Producer in a Game Project

A Game Producer's Workflow Case: Under

John Nousiainen

Thesis, May 2013
Business information systems
Digital media / Game production

ABSTRACT

Tampere University of Applied Sciences
Degree Programme Business information systems
Option of Digital media / Game production

JOHN NOUSIAINEN:
Producer in a Game Project
A Game Producer's Workflow Case: Under
Bachelor´s thesis 32 pages
October 2013

This thesis was intended to serve as a “mental map” for production work in video gaming industry, or in a similar multimedia production that comprehensively combines audio, art and movies via interactive user engagement. This Thesis also describes workflows that the author adopted in his own production work, most of which are also very relevant in modern software development processes.

The methodologies described here are a combination of various methods of producing, some of which were developed and are actively used by the author.

The goal was to achieve a straightforward and clear production workflow that is most efficient when used with small sized organizations of 2 to 6 people. However, the results and methodologies were also compared and analysed from the viewpoint of medium and large sized projects.

The process of building a working workflow was described and analysed through a case study of a game called “Under: Dark Void: Chapter 1”. The phases of building a modern first-person game with a small team in a limited time span were decribed. As the team worked as an indie developing team with no costs or overheads, budgeting and outsourcing were not included in the analysis. However, a hypothesis of how money would have changed things during the production is presented.

Keywords: game production, producing, software development, game development

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma
Digimedia / Pelituotannon suuntautumisvaihtoehto

NOUSIAINEN JOHN:
Producer in a Game Project
A Game Producer's Workflow Case: Under
Opinnäytetyö 32 sivua
Lokakuu 2013

Tämän opinnäytetyön tavoitteena oli laatia eräänlainen henkinen kartta tuottajille, jotka aloittelevat videopelituotannon parissa, tai ovat tekemisissä samantyylisessä projektissa, joka yhdistää kokonaisvaltaisesti niin ääntä, taidetta kuin elokuvaa loppukäyttäjän interaktiiviseen kokemukseen. Opinnäytetyössä kuvattiin myös työmenetelmiä, joita sen tekijä on omaksunut omaan työhöni tuottajana. Suurin osa tästä on myös erittäin oleellista nykypäivän ohjelmistokehityksessä.

Kuvatut metodit eivät ole suoraan yhden säännön ketterien menetelmien soveltamisaloista, vaan ennemminkin yhdistelmä eri tapoja tuottaa. Osa metodeista, jotka työn tekijä on huomannut toimivan omassa tuotannon työssä, ovat hänen omiaan.

Tavoitteena oli saavuttaa suoraviivainen, selkeä tuotantomenetelmä, joka toimii erinomaisesti työskenneltäessä pienissä, noin 2–10 henkilön organisaatioissa. Kuitenkin myös tehtiin vertauksia ja analyysejä saavutetuista tuloksista ja metodeista suhteutettuina keskisuurin ja suuriin projekteihin. Näissä tapauksissa nojattiin suurimmalta osin toisen ja kolmannen käden tietoihin, joita on kerätty laajasti eri julkaisuista, tuottajien puheista tai internetistä.

Tämän opinnäytetyön tarkoitusta luoda toimiva tuotannollinen työmenetelmä kuvattiin ja analysoitiin case-tyyppisen projektikuvauksen kautta. Tästä pc:lle luotu peli ”Under. Dark Void. Chapter 1” toimi esimerkkitapauksena, josta kuvattiin koko tuotannollinen prosessi. Tämä moderni ensimmäisen persoonan peli tehtiin pienellä tiimillä ja nopeassa aikataulussa. Esimerkkitapaus ei sisältänyt budjetointia tai ulkoistamista, koska se luotiin riippumattomana tiiminä, jolla ei ollut budjettia tai kuluja (aikaa lukuun ottamatta). Tämän on kuitenkin otettu huomioon koko esimerkkitapauksen ajan, ja opinnäytetyön lopussa heijastan hypoteeseja siitä, miten raha olisi mahdollisesti muuttanut asioita tuotannon aikana.

Asiasanat: pelituotanto, tuottaminen, ohjelmistonkehitys, pelikehitys

CONTENTS

1 PREFACE

This Thesis is about writing and analysing the role of a producer in a game-making process. It has also strong connections to other production areas, especially in software development. However, it differs in very precise key areas and expands the whole process of development so, that in the end the similarities are limited.

I have been involved in many game-projects as a producer, and I have also had a few other projects where I was managing the production of teams varying from 4 to 24 people. This background and experience also influenced me to choose it as a subject for my thesis.

There is also included a specific case study about production workflow, where I was the lead producer of a game-project for the duration of the whole process. I explain the process from a producer's point-of-view and analyse the results of the project and its team members. The game project for the case study is a multi-episodic web-based PC-game, named "Under". My production responsibility in this case-study is about the production of the first chapter of the series.

This thesis is helpful for aspiring game producers or producers who are searching for knowledge in managing smaller scaled game projects, specifically with remotely working production teams.

2 GAME PRODUCTION

2.1 Software and game producer overview

Today, the tools for making a complex game for various platforms are becoming more available and easier to use, thanks to such game engines as the Unreal Development Kit or Unity 3D. It has also inspired many start-up companies, indie developers and even groups of friends to start building games on their own. However, when orienting to a game project in a professional sense, especially with a budget that is dependent on income/funding and costs, it is imperative to have an extremely accurate plan, both production-wise and overhead-wise. This is where the role of a producer is needed. A work that someone does to carry out the project from start to finish.

A successful producer is guaranteed to make the project and its end-product(s) better. It will not guarantee a disastrous project to become a multi-million dollar success-story, but it will make it better in every aspect nevertheless. It will provide a better cost-efficiency, it will have the project become finished a lot earlier (and in time), it will make the end-product have a lot more quality, and most importantly, it will leave the people working on the project feeling a lot more satisfied and happy about their work on it. There is not a part in the process of software development or game development that wouldn’t benefit from the involvement of a dedicated producer. This is something that seems to escape many starting teams today, especially in the field of game developing.

