Rapport

Working with Serious Games

Guidelines for conducting a successful serious game project.

Teknisk rapport

Editor: Björn Berg Marklund, Marcus Hellkvist, Per Backlund, Henrik Engström, Marcus Toftedahl

2017-09-19

Acknowledgements

This handbook is written as a part of the EU Interreg Öresund-Kattegat-Skagerrak funded project *GameHub Scandinavia*. The aim of the project is to provide resources and services to developers, educators, researchers, and supporting actors that are involved in the Scandinavian game industry. This text in particular is intended to help serious game developers and clients communicate better during their project by fostering a mutual understanding of the challenges and benefits of using serious games, and hopefully to encourage people to examine if serious games might provide useful benefits to their working processes or their local communities.

Authors
Björn Berg Marklund, PhD
Marcus Hellkvist
Per Backlund, PhD
Henrik Engström, PhD
Marcus Toftedahl

Dr. Björn Berg Marklund has been working with serious games, in particular educational serious games, since 2010. In his thesis, *Unpacking Digital Game-Based Learning*, he studied how both developers, teachers, and students approach and experience the development and use of serious games, and focused on establishing a pragmatic overview of what it actually takes to create something as technologically advanced as a digital game, make it engaging while still representing educational subject matters accurately, and also implementing it and making it work in a classroom environment. In his work, he emphasizes the importance of employing a broad systems-oriented perspective when thinking about serious games, in order to ensure that a created game isn't just engaging and hypothetically educational, but also efficient, reliable, and realistically useable so that it actually can make a positive impact in its intended context of use and reach its target audience.

Marcus Hellkvist has studied game development since 2013. He holds a BSc in arts with a major in Media Arts, Aesthetics and Narration, as well as an MSc in Serious Games Development. In his master's thesis project, he co-developed the serious game *Braille Hero*, and during the project he studied smartphone game design and UX with special consideration taken towards visually impaired players. The project aimed to both refine smartphone interfaces to provide better accessibility for visually impaired individuals to ultimately create a game-based, engaging way of teaching braille.

Dr. Per Backlund is an associate professor of informatics at University of Skövde and the head of the university's Interaction Lab. He has been working in several serious games projects since 2005. These projects have been in a wide variety of areas, such as traffic education, healthcare, vocational training, and general education. Per is also one of the co-developers of the Serious Games MSc programme at the University of Skövde and he has been a lecturer at the programme since the start in 2008. Per has supervised several serious games related theses projects at both MSc and PhD levels.

Dr. Henrik Engström has a PhD in Computer Science from the University of Exeter and he has conducted game-related research since 2001. His research interest is focused on the game development process with a particular focus on its entangled, multidisciplinary nature. As part of the research, Henrik has had the role as project leader, producer and developer in a number of game projects.

Marcus Toftedahl is a game researcher, developer and lecturer at the University of Skövde, Sweden. His main competences lies within game production, game design and narrative design. His main research interests include serious games, gamification and the processes found within indie game development and production. Marcus has worked at the University of Skövde since 2009 and has since he started co-developed the world's first full concentration game writing education at University level as well as teaching game design and game production in general. Marcus have led multiple game development projects as a part of the University's digital game research, focusing mostly on using games in public spaces. Since summer of 2016 Marcus is working on his PhD project in socio-technical systems researching indie game production with a special focus on localization.

EXECUTIVE SUMMARY

This report has been written to explore different ways in which games can contribute to various developments in society, businesses, health, or in other areas. This report explore the origins of serious games and gamification and their application areas. Furthermore, a detailed explanation is given of what serious games are and how they can be used by illustrating the practicalities, benefits, challenges, and problems connected to serious games development. The most essential steps such as pre-production, production, and post production is brought up. The various stages are explained in detail. What to do in the beginning of a project, what makes a serious game effective and useable. How to practically develop serious games, and how to use and evaluate them once they are implemented into their right context. This report provides examples from serious games and gamification projects to highlight various challenges and problems which can arise from dealing with game development and serious games.

Table of contents

1 Introduction

If you are reading this report, it means that you at the very least have a passing interest in exploring and understanding the various ways in which games can contribute to various developments in society, businesses, health, or in other areas. Maybe you are approaching this from a developer's point of view, and want to examine if serious games and gamification can be a part to your studio's business plan. Or, maybe you are approaching it from the standpoint of an educator, employer, marketer, or organizer who wants to explore whether games can help improving your working processes or expand the audience you reach with your messages and products. The purpose of this report is to prepare you, regardless of which perspective you are approaching this subject from, to be able to work with serious games and gamification efficiently and confidently. We will briefly describe the origins of serious games and gamification and their application areas, and we will subsequently go through and describe every phase of planning and development that is involved in working with serious games and gamification.

Throughout the report, we will use text boxes such as this one to describe examples from serious games and gamification projects that we have conducted ourselves to highlight specific challenges or solutions that we have encountered and that might be useful for you as well. The textboxes will also be used to summarize what we consider to be particularly important takeaways and considerations from the different chapters.

1.1 Important terms

There are some general concepts that we will be using throughout this report that might need some clarifications. The field of serious games and gamification can be riddled with ambiguous and misappropriated terminology, so it is important that we account for what *we* mean when we are using some key terms in this report.

- **Serious Game, Gamification or Game-based Solutions**: one of the most contested terms in the field of games with serious purposes, is what they should actually be called. Many prefer the term 'serious game' as it is one of the older terms, whereas many are starting to use 'gamification' more and more to often refer to the same type of games. In this report, for the sake of simplicity, we primarily be using serious games as the default term to refer to these types of games. We also provide a more in-depth disambiguation of the terminologies in chapter 2.
- **Developer**: in this report, we are using the term 'developer' much in the same way it is used in the games industry in general. The developer is the one that creates the product - the one that makes the game function and run properly. This involves anything from creating all the assets and models that are needed for the game to look nice, designing and balancing the game mechanics so that it plays well and "feels right", and doing all the programming to make the game actually function.
- **Client** or **Customer**: it is always difficult to have a nice, easy-to-use term that captures everyone that is not a developer in a serious game project. A teacher that wants to start using a serious game in her classroom might not feel like a 'client' or 'customer' since she plays such an active role in the use of the game. The same is often true for most projects: serious games projects are often co-development projects where teachers, physicians, legislators, and

journalists partake in the development process. With this in mind, 'client' might sound a bit commercial or maybe makes it sound like non-developers are passive buyers when it comes to serious games. Even so, we have not really found a better term to describe the persons who commission or use serious games. So in this report, the word 'client' is used to mean the persons in the serious game project that is receiving, purchasing, commissioning a serious game from a development studio, or co-developing a serious game with a development team.

- **End user**: the person who makes or buys a serious game is not necessarily always the same person who will be using it in the end. For example, a hospital administration might order or purchase a serious game that will later be used by its patients or physicians. In this case, the patients or physicians is called the 'end users' (while the hospital administration is the 'client').
- **Context of use** or **use context**: simply put, the context of use is the setting/environment in which the serious game is ultimately going to be used, for example a classroom, a corporate retreat, a patient's home, a museum, or on the streets of a city. Every context brings its own unique challenges and opportunities, which is why we will spend a lot of time discussing it in this report.

This report is written as a part of the EU Interreg Öresund-Kattegat-Skagerrak funded project Game Hub Scandinavia. The aim of the project is to provide resources and services to developers, educators, researchers, and supporting actors that are involved in the Scandinavian game industry. This text in particular is intended to help serious game developers and clients communicate better during their project by fostering a mutual understanding of the challenges and benefits of using serious games, and hopefully to encourage people to examine if serious games might provide useful benefits to their working processes or their local communities.

2 Serious Games and Gamification

It is always difficult to know where and when to start when it comes to thinking about the creation and use of game-based solutions. If you are a developer who is just starting out on the serious game and gamification market, it can be tough to know whether to focus your energy thinking about game designs, subject matters, potential application areas, or how you are supposed to manage your marketing and how to even find clients. If you are a person or organization that might be flirting with the idea of using or creating a game yourself, you are probably facing a similar dilemma: do I start looking at the commercially available games and see if one suits my needs, do I try to find a developer to collaborate with to create something custom made, and what type of work and what type of resource investments might I be looking at? In short, it is difficult to know where to start when it comes to the topic of serious games and gamification. In this initial chapter we aim to clear up some common misconceptions surrounding them, and hopefully equip you to be able to more clearly state what you want and what you are looking for when you are talking about game-based solutions.

2.1 Why are serious games and gamification popular?

Games are often celebrated for their unique capacity to represent and simulate complex systems and invite players to experience and interact with the first-hand in engaging and interesting ways. They allow the player to form an understanding of intricate subject matters based on participation and experimentation, rather than mere observation. It is for these reasons that games are often argued to have great potential as tools for information, education, persuasion, and positive behavioral change, and many other beneficial endeavors. In a game, the player can take on the mantle of a medieval ruler,

a soldier in the midst of historic conflict, a business tycoon, or essentially any other human or non-human individual that can be imagined. If the game is well-crafted, the player can spend hours upon hours engrossed in it, trying to master whatever challenges that game contains. Many games are naturally designed to become progressively more challenging to keep the player interested too. They introduce new concepts, items, maneuvers, or characters that the player need to experiment with and figure out in order to be able to confidently put them to use and to continue progressing in the game's narrative. Because of this, games seem to inherently possess many of the qualities of a good learning environment, such as positive reinforcement, engaging and escalating challenges, and immediate feedback.

Digital games and game application are also used by an increasingly diverse demographic throughout different levels of societies and cultures. As technologies that can support game applications become increasingly ubiquitous and accessible, the ways and places in which people engage with games are rapidly diversifying; people play on the bus, at home, in classrooms, and at work, and gaming is thus starting to reach domains, public spaces, and social settings that they were previously excluded from. Furthermore, digital games also constitute an industry of growing cultural and economic significance. Most often, digital games are thought of as entertainment products, which of course is true. A large part of the games industry is devoted to producing entertainment software, and it is primarily the marketing campaigns of big blockbuster game titles that become visible to the general public. This growth of the entertainment games industry might be one of the reason as to why serious games and gamification are starting to gain more traction.

With the rapidly growing interest in game-based and gamified solutions, a lot of developers who may not have experience creating games for specific 'serious' purposes, as well as companies and organizations who might not have previous experience working with game development, want to start working with serious games. While this is, of course, a very positive development - it can lead to some issues. Inexperienced people, and perhaps in particular people with knowledge in related field that they assume are directly transferable to another field, are more prone to make mistakes; this is precisely why this report exists, to expand your understanding of both the complexities and the potentials of serious game development. We will lead you through different types of development project, and present common challenges and pitfalls so that you are better equipped to predict and deal with them as soon as they emerge.

The rise of the now-ubiquitous term "gamification" is a good indicator that we still are not really in agreement of how to treat games as tools for things beyond escapism. New terms that try to encapsulate what games can do for education, healthcare, and society at large emerge every now and then and each one carries new values, guidelines, and agendas with it. Throughout the edutainment, game-based learning, serious games, and gamification paradigms there has always been a myriad of different opinions regarding how to do things right.

2.2 What is serious games and gamification?

There has been an ongoing debate in the research community regarding the merits of both terms and the relationship between them. The difference between the two concepts have become increasingly ambiguous in recent years, and because of this, they are often treated synonymously. However, in their original conceptualization, serious games and gamification described two quite different practices. They can be hard to distinguish from one another, but in essence, serious games refers to

the practice of translating subject matters, activities, or events into full game experiences so that players may experience, practice, or experiment with that through the medium of games. Gamification on the other hand, is the practice of attaching elements that make up game experiences onto existing activities to make them more engaging and motivational, or to encourage people to perform and reflect upon them in new ways.

This difference can be illustrated by giving the same problem two different solutions. E.g. if we wanted to encourage people to be more mindful of their use of electricity in their homes. A serious game approach to achieve this goal could be to create a game where the player gets to walk around in a house, operating different apparatuses in a simulation of day-to-day activity, and then see how it effects their electricity meters, budgets, and the environment at large, thus allowing them to see and experience electricity consumption. The "gamification" approach on the other hand could be to simply attach a scoring system to light switches and appliances in a home, and reward the inhabitants with points depending on the duration of time that the light are turned off during daylight hours. In the former example, the real-life activity of electricity consumption management is made into a game, and in the latter, the activity is still the same but gets framed with game-like elements intended to encourage and motivate the type of behavior and awareness we aim to promote. That, in essence, is the fundamental difference between serious games and gamification:

> *In serious games, activities are translated into a game experience. In gamification, the activity itself is unchanged but an extrinsic layer of game-design is applied "on top" of it.*

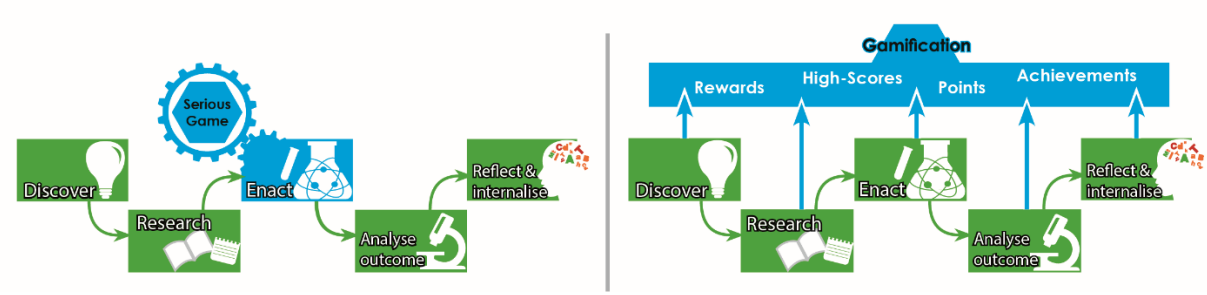

Figure 1. Whereas developing a serious game is a process of transforming an activity to make it into a game experience in which participants can experiment with a subject matter in new ways, gamification is the process of creating a gamified layer that doesn't change the activity itself, but aims to change the end users' engagement with the activity in some way.

2.2.1 Serious games

As previously mentioned, many attempts have been made to encapsulate and define the term serious games, which have led to countless hours of debate on how it should be defined and used. But the most common term, which most of them nowadays tend to gravitate towards is that games which engage the user, and contribute to the achievement of a defined purpose other than pure entertainment, should be classified as serious games. Interesting enough, a game's purpose may be formulated by the user or by the game's designer, which means that a commercial off-the-shelf game, used for non-entertainment purposes, may also be considered a serious game. It is worth noticing that the utilization of game and visualization technologies, simulations and virtual worlds for purposes beyond entertainment can be included under this definition as well. That being said, these games should by no means be boring. The defining moment of a successful serious game is not the

achievement of any given purpose per se, but rather, the effectiveness and ability to captivate its user to increase engagement and heighten motivation within a subject. When serious games are discussed, digital games and solutions comes to mind, nonetheless, analogue board and card games could also be included into the realm of serious games as well. This report however, will primarily focus on the former. As concluded from the previous statement, interactivity and gameplay is what separates serious games from gamification. Applications of serious games can come in many different forms, such as education, training, rehabilitation, information, policy making, marketing, and recruitment. This can be used in (but not limited to) healthcare, social change, advertisement, organizational training, and educational environments followed by their respective sub-categories.

