guidance, the project may lack medium-to-long term goals and be too focused on each iteration. The progress of the project may end up resembling a random path that has no end. This makes it difficult to es-

timate what resources are required to complete the project and when it actually going to be completed. Also by not being able to show how each iteration contributes to developing the final product, a project team may lose its focus and the support of stakeholders. (Bittner & Spence 2005.)

Iterative model is usually suitable for small newly established companies, because it reduces the risk and emphasises delivering value to customers as soon as possible. When the company only has funding for developing one product, it is important for them to be able to adjust the project towards success. Overall agile approach delivers more value quicker with lower risk than waterfall. (Cook 2007.)

4.4 Stage Gate model

A company using the stage gate model launches several products similar to standard portfolio model so that it has multiple chances to succeed. But instead of waterfall approach, project teams utilize iterative development techniques for developing individual products, which enables them to direct every project towards success. On top of this, stage game model adds project stages and kill gates, which are used to manage the development process of products. (Cook 2007.)

Kill-gates are decision-making points, which occur at the end of every project stage. Each kill gate has its own set of requirements, which are usually company wide. Every project is evaluated against these requirements to decide if it should continue to the next stage or if it should be cancelled. The requirements become stricter on every subsequent gate so that a company can better identify the projects that can actually become successes. Figure 11 displays projects progress through the stage gate process, some are cancelled during development and only few projects are completed. (Law 2006.)

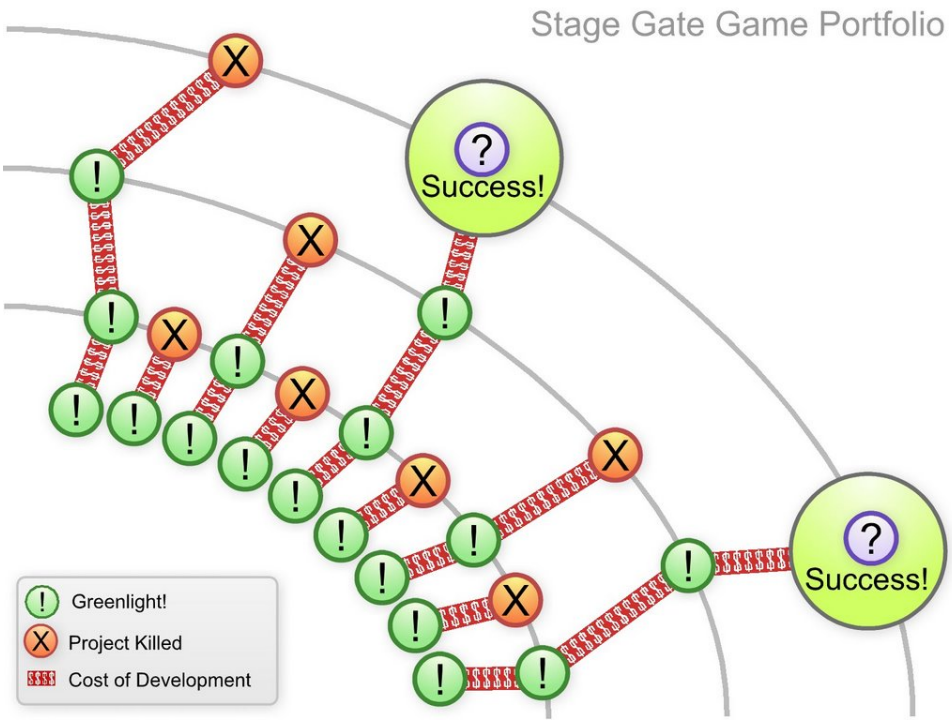

Figure 11. The progress of multiple projects through stage gate process. (Cook 2007)

Stage gate model enables companies to improve their product portfolio and techniques used to vet the products. The goal of the model is to allow companies to rapidly and efficiently develop new successful products and to maintain diverse portfolio. Each product in development is targeting a different market section and its own set of customers. With the proper product diversity the company increases its chances of building a product that finds its target market and becomes a success. (Cook 2007; Cooper 2009; Law 2006.)

The project teams building the products in the stage gate model utilize agile development practices. This will keep the feedback loops short, and the product in a proper state for gate evaluations. Agile approach also enables the teams to add incremental customer value to the product and gain more knowledge about it. This information can be then used to compare the product against the company's requirements on each kill gate. The products that meet their goals and display the most potential, are developed further. The projects that did not reach their goals are cancelled and the recourses from those are invested on the successful projects or used to start new projects. (Cook 2007.)

