

KARELIA UNIVERSITY OF APPLIED SCIENCES

Business Information Technology, Game Programming

Antti Kovanto

The Improvements for Indie Game Development

Bachelor's Thesis

DATE 2013



THESIS
January 2013
Degree Programme in Business Information Technology
Karjalankatu 3, 80200 Joensuu
FINLAND
Tel. +358- 50 311 6310

Author(s)

Antti Kovanto

Title

The Improvements for Indie Game Development

Abstract

The goal of my research was to combine the vast but defragmented knowledge sources related to Indie Game Development to find a commonly preferred approach.

Insight information was gathered from Game project postmortems to find specific information on the problems and successes occurring in the projects. From these results an issues of major concern was selected and improvements were suggested for them in discussion chapter.

Later in the research I formed a focused guideline based on the information gathered and results founded. The guideline was formed so it is easy to practice in smaller Indie projects, without massive efforts. The focused guideline aims to improve the areas which Indies lack the most, without being too heavy, authoritative, strict or inflexible.

Indie Development problems and successes were also discussed from an ethical and moral perspective as it was important for this research to reflect Indie Development also to Indie ethics involved in Indie lifestyle and culture.

The research and its results were connected in practice by a real-life Indie game development. Together with my team we created a game prototype according to my game design and after finishing the project I wrote a postmortem reflecting on many issues pondered in the actual research.

Language
English

Pages 80
Appendices 5

Keywords

indie, games, postmortem, comparative

Table of Contents

Acknowledgements.....	1
1. Introduction	2
2. Literature review	3
2.1 The brief history of indie game development.....	3
2.2 Definition of 'indie'	4
2.2.1 Indie as a lifestyle	4
2.2.2 Apart from mainstream	5
2.2.3 Active indie community	6
2.2.4 Ethics and values.....	7
2.3 Creativity emphasized	10
2.3.1 Out-of-the-box thinking	10
2.3.2 Creative freedom	12
2.3.3 You make your own rules	13
2.3.4 Being creative on marketing	13
2.4 Project management and agile development process.....	17
2.4.1 Prototyping	17
2.4.2 Milestones	18
2.4.3 Development process models.....	19
2.4.4 Team commitment and motivators	20
2.4.5 Multitalented developers	21
2.4.6 Outsourcing, hiring specific talents	22
2.4.7 Office working vs. remote working.....	22
2.4.8 Team work.....	26

2.4.9	Quality assurance	27
2.5	Technology	29
2.5.1	Free or affordable tools.....	29
2.5.2	Game engines	29
2.5.3	Version controlling.....	30
2.5.4	Bug trackers	31
3.	Methodology	31
3.1	Purpose of research.....	32
3.2	Procedure	33
3.3	Measure	35
4.	Results	37
4.1	Project area management results	37
4.2	Business area results	40
4.3	Development area results	41
4.4	Design area results.....	44
4.5	Problems in indie and AAA game projects	46
4.6	Successes in indie and AAA game projects.....	47
4.7	Problems found in problem type groups	49
4.8	Successes found in success type groups.....	52
5.	Discussion	54
5.1	Thoughts on indie culture, lifestyle and ethics	55
5.2	Thoughts on designing and creative freedom.....	56
5.3	Thoughts on indie business	58
5.4	Thoughts on development.....	59

5.5	Thoughts on project management	61
5.6	Thoughts on comparative analysis results.....	63
5.7	Improvement ideas and suggestions for indie projects	65
5.7.1	Scheduling was done poorly.....	65
5.7.2	Finding new employees	68
5.7.3	Launch and Post launch problems	70
5.7.4	Team work success	72
5.7.5	Focused vision	75
5.7.6	Correct design decisions	76
6.	Conclusion	79
	References.....	81
	APPENDIX I (Postmortems list and data grouping table)	89
	APPENDIX II (Focused Guideline for smaller indie projects)	94
	APPENDIX III (“Monx in Mandala” prototype’s decisions)	96
	APPENDIX IV (“Monx in Mandala” postmortem)	98
	APPENDIX V Glossary	100

Acknowledgements

I am eternally thankful to my Responsible Teacher Seppo Nevalainen at Karelia University of Applied Sciences for guidance. Without his support during my studies none of this would exist now.

I also want to give special shouts for my family and especially to my wife Nonmanut Pongsakdi who gave me invaluable insight into academic writing and Tuomas Kärkkäinen who has given me alternative views on many topics connected to the subject. Tuomas has also made a gigantic effort in proofreading and polishing the language.

Lastly, I want give thanks to God for giving me strength and ideas for this project as well as the vast amount of people who supported me.

1. Introduction

I started this research because there were only a handful of sizable studies made on the subject and most of the knowledge available is scattered in small parts. This scattered knowledge makes it difficult for starting Indies¹ to have a focused overall vision on different areas related to Indie Game Development.

This research was also needed to find improvement ideas on some of the problem types that occur in Indie Game² Projects. On a personal level doing this research felt like a natural way to help the Indie community and to help Indie ethics and virtues flourish.

The definition of Indie is observed from multiple perspectives and the history of Indie Game Development is briefly explained. During this thesis I try to open up the whole Indie concept and to continuously improve the awareness of the deeper meaning of “Indie” for the reader. A structured concept of Indie Game Development is formed from out-of-the-box thinking, creative freedom, team work, ethics, values and creative marketing.

With the information and data available, it was possible to find areas of concern and provide useful insights to many sides of Indie Development and life. An ethical perspective was kept as an important part of this research and it was connected firmly to different topics throughout this whole thesis.

Many technical aspects are also presented, but as this thesis' main focus was not on detailed technical information, the details of this information were kept abstract enough even for non-technical readers.

Focusing more heavily on Project Management, Development Process Methods and different Design issues, multiple problem type groups were investigated and a valid set of improvement suggestions were made for each of them. Later, according to these suggested improvements and the other information gathered, a focused guideline was created to support smaller Indie companies.

In the discussion part I share the most important thoughts I had on the research results and Indie Game Development and lifestyle as a whole. I sincerely wish that information provided in this research and my thoughts on the subject will help people deepen their connection to each other as a community or as a team and re-evaluate the ethical aspects of their life if needed.

In the Appendix section, I have included our prototype³ project's postmortem⁴ where I analyze critically our team development process and different actions taken during the project.

2. Literature review

In this chapter my main focus was the many requirements for successful Indie Game Project. As the scope was rather large, I focused on Creativity, Project Management and Development Process.

During my research I quickly noticed that no “silver bullet” solution exists in Indie Game Development, but rather there is a large pool of fragmented “common wisdom” that could be gathered and formalized. Finding and collecting this data was easy due to the fact that Indie ethics supports free sharing of knowledge.

2.1. The brief history of indie game development

When tracing roots back in history to the point where Indie Game Development begun we cannot deny that it has been there already since the beginning. The very first computer games were actually indie games.

When tracing roots back in history to the point where Indie Game Development began we cannot deny that it has been there already since the beginning. The very first computer games were actually indie games^{1 2}.

However it is said that Indie Game Development was officially born in the early 1990's when PC gaming was growing rapidly and creating video games became profitable for the PC. Indie developers started to make games, but reaching a certain level of quality was difficult because the Internet as a source of free information had not yet completely developed. In 2011 the indie game industry had finally started to mature, thanks to new opportunities created by the iOS App Store, Xbox XBLM, PSP minis, and the PlayStation Network (PSN)^{3 4 5}.

2.2. Definition of 'indie'

2.2.1. Indie as a lifestyle

According to the dictionary, 'indie' is an informal version of the word independent.

However, the word means more and it is connected to certain attitudes. Generally, it symbolizes originality and forward-thinking, especially in music and design. The name 'indie' is commonly used for a person who thinks, and cares about issues one wouldn't find in the mind of most adolescents.

An indie is any business, developer or designer that is not associated with a large corporation, especially a global one. A consumer who chooses to support small businesses, independent record labels, and handmade items rather than shopping at big-box stores can also be called indie⁶.

Robert Boyd, co-founder of the two-man studio behind breakout XBLIG hits *Breath of Death VII* and *Cthulhu Saves the World* states that: “ An indie developer is an individual or small group that is not owned by another company that makes games, an indie game is a game made by an indie developer”

K. Santiago, co-founder of TGC says: “Indie is when you are innovating in some way on how games are made”. According to Santiago this can either be innovation from the creative or the business perspective.

A. Saltsman, the one-man indie developer behind the genre-defining game *Canabalt*, gives a personal definition:

“An Indie game is a project that is the sole product of a passionate creator, and one that’s unfettered by outside forces”⁷.

And David Rosen clarifies Indies to have these two key characteristics:

1. Motivated by passion⁵, not money - Money is always a factor, but for indie developers it's an incidental logistical concern (i.e. the project can't continue if we starve to death), not the primary goal.
2. Designed from the trenches - The developers in charge of the project's direction are also the ones doing the grunt work, such as programming and creating artwork⁸.

Later in this thesis, when talking about 'indies', we are concerned only with the specific group of independent game developers. In this thesis the term has a much smaller scope of people than usually.

2.2.2. Apart from mainstream

Indie Game Developers avoid big investors, taking big loans and following the beaten path of mainstream game development. They want to make their own rules without listening to commandments from above, neither do they want to follow trends.

Some might even dare to predict the “Renaissance” of computer gaming in the future caused by Indie Game Developers. A certain rebellious “guerilla” attitude against mainstream game business is popular among Indies.

The idea of every human being’s birthright to live their life how they want without depending on mass media propaganda or capitalistic dogmas are highly emphasized among 'indies'. Human dignity is raised above materialistic gain. This is not how mainstream culture usually works.

However, the mainstream game industry still wants to be part of this current indie hype. Electronic Arts (EA) has released its own Indie Bundle⁹. Over the years EA has become infamous of being the polar opposite of the indie “scene”. EA has historically followed very questionable ways of taking care of their employees. Many sources insist that EA is actually ripping off the developers and taking huge advantage of them through inhumane crunches⁶ and other ethically dubious working policies^{10 11 12}. Actions like EA’s Indie Bundle make the thin line between mainstream and indie more obscure. When people became more liberal on what they accept as “indie”, the word indie itself began to lose its meaning. This same progression has happened earlier in other forms of indie business, such as indie music¹³. It is modern colonization, where instead of cannons and rifles they use more sophisticated ways to conquer.

2.2.3. Active indie community

Indie Game Developer’s Community (or Indie Game Scene) is based on the Internet, organized as web pages, portals, blogs, Twitter, forums and IRC channels. The community also embraces community events and so called developer’s nights. These events and “nights” can vary from a few developers drinking beer in a bar to huge professional gatherings of Indie developers.

Indie Game Developer’s Community is usually closely connected to “Demoscene⁷” and Open-Source⁸ communities. These three factions work in supportive symbiosis⁹ providing each other the best technology possible with dignity. As a matter of speaking new technical innovations in graphics programming and related fields, often originate from the “Demoscene”¹⁴. This technical knowledge eventually reaches mainstream game development and I agree that it is true:

“They’re supposed to be modern. They’re eye-catchers. We’re talking about games. The explosion of graphics in games is getting bigger and bigger. Games such as Crysis, Mafia 2, Far Cry 2, or the Nintendo DS title Nanostray 2, play in their own league on their respective platform. Responsible for their visually impressive

appearance, are people of the “Demoscene”. Those people have talent, a good grasp of design and visuals, as well as the required expert knowledge”¹⁵.

Meanwhile, the Open-Source community is doing important work creating inexpensive choices of tools and software for developers. These factions naturally provide important support and possibilities for the Indie Game Developer’s Community.

The existing community of people who want to see the scene flourish is an enormous advantage of being an indie game developer. For Indie developers it is highly beneficial to keep close relationships with the local community and to create a firm fan base¹⁰ for their company. Looking down on your fans’ opinions or wishes is the worst kind of mistake you can do. An Indie Game Developer has to understand the fact that fans do invaluable marketing, promoting and numerous other efforts for free so you can make your living. Moreover they will always buy your games. Never underestimate the power of real fans¹⁶.

Indie developers benefit greatly from the support of their community in the form of feedback, word-of-mouth¹¹ marketing, fan bases, game play testing, reviews, and finding new developers. Naturally, Indies should always keep their community active and recognize their fans; it is still important to acknowledge that organizing community can be a difficult challenge¹⁷.

It is also worth mentioning that an Annual Independent Game Festival (IGF) and Indiecade, an International Festival for Independent Games are organized every year. They are great showdowns for any Indie Game Studio^{18 19}.

2.2.4. Ethics and values

Indie Game Developers borrow many of their ethics and values from Open-Source Development and further from Hacker ethics.

“The Belief that information-sharing is a powerful positive good, and that it is an ethical duty of hackers to share their expertise by

writing open-source and facilitating access to information and to computing resources wherever possible”²⁰.

This radiates inside the Indie Game scene in ways of active knowledge sharing communities, Open-Source friendly attitudes and a certain level of transparency between fans, customers and developers, with the help of blogs and other social network services.

Prof. Himanen observes hacker ethics from a different perspective. In his approach hacker work ethic is an opposing force to our society’s current Protestant work ethic²¹.

According to Prof. Himanen, hacker work ethic is becoming more important among information professionals. Instead of Protestant work ethic features and values: money, work, optimality, flexibility, stability, goal orientation and result accountability, hacker ethics appreciate 7 different values: passion, freedom, social worth, openness, activity, caring and creativity.

“The Protestant work ethic, as every first-year sociology student knows, is what made western capitalism so powerful. When it comes to accumulating profit, what could be more perfect than hard work, self-denial and the threat of eternal damnation for the lazy?” wrote Oliver Burkeman in the Guardian newspaper.

The greatest problem in Protestant work ethics seems to be however the built-in idea of guilt of being lazy and worthless. This supports a puritan attitude which measures our responsibility as a worker, not on how much we really can achieve during a day but rather how hard we are being on ourselves and how guilty we constantly feel. Protestant ethics’ support of the sacrifice of mental, physical and spiritual health on the altar of capitalism is soon coming to its end as more humane ideas win ground.

Schedule more free-time activities you truly enjoy: weaken the Puritan link between "working hard" and "making life awful", and the work part gets easier,

too. The Puritans, Pallotta concedes, "had a strong work ethic. [But] they also burned witches at the stake... We need new role models"²².

According to Prof. Himanen, most hackers do not see money as value in itself and are willing to create something valuable to the community. Serious Indie Game Developer often shares similar thoughts and agree with the values that Prof. Himanen declares. We can reasonably state that Indie Game developers aim to reach higher virtues in their lives than Protestant work ethic can provide²³.

As studied in the 1990s, decadence is affecting Hacker ethics, which naturally radiates throughout the Open-Source community also into indie culture²⁴. Ethics and values have to be reconsidered and 'indies' need to overcome the constant pressure coming from modern society and the mainstream business world. If 'indies' cannot stay true to their vision and principles, the whole culture might be endangered and the self-humiliating practices described above will gain footing.

There has been currently also an online discussion around sexual harassment in Game Companies. This kind of harassment is one of the many negative effects caused by a twisted attitude towards employees. Company CEOs who feel a wrong kind of ownership over their employees can go even this far in humiliating practices and the results are often disastrous^{25 26}.

During my research on AAA₁₂ postmortems I found out how Protestant work ethic has created inhuman Project management principles. In order to maximize optimality and fast results, mainstream companies have started supporting humiliating practices on their employees such as harsh "result evaluations".

The idea of these practices is to reward employees with the highest achievement and to humiliate the employees who have achieved least. Evaluation is done on a weekly basis and "improvement needed" scores are given to the lowest achievers. The irrational fact is that even a team doing a supreme job or one improving its efficiency and quality constantly; always must give some members the rank of "improvement needed". A punishment is also given to employees who receive this "rank" multiple times²⁷.

The original idea of result evaluations is to make employees pressured to improve their work and avoid being ranked. This working culture is often called a cannibalistic working culture and has been harshly criticized for being a hazardous strategy to any company practicing it. The reasons are obvious: instead of respecting your employees and their work efforts by giving them peace and security which they need to be creative; a constant battle field is created inside the company. Rather than making efforts to beat competitors, the employees are forced to compete with each other. Needless to say, this kind of setting quickly affects the production and turns against itself²⁸.

2.3. Creativity emphasized

2.3.1. Out-of-the-box thinking

When you're independent you have the ability to do something different. The best and most loved Indie games are usually the ones having a completely new experience to offer. This new experience is created by providing an interesting new concept and design.

For Indie developers it is important to be able to think out-of-the-box. As an Indie you question all old "traditional" solutions and instead of making a copy of an old idea make a game of your own. When you recognize your game as a form of art, this leap is easier to take²⁹.

This does not mean however that you could not make a game fitting in some existing genre. Instead, it means to create your game in a fresh and innovative manner. A game called Perspective is a perfect example of how out-of-the-box-thinking works. The game is a platformer game that has 2D and 3D working together in a really interesting way. Instead of running from left to right and jumping around, the player has to make her path with the help of both 2D and 3D dimensions. This adds a special puzzle element to the game³⁰.

In very many cases, Indie developers have to invent new kinds of game mechanics, just like in Perspective, because they cannot compete with other technical areas such as high-detail real-time rendered graphics. This necessity

among some other aspects related to game design is sometimes even referred to as a renaissance of game development. The core of the games was originally focused on new game mechanics and currently the trend seems to be that the game developers are returning to their roots.

Out-of-the-box-thinking can be applied also to creating designs. Indies should consider creating designs which are less time consuming to produce (small teams) but still impressive in their own unique way. Many Indies strive to create their own style and graphics that have a personal touch to them. Personal touch and vision are highly important in Indie development. These lead to new innovations that are often re-used in AAA game development later on, if proven to be profitable.

Furthermore, the fans need to know that there are real human beings behind the games they love. The fans need trustworthy and authentic Game Development Heroes to support and love. Instead of watching people who love what you're doing from your ivory tower, talk to them openly and use social media such as Twitter to share your life as a game developer with them. This more natural approach to dealing with your fans is often practiced among large mainstream game studios as well³¹.

The out-of-the-box approach can be expanded to all other development areas as well. Indies should use their imagination to create effective and low-cost marketing campaigns, ways to promote their game and ways to cut down development expenses. It is worth mentioning that one Indie game company made their whole project from a third-world country, so as to cut their living expenses to a bare minimum³².

Decisions like the one mentioned above are however, quite extreme and rare, but it is still a good example of out-of-the-box thinking. Even the craziest idea can prove to be perfect for the situation or requirements of the company.

2.3.2. Creative freedom

One of the main differences between Indie games and AAA games is creative freedom. Because huge amounts of money are usually not involved in Indie game development, developers tend to make more radical decisions in the areas of game design, graphics and sounds. Indie game developers are not obliged to milk the poor old cow forever unlike the AAA game development studios.

In AAA business where every decision is done to maximize profits, a lot of compromises are made and the same popular concepts are repeated over and over again. AAA studios usually design games to fit mainstream consumers' needs and once one game concept has sold millions it is repeated as long as it keeps selling. The result is that most of the games are more or less similar with little difference in graphics, mechanics or features. As the main focus is on making big profits, risks including creative freedom are avoided^{33 34}.

This creative "prison" can be completely avoided in Indie Game Development. In fact the best Indie games are those which break old concepts and boundaries; games that have truly invented something new and innovative. When it's not mandatory to include typical clichés in your designs and storylines, something more beautiful, vivid and meaningful can be created. Also due to Indies lack of manpower and resources, interesting innovations in terms of simpler designs and game mechanics are born³⁵.

Indies have control over their intellectual property and as there are neither investors nor publishers who are mandating guidelines and restrictions, a real creative freedom is possible; this is foundation for all gaming paradigm shifts and innovations^{3 4}.