To develop or use a serious game in one of these applications areas, different approach strategies can be used or applied in order to achieve positive results. One could use existing or adapt commercial off-the-shelf (COTS) games to fulfill an educational purpose in a classroom environment. The serious game experience can be hired as a service where specialized companies proved the game and the associated infrastructure. One could implement pre-existing serious games if it suits the specific pedagogical goal.[1] Or, one could (which is the focus of this report) develop their own serious game which provides control and stability for the project but requires multidisciplinary skills, time, and resources.

2.2.2 Gamification

One example of a game element that is commonly used to increase users' motivation and engagement with an application or activity, is to tie it to a scoring/reward system. This has the benefit of clearly visualizing goals, that might otherwise be difficult to codify or "see", for the user, which in turn can increase the user's motivation to want to achieve better results, and foster a higher retention with an activity (such as practicing an instrument, or maintaining healthy habits). Designing these types of reward systems, however, is difficult and takes a certain type of deliberation and expertise during the development of an application. It is a delicate matter to design a scoring system that keeps the player informed and motivated throughout the entire game, and they have to be carefully designed to promote desired behavior and create a meaningful experience that keeps the user on the task.

While a scoring system is most often the element that is first thought of when discussing gamification, it should be noted that scores can be a double-edged sword as many people do not become motivated by scoring systems alone, and may in fact be deterred from an activity due to the presence of constant quantitative assessments of their performance. Other examples of so called game design elements (as in concepts being imported from or inspired by what is often present in games) that are being used are missions to keep the player motivated to pursue an activity and badges or medals which the user can earn and show off as tokens of success.

2.3 Application areas

Serious games and gamification can be used in a great variety of ways, for a great variety of purposes. Games have been used in education, healthcare, social change and journalism, advertising, and organizational training for quite some time already, so there are definitely some precedent when it comes to creating games for specific purposes. That being said, serious games and gamification are still in a very exploratory phase, and as new game technologies are continuously developed, new

[1] A. Calvo et al. 'Tools and approaches for simplifying serious games development in educational settings', *IEEE Global Engineering Education* Conference, vol. 10, 2016, pp. 1188-1197.

potential areas of applications and ways to solve problems with the help of games are constantly cropping up.

One major change in the last 25 years is the increased focus on digital solutions that has come in the wake of the success of digital entertainment games. This has led to more attention as well as new opportunities. However, developing a digital solution also entails new challenges; both technological and organizational, i.e. when and how they should be used. It is typically hard to distinguish these challenges from each other. More often they are intertwined and need to be addressed simultaneously. We argue that it is equally important to consider the organizational context in which the solution is meant to be used as it is to develop the actual technological solution. This is a well-established fact in areas of traditional IT use in organizations, e.g. digital information systems for keeping track of sales and purchases. What we try to say in this report is that when you develop something like a serious game or gamify a work task, you actually combine the two complex tasks of developing a games as well as a utility solution. It makes sense to think of this as a complex process in its own right.

Whether you are a game developer, in the midst of creating a serious game, or a client looking for a digital solution, three crucial aspects should be taken into considerations before creating or choosing a serious game:

- Context – In which field or application area should the game be used. (E.g. education, healthcare, military, etc.) The context of use heavily dictates the game's design.
- Purpose – The serious game need to have a purpose. I.e. increase math and algebra skills, or heighten the intrinsic motivation for the American Revolution. It is difficult to implement any relevant information into the game without a solid purpose.
- Audience - Who should play the game? Is the game for the general public, professionals, or students? What is the age group? All these aspects affects how the game looks, how it should be played, and what type of platform should be used etc.

It would be weird (and ineffective) to use a serious game aimed to educate high school students about the civil war, and use it in a military environment. Even though the purpose is the same, the market and the audience plays a huge role in how the game is designed and issued. Therefore, the digital experience need to be tailor made for a specific purpose, application area, and audience, which creates for unique challenges and benefits.

2.3.1 Education

One of the more popular and well-known applications of games in "serious" settings is the use of games for educational purposes, either as a product to be used by families at home or as a teaching tool in classroom environments. Using computer and video games for educational purposes is not a particularly novel concept. You are likely aware of some of the "edutainment" titles that were available in the '80s and '90s yourself – some of the more wide recognizable ones are *Where in the World is Carmen San Diego* and *Math Blaster* (or maybe *backpacker, Chefrens pyramid,* and *Krakel Spektakel* series are more familiar if you went to school in Scandinavia in the late '90s). But the educational genre of games have actually been around for about as long as entertainment games, and they have both evolved alongside each other since the early '70s.

The game *The Oregon Trail*, for example, is a seminal title in the educational game genre. It was first created in 1971 specifically for use in classrooms at a school district in Minnesota. Not only is Oregon Trail one of the first educational game titles released on a digital platform, it still regularly gets updated and re-made to this day. The significance of Oregon Trail is that it was devised around the same time as video games started becoming available as consumer products for home entertainment. There had certainly been a few primitive, primarily mechanical, games for arcades and other establishments a few years prior, but in the early 70s video games became a more open consumer market thanks to the advent of home consoles with the Magnavox Odyssey being released in '72 and the popularization of home computers with the release of the Apple II in '77.

In these early days, the ambitions regarding how educational games (back then often referred to as 'edutainment') would change the educational landscape were very high. As computer simulations rapidly became more and more advanced, the ideas of what games could achieve continued to grow. The problem with inflated ambitions, however, is that it's hard to actually fulfill them. Educational games failed to keep up with the rapid increase in quality of entertainment games that received much larger development budgets, and controversies around inappropriate content in certain games made many parents and teachers wary of putting them in the hands of children. These factors, among many others, led to a rapid decline in the interest in educational games and in the very end of the 90s, the edutainment game market went from being a billion dollar industry to essentially evaporate completely.

Only a few years after the late '90s collapse, however, interest once again started to pick up. More and more research pointing to the benefits of using games for healthcare, training, rehabilitation, and education started to emerge in the early '00s, and games also became more easily available to a wider audience through social media and mobile gaming. Interest in educational games is currently very high, and new examples of games being put to interesting uses in schools pop up frequently from places you would not have expected a few years ago.

Given these qualities and the wide variety of game genres out there, it seems as though games could find a natural place in classrooms to teach a wide variety of subjects in a hands-on and participatory manner. Games seem to correspond nicely to most buzz-words frequently thrown about in the debates surrounding education – the "new" era of education should be engaging and motivating for students, it should be participatory and active instead of passive, and it should invite students to interact with new technologies to give them the "21st century skills" that nowadays seem essential to surviving in contemporary society. While all of this rings true to some extent, it is dangerous to assume that merely throwing a game into a classroom will create a positive learning environment that embody all those desirable values. Games are complex technologies, and while they do provide many exciting new opportunities they also have their own limitations and unique requirements that both developers and educators need to be aware of if they want to use them effectively.

Games *can* certainly be put to good use in classroom environments. But it is important to keep in mind that a game is a tool; a tool with some unique and endearing properties certainly, but a tool nonetheless. As with any other tool it needs to be used *correctly* in order to function efficiently (or to function at all).

Games in education can:

- Provide interesting alternatives to subject matter exploration
- Change and challenge classroom dynamics
- Provide a venue for first-hand experimentation of abstract or impossible-to-stage concepts, objects, and events
- Immerse students in subject matters, rather than passively showing them
- Revitalize students or classrooms (wow-effect)

Common challenges:

- School infrastructures, hardware availability and performance
- Budgetary constraints
- Subject matter expertise is important
- Teacher and/or school collaborations can be challenging
- Maintenance / IT departments
- Rules and regulations
- Parents/ teachers attitudes towards games
- Teacher's technology literacies
- Children's tech literacies and individuality

2.3.2 Healthcare

Digital games for health has caught much attention and is a growing application area. There are examples of how games have been used in treatments and rehabilitation as well as for spreading knowledge and information. There are summaries of examples where games have been used with positive effects in different areas of medicine.[2] Some interesting applications include using games to distract children during painful treatments, such as chemotherapy for cancer. The thing to point out here is that the degree of attention needed to play some videogames is high enough to distract from pain, which tells us something about the power of engagement in playing. One strength of this application of games is that any game which is enough engaging for the player can be used.

Another way of applying games in a healthcare context is to develop games with a specific topic or purpose in mind. Examples of this approach is games targeting children obesity,[3] which can roughly be categorized into two categories: exergames and educational/persuasive games. The former are games that include physical activity as a part of the activity, for example dance games; the latter are games that aim to inform about obesity and the effect that lifestyle may have on obesity in order to induce lifestyle changes. The goal of an exergame would hence be to make children physically active with the help of the game. The general idea is to create a game that requires physical activity which is in some sense similar to any physical play activity or sport in that the movement is inherently present in the game. The educational approach, on the other hand, aims to inform the player about, for example, the amount of calories in different food products in order to make them aware of what to eat and reduce daily intake. Irrespectively of which approach the key issue for any serious game or gamified approach is to have an impact on knowledge, motivation, behavior, attitude or some other key factor relevant to the field in question. This is a key challenge in the development of gamified solutions which entails

[2] M. Griffiths, 'Video games and health', *British Medical Journal*, vol. 331, no. 7509, 2005, pp. 122–123.
[3] S. Guy, A. Ratzki-Leewing, F. Gwadry-Sridhar, 'Moving Beyond the Stigma: Systematic Review of Video Games and Their Potential to Combat Obesity', *International Journal of Hypertension,* vol. 2011, 2011, pp. 1–13.

an understanding of the application domain in question. That is, what is the problem to be solved and how will the game solve it, or contribute to the overall solution. In the following we will illustrate this with an example from one of our projects in the healthcare domain, namely stroke rehabilitation.

In one of the projects undertaken by the authors of this report, the Elinor project, we addressed stroke rehabilitation, more specifically what we termed game-based stroke rehabilitation in the home. The vision of the project was to provide an easy-to-use game console with games that were fun and motivating while at the same time being beneficial for coarse motor training of the arms of the patients.

Stroke is a condition when poor blood flow to the brain results in a brain injury. A large portion of those who suffer from a stroke face a long rehabilitation process as recovery of motor abilities requires repetitive movement therapy over time. In fact rehabilitation training can have an effect a very long time after the injury and hence motivation becomes an issue. In stroke rehabilitation there is an interest in so called virtual rehabilitation. Virtual rehabilitation refers to the use of information technology for rehabilitation and is very often carried out in clinics with the help of advanced equipment, for example, haptic devices for motor training. The focus is on medical aspects rather than motivation, which of course makes sense in the context of therapy. However, we claim that there is a huge potential in addressing the question of game design more carefully. How can game aspects be better addressed in the development process so that the full potential of the serious game is reached?

Throughout the project we dealt with challenges having to do with hardware and software development, game design, stroke rehabilitation, health care regulations and work procedures. One of the first major challenges in the development process was to understand the actual problem and the problem domain. Our vision was to utilize game enjoyment as motivation for motor rehabilitation in patients' home. However, this vision comes with some challenges: firstly, there are a lot of specific challenges associated with health care requirements. As an example, we need to deal with a different set of ethical considerations when working with a target group that may suffer from both cognitive and physical impairments. Issues such as what is the effect of certain types of stimuli? We can't expose patients to uncomfortable or harmful experiences. The solution must be acceptable to health care professionals as well as patients. These are all issues that arise early in the design process, long before we are in a position to make any claims concerning the effect and effectiveness of the solution. Secondly, there are a number of challenges associated with designing games for people with motor and/or cognitive impairments. Examples include speed, colors and interactions. Typically, we expect a good game to be engaging and put the player in a state of flow where s/he does things without even thinking about them. However, in the context of stroke rehabilitation this can be a double-edged sword since patients in this group may be prone to repetitive behavior which in the long term may lead to stress. In our case, we chose to deal with it by imposing breaks in the games. This is bad from the viewpoint of engagement and flow but necessary in order to avoid stress related injuries. Thirdly, the target group for our stroke rehabilitation games are typically not gamers, some of them are not even acquainted to computers. So many things that are taken for granted in the gamer community may be completely unknown to the target group. E.g. that the spacebar on a keyboard is often associated with the action of jumping in many games. Finally, we also face a number of challenges that come with the fact that the system is self-administered, by the patients in their homes. This forces us to build something that is really easy to use and accessible. We aimed to capture this in the design vision "instant fun" which means that the player should be engaged in a play session within one minute from

starting the console without having to make a lot of adjustments and navigate through endless menus. Home environment and self-administration also entails high requirements on dependability.

During the Elinor project we identified three approaches to what is referred to as virtual rehabilitation: high-end and mid-range systems, which are typically designed with stroke rehabilitation in mind and a third approach which is about using existing (entertainment) games as they are. Each approach has its pros and cons, so the more detailed needs and resources will have a strong impact on what can be done. In one sense this means that all types of systems are useful and motivated in different parts of the stroke rehabilitation process, all depending on the status of the patient. High-end systems are specialized and expensive and have highly specialized interactions and exact tracking of movements which means that exact logging and analysis of movements is possible. This is advantageous from a rehabilitation perspective since it makes it possible to make careful adjustments with respect to the rehabilitation goal as well as to track the progress of the patient in a very exact manner. For example, it is possible to see if the fine motor abilities in the hand are actually improving. Another example of a high-end system is robot assisted stroke therapy. In this type of therapy it is common to use a robot arm as a sort of exoskeleton on the injured arm to support the patient in making the required movements. This type of system is hence composed of specialized hard- and software for the actual robot arm. This is a device that takes a lot of space and is not intended for home use. Another drawback is that the game aspect, which is to create the fun and compelling activity is put in the second room. At the other end of the spectrum we have commercial entertainment games with physical interaction played on regular game consoles (e.g. Nintendo, Xbox, or PlayStation consoles) which are used in the rehabilitation process. The upside is that the price is relatively low and that there is an abundance of games available. Drawbacks are lack of precision and tools for the purpose and that most games are probably too hard for the target group, even at their easiest levels. The general purpose of this approach is to activate the patient but there is generally little or no connection to the therapeutic process. As an option between these two approaches a mid-range system (such as the Elinor console or adaptations of commercial platforms such as e.g. Microsoft Kinect) may offer more freedom to focus on games and playful content while still paying some attention to providing limited support for logging and analysis. The general idea of this approach is to provide some of the therapeutic benefits from high-end systems but still have the relatively low cost and ease-of-use that will allow training in the home of the patient. An additional benefit of a device that is more adapted to games is of course that it will lend a stronger support for the motivational aspect of the rehabilitation process.