Even if a project is cancelled it does not mean it becomes complete waste. Stage gate model has an item called concept bank. Sometimes the market situation changes or a company identifies new emerging market, which is a good opportunity to revisit old cancelled projects. It is possible that when the projects were initially developed the time was not right for them, but they can still be turned into successful products. (Cook 2007.)

When all the aspects of stage gate model are combined it becomes a funnel (figure 12). The funnel filters the poor performing products and places them into the concept bank. The products that met their goals continue down the funnel. In the end only a few products will actually be launched to the market. (Law 2006.)

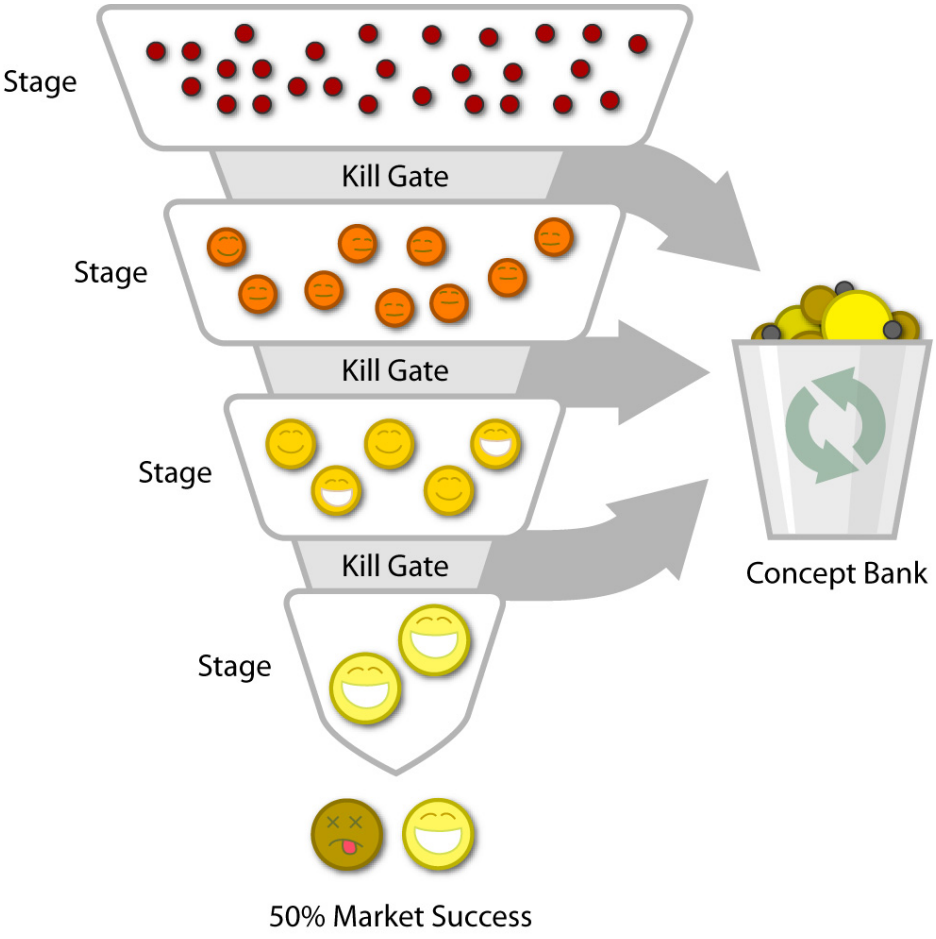

Figure 12. Stage gate model. (Law 2006)

One of the biggest benefits of stage gate model and the funnel is the learning it enables. Because the small agile teams in the company are developing products to multiple directions, there are more opportunities to learn compared to either iterative or standard portfolio model. (Cook 2007.)

The stage gate model is also less costly than standard portfolio model. Agile practices and kill gates make sure that poor performing products are identified and cancelled as early as possible. In standard portfolio model products are always finished and launched, which makes especially the unsuccessful products very costly. Instead of wasting the resources, stage gate model enables a company to invest those into exploring new opportunities or developing the more potential products. (Cook 2007.)

Compared to purely iterative model, stage gate model increases company's success rate. Iterative model focuses on developing a single product, which through several iterations should reach its target and become a success. Unfortunately not all products will have equal success, so instead of sticking to a single product stage gate model encourages a company to explore multiple directions. A company that is using the stage gate model is more likely to build a successful product and be faster in doing so, than a company using iterative model. Also by heading to multiple directions, the company is more likely to identify emerging market opportunities and shift resources into the projects heading to that direction. (Cook 2007.)