As in all software productions, the understanding of the whole process from the start to the finish is crucial. Everything that is required to achieve the needs of the end product(s) has to be thought of and planned before starting the pre-production process. This is a crucial area, where a dedicated project producer or a team of producers work first. This is also the state where the projects outcome is mainly defined.

My role as a producer has been, and is about people management, efficiency (of work and costs), quality assurance, responsibility towards every aspect of the project (team members and clients) and inspiration. In my opinion, these could also be described as the overall qualities that every producer should have and work towards to. A member of the team that holds these elements as their workforce will have a direct influence on the success of the project in every aspect.

2.1.1 Producing game projects

Games usually involve much more than just straightforward software programming. There are artists and audio designers, music composers, modellers, animators, not to mention the teams of various designers for game mechanics, game design or level design. Even the programming teams can have so many multi-layered tasks that they are involved in, such as tools-programmers, game-engine programmers, AI-programmers etc. There might also be a need to establish community services for the game with websites, Twitter and Facebook accounts, chats, community boards, and various other things that all require dedicated work to create, update and manage.

A producer is the key to bringing in a good collaboration between such various teams and also to inform the teams about their schedules and describing the final product as accurately as possible for the duration of the whole project. Every member working on the project should have a clear vision and understanding of what they are expected to deliver and when. Also a very important but sometimes overlooked information for the projects workers is why. A producer can and should explain the workers about things that relate straightforwardly on their daily work. Why are we using a certain programming language? Why it should be done as an object-oriented code? Why the graphics have all a greyish appearance in the artist sketches?

2.1.2 Various producing tasks during game production

Obviously when making a game with the scale that would involve tens or hundreds of workers for a duration of years, it is impossible for one producer to do everything that is required of the production as a whole. There are many layers of production work, a few of which I will explain here briefly.

LINE PRODUCER

This production work usually consists of the most hands-on type of production on the project. It might be something like ordering food for the teams that are working overtimes, getting plane tickets for the programmers on their recreation day, keeping the teams up-to-date on various happenings in the project, or distributing work when a worker falls ill. This work is essential, and something that will have a great effect on the people working on the project. A successful line producer increases the amount of actual work that the team can do with helping out in everyday daunting tasks that require doing, but cost time to do.

ASSOCIATE PRODUCER

This is the most common production job when working on bigger projects. Associate producers work is similar to that of the executive producers, but with a smaller influence on the bigger picture. This job usually involves a lot of timetables, planning and highly important tasks of communication between the teams and the executive producer. Usually these works are also divided into different layers, and every company has their own form of hierarchy that divides these jobs. These might be something like assistant associate producer or senior assistant producer.

EXECUTIVE PRODUCER

This work is usually done by the company head(s) or the founder of the company. However, a hired producer can do this work also. Usually this is the case with small companies or small projects, involving a few to ten workers. Regardless of the post and its title, one similar aspect to this is the accounting and the costs and overheads of the production. This job has the responsibility of the profit-making abilities that the project has, and making everything in the producer’s power to see it does so.

2.1.3 Single producer´s role

While bigger scaled projects have various producers, each of which have divided production work, they tend to focus on their own areas very tightly. A single producer on a smaller project has a lot more areas (all of them actually!) to manage by themselves. The workloads of the single producer and a line producer might be the same, but the variance in skills and the overall knowledge and responsibilities about the project are totally different.

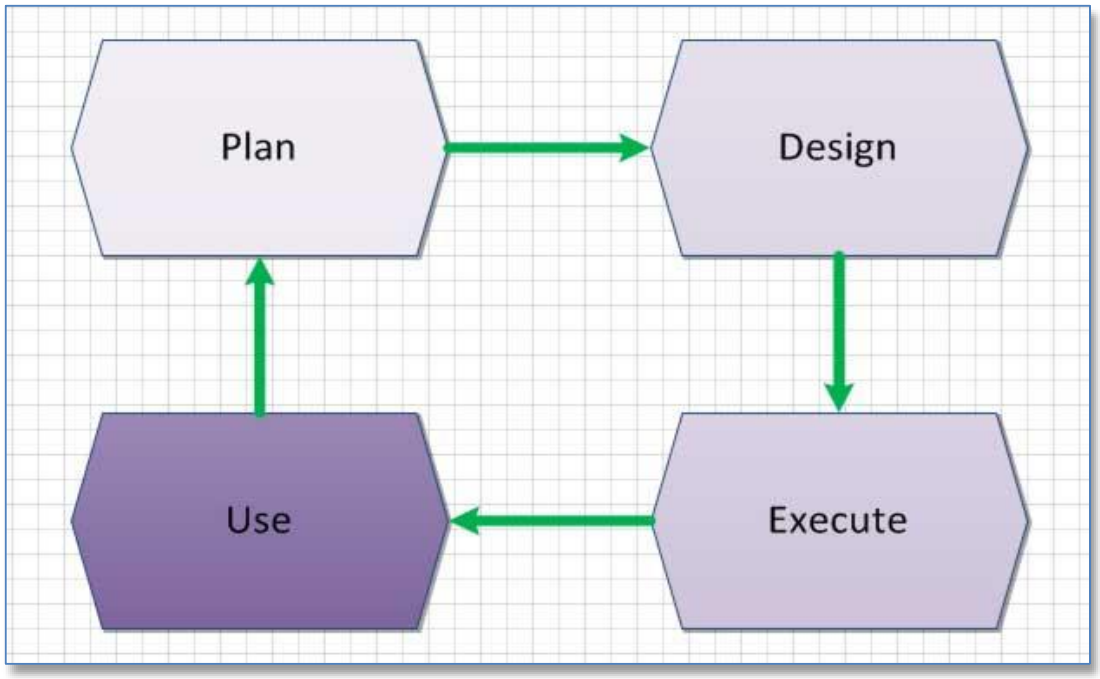
There are also some benefits that derive from single producer projects. First, it is easier to point your finger to a single person whenever there´s trouble or difficulties. It straightens the line of communication and thus, speeds the whole process up and gives the workers a mind-set that this person has the responsibility and will give a full 100% backing on them, no matter the subject. Multi-producer projects tend to lose on this one, as responsibilities are often divided and some producers might not even have the authority to solve the current problems.