Whichever of the above approaches that is used, evaluation is of greatest importance. The evaluation of serious games or gamified solutions is special in that it involves aspects from the gaming sphere (is it an engaging game?) as well as from the domain specific utility sphere (does it solve the problem?). In the context of the Elinor project we evaluated three aspects: gaming behavior, rehabilitation effect and user acceptance. Designing and carrying out such an evaluation is a challenge in itself as it involves different areas with different traditions. Yet, they have to be integrated in some way in order to get the full picture of the situation. In our case, this meant to use methods and evaluation aspects from the different areas. Since we have our starting point in game development we included gaming behavior as an important thing to evaluate as we were interested in learning more about how our games were perceived by the user group. This is of particular interest since we were targeting an atypical audience, which in itself is a design challenge. We delivered Elinor consoles to a total of five patients who had them in their homes for a total of five weeks. We looked at two things: play time and flow. Playtime can work as one measure of engagement even though it does not capture information

of what players think of a game. Since our subjects had no instructions concerning when and how much to play we considered playtime to be a relevant measure. In general, players seemed to enjoy the games and the actual playtime exceeded our expectations by far. We had a vision of participants playing for fifteen minutes per day but the actual figure was five times as high for some participants. We also asked the players what they thought about their experiences and it was interesting to note that they talked about how they became so engaged in some of the games that they forgot about time and place. The rehabilitation effect was evaluated with the help of healthcare specialists using methods and measurements from their domain. We actually found some improvements in patients' mobility and quality of life. Finally, the acceptance of a new system is crucial with respect to its use. If users do not understand or for any other reason do not want to use the system it becomes completely pointless. In the case of a rehabilitation game such as Elinor there are at least two group of stakeholders who need to accept and use the system, patients and healthcare professionals. In the Elinor case it seems that both healthcare professionals and patients believed in the idea and that the concept would be useful.

Games in healthcare can:

- Lead to higher retention by making otherwise boring rehabilitation and/ or diagnostic tasks more engaging
- Be a good way to administer at-home patient care

Common Challenges:

- Games for healthcare can be difficult to design
- Compared to other application areas, precision and validation is a higher requirement in healthcare. Due to this, development costs can be high.
- Accommodating for a heterogeneous audience which might have disabilities that make gaming difficult.

2.3.3 Social change and the spreading of information

Games focused on social change and the spreading of information attempt to raise awareness of important social issues. The focus of these games is to transform the players' views or attitudes to inspire social change and to affect real-world behaviors. With this, many different areas can be addressed such as global issues, politics, poverty, social environmental risks, and human rights.

Depending on the level of change, goals, complexity, or target audience; games focused on social or behavioral change can be a bit more challenging compared to other application areas. In contrast to education and development, social or behavioral change typically refers to a much broader phenomenon connected on a deeper, more meaningful, and intricate level with peoples' opinions, attitudes, and everyday lives, often in conjunction with family or community members.[4] Instilling a healthier diet might be easier to put into effect than changing the society's position towards global warming. In order to create social change, multiple goals that ultimately affect a person's attitudes and opinions needs to be addressed, and for that to be successful, finding motivation and the will to do so are crucial and not something that will happen overnight. Because of this, randomly providing new

[4] C. Klimmt, 'Serious Games and Social Change: Why They (Should) Work', in U. Ritterfeld, M. Cody, P. Vorderer, in *Serious Games: Mechanisms and Effects,* Routldge, 2009.

knowledge and information about a certain subject is simply not enough when dealing with social change.

How many times have you heard someone utter the phrase "today I'm going to quit smoking" if a game were to help and change with this behavior it would require a lot of effort involving informing you about consequences of smoking, how it affects you, your quality of life, and everyone around you. The game need to do this, and simultaneously keeping the user motivated and engaged at all times, which is a challenge in itself. Adding the fact that changing a social behavior, deeply connected within a human's moral compass makes the process of change even more difficult. We are not saying that serious games and social change is an impossible combination, but depending on what level and type of change we are dealing with, it can aggravate and dictate the situation.

Social change and the adoption of a new behavior is a 5-stage process ranging from knowledge gathering, persuasion, decision, implementation, and confirmation of a behavior.[5] The process starts when an individual is exposed to the issue and gains understanding of it. Followed by persuasion where the individual forms an opinion or attitude towards the given subject. Decision, is the third step of the process and occurs when the individual engages in activities that lead to adaptation or rejection of the issue. Implementation is when the new behavior is put into use, and lastly. Confirmation where the person seeks reinforcement. With the use of serious games, the stages involving persuasion, decision, and implementation can all be designed and used as game elements to instill social or behavioral change within an individual.

Play and interactivity can increase users' emotional engagement with a subject matter and helps bring out affective responses that are crucial to achieve social or behavioral change. Games allow the player to embody (basically anything) in the game world in a playful way without consequences. This behavior fosters curiosity and experimentation and gives the player the opportunity to explore multiple identities and endless possibilities.[6] Since the rules and the consequences from the real world does not apply, sensitive and serious topics can be explored and help bring a new uncharted perspective to the player. The player is more likely to understand a certain subject or problem if he or she is at the center of attention and experiencing it first-hand. Such perspective is hard to achieve with traditional mass media where the recipient usually is a passive observer. A good example of this would be the game *Peacemaker*, where the player is forced to deal with the Israeli-Palestinian conflict, in which the player assumes the role of the leader of either party. The player then has to deal with events presented as real world pictures and footage by making social, political, and military decisions. The reason for the game's success is because of its ability to put the player in the leader's shoes and actively show the impossibility of the task at hand. Another game, with a similar approach is *Food Force*, where the player distributes food in famine affected countries and help them become self-sufficient. The game teaches the player about hunger in the real world and how to prevent it. As previously discussed, to have the ability to embody someone or something else with the use of active participation is a great way to persuade someone and force them to see the problem from a new and different perspective. Furthermore, in cases like these, it is of the utmost importance that accurate and objective information is used and implemented into the game. What is the point of having a game about social or behavioral change if the information is incorrect and wrongfully used? However, in some cases it might be

[5] E.M. Rogers, *Diffusion of Innovations*, 4th edition, New York: The Free Press, 1995.

[6] H. Wang and A, Singhal, 'Entertainment-Education Through Digital Games', in U. Ritterfeld, M. Cody, P. Vorderer, in *Serious Games: Mechanisms and Effects,* Routldge, 2009, pp.271-292.

problematic to show the whole picture due to its complexity or comprehensive material, which forces you as a developer to simplify the solution without affecting the content and intended outcome.

2.3.4 Advertising Games

Games' ability to communicate through interaction can be highly valuable in various forms of marketing. Advertising through games can happen on many different levels. Games can be built from the ground up with the explicit purpose of advertising a product or service. Advertising can also be made part of a game later in the development cycle as natural synergies between the game and other products and companies start to emerge.

There are plenty of past examples of both of these mergers of games and advertising available to draw from when discussing this topic, as 'advergames' has been around since the early '80s. Early examples of advergames were, in essence, fairly rudimentary games in which the main characters are company mascots (e.g. Kool-Aid Man from 83, McDonald's M.C Kids from 92, and the Coca Cola's Cool Spot from 94). These games often borrowed their core game elements and concepts from other popular titles, and used the popularity of these game genres to convey company branding and characters.

The sophistication with which games have been used for advertising has, however, increased greatly in the past few years, and advergames now go far beyond just superimposing company branding on top of already popular game types. Modern advertising games are often more deliberately built around showcasing a company's or product's philosophies, functionalities, and how they can provide value to the consumer. This means that the designers of such games often make very different design choices than 'entertainment' game designers do. Making the game accessible and quick to start and finish is of special importance - players rarely spend extended periods of time on smaller game experiences, so it is important that the advertising messages comes across quickly and clearly, and that the advertised item is associated with something exciting and pleasurable. These small, powerhouses of super-efficient gameplay experiences makes for an easy distribution across multiple platforms.

In most cases, advertisements are seen as something intrusive and forced. People rarely appreciate advertising. Therefore, in advergames, it is important to try and captivate the player with a good game first, before trying to sell or raise awareness of your product. No one wants to play a boring game, especially if it's full of advertisements. Red Bull had an interesting approach to this problem. In which they created a racing game (*Formula Face*, sharing similarities with *Mario Kart*), built on a unique webcam gameplay mechanic. The game was played with facial controllers via webcam tracking. In order to steer the vehicle the players tilted their head left or right. Special abilities, such as turbo boost and power ups could be activated by smiling, blinking or making grimaces. The game also implemented social features to reach out to more people, like in-game photo sharing and the ability to challenge others via Facebook. The (at the time) unique gameplay mechanics made for an interesting experience despite the fact that everything in the game was covered in advertisements and Red Bull cans.

This focus on entertainment separates advergames from other types of serious games. In education for instance the serious game experience is usually mandatory and a temporarily substitute for traditional learning methods. However, no one is really forcing you to play advergames. Therefore, the aspect of entertainment and fun in advergames needs to be treated with more care and consideration compared to serious games found in other application areas. Players left with a negative attitude

evoked by the exposure of an advergame could translate to the brand or product itself, resulting in avoidance.[7]

Advergames can be categorized into three different types; associative, illustrative and demonstrative.[8] In associative advergames, the brand, product, or logo is placed in the game in such a way to support brand awareness through lifestyle association and the idea is to connect the brand with the activity depicted in the game outside of the gameplay. In illustrative advergames the brand or product appears within the game as a prominent feature and plays a significant role in the gameplay. In the demonstrative type the product is presented in its natural context and the player can interact with the product in a "try before you buy" approach. The racing advergame *BMW M3 Challenge* uses this demonstrative approach where the players experience the car and has the opportunity to interact with its features.

The main difference between advertisement in games and advergames is in the development process and ownership of the game. The development for advergames is funded and controlled by the marketer, while in regular games the advertisement company has to rely on independent game developers and third parties.

Disambiguation: Advergames vs. In-game Advertising

With the continued evolution of mobile games, advertising has also become a way of generating revenue from a game title. This is a type of advertising that is done with very little collaboration between developers and the creators of the advertised products. It is essentially the same relationship a film's director might have with the coffee makers who advertise their products in commercial breaks when the film is televised: none at all. Since this is only a form of revenue generation for game developers of free-to-play games, it is not something we will cover at length in this report.

2.3.5 Organizational training

An area of serious games and gamification that deals with relatively complex goals, is the one of organizational training. There are many different ways in which games have, and can be used for organizational purposes. In this report however, we have decided to focus on the biggest application area for organizational training, mainly the military, which has a long history of using games for multiple purposes.

One of the earliest wargames is called *Chaturanga* and was played in India over four millennia ago. The game shared certain similarities with modern day chess, which might be one of the most well-known military training games. Leaders and officers were taught chess to improve on the battlefield and prepare them for battle. Around the 17th century, different chess-like games started to circulate which reflected military units and complex terrains in a better and more accurate way. From this, more and more sophisticated and complex games and simulations evolved. Aside from chess, many of these games never made it to the public as entertainment.

[7] J. Marti-Parreño, et.al, 'Factors contributing brand attitude in advergames: Entertainment and irritation', *Journal of Brand Management,* vol. 20, no. 5, 2012, pp. 374-388.
[8] J. Chen and M. Ringel, 'Can advergaming be the future of interactive advertising', 2001.

Games gave the possibility to experiment with different strategies and analyze the outcome without risking troops and resources. Wargames was also a great way to discover and exploit weaknesses. "If he do this, then I can take advantages of this like that." Which gave a clear overview of the battlefield and let the commander experience things from a different perspective and subsequently assess the situation differently. Given the possibilities and value of games in battle they grew and became more complex. *Kriegspiel* was one of these complex games used to train officers in the Prussian and German armies. The game was played on a specially designed table with the playfield divided into a grid system. The game consisted of different combination of modular terrain types, a system to simulate "fog of war", special gaming pieces and different dice. The game used topographical maps drawn at a 1:8000 scale. Blocks were used to represent different military units and were assigned various movement speed. To make the experience even more realistic, the rules stipulated that units out of sight could not communicate with each other.

A shift from analogue to digital became inevitable as technological advancements increased and the use of computers became more common. In the beginning, the usage of these games were fairly limited, but as timed passed, just as the analogue games before them, more sophisticated and complex wargames surfaced, and in the late 20th century, realistic 3D video games paved the way for new training and teaching methods. For a more in-depth explanation about the usage of serious games in the military, read David Michael and Sande Chen's book "Serious games: Games that educate, Train, and Inform" and Roger Smith's article "The long history of gaming in military training"

Aside from wargames, different types of flight, tank, and vehicle simulators have been used to train military personnel. Simulators and simulation games are often used in military training because it saves time and reduce costs. Setting up a military training camp and mobilize necessary equipment requires a lot of time, effort, and resources. Furthermore, the margin of error is huge and can sometimes even be fatal. E.g. flying a plane for the first time would feel a lot safer if you had had the chance to practice countless of hours in an accurate flying simulation before actually trying to fly the plane. There is of course going to be a different type of experience, but at least you will have some experience compared to none. In addition to this, once the regular training session is over, everything needs to be disassembled and moved back to its original position for further use. In a simulated environment, this can all be controlled with the press of a button, where the user can practice and experience the same scenario multiple times for a fraction of the cost, time, and effort compared to regular training. Simulations also have the ability to induce certain events to streamline a particular type of training.

But should simulators really be classified as serious games? Simulator does not have any gaming properties since it is supposed to be an authentic representation of the real world in which it operates. But does a game need game elements and gaming properties to be classified as game? Microsoft Flight Simulator (which has been used by the military) can, for instance, both be seen as a game and a simulator depending on who you talk to. Based on that argument and our definition of serious games; purpose contra entertainment, we have to include simulators and simulation games into the mix of serious games as well. One could also categorize serious games along a continuum of gaming characteristics ranging from video game purposes to environmental experiences and emotions.[9]

[9] T. Marsh, 'Serious games continuum: Between games for purpose and experiential environments for purpose', *Entertainment Computing,* vol. 2, no. 2, 2011, pp. 61-68.

Despite differences and points of view, both games and simulators have facilitated military training for many years.

Back in 2003, the U.S. Army invested $4 billion annually on simulation equipment and wargames.[10] One of these investments were for the game *America's Army*. Which is probably one of the most popular and successful serious games of all times. Instead of purposeless killing the enemy with blatant combat scenes (which is usually the case in wargames), America's army tries to educate the player about the army life by simulating the experience and let player explore different scenarios and career paths. The game awards behavior essential to the army experience such as teamwork, responsibility and good values. The purpose of the game is to recruit new people to join the army by distribute accurate information about the army experience and the opportunities thereto. The game showed surprisingly well results and led to a more efficient and cheaper recruitment method for the U.S. Army. Furthermore, the game's success led to modifications to fit other areas within the military, such as mission preparation, bomb defusal practices, and virtual medicine training to name a few.