"Indie games tend to be more personal," Terry Cavanagh argues, "Every small detail in a game gives you a sense of the person who made it, and that's just something you don't get in a game that's made by a big group of people"³⁶.

2.3.3. You make your own rules

As an Indie developer you are responsible only to yourself and your team. You have full control over your work practices, development processes and how you run your company.

This allows company employees to control their ways of working and adapting themselves when practices are not working well or do not feel correct. There is virtually no game design that you could not develop as an Indie, but it is still mandatory to keep your scope small enough and to avoid feature creep. Also it is important to do your work tasks actively and with high morale. As Friedrich Nietzsche states: “Freedom is the will to be responsible to ourselves”.

On the contrary, there is also the negative side of all this freedom given to Indies. Sometimes when things are scheduled loosely or not at all, launch deadlines can be postponed, possibly multiple times. This enormous freedom towards working can sometimes cause a lot of unwanted stress as the small Indie team tries to struggle without proper project management.

The rewarding factor of this freedom given is of course not only the freedom itself but also the feeling when you finish the game. On launch day you can say: “We did this and this game is our personal achievement. It is a projection of us”. Especially if the public receives the game positively and loves it, the reward is overwhelming^{4 37 38}.

2.3.4. Being creative on marketing

As an Indie you don't have millions to invest on marketing, it is mandatory to do it low-cost. There are many things that need to be considered when laying out the marketing plan for the company.

Whenever making a game for a certain platform you make a brief research on how the online market for that particular platform works efficiently. Certain

actions need to be avoided when using services such as App Store, Android Market or Nokia Store. This brief research will tell you how to use your sales place and helps avoid mistakes.

In case you need to do any form of traditional marketing you always need to have real business decisions behind them. If you don't carefully investigate your possibilities there is a great risk that you're throwing your money away on inefficient marketing methods.

When using Banner Advertisements, it's important to select the correct sites. Keep track of the efficacy of your marketing campaign. If some site is not bringing you any visitors or money, don't hesitate to drop that site from your Banner Ad site list. If you can do an animated ad, you should do it, as they catch the eye more effectively than static ads.

Reviews are crucial and it's important to have connections to the correct review sites and Press people. Naturally if the game gets good reviews and high ratings people notice it more easily and get interested.

Social marketing can be done with a minimal budget by using word-of-mouth marketing. Services such as Twitter, Facebook, blogs and forum threads are good tools to build awareness for the game. You should always investigate the correct tricks on the service that you use for marketing. Using hash tags, shout out thanks for re-tweets and tweet timing are basic knowledge that Indies should be aware of when using Twitter for example.

Also you should consider crowd funding possibilities. There are many great sites for getting crowd funding such as Kickstarter. Whenever you try to collect crowd funding remember to include enough content in the form of screenshots, concept art and a possible prototype of your game. Including content already in the early phase will improve the likelihood of reaching your goal. Also keeping your Kickstarter actively presented in social media is important³⁹.

A website is mandatory for an Indie Game company. The comforting news is that it does not need to be anything astonishing. These days a lot of Indies use simple WordPress blogs to promote their games. Make sure you have all the necessary links to the places that sell your game and your own unique domain name.

Prepare promotional material such as screenshots and a trailer already in the early stages. Especially with the trailer outsourced help comes in need, in case you don't have an AV-expert on your team. Trailers are a great way of building up hype before the actual launch. A Press Kit is also worth making beforehand. You can also make a so called Announcement Trailer to get attention for your project from the start.

Making Press Releases is an efficient way for getting the word out about what you're doing. It is a good idea to put out a Press Release for pretty much anything. Study about writing Press Releases from internet articles. It is recommended for Indies to write their own Press Releases, because you know best what you want to announce and you can save some money as well. However if English is not your first language or you're not really a good writer outsourcing Press Release writing might be worth the money.

Publishing Leaderboards is worth doing especially if your game is community-based or competitive. Rewarding top players on the Leaderboard, holding contests, announcing winners on your blog are good ways of getting extra visibility for your game.

Having good contacts in the Press and media is very important. Whenever possible try to get good contacts and get also mainstream publicity for your game company. When you're Launching, you can use your contacts to get the word out for your new game.

However, there are many sketchy and shady businesses related to reviewing and marketing, such as paid reviews, guarantees and micro services. It is mainly a developer's ethics that are deciding which amount of these services he

uses, if any. The bad side of using these kinds of dubious services is that you cannot anymore track really your own success and pinpointing your mistakes can be more painstaking⁴⁰.

Maintaining all of these marketing methods can be full-time work and while wearing many hats it can start to be painful. In case you have a small team, dividing tasks is of course the correct way and maybe you have to consider outsourcing some separate tasks⁴¹.

However, traditional marketing strategies can be ambiguous to some companies. According to Mountain Sheep's co-founder Jouni Mannonen at Assembly Summer 2012 ARTtech seminars, marketing should not be aggressive and pushy. Instead Mannonen empowers the importance of providing gamers those games they know people want and need. As the games themselves are done with passion and in high quality, Mannonen notes that the games will speak for themselves and the traditional marketing strategies, such as banner advertising and big marketing campaigns are no longer needed.

Mountain Sheep has not followed any common ways of promoting their games. Their strategy is very money-wise and different. Mountain Sheep has been able to gather a rapidly growing community of gamers to wait continuously for their new releases.

It is important to share information already in the early phases to make people wait for the game and start talking about it on social networks. When the hype has been created among people the wheel of viral marketing starts to move by itself and in order to keep that wheel moving they have followed few simple rules. Mountain Sheep goals are to deliver the best gaming experience and a feeling that people are familiar with. Also their longer post production periods are a way to satisfy their audience.

Even there will be no direct profits from the people who have bought the game already; long post production times are a really important issue for Indie success.

As the game continues to be polished and perfected, old customers will play their games longer and importantly show the game and praise it to as many friends as possible. This highly efficient viral marketing will boost the sales of the released game and it will gather new fans for the fan base. The main thing is to get the growing amount of people playing the games and to keep them playing and being interested on the future releases.

The familiarity of the game play feeling is a crucial part of their marketing strategy. Mountain Sheep's main goal is to provide a familiar gaming experience for their customers. There must be a perfect balance of an old and new in the still personal and very unique game designs. They have followed their instinct well as the Mountain Sheep has had already four TOP#1 App Store releases during the last few years⁴².

2.4. Project management and agile development process

2.4.1. Prototyping

It is vital for an Indie game developer to practice active prototyping. There are many reasons why making prototypes is important. One is that your idea becomes something concrete, an actual playable product; the first version of your game.

Indie developers as well as AAA developers support using techniques that enable so called "fast prototyping". This means that a prototype is created with minimum effort using already existing technology. Game Engines and APIs are efficient tools for fast prototyping of your game ideas, just to name a few: Flash, Unity3d, Box3D and XNA.

It is also highly important to select the correct plugins and add-ons to the selected Game Engine. There are massive amounts of ready-wheel to be taken into use instantly. In case you use engine such as Unity3D, observe these free plugins: iTween, Othello2D Framework, A* Pathfinding project and these low-

cost plugins Ex2d, Prime31, PlayMaker System and SmoothMoves. Using correct tools from the start will save you immeasurable amounts of programming time⁴³.

Making a prototype focusing on main new features that your game idea has is also a good practice. If the prototype does not give insight to the core questions raised from the game idea and its features, it's rather useless.

A good prototype can work also as a Proof of Concept and can give new insight to your concept that has earlier been only on paper. Also it is worth acknowledging that a developer who can make a prototype can most likely continue that work to an official release.

Prototypes are also invaluable promotional material in case your Indie Company needs funding from outside sources. Achieving publisher deals is naturally easier when you already have real content to demonstrate^{44 45}.

2.4.2. Milestones

Milestones are often used in Game Development. They're an important scheduling tool for both Indie and AAA game development. The basic idea of a milestone is to set a specific level of achievement in the game development cycle. A milestone could be set for example when completing prototype or when completing the first beta release. There is no minimum or maximum number of milestones that a project can have, but usually less than ten are used.

The benefits for using Milestones come from the obtained structure to working process as well as the ability to schedule your project more efficiently. The schedule and time-estimations are easier to make when a project can be broken into smaller sets.

Milestones can be used alone to provide most agile development. Milestones suit well to a smaller passionate team of developers who need maximum agility and flexibility in their working processes. However, when teams get bigger and

more people are involved more structure is often needed. Development Process Models are often taken into use^{46 47}.

2.4.3. Development process models

Agile development process models (Agile methods) can be used in Game projects. There are already multiple agile methods such as Scrum, Kanban, Extreme programming, TDD (test driven development). Especially in game development these methods are often customized to fit the current project or situation. This customization of the methods is often required because sometimes practicing agile method fundamentally can increase overhead in project management tasks^{48 49 50}.

Much older non-agile methods such as the waterfall model are no longer seen as a feasible practice for Game development. Because the game development process is highly iterative, the older models are not working very well. So called hybrid models that combine older software development models and new agile methods have also been created to solve the issues behind fully non-agile methods.

Due to the emphasis of audiovisual experience in games, a rapid incremental and iterative process is needed. Usually there are smaller iterations inside each Milestone which helps to split this Milestone into smaller parts and to gain more insight on scheduling it.

Inside these iterations, each new feature and level can be tested with maximal effort. Games also need a lot of time for balancing, tweaking and polishing everything from controls to difficulty levels and AI, so it is obvious that the development model needs to allow rapid changes and provide proper process cycle to enable maximal efficacy. For this reason the modern agile methods suit better the needs of game development^{51 52 50}.

2.4.4. Team commitment and motivators

A dedicated and motivated team is a crucial part of Indie Game development. The whole team has to believe in the vision of the project and the goals set⁵⁴. The best team is the one that knows each other well and has worked together earlier.

When everyone knows the goals and has a feeling of being an important part of the team, knowing his personal efforts are respected, the motivation increases naturally. Indies sometimes even see their development team as a “band” or as an organic collective which makes the game together in efficient synergy¹³. An Indie development might be compared to a rock band or even a bigger orchestra working together^{55 56}.

The importance of great collaborators also increases ethical project management. Instead of seeing employees as replaceable workers, each member is valued as an individual who has unique strengths. Team members have no need to compete inside an Indie Game Studio when everyone knows their area of expertise.

When an old fashioned hierarchical company structure is avoided the connection between all participants is enhanced. This enables free communication and creative freedom as employees are no longer under artificial hierarchies that restrict the positive flow of development. Developers should have full control over their work and share their visions in a proper creative atmosphere.

In the best situation no external influences such as investors or board members affect the team’s performance. The team chooses its own path and they are responsible only to themselves. As the team collectively feels the project is their own creation and a unique piece of art work, their motivation and dedication grows. The Team can focus efficiently on their goal which naturally improves motivation among team members.

Poor motivators such as money should be avoided and motivators such as freedom, flexibility and passion should be used instead. There have been studies insisting that more money does not solve any motivational problems inside a company but more over people do things better if they find them challenging and interesting. Personal professional development is a great motivator that a wise team leader can use to make employees feel self-pride of their work.

Money should not be used to blackmail workers for better results in a goal oriented manner, moreover enough money should be provided for the living expenses of the worker, so the money does not cause extra stress influencing the work negatively in the future. There should not be huge gaps in salaries between team members, as there are in mainstream companies following protestant working ethics and company hierarchies; instead providing everybody according to their needs in their current life situation^{57 58}.

The company should be able to react in a respectful manner to different situations, fathers should have time to be with their young children once in a while and the company should think the best for its employees if some unexpected issues or situations occur. And as your team consists of talented intelligent people, over patronizing should be reduced as team lead should trust each developer's ability to make the best decisions⁵⁹.

2.4.5. Multitalented developers

When the team is small and there are less human resources available; different team members have to wear Many Hats. A perfect Indie team is a collection of multitalented people who can be productive in as many development areas as possible⁶⁰.

Balancing how many Hats the developers should wear has proven problematic. In many Indie projects a typical problem was wearing Too Many Hats¹⁴, meaning the working tasks were too fragmented and there were too many

completely different areas to take care of. This is not an uncommon scenario but one to avoid^{61 62}.

2.4.6. Outsourcing, hiring specific talents

Even if your team members are multitalented and basically could handle most of the critical development tasks, outsourcing freelance workers or consultants might be a good business decision. Outsourcing has proved to be efficient when your company needs a specific single skill that it cannot provide by itself.

However outsourcing talents from other countries can be risky and there might be many surprise costs which you might not be aware of. One possible scenario is when ordering art assets¹⁶ from a distant land. The amount of specifications you might need to prepare for the outsourcing company can be a lot of extra work⁶³.

Usually outsourcing is used in areas such as narration, story-writing and marketing. Sometimes also a testing team can be outsourced even though it's recommended to have an in-house testing person(s) or to arrange gameplay testing sessions regularly with focus groups or fans^{64 65}.

2.4.7. Office working vs. remote working

Remote working has been a controversial issue for companies for a longer time; naturally those who support protestant working ethics. If the company does not forbid remote working completely, they tend to be inflexible with it at least.

One reason that companies are not enabling remote working for employees in a proper scale is the belief that communication is not possible elsewhere than in the office.

This might be completely true in work that needs complete physical and verbal real-time interaction with its workers (for example work in theater or opera). However when working with computers, it is no longer mandatory for workers to

be physically present at the office every day. It is important still to have physical meetings when needed⁵⁷.

There are many possibilities how companies can arrange the remote working for its employees, just to name few: VPN connections, Online meeting rooms (like WebEx or Adobe Connect), Skype etc.

The table on the next page shows the view differences towards remote working depending on which ethical view is practiced:

Table 1: View differences on remote working^{57 66}

View differences on remote working	
Protestant ethic view	Hacker ethic / Open Source business view:
Pros:	Pros:
+ Productivity is increased (less distractions)	+ Proper flexibility provided which matters
+ Morale increases thru freedom	+ Some working tasks just are better to do from home
+ Some working tasks just are better to do from home	+ Eco-friendly way of working
	+ Productivity is increased (less distractions)
	+ Morale increases thru freedom
	+ Enabling remote possibility is a future investment
	+ Time scheduling for the employee family (due to not wasting time in traffic jams (commute), you have more time for you family)
	+ As money is not the main motivator, employees are fine to not get the top notch salaries; in chance more flexibility is provided
	+ Employer trusts the employees, no huge efforts on monitoring is needed
	+ Instead of achieving "ownership" status over employees, dedication and passion are more important

	+ More humane working life benefits create more motivated and dedicated employees
Cons:	Cons:
- Lacking in “ownership” of the employees	- Communication can be an issue, if not supported and practiced properly
- Monitoring workers	- Avoid pressure and unhealthy competitive atmosphere that leads to over working
- To trust employees is needed	- Collaboration (sometimes a fast interaction between team members are in need)
- Standing out in the company as remote worker (in reaction to competitive working culture, you might need to actually work harder than others)	- Regular work lunch
- Communication can be an issue, if not supported and practiced properly	
- Collaboration (sometimes a fast interaction between team members is needed)	
-Regular work lunch	

2.4.8. Team work

Successful Indie Game Development relies entirely on its team; most of us are not multitalented-geniuses. An Indie Team must consist of motivated and devoted team members that can work fluently with each other. It is a plus if the team members have worked with each other earlier so they know each other already on a mental and practical level.

Properly functioning team working reduces the amount of work load on individual team members. One of the benefits is that when developing together the individual tasks are smaller and the work load smaller. This helps also scheduling as it will increase productivity and reduce the time taken on separate tasks.

Another positive effect of teamwork is that it increases an atmosphere that has healthy competition. This healthy competition improves the productivity of a company. Working towards a common goal and at the same time competing with each other to do better can help foster ties at the workplace. However there is a thin line between healthy competition and unhealthy competition. Every time there is a punishment, humiliation or feelings of worthlessness involved you know you have crossed this line as mentioned in the Ethics and Values chapter earlier^{67 68 69}.

It is really challenging to find the correct people to begin game development. Devoted and trustworthy employees are really difficult to find; especially for Indie companies. I share my own experience on this topic in the Postmortem in Appendix III.

It is almost a cliché to say Indie team should be like a living organism that can react and work in the most agile way; like a barrel old rock band¹⁷ in common terms³⁸. However according to many Indie postmortems this seems to be the ideal work practice and companies count it as a success always.

Important matter is to lift each other up and keep a “We can do this!” spirit growing inside the team. In difficult times, when doubt on the success or completion of the project comes, the team members must support each other so the project can continue normally. Recognizing positive things happening rather than sticking to the negative will improve the team spirit.

The psychology related to team management is also an important issue as there will be conflicts inside the team as all team members are different personalities. There should be one respectful and experienced developer in every team that could solve the issues as they are raised and who can carry on even through the hard times and give faith to other team members^{70 71}.

2.4.9. Quality assurance

For small Indie game studios Quality Assurance (QA) is really important. The game needs to be tested as much as possible before the big launch. The Quality Assurance should include a proper Test Plan and necessary tools for testing; also it is important to frame up the test cases.

Scheduling time for the quality assurance or outsourcing will pay you back always. The more bugs the game tester(s) can catch before the game is launched the better. All bug tickets should be reported and documented in a proper organized manner for example using Bug Tracker software. Ryan Keable tells:

“This is the most important and most obvious part of the QA process. When you are writing down bugs, you are identifying and categorizing an issue in your game that needs to be fixed. Reporting bug information correctly and categorizing it clearly will make the job easier for those who will be required to solve it”⁷².

Testing is categorized to black-box or white-box testing. When making black-box testing the information about the internal workings of the software is very limited; the information that tester has is similar to end-user.

White-box testing instead has more information about the software. The tester can use internal functions or a second software suite (usually a debugger) to track issues while the software runs, providing information to the tester or collecting it to a test log.

“The biggest misconception about software testing is that any one method of testing is better than another. There is both an art and a science to software testing, and neither of them should be ignored” states David Wilson in *Gamasutra*⁷³.

Testing can be often also automated so the human factor can be reduced. It is also a good practice to take quality assurance as an active part of your development process or testing more heavily on a particular Milestone point.

The team leader should also encourage all team members to do actively gameplay testing on new builds. Once in a week specific gameplay sessions can be arranged, for example on last hours of Friday⁷⁴.

Bug free game will most likely get better ratings and reviews than a buggy one. Also testing your game from the beginning as a part of your development process allows you to do better scheduling. As programmers can constantly fix bugs rather than fixing huge amount of bugs that were found late in the development cycle, schedules are easier to keep⁷⁵.

Arrange game play testing sessions to get feedback from players. Usually volunteers are used for prototype/alpha game play testing and later on beta stage specific focus group tests are arranged. These tests will give valuable feedback on the mechanics, feeling, controls, graphics, artificial intelligence etc. and if the target/focus group likes it or not^{76 77 78}.

Lastly, quality assurance people do important work so the game can be bug free and successful. Company should not practice bad ethics on them, nor use them only as human resources rather than real team members⁷⁹.

2.5. Technology

2.5.1. Free or affordable tools

It is crucial for an independent game developer to be aware of the free or low-cost tools available for creating games⁸⁰. As an indie, you have to keep your budget as small as you can and paying huge amounts of money for 3d modeling software is not feasible⁸¹.

Before starting an Indie Game Project you should always make a brief research what parts of the development you can do with free or low-cost tools or just update your knowledge about the best solutions at the current time. There is enormous variety of free and low-cost tools for almost every part of the development. Choose wisely the ones that are suitable for your needs^{82 83}.

2.5.2. Game engines

Often indie game development cycle is not as long as AAA-titles development cycle. So it is natural that Indie companies often use existing game engines. The benefit is that you can shrink the development costs and times, because you do not need to “re-invent the wheel from scratch”, but instead can start immediately creating your game.