The downside of stage gate model is that it costs more than pure iterative model. Some of the company's resources need to be invested in exploring new opportunities. This is usually offset by the fact that the company is faster in developing successful products and capable of identifying the bigger success opportunities. Especially in mobile games market this is beneficial, because it is a hit-driven market and developing a good product is not enough to sustain most companies. The top ten games generate 25 % of the global revenue and a niche success only generates a small fraction of the top game's revenue. (Cook 2007, Nutt 2015.)

Another downside of stage gate model is that it can hinder innovation. The requirements and criteria used to evaluate each product on kill gates are based on company's previous experience and some very innovative ideas might not pass the gates. To avoid this downfall companies need to constantly keep improving their kill gates and tailor them to suit innovative ideas. (Cook 2007.)

Due to the significant benefits over more traditional product development models stage gate model has become the most popular model in industries where rapid and efficient new product development is integral part of the business. This is also the case for mobile game industry where product life cycles are short, competition is fierce, and customers are demanding more. Building successful game products has never been as difficult. According to

Law (2006) only 20 % of products launched actually turn a profit and 3 % of the launched products become successful enough to keep a game company afloat. So it is important that game companies utilize a model that enables them to improve their product portfolio and techniques used to manage the products. (Cooper 2009; Telfer 2014.)

5 WOOGA'S HIT FILTER

Some of the most financially successful mobile game development companies, like Supercell and King, already have well-defined processes in place for developing and evaluating games. They have been quick in adapting to the market changes and able improve their process over-time. This has enabled them to build extremely successful games and continue doing so. (Borison 2014; Street 2013; Supercell n.d..)

This chapter will describe the framework Wooga GmbH, another major mobile game development company, utilizes to continuously develop successful mobile games. Their framework is based on the stage gate model and the purpose of this chapter is to give a concrete example of how stage gate is applied currently in the mobile game industry.

5.1 Wooga GmbH

Wooga GmbH is a mobile game development company that was founded in 2009. The company is based in Berlin and currently employs over 250 people from more than 40 nations. Wooga's focus is on developing free-to-play games for mobile devices, and this far they have managed to create six very successful games. (Begemann 2014.)

The company started first by developing Facebook games but has since shifted to developing only mobile games due to the changes in game industry. The transition was caused by the declining success of Facebook's gaming platform, while the mobile game market was growing rapidly. This attracted many game development companies, including Wooga. (Begemann 2014.)

When Wooga started to transition to mobile game development in 2012, they also had to rethink their product development model. The first reason for this was that they had just cancelled their first game that was supposed to be launched on Facebook. The project team and the company executives realised that the product would not become a success, and they had to be cancel it. The issue was that they had no process in place for this. Even the first five mobile games Wooga developed followed the waterfall product development model and only one of them became a hit product. (Begemann 2014.)

The second reason was the more competitive and hit-driven nature of mobile game market. Currently there are over 1000 new mobile games released every week and only very few of them become profitable. For professional game development companies developing a barely profitable games is not enough, the games have to also generate enough revenue to sustain the company and cover the cost of failed projects. Figure 13 displays Wooga's estimate how daily revenue of mobile games is distributed. (Begemann 2014.)