Since the military use simulations for the majority of their training, it is important to notice the challenges within knowledge transfer between different types of media. In most cases, the training environment and the actual workplace are two vastly different types of environments experienced under different circumstances. The surrounding in which the training occurs, affect the outcome. Therefore, a poorly designed simulation with an inaccurate image of the real world would not be as effective as the opposite. It is therefore important to understand what they are training for, and how that knowledge is going to be used in the workplace when designing simulators and simulation games. I.e. the best possible training is in the field or workplace. However, most of the time, this experience is impossible to achieve (e.g. military surgery during battle), therefore simulations are usually the next best scenario.

These military training session are obviously different from each other and various social and environmental aspects are decisive for the process in which learning occurs. However, in most cases an instructor or supervisor is usually present during simulations and game-based training. During our work with serious games we have identified and formalized training practices to facilitate learning. It is called the coaching cycle[11] and consists mainly of three phases:

- Scenario preparation – The instructor prepares the training session. The most prominent part of this phase is the creation of the training scenario. Depending on the type of training, different tasks need to be addressed.
- Gameplay – During this phase the instructor makes sure that the trainees are playing the game as intended and monitoring their progress which later can be used for debriefing. The instructor can also give continuous feedback on certain training aspects during this phase.
- Debriefing – This allows the participants to reflect upon the training experience by reviewing the game process and discuss alternative solution. Debriefing is crucial for any experience-

[10] D. Michael and S. Chen, *Serious Games: Games that Educate, Train, and Inform,* Boston, Thomson Course Technology PTR, 2005.

[11] A-S. Taylor, P. Backlund, and L. Niklasson, 'The Coaching Cycle: A Coaching-by-Gaming Approach in Serious Games', *Simulation & Gaming,* vol. 43, no. 5, 2012, pp. 648-672

based learning situation. The most common method for debriefing is simply a discussion between the instructor and trainees.

This cycle is not limited to military exercises but can be leveraged to other vocational game-based training domains as long as you adapt to the newly found context and purpose in which it is going to be used.

The serious games used in the military is not solely focused on combat and war. Games have been used to train soldiers overseas about cultural difference and languages to facilitate and help those in need in a better and more efficient way. The commercially released game *Full Spectrum Warrior* has been used by the military for leadership and decision making training.

2.4 Formal and informal application

It is important to also point out that, regardless of specific "topic" or "genre" of application, a serious game can be used in formal or informal ways, both of which place very different types of requirements on the serious game.

Take, for example, the challenge of creating a serious game created to facilitate physical rehabilitation. This type of serious game solution might need to work around specific types of hardware (e.g., crutches, wheel chairs) and perhaps mobility tracking and diagnostic software that might be present in a lab or rehab environment. But on the other hand, there could be immense value in attempting to create a serious game that makes it easier and more engaging for participants to do physical rehabilitation in their home environment at their own leisure. The issue with these two different ways of facilitating physical rehabilitation is that it is very difficult to accommodate for both of these things simultaneously. A serious game that works well in a lab environment needs to integrate and function well with the technology and procedures that are already present in that setting. Just creating something that does not take the context of the lab into account will at best feel superfluous or peripheral, and at worst obtrusive and unhelpful. But, if you develop your game to integrate well into the lab context, it might become difficult or impossible to use for individual users in their home environments where the same technologies and supporting processes are not present. The formal setting (in this example a rehabilitation lab) and the informal setting (the individual's home environment) are very different in terms of application even though the subject matter of the serious game might be the same.

Understanding the context of application is essential for a serious game project to be successful. A serious game is seldom implemented into a vacuum, and there are pre-existing processes and conditions to keep in mind. As a developer, you can either focus your efforts on creating a serious game that is specifically tailored to excel in a limited number of specific settings, or you can try to create a game that has broader usefulness and can fit into many different types of settings and subjects matters. In the following discussion, the approaches will be referred to as creating "tailor-made" and "adaptable" games respectively.

Both approaches have their own benefits and shortcomings, and in some sense you decide on your general approach as soon as you choose the business model for your educational game studio. When you create tailor-made serious games for education, for example, you often work closely together with different educators, schools, or different branches of municipalities – which means that you work on specific commissioned projects rather than coming up with your own products from scratch. In this

case, your consumer is already decided for you since it is the client you're working towards, and even though you might get to own the created intellectual property after the project is over it is often difficult to spread a specifically tailored educational game to a broader market. The internal workings of different schools can be very specific, so a game that is tailor-made to fit into one school might not work very well in others. The same goes for most types of organizations; there are often a lot of procedures, cultures, and practical situational factors that are unique to individual organizations, or even sub-groups within an organization.

If you create adaptable games, you have more opportunity to work on your own, but you take a bigger risk since your development process might not be funded by a project or an already existing client. But on the positive side your game will likely have better potential on an open market once it is completed since it can fit into many different educational settings – the caveat here is that it obviously places some additional responsibilities on the side of educators since they will need to figure out how to use your game in their educational setting themselves. To make your game more accessible in this regard you can provide some guidelines and examples of how a teacher can use your game. This can help you reach out to new clients since they might have no other way of understanding your thought process and intentions when creating your game.

So from a developer's perspective, there is a significant difference in how tailor-made and adaptable games provide revenue. Tailor-made games are usually created by commission from a client in education. The general marketability of the created game is likely to be limited and thus the game itself won't provide much revenue, but the development is somewhat safer since it's paid for by a client and the risks for you as a developer are thus very low. Adaptable games, conversely, have a wider marketability after their completion and can be a worthwhile pursuit for companies who do not wish to work on a contractual basis. A couple of companies that exemplify each of these principles are Serious Games Interactive in Denmark that primarily creates tailor-made games, and the Finish/American studio TeacherGaming LLC who are working with a highly adaptable game.

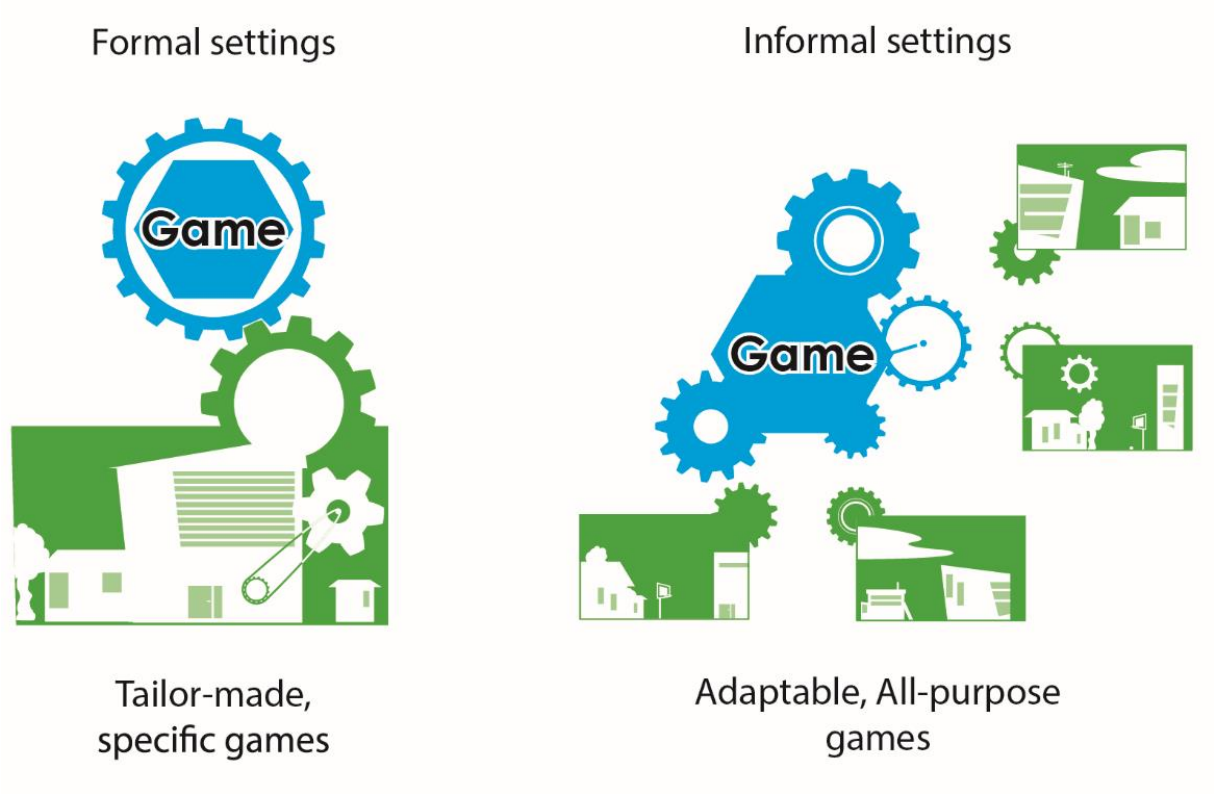

Figure 2. Formal and Informal settings place different requirements on a serious game project, but also presents different opportunities. A tailor-made game has a limited application area but might be easier to stream-line and develop, whereas an adaptable game might be more difficult to create but might reach a broader array of contexts and users.

In short, creating a game for a formal setting requires you to have a strong understanding of that particular setting. Your serious game needs to integrate and function well with the setting's peculiarities to be considered useful and impactful for the people within that setting. For informal settings, you instead face the challenge of making something that is broad and adaptable to fit into a broader amount of potential settings. For the latter, your potential market space increases, but at the same time your development costs will likely increase as your game needs to account for a larger amount of potential usage situations.

2.5 Effectiveness and shortcomings

The tricky thing about discussing the effectiveness of games is that people often want to see a universal, definitive proof that clearly states that "games are educational", "games are good for your health", or "games are good for brain development." The problem is that such sweeping statements cannot really be made, for the simple reason that games, the problems they are used to solve, and the people who play them, are all incredibly varied. Attempting to answer "how good are games for your health?" is as difficult as answering the question of "how good are pills for your health?" A pill can include a wide variety of chemicals in millions of different combinations and proportions, which makes the question impossible to answer without further clarifications. What type of pill is it, what type of health issue is it intended to help, and what type of person is taking it? The same is true for games: they can contain a million different elements in different combinations, belong to different genres, and they can be played on different platforms, with different interfaces, in different contexts, and by different types of people.

When discussing the effectiveness and shortcomings of games, it is important to keep this in mind, and not to expect simple answers. Even if we manage to identify research and development projects that proves the effectiveness or efficiency of serious games, we still have to battle the practicalities of implementations, user acceptance, and content validity. For starters, the level of difficulty and convenience of opening a book compared to starting a game is huge. However, if we disregard the previous discussed problems and focus only on the game itself, we can see that games, serious games, and gamification is a great way to promote knowledge acquisition across a wide range of topics and to a lesser degree skill and social skill acquisition and behavior change.[12]

Playing games can promote a wide range of cognitive skills, such as faster and more accurate attention allocation, higher spatial resolution in visual processing and enhanced mental rotation abilities. In addition to this, games have been speculated to be excellent tools for developing problem-solving skills and to enhance the player's creativity.[13] Immersive gaming sessions which perfectly balance challenge and frustrations with success and accomplishment show positive motivational tendencies, in which the player continuously tries to overcome obstacles presented in the game. Playing games can also improve player's mood, promote relaxation, and even moderate anxiety. The social stigma of playing video games have changed drastically over the past years and gaming is no longer seen as a socially isolated event. Instead, many people nowadays prefer to play games with or against their friends which could increase social skills and foster prosocial behaviors.

Serious games can encourage people who lack the interest and self-esteem and increase motivation and user engagement in certain subjects and application areas. They can also allow the users to experience something that can be too expensive, risky, or even impossible to achieve in the real world. I.e. using a simulated version, where players can practice multiple times on how to deal with and manage a cardiac arrest in a health environment without having to worry about the consequences. Games can form a unique learning experience based on participation and experimentation rather than mere observation,[14] which supports active, experiential, and problem-based learning. Games are, in their nature, *replayable* or repeatable. This means that the users can play the game again if the content or meaning was misinterpreted. Which also makes them useful for scenarios which requires multiple training sessions.

In multiple applications areas (education, health and exercise, and organizational systems and work to name a few), gamification has positive effects on motivational affordances and user engagement. However, the effects are greatly dependent on the context in which the gamification is being implemented, as well as who's using it.[15] Similar gamification solutions and implementations, transferred to a different context, does not always yield the same results.[16] In some cases, a gamified

[12] E. A. Boyle et al., 'An update to the systematic literature review of empirical evidence of the impacts and outcomes of computer games and serious games', *Computers and Education,* vol. 94, 2016, pp. 178-192.
[13] I. Granic, A. Lobel, and R. Engels, 'The benefits of playing video games', *American Psychologist*, vol. 69, no. 1, 2014, pp. 66-78.
[14] B. Berg Marklund, *Working with Educational Games: Fundamental guidelines for developers and educators interested in working with educational games,* University of Skövde, School of informatics, 2014.
[15] J. Hamari, J. Koivisto, and H. Sarsa, 'Does gamificaion work? – A literature review of empirical studies on gamification', *Proceedings of the Annual Hawaii International Conference on System Sciences*, IEEE Computer Society, 2014, pp. 3025-3034.
[16] K. Seaborn and D. Fels, 'Gamification in theory and action: A survey', *International Journal of Human Computer Studies,* vol. 75, 2015, pp. 14-31.

solution might not improve educational outcomes and can even decrease students' motivation, satisfaction, and empowerment.[17] In the fields of higher education, only a small percentage of the teachers use gamification on a regular basis, even though their attitude towards gamification is mainly positive. The main reason for prevention of technical solutions is lack of time, required technical training, and economical support.[18]

A fair amount of evidence has shown the positive effects of serious games and gamification[19] which goes to show that even if they are not superior to traditional teaching methods and learning materials, they can still be used in their own right effectively. However, there is a lack of longitudinal studies that can prove retentive learning and assess the long term effects of serious games exposure. Despite everything previously discussed, it is safe to say that, serious games is a new, challenging, and exciting research field with untapped potential and unlimited possibilities.

We have probably convinced you that serious games is the solution we have all been waiting for. That is not the case, serious games and gamification should not be seen as a "silver bullet" that magically solves all your problems. You cannot throw it against any obstacle, and hope that the mysterious magic of games dissolves it. It requires meticulous and careful planning between different disciplines and contexts in order to stand a chance against traditional working methods and processes, or to be effective on its own. But if used correctly, it can enhance the learning experience by increasing the motivation and user engagement, and subsequently, increase the pedagogical outcomes.

The area of serious games and gamification is not black and white, instead, there are many small intricate bits and pieces everywhere, whether it is between different people and their respective expertise, or the context of which it's being used that needs to be working in conjunction which each other in a harmonic and effortless way to be useful. As discussed throughout this chapter, serious games and gamification is not without its flaws and should not be seen as a substitute for traditional methods but rather, work as a compliment to strengthen current teaching and learning methods to create new and interesting ways of gathering information and acquire new knowledge. In our experience, games only truly fail at what they set out to do if the team seek out easy answers.