There is already a huge variety of game engines that will fit on most developing needs. Before starting a project, you should make a brief technical research about the Game Engines that would best fit your project. The prices of game engines can vary dramatically, but there are many quality low-cost engines available^{83 84 85}.

As an Indie developer it is sometimes important to be able to make compromises, such as designing game to fit the current technology available and not vice versa³⁸.

This approach will not mean that you cannot create interesting and fresh games that fulfill your visions; it's a guidance to not “think too big” and to have realistic

scope for your designs. However when you need to develop completely new engine or more time consuming custom made features to existing Game Engine you have to have real business reasons for making such decisions^{86 87 88}.

2.5.3. Version controlling

Many of the postmortems studied stressed how important it is to always have a working build of the project available for all team members. This can be achieved through version controlling.

Version controlling becomes crucial in any Indie Game project that includes a team, big or small. Developers require newest versions of the source codes and other game assets so that simultaneous development is possible for the team members. There are many good options how to do version controlling freely, such as Mercurial or Git.

Services such as DropBox or Google Drive can also be used, but it is recommended as the project goes any further than a sketchy prototype, that a proper version controlling is configured and taken into use. There are some free or low-cost cloud services also available for Indies in case making your own Git server is out of the question^{89 90 91}.

When developing with existing technology such as Unity3d game engine make sure you take advantage of the external version controlling features engines provide. In case of Unity3d it is a straightforward task to get your project configured to support version controlling.

Remember to ignore all the unwanted files so they won't cause unnecessary conflicts later on between your branches. This is really important for fluent development. Keep your version controlling software also updated. Some developers also insist that it is better to keep assets away from the version controlling to avoid conflicts⁹².

2.5.4. Bug trackers

Bug Trackers are an important part of every development pipeline. They allow all bugs and errors to be reported and prioritized and are one of the main basic tools of QA testers and Game Programmers. Having access to bug reports from a common place is a necessity for all game programmers. Usually this service is available online.

There are many detail levels of the bug reports, but their main goal is to demonstrate the programmer how the bug can be reproduced and what is the severity of the bug. If possible bug report should demonstrate simple steps that trigger the bug to occur in the execution of the game. The programmer has to be informed also if the bug is not reproducible or that it can be reproduced only occasionally. All these bug reports are gathered inside a system, called a bug tracker^{93 94}.

There are many good free bug trackers such as Mantis and Bugzilla. These can be used for even bigger projects and for Indie developers, free software is always welcome. There are also bug tracker cloud services available in case you don't want to maintain your own server⁹⁵. Bug tracker comparisons can also help you to select the bug tracker that fits the requirements of your product⁹⁶.

3. Methodology

This research used both Literature Review and a Comparative Analysis of Postmortem data as a basis for the results later provided.

As the research was mostly qualitative, many areas were difficult to measure comprehensively and the result chapters 4.1-4.4 should be acknowledged to be based on the information gathered from articles, seminar speeches and documentaries.

However, chapters 4.5-4.8 provide results from a more quantitative perspective as the Postmortem data could be measured better. This data was measured in

simple units, occurrences and grouped to meaningful groups to make data mining possible.

The Literature Review gave an overall picture of the subject and important information for observing the approaches and practices currently used in Indie game development.

On the contrary the Comparative Analysis aimed to find out more in-depth information of the Indie projects and compare that information with the one collected from AAA projects. As the AAA and Indie project's postmortems are made in similar form, the data collected could be compared in a sensible way.

In my Analysis I compared the problems and successes of both Indie and AAA postmortems. As every common postmortem includes both of these areas I had a freedom to select my postmortems randomly to provide fair results and observations.

3.1. Purpose of research

The research was needed due to the rapid expansion of Indie Game Development culture and the fact that so many Indie projects have a history of containing critical problems, called pitfalls. In my research I wanted to look into the world of Indie projects through current information available and by studying the data included in the postmortems of Indie and AAA projects.

The goal of this qualitative research was to define a focused guideline and necessary practices for possible future production of our prototype and to provide improvement ideas for critical problems of Indie game development found in postmortems. This guideline was constructed by first defining commonly preferred approaches for Indie development and by using comparative analysis method on Postmortems to access deeper knowledge around the topic. The focused guideline is reflecting the improvement ideas for critical problems found in comparative analysis. This guideline can be found in the Appendix II.

The Research aims to answer these three questions:

- Research Question 1: What are the commonly preferred approaches to Indie Game Development?
- Research Question 2: What are the Pros and Cons of both AAA and indie game projects? Where did they succeed, where they did not succeed?
- Research Question 3: What improvements are needed for Indie Game Development?

3.2. Procedure

The data for this research was collected mainly from articles, books and postmortems. The most relevant and highest quality sources were selected. The articles were gathered from popular web portals such as Gamasutra, Eurogamer and other sources that were in the scope of the research. As there were not many earlier researches or studies made from Indie Game Development, I needed to include blog writings and opinions from experienced Indie Developers for more insight.

One perfect example of such data is the design opinions written by Veteran indie game creator Edmund McMillen. Also professional seminar recordings and an Indie Game Documentary were used as data.

For defining common preferred approaches (chapters 4.1-4.4), I gathered information based on the literature review. The information such as pitfalls, quick wins¹⁵ and To-Dos were included and constructed to commonly preferred approaches.

However, for framing my focused guideline (Appendix II) and improvement suggestions (chapter 5.7), I made comparative analysis which was based on a total of 20 Indie and 20 AAA postmortems. This analysis pinpointed problems and successes in each Development Area and which specific Type Groups the occurrences belonged to. Analysis also gave deeper information on single occurrences and these occurrences were later discussed in detail in the Discussion chapter.

Each postmortem was studied and analyzed in order to collect a database of Pros and Cons of Indie and AAA game projects. In my analysis I compared Indie and AAA projects to find differences between their Pros and Cons. This method of analysis was selected as it was best suited for my purposes.

It was less difficult to group the data to the larger Development Area groups as it was clear which area of development the occurrence was related to.

However, while forming the more specific Type Groups, I needed to make certain decisions related to postmortem data grouping and difficulties were encountered since some of the information found could belong to two or more Type Groups.

I aimed to group these problematic occurrences using my best judgment. Rational compromises were mandatory, especially with the problems in Design and Development areas. For example instead of having separate problem type groups “Bad Game Design” and “Gameplay Issues”, I decided to merge these two. It was justified to do so, because gameplay is a specific sub-part of game design.

To keep the provided charts readable I selected only Top 10 most common Type Groups on these charts. There was a vast amount of minor Type Groups that are not included on these charts but which can be found in more comprehensive table in Appendix I.

Based on the comparative analysis of Postmortem Data, I selected 6 occurrences to investigate in-depth. I reflected these occurrences with the commonly preferred approaches gathered from the literature review and to other postmortems that included possible solutions for these issues. In addition to the commonly preferred approaches, I introduced improvement ideas and possible solutions for the Type Groups chosen. The improvement ideas and suggestions were based on the data and research results.

As my research was heavily qualitative it was sometimes difficult to weigh the importance of different Type Groups as there was a vast amount of information available in the data gathered. This made it a cumbersome task to decide which Type Groups are included in the deeper investigation.

It was also complicated to decide the best principles and practices to suggest for improvements as there was a large amount of information available which did not always follow a certain pattern. However I aimed to open the ideas, reasons and principles when selecting the Type Groups for deeper investigation and the improvements suggested in Discussion.

From the research I constructed a focused guideline for our own future Indie projects. This guideline was framed to help evolving the prototypes to actual games and was tailored to suit our purposes and needs. It included 20 important topics to remember in Indie Game Development.

In my Postmortem (Appendix III), I have analyzed the development of NoBreak Media's puzzle game prototype based on my own experience. I reflect my experiences on the prototype development to the commonly preferred approaches and analyze which approaches were the most beneficial to our prototype and which approaches we failed to follow; clearly describing the costs we had to pay when we overlooked these issues during our development.

The Postmortem also reflects NoBreak Media's prototype development process against the information gathered from the Postmortems and the improvements suggested. This way the Prototype was connected firmly to the topics discussed.

3.3. Measure

The Postmortem data was randomly selected, which means I did not select Postmortems from a certain game genre or year. This was a way to avoid my personal opinions and likings when selecting postmortems. When selecting the data I followed a basic rule to avoid Postmortems aging more than 15 years.

With these decisions I focused to combine a data set which would represent overall game development fairly, give a bigger picture of it, rather than a narrower specific portion of a certain genre on a certain year.

In total 20 different Indie Postmortems and 20 different AAA Postmortems were selected each containing information on Pros and Cons (typically five each) of the game project. This data of Pros and Cons was first grouped into larger Development Area groups. At the beginning five Development Area groups were made and later more detailed Problem and Success Type groups (sub-groups) were created for further analysis of data.

The charts' values are provided as percentages to avoid sampling error. This action was needed because of a small deviation of occurrences as some postmortems used as data had more than five or less than five Pros or Cons. Because of this slight deviation in the data (total occurrences amounts) between Indie postmortems and AAA postmortems, I decided to present all information as percentage values to avoid sampling error.

Each problem and success in the different development areas was analyzed and more specific problem and success type groups were created. All occurrences were carefully grouped into these type groups. The Problem and Success Type groups were designed to be generic enough, so they would present the occurrence deviation correctly.

Deciding how specific groups were needed was problematic in the beginning. I decided later in the research to merge some of the Type groups into more generic groups, which were too specific. Merging was required in order to achieve comparable results for each group and was one of the best decisions I made during research.

I selected a few specific Type groups of major concern for deeper investigation. Justifying the importance of these specific groups was based on a difference amount between AAA and Indie occurrences of that type group. This variable was used to rank Type groups that were of special importance.

An exception was made with the Scheduling Problems Type group as it was selected by other variable; in this case the research's special interest towards project management and that it belonged to the Top three Problem Type groups. For these selected Type groups multiple improvements were suggested.

The findings from Postmortem data together with observations made in Literature Review are meant to provide a wider portrait of the whole Indie game development and give perspective on the subject as whole. The data and data grouping are presented in Appendix I in more detail.

4. Results

The following Results chapter will represent the results which were found from the Literature Review (chapters 4.1-4.4) and the Comparative Analysis of Postmortem Data (chapters 4.5-4.8).

The results on chapters (4.1-4.4) answer to Research Question 1: "What are the commonly preferred approaches to Indie Game Development?"

The results provided in chapters (4.5-4.8) answer to Research Question 2 "What are the Pros and Cons of both AAA and indie game projects? Where did they succeed, where they did not succeed?"

These results of both Literature Review and Comparative Analysis are later discussed in Discussion Chapter. The results of the Literature Review are discussed in chapters 5.1-5.5 and the results of the Comparative Analysis are discussed in chapters 5.6-5.7 later in the Discussion.

4.1. Project area management results

Lean Project Management was a requirement for a successful Indie game project. This Project Management includes scheduling, planning and making

sure we can reach our deadlines and complete our Milestones. As creative persons, Indies are not always using enough effort on a proper level of Project Management and this causes a lot of problems as we later will see in Comparative Analysis results (Chapters 4.4-4.8).

The articles showed it was commonly preferred to wear many hats; to have many positions. It is not recommended to have strict position titles but instead to share responsibilities and even single working tasks. The importance of team is emphasized and it is important to have a good team where optimal situation would be if the workers have earlier working experience together.

It was recommended in the articles, to have one strong leader in a development team who would have possibility in technical terms to finish the game alone. Project management skills are also recommended as well as having an actual project manager. All conflicts that occur should be handled in a proper manner and a constant transparency inside the company was a requirement. Everything has to be open and continuous conversation needs to be supported.

Game developers were recommended not to treat the Scrum method as a solution for all problems (so called Silver bullet) but instead to acknowledge Scrum methods as a set of tools for the development process. The Scrum model was recognized as problematic when a game is entering the production phase and teams often blend back to some extent to waterfall models. However it was recommended to practice alternative agile methods (Lean and Kanban) in order to fix the production issues that Scrum is having and still continue using agile methods. Making milestones was regarded as a mandatory approach to any Indie Project as they help to schedule and organize the development process and to provide clear goals to developers to avoid losing focus.

Outsourcing some of the game content or areas of development is considered as a good possibility for smaller teams but on the other hand working in the same location or office was also acknowledged to be important. For starting Indies getting a highest paying day job as possible was emphasized, which would support you in the beginning of an Indie career.

Many professionals also guided Indies not to overlook the post production as it is as important part as any other part of the project. Mannonen stated this same principle clearly in his speech at Assembly 2012, as discussed in earlier chapters.

Many professionals of the business are emphasizing also the importance of flexibility. This means an ability to adapt to different situations, surprises or changes. From the vast amount of postmortems and articles, the importance to be able to correct your path was clarified. If things are not going like they should an action was suggested to do as soon as possible, even if it would mean dropping the whole project in a worst case scenario.

It was encouraged in the Literature Review to have critical thinking not only to your game designs but also your ways of working. There is always an opportunity to try different approaches and instead of keeping going with development process methods that are not working properly you should break them in parts and analyze where the bottlenecks are or what is slowing your progress. After pinpointing your root problems you should investigate ways to fix them. Taking action as early as possible is important.

For individual workers flexibility can also mean their ability to make compromises and to see conflicts or decisions from another perspective than their own. As there was as a common preferred approach the game project has to always be put in first place instead of personal agendas or visions. The ability to put your own ego and pride aside and really think what is the best for the future of the game is crucial for Indie Developers. Voting has been recommended as a common preferred approach to give everyone a voice if there is a critical conflict about some feature or aspect of project management.

Flexibility can also mean the way the company is run as explain in Literature Review. As it is important that usually employees would work as much simultaneously as possible this should not be written in stone. As mentioned earlier in the thesis, there is also another side of life that needs to be taken care of besides work and as long as this private life has balance; the employees

continue being motivated. Nothing is less motivating for an employee than a feeling of powerlessness over his own work and constant stress over issues that would require flexibility from the company's side.

4.2. Business area results

According to Literature Review, it becomes clear that certain level of clarity is preferred. Each Indie game studio should have at least one person familiar with the business side who can take correct actions to make the game profitable. A business plan plays an important role as the team needs one.

Another preferred approach was to have a real marketing strategy and to keep things organized and evolving. This can include many areas as mentioned before in the literature review such as community, traditional marketing, viral marketing etc.

Healthy Indie game company should practice cash flow summary or book keeping in more general terms. Indie Company should always be aware of its expenses and what actions it takes from the team to keep the business going on. It was said, that making a research on how much profit can be expected by studying similar games done before. Some amount of analytics was emphasized, so you can know details from your common customers.

As a common pitfall an Ad-based marketing was discouraged as it was regarded as costly and ineffective. Instead viral marketing is definitely a wiser investment as it is low-cost but can be really effective; especially when you have a firm fan base, instead you should put more efforts on Post-production as explained in theory part.

However, these tasks can and should be divided to different team members, in case the company does not have money to invest on a professional marketing person. The one who has charismatic voice and speaking talent can make all trailer voice overs and the one who has proper writing skills can take care of blogs and newsletters. A member who is creative, witty and funny would fit

perfectly the viral marketing hat. As Indies are usually tight with budget the company should use resources efficiently.

According to articles, selling your games can become emotional and in the worst case you attach your self-worth to how your game sells. Golden rule given was not to take any success or failure personally and prepare for good and bad times. Do not let reviews or statistics define you as a game developer but rather learn from them as constructive criticism without getting depressed.

Efforts on Post Production are also preferred as they will keep customers attached to your games. Post Production allows you also to cross advertise your new games using your old games and vice versa.

Always communicate with people in a professional way and do not respond aggressively to criticism. As you cope with people and treat them as you want yourself to be treated longer lasting relationships will be built. Keep in touch with people and if possible be overwhelmingly kind to everybody.

Do proper marketing research and trust yourself on the decisions you make. In case of a mistake you will then know that at least you did your best and you were aware of the pros and cons of the particular decision.

4.3. Development area results

An important approach preferred in development side was the use of existing game engines instead of inventing the wheel yourself as it saves a lot of development time and gives you access also to plugins and add-ons.

It was said that creating a custom engine with a small Indie team should have always a good business decisions behind it and specialized engine programmers are then needed in the team. This is a long path to go and more comfortable decision is to use existing technology such as engines and tools. There are many possibilities for using this technology efficiently and feasibly in your game projects.

Another approach preferred was to allow all team members to have access to the newest working builds. This requires of course Version controlling and will benefit the QA process efficiently as the newest versions are always ready to test. Fixing bugs immediately is also preferred as you will not have any crashing builds available; instead, only working ones.

Having a good time during development was regarded as a matter of importance. Your team should have fun together and keep passionate what you are achieving. As mentioned in the theory, avoiding strict protestant working ethics and supporting hacker ethics instead will benefit on your team's well-being and bring more joy inside your company.

As I read through Indie and AAA postmortems one really important issue was mentioned repeatedly; the importance of communication between team members. This is really important to point out as with a good flow of communication, open conversation between co-workers and transparency inside company, many bottlenecks in the pipeline and endless re-making of features or designs can be avoided.

The team has to keep focused the whole time of the development cycle and without good communication this is hard to achieve. Proper communication over detailed design documentation is preferred in most of the projects and this clearly states the great importance of the subject.

It was commonly preferred approach to make the game playable from the start and allow the developer to study the game mechanics and player experience. Creating a prototype when using this approach is of course mandatory. The prototype will immediately tell if the game works out or not. Also gameplay testing sessions can be arranged. When you have your first proper prototype ready, get external feedback.

Feedback from focus group testing supports developers in choosing the correct direction. Instead of creating a skill game you might need to correct your direction and re-design the game to suit your target audience better. Game

designing is always a meticulous balancing between skill game and labor game and focus testing can tell you quickly which side you must put efforts to.

The lesson is to fail fast and as often as needed. If the prototype and its mechanics or game design do not impress your team or the actual players (gameplay testers), the prototype is probably not worth of continuing further. Failing fast let you move on to another, better concept without wasting your time on a design that surely won't make it. There is a vast knowledge base related to prototyping benefits, good practices and its common mistakes available on the internet for further reading.

As it was emphasized to do actively fast prototyping it was also a preferred practice to aim to reach your Milestones. Project's scheduling and time management has to be enough good so that it is an exception if the team cannot reach the Milestone in promised deadline.

If your Milestone(s) is missed you need to make a serious investigation and analysis on the reasons behind it. Analyze what was blocking your way and how you can fix it as a team in the future. You might have to do development decisions such as re-focusing your vision, create tools for more rapid development or make compromises (cutting features or content that are difficult to implement). Some of your team's developing practices might even need re-evaluation or changing.

Inside each bigger milestone developers should take vast amount of smaller steps with tiny intervals to allow constant process in all development areas. Old non-agile methods, such as Waterfall model cannot provide effective tools for needed process model qualities and using them is highly discouraged in modern development.

Usually a customized agile method together with Milestone approach mentioned above fulfills the needs for even more ambitious Indie game project. Implementing full-scrum in Indie Projects is not usually preferred however, due to the rather big amount of extra managing work involved.

It was encouraged to pinpoint the bottlenecks of your development processes as early as possible and make enough effort to solve them properly. Be brave to practice and try also new ideas, have an open mind about what you're doing.

4.4. Design area results

Common practices on design area of development emphasize the importance of having clear responsibilities and good collaboration skills. The design process can have better focus if the game design is not spread to many designers or voted inside the team.