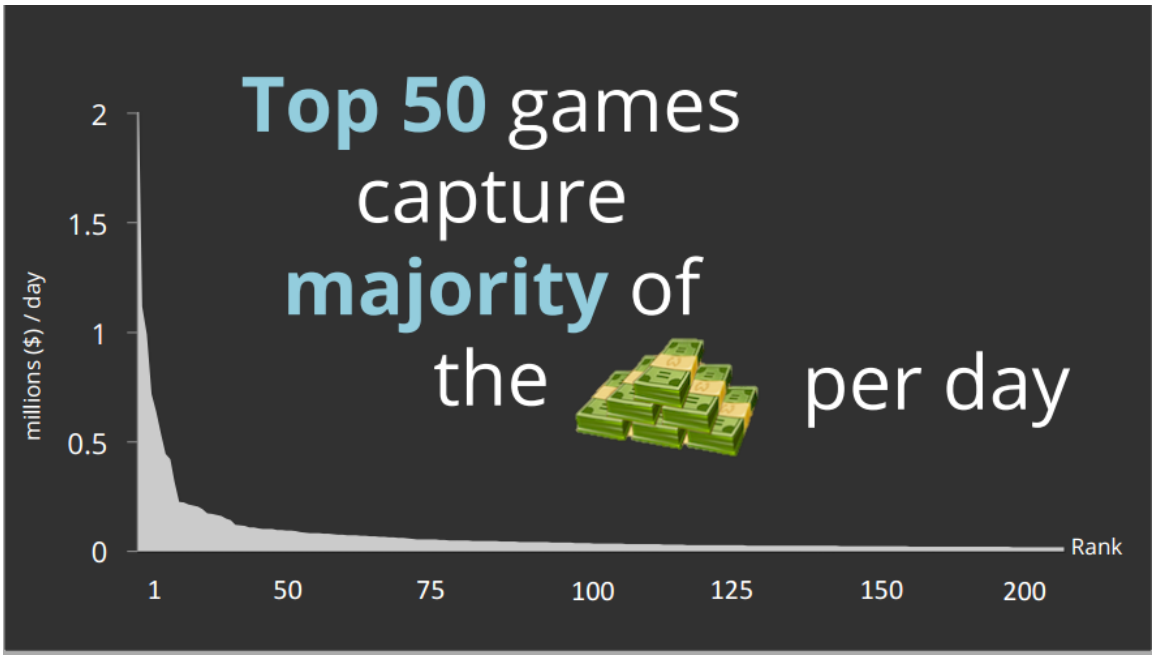

Figure 13. Daily revenue distribution of mobile games. (Kriese 2014)

The graph shows that top fifty games generate hundreds of thousands of revenue per day. Developing even one product that reaches that critical level of success can sustain a company the size of Wooga. The lack of proper process, changing market situation, the need to create hit games led to Wooga creating a product development model they call hit-filter. It enables them to develop critical hits and cancel games that do not display enough potential. (Begemann 2014.)

5.2 Hit filter

Wooga's hit filter is a system that fosters a culture that creates hit products. The system makes cancelling projects normal and allows the company to focus on developing critical hits. For Wooga developing a good game is not enough in the current market situation, they want to build great games that become big successes. (Begemann 2014.)

The idea behind hit filter (figure 14) is that it whittles down lots of game ideas into a few very potential ones. This resembles closely to the stage gate model described in chapter 4.4. (Wooga n.d.)

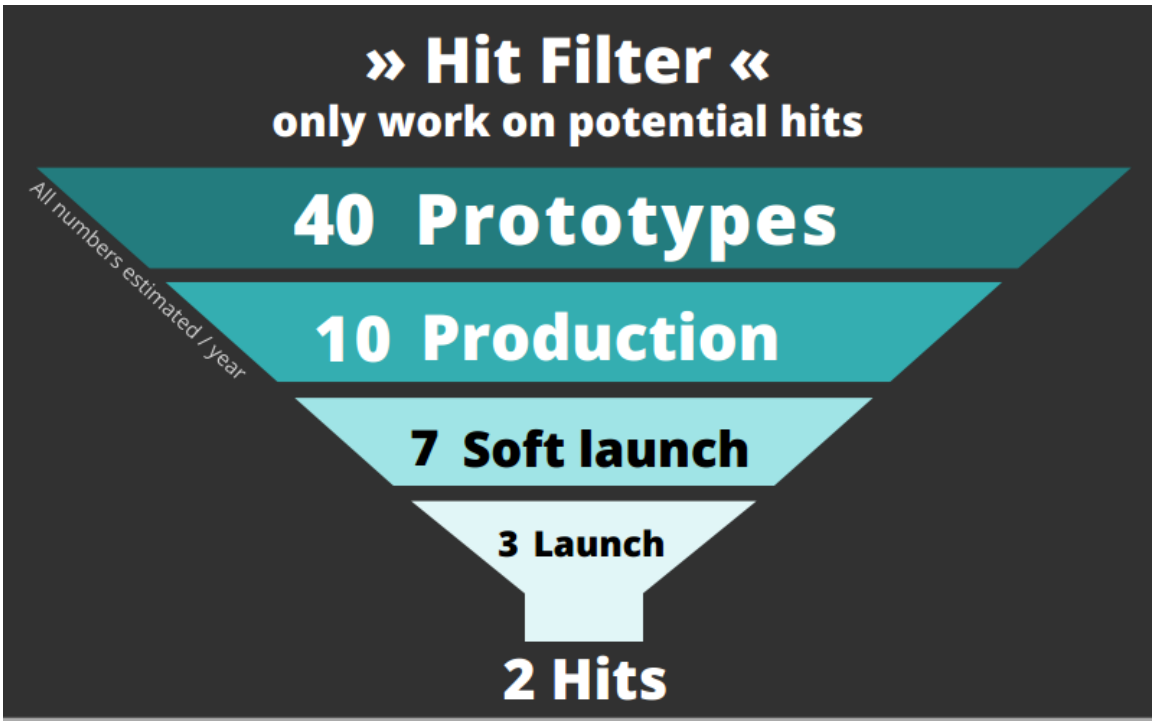

Figure 14. Wooga's hit filter. (Kriese 2014)