Second, I believe that single producers have an advantage in simplifying things, so that the end-product matches more to the given vision or goal. In this aspect it is healthier to have one driving force to fulfil the goal, than many working first on homogenizing their vision, and then driving towards it (not to mention projects where the planned vision changes multiple times during the production).

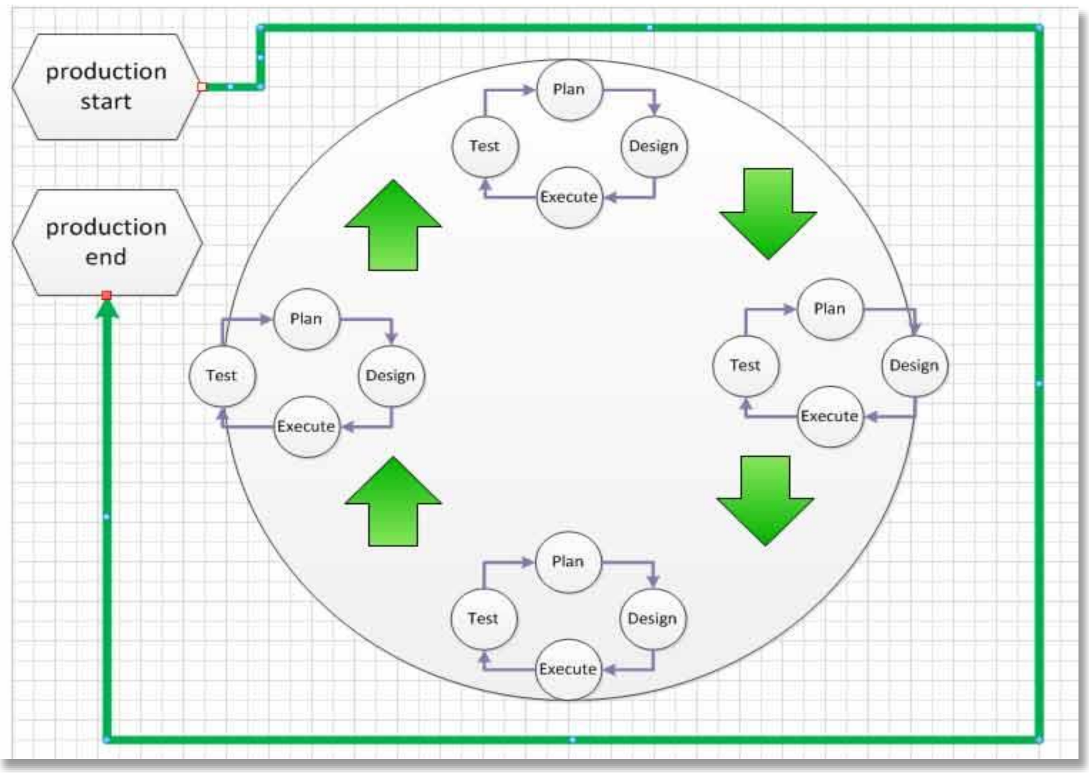
Thirdly (and this one strongly relates to points 1 & 2), there is nothing in the industry that anyone would hate more than failing. Nevertheless, failing happens to be a very relevant part of another process; becoming stronger and better at what we do. I don´t mean we should strive towards failures, but we should accept the fact that not everything we do will turn to gold. When a project with a single producer fails due to production issues (and not because of e.g. funding), the one to carry the blame should (justifiably, but not exclusively) be the producer. How the failure is handled after the project is announced dead, depends on many aspects. I myself think that as human beings, we all tend to have the urge and need to share the blame and negativity in a very social way. We don’t instinctively want to keep the negative things in our life to ourselves, but we want to share them and even form groups where the defeat or failure is sympathized, so that it makes us feel better about ourselves. Larger projects that face these unwelcomed circumstances have a whole horde of producers thinking about what went wrong and then there’s already a lot of shared blame going around. Finding the true cause of the failure (and that is the single most important thing that could be now learned from) is often pushed aside. On single producer projects, the trace to the root of the defeat is much easier to do. In my opinion, the most demanding posts can sometimes lie in the smaller projects, if the overhead capital is not taken into account.

3 PRODUCTION PROCESS IN STAGES

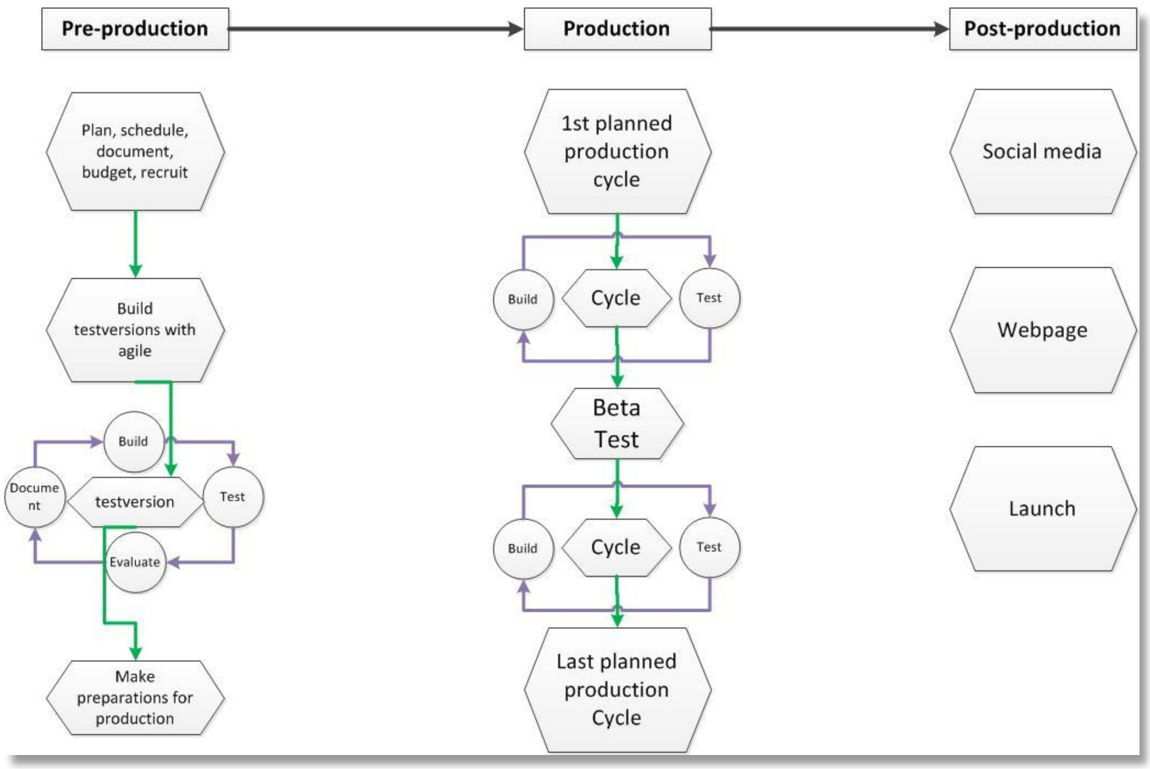
3.1 Example of production cycles used in game production.