It occurred frequently in Literature Review that the game audio should not be overlooked and the cost of art creation should not be underestimated. Reaching a level of quality in your designs and making your game design accessible also for mainstream gamers was encouraged; but still designing from your heart, realizing you are making art. A "Fun over features" approach was recommended; you should support game designs that provide highest fun with minimum amount of features. This keeps your Indie budgets in control and your development cycles shorter.

Avoiding feature creep was important in order to keep on budget and to keep hitting the deadlines. Also supporting game designs that are fast to make and which are minimalistic enough were preferred. It was told to think critically as game designing is mostly critical thinking. According to common ways it was also necessary to design to fit the technology not vice versa. Working openly with people and in communication with technical persons related to designs was suggested. As a designer you might not know the technical possibilities or restrictions well enough and a programmer or such can guide you to keep your designs feasible.

Making many hundred pages of documentation was not supported as no one has time to read it or update it. Instead open communication and more iterative

design processes should be encouraged. The Literature Review did not however contain idea that the documentation should be completely neglected. It was also important to practice networking and learn from others. Joining communities and observing what other designers are talking was a counted as a good practice. Learning out-of-the-box thinking from other developers in your community was encouraged. Also going for a walk to enjoy nature in order to have a fresh sight over things and finding sources of inspiration were preferred for Indie developers.

Designers should also keep themselves balanced and to let their brain have rest as the creative persons are prone to depression and other problems. This correlates directly also to your company's working culture and what working ethics are held in your company. Keeping yourself in balance also means you are excited about the game. In case you are not excited about the game you need to wonder why and why you are doing it at all.

Finally you should not avoid taking risks. Dissect existing formulas as all game "genres" are formulas. Level design, teaching rules, jumping patterns: it's all according to a formula. Investigate those formulas by breaking them into parts and observe how they work. Play games frequently to find out what elements you like. Decide why you like those elements, and then redesign them.

Deconstructing a game formula can teach you important issues from game designing that you might not notice otherwise. It was even somewhat harshly stated in the articles that chances are you are not a child anymore, so if you really prefer to do something more controversial, take a risk.

4.5. Problems in indie and AAA game projects

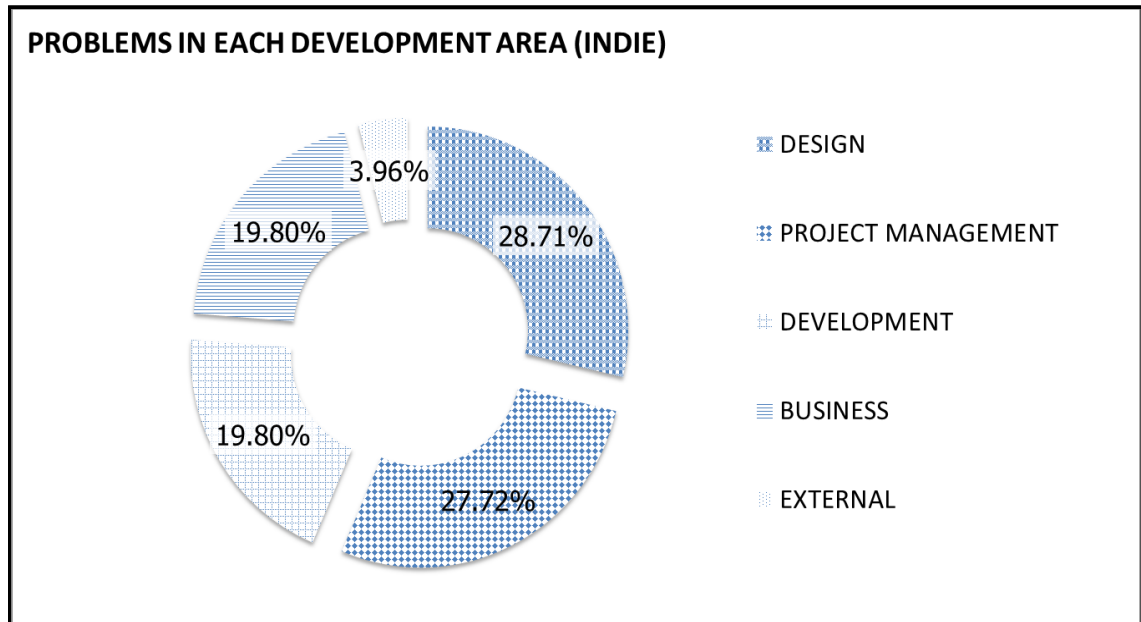


Image 1: Problems in each development area (Indie)

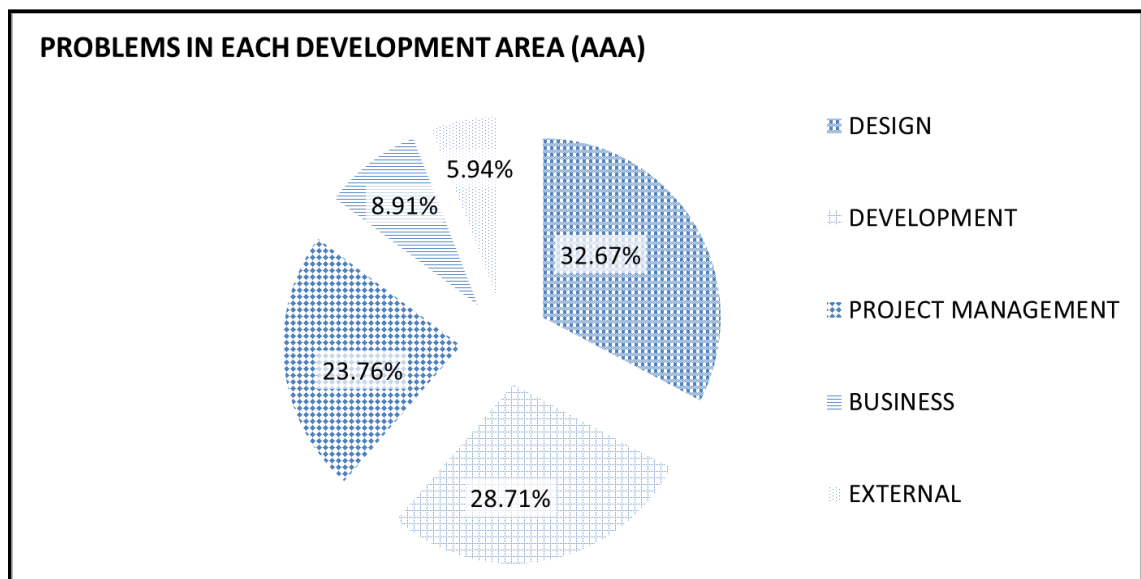


Image 2: Problems in each development area (AAA)

As shown in Images 1 and 2, the largest problem numbers were in Design and Project management areas. Development and Business areas were having significant amounts of occurrences. Problems in External area were only a minority. Compared to AAA postmortems, Indies major differences fell into Project management and Business areas, where Indie Projects had more problems than AAA projects.

Indie projects had fewer problems in Design and Development than AAA projects. Development had the greater difference between these two.

4.6. Successes in indie and AAA game projects

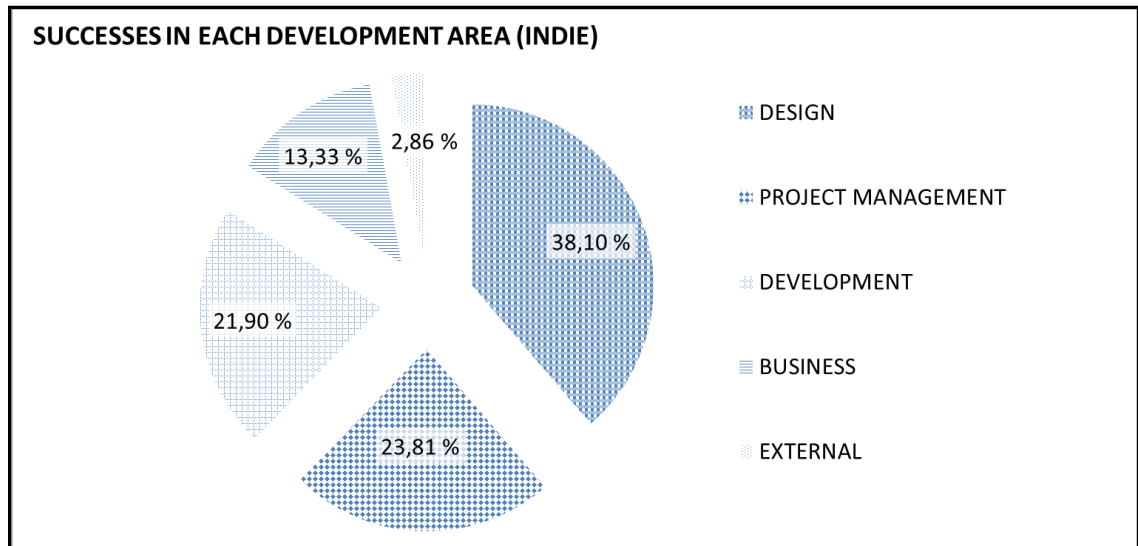


Image 3: Successes in each development area (Indie)

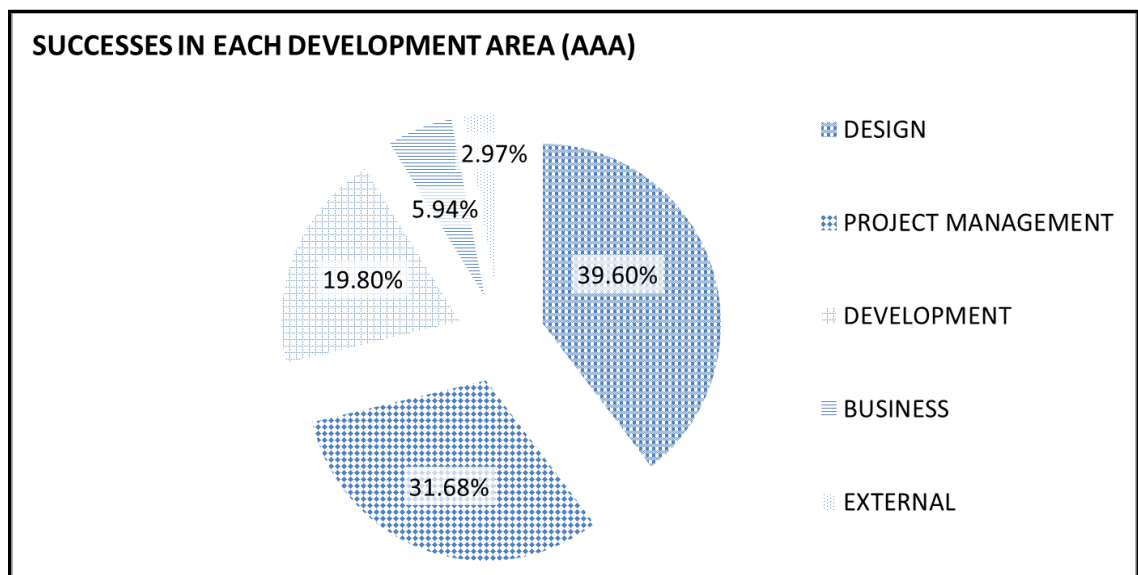


Image 4: Successes in each development area (Indie)

The above charts (images 3 and 4) show that both Indie and AAA projects had a majority of successes divided mainly to Design, Project Management and Development areas. Both shared the same Top 3 success development areas but major difference was found in Project Management. The research found that

there was significant difference in these success occurrences; as the AAA projects had more successes listed in this area.

Another major difference was found in Business where Indie projects listed more successes than AAA projects. A difference worth mentioning was also Indie projects succeeding slightly better on Development than AAA projects.

4.7. Problems found in problem type groups

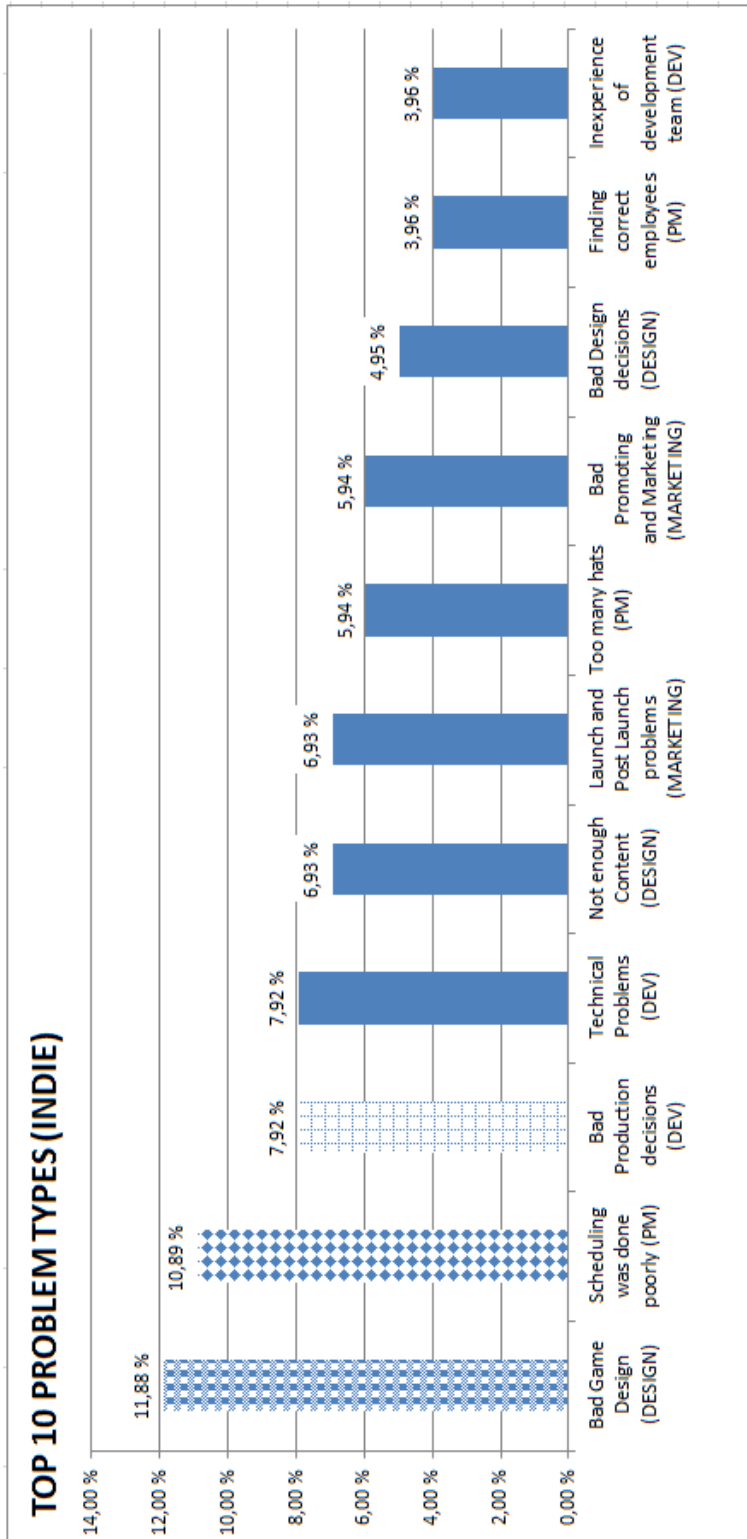


Image 6: Top 10 Problems types (Indie)

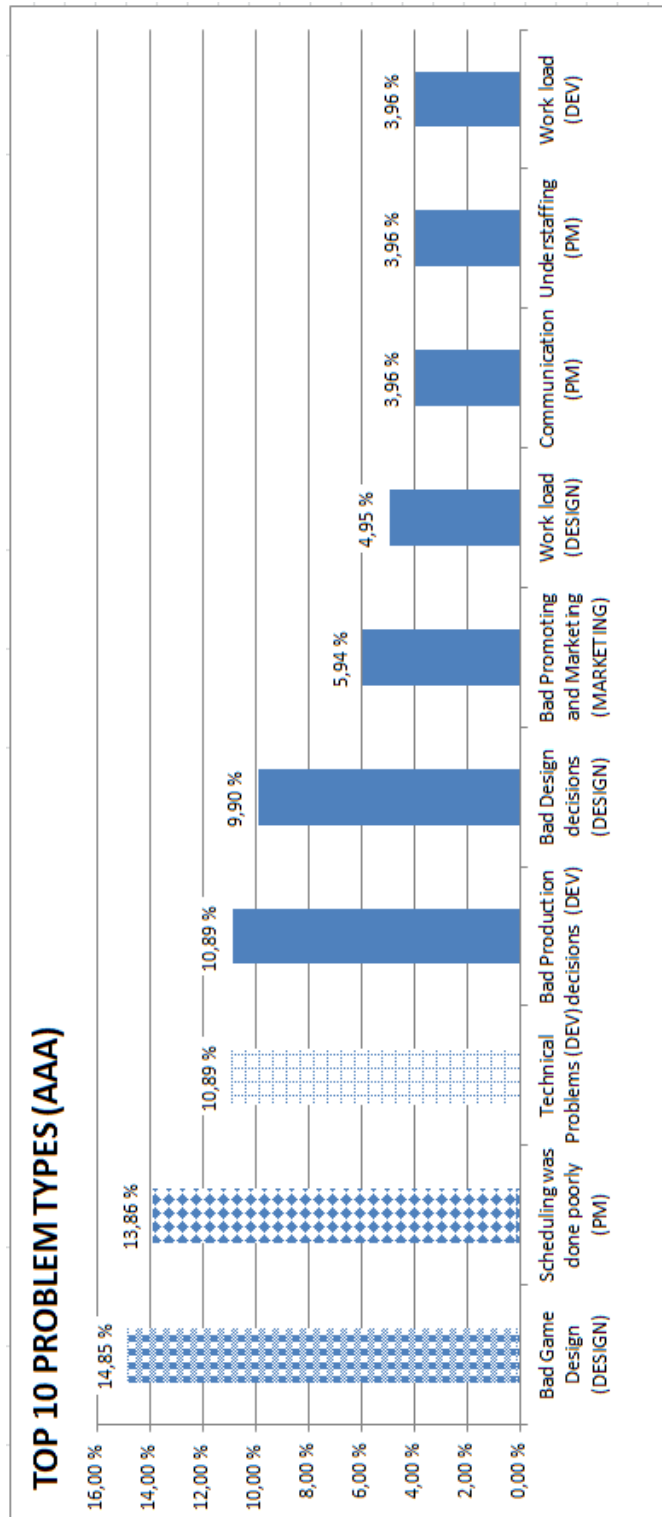


Image 7: Top 10 Problems types (AAA)

According to images 6 and 7, both Indies and AAA project shared the same Top 4 Problem type groups, not in the exact order though. There was not so much difference in the Top 4 groups mentioned and surprisingly Indie projects had lesser problems in these biggest groups than AAA projects.

The largest problem type group was Bad game design. This problem type group was part of the Design development area problems and contained issues from gameplay, controls, mechanics, multiplayer support, tutorial systems, unnoticed content, UI and balancing mistakes.

The second largest problem type group was related to Poor scheduling. This problem type group was part of the project management problems and it contained issues related to bad timing, crunches, inexperienced scheduling, underestimating work amounts and last moment features.

Bad Production Decisions was the third largest group. It belonged to Development problems and contained issues from unwanted compromises, implementing scripted sequences, inefficiency with assets, lack of iterations on tutorials, decentralized development and other failures in making optimal decisions.