3 The fundamentals of Game Development

Even if you are commissioning a game and are not the one building it - it is important for you, as a client, to have a good grasp of some of the fundamentals of game development. Understanding game development will significantly increase the likelihood of your project succeeding, as you will be able to communicate better with the development team and will have a better understanding of what you can expect. A lot of serious game projects run into trouble because they are built on a fundament of flawed expectations and requirements - so knowing how complex game development can be will hopefully make you better at defining realistic requirements and also better at describing them to the development team. This chapter will be a brief and general introduction to game development to start

[17] M. Hanus and J. Jox, 'Assessing the effects of gamification in the classroom: A longitudinal study on intrinsic motivation, social comparison, satisfaction, effort, and academic performance', *Computers and Education,* vol. 80, 2015, pp. 152-161.

[18] J. Martí-Parreño, D. Seguí-Mas, and E. Seguí-Mas, 'Teacher's Attitude towards and Actual Use of Gamification', *Procedia – Social and Behavioral Sciences,* vol. 228, 2016, pp. 682-688.

[19] P. Backlund and M. Hendrix, *The Educational Value of Serious Games: Summary and Overview of Current Research,* Lifelong Learning Programme Education and Culture DG, University of Skövde, 2010.

establishing a common report in serious game projects. More detailed information regarding each individual part of a game project will also be offered in subsequent chapters of this report.

3.1 Goal specifications and requirements in game development

Game development constitute a unique merger of creative crafts and software engineering. One of the ways this unique merger manifests is when it comes to requirements engineering (that is to say, specifying what the produced solution is supposed to *do*). In software development, requirements are often highly utilitarian in nature. The software is required to securely allow a user to order products online, print documents from their computer, help a company keep their pay-rolls up-to-date, or keep security doors locked until they receive the right password. Since these requirements are firm and easy to test, it is often relatively easy to determine whether the software has achieved those goals. A large requirement on games, however, is to be engaging, challenging, fun, aesthetically pleasing, relaxing, or any other nebulous and subjective term. It is both hard to form a unified baseline for exactly what these requirements are - what one person considers to be challenging, fun, or relaxing might be considered easy, boring, or stressful for someone else. Since the requirements are so nebulous, it is also difficult to precisely identify when the game has fulfilled them.

This is where a lot of challenges in serious game development come from - it is hard to communicate goals, when the goals of the projects are often largely subjective. Compounding this challenge is the fact that the client or commissioner (which is a subject master expert in their own field) often operates within a widely different frame of reference than the developers of the game.

3.2 Iterative development

Even though games are a type of software, the craft of game development is quite different from software development - and this can sometimes take people by surprise. Game development is, first and foremost, an *iterative* process. Games rarely get created in a linearly progressing way, and they often get continuously reinvented throughout their development process. This is partly a side-effect of the subjective nature of requirements in game development. Since the end-goals are rarely 100% clear, game development requires room for improvisation and exploration. Exploration and improvisation helps a development team to clarify their goals over time. It is hard to know at the start of a project how different game mechanics and aesthetics will feel and play (what might sound fun and challenging on paper might be boring and tedious in practice). It is primarily through continuous testing and revision that game designs are refined to ultimately become a good end product.

These iterations are also dependent on where you are in the development process. In the early stages of the development it is easy to completely reinvent or revise an entire design of the game, such as changing the theme, switching between platforms, or rewriting the whole narrative. But as the project progresses these iterative changes becomes smaller and not as revolutionary. It may be about enemy locations, jump heights, or small gameplay mechanics.

A common misunderstanding of iterative development, however, is that it *just* means that you need to open yourself up to remain flexible during a project. The truth is that iterative development is a broad “genre” of development processes with defined structures. Iterative development can come in many different shapes and flavors (e.g., Agile, Scrum, Rapid Application Development, Spiral Development, etc.) that each bring their own unique variations and planning challenges to these

structures, but there are some fundamental truths that unify them and characterize any successful iterative project.

In traditional software development, development progresses through a series of hierarchical phases. The general agreement is that development does not proceed to the next phase of development until an earlier phase is fully completed. Iterative development is especially suitable for game projects, since games are made up from elements heavily connected through multiple disciplines (graphics, networking, programming, animation, music and sound etc.). This multidisciplinary relationship must be set up in an almost seamless and effortless way in order for the project to be successful and for the gaming experience to feel coherent.

Even though goals, requirements, and processes are more ephemeral than traditional software development, game development does follow a general framework of phases, and the early design work done in the pre-production phase is of course very important for the overall progress and success of a project.

Iteration means *kill your darlings*

One thing we will likely repeat many times in this report is: do not get too attached to your first ideas. Game development needs to be iterative to be successful. A game experience is a machinery that is made up of an immense amount of moving parts, and it is impossible to know how all these parts will flow together and how they will *feel* before you have started actually testing the game. In many cases, some ideas that look good on paper will turn out to be detrimental to the overall feel of a game and might need to be revisited and revised, or perhaps thrown away entirely. Clinging on to a poorly functioning game feature for too long might hamstring the entire project's progress.

It is important for the well-being of a project that team members don't get too attached to specific details of the project, and that they don't lose their motivation and creative drive if one of their ideas end up not being used. The best way to combat this is by establishing a mutual understanding of the transient nature of the game development project, and by maintaining a team culture of open and transparent communication.

3.3 The phases of game development

Broadly speaking, game development is divided into three phases: Pre-production, Development, and Post-release/Post-production.

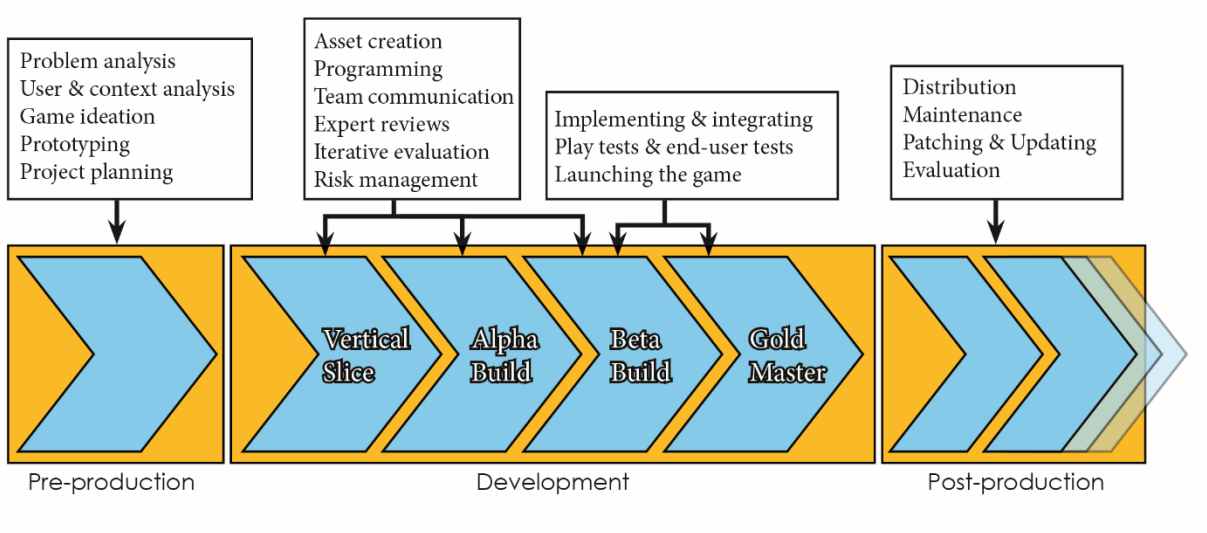

Figure 3. Overview of the different development phases and the general tasks that might be associated with each phase.

Projects start with a pre-production phase, which involves ideation and concept specification. In this phase, since little funds have been put towards producing content, the team working on the game has fairly free reigns to explore design ideas. The decisions made in this phase establish the general framework of the game project as a whole - the team needs to decide which technological platform they want to work with, which audience the game is intended to reach, the general theme and art direction of the game, and the time frames of the development cycles, and so forth. While a lot of choices are, as mentioned previously, often refined and revised during the iterative development process, some of these fundamental choices for the game project are irreversible. Re-building a game for a new technological platform or changing the art direction for an entire game, for example, often requires a total rehaul of assets and code, which often is not feasible during a development project.

After the initial trajectory of the game project has been established in the pre-production phase, the game starts iteratively taking shape during the production phase. An important thing to realize about game production is that early playable prototypes can often be produced relatively quickly. But, it is important to realize that a playable prototype is just that; a prototype. A prototype is meant to showcase the core theme or general 'feel' of a game to ensure that the project team are all on the same page. Thus, early prototypes constitute a pivotal point in the production process - and it is a situation that can really put a project's communication to the test. If you are a client or commissioner of the developed game, a prototype can come as an unpleasant surprise since it often looks woefully unfinished. It is important to keep in mind that a prototype is meant to be a cheap and effective early test of some of the fundamental ideas that are going to be a part of the final product. Thus, it is important not to get too hung up on superficial aspects of the prototype, and that the prototype is seen as an early framework of what the game might ultimately become.

After early ideation and prototyping has helped the project team decide on a unified vision for the end product, the nit and gritty production phase can start. As described previously, even though development will need to be kept relatively flexible, having a structure for how the different iterations development is going to go through is important to establish. For the remainder of this report, the different phases of production will be described in more detail.

Digital or analog prototypes?

Prototypes do not necessarily have to be playable on the target technology platform, and game development teams often employ pen-and-paper tools, animation programs, storyboards, or other cheap means to create prototypes early on. A prototype is often primarily used to exemplify or to hone in on the core design goal of a game and to see what types of gameplay situations a certain type of game mechanic results in (for example, seeing how a certain type of resources system would play out, or how a level layout, color scheme, or competitive aspect of a game feels). This can often be done quicker and more efficiently through analogue tools, and it is often easier to rapidly iterate and change physical components than digital ones.

Another benefit of analogue prototyping is that developers can showcase it fairly easy and early on with clients during early ideation workshops. Physical objects are easier to manipulate and play around with than digital ones, so this might have the added benefit of inviting clients and end users to the design and ideation process early on in the project.

4 Starting a Serious Game Project

Embarking on a serious game or gamification project can be a daunting task, and it can often be difficult to know exactly where to begin and what the first steps are. Depending on whether you are a teacher looking for a digital solution to inspire your students, or an independent game developer trying to help visually impaired people learn braille, or a well-established game studio trying to expand into serious games. These first steps of your game project will look slightly different. Of course, a lot of the things that go into the project will be figured out collaboratively between developers, commissioners, and users as it progresses. Whatever the case might be, this chapter will take you through the fundamentals of serious games development, to make sure that your project starts out with all the things it needs to ultimately succeed.

Regular games are mainly developed to entertain the player, they rarely need to adapt to special frameworks in order to be played properly. As long as the games are entertaining, they can be released with little knowledge about user context and target groups and still be successful. Designing serious games on the other hand requires a slightly different perspective. A serious game will be used (in most cases) within a functional organization with bits and pieces already working well together, competing against books, films, lectures and other types of perfectly fine media. Apart from the entertainment aspect, serious games also have the obligation to fulfill certain promises effectively in order to be successful. For this to happen, you need to carefully think about the game's purpose, it's target audience, and how it's supposed to be used and in what context. These are all valid points for regular game development as well, but the margin for error is way smaller for serious games than regular games. Therefore, conducting meticulous research on who, when, why, and how your games is going to be used is important. To have a proper understanding of your problem domain will facilitate difficult decisions along the way. If you are going to add another cogwheel to a working machine, you got to be sure it fits first.

Solid groundwork and research was an integral part during our time working with Karlsborg's Tourism board. The serious game was developed to enhance the visitor experience of an adventure tour within the historic military fortress located in Karlsborg. The game had a clear target demographic: young kids

with an interest in game technologies, but whom might not find the more passive experience of looking at a fortress particularly exciting. But rather than creating a traditional sit-down game experience or tablet/phone-based game, a novel interface was used and the design was carefully chosen to make the game novel and inviting for everyone. The proficient game literate kids could not really "take over" the experience, since it was new to them too. For the game to be successful it also had to be carefully designed around a pre-existing context. With this, the novel interface and historical context presented unique opportunities, problems, and challenges which were attentively evaluated and ultimately solved using previous research and experience. Since both the game and adventure tour were interdependent of each other, meticulous planning was used in order to create a fun and entertaining user experience where the game borrowed different elements from the adventure tour and vice-versa. Due to the uniqueness of this project, many of the difficult decisions could not have been made in later iterations of the project. The team could not, in the last few months of the project, have realized that the unique J-Dome interface would be suitable for the project goals, or realized that the transmedia components and cross-overs between the users' experiences in the Karlsborg's fortress were important to the game's storyline and added it in retroactively.

Figure 4. J-Dome interface (left), in-game picture (right)

Another example of these types of decision pipelines in serious games can be found in the game Braille Hero[20] (created by one of the authors of this report) where previous research heavily dictated the game's design. Even though the game was not developed in cooperation with an organization, the game still had a specific purpose, targeted toward an explicit demographic of end users with unique requirements and preferences. The purpose of the game was to teach blind and visually impaired people braille through smartphone games. Designing a functional smartphone user-interface for visually impaired people was proven quite difficult. Regular user-interfaces are often based on visual feedback and stimuli. However, in this case, these conventional design methods could not be utilized due to the fact that the target demographic was blind. Instead, the game's visual feedback had to be "transformed" into auditory and tactile feedback in an intuitive and entertaining way. By using previous research as guidance in the pre-production stage, fewer mistakes were made and certain elements and design choices could be assured in a different, more meaningful way compared to going into the project unprepared. By doing diligent research before the production of the game began, the development team could have a clearer understanding of who exactly their target end users would be, what their needs and requirements would be from the serious game, and how the interface and

[20] M. Hellkvist, *Braille Hero: Feedback modalities and their effectiveness on alphabetic braille learning,* Master Thesis, University of Skövde, Sweden, 2017.

gameplay would need to be designed to accommodate for these things. If the project had started without this type of premeditation, the project would likely not have succeeded; it would have been very difficult to realize the unique requirements of visually impaired end users late in the development process, and adapt a half-finished game to fulfil those requirements.

Building an initial familiarity with the application area and end users

Both Braille Hero and Karlsborg's Fästningsäventyr projects show the importance and benefit of conducting rigorous research in the pre-production phase. The important thing to know about this step, is that just a little effort and resource investment at the outset of a project can have immense benefits for the success of a serious game project as a whole. Sure, it is possible to spend several months doing diligent research and state-of-the-art overviews to be absolute sure that all the parameters of a project are mapped out - but pre-production research does not have to be that huge of an undertaking. Spending a few workdays or a week doing searches on Google Scholar, or getting familiarized with the use context and the target end users, can be enough to move a serious game project from just essentially being "risky and ad-hoc guesswork" to being "deliberate design and development".