PICTURE 1. This is a very basic production cycle, where the production cycle goes through stages of planning, designing and the actual execution before finishing. Another cycle then iterates on top of that.



PICTURE 2. This is how you could divide your production, following the basic agile methods. This diagram shows the use of production cycles during the whole production.


Pre-production
Production
Post-production
Plan, schedule, document, budget, recruit
Build testversions with agile
Build
Document
testversion
Test
Evaluate
Make preparations for production
1st planned production cycle
Build
Cycle
Test
Beta Test
Build
Cycle
Test
Last planned production Cycle
Social media
Webpage
Launch

PICTURE 3. This is how the case: Under was produced. Any game could be produced with similar methods.

Breaking down different tasks that will be going on during production is a critical part in planning out the schedules and different teams. The same cycle processes can be used to mark bigger, more overwhelming tasks in a project, such as character controls, game level 01, enemy AI etc. Also, smaller *micro* processes use the exact same production methods, e.g. such as character look (as a part of character controls), level 01 terrain production (as a part of the whole level design) or enemy AI at idle stage (as a part of the whole enemy AI).

3.2 Pre-production

Pre-production is the stage in any project that will have the producer(s) fully occupied. Even if the overall production is using a fast iteration and more improvised approach, the pre-production is the part that will almost certainly define the projects outcome. Bad pre-production will result in a bad game, mediocre at most. However, a good pre-production will not guarantee a good game, but it will definitely shape and guide the project towards success.

There are many different things that will be prepared during pre-production, and the most critical ones that the producer needs to be aware of are:

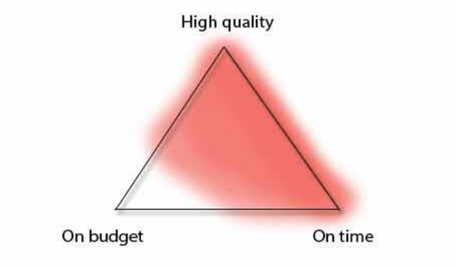
- scheduling
- financing
- recruiting.

Pre-production should also include game art through various iterative stages, some audio demos or samples and possibly even sound-effects that the game will have. The producer will also work very closely with the game designer(s) and art departments to have all the necessary data available as the teams are introduced to their respective work areas at the end of this stage.

3.2.1 Scheduling

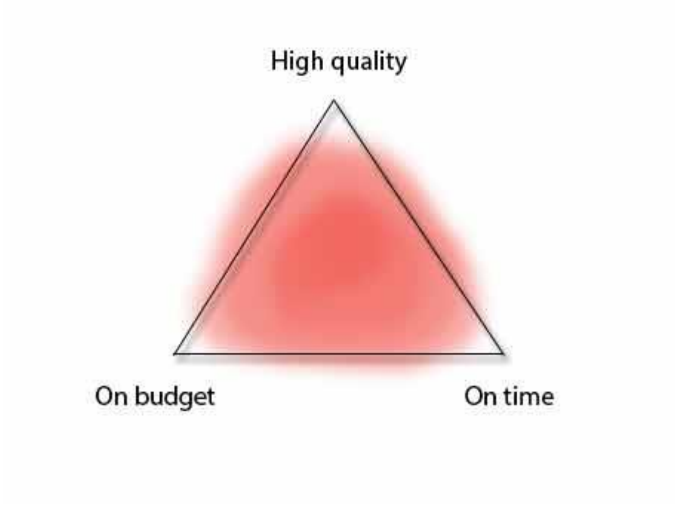
Scheduling can be done in many ways, using Gantt-charts, Kanban principles, Excel-charts or a combination of various different techniques. The main thing that will need focus from the start is the ending of the production; a deadline date that is defined by the amount of work that will be needed versus the amount of finances that will allow the project to be made in a scheduled time span.

Bethke (2003) describes in his book *Game Development and Production* the three main production qualities as an interdependent triangle. High-quality, on budget and on time are positioned at each of the points of the triangle, and you may choose 2 of them that you want to accomplish in the production. One of the points is sacrificed to accomplish the two. The following diagram shows the idea behind this.



PICTURE 4. The overall choices that a production has, versus the cost that those choices will affect. E.g. if you would choose the production to be finished on time and with a high quality, you would need to sacrifice the budget somewhat to achieve this.

Now, I don’t wholly accept this statement, even if it is an exaggerated example. During scheduling you can greatly affect the points that you wish to drive towards to and when planning the whole process, you should be able to find a place that is very close to all 3 points. None of them fully at 100%, but all at least near 90 to 95%. All of the fore mentioned points affect to each other’s causally, but I have found that by concentrating on the middle will bring you the utmost best overall result. You start being on budget when you are on time, and you are on time when you are happy with the quality. I do put more emphasis here for the quality as the driving factor, but only so that the **planned** quality will strictly respond to the **produced** quality. I might describe my take on the same diagram by starting a spiral of production from the centre of the diagram, and working to-



wards the outer points from there.

PICTURE 5. This is how I think that the production could be balanced in comparison to the Picture 4 diagram.