However the major differences between Indie and AAA postmortems' problems were not in the Top 3 Problem Type groups at all. This surprised me as there were much greater differences in the smaller Problem Type groups. The significant differences were on:

- Not enough content (Design)
- Too Many Hats (PM)
- Finding Correct Employees (PM)
- Launch and Post Launch Problems (Business)

4.8. Successes found in success type groups

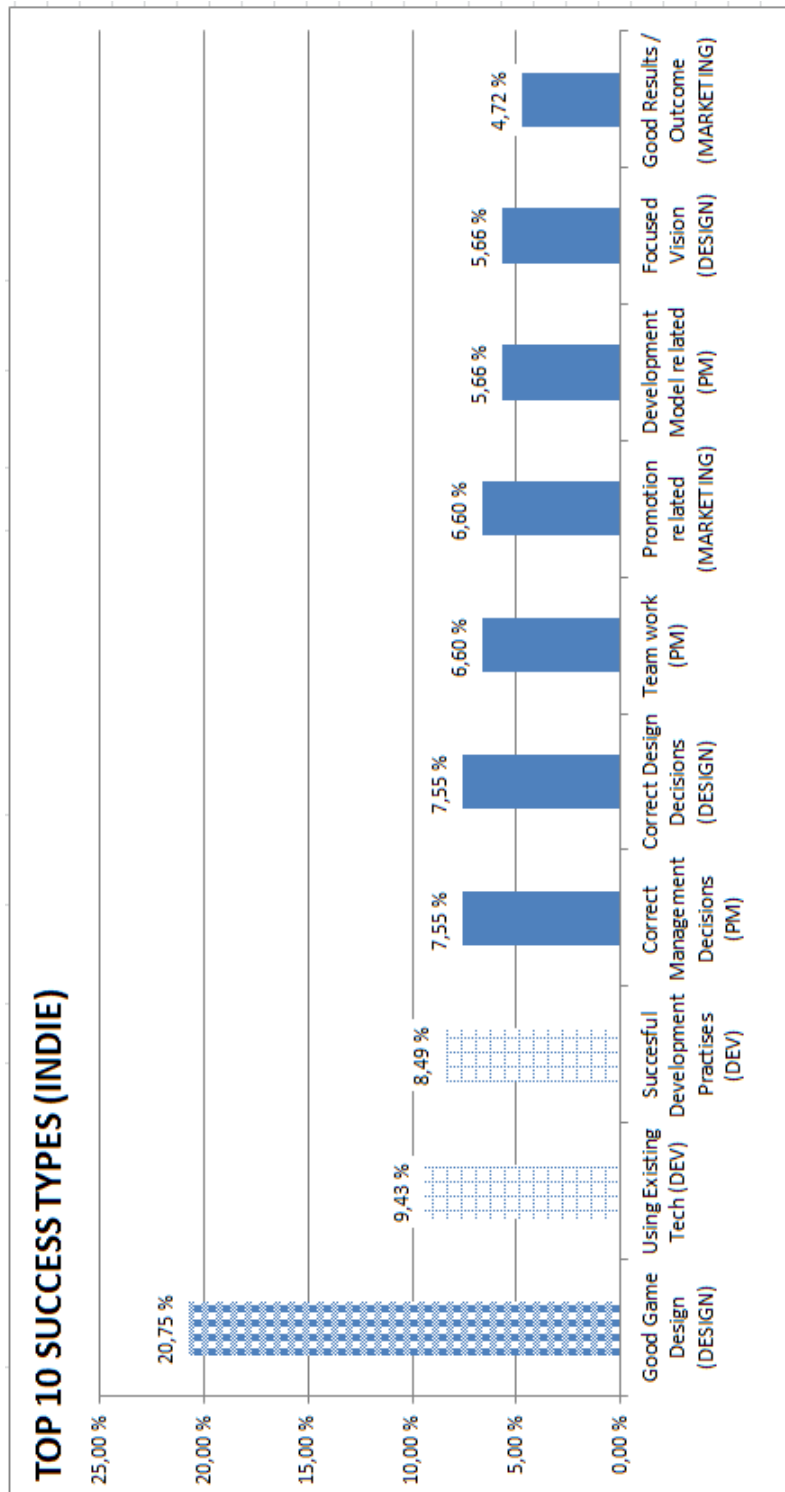


Image 8: Top 10 Success types (Indie)

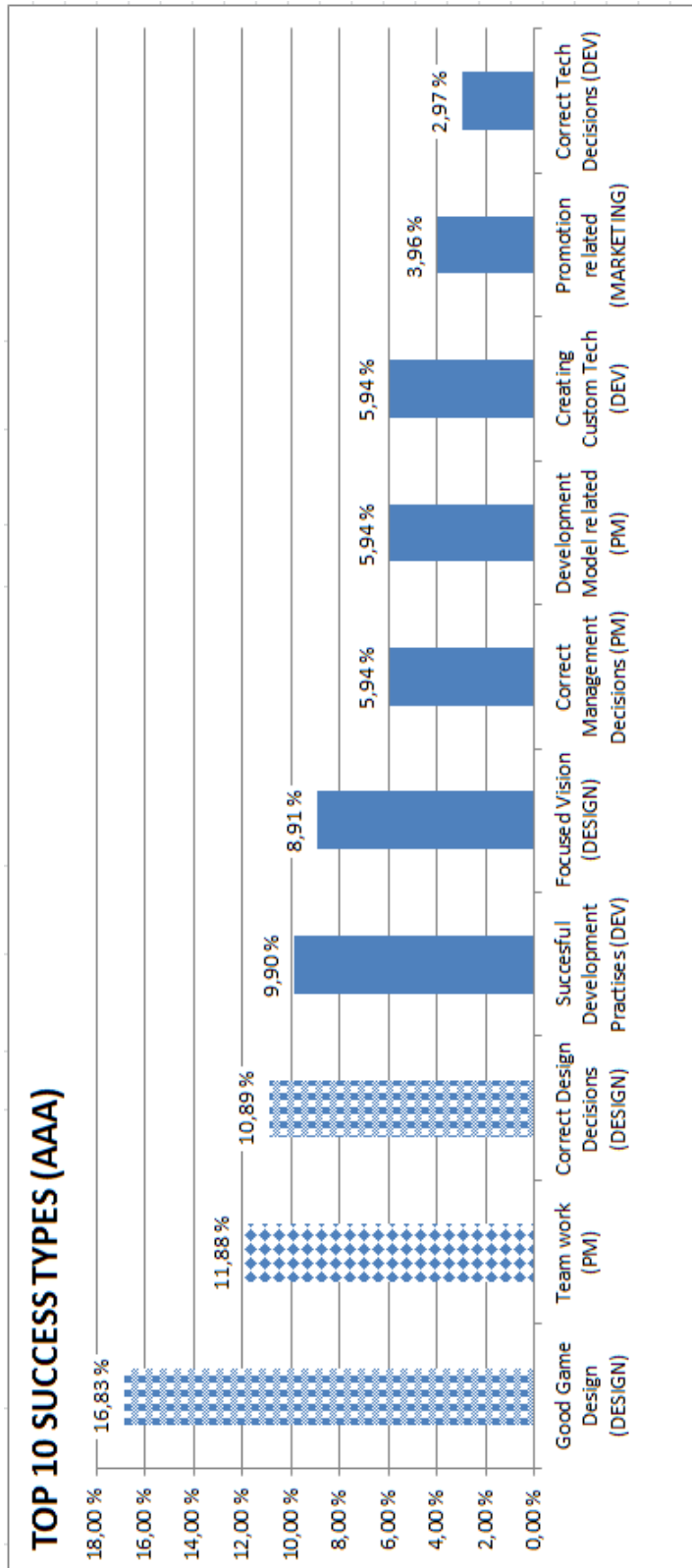


Image 9: Top 10 Success types (AAA)

In the Success type groups (images 8 and 9) there was a real change on the Top 3 Success type groups between Indie and AAA. Both had Good game

design as the largest group and here Indie projects did better than the AAA projects.

However the second largest success type group was completely different as Indie projects had successes on using existing tech and AAA projects had successes on Team working.

The major negative differences in these Success Type groups were in the Team Work and Creating Custom Tech groups. Minor, but still significant differences were on Correct Design Decisions and Focused Vision groups.

However major positive differences where Indies performed stronger could be found from Using Existing Tech and Good Game Design groups; the former having a greater difference.

5. Discussion

In this chapter, I discuss the results and observations done during the research. In chapters 5.1-5.5 the discussion is based on the Literature Review results (chapters 4.1-4.4) and the remaining chapters (5.6 & 5.7) of this discussion are based on Comparative Analysis results (chapters 4.5-4.8).

It should be acknowledged that the discussion in chapters 5.1-5.5 is based on Literature Review and is more generic than the latter part of the discussion (chapters 5.6 & 5.7) which is targeting towards more specific issues found in Comparative Analysis.

To be more specific, the latter part of the Discussion (chapters 5.6 & 5.7) aims to answer the Research Question 3: "What improvements are needed for Indie Game Development"

5.1. Thoughts on indie culture, lifestyle and ethics

The common definition of 'indie' is a bit lacking in my view. A more comprehensive definition would be a lifestyle that favors economic independence, freedom from mainstream culture, passion over money, sharing knowledge, indie community, artistic freedom and intellectual thinking. A certain bohemian (or even hippie) state of mind is often shared among 'indies' as well as values, principles and ethics which the mainstream business world hardly ever can offer. Nevertheless to achieve an absolute definition of the word "indie" is close to impossible, in my opinion even useless.

Personally I prefer to use more abstract term "indie spirit", when describing wider meaning of indie life. "A little bit of gold rush mentality. Look! The grass is greener on the other side of the fence! That sort of thing" as co-founder of Epic Games Mark Rein defines.

Often you see the term 'indie' connected to a certain trend at the current momentum; honestly I think that these inventions originate from the minds of marketing people. They want to sell their products to ignorant consumers who desire to be labeled as "individuals". Selling individuality or independence is one of the most common marketing tricks. It is clear that mainstream trends can hardly ever be true to the ethics and philosophical views of indie lifestyle.

Things that matter to me are however more spiritual still as the developed game should not support wrong values or stand against The Moral Law⁹⁷₁₈. It is important for me as an Indie to promote content which in the best case benefits the players also mentally and spiritually or at least lifts their mood higher rather than making them feel worse or addicted.

My personal opinion in the discussion around Protestant Working ethics is that this path of being a modern flagellant is coming to its end and change is inevitable. Indie game developers are among the first groups to bring the more humane work ethics also into the business world. Those companies which are incapable to comprehend, accept or adapt to the changing world will die away.

There is a big legion of people already who think alternatively on the priorities and values of life in our present time²².

The most terrifying trend I have seen happen during the last years is games that are designed on purpose to support gamers becoming addicted. The games I want to develop should support neither hatred nor corruption of the youth. These matters are highly important for the future success of our societies as well and should not be despised carelessly.

Ethics should be considered on the business and marketing side as well. Getting rich on however cruel terms are something that I personally will avoid and also encourage others to avoid as well. There is nothing wrong with success, but in case you're taking clear advantage of others just to fill your bank account you should re-evaluate what you're doing as an Indie²³.

Personal motives like these can be only achieved through Indie game development, because the mainstream AAA studios are rarely working according to the Moral Law mentioned. It is also a form of creative freedom to be able to support basic ideas that you grant as good in your game designs.

5.2. Thoughts on designing and creative freedom

In my opinion Indie Developers should flow and try to pinpoint the quirky weird things on the edges of our culture and industry. Whenever an Indie finds some strange idea worth trying, he should note it down and if he still likes the idea after few weeks or a month, create a prototype.

It is all about standing out from the crowd and to achieve it you have to be able to think out-of-the-box. Self-learning to open doors on all aspects of life rather than closing them is going be highly beneficial for Indie game developers. Having narrow, puritan or conservative attitudes towards things is closing many doors on what you can do as a game developer and designer.

In my opinion as we are making an interactive form of art as Indie Game Developers we should be able to enjoy the Creation Process all the time and to feed our inspiration without boundaries that capitalism creates. The game designers should have the guts to stand up for their unique vision rather than compromising on everything just to please mainstream consumers^{98 99}.

However with complete freedom to design a game I want to point out that freedom comes with great responsibility as you need to make ethical and moral decisions over your content. For example, a game that supports reckless use of alcohol and drugs, ultra-violence, free sex or racism of any type just to name a few, can perhaps make your bank account big but still be ethically dubious or even dangerous to players; adults or children. Aim to send a meaningful message through your games⁹⁷.

Still in many cases the beauty is in the eye of the beholder and you should not design your games simply to please the worldly ethics and morals on the present moment due to the fact they tend to be often hypocrite methods to maximize the profit. Rather design from your heart, because it is you who should know the difference between right and wrong; not media's ever changing opinion on issues.

A proper approach taken is something you don't need to regret later and a path you can stand behind as team. The whole team should agree on the path taken and the common values supported. Also it should be clarified to every member which are the topics to avoid on designs^{100 101}.

In my comparative analysis of Postmortem data I found out that Indie projects had lesser occurrences with Bad game design problem type group than AAA projects. I honestly think that the main reason behind this positive difference is the attitudes towards the process and creative freedom. Also Indies might realize that they are making art more strongly than AAA developers, which boosts more quirky Game Designs³⁸.

5.3. Thoughts on indie business

One missing common preferred approach was very obvious and important; using professional consulting through different business guidance programs. It is important to know that there are International growth programs such as Luovimo¹⁰² in Finland for creative industry companies focusing on innovative services and content business available in most modern countries.

By joining these programs, companies can get specific business related knowledge and guidance they could not afford or reach otherwise. These programs can focus on the most critical areas for a company's success such as internationalizing or networking and help on framing a proper business plan^{102 103}.

Also I have an opinion that the commonly preferred ways related to marketing are just scratching the surface and it might require having a person with knowledge to organize the whole marketing and promotion. Having a skillful person to do marketing is really important as there is a huge amount of extra work needing to be done in the form of viral marketing, blog writings, newsletters, organizing game exhibitions.

As an Indie you should be more concerned about ethics, human rights and also environmental issues. I think it's important for Indie companies to plan their marketing strategies according to these aspects as especially in the world of business, ethics and morals are often overlooked or shamefully forgotten.

Instead of making your game fan products with child labor in third world countries, choose otherwise. Maybe you can also consider making your fan products in an ecologically sustainable way. For example instead of producing tons of plastic waste, make a little Etsy shop where you sell exquisite quality handmade toys. You can perhaps make some local handicraft artisan really happy by outsourcing your manufacturing to people around your home city or town.

Try to find other means to practice far-sighted business which do not aim only to maximize the profit as there is always room for being creative on business. This can improve your company's image and have extra value also to your brand when you get your message through to your community and players^{99 104}.

5.4. Thoughts on development

In my opinion it is important not to restrict or narrow any future possibilities when making prototypes. For example, the development can continue after the prototyping with the current technology or with completely different option. Making a decision to drop the current technology or continue with it is easier when you have not made a herculean effort on it yet in this phase^{44 45}.

Commit to your Indie community when using Open-Source engines such as WinterMute engine. In case you develop interesting new features for the engine you use, such as tools or some other form of technology; it is appreciated to let community enjoy your inventions as well. Whenever you're developing an engine of your own, you should consider making it Open-Source if possible. Not only as these actions are in line with Indie Ethics, are they also important extra advertisement for your Game Studio. These actions support the Indie community flourish around you.

In my opinion the game should be tested from the beginning and testing sessions should be constantly kept. There should also be a bug tracking system involved when organizing bug tickets starts becoming unmanageable. Team should play their own game as much as possible and a golden rule is that if they like it still at the end of the development cycle the game will be successful¹⁰⁶.

Sharing your custom made tools with Indie community or making them Open-Source is highly valued. This can give your Game Studio free publicity in the Open-Source community and help you building your company's image.

Use Git as a version controlling system instead of SVN. A good practice is to work in your local repository and branches, while keeping your remote repository always updated with the newest working and tested features. The

remote master branch (called origin) should always be available so that any team member can access the current version of the game.

I prefer to constantly make branches and never make my modifications to local master. In case I want to push something to origin, I will first make a commit in branch X and then switch to local master. In local master I first pull the latest version from the origin and only after that merge my commit from branch X with local master. After the merge has succeeded I push my changes finally to remote origin. This helps to tackle with the conflicts quite efficiently.

Making a remote repository “bare” is also good option that prevents any other Git command executing that pulls and pushes. This helps to avoid making modifications directly to the master.

Learning to use ignore lists, stashes, merging branches and reverting your commits are skills that you definitely will need in future. In case you do not enjoy working with command prompt, there are many quality Git Clients available for different operating systems.

When selecting Game Engine to use it is important to study which Version control systems it supports. You have to get version controlling working even you're just making a small game^{105 107}.

Rule of thumb: study the best approaches on implementing the agile methods, keep the agile practices that really work^{110 111 112} for your project (such as refactoring or scrum wall) and don't be afraid to discontinue using unfeasible practices^{55 108 109 113}.

I would include in the good communication also the constant learning from your team members. The hacker ethic's principle that all information and knowledge must be distributed to others helps to understand in deeper way the great benefits of sharing your skills with your fellow workers. This does not mean that everybody has to wear all hats; rather it means to have an open attitude towards knowledge sharing whenever someone needs to learn something from

you. This approach will boost developing skills of the whole team and make it stronger^{20 23 24}.

The game development practices should not be treated as something occult required to keep hidden either, instead when you found a neat way to do some specific parts of your development; write an article about it. Sharing knowledge will definitely give your game studio a nice boost as being Indie professional. Aim to maximize the hype⁵⁷.

5.5. Thoughts on project management

If the conflict or argument is really severe bottleneck and an absolute correction to project's path would be needed then voting is not good practice, because it might be based on feelings of individuals instead of knowledge. In this kind of problematic cases an acceptable level of research about the subject should be made. For example if the pipeline in development is not working it is not recommended to vote which action to take, but rather make a small comparison or analysis of the different possibilities in order to find the best approach.

In my opinion working in equal spirit with people you know and trust is a lot more fruitful than being under a messy company structure where agility and transparency might be reduced, sometimes even suppressed by shady business decisions. It is more important to get team members connected and increase bond between them than to build unnecessary walls through company hierarchies and old fashioned policies which are suppressing creativity and natural flow of process^{42 62 99 101 114}.

Taking advantage of good project management software is also important. A program such as Trello or Blossom can improve your teams working and provide an efficient way to handle agile development process. In case your team works remotely or it is important to reduce the noise levels in your open office, real-time chat software can help on constant communication. One free option is mIRC, but programs such as HipChat might also be a good option.

The matter of concern is to make communication working in between your team members.

I personally, emphasize supporting remote working on a proper scale for Indie game development as it is not a defect for the company when done properly. It sure needs some specific arrangements, but the benefits gained through this effort are long lasting.

It seems companies (practicing protestant working ethics) are not supporting and practicing remote working in a larger scale because of their lack of trust for their employees and their need of mentally controlling their employees. Where everything is counted as maximizing the profits, remote working is treated as something radical and anarchist. Companies are scared of losing the status quo if employees are given more freedom; in worst cases they see remote working even as an enemy or boogiemans⁵⁷.

This is not how Indie Game Company should run as it should practice different and more humane working ethics, described earlier. Indies together with Open source developers are leading the path for the rest of the world.

Flexibility on both remote and office working is one important part of Indie ethics and to it might be the crucial step for these ethics to spread and be more commonly practiced and accepted.

However, clear simple guidelines need to be laid on remote working which can be understood and accessed by every worker in the team. If these simple guidelines are missing, there will be later misunderstandings causing negative atmosphere and chasm inside company.

In my view providing real remote working possibility or at least proper flexibility towards it is a key to future success. New technology provides new better ways to work and the whole concept of working is on the edge of a paradigm shift; perhaps resulting in a better working ethics for all²³.