Every year at Wooga, approximately hundred game ideas are created. From those ideas only forty are turned into prototypes. Every stage the hit filter keeps whittling down more and more ideas that do not make pass the kill gates. Only ten ideas make it into production and seven of those into soft-launch stage. In the soft launch stage Wooga collects user data to measure a game's retention to decide whether it should be launched or not. From all the ideas created each year, Wooga's goal is to launch three that have the potential to become hits. But they only expect that two of the launched games actually become hits products. (Begemann 2014.)

It is also worth noting that each stage of the hit filter matches game development project stages described in chapter 3. Wooga is not reinventing the wheel but applying the best practices to its development process. The company has identified that the most important stages for developing successful products for them are: prototyping, production, soft-launch, and full launch. This makes it clear for each project team how they should structure their project. (Wooga n.d.)

5.3 Managing failure

The hit filter also creates challenges for everyone involved. Wooga is cancelling multiple projects each year, which has led to a result that almost everyone in the company has worked in a project that was cancelled. This can be very demotivating experience. (Begemann 2014.)

Handling failure is difficult even in professional work environment. Project teams often put their best effort in developing a game that turns out to perform poorly. Most of the projects will never even get to the hands of customers, which leads to employees having bare resumes. Also when a project is cancelled, the company has to be ready to move employees to other projects, so that employees do not feel insecure about their jobs and resources are not going to waste. (Telfer 2014.)

Wooga's hit filter has built in features to address all these issues and to leverage the benefits of cancelling projects. The most important of these features is the review meeting. Review meetings happen at regular intervals and at every kill gate. It is the meeting where a project is evaluated, and based on that decided if it should continue or be cancelled. What Wooga does differently to many companies, is that they empower the project team. Each project team has a product lead that has the final say about continuing or cancelling a project. This removes the pressure from the project team to sell the game to the company's executives and lets them focus on building a great game. In the review meetings the expert board merely provides their feedback and suggestions. It is then up to the project team to decide the next steps. (Begemann 2014.)

Another important feature of the hit filter is knowledge transfer. It is beneficial to the whole company that the lessons from cancelled projects are shared to everyone. This will make the future projects better, and allow the team members to feel that they are contributing to a greater good even if their project was cancelled. Usually the lessons are shared in quick meetings and by transferring the team members into other projects. (Begemann 2014.)

For employees that cannot immediately jump into a new project, Wooga has created lab time. Its purpose is to reduce job insecurity when an employee left without a project. The idea is that the employee can work on improving their personal skills or develop useful tools for the company. This ensures that there is always work for people who are in between projects. (Telfer 2014.)

The hit filter has helped Wooga to create a culture that accepts failure and where cancelling projects is seen normal. This is essential for creating hit games and to be able to compete in the mobile game market. The benefits of the hit filter are numerous. It enables everyone in the company to learn more about the market by seeing the successes and failures. Everyone can also start to understand what is required to develop a hit game. If the market changes, the company can adjust the hit filter to meet the new market requirements. And by filtering the poorly performing projects, everyone in the company knows that the projects that are being worked on are important and the company believes in them. All the projects that are being worked on are important for the company and everyone knows they are contributing. (Telfer 2014.)

6 SUMMARY AND ANALYSIS

The goal of the thesis was to describe a product development model that enables mobile game companies to, rapidly and efficiently, develop new game products. The model should also help companies to refine their success criteria and monitor progression of projects during development. All these aspects combined should allow companies to make well-informed product decisions and develop successful games.

The first two chapters inspected the game development process and project stages used to manage game projects. The purpose was to identify which development process, waterfall or agile, is more suitable for mobile game development and to describe how a game project is structured with stages. The conclusion was that agile approach works better for developing new products, by delivering customer value faster and enabling the game to be evaluated throughout the development process. In addition it was found out that standard agile methodology does not use project stages, but game development teams, utilizing agile methodology, have integrated the stages into their process due to the complexity of game development.