This premeditation process can, ideally, be done cooperatively by clients and developers, so that they discover and form an understanding of their application area together as a team. But, it can also be done by clients and developers individually, or even before a full project team is actually assembled. If you are a client, the development team you start working with will certainly appreciate it if you come to initial discussions with insights or "core readings" that help push the project in the right direction. And the same goes for developers: approaching a client with a clear understanding of their use contexts, end users, and the nuances of their area of application goes a long way in early communication.

4.1 Why pre-production is important

The importance of pre-production is sometimes overlooked by people who are new to software projects, and perhaps to game development in particular. Projects often start out with a strong momentum; a creative spark in someone has ignited an initiative to create a game-based solution, and it is easy to be swept away by that momentum and to want to start getting to work realizing the idea as soon as possible. But, this positive rush to start creating can sometimes lead to future problems for a project and its team members. Giving the project team enough time in the pre-production phase to plan out the framework for the project is important to ensure that all the creative momentum and energy is channeled in the right direction. In essence, pre-production might seem like it can halt your creative momentum, but it is sure to save you plenty of headaches and unnecessary expenses in a project in the long run, freeing your team up to pursue creative directions instead of stumbling on obstacles that could have been easily avoided.

Too much short-term decision making at the outset of a project, without taking the time to make sure that the various important aspects that will affect the project in future stages are accounted for, can lead to unnecessary costly problems. It is often harder and more expensive to fix problems late in a project than it is to anticipate them, plan for them, and find ways to circumvent or handle them early. For this reason, it is always a good idea to have an agreed upon plan where the project has a complete "feature stop" – a threshold after which no new features or radical changes are introduced to the

game, and the development solely focuses on polishing and refining features and content that is already implemented. Introducing new features and ideas will simply become too cost-ineffective and risky after a certain point, so it is important to plan the development schedule accordingly. In ideal situations, roughly a third or a fourth of a development schedule should be devoted to the "post feature stop" period.

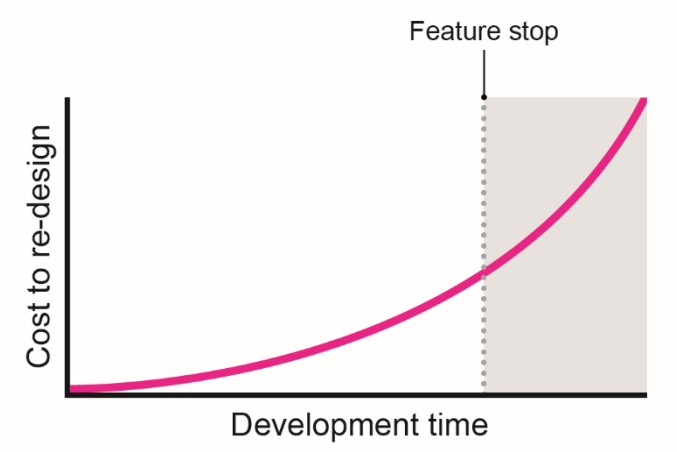

Figure 5. As the amount of content in a game increases, implementing or re-working ideas and features into it becomes increasingly costly as re-working aspects of a game at a late stage often requires re-working a complex system of game object relationships. For this reason, it's important to plan for a definitive feature stop date in a project.

The reasons for the rapid increase in costs is that inclusion of new ideas, or revising old ones, become progressively harder the further along a project is. Even though game development is, as mentioned in the previous chapter, a highly iterative process and that game projects often evolve and change over time, it does not mean that revisions of ideas or drastic changes of directions cannot have a significant negative impact on a project. In game development, changing components late in production can have several resource-heavy implications:

1. Even doing superficial changes means revising or completely re-creating assets that were time consuming to create in the first place, essentially meaning that valuable project time was wasted creating something that was not needed or creating something incorrectly.
2. Changing more integral components sometimes means revising an entire complex system of relationships between game objects. Game objects are parts of large interconnecting systems, so changing things are seldom as easy as just cutting something out or putting something new in; the entire system often needs to be adjusted.
3. There are some things that can be almost impossible to change without starting over from scratch. Changing platforms or implement a network solution usually requires a large amount of new code, simultaneously as old code need to be rewritten or removed. In addition to this, a lot of game assets need to be changed as well to fit the new format.

The first step in making sure that your project does not suffer needlessly from these types of costly mistakes, is to establish a firm understanding of your problem area, and by extension understanding what you aim to solve with your game-based solution.

4.2 Establishing a framework: analyzing the problem area

When beginning your project, it is of crucial importance that you understand its fundamentals. What is the problem you are trying to solve with the project? Why do you feel a game might be the right way

to do so, and how might that game look? Who, in the end, is going to be playing your game or benefit from it, and where is the game going to be played? In essence, the first thing you need to do is to be able to answer the **what**, **why**, **who**, **where**, and **how** of your project.

The specifics of these questions are unique to every project, and it takes some solid groundwork and research to ensure that they are answered well. You can, of course, figure many of these things out together with your development team, and you do not have to have all the answers to these questions yourself before starting a dialogue regarding a project. As a client, the main things you should have given some thought before contacting a developer is what your problem is, why you think a game-based solution might be a good way to solve it, and who the target audience for the game is. But beyond these fundamental ambitions and goals, how they might be best achieved is a good conversation to have with the development team in its entirety.

What: understanding what problem you are intending to tackle with a game-based solution, or "creating a problem statement" as it can be referred to in literature, is a natural first step in starting a serious game or gamification project. Problems can range from something small, like wanting a better way to teach geometry to your students or increase productivity in a specific factory pipeline, to something larger and more general, like creating an installation to entice visitors to come to your museum or creating a mobile healthcare application that is directed at the general public. This means that problem statements can either take the form of an embryo of an idea written on a napkin, or be lengthy essays that goes in-depth on the nuances of the problem. The bigger and more general the problem is, the more in-depth your problem statements needs to be so that the development team understands the nuances well as early as possible in the project timeline. Literature reviews, field work, expert reviews, and competition analysis can all be used to create a better understanding of your problem and problem area.

Why: after understanding what your problem consists of, you need to know why you think games might be the most attractive and effective solution for solving it. As we described in Chapter 2, there are many possible reasons for why one might want to turn to games to solve a wide variety of problems. It could be that games provide a certain "wow!" effect that can draw in customers or visitors for your business, museum, or event or help you stand out against competitors. It could be that games contain qualities that are uniquely suited to your problem, or that they seem to be the best way to reach your target audience. Knowing exactly why games is the tool you are choosing to use to solve your problem rather than books, films, lectures, or other types of software allows your team to start focusing their development resources on the right things to make a good game experience.

Who and where: refers to the end users who will be using, or playing, the developed game, and in what type of use context these activities are going to happen. Understanding these two things are crucial for ensuring that your final game solution can contribute to its usage setting in an appropriate way. A game, no matter how good the game might be or how impressive its production values are, will not be able to benefit its end users if it is poorly adapted to its use context. Poor context awareness when applying game-based solutions can essentially be like using a high-powered and expensive jackhammer to fix a leaking roof; an unnecessary and unruly way to solve a task that would be easier and more cheaply solved with a simple hammer. Instead of focusing your resources to create what seems impressive, or to try to mimic the production values of other high-tier video game productions,

it is important to think about what the context and the end users actually need and focus the design and development process to specifically provide those things.

How: refers to how the game will actually be used, and which types of working processes it is going to fit into. Teachers already have their own working processes and preferences of how to do things, as do instructors, organizations, healthcare professionals, and end users. How they will actually be interacting with the serious game, and how it will fit into their pre-existing habits, working processes, and lives are important to consider to ensure that the developed game does not feel like a complete departure from that the end users expect, or at worst an inconvenience for them.

All in all, answering all of these things in the pre-production stages of a serious game project is crucial for ensuring that the serious game can actually be impactful. A great looking game that plays like an amazing block-buster experience, might still fail to be impactful if the end users do not see it as being useful to them, if it functions poorly in its intended use context, or if it does not address its educational or informative purpose well. A game that excellently represents a subject matter in an authentic way, might still fail if it does not match the expectations and preferences of its end users.

4.3 Early communication and game ideation

Once you understand the basics of the problem you want to solve, it is time to start thinking about how the characteristics of the problem can start being translated into a game design document and a development plan. This is what is often referred to as 'early ideation' in game development. Ideation is a process that game scholars are having a somewhat difficult time mapping out. One of the reasons for this might be the simple fact that the origin of ideas, and the process of solidifying them into plans, are very abstract processes whose shapes vary between both individuals and teams. However, there are some helpful models, methods, and theories that can be helpful if you find yourself not knowing how to start codifying your general serious game ideas into workable development goals.

One method that can be useful for clients and developers to use together at the outset of a project is one called the Design Box.[21] The method encourages developers and clients to "box in" the design of their game by thinking about it through four different perspectives: what technology would make the most sense to use, who are our target audience (or end users), what is the question or problem we are intending to solve, and which aesthetic themes should the game employ to accommodate for these factors.

[21] R. Altizer and J. Zagal, 'Designing inside the box or pitching practices in industry and education', *Proceedings of the 2014 DiGRA International Conference,* vol. 8, 2014.

Figure 6. The design box consists of four walls (or constraints) that your project exists within. These walls are related to technological aspects, aesthetic goals, target audience preferences and their means of participating in gameplay, as well as the problem statement that defines the project's purpose.

The design box primarily helps a project find its ultimate goal and narrow down its scope to focus on that goal more effectively, and is a good way to establish the most necessary constraints that the developers will need to accommodate for. This can be valuable in a project that has a hard time finding its direction; a common issue with serious game and gamification projects is that goals remain too ambitious, or that the project plans include features that are either unrealistic for the project budget or unnecessary for what the game-based solution is intended to solve. In essence, the Design Box encourages a team to, in a way, go against the old axiom of "thinking outside the box", and to think practically and realistically about the constraints for which they need to accommodate. By first establishing the general framework for the project, by stating the basic technological requirements, end user characteristics, problem statement, and desirable aesthetics, the project team will have clearer common boundaries in which they can discuss the nuances of their project. It can be difficult and unruly to talk about design ideas and development plans purely from a "blue sky" perspective where everything is on the table, and "walls" of the design box helps establish a unifying perspective on the project and narrows the discussion down so that it can be more productive.

Technology Wall - This wall off is highly affected by external circumstances that the development team and clients might not have that much control over, since many organizations already have technology in place that the project will create a game for. This wall is established by asking: what platform (Mobile, tablet, PC, console) will the game be played on; are we required by the client to use any specific software; and what are the technological constraints and limits to this project?

Aesthetic Wall - This side of the box is about defining the emotions the player will feel and how the content should ideally affect the player. Does the client want a game that encourages fellowship and discovery between players? What type of feeling should the game evoke?

Audience Wall - This helps the team understand the target audience (or end users) to help them better make design and development decisions accordingly. Does our client have any particular groups of players in mind? Who will be playing the game, and where will they be doing so? Do they have

particular needs or preferences that should be driving the core design of the serious game or its interface?

Question/ Problem Wall - This side is connected to the purpose of the game and its expectations and what it is suppose to accomplish. It can either be a game mechanic, a genre, a theory, or a question. Have we been asked to make a specific type of game? What is the problem we are trying to solve with our game?

After the limitations and constraints have been established, an iterative process where old ideas get discarded or renewed starts. From this, new limitations and constraints are formed. This process then continues until the team has found and established a reasonable scope for the project. As previously discussed, formulating a problem statement can help with the process of narrowing down and finding your project's scope. A good problem statement tells both your goal and constraints. "How can we make a mobile-based game facilitate an understanding about braille amongst visually impaired?"

With this statement we have captured our goal, which is to facilitate an understanding about braille; we have also limited ourselves to certain constraints like mobile platforms and visually impaired players. With this, certain contextual and design constraints appear. Furthermore, a good problem statement leads to broader creative space, clear measurement, and better communication.[22] Which means that, with a well formulated problem statement, you can explore and find better and more suitable solutions. You have the ability to compare your proposed ideas against your statement to assess how well they solve your problem. You avoid confusion between team members and clients. With a definitive and clear problem statement a common goal is established.

For developers, it is important that you clearly communicate the value of your game to the clients. The best way to do that is by showing examples of how your game will be used in the final use-context. This helps maintain healthy expectations on the game and clearly communicates the purpose. It is important to know that it is more difficult to reduce the scope of a game mid-development than it is to increase it. Early ideation should focus on crucial components. Additional elements of complexity can often be added later on. It is easy to become too ambitious and get carried away with fancy solutions and intricate gameplay mechanics. Serious games, however, does not necessarily have to be revolutionary in terms of design and aesthetics. Conducting a "competitor analysis" in the beginning of your project can be helpful to identifying games and to find inspiration. Furthermore, showing a client "your game will have features much like X, Y, Z in this game" can be a good way to explain early ideation concepts.

'[I]t is important for serious game developers to understand that not all the learning needs to be in-the-box. This idea is not unfamiliar within the realm of entertainment focused games. Together with playing games, avid gamers read magazines and strategy guides, visit websites and share their knowledge and experiences. Players they learn both within and without the game.' [23] Games that only heighten motivation and engagement are not necessarily bad, they can create interest and curiosity to further explore the subject. Commercial off the shelf games (COTS) games are diligently used in the classroom because of this. Games such as *Civilization* and *Age of Empires II* have been used to teach

[22] J. Schell, *The Art of Game Design: A book of Lenses,* Burlington, Elsevier, 2008.
[23] A. Stapleton, 'Serious Games: Serious Opportunities', *Health Care,* vol. 1, 2004, pp. 1-6.

history, *The Sims 2* has been used for making and understanding complex social relationships, and *Rollercoaster Tycoon* to highlight engineering and business management.[24]

4.4 Understanding your end users and use context

One specific point of serious game and gamification we will frequently mention throughout this report is the importance of understanding how varied your group of end users will likely be. A common misconception about games and the people that play them, perhaps particularly when it comes to younger players, is that they are all very similar in their gaming literacy, interests, and technology proficiencies.

One prime example of these types of misconceptions is the notion of the "digital native" - a term that was introduced by Marc Prensky in his book "Digital Game-Based Learning" in 2001. The term gained significant traction in the following years, and is still widely circulated today as a truism of the younger "net generation's" technology proficiencies and interests for computers and games. Shortly summarized, the term is used to argue that people born after the mid-1980s, at a time when the world started becoming increasingly saturated with digital technologies, have a natural affinity for, and high interests in, technology. In essence, they are born into the "society of technology", and are thus natives to the world of tech and all that it entails.