It might occur that outsourcing workers from countries that offer cheap labor could be a risky decision. Taking this path will directly impact your company's

image and reputation, you should be extra careful when making such decisions. Many people dislike the idea of using cheap workers who have random working conditions; sometimes close to modern slavery⁹⁷.

5.6. Thoughts on comparative analysis results

As it was found out in the comparative analysis of Postmortems, Indies had more difficulties in Project Management and Business areas. On the contrary the research showed that Indies had fewer difficulties in Design and Development areas. Both Indie and AAA projects had minor amount of problems in External areas.

The major differences compared to AAA projects were in Business area problems. This was no surprise as it is a common knowledge that Indies usually lack on business and marketing skills. For Indies making fun games is the most important issue and often marketing and business is overlooked or not properly studied at all. Due to limited manpower, knowledge and funds this is very understandable. There is no silver-bullet here but as introduced earlier in Literature Review and Results chapters, some feasible and efficient practices do exist^{34 42 101}.

The Project Management problems however are issues which the Indie teams could fix by getting more organized and taking correct practices into use. As proven by earlier chapters there are vast amount of information available to help on this issue^{51 62}.

However, it seems to be a commonly accepted way to work in “creative chaos mode” in Indie game development projects, but it might not always be the best way as the projects get bigger^{111 114}.

In my opinion the lesser amount of problems in Design area can be explained with creative freedom, control over IP and smaller economical risks. As the money is not usually the main focus in Indie game projects the risks can be taken more easily than in bigger AAA projects. The biggest notice here is

however the full control of the IP which leads to creativity as explained in Literature Review ^{34 58 59 99 100}.

In the Development area, the main reason for fewer problems is that projects are not as big as AAA project are. Indies also use existing technologies instead of making custom in-house tech more commonly, which can naturally also reduce many problems such as bugs.

Even the difference in this specific Type Group was enormous compared to AAA projects I decided to leave it out from deeper investigation, due to its nature of being more or less self-explanatory^{82 83 84 85 86 89}.

As the research also found out, major differences in Problem Type groups were not found in the biggest groups. Instead, the following minor Problem type groups had the most significant differences:

- Not enough content (Design)
- Too many hats (PM)
- Finding correct employees (PM)
- Launch and Post launch problems (Business)

In my opinion the first two problem type group differences are quite obvious as the Indie teams are smaller and their budgets are lower. However, the last two differences seem to have a special importance, since there are no clear technical or economic reasons why Indie projects should have more problems in these groups.

Finding correct employees might be more difficult for Indie projects for multiple reasons. Maybe Indie companies can't pay high enough salaries; maybe they don't have fancy office spaces. Developers might want to get themselves in bigger projects or the AAA companies themselves have better networks and systems to hire necessary employees.

So it is plain and simple that obtaining dedicated committed employees to ensure the success of your project and future of your company can be a

cumbersome task to tackle. This issue does not become less complicated due to the need of making sure you select the sidekicks and partners who are trustworthy and easy to work with^{67 68 69}.

It was interesting to note out that AAA projects Top 10 Success type groups chart did not contain Using Existing Tech group at all. The fact that it did not reach the Top 10 is simply that AAA studios have more budgets to create their custom engines and often they tend to make that decision. This can be clearly seen on their peak in the Creating Custom Tech Success Type group³⁸.

In the next chapters I am going to suggest a valid set of Improvement ideas for some of the more significant Type Groups noticed during research.

5.7. Improvement ideas and suggestions for indie projects

I selected three Problem Type groups and three Success Type groups for a deeper examination. All these Type groups were selected according to few important variables. The used variables were occurrence difference between Indie and AAA projects, special relationship to the topics covered in this thesis and total amount of occurrences in each Type group.

Problem Type groups selected:

- Scheduling was done poorly
- Finding new employees
- Launch and Post launch problems

Success Type groups selected:

- Team work success
- Focused vision
- Correct design decisions

5.7.1. Scheduling was done poorly

The data collected from postmortems showed that underestimation of time consumption, having unrealistic project scope and decisions to add more

content and features at the last moment caused serious problems in the Indie projects. Also because of a loose and faulty scheduling crunches were needed, even not intended.

It was surprising to note a detail here though. Instead of Indies failing more often on scheduling than AAA studios, they actually had fewer problems. Even though Indies were obviously less organized in scheduling and project-management, they still could cope better in their schedules. The main reason for this might just be the fact that projects are usually smaller, faster to develop and easier to schedule and estimate⁴². The scheduling problems were still critical concern for both Indies and AAA studios.

However, scheduling successes were achieved when Indies followed a development process model that was made from multiple short iterations; a customized agile approach. Designing for the current technology and not vice versa helped Indie projects to keep in schedules. Keeping the project inside the scope and budget was important, helping to avoid feature creep.

Completing various micro-projects on free momentums was also helping Indies on a larger scale. Schedules called roadmaps and a decision of forced delay on launching was recognized as an important compromise in order to get the game to a level of quality.

The data revealed that AAA studios were having success on scheduling when practiced accurately and/or when a realistic schedule was made. Also on the other hand flexibility was endorsed in a macro level scheduling. AAA studios tend to also eagerly analyze the reasons and react with necessary actions if a milestone was missed.

The AAA studios emphasized aligning the pipeline so the multiple parts of development such as asset creations, game prototyping and implementation and game design would work in a fluent way.

Stable production environment and availability of the complete build for all was often considered a necessity. The production structure designed to be flexible to allow establishing production cycles inside bigger milestones had also positive effect¹¹⁴.

My suggested improvements for poor scheduling in Indie projects are to take customized agile practices as a part of your development process and to optimize your pipeline by fixing the bottle necks⁵¹. Also if the team misses a milestone, a proper analysis needs to be done and correct actions taken. Indies need a bit more organizing and a bit less creative chaos mode.

There is no need for obtaining agile practices with a puritan mindset. As they are only tools and not some kind of religion a person should pick from that toolset the best practices that work. For example the scrum wall that is accessible to all team members is really efficient tool for scheduling the near future working tasks.

As there might be remote workers in Indie teams this wall is better to be accessible online. Google Drive is a good and free service that can be used. You can create the scrum wall backlog differently for each milestone and also have a miscellaneous section that can be used for sprint tasks that are currently difficult to place under any milestone in the backlog.

This approach enables a clear visibility on the progress of each milestone of the project and combines them together with the scrum backlog. It is a great way to keep things organized and to see if you're efficient as the postponing of sprint tasks will soon tell about your performing and if you will hit Milestone target or not.

Hitting your milestones on time is important and instead of having a big crunch that everyone hates in the end of the development to hit your Launch, you should aim to hit every milestone in your project. In order to hit the milestones you need to have your pipeline in order as mentioned above, your people motivated and maybe smaller crunch times at the end of each milestone.

Keeping smaller crunches in each milestone is a better tradeoff than having one big crunch in the end in my opinion.

In addition to the scrum wall and milestone also a common calendar is important for scheduling the project and to see the bigger picture. All deadlines should be on the wall, visible to everyone in the project.

Keeping your team focused on the milestones is also important as if the team loses its focus, pipeline problems and missing the deadlines will follow¹¹⁰. In order to keep employees focused on what they do, regular sprint planning meetings might be good to include in your agile toolset. The sprint planning meetings is a must have even for a minimalistic version of customized scrum. This will focus your team for the next sprint/iteration to complete certain tasks and a rule of thumb here is that 20-25 hours of backlog tasks should be included for one sprint week of employee. However if your sprints are longer than one week a scaling can be done easily to this rule.

An important guidance found from Indie postmortems was not to waste your time on sketchy designs. Nothing is more time consuming and frustrating than doing things over again, multiple times in some cases. It is really important that new features are designed in enough deep level so that only tweaking instead of complete rewriting is needed.

However sometimes it's natural to redo parts of your game such as art assets, sounds or even code optimization. This can be caused by a major budget change etc. but can still take its toll on the developers' nerves, cause extra stress and bring bad blood inside the team^{38 99 101}.

5.7.2. Finding new employees

The insight investigation to this Problem Type group revealed that being lazy on networking and researching for proper outsourcing companies can cause problems later in the project cycle. There were cases where a company needed a specific talent for an important part of the project but failed to find the worker

in a feasible time. This of course affected the schedules and the projects on a whole. For example finding quality voice acting talents was considered a difficult and time consuming process. It was emphasized that enough time needs to be prepared for evaluating these different voice actors.

The relocation of a company's office helped one Indie Company hire new talented people. In this particular case their office was in some small city where there were not enough game developers to choose from. There simply were not enough skilled developers available. The decision to relocate to a bigger city opened possibilities to hire better workers and benefit the project and the company. Outsourcing game audio was also very successful decisions to another project, as they had good experience with freelancers³⁸.

Finding new employees can be a difficult task for starting Indie companies that are struggling with their budget and cannot pay competitive salaries.

Make efforts to connect your local Indie Game scene and also game industry as whole. Networking your company and yourself to different areas of creative industry can have an enormous help later on. It is always good to know people in the industry. A good option is to work together with schools in smaller projects. This can give your studio low-cost work force for example on prototype projects and such. It can also give you peek behind the curtains of the newcomers, so you can hire the best talents directly from school.

Make your company accessible through the internet. Branding your game company to a certain level is the key to attracting new people. You should trust your instinct and vision and believe what you are doing. Focus your business vision, without doing everything at the same time.

Probably the benefits that you can provide better than bigger studios are more or less immaterial and cannot be calculated directly in money¹¹⁴.

In order to make your company more tempting to employees finding work, you should consider what special can be provided that bigger companies cannot or

are not eager to provide. You should consider supporting remote working possibilities, flexibility, proper recognition and respect for all employees together with Indie ethics and values^{23 57}.

An ideal Indie company understands that there is life outside the work and would be able to adapt to changes in life. Employees should have enough time also for family life and more control over their working, so the inhumane working times such as crunches could be avoided^{10 12 57}.

Naturally creative freedom should be supported and emphasized in your company as it is often considered important for developers in Indie Game business.

It is also important to mention that Indie game development requires you to wear multiple hats. This can be used to make developers interested on your company. Many developers have multiple skills and when they are able to participate in multiple areas of game development it can make their working life more fun. Wearing Too Many Hats should still be avoided if your aim to keep working fun for all the time.

Finally you should advertise your company as a fun place to work and remember that money alone is a really bad motivator today and also in the future^{38 54 56 58 59}.

5.7.3. Launch and Post launch problems

The research showed that Launch and Post Launch problems were more common in Indie projects than AAA projects. There are many underlying reasons. These reasons were often related to budget and money but also inexperience in marketing, dealing with the Press and Media and the whole Launch process.

However, one must understand that Indies do not have a massive machinery to handle all this promotional and marketing work and that is the biggest reason

they are sometimes performing inadequately related to this Launching and Post-production.

Using more effort in marketing, for example by making an early trailer gave Indies good results. These actions built hype around the upcoming games and raised awareness of them. This is a proof that marketing is not rocket science Indies could not manage, but rather if wise actions are made; positive results will follow. Positive response from press and audience and good reviews were mentioned also as keys to succeed on the Launch and Post-Launch period and they are partly connected on building up the hype^{39 40}.

Both Indies and AAA studios mentioned participation on special game events such as Summer of Arcade being good for promoting game and get publicity. These events are efficient ways to create hype around the game and your game studio.

The AAA studios also used the announcement trailer trick to get people interested on the project. The announcement trailer got people waiting for the game since the very beginning and was counted invaluable. Facebook was also used to build a global online fanbase for an AAA project which of course was an efficient way to share news and built hype as much as possible^{38 39}.

Still, Indies cannot often make some of the launching related actions that AAA studios can. Simultaneous worldwide release is one of them. It is probable that small Indie studios cannot organize or afford global scale business efforts like this.

To get Launch as smooth and successful as possible I claim that using enough on QA efforts, time in Social Media and promotional materials will raise your possibility of achieving success in Launch. Also Indie companies should have as many contacts from Press and media as possible and keep the connections warm constantly. These connections can be invaluable when launching the new game.

Post Production problems could be efficiently avoided by having a real Post Production budget. There were cases where Indies had not prepared sufficient budget for Post Production phase.

Post Production is highly important in terms of Customer Experience, Cross promoting and viral marketing. Many AAA projects are doing this part better. For example in case of updating your older game you can promote your upcoming titles at the same time and vice versa. This was considered a really important part of marketing as it benefits multiple of your products at the same time^{41 42}.

Updates are also important in Post Production. Typically when you release an Indie game you might not have as much content as big AAA projects do. In this case you can later on update and upgrade your game, include more content. There was cases that players actually wanted more and when Indie studio could not provide them more, the gamers left to another titles. Keeping your game updated long enough will increase positive Customer Experience and also give longer lifespan for your game in the Social Media. Viral Marketing will not work, if you don't give the players new reasons to talk about your games^{39 41 42}.

Twitter, Facebook and other Social Media services are important to take in active use. For some reason Indies tend to prefer Twitter, but the SoMe¹⁹ services are as important as the research found that AAA studios are benefitting from having a global fan-base on Facebook^{38 39}.

5.7.4. Team work success

In the postmortems, Indie projects had less Teamwork success than AAA projects³⁸.

Concerning on Indie team working there is a need for improvements and by extracting the detail level occurrences from this Success Type group, I was able to find out that when Indie teams could achieve a deeper bond with each other as a team as well as share freely knowledge between all team members, a better results followed.

Seeing the team as a band or in other words as an organic synergy was allowing this open knowledge transfer and the deeper bond between members. It was said that earlier work experience together was beneficial as the team members already knew more about their band members. Keeping the team size small was mentioned to be important; instead of a gigantic orchestra, an image of progressive rock or jazz band was resonating through the postmortems³³.

The data collected from AAA projects' team working successes gave insights on how they are performing better than Indies. For example sharing resources between other projects of the company was mentioned to be effective. However, as the Indie companies are small and they tend to do one project on time typically this cannot be directly applied to Indies. Something similar can be achieved for Indies by creating larger scale networks and portals that provide freelancer connections and high quality open-source resources such as assets, plugins, modules, sounds etc. One perfect example is freesound.org website which provides free sound assets in different formats.

Smaller “strike teams” were also used in bigger AAA projects to cope with various difficult problems. This approach is also quite infeasible for small Indie teams but if the Indie teams get bigger the method could be applied. However networking with other developers will help Indies tackle some more specific problems as there is always someone in your network that has the know-how or can guide you in the correct way.

It was also suggested in the AAA postmortems how important the communication and agile methods supporting it are. It seemed to be a rule of thumb to minimize your documentation and the time needed to keep them updated and instead practice active communication.

In my view, communication is a necessity for every team. The company should support open discussion, commenting and opinions. Transparency and openness inside a company and between teams is also important. Roadmaps and future plans and visions should be freely shared and discussed in order to create team spirit and to support focused development and promotion⁷¹.

Lastly, AAA postmortem data included information that embraced the determination of leadership. A good leader was said to be a person who can keep the morale and productivity high during the project. Keeping focus was mandatory and should be applied not only to development itself^{54 56} but also to job interviews, where approach measuring applicants with alternative variables such as how they would fit into the team, instead of skills only was preferred³⁸.

My opinion is that to avoid conflicts in the team, members should practice subtle ways of commenting and giving opinions. Criticism is mandatory of course and when giving this "negative" feedback, the way it is presented is important. When giving criticism for a certain design or implementation always give clear reasons why something does not work out and if possible open a discussion around the problem in a good manner. Nothing can ruin your team work faster than unjust or harsh comments and opinions. It will benefit the development if team members possess healthy values and morals.

A certain "lone wolf" attitude might be rooted in the mindset of Indies and sometimes also artistic egos can become an issue. Especially with highly creative people receiving negative feedback can become a stumbling stone as all creations might be representations of your artistic personality and ego. However, each team member must learn to receive criticism and negative feedback without taking it personally. This does not mean that feedback can be given recklessly as mentioned above⁶⁹.

Avoiding position titles such as "lead" or "senior" can help. Basically you need to work as a band of brothers and sisters where each member of the team is important. As everyone in the team feels that his work and skills are valued in the company an unhealthy competition inside the team is no longer needed. Team members should not raise themselves higher or bring others lower, everyone is important^{67 68 70}.

Sometimes Indie teams can have remote workers involved and they should not be left out from the design discussions or brainstorming sessions. Use

microphones, speakers and video connection to contact the remote workers during the meetings. Provide necessary connections such as VPN or cloud based services so that remote workers have access all the time to newest project data and information related.

Especially when remote workers are involved a high level of transparency and openness must be provided also towards them. These decisions are important as they connect your whole team together and help to establish a team spirit, so that team "can move mountains" together.

5.7.5. Focused vision

The Research found out Indies had fewer successes on having a focused vision than AAA studios. Indies succeeded keeping focus usually by being aggressive on the artistic vision, trusting their instincts, not changing easily the original vision, identifying key elements and designing set of rules for the designs.

However AAA studios were eager to set clear goals and having realistic expectations. This helped AAA studios to focus on the strengths of their team and game concept^{54 55}. There was also a case where a AAA studio created a "diorama" that was giving birth to their game design and which was their primary reference on how the game should turn out in the end.

They also used a technique called "design by playing" that let them evaluate the features constantly; helping them to keep only ones that worked. I think this approach could be used also in Indie Game Development more frequently³⁸.

Focusing on the core mechanics as well as first priority features and content helped them to get a firm basis for later development³⁴. There was also a case where storyline was created by a committee to avoid plot holes. I think it was a great idea.

To have more focused vision, project should have only one Game Designer. Decision to have only one Game Designer does not mean communication,

opinions and open discussion should not be company's principles. What it means is that in the end it's still up to Game Designer what decisions related to game designs and features are made.

This also removed the need for voting or other obscure methods that might cause the team to lose focus. Nevertheless I support creating the storyline together as mentioned earlier. The feelings or current moods of the team members cannot affect so randomly the game design progress when the strings are on the hands of one focused Game Designer³⁴.

Also when vision is focused, amount of sketchy designs is decreased as there is only one person concentrating on every design. As he is basically responsible on each design decision, he must make sure and believe in his intuition on every decision that he makes maximizes the fun.

Early play and focus group testing can also keep Game Designer focused on the core concepts of the game. These tests will tell Game Designer which features will add more fun for the target audience^{38 99 100 101}.

5.7.6. Correct design decisions

AAA studios had performed better on making correct design decision during the development cycle. There was numerous interesting decisions been made which I am going to present some of the best ones.

It seems the greatest surprise here was the case where AAA Studio took multiple risks which all turn out to be worth taking. The interesting detail here is that this occurrence was not in Indie successes but in AAA successes. Risk taking is emphasized often in Indie development related articles and on the other hand AAA studios are commonly blamed of playing it too safe³⁴.

Outsourcing was considered a worthwhile decision as well as stripping out features such as online gameplay. Decision to make a fast and efficient pipeline

for artists and integrating multiplayer mode directly to original design were counted as successes also.

Case where making storyline optional and not giving up on a good feature which was difficult to implement were also decisions need to not regret. Some of the decisions were beneficial compromises which however did not support creative freedom; one example was a game design that was very similar to the prequel. Making such compromises might be important in business term, especially when you feel the original design does not need huge changes. I can think about the old Sierra adventure games that were almost similar to each other in terms of technology and game mechanics but still being unique works of art.

I think it is not mandatory for a good game always include “fresh” gameplay features or technical innovations; but I still understand those who think the opposite. However I believe creating re-usable game technology, such as controls and mechanics can be important surplus for any Indie, but definitely not an issue which you should frame your game designs³⁴.