The third chapter covered four product development models employed to create successful game products and to manage a company's game portfolio. The result was that two of the models were based on waterfall process, which is expensive and inefficient way of developing new products. The other two models, iterative and stage gate, were based on agile methodology and therefore more suitable for mobile game development. Both of the models can be used to develop successful games, but iterative model lacks in portfolio management and identifying opportunities. This means that especially for medium or large-sized mobile game company stage gate model is the best option.

The fourth and last chapter provided a concrete example of how stage gate model has been adopted by the successful mobile game company Wooga. It described Wooga's approach to managing product development and its hit filter process. This showed that it is important for a mobile game company to have a product development model that enables them to explore multiple opportunities simultaneously and allows them to cancel poorly performing projects, if they hope to build successful games.

Based on the observations made in this thesis, it can be concluded that the combination of agile game development practices and stage gate model enables mobile game companies to build successful games faster and more efficiently. By adopting this combination companies can start to improve their development process and focus on games that display hit-potential. With better development process and product development model, companies can expect to be able to experiment with more products and have a good chance at succeeding with the games they choose to launch to the market.

The sources used for this thesis mainly consist of literature pieces and online sources, from respected authors in the game industry. Game industry has adopted a lot of practices from other industries that are more mature and have more experience about new product development, therefore there is plenty of trustworthy material available. For the mobile game development specific topics, it was necessary to settle for online sources that provide up-to-date information, due to the young age of mobile game industry. Fortunately mobile game companies have been open about their product development and there is also plenty of data available that can be used to verify the current state of the market.

The main challenge for me when writing this thesis was to describe game development processes and product development models on a high-level. The perspective in this thesis is on a company level rather than team level. I have worked three years in game development industry, in the project teams, and observed these models from the team's perspective. Trying to not get tangled on the smaller details was hard but also gave me a good understanding to the theories behind the processes and models.

Researching and writing the thesis has enabled me to better understand processes and models required to develop new successful games. I am also sure I can utilize this knowledge at my everyday work at Wooga and try to improve the company's hit filter. It would be also interesting to further research the product development models used in mobile game industry by interviewing other companies to discover more best practices and possible alternative models.

The information presented in this thesis can also be useful for any companies struggling to produce successful games or new companies establishing their product development model. The goal of every game developer is to create successful games and choosing the most suitable product development can both save time and money. Understanding the models that are currently prominent in the game development industry paves the way for future success. The industry will keep evolving and it is important to keep up with it. By having the knowledge how the processes have evolved and the reasons behind the changes, it is possible to be prepared for the future.

SOURCES

Adams, E. 2010. Fundamentals of Game Design, Second Edition. United States of America: New Riders

Agile Methodology. (2008, 23rd October). Agile Methodology. Accessed 26.10.2015, on the site Agile Methodology, internet access: http://www.agilemethodology.org/

Bates, B. 2004. Game Design. United States of America: Thomson Course Technology PTR.

Begemann, J. (23.12.2014). Critical hits and new misses. Internet address: https://www.youtube.com/watch?v=VN4fzCZ_Cek

Bestebroer, P. (2015, 20th October). The four phases of game development. Accessed 29.10.2015, on the site Gamasutra, internet access: http://gamasutra.com/blogs/PascalBestebroer/20151020/256765/The_four_phases_of_game_development.php

Bethke, E. 2003. Game Development and Production. United States of America: Wordware Publishing.

Bittner, K. & Spence, I. (2005, 5th May). What is iterative development? Accessed 5.11.2015, on the site IBM Bluemix, internet address: http://www.ibm.com/developerworks/rational/library/may05/bittner-spence/

Borison, R. (2014, 20th May). The 15 Highest-Grossing iPhone And iPad Games. Accessed 13.10.2015, on the site Business Insider internet address: http://www.businessinsider.com/highest-grossing-iphone-and-ipad-games-2014-5?op=1&IR=T

Bonin, H. (2014, 28th April). In Search of Bigfoot (The eight elusive phases of game development). Accessed 26.10.2015, on the site Gamasutra internet address: http://www.gamasutra.com/blogs/HarvardBonin/20140428/216386/In_Search_of_Bigfoot_The_eight_elusive_phases_of_game_development.php

Cagan, M. (2005, 15th February). The Waterfall Product Development Process. Accessed 1.11.2015, on the site Silicon Valley Product Group, internet address: http://svpg.com/the-waterfall-product-development-process/

Chernyak, U. (2014, 13th October). Understanding "Games as a Service" for Game Development. Accessed 31.10.2015, on the site Gamasutra, internet access: http://www.gamasutra.com/blogs/UlyanaChernyak/20141013/227536/Understanding_quotGames_as_a_Servicequot_for_Game_Development.php