3.2.2 Risk management and control

There is no production that should go from pre-production to production (actually, even before pre-production there should be one!) without mapping the possible risks. This is crucial in software productions, and even more so with game productions.

The single biggest risk for any game production today is technology. Today’s AAA-games (the biggest budgets and productions) could be described as the representations of the current hardware evolution. Games that show us what can be accomplished when we combine the calculative powers of cutting edge technology together with real-time rendering powers of the video cards and pack it into a software. This is not entirely true, and this is one of the biggest things that every producer must be aware of.

First of all, the hardware changes are so rapid (and I mean now only strictly the hardware, not e.g. the current public demand for any given time (such as mobile / table business today), which will spice the soup up even more) that the games of today are actually based on a few years older technology than the date that they are published. When starting the whole production you need pretty strict guidelines at what kind of environment you are actually launching it. You could make a guess at the start of 2014 on what the common hardware will be in the launch year of 2016, but the risks are great here. Many big companies do have a very open relationship to many hardware manufacturers because of this. This is something that has been a common risk with PC-games since the start in 1990´s, but now the mobile market is transforming into the same. The tablets and phones are at least 100 times more powerful in 2013 that they were in 2008, and they will most likely continue to evolve at the same rate for some time. TV is the second one that is as big a question mark as the tablets were five years ago. The streaming cloud services could open up the gaming to be only controller dependent, as there might be a possibility that no hardware is actually needed.

Some ideas there might seem a bit far-fetched, but not very far. That is just one aspect of managing risks! If you start building a game now that will be in production for a year, and it has a planned patching and updating schedule reaching to 3 years (provided the game is profitable enough), you could be facing a situation where a year after the launch, the TV-gaming revolution goes through the roof, and your game is just now reaching the peak of its lifetime. How do you adapt? At least 2 more years of the games lifetime is going down the drain (with the dollars), as people would like to play your game, but not with the “older” technology. Of course I don’t know the answer to that, but I would say that it was a very wise decision to keep that possibility in mind when planning the production. At least you now have something to look into if needed.

The other factors on the risk management and assessment list are pretty straightforward: how many sick leaves simultaneously will start affecting the production, or how many different backup scenarios are used to secure all data. I keep in mind a very good advice that I had during my education in risk management; every production will face at least one catastrophe. The fact that you missed to write it on your risk management document will not make it any less relevant. Every possibility should be written down, so that when something comes up (and it will!) your whole team can look into it and know what to do. Losing the beta-build and the backup files will halt your production for sure. For how long? That might depend on if the risk management document had "…and do at least 2 daily backups of everything. One to server 1 and the second to DVD that goes to John Doe´s safe."

3.3 Production

During the actual production stage, the producer can switch from planning and scheduling into a more interactive figure in the project as different teams start their work on various things, and everything concrete that is actually needed in the game will be developed. Graphics, audio and codes are implemented now, with vigorous testing after each part is completed. Producers usually have an artistic production and game designing as their secondary roles in the development process. This stage sees a switch in their role as the planning, hiring and scheduling are transformed into team communications, reacting to scheduling or budgeting changes and into an overall project handyman tasks.

Unger and Novak define the key goal of production phase at page 176 in *Game development essentials: Mobile game development* as having the game up and running as fast as possible is essential to solve unseen issues. Having working builds out every week to two weeks and iterating upon them is the way to go. I wholeheartedly agree that this is the way to build modern day games and this is where the producer is still a very viable, though not so dominating part of the whole production.

I have found that even as critical as the pre-production stage is, and how much of the project is defined there, the best producers work really intensively during the production stages too. This in my opinion can lift the project to another level of quality; from being good to being awesome. I myself have seen such changes during productions, where the producer's presence gave the teams a positive morale boost. Also the fact that there is someone that you know you can contact and ask for guidance at any time during the production is a massive boost to the whole project. I would compare this work-wise to the similar situations as medical surgeons and doctors might have. You are constantly on alert and ready to work if you get a call. Producers who spend their projects production stage absent from the actual teams have nothing to gain but simultaneously have a lot to lose for it. This is actually the best time for the producer to actually get to know the people who are working on the project, and usually these same people work for other projects in the future too, so forming basic social relations is also a big motivating factor. Modern software development doesn't cope too well with "faceless" producers.

3.4 Post-production

Most of the production work is done at this stage, and the post-production stages will see a lot of the work that is the most visible for the customers / fans. The on-line communities are established, the posters and game case arts are revealed and the cinematic trailers and teasers are being released. If the publishing date has been on an accuracy of a month, the actual date is revealed that the game will be available.

One big change that the gaming industry is currently going through is the localization of the games. Post-production stage will see these implemented in the game also. Usually this means tampering with texts in the art or in the game, but audio dubbing is a lot bigger process, and this would actually have been completed during the production stage and would be finalized during this stage.

For the development teams this is a fun, but crucial part, where everyone can now actually see where they have put their working hours for the past weeks/months/years. The 99% completed game is revealed for them and everyone can now actually play it. Not only the levels that your team was responsible, not only the packages with the assets that you were making, but the whole game that is coming alive through arts, animations, audio, modelling and coding. In some worst cases (as if your game is revealed to be under the quality that is acceptable) here is a point that will give you time to negotiate for a second point of no return (the first one being the point in production where you cannot turn and change any big things, and as time goes on, so does scale of the changes). Putting together a big team that might be able to change things for the launching patch of the game might be one option. But generally at this stage a win is a win and a loss is a loss.

3.5 Ending production to start it again

With this part I mean mostly about updating the game as in the modern mobile markets and adjusting it through the use of metrics (in short; a collection of data from each users device that holds information about various things that the customer has done, such as playing time, chosen difficulty, where the player dies most often, where they quit playing most often etc.).

Patching is also needed, as the modern schedules for game development are implemented so that a core team of coders make the last changes to the game during and after post-production. This can be anything from big packages that might correct massive errors, to a more common ones that hold only slight balancing changes to variable values in the code, or they speed up some parts of the multiplayer server codes etc.