Correct Design Decisions in the Indie projects were mostly gathered around creative freedom, re-doing sloppy art, exploring the concept, cutting content away or supporting minimalistic designs and focusing on reusable features. A decision to set the gameplay and controls as a foundation of the whole game design was counted very helpful. The idea here was to let game tell what it wants to become. This approach is really fundamentally agile in my view^{34 101}.

As the Play and Focus group tests are important for Game Designer to keep focused vision, it is as important to make correct design decisions.

When deciding on features to implement Game Designer has to know those features that give maximum fun, with low cost. In order to analyze and estimate the costs of each new feature a close team work with other artists and programmers need to keep¹⁰⁰.

For example some feature that would improve fun might be very costly to make in practice but still would not improve fun for target audience than some other low cost feature. If there are any quick wins to please your target audience, make sure you use them efficiently.

It is also necessary for a Game Designer to be able to fail fast; to let go of a design that probably won't work out. A proper prototype will give insight if the design is enough good to be taken into production or if there is no need to continue further^{38 99}.

6. Conclusion

In the beginning of this research the information of Indie development was fragmented on multiple sources such as articles, writings and postmortems. During research I collected vast amount of information which was analyzed. In comparative analysis of postmortems I also found out that there were major differences in some parts of the indie development compared to AAA development.

These differences such as finding new employees, launch and post launch problems, team work success, focused vision and correct design decisions were then discussed and reasons behind them were presented. As the Indies cannot compete in many areas with AAA studios that have a lot of money and funds available, Indies have to be witty and creative in terms of handling their development and business.

To be aware of the best design and development approaches is crucial to Indies, but it is not usually enough to compete with AAA studios. To compete and stand out, Indies have to practice better ethics than AAA studios. Indie ethics that derive from hacker ethics seems to become a crucial corner stone on competing with multi-million dollar AAA companies in the future.

Indie Game Development is recognized as a combination of technical skills, creativity and ethics. In order to make a living out of Indie Game Development one must possess multiple skills and have an open mind to multiple aspects of life.

By putting more efforts into project management, pipelines and the development process, Indies can improve their ways of working. Using simple, ethical and efficient marketing strategies will help them stabilize their business so global companies and investors cannot take advantage of them easily.

Indies who practice creative freedom, passionate developing and personal intuition in sharing their vision to the world are modern game culture patriots. Their source of motivation is not primarily material, instead it comes from passion on creating and life itself.

The public is slowly beginning to recognize Indie Games as invaluable experiences as well as a form of Art. Time will tell how publicity and the growing requirements from the mainstream audience will affect the many beautiful and high virtues and principles of the Indies.

References

1. Gamasutra Staff. 2011. Indie Game History You Never Knew (http://www.gamasutra.com/view/news/125185/Feature_Indie_Game_History_You_Never_Knew.php#.UHNROk3MiyU)
2. Parkin, S. 2011. The Story of ANMA: Two Schoolboys And a Game Development Dream (http://www.gamasutra.com/view/feature/6394/the_story_of_anma_two_schoolboys_.php)
3. Bone. 1997. John Carmack Archive – interviews (<http://fd.fabiensanglard.net/doom3/pdfs/johnc-interviews.pdf>)
4. Documentary of Id Software. 2011. – How it happened (<http://www.geek.com/articles/games/id-softwares-history-now-available-as-youtube-documentary-20110320/>)
5. Litten, M. 2012. Humble Bundle V is Like the Avengers of Modern Indie Gaming (<http://www.vgbblogger.com/humble-bundle-v-is-like-the-avengers-of-modern-indie-gaming/15725/>)
6. Urban Dictionary. 2012. Definition for Indie (<http://www.urbandictionary.com/define.php?term=indie>)
7. Dutton, F. 2012. What is Indie? (<http://eurogamer.net/articles/2012-04-16-what-is-indie>)
8. Rosen, D. 2009. What are Indie games? (<http://blog.wolfire.com/2009/08/what-are-indie-games/>)
9. Game Zephyr. 2012. EA's Indie Bundle Receiving some hostility (<http://gamezephyr.com/en/2-uncategorised/176-ea-s-indie-bundle-receiving-some-hostility.html>)
10. Steven. 2011. When it comes to the crunch – Unpaid overtime in the game industry (<http://libcom.org/library/when-it-comes-crunch-unpaid-overtime-games-industry>)
11. MadModMike. 2012. Working Environment for a Development Team (<http://truthofgaming.blogspot.fi/2012/05/working-environment-for-game-developer.html>)
12. Tanner, N. 2011. The Real Housewives of game development (<http://m.ign.com/articles/2011/05/02/editorial-the-real-housewives-of-game-development>)
13. Grayson, N. 2012. Opinion: Why 'Indie' Has Become A Bad Word (<http://www.rockpapershotgun.com/2012/05/03/why-indie-has-become-a-bad-word/>)

14. Bobic of 4sceners & Ghandy of Moods Plateau. 2012. The Demoscene and modern games (<http://zine.bitfellas.org/article.php?zine=13&id=19>)
15. Lopez, M. 2012. Demos in the Eye of a Games Journalist (<http://zine.bitfellas.org/article.php?zine=13&id=22>)
16. The Technium. 2008. 1000 True Fans (http://www.kk.org/thetechnium/archives/2008/03/1000_true_fans.php)
17. Graft, K. 2010. Nourishing Your Indie Community (http://www.gamasutra.com/view/news/118537/GDC_Nourishing_Your_Indie_Community.php)
18. Indie Game Festival. 2012. (<http://igf.com/>)
19. Indiecade – International Festival for Independent Games. 2012. (<http://www.indiecade.com/2012/conference/>)
20. Webpage. 1996. The New Hacker's Dictionary (http://www.outpost9.com/reference/jargon/jargon_toc.html#SEC22)
21. Britannica. 2012. Protestant Ethics (<http://www.britannica.com/EBchecked/topic/479867/Protestant-ethic>)
22. Burkeman, O. 2010. This column will change your life: The Protestant work ethic (<http://www.guardian.co.uk/lifeandstyle/2010/sep/11/pain-gain-work-ethic-burkeman>)
23. Himanen, P. 2001. The Hacker Ethic and the Spirit of the Information Age
24. Mizrach, S. Is There a Hacker Ethic for 90s Hackers? (<http://www2.fiu.edu/~mizrachs/hackethic.html>)
25. Cox, K. 2012. PC Gaming Studio said she ruined their game but only after she sued the boss for sexual harassment (<http://kotaku.com/5940401/pc-gaming-studio-said-she-ruined-their-game-but-only-after-she-sued-the-boss-for-sexual-harassment>)
26. Lumb, J. 2006. Itagaki Trial – Sexual Harassment (<http://www.1up.com/news/itagaki-trial-sexual-harrassment>)
27. Hao, W.D. 2003. Postmortem: Ubisoft's Tom Clancy's Splinter Cell (http://www.gamasutra.com/view/feature/2830/postmortem_tom_clancys_splinter.php)
28. Vanity Fair. 2012. Microsoft's Downfall: Inside the Executive E-mails and Cannibalistic Culture That Felled a Tech Giant (<http://www.vanityfair.com/online/daily/2012/07/microsoft-downfall-emails-steve-ballmer>)
29. Bernacki, E. 2002. Exactly what is "Thinking outside the Box"? (http://www.canadaone.com/ezine/april02/out_of_the_box_thinking.html)

30. Matulef, J. 2012. Mind-bending puzzler Perspective emerges from Portal alma mater Digipen (<http://www.eurogamer.net/articles/2012-06-21-mind-bending-puzzler-perspective-emerges-from-portal-alma-mater-digipen>)
31. Cook, D. 2010. Why Indie games are good for fans (<http://www.lostgarden.com/2010/02/steambirds-why-indie-games-are-good-for.html>)
32. Bordeu, C. 2009. Postmortem: ACE Team's Zeno Clash (http://www.gamasutra.com/view/feature/4156/postmortem_ace_teams_zeno_clash.php)
33. Adam. 2012. Indie Experience Points – Creative Freedom in Indie Game Dev (<http://egamer.co.za/2012/06/indie-experience-points-creative-freedom-in-indie-dev/>)
34. Pajot, L., director/producer; Swirsky, J., director/producer, Directors/Producers. 2012. Indie Game – The Movie (<http://vimeo.com/indiegame>)
35. Waisgluss, I. 2012. Eye on Culture – Indie Games (<http://kaleidoscopeflux.blogspot.fi/2012/06/review-snakes-on-cartesian-plane.html>)
36. Cantoni, C. 2012. Interview – Terry Cavanagh (http://www.indiecade.com/interviews/terry_cavanagh/)
38. Appendix I (Postmortems list and data grouping table)
39. Hangartner, J. 2011. Indie Game Marketing: Article I – Social Marketing (http://www.gamasutra.com/blogs/JeffHangartner/20110830/90107/Indie_Game_Marketing_ARTICLE_I__Social_Marketing.php)
40. Hangartner, J. 2011. Indie Game Marketing: Article II – Traditional Marketing (http://www.gamasutra.com/blogs/JeffHangartner/20110902/8350/Indie_Game_Marketing_ARTICLE_II__Traditional_Advertising.php)
41. Hangartner, J. 2011. Indie Game Marketing: Article III – Game Related & Maintenance (http://www.gamasutra.com/blogs/JeffHangartner/20110907/8386/Indie_Game_Marketing_ARTICLE_III__Game_Related__Maintenance.php)
42. Mannonen, J. 2012. Four successes in a row can't be a coincidence - secrets of Mountain Sheep - Assembly 2012 ArtTech Seminar (<http://archive.assembly.org/2012/seminars/four-successes-in-a-row-can-t-be-a-coincidence-secrets-of-mountain-sheep>)
43. Gray, K. 2005. How to Prototype a game in under 7 Days (http://www.gamasutra.com/view/feature/2438/how_to_prototype_a_game_in_under_7_.php)
44. Intermediaware. 2011. Why Prototyping is important is so important in Game Development (<http://intermediaware.com/blog/why-prototyping-is-so-important-in-game-development>)

45. Noel. 2010. Prototyping: You're (probably) Doing it Wrong (<http://gamesfromwithin.com/prototyping-youre-probably-doing-it-wrong>)
46. Mullich, D. 1997. Milestones and Glass Houses (http://www.gamasutra.com/view/feature/131599/milestones_and_glass_houses_.php)
47. Browncoat, J. 2011. Ultima Game Developer: Milestones (<http://lycaeum.ultimaaiera.com/milestones/>)
48. Keith, C. 2008. Beyond Scrum – Lean and Kanban for Game Developers (http://www.gamasutra.com/view/feature/3847/beyond_scrum_lean_and_kanban_for_.php)
49. Clinton, K. 2007. Scrum and Long Term Project Planning for Video Games (http://www.gamasutra.com/view/feature/3142/scrum_and_long_term_project_.php)
50. Demachy, T. 2003. Extreme Game Development (http://www.gamasutra.com/view/feature/131236/extreme_game_development_right_on_.php)
51. Esmurdoc, C. 2010. Postmortem: Double Fine's Brutal Legend (http://www.gamasutra.com/view/feature/132696/postmortem_double_fines_brutal_.php)
52. Cage, D. 2006. Postmortem: Quantic Dream's Indigo Prophecy (http://www.gamasutra.com/view/feature/2731/postmortem_indigo_prophecy.php)
53. Cheng, J. 2006. Postmortem: Klei Entertainments's Eets: Hunger. It's emotional (http://www.gamasutra.com/view/feature/2687/postmortem_klei_entertainments_.php)
54. MindTools. Locke's Goal Setting Theory (http://www.mindtools.com/pages/article/newHTE_87.htm)
55. Austin, A. 2004. Postmortem: Chronic Logic's Gish (http://www.gamasutra.com/view/feature/2176/indie_postmortem_chronic_logics_.php)
56. Lunenburg, F.C. 2011. Goal Setting – Theory of Motivation (<http://www.nationalforum.com/Electronic%20Journal%20Volumes/Lunenburg,%20Fred%20C.%20Goal-Setting%20Theoryof%20Motivation%20IJMBA%20V15%20N1%202011.pdf>)
57. Chacon, S. 2011. The Business of Open Source (<http://vimeo.com/reaktorfi/businessofopensource>)
58. Zhang, P. 2009. Motivations in Open Source Software communities – The Mediating Role of Effort intensity and Goal commitment (http://www.academia.edu/984374/Motivations_in_open_source_software_communities_The_mediating_role_of_effort_intensity_and_goal_commitment)

59. Krogh von, G., Haefliger, S., Spaeth, S., Wallin M.W. 2012. Carrots and Rainbows (<http://sspaeth.de/uploads/CarrotsAndRainbows.pdf>)
60. Conditt, J. 2012. Five Legendary Indie Developers Walk into a room (<http://www.joystiq.com/2012/03/09/five-legendary-indie-developers-walk-into-a-room/>)
61. San Filippo, A. 2012. Going Indie – 5 Lessons I learned from AAA (http://www.gamasutra.com/blogs/AaronSanFilippo/20120615/172495/Going_Indie_5_lessons_I_learned_from_AAA.php)
62. Richards, T. 2009. Indie Game Developer's Guide – Team Building (http://www.indiezen.org/index.php?option=com_content&view=article&id=119:indie-game-developers-guide-team-building&catid=36:tonys-blog&Itemid=59)
63. Culp, P. 2012. The Secondary Costs of Outsourcing (http://www.gamasutra.com/blogs/PaulCulp/20120221/91139/The_Secondary_Costs_of_Outsourcing.php)
64. Nutt, C. 2011. Virtuous Setting the Record Straight On Outsourcing (http://www.gamasutra.com/view/feature/134696/virtuos_setting_the_record.php)
65. Dunning, T. 2009. Small Developers Minimizing risks in Large Productions (http://www.gamasutra.com/view/feature/132583/small_developers_minimizing_risks.php)
66. Henson, G. 2009. Working Remotely: Pros and Cons (<http://www.guahanweb.com/2009/11/16/working-remotely-pros-and-con/>)
67. Boyd, R. 2012. So you Want to be an Indie Game Developer (http://www.gamasutra.com/blogs/RobertBoyd/20120214/91082/So_You_Want_to_Be_an_Indie_Game_Developer.php)
68. Doulin, A. 2010. Building a Strong Indie Game Development Team (http://www.gamasutra.com/blogs/AlistairDoulin/20100107/4040/Building_A_Strong_Indie_Game_Development_Team.php)
69. Grazier, P. 1997. Conflict (http://www.teambuildinginc.com/article_conflict.htm)
70. Martin R. & Hixson, J. Personality and the Team – Value the Person (http://www.teambuildinginc.com/article_mbti.htm)
71. Nair, T. 2010. Importance of Teamwork in the workplace (<http://www.buzzle.com/articles/importance-of-teamwork-in-the-workplace.html>)
72. Keable, R. 2011. An Indie Guide to Quality Assurance Testing and Procedures (http://www.gamasutra.com/blogs/RyanKeable/20110720/8028/An_Indie_Guide_to_Quality_Assurance_Testing_and_Procedures.php)
73. Wilson, D. 2009. Quality Assurance: A Methodology for Wide-Spectrum Game Testing

(http://www.gamasutra.com/view/feature/132398/quality_quality_assurance_a_.php)

74. Rees, E. & Fryer, L. 2003. IGDA: Best Practices in Quality Assurance / Testing (http://www.igda.org/sites/default/files/IGDA_Best_Practices_QA_1.pdf)

75. Fristrom, J. 2003. Production testing and Bug Tracking (http://www.gamasutra.com/view/feature/131238/production_testing_and_bug_tracking.php)

76. Dunn, T. 2012. Focus group vs. usability testing – what, when and why? (<http://www.webcredible.co.uk/user-friendly-resources/web-usability/focus-vs-testing.shtml>)

77. Donovan, T. 2011. Focus Groups, Testing, And Metrics: Developers Speak (http://www.gamasutra.com/view/feature/134870/focus_groups_testing_and_.php)

78. Keable, R. 2011. An Indie Guide to Quality Assurance Testing and Procedures (http://www.gamasutra.com/blogs/RyanKeable/20110720/8028/An_Indie_Guide_to_Quality_Assurance_Testing_and_Procedures.php)

79. Thang, J. 2012. The Tough Life of a Games Tester (<http://www.ign.com/articles/2012/03/29/the-tough-life-of-a-games-tester>)

80- Hefner, T. 2007. Linux Game Development part1 (http://www.gamedev.net/page/resources/_/technical/game-programming/linux-game-development-part-1-r2372)

81. Marsh, D. 2008. Nine Paths to Indie Game Greatness (http://www.gamasutra.com/view/feature/131952/nine_paths_to_indie_game_gr_eatness.php)

82. Gloobus. 2012. Game Development using free tools part 1 (<http://gloobus.net/game-development-using-free-tools-part-1/>)

83. Gloobus. 2012. Game Development using free tools part 2 (<http://gloobus.net/game-development-using-free-tools-part-2/>)

84. Itterheim, S. 2012. The Game Engine Dating Guide: How to find the right engine for your game (<http://www.learn-cocos2d.com/2012/05/the-game-engine-dating-guide-how-to-find-the-right-engine-for-your-game/>)

85. Stead, C. 2009. The 10 best Game Engines of this generation (<http://www.ign.com/articles/2009/07/15/the-10-best-game-engines-of-this-generation>)

86. Vaggelis, G. 2011. The Best Engines for Indie Game Developers (<http://nuverian.net/2011/01/17/the-best-game-engines-for-indie-game-developers/>)

87. King, Y. 2011. Why on earth would we write our own game engine (<http://www.altdevblogaday.com/2011/12/17/why-on-earth-would-we-write-our-own-game-engine/>)

88. Caoili, E. 2012. Game Engine Torque 3D will soon be free open source (http://www.gamasutra.com/view/news/177431/Game_engine_Torque_3D_will_soon_be_free_open_source.php)
89. Bartley, E. 2011. Opinion: Essential Tools for Indie Developers (http://www.gamasutra.com/view/news/124678/Opinion_Essential_Tools_For_Indie_Developers.php)
90. Stanton, P. 2008. To Version Control or not (<http://boagworld.com/dev/to-version-control-or-not/>)
91. Paraev, A. 2009. The Ultimate Guide to Version Control for Designers (<http://sixrevisions.com/project-management/the-ultimate-guide-to-version-control-for-designers/>)
92. Zamojc, I. 2012. Using Version Control with Unity3d (<http://mobile.tutsplus.com/tutorials/game-engine/using-version-control-with-unity3d/>)
93. Spolsky, J. 2000. Painless Bug Tracking (<http://www.joelonsoftware.com/articles/fog0000000029.html>)
95. Jimmy Drago. 2012. Cloud Based Bug Tracker: Tips for Choosing the Best Tracking System (<http://ezinearticles.com/?Cloud-Based-Bug-Tracker:-Tips-for-Choosing-the-Best-Tracking-System&id=7243179>)
96. Flanakin, M. 2011. Compare Web Trackers (<http://www.michaelflanakin.com/articles/CompareWebTrackers.aspx>)
97. Lewis, C.S. 1952. "Mere" Christianity
98. ADD Opinion: Indie Game Design Do-s and Don't-s: A Manifesto
99. Walter, R. 2012. 10 Pitfalls to avoid when making your own Indie Game (<http://www.iphonegametutorials.com/2012/01/11/10-pitfalls-to-avoid-when-making-your-own-indie-game/>)
100. Doolwind's Game Coding Blog. 2010. Fun over Features – Manifesto for Agile Game Development (<http://www.doolwind.com/blog/fun-over-features-manifesto-for-agile-game-development/>)
101. Canfield, C. 2005. The 6 Indie Mistakes (<http://www.kuro5hin.org/story/2005/7/28/235328/642>)
102. Finpro homepage. 2012. Business Consulting (<http://www.finpro.fi/web/english-pages>)
103. AppCampus homepage. 2012. Business Program (<http://appcampus.fi>)
104. Hangartner, J. 2011. Indie Game Marketing: Article IV – Psychology (http://www.gamasutra.com/blogs/JeffHangartner/20110915/8445/Indie_Game_Marketing_ARTICLE_IV__Psychology.php)
105. Fiorito, J. 2008. Postmortem: Imsomniacs's Ratchet & Clank Future: Tools of Destruction