Cook, D. (2015, 12th April). Minimum Sustainable success. Accessed 4.10.2015, on the site Lostgarden internet address: http://www.lostgarden.com/2015/04/minimum-sustainable-success.html

Cook, D. (2007, 18th February). Rockets, Cars and Gardens: Visualizing waterfall, agile and stage gate. Accessed 30.9.2015, on the site Lostgarden internet address: http://www.lostgarden.com/2007/02/rockets-cars-and-gardens-visualizing.html

Cooper R. (2009). How Companies are Reinventing their Idea-to-Launch Methodologies [pdf-document]. Accessed on the site Stage-Gate, internet address: http://www.stage-gate.net/downloads/wp/wp_38.pdf

DiCaterino, A. & Green, D. (1998, 1st February). A Survey of System Development Process Models [pdf-document]. Accessed on the site Center for Technology in Government: http://www.ctg.albany.edu/publications/reports/survey_of_sysdev/survey_of_sysdev.pdf

Dryer, B. (2013, 8th April). Going Agile - The 7 Simple Stages of Why and How to Get it Done (I'm not saying easy). Accessed 27.10.2015, on the site Gamasutra internet address: http://www.gamasutra.com/blogs/BrianDreyer/20130408/190077/Going_Agile__The_7_Simple_Stages_of_Why_and_How_to_Get_it_Done_Im_not_saying_easy.php

Fischer, J. (2014, 30th April). Competitive Advantage and the Productivity Frontier, Or Why Dark Souls is the Ikea of Game Development. Accessed 29.10.2015, on the site Gamasutra, internet access: http://www.gamasutra.com/blogs/JustinFischer/20140430/216624/Competitive_Advantage_and_the_Productivity_Frontier_Or_Why_Dark_Souls_is_the_Ikea_of_Game_Development

ISTQB Exam certification. (2012). What is Iterative model- advantages, disadvantages and when to use it?. Accessed 3.11.2015, on the site ISTQB Exam Certification, internet address: http://istqbexamcertification.com/what-is-iterative-model-advantages-disadvantages-and-when-to-use-it/

Kain, E. (2015, 28th August). 'Angry Birds' Developer Layoffs No Surprise In Volatile Mobile Industry. Accessed 7.10.2015, on the site Forbes internet address: http://www.forbes.com/sites/erikkain/2015/08/28/angry-birds-developer-layoffs-no-surprise-in-volatile-mobile-industry/

Keith, C. 2010. Agile Game Development with Scrum. United States of America: Addison-Wesley Professional.

Keith, C. (2014, 22nd September). Why Agile Game Development. Accessed 27.10.2015, on the site Agile Game Development internet address: http://blog.agilegamedevelopment.com/2014/09/why-agile-game-development.html

Kriese, S. (4.11.2014). Killing as Usual - Wooga's Hit Filter. Internet address: http://www.gdcvault.com/play/1020855/The-Art-of-Killing

Law, L. (2006). Group Report: Building Innovative Games That Sell. Accessed 8.10.2015, on the site Project Horseshoe, internet address: http://www.projecthorseshoe.com/ph06/ph06r5.htm

Lotz, M. (2013, 5th July). Waterfall vs. Agile: Which is the Right Development Methodology for Your Project? Accessed 17.10.2015, on the site Segue Technologies internet address: http://www.seguetech.com/blog/2013/07/05/waterfall-vs-agile-right-development-methodology

McGuire, R. (2006, 28th June). Paper Burns: Game Design With Agile Methodologies. Accessed 30.9.2015, on the site Gamasutra internet address: http://www.gamasutra.com/view/feature/131151/paper_burns_game_design_with_.php?print=1

Mobiledevmemo. (2.9.2013). Soft launch product development using the Minimum Viable Metrics. Internet address: http://mobiledevmemo.com/soft-launch-product-development-using-the-minimum-viable-metrics/

Nutt, C. (2015, 21st October.) The top 10 games take 25 % of global mobile app revenue. Accessed 14.11.2015, on the site Gamasutra, internet address: http://www.gamasutra.com/view/news/256997/The_top_10_games_take_25_of_global_mobile_app_revenue.php

Pathak, R. (2014, 25th March). Top 5 Project Management Phases. Accessed 28.10.2015, on the site Project-management, internet address: http://project-management.com/top-5-project-management-phases/

Pearson, D. (2014, 22nd October). Report: Mobile become gaming's biggest market by 2015. Accessed 13.10.2015, on the site Gameindustry internet address: http://www.gamesindustry.biz/articles/2014-10-22-report-mobile-to-become-gamings-biggest-market-by-2015