Modern games might have an approach that there will be some dlc (downloadable content) material that will hold an update for the game as a playable bonus or as an addition to the main game, and this is something that needs careful planning throughout the development. Starting some parts of the production again might be really costly, such as hiring voice actors again, preparing motion capture studios for another recording process, etc. If this has been a planned strategy and an anticipated one during the pre-production stage, all the crucial and costly material should be done during the main games production. If this is not the case, then it generally needs a development process that is comparable to the actual main game, and is another process to start with, only this time you have a lot more assets and material to work with, but you will probably have a lot smaller team to work with it too!

4 CASE: UNDER – CHAPTER 1 – DARK VOID

4.1 Preface for case Under

Under is a PC-game, made with Unity 3D game-engine as a case study for this thesis. The game is a first-person adventure/survival/horror, which will be completed and released during my work on this thesis. The team working on the project is called Untrained Monkeys, and it consists of 5 people, myself included. The team is roughly divided in the main areas of game-making so that we have 2 programmers, 1 3D-modeller, 1 tester, and me. I am doing everything from scripting, modelling, rigging, animating, texturing, audio producing, composing and game design to testing. On top of that I am the lead producer and community manager for the games webpage and Twitter account. In this thesis, I am focusing only on the producing side of the project, and I will use it as an example case of my production methods.

4.1.1 Case Under. Before pre-production

Lining out the games concept was done in a few meetings. We developed some basic mechanics, which we would like our game to have. We talked about the settings, storyline and graphical style that we would strive towards to. All in all, after a few meetings and a week of pretty good detailing, we had a solid idea and a unified vision of what we would do, with what game engine and with what schedule. We also decided on the different posts that we would occupy for the rest of the process. My work as the lead producer begun then.

First I needed to think about the scheduling, costs, team management, communication and the engine we were using. I had already a pretty good knowledge about working with Unity 3D, but the rest of the team had no experience with it. In fact, they had no experience with any game engine and none of them had ever even developed a game for any platform. I decided then, that education should be my starting point for this project, and educating the use of Unity 3D proved pretty easy. We got a lot of material to study from, and I held a few training sessions where I also talked about some basics of game developing in general, and what we would need to accomplish before actually going into pre-production phase of the project.

I took it as my responsibility to create the base documents from schedules to high-concept, and finally the actual game-design document itself. The team provided most of the input to these documents in two meetings, but the basic outlines for what they would hold were my first task.

Financially it was clear from the start that this project would be done in zero budget, so that was really the first point that I was working on when scheduling. I also had to make it very clear for the whole team that due to this, the production would have to be vigorous, and all the more complex ideas would need to be straight lined or cast aside.

4.1.2 Case Under. Analysis of the start:

The start of the whole process was not an optimal one, as I was the only member who knew what it would take to actually make a 3D first-person game with all the mechanics we were planning for. Fortunately the team was enthusiastic enough to learn a lot of new material and techniques in a relatively short time span.

One month was used for education and learning of the basics about Unity´s tools, but obviously it was not nearly enough time to learn every aspect of what we would need. Still, I trusted that the people’s enthusiasm would provide for their lack of knowledge. What I didn’t expect, was the amount of work I would put myself through in order to get the team ready for the pre-production phase. This would backfire in the production-phase later on, and in the finalizing stages of the mechanics scripting. Documentation for the game was also a critical part to get things on the right track from the start.

5 CASE UNDER. PRE-PRODUCTION

The project started with mapping out and sorting the key mechanics that we wanted to polish and create for testing before the actual production would start. This way we could see what elements were working, which ones were missing, and also we could see how our working flow would turn out for the rest of the project.

Biggest thing that the pre-production planning needed to sort out was the fact that the team was working remotely. Everyone besides me had already full time jobs to attend to as a priority, so that would prove a massive challenge to deal with. It would affect the whole production aspect from scheduling to finishing, and it would have to be executed in such a way that the software builds and versions would not overlap each other and the versions would stay coherent from build to build.

5.1 Scheduling the project

Sliger and Broderick describe time management in *The software project manager´s bridge to agility* on page 85 as "… a negotiation between the customer (or the business) and the team from one iteration to the next, based on product feedback, changing market conditions, and insight into how rapidly a team can create working features (known as "velocity").". What a great summary of main things related to scheduling.

First priority was to manage an overall schedule that we would be working with. The start of the pre-production to the release and patching of the final game. I made an excel sheet for this purpose, that was mapped roughly on the date where we planned the initial release. The map was divided into a weekly schedule where the main things that needed to be finished at each stage were mapped out. This rough schedule showed me straight away some points that were probably going to need more attention than others, such as texturing and modelling, as we had no real expertise in those areas, apart from myself.

This schedule was then divided into sections of production cycles, meaning that in the end of each cycle (a two week or three week period) we would have a build that included all the things that were planned for those weeks. Also I made sure from the start that the first production cycle had the tightest schedule planned, whereas the next ones would each have a little more space in them, just to make up for upcoming errors, problems or late deliveries. This was based only on my feeling that we were most likely doomed to see those problems during the production, and now I made sure that we had some time each cycle to see that those things would get fixed before we went too far ahead in the production. If we would not get any errors during a cycle, great! The team could then take a little leisure from the project, as it would prove a hard one to do in addition to day jobs anyway.

Some things proved to be a massive undertaking to map in terms of scheduling. The amount of time to allocate on things that the team had had no experience with was mostly based on my past experiences working with student projects, and then adding a little extra to see that it would be enough.

The overall scheduling was then distributed to all the team members, with a more detailed personal schedule included for each team member. This way they could really delve deeper into the current production cycle´s needs, and they could communicate with me if there was any need to change them.