While this argument might hold some merit, many researchers (ourselves included) would argue that it is a gross oversimplification of the extremely diverse nature of people.[25] [26] [27] Even in a single classroom of children, for example, you will find different wills, backgrounds, and interests. All of these characteristics significantly impact the ways they interact with technology. Due to all this diversity regarding audience, setting, and context. What works in one environment, does not necessarily work in another, Even if the organizations operate in the same application area (e.g. education), there will still be differences that set them apart. It can be anything from rules, policies, administration, staff members, IT-department to curriculum, technical accessibility and knowledge, and the way they operate. Furthermore, there will be a difference between a serious game developed for primary and secondary school due to age, interests, and what they considered to be entertaining. Together with the purpose, this heavily influences the game's design and its development. Under these circumstances, what would be the most appropriate solution? What genre and game type do you focus on? In some cases basic puzzle games can be effective, and sometimes games with higher immersion, interactivity, and fidelity is needed. Should the game contain violence? People are affected by violence in different ways, a heavily violent game might not be suitable for younger audiences. The school might even have strict policies against violence. Will the intended audience like this game enough and will it facilitate learning within this specific group players? What's fun and intriguing within one user group does not necessarily have to be true for another. Understanding your audience and their environment is a vital, and crucial part for a successful project.

[24] F. Bellotti, R. Berta and A. De Gloria, 'Designing Effective Serious Games: Opportunities and Challenges for Research', *International Journal of Emerging Technologies in Learning,* vol. 5 no. SI3, 2010, pp. 22-35.

[25] B. Berg Marklund and A-S. Taylor, 'Educational Games in Practice: The challenges involved in conducting a game-based curriculum', *The electronic Journal of e-Learning,* vol. 14, no. 2, 2016, pp. 122-135.

[26] Guthrie, C.H. "Who Are We Teaching? The Learning Expectations of "Digital Tribes" in the Classroom", International Journal of e-Education, e-Business, e-Management and e-Learning, Vol. 4, No. 2. 2014.

[27] C. Jones, R. Ramanau, S. Cross, and G. Healing, 'Net generation or Digital Natives: Is there a distinct new generation entering university?', *Computers and Education,* vol. 54, no. 3, 2010, pp. 722-732.

One of the most important things to mention about end users when it comes to serious game development, is that it is crucial for developers to understand that they are creating something for a highly non-traditional gaming audience. Since a lot of serious game developers have a background and interest in gaming and "regular" game development, they can sometimes lose track of how their own expertise influences their design choices and assumptions about how games are played. A lot of serious games are created to be put in the hands of people that do not have a lot of previous experience playing games, and game mechanics and designs that might feel very rudimentary or "intuitive" to a game developer might feel foreign and indecipherable by someone who has rarely, if ever, played a video game. Simple concepts such as "WASD movement" coupled with mouse aiming, or various genre conventions that developers are accustomed to, can be new to end users. Building a strong connection and understanding of end users early in the development process is thus crucial in steering the serious game's design in a direction that the end users will not only find engaging, but also be able to enjoy.

Building a *real* understanding of end users and end contexts

For some, this might sound somewhat strange, but we would highly encourage any serious game team to have a "field trip" early on in the development process where the development team gets a chance to see and experience how the organization they are working with operates on a daily basis. Reading about, or discussing, use contexts and end users can be informative, but there are a plethora of different small nuances that text and words cannot really capture and which can only be understood by direct observation. A game developer will see things in a use contexts and end users that the client, who is already very familiar with these things, might understand at such an innate level that they do not think of them as novel, interesting, or unique. Furthermore, developers can look at a context through their own "lenses" of expertise, and either identify possibilities or constraints in terms of game interactions in use contexts that a client might not have thought of themselves.

This advice was best described to us by a developer in a workshop we held at the University of Skövde together with representatives from local serious game studios. In their advice, they described how their development team would have opportunities to visit a school they were collaborating with, and that through just a few visits the team had gotten insights they never would have arrived at otherwise until way later in the development process (where it might have been too late to put them to good use). They realized that the technology in the classrooms was more limited than they expected in some ways, and that the students at the schools had skills and preferences they could incorporate in their design to make the game more suitable for them. In short, visiting the use context and end users helped their team designing their game with the peculiarities and characteristics of these in mind.

4.5 Setting up a team

As we have already mentioned several times, serious games and gamification are a unique merger of game experiences, playful representations of subject matters, and purposeful real-world application. These different components of serious games need to be reflected in the makeup of the team that are developing them. Whereas "regular" game projects can usually function well as long as the developers have the skills necessary to ideate and create an engaging game product, serious game and gamification projects also require the development team to have expert knowledge in different application areas and subject matters, and an understanding of the complexities of real-world

technology implementations. In essence, the team needs to be highly interdisciplinary and knowledgeable in topics that reach far outside the field of game development.

Of course, the knowledge requirements for a team varies depending on what type of game you are creating, and in what sort of setting it is to be used. Often, the knowledge requirements will be met fairly naturally by many serious game projects, since they often start out with a customer from a business, museum, school, or other organization seeking out a game developer to collaborate with. Thus, the client can contribute their own understanding of their own field of practice to the game project, and together with the developers the team can pool their knowledge to build a healthy understanding of the problem space and realistic expectations on the developed product. However, this type of knowledge merger should not be expected to happen automatically at the start of a project - each party needs to deliberately contribute their expertise, and be open about aspects of the project of which they are still uncertain. Members of an organization, for example, might not be 100% aware of the different types of factors that might complicate or affect the implementation and use of a serious game and game designers might not understand how the business is organized and what is required to keep it floating. Meeting halfway and, exchanging knowledge and expertise is crucial for success of the project.

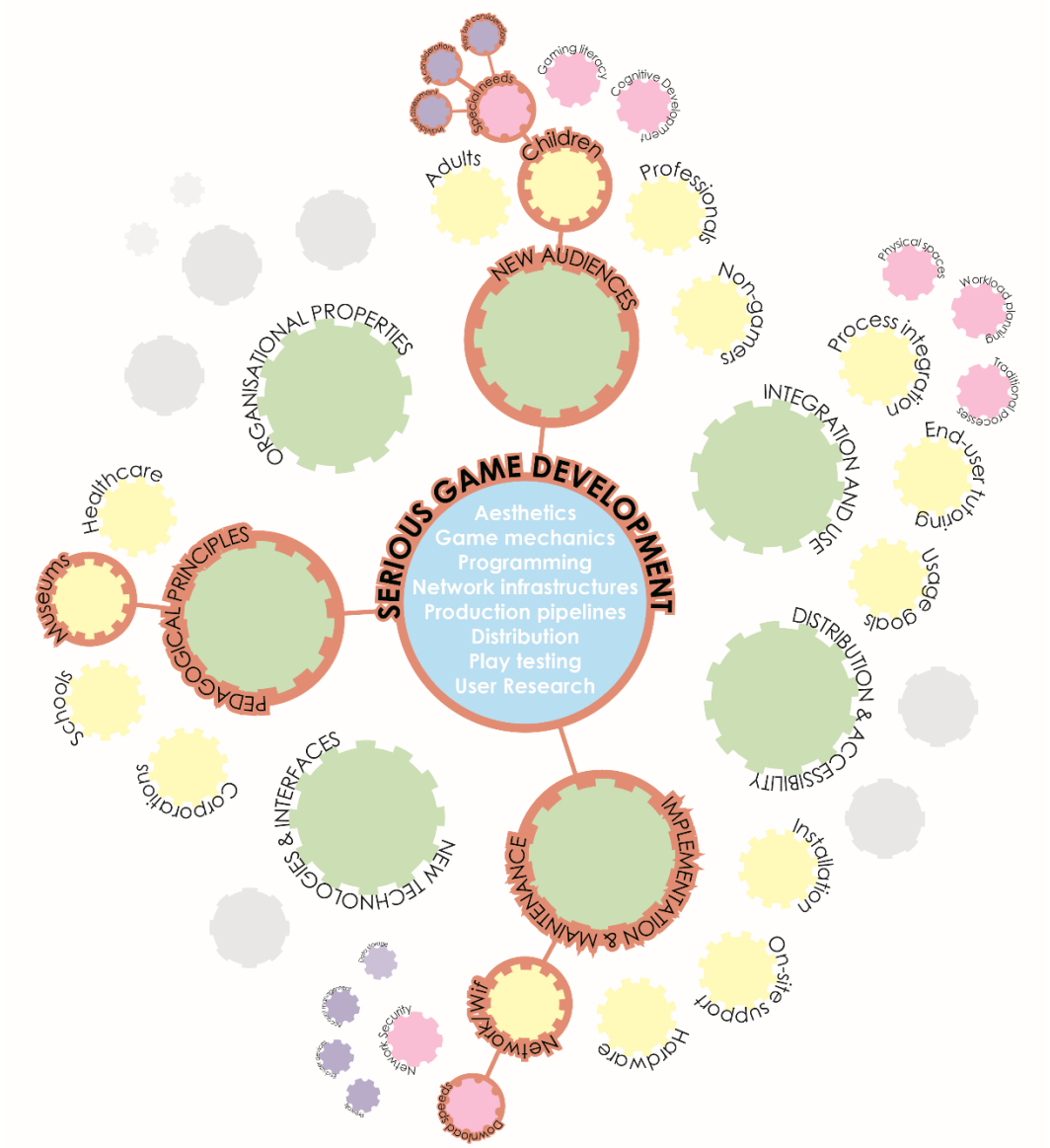

Figure 7. There are countless amounts of knowledge disciplines and areas of expertise that can be relevant to serious game development. A specific combination of knowledge and skills will be needed depending on the specific purpose and problem statement of your project. It is important to take a thorough inventory of the expertise already present in your project team and to highlight gaps in your communal knowledge where you might need additional consultation.

In essence, a team that is creating a game-based solution need to be able to connect the different "dots" that are pertinent to their specific project. Since every serious game project has a unique constellation of dots (e.g., specific requirements, usage environments, characteristics of end users, etc.), the best team setup for any given project will also need to be fairly unique. The team knowledge needed to create a great game-based museum installation piece will be quite different from the team knowledge needed to create a good game-based solution for at-home healthcare, for example. They will share some core similarities when it comes to game development processes, but accommodating for the physical space and technological maintenance procedures of a museum, as opposed to the particular needs, technology proficiency, and preferences of the target demographic of a healthcare solution, are two entirely different challenges for a serious game team.

Managing and planning for the type of expertise and knowledge that a project will need, is another reason for why a thorough problem analysis in the pre-production phase of a project is incredibly important. Small knowledge gaps can lead to large issues in later phases of a project. Whether it is because the team failed to account for some situational factors in the solution's use context, or technology proficiency and preferences of its end users, re-designing or revising a game-based solution late in a project is far more expensive than accommodating for these challenges early on when doing design work and planning the development pipeline.

From our experiences, we have seen many promising serious game projects fail in late stages due to unforeseen issues that could have easily been avoided. Maybe the team did not properly account for the limited technology available at a school, creating something that is too complex or with too high system requirements to be useful for teachers and students. Or maybe the team did not consider the budgetary constraints of an organization when it comes to acquiring and doing maintenance on software, making it difficult for their primary consumers to purchase and use their serious game. Including end users, consultants, and subject matter experts in whatever field you happen to be working in early discussions and workshops can help the project team circumvent or foresee these types of potential issues and make them more manageable than they would be if they emerged as a surprise in the late production or post-production phase.

SAREK Ambulance Training; a case of interdisciplinary work

The importance of an interdisciplinary team can be exemplified with one of our more complicated projects, the SAREK ambulance training center project. Shortly summarized, SAREK aimed at gamifying parts of the staged training exercises for ambulance drivers to make the situations feel more immersive and more in line with what the ambulance drivers might encounter in real-world situations.

Figure 8. Overview of the SAREK training facility and its tools. SAREK consisted of both real-world physical objects (the ambulance and the medical training doll) and virtual objects (specific scenarios and videos). By using 'live action roll-playing' inspired ways of presenting these scenarios and videos, SAREK aimed to make the training exercises more impactful and efficient.

This project included various context- and subject-specific aspects that reached far beyond what one would encounter in a "regular" game development project. Knowledge of health care, of ambulance drivers working procedures, the teaching styles of instructors, as well as methods of implementation and validation of training methods, were only a few of the core components that was involved in the gamification project.

The project managed to overcome these challenges by establishing clear roles of research and communication, primarily based on the expertise of the people involved in the project. Developers were given the main responsibility of describing what type of game-based elements could best contribute to the training experience, essentially getting the rest of the project team up to speed both with what gamification is, what sorts of solutions and implementations they could expect, as well as the challenges the project might face going forward. Other members of the team were tasked with researching why the training procedures were built they were originally, to determine which parts needed improvement and how they could be modified. Members who worked in healthcare, and instructors that dealt directly with ambulance drivers, were tasked with describing their work context and their working procedures, so that the project team knew what type of environment and what type of working procedures the gamified solution needed to fit into.

By setting up deliberate processes of both shared and divided responsibilities, ensuring that all involved stakeholders were able to share their expertise, and ensuring regular communication between the different members of the project, a complicated system could be mapped out and a suitable development process could be established. In essence, the complicated constellation of "dots" were connected through thorough pre-production work.

4.6 Choosing technologies

Choosing which technological platform or device your game is going to be created for is not a choice that should be made lightly. More so than any other choice you will be making in the early stages of a project, your choice of technology is something that likely cannot be changed down the road without wasting a lot of invested resources. As mentioned earlier, different technologies present a project with different types of challenges or benefits. Working with well recognized and ubiquitous technology (e.g., smartphones or personal computers) can, for example, make the distribution of your game easy and cut down on development time - but you will also be competing against other pre-existing products and users' habits on those platforms, and you will need to work within technological restrictions inherent to those platforms as well. You can also work with more obscure technologies or create customized technologies for your specific project - this gives you more control over the technology and an ability to ensure that it suits your users' needs and can give your game a more novel and attractive feel. At the same time, creating tailor-made technology can be very expensive both to develop and to maintain since the people involved need to understand the specific pieces of hardware to be able to work on it. If you opt for a tailor-made technical solution, it is essential that you understand how it work in its given context, because compared to "regular" platforms and technologies, a customized system biggest weakness is its lack of use and the uncertainty that follows.

Every company operate on a strict budget and, in most cases, they prefer to use what they have instead of investing into new and better things for the sake of performance and speed. Which means that these serious games also have to function and run smoothly on older and often outdated hardware.

Sometimes, the organizations only have a handful of computers available, which means that the games might be played by groups of students instead of individuals. These contextual aspects are more than likely to interfere with development and early design processes. Normally, the purpose, application area, and intention are likely to dictate both the hardware platform (computers, laptops, or smartphones etc.) and input device (touch screen, joystick, web-cams and motion tracking systems). In some cases a regular computer will do just fine, while in other cases (such as Karlsborgs Fästningsäventyr mentioned earlier) a more elegant, context-appropriate, and custom-made solution is needed.