(http://www.gamasutra.com/view/feature/132282/postmortem_insomniacs_ratchet_.php)

106. Schaefer, E. 2000. Postmortem: Blizzard's Diablo II
(http://www.gamasutra.com/view/feature/3124/postmortem_blizzards_diablo_ii.php)

107. White, S. 2002. Postmortem: Naughty Dog's Jak and Dexter: the Precursor Legacy
(http://www.gamasutra.com/view/feature/2985/postmortem_naughty_dogs_jak_and_.php)

108. Fowler, M. 1999. Refactoring: Improving the Design of Existing Code

109. Mäkelä, J. & Peurala, V. 2011. API Design done right
(<http://reaktordevday.fi/2011/slides/ApiDesign-ReaktorDevDay.pdf>)

110. Miller, P. 2008. Top 10 pitfalls using Scrum Methodology for Video Game Development
(http://www.gamasutra.com/view/feature/3724/top_10_pitfalls_using_scrum_.php)

111. Cohn, M. 2008. Agile and Scrum for Video Game Development
(<http://www.mountangoatsoftware.com/presentations/agile-and-scrum-for-video-game-development>)

112. Noel. 2004. Dealing with chaos in the real world
(<http://gamesfromwithin.com/agile-game-development-dealing-with-chaos-in-the-real-world>)

113. Swan, C. 2010. Postmortem: Blitz Games' Droplitz
(http://www.gamasutra.com/view/feature/132623/postmortem_blitz_games_droplitz.php)

114. Doolwind's Game Coding Blog. 2009. Guide to Becoming an Independent Game Developer (<http://www.doolwind.com/blog/guide-to-becoming-an-independent-game-developer/>)

APPENDIX I (Postmortems list and data grouping table)

Indie Postmortems:

1. ACE Team's Zeno Clash
2. Bane Games' Flick Buddies
3. Blitz Games' Droplitz
4. Chronic Logic's Gish
5. Frozenbyte's Trine
6. Gaijin Games' BIT.TRIP BEAT00
7. Irrational Games' System Shock 2
8. Klei Entertainments's Eets: Hunger. It's emotional
9. Mediatonics' Monsters (probably) stole my princess
10. Mind Control's Oasis
11. Mommy's Best Games' Explosionade
12. Mommy's Best Games' Weapon of Choice
13. Over the top games' NyxQuest: Kindred Spirits
14. Ray Ardent's Science Ninja
15. Reflexive's Wik & The Fable of Souls
16. Ronimo Games' Swords & Soldiers
17. Tale of Tales' The Graveyard
18. Tale of Tales' The Path
19. Twisted Pixel's Splosion Man
20. Wadjet Eye's The Blackwell Converge

AAA Postmortems:

1. 2K Boston / 2K Australia's BioShock
2. 8Monkey's Darkest of Days
3. Avalanche Studios' Renegade Ops
4. Bioware's Baldur's Gate II
5. Bioware's Neverwinter Nights
6. Blizzard's Diablo II
7. Bohemia Interactive Studio's Operation Flashpoint
8. Capcom / GRIN's Bionic Commando Rearmed
9. Capcom's Okamiden
10. Double Fine's Brutal Legend
11. Insomniacs's Ratchet & Clank Future: Tools of Destruction
12. Lionhead Studio's Black & White
13. Monolith's No one lives forever
14. Naughty Dog's Jak and Dexter: the Precursor Legacy
15. Pipeworks Software's Deadliest Warrior
16. Quantic Dream's Indigo Prophecy
17. Sega/Other Ocean's Super Monkey Ball 2
18. Square Enix's The World Ends With You
19. Ubi Soft's Tim Clancy's Splinter Cell
20. Vicious Cycle's Matt Hazard: Blood Bath and Beyond

PROBLEM OCCURENCES DATA (INDIE)				
INDIE	Problems	occur.	% of group	% from all
DESIGN	Bad Game Design	12	41,38 %	11,88 %
	Not enough Content	7	24,14 %	6,93 %
	Bad Design Decisions	5	17,24 %	4,95 %
	Too Much Content	3	10,34 %	2,97 %
	Design Process Related	2	6,90 %	2,97 %
		total	29	
PROJECT MANAGEMENT	Scheduling was done poorly	11	39,29 %	10,89 %
	Too many hats	6	21,43 %	5,94 %
	Finding correct employees	4	14,29 %	3,96 %
	Other	7	25,00 %	6,93 %
		total	28	
MARKETING	Launch and Post Launch problems	7	35,00 %	6,93 %
	Bad Promoting and Marketing	6	30,00 %	5,94 %
	Target market not clear	3	15,00 %	2,97 %
	Other	4	20,00 %	3,96 %
		total	20	
DEVELOPME NT	Bad Production decisions	8	40,00 %	7,92 %
	Technical Problems	8	40,00 %	7,92 %
	Inexperience of development team	4	20,00 %	3,96 %
		total	20	
EXTERNAL	Delays related to Platform & Stakeholder procedures	2	50,00 %	1,98 %
	Piracy	2	50,00 %	1,98 %
		total	4	

TOTAL OCCURENCES 101

SUCCESS OCCURENCES DATA (INDIE)				
INDIE	Successes	occur.	% of group	% from all
DESIGN	Good Game Design	22	53,66 %	20,75 %
	Correct Design decisions	8	19,51 %	7,55 %
	Focused Vision	6	14,63 %	5,66 %
	Enough Content	3	7,32 %	2,83 %
	Other	2		
		41		
PROJECT MANAGEMENT	Correct Management Decisions	8	32,00 %	7,55 %
	Team work	7	28,00 %	6,60 %
	Development Model related	6	24,00 %	5,66 %
	Scheduling was successful	3	12,00 %	2,83 %
	other	1	4,00 %	0,94 %
	total	25		
MARKETING	Promotion related	7	50,00 %	6,60 %
	Good Results / Outcome	5	35,71 %	4,72 %
	Investments on Marketing	2	14,29 %	1,89 %
	total	14		
DEVELOPME NT	Using Existing Tech	10	43,48 %	9,43 %
	Successful Development Practices	9	39,13 %	8,49 %
	Correct Tech Decisions	2	8,70 %	1,89 %
	Creating Custom Tech	2	8,70 %	1,89 %
	total	23		
EXTERNAL	Fan related	2	66,67 %	1,89 %
	help from outside	1	33,33 %	0,94 %
	total	3		

TOTAL OCCURENCES 106

**PROBLEM OCCURENCES DATA
(AAA)**

AAA	Problems	occur.	% of group	% from all
DESIGN	Bad Game Design	15	45,45 %	14,85 %
	Bad Decisions	10	30,30 %	9,90 %
	Work load	5	15,15 %	4,95 %
	Too Much Content	3	9,09 %	2,97 %
	total	33		
PROJECT MANAGEMENT	Scheduling was done poorly	14	58,33 %	13,86 %
	Communication	4	16,67 %	3,96 %
	Understaffing	4	16,67 %	3,96 %
	Pitching	2	8,33 %	1,98 %
	total	24		
MARKETING	Bad Promoting and Marketing	6	66,67 %	5,94 %
	Launch and Post Launch problems	3	33,33 %	2,97 %
	total	9		
DEVELOPME NT	Technical Problems	11	37,93 %	10,89 %
	Bad production decisions	11	37,93 %	10,89 %
	Massive work load	4	13,79 %	3,96 %
	Inexperience of development team	3	10,34 %	2,97 %
	total	29		
EXTERNAL	Competition	2	33,33 %	1,98 %
	Lack of control over product	1	16,67 %	0,99 %
	Lawsuits	1	16,67 %	0,99 %
	Other	2	33,33 %	1,98 %
	total	6		

TOTAL OCCURENCES 101

**SUCCESS OCCURENCES DATA
(AAA)**

AAA	Successes	occur.	% of group	% from all
DESIGN	Good Game Design	17	42,50 %	16,83 %
	Correct Design Decisions	11	27,50 %	10,89 %
	Focused Vision	9	22,50 %	8,91 %
	Enough Content	3	7,50 %	2,97 %
	total	40		

PROJECT MANAGEMENT	Team work	12	37,50 %	11,88 %
	Correct Management Decisions	6	18,75 %	5,94 %
	Development Model related	6	18,75 %	5,94 %
	Scheduling was successful	5	15,63 %	4,95 %
	Other	3	9,38 %	
	total	32		

MARKETING	Promotion related	4	66,67 %	3,96 %
	Good Results / Outcome	2	33,33 %	1,98 %
	total	6		

DEVELOP M E N T	Successful Development Practices	10	50,00 %	9,90 %
	Creating Custom Tech	6	30,00 %	5,94 %
	Correct Tech Decisions	3	15,00 %	2,97 %
	Other	1	5,00 %	0,99 %
	total	20		

EXTERNAL	Feedback	1	33,33 %	0,99 %
	Fan related	1	33,33 %	0,99 %
	Help from outside	1	33,33 %	0,99 %
	total	3		

TOTAL OCCURENCES 101

APPENDIX II (Focused Guideline for smaller indie projects)

The following guideline presents a focused listing of important practices. In this list, I do not repeat the theory or reasons behind selecting them; rather only define a list that would be benefiting Indies and that could be taken into use also in small teams.

This focused guideline is based mainly on the information provided in chapters:

- 2.2.4. Ethics and values
- 2.3.1 Out-of-the-box thinking
- 2.3.2. Creative freedom
- 2.4.1 Prototyping
- 2.4.2. Milestones
- 2.4.3. Development process models
- 2.4.5. Multitalented developers
- 2.4.8. Team work
- 2.4.9. Quality assurance
- 2.5.1. Free or affordable tools
- 2.5.2 Game engines
- 2.5.3. Version controlling
- 2.5.4. Bug trackers
- 4.1. Project management results
- 4.2. Business results
- 4.3. Development area results
- 4.4. Design area results
- 5.1. Thoughts on indie culture, lifestyle and ethics
- 5.2 Thoughts on designing and creative freedom
- 5.3. Thoughts on indie business
- 5.4. Thoughts on development side
- 5.5. Thoughts on project management
- 5.7. Improvement ideas and suggestions for indie projects

Focused Guideline:

1. Use only the agile methods that fit to your project
2. Schedule your development into bigger Milestones
3. Take Version controlling (Git preferably) into use from the beginning
4. Use existing technologies
5. Do a lot of QA constantly and organize focus group testing sessions
6. Use bug reporting tool or shared document for bug reports
7. Develop immediately for your target device(s)
8. Support active communication instead of massive GDD
9. Support immaterial work benefits
10. Support High Fun / Low Cost designs and features
11. Fail Fast prototypes that do not work out

12. Design to fit tech available, not vice versa
13. Avoid perfectionism, but keep high level of quality (compromises)
14. Prepare Post Launch budget to keep your product updated
15. Do games that are truly fun, so your products promote themselves
16. Check your course as a team often enough
17. Pinpoint and solve the issues that are causing bottlenecks
18. Avoid unethical profit gaining
19. Have a small efficient team that cover all core skills
20. Make decisions that do not conflict with your company ethics and values

APPENDIX III (“Monx in Mandala” prototype’s decisions)

Our main decisions for prototype project were:

1. Using Existing Tech (unity3d)

This decision was really good as we get things done more rapidly than we would have with the original JLWGL (Java Light Weight GL) which was in use in the beginning. We would not have been able to complete all our content and features intended for prototype without this decision.

2. Using Version Controlling SVN, later Git

We used free Unfuddle cloud service to provide Version Controlling for the team members. This was a mandatory decision and it was important that we investigated this issue in the very beginning. Later as I see it, it would be better to start using Git immediately at the start, instead of SVN.

3. Only Milestones, for maximum support of agile and flexible development

We had focused milestones, each containing one clear phase of development. We divided our milestones into M1: Basic Mechanics, M2: Advanced Mechanics, Adding and improving content, M3: Minor Content adding & polishing and M4: Gameplay testing with 50-100 testers.

In each Milestone we had the tasks necessary to complete before Milestone was ready. We were able to hit our Milestones well, but last Milestone “Gameplay testing with 50-100 testers” we decided to arrange only online, because we were running a bit late in schedules.

4. Free information sharing among team members

We used Google Docs to share information such as design documents, mock ups, concept art, art references and knowledge. Google Docs was small but important part in our development and also free.

5. C# language was used for scripts

We made a decision to use C# language in our scripts. One reason behind this was that we had earlier experience in C++ and C# languages but also the fact that it’s strongly typed language.

In the end this was a correct choice and our code remained being understandable.

6. Made the level loader on early phase

We saved enormous amount of time with this decision as combining levels only with Unity3d Editor is slow and bound to errors.

Making a level loader that used prefabs was good decision that saved a lot of time.

7. Project was organized in different scenes and assets, prefabs and scripts were organized properly

We organized our assets in proper hierarchies and folder

8. Meetings were kept (online and face-to-face)

Especially at concept phase and later when we got reinforcements (extra programmer), meetings were golden.

9. Extra programmer was taken into team in order to ensure completion the project.

We got our reinforcement programmer that was really critical in order to finish this prototype as I was offered suddenly programmer position from another company.

10. We were taking it easy enough and had fun during prototype development

Even a lot of surprises and the fact that we were lacking gfx resources, we managed to survive without hardly any conflicts and progress further as a team. We were not pushing each other in a negative sense too much, even still laying out a certain focus and milestones to achieve.

11. Arranging a Gameplay testing for n number of players

Since the start we had an idea for arranging a play testing for real players to find out more insight information from our prototype and on people's opinion towards it. This play testing was decided to take in to action when the prototype was ready.

The Gameplay testing was arranged with Unity3d web build thru web hotel service and the participant were informed on the testing request via email or online chat (mIRC, Skype, GoogleChat and MSN).

12. We decided to go 3D

In order to ease up transforming of game objects we decided to use 3D objects with perspective main camera. This decision was also related to the fact that Unity3D being not very famous on its built-in 2D features.

APPENDIX IV (“Monx in Mandala” postmortem)

Our PROS:

1. Our base concept was a fertile ground

From the beginning we think that our game concept was a concept of many possibilities. The concept provided fertile ground for aesthetics as well as funny core mechanics. I think that selecting a bit wacky concept for this prototype project made us all think in a fresh way and to process many issues we would not have otherwise ever thought.

2. Version controlling from the beginning

Taking our Unity3d project into version controlling was a really important decision and it was good that we did this since the very beginning of the project. This enable all developers have newest working build available and enable also simultaneous developing.

3. Got extra programmer in early phase

Hiring another programmer was of great significance to finish the prototype on time and hitting our Milestones better. Our new programmer was proven to be efficient and his work quality was also good. He also committed to development as much as was necessary.

4. Got better assets in the end

After losing our 3D graphics artist we were long time without any good models. However at the end we got another artist interested on making better models for us that we had been hoping for a longer time. These models were really important to get the prototype to a level of quality that we could let other people test it.

5. Using existing tech was a good choice

We saved enormous amount of time by using Unity3d game engine. This decision was really important in order to get prototype in a feasible timespan. Unity3d turned out to be very good tool for our project and porting builds to different platforms was also easy. I raise my hat for Unity Technologies.

Our CONS:

1. Not having GFX artist from the beginning

Prototype was not meant to be super graphical from the beginning, but soon we found out that a proper graphics person inside the team would have been great help.

We had major difficulties on getting any graphics person committed to prototype and took long time before we even had proper models in. However we had basic test models that we could use, so the issue did not become really a bottleneck.

2. Not developing immediately for the actual device

In the beginning we were not set up our mind which input controller the game would be using in the end. We wasted really much time on making gamepad controls that were later necessary to change into mouse controls.

We noticed the issue far too late and the side-effects that it caused. Some parts of our game code and mechanics needed to be rewritten. However, the controls turn out to be rather good in the end.

3. Bad Design Decisions

To be honest, we were lacking focus on the details of our core mechanics and how we wanted everything to be in the end. I would like to think this as an iterative agile progress, but I must admit that we were not framing our technical details enough sharply for the game.

Partly I think reason behind having sketchy designs was that we were using couple of hats too much in the project and each developer impacted on the game design. We should have had a clearly one Game Designer who would make decisions in a focused manner.

Also, we should not have taken the 3D graphics path from the beginning. Even the lack of resources in the 3D graphics side did not create a bottleneck it still makes me wonder what kind of interesting things we could have implemented if we had decided to use 2D graphics on our game instead.

In other words, we would have much more content & feature rich demo. Using 3D was probably the most stupid thing to do as I now later looking back on our development process.

However, we could not guess that our first 3D graphics person did not have time to contribute to the prototype for longer time.

4. Surprise changes in the plans

As the surprise was positive, it also caused a lot of hassle to our development. At the half way of our prototype development I was offered a game programmer position. Naturally I needed to react quickly on the surprise as I would not have any more possibility to work 100% with the prototype. We got another programmer into our team.

After I started working in my new position I was not able anymore to push huge efforts for the prototype and suddenly I became more or less a producer than a

programmer for it. It was really important that I reacted fast to the sudden changes and could arrange the project to continue and reach completion.

5. We should have used more Scrum methods

In the beginning of the project we decided to frame only Milestones to achieve. However we did not hit the last two Milestones on time and one of the reasons behind this was that we having too loose project management. I think iterative development were every week would have own tasks to commit could have helped us.

Due to the working style of ours a full-scrum would not be correct option, instead methods which concentrate on time management and scheduling would have been helpful. Possible methods to use could have been the sprint planning and the Scrum Wall, without sprint retrospectives or everyday Scrum meetings. However one agile method we really did well and that was code refactoring. We kept our code base tidy for most of the time and often enough did refactoring work on it.

(*) As a special note I was really happy about the feedback got from the arranged Play Testing Session and got many good ideas on what people liked on the prototype and what was not working so well.

APPENDIX V Glossary

1. Indie = Independent Game Developer
2. Indie Game = Independent Game made by an Indie Game Developer
3. Prototype = simplistic and small early version of the game that shows the core mechanics and idea. Prototype is meant to explain the idea and mood in a practical example and can be used to early stage play testing
4. Postmortem = The report containing 5 pros and 5 cons of the game project after the game has been finished
5. Passion = interest on the subject that breaks all boundaries. A person who has passion towards his work is doing a task of great importance for himself which gives his life a new meaning"
6. Crunch = longer period of over working at the end of the project cycle before launch.
7. Demoscene = a community of people who are making demos, audiovisual experiences that include art, music and programming
8. Open-Source = a software that offers its source codes for public use
9. Symbiosis = a situation where different factions gain benefit from each other
10. Fan base = a community of fans following your company and the games you're publishing
11. Word-of-Mouth marketing = viral marketing for example in social medias such as Facebook, Twitter, Blogs and Digg

12. AAA = big budget studio games titles, such as Alan Wake or Assassin's Creed. AAA titles are typically funded by investors.
13. Synergy = harmonic Co-operating in the team
14. Too Many Hats = person has too many responsibilities to take care of in the project
15. Quick wins = fixes or features that give a rather important economic benefit compared to their required effort from the team.
16. Asset = a file containing game graphic, a sound, music etc.
17. Barrel old rock band = in this context means a team that can work fluently together
18. Moral Law = C.S. Lewis' definition on what people know by nature as Good/Evil
19. SoMe = Social Media