5.1.1 Personal timetables

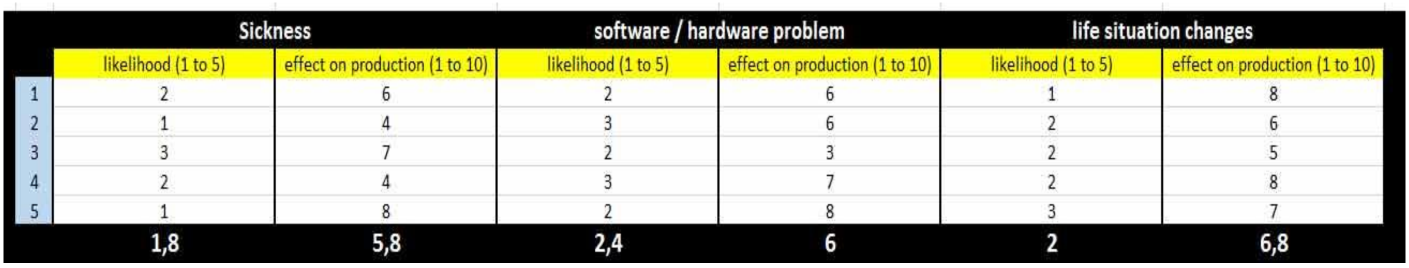
I had a small amount of experience into a technique called Kanban, which is a production method used in Just-In-Time (JIT). This is a card based methodology where the current task is the represented in a card, and this is moved on a table divided into different stages of completion. The stages can vary, but mainly there’s at least 3 different stages of “waiting”, “in work” and “completed”. Main thing here is that the board or wall is visible to the whole team, so that production advances and stalls can be tracked quite easily by all team members. I used this type of scheduling methodology to make the personal timetables for each team member. Everyone could then track what others were currently doing, and as the production was done remotely, this was the most fluent way of keeping the team members communicating and on track for the duration of the production. This also took a lot of work off of myself, as the need to communicate through me was reduced to a minimum.

5.2 Risk mapping and management

Some of the risks were pretty obvious from the start. We had a team with no real experience in game making, working hours would have to be evaluated again and again after each iteration cycle, and some things like animation, modelling and texturing required learning as we worked. On top of that we knew that we couldn’t afford to lose too much time on any of those risks.

Finding out a perfect solution to manage the risks was not an easy task. To back the production up, I decided to do a few versions of our original timetables. One was a version where we would suffer a one to two week loss due to the fore mentioned risks. Other would be a “disaster” model, where things would fall back for four weeks. At least the production would not stop completely at any point, and there would always be at least some kind of plan to fall back to.

I would then use a very basic Excel sheet to pass around the team and let them evaluate the basic risks that every software production has. Each risk would be given a score between 1 to 5 for likelihood, and 1 to 10 for the affect that it would have on the production. I then calculated averages for each basic risk to see where we would be at each case.



	Sickness		software / hardware problem		life situation changes	
	likelihood (1 to 5)	effect on production (1 to 10)	likelihood (1 to 5)	effect on production (1 to 10)	likelihood (1 to 5)	effect on production (1 to 10)
1	2	6	2	6	1	8
2	1	4	3	6	2	6
3	3	7	2	3	2	5
4	2	4	3	7	2	8
5	1	8	2	8	3	7
	1,8	5,8	2,4	6	2	6,8

PICTURE 6. This is the basic risks table from Excel that I used to get some clues that what we might be expecting during production.

5.3 Creating the game design document

Doing a good game design for a project where the team would work remotely for the whole production was probably the most crucial document in the whole pre-production. This is not related to producers work in most cases, but there are still many cases where the producer works tightly with the game designer for this document. In this case, I was responsible for both, and thus it seems proper to be described here, but mostly from a production point of view.

The design document would have to be a little more thorough than in cases where the team would be working together in a same space. To ease the production, some mechanics and their execution were really detailed. Also, to ease the producers work in the middle of the production, it would be essential for the team members to understand exactly what was needed for the game.

As a producer, I found it really helpful and liberating to have a really thoroughly detailed design document available for the production cycles. As we were iterating things in each production cycle, the design document would see some changes here and there, but the amount of work to update it was really minimal. Also the main points about our actual release, such as game resolution, platform or controls were there for the whole team to see whenever they were working on their personal parts and build of the game.

5.4 Creating the software base with Unity 3D

I decided early on that we would start building our games base by coding as much of the basics as possible as early on in the project as possible. Both, to ease our actual production and to develop the skills necessary for our coders to work more efficiently.

We approached the base with some tricks that helped our workflow in Unity 3D. As a free version platform, it has numerous tedious flaws that can really ruin a bigger production where many codes and code parts change frequently. We would use a few 3rd party file-comparing tools to check our updated files and edit them if necessary. Also the hierarchy for actual editor data was agreed to have certain name extensions that would separate our base objects from dynamical ones.

Building up the base was part of our pre-production from the start to the very finish. It proved enormously helpful, and it also gave the team a deeper and more thorough view of the whole project for the actual production phase. Designing the way that our game would be managed together with the actual controls and physics set a good and inspiring tone for the rest of the production.

6 CASE UNDER. PRODUCTION

As the preparations were underway and the actual production started, I took it as my responsibility as the producer to start building material and preparations for the actual launch of the game and the whole post-production phase. This is how I wanted to approach the whole production as a producer: being one step (or at least half a step) further than the rest of the team.

During the production, however, there were a lot of things that would begin to hinder our progress. The team's inexperience would inevitably start to show, and I was multitasking in way too many areas the project. There was no real solution that could let us somehow pass these hindrances, so after a few cycles, we decided to add 3 weeks to our production time.

Otherwise we were proceeding to our plans, and were actually getting towards our goals now more realistically. Testing all the iteration builds of the game proved really useful, and we would manage to polish some mechanics that were reporting flaws.

Managing the team during the production was really difficult. Despite our good game design document and reports, we still had the disadvantage of not being able to discuss many of the productions main difficulties face-to-face. The meetings between our iterative production cycles weren't as constructive as planned, plus they weren't nearly as long as they should have been.

However, as we got to the latter stages of the now revised production phase, we did pick up the pace (quite usual in these type of productions, actually) and manage to have the build readied for the release date.

7 CASE UNDER. POST PRODUCTION

At the start of the whole production, some main parts were agreed upon that would define the way that the post-production was supposed to be handled.

The publishing of the finished game as a WebPlayer version from Unity free would need a webpage that the team didn’t yet have. This lead to whole new area of actually establishing a web-page for the team itself, and publishing the game under that domain. This would also allow the team to build more projects and continue the Under episodes in the future with a more free-styled approach.