Pocketgamer. (2015, 1st November). Count of Active Applications in the App Store. Accessed 1.11.2015, on the site Pocketgamer, internet address: http://www.pocketgamer.biz/metrics/app-store/app-count/

Rouse, M. (2014, 30th June). Product development (new product development) definition. Accessed 1.11.2015, on the site Techtarget, internet access: http://searchcio.techtarget.com/definition/product-development-or-new-product-development-NPD

Segue Technologies. (2015, 25th August). 8 Benefits of Agile Software Development. Accessed 27.10.2015, on the site Seguetech, internet access: http://www.seguetech.com/blog/2013/04/12/8-benefits-of-agile-software-development

Shapina, A. & Shpilber, P. (2013). Mobile Application Launch Checklist (Marketing and Analytics) [PowerPoint-presentation]. Accessed on the site Slideshare, internet access: http://www.slideshare.net/pshpil/mobile-launch-checklist

Sims, G. (2015, 20th April). Google Play Store vs the Apple App Store: by the numbers (2015). Accessed 1.11.2015, on the site Android Authority, internet address: http://www.androidauthority.com/google-play-store-vs-the-apple-app-store-601836/

Sivak, S. (2013, 5th March). The Future of Games as a Service. Accessed 31.10.2015, on the site Gamasutra, internet access:

http://www.gamasutra.com/blogs/SethSivak/20130305/187766/The_Future_of_Games_as_a_Service.php

Statista. (2015). Number of available applications in the Google Play Store from December 2009 to July 2015. Accessed 1.11.2015, on the site Statista, internet address: http://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/

Street, Z. (2013, 7th October). Understanding King.com's production line. Accessed 13.10.2015, on the site Gamesbrief internet address: http://www.gamesbrief.com/2013/10/understanding-king-coms-production-line/

Supercell. (n.d.). Our Story. Accessed 21.2.2015, on the site Supercell internet address: http://supercell.com/en/our-story/

Telfer, A. (11.8.2014). The Art of Killing Games. Internet address: http://www.gdcvault.com/play/1020855/The-Art-of-Killing

Tran, J. (2014, 6th May). How digital distribution is changing the scope of gaming. Accessed 28.10.2015, on the site Venturebeat, internet access: http://venturebeat.com/community/2014/05/06/how-digital-distribution-is-changing-the-scope-of-gaming/

Wooga. (n.d.). The Hit Filter: How We Build Games. Accessed 30.10.2015, on the site Wooga internet address: http://www.wooga.com/about/hitfilter/

Wright, C. (2009, 2nd January). A Brief History of Mobile Games: 2007/8 - Thank God for Steve Jobs. Accessed 27.10.2015, on the site Pocket Gamer internet address: http://www.pocketgamer.biz/feature/10723/a-brief-history-of-mobile-games-20078-thank-god-for-steve-jobs/

York, T. (2012, 17th April). Making Lean Startup Tactics Work for Games. Accessed 29.10.2015, on the site Gamasutra, internet access: http://www.gamasutra.com/view/feature/168647/making_lean_startup_tactics_work_.php

Figure Sources

Figure 1. @KaeTheDev. (18.3.2014). The Game Creation Process, Part 2: Designing the Idea. Internet address: http://fireside.gamejolt.com/post/the-game-creation-process-part-2-designing-the-idea-viq5rk2t

Figure 2. Agilemanifesto. (2001). Manifesto for Agile Software Development. Accessed 20.10.2015, on the site Agilemanifesto, internet access: http://www.agilemanifesto.org/

Figure 3. Keith, C. (2010). Agile Game Development with Scrum. United States: Addison-Wesley Professional.

Figure 4. Keith, C. (2010). Agile Game Development with Scrum. United States: Addison-Wesley Professional.

Figure 5 - 11. Cook, D. (2007, 18th February). Rockets, Cars and Gardens: Visualizing water-fall, agile and stage gate. Accessed 30.9.2015, on the site Lostgarden internet address: http://www.lostgarden.com/2007/02/rockets-cars-and-gardens-visualizing.html

Figure 12. Law, L. (2006). Group Report: Building Innovative Games That Sell. Accessed 8.10.2015, on the site Project Horseshoe, internet address: http://www.projecthorseshoe.com/ph06/ph06r5.htm

Figure 13 - 14. Kriese, S. (4.11.2014). Killing as Usual - Wooga's Hit Filter. Internet address: http://www.gdcvault.com/play/1020855/The-Art-of-Killing