The team also wanted visibility in the Twitter, so an account was established to inform the world about the coming project, its progress and the team behind it. The tweets would also give the team a nice look-back to the whole production after we were finished.

The project builds and files were not closed, so that future updates or the planned possible sequels could utilize everything we already had made. Also publishing the game on a different platform in the future could be done with more ease in that way.

The game was finally uploaded to the teams newly established webpage of http://www.untrainedmonkeys.com to an address under that at http://www.untrainedmonkeys.com/under.html

8 MAIN PROBLEMS IN THE DEVELOPMENT

Productions of these type usually have a long list of problems after the dust of the whole process has landed, and we were no exception to this. However, I’m not going to list any detailed bug-lists of the software or anything really specific, but point out really the bigger picture of problems that we faced in our production.

First problem that we had (and this was even taken in some ways into account before starting) would be the inexperience. The challenge that the team had in making a game of this scale with as little experience as we had, was massive. It would also escalate into many of the bigger problems we had. The inexperience would show in many areas of the actual production, which forced for some massive parts of the game to totally differ from what was planned. Also the inexperience and difficulties of remotely communicating proved hard, as the whole team would have to learn to communicate with each other in a relatively small timespan as efficiently as possible. This slowed certain designs and parts really too much. One might ask that why would we even begin a project of this magnitude with a team that has so little experience? The goal was to make a proof of concept about a bigger picture concerning the establishment of a working game studio.

Second major problem that we faced was struck by Murphy´s law in our risk management tables. Even though the risks were really thoroughly mapped and all of our production plans had a lot of time to prepare for all kinds of mishaps, it just couldn’t cope with illnesses that would happen almost simultaneously, to almost to the whole team. This was really a second out of control situation that just pushed the schedules further than was planned. Production would suffer from this, but the actual work was picked up really fast and efficiently as the team got better.

A more small scaled but nevertheless great hindrance was our lack of documentation in some parts of the production. We didn’t have a tightly documented tables for our sound designs, models, animations or art. In a different working environment with more social aspects available, this would have been fixed during the production. Now it left our production really vulnerable to any major changes that the fore mentioned tasks would need. This really pushed my own work as a producer to come up with the documents in the mid-production. This also meant that I had to look at the already finished parts really thoroughly, imagine what they might look like a few weeks ahead and start implementing things that would see the mechanics fit in their place. Not a very effective way to get the job done, but it did get the parts in the game.

9 ANALYSIS

9.1 Looking back

As a whole, the project was really tough and the game itself proved really difficult to produce. Because I had so many different parts in the whole production, I cannot say that I'm extremely pleased with results, but neither am I in any ways disappointed by them. The published final version is stable and working, with most of the intended mechanics really well implemented. Graphically we were already at the beginning way over our heads, but managed to put up a quite beautiful underwater environment. I would like to think that overall my work as a producer was really good and efficient, with no major flaws that could have seen the whole project collapse.

At the start of the project I knew what the team was after, what would be required to get to that point and at what cost. Probably because I knew a lot of aspects about the production, even before it had started, I was able to map out the plans and designs really quickly and make them as thorough as possible to compensate for the lack of experience in the team. I knew that the documenting was crucial, but couldn't have ever anticipated that it became the production area that would prove the most lacking. As much as I tried to have the documents match the needs of the team and the project, they weren't quite up to the task of dominating the whole production.

Now looking back at it, our production method was more about iteration, and the strict documenting policy was somewhat against it. By trying to fit these two methods together, it became a little unclear at some point to see if the project was done like documents stated, or like it was iterated at each cycle. Frequently changing the design document wasn't anything unexpected, but starting the project strictly by it, probably lead to more problems that gains.

9.2 Funding and how it would have affected this project

The whole production of Under was done with a zero-budget. We had no investments or funding, and this was the single biggest thing that affected our whole project from day one. This did enable our team a little more freedom, and we did use it to our advantage as much as possible. But there were also some really obvious things that were sacrificed for those gains.

First of all, I think that our game had very good base built at the end of the pre-production phase. What we could have gained through funding, would have really helped to elevate the games overall production values in graphics, sounds, 3D-models and animations. These are the usual things that get outsourced to companies that have expertise in these fields. Also our tools with Unity 3D free version were getting more limited the further the production went. A paid license would have helped with that too.

To list a few examples where the game suffers from the lack of funding:

- Graphically the UI could have used a more professional touch, and the overall textures might have needed a little more polish.
- Sounds and music could have been produced with a little more time and they might have been mixed better.
- Some models and animations would have definitely gained from a more professional touch.
- Developing the game for different platforms could have been possible.

Taking all the fore mentioned things into account and judging the end-product from that viewpoint, the game did get finished with good balance and execution. The lack of polish is not as big as it might have been. Finally, as a producer I am very pleased that a new and inexperienced team was able to accomplish such a complex game in a relatively small time frame with my guidance.

9.3 Establishing the requirements of a thesis with case Under

I believe that this thesis provides a good basic knowledge about one type of production management, and in my opinion these fundamentals can endure time even in this rapidly changing field of game making. The core knowledge that I have written here comes from a combination of experience, theory and practice. The accomplishment of finishing the game is a good proof of concept for many production workflows I have presented here. Also I would like to point out that as a learning experience the production and the writing of this thesis were invaluable learning lessons, which have again elevated my knowledge of producers work to a new level. I also believe that a starting generation of game producers can find an insight from my work, and that this thesis will help them develop their own methods, and also refine the ones presented here.

10 REFERENCES

Bethke, Erik. 2003. Game Development and production. Wordware publishing Inc. USA.

Gates, Richard. 1992. Production management for film and video. 2nd edition. Reed educational and Professional Publishing ltd. Great Britain.

Sliger, Michele & Broderick, Stacia. 2008. The software project manager´s bridge to agility. Pearson education Inc. USA.

Unger, Kimberly & Novak, Jeannie. 2012. Game development essentials: Mobile game development, international edit. Canada

Under – Dark Void – Chapter 1. 2013. Untrained Monkeys. Finland. http://www.untrainedmonkeys.com/under.html