Designing for end users: custom hardware and "One button play!"

In the project Elinor we developed a library of games as well as a gaming console to be used for the purposes of stroke rehabilitation. While creating customized hardware was a time-consuming process, it was deemed necessary since it allowed the product to accommodate for the end users' unique needs. The end users had a few unique characteristics that no off-the-shelf technology could accommodate for, but two of the most important ones were their limited interest in technology and low capability of interacting with modern computer interfaces. In essence, if they were to interact with our rehabilitation solution, it needed to be as easy as possible for them to access and use it.

Figure 9. Game interaction with the game developed for the Elinor project.

With this in mind, the hardware of Elinor was designed to favor usability and remove everything that was not directly contributing to the rehabilitation process. Every single component or function that was superfluous to this aim, even though it might have made the games look more visually impressive, or give the game cabinet some extra flair. An important thing to keep in mind is that everything added on top of the core concept of a serious game has the potential to be distracting and detrimental to the game's ultimate goal. For Elinor, which had end users with very specific cognitive and physical limitations and requirements, the purpose of every component was carefully evaluated, and everything that did not directly contribute to the project goals was disregarded.

4.7 Prototyping

There is no such thing as bad ideas. They can be the starting point of something great. Bad ideas can spark new, interesting, and challenging ideas within your development team. Prototyping is a way of testing and evaluating these ideas to predict their ability to fulfill its purpose and to get an understanding and feel for how they will work once they are realized. A well-designed prototype can demonstrate the practical usefulness and benefits of the solution. However, as a developer you will not have time to prototype and evaluate every idea you come up with. So it is important to sort, prioritize, develop, and test only those you believe in.

So why do we prototype? Can't we just start by creating the game? Well, you could. But that would not be a sustainable, and in the long run effective solution. As we have discussed previously, changes made during the production phase and later is problematic, costly, and removes countless of hours of hard work and dedication which could have been spent more wisely from the beginning. I.e. it is much easier to change the screenplay, than to reshoot the whole movie. A prototype is a cheap and effortless way to create a proof of concept and to show abstract things in a practical and graspable way. To describe an idea or a game mechanic to someone can be complicated (especially if they are not experience in games or software development), it is much easier to just show them the idea and let them experience it first-hand. With a prototype, both parties (developers and client) can communicate their requirements, possibilities, and limitations of the technology in use. Prototypes are supposed to be lightweight, fast, and cheap to develop. This does not make them useless. Instead, it forces you to focus on what is important, mainly the core experience. It is a real challenge to turn a mundane gameplay idea or core mechanic into something fun and exciting by adding more elements and complexity to it. You will still have the same old boring core in the middle, but now, covered in glitter. Prototypes help with this process and make sure that we move on with a solid foundation and structure to ensure a successful project.

When creating an early prototype, it might actually be important to not make it "too good". Features that look too polished might be misinterpreted as final versions of those features, clearly demarcating components as being placeholders can sometimes be important. This is, of course a bit of a double edge sword, you do not want to have a prototype that looks bad enough that make the client doubt the project. It is all about finding a good balance and (as always) communicating clearly.

Prototypes for games can take many different shapes and forms, and they do not necessarily have to be digital. In most cases, a simple paper prototype will do just fine. They are fast and can often capture the essential part of the gameplay, which makes it easier to find and pinpoint problems early on. Almost every game or gameplay mechanic can be translated into a paper prototype. Real-time games for instance, can be adapted to a turn-based play style. The experience will obviously be different, but a well-crafted prototype can still manage to capture the core values of the gameplay. To demonstrate this, we have described a way in which the classic arcade game *Bomberman* could be prototyped:

Start out by drawing your Bomberman maze on a graph paper and build various game pieces to represent the different players, enemies, and obstacles. To play the game, you are going to need at least two people controlling the different characters. You can make the movement and actions either turn-based or time-based. I.e. you can use a dice or a clock to determine how and when you are allowed to move the different characters. The gaming experience is obviously going to be slower than the real game, but it gives a good overall feeling of what is working and what is not. By using this

method, you can also test for different map sizes, layouts, and power-ups to see how that will affect the game and how the player interact with it.

Just like everything in life, there is a good way, and a bad way of doing things. This is also true for prototyping and some of the most important things to avoid are:

- Too few prototypes – They are meant to be fast and cheap, so why only make one? It is better to weigh multiple prototypes and their strengths and weaknesses against each other instead of sticking with the first "okay-looking" one.
- Taking too long – This is obviously related to the scope of the project. A prototype for a two month project will look different from a three year project. Bigger projects usually require solutions to more complex problems. Therefore, a more elegant and sophisticated prototype is more appropriate. It is also important (but challenging) to understand when enough is enough. A maximum of 20 percent of your project schedule can be spent on prototyping, after that you should have a playable version of your game. [28]
- Inefficiency – Focus on what is important. Do not spend valuable time on non-essentials, do not worry about straight lines or color of various game pieces. Everything does not have to be perfect. If you can do it quickly, do it quickly.
- Group prototyping – Staying in one big group during prototyping is a mistake. It is more beneficial to divide the team into smaller groups (or even individuals) and work on separate prototypes. This goes hand-in-hand with the first statement. The more prototypes you produces, the bigger is the chance of creating something really good. With smaller and more groups the odds of creating the same type of prototype is also very slim. The ability to share and combine different ideas afterwards can also be helpful for further prototype iterations.
- Be ready to throw it away when it is done – It can be difficult not to become attached to a prototype, especially if it works well and feels good to play. However, sometimes continuing development with a prototype as a foundation can lead to problems down the road. A prototype is seldom built rigorously (e.g., patchwork cod and designs), and building a game upon a hastily put together core can result in an unreliable or unruly development process; a prototype's functionalities are seldom created with long-term expansion and scalability in mind, so creating co-dependent game mechanics on top of them can lead to expensive problems.

To summarize, prototyping is rapid, highly iterative, and is meant to be a quick-fire way to test out the boundaries of what the final product might look like. The prototyping phase is when you put all your biggest and craziest ideas to the test – you are free to change themes, include brand new mechanics and game systems, and change storylines entirely. It is highly unlikely, or at least very unfavorable for the project, if big ideas are only introduced later in a project, so the early prototyping phase is where you have got a low-risk environment to experiment.

[28] K. Salen and E. Zimmerman, *Rules of Play – Game Design Fundamentals.* Cambridge, Massachusetts Institute of Technology, London: The MIT Press, 2004.

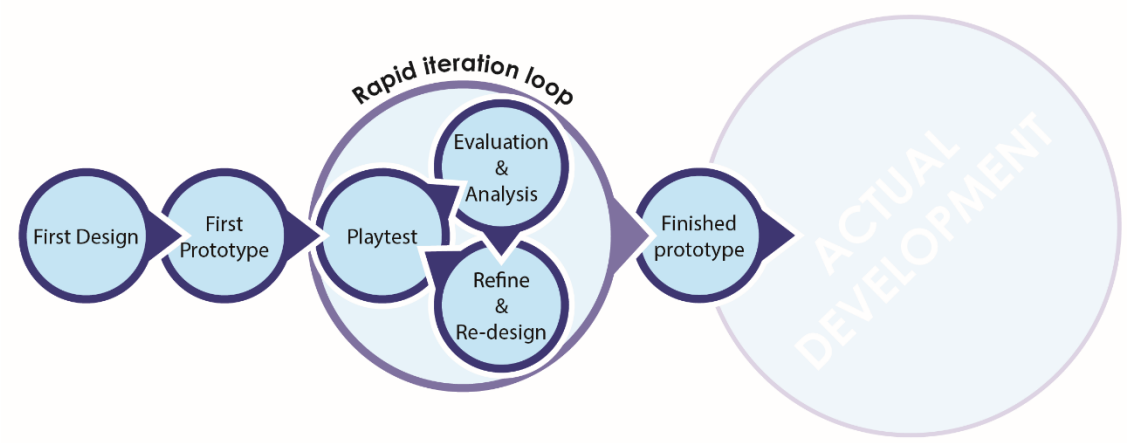

Figure 10. The different steps involved of rapid prototyping. As with everything else relating to game development, it is important to keep the process iterative. In prototyping especially, iterations should be very rapid. Depending on the project's size, they can be weekly, daily, or even hourly. The main goal is to quickly try out ideas to get a better understanding of what type of serious game you want to ultimately create with the project as a whole. Asset polish and finely tuned gameplay balance is largely irrelevant at this point, and it's important to keep an eye towards the bigger picture rather than details.

In essence, a prototype is all about setting your project, and the expectations and vision for the entire project team, up in the right direction. It serves to give a team a unified view of what game they are actually going to be making, what makes it interesting and unique, and which types of aesthetic themes, emotional responses, and goals they will be aiming for with everything they do during the project's future development journey.

Prototyping means RAPID iteration

The more iterations you go through, the more you test and explore your game ideas, and the better the end product will be. The development of your serious game should only begin once you have finished a functional prototype that exemplifies the type of game you would like to make.

Important to note, is that the prototype should not be seen as the foundation of the actual product you are making in practical terms. Prototypes should be built rapidly without much care taken for scalability and long-term stability – it is only built to showcase how ideas and solutions will be shaped. This means that it will contain shoddy code and hastily put together assets. When the prototype has served its purpose: throw it away and start a new so that you can start building a solid foundation of your product now that you have a better understanding of what the end product is going to look like and what types of features, technology, and end user interactions it is going to include.

5 Designing and Developing the Serious Game

The creation of serious games is an act of balance between subject matter authenticity, educational or informative content, and engaging gameplay. Each of these components are equally important, and constantly need to thought of as a mutually co-dependent system throughout a serious game project. If the game is entertaining, but not educational, then what is the point of using it? If the game is educational, but not authentic, then what is the purpose? If the game is authentic, but not

entertaining, who will want play it and why would we use a game instead of some other, more traditional, solution or tool?

While these components are all crucial for a serious game to be impactful, they can be hard to unite. Sometimes, purposeful content is difficult to make engaging, and sometimes engaging gameplay can be distracting or harmful to an educational or informative goal. Uniting these components require frequent compromises, good communication in a team, and a good understanding of both the serious game's subject matter, its gameplay potential, and its target end users. During the development process, certain decisions often need to be made where you are forced to make compromises and find ways to reach a good balance between engaging gameplay and purposeful content (whether it is to educate, inform, motivate, or convince). Some choices will affect the educational aspects more, than let's say the authenticity. Do we favor an entertaining gameplay mechanic in this disputation? Or is it more suitable to opt for a more authentic, but maybe boring approach? The better of two worlds would obviously be an entertaining gameplay mechanic that intentionally depict the subject matter in an authentic way. However, this is not always possible.

5.1 A story of balancing gameplay and purpose

In a serious game project we conducted together with local game developers,[29] these three components were carefully balanced against each other to create the game Testament. The game is an action role-playing game which focuses on the story of the Old Testament. It also contains a lot of interesting, and maybe unusual, content, which led to a lot of interesting and difficult compromises and design choices that exemplify how tricky serious game development can sometimes be. Here, we will use the Testament project to show how terms like immersion, gameplay, game mechanics, flow, narrative, avatars, end users, authenticity, and game assets often manifest in game development and factor into a serious game project.

The game was designed to be used during teenagers' confirmation education (the process of confirming your religious faith) in the Church of Sweden. The ultimate purpose of the game was to increase the end users' interest and understanding of the Old Testament, a part of the Bible which can sometimes be particularly difficult to grasp due to the many events and names that are correlated in complex ways, and it can perhaps also be a bit wooden. Essentially, the Old Testament is to the Bible what the Silmarillion is to the Lord of the Rings. This core goal, and a profile of the target end users, already set the tone of the development project: we needed to prioritize engagement with the subject matter, but at the same time not make too many concessions in subject matter authenticity.

During this project the most challenging and difficult decisions were to: decide who or what the main playable character should be, portraying the story of David and Goliath, and what to exclude from the old testament to the benefit of streamlining gameplay and reduce the scope of the project to something more manageable. These choices were continuously negotiated and discussed with the representatives from the Church of Sweden. These changes ultimately affected many aspects of the game such as, characters, dialogs, properties, and environment.

[29] IUS Innovation, formerly known as Immersive Learning

5.1.1 Player avatars, immersion, and authenticity

For the development team and the client, the choice of the player's avatar[30] was between recognizable people from the Old Testament, or a nameless angel. The core dilemma of this choice was that playing as a known character from the Old Testament might have been more immersive and authentic compared to playing a more inconspicuous unknown angel. At the same time, however, it would also have been impossible and immersion-breaking to play a known person throughout a whole storyline that unravels over a timespan of roughly 3600 years - a person rarely lives that long, so the player might have a hard time being immersed in the game if their avatar, a human being, is totally unrealistic in this way. The only way to make a regular person work as an avatar without really challenging the player's immersion would be to have the player to take control of different known persons between different time periods. This could, however, disrupt the players' connection to their avatar and disrupt the flow of the game as they would need to reconnect and identify with a new person for every playable time period. Jumping from one character to another like that is also very unusual in role-playing games. For these reasons, the team decided on having a nameless immortal angel as the playable character. By doing this, they avoided a lot of the problems stated above to favor immersion and character progression, but they perhaps did so at the cost of authenticity since the character was no longer an authentic biblical figure but rather an angel invented by the development team. These types of compromises between immersion and authenticity frequently need to be made during development projects, and there is seldom a clear right or wrong solution to design issues such as these. This is one of the many reasons pre-planning, iterative work, and open communication between the people involved in a project is important.

5.1.2 Gameplay, engaging narratives, and authenticity

The battle between David and Goliath was a particularly action filled event in the game, and was deemed to be a suitable climactic ending scenario for the whole game. However, according to the Old Testament, the battle between David and Goliath itself was only one part of a longer series of event, and many significant things that are central to the Old Testament happened after the battle. For example, the arrival of the prophets, whom are crucial for conveying the core messages and teachings of the Old Testaments, happened after the battle. So, from the perspective of gameplay and the game's narrative, portraying the events in a strict chronological order would not work particularly well, which put engaging gameplay at odds with authenticity. The team concluded that continuing the game after the climactic defeat of Goliath would be too anticlimactic of an ending after so much build-up throughout all the action scenarios the game contained. So, instead of having the prophets appear after the battle, they chose to introduce them much earlier in the game by allowing the players to peek through time to visit and converse with the prophets. This solution was both against the authenticity and the chronology of the Old Testament, but it made for a better and more pleasing ending, where engagement and narrative build-up was prioritized over subject matter authenticity.

5.1.3 Development scope

Beyond making choices in keeping content from the source material authentic and creating an immersive, cohesive, and engaging narrative and gameplay progression, the development team also needed to ensure that they kept the game inside a manageable development scope. The Old Testament can be pretty unruly in its length and complexity, and it would be impossible to represent

[30] The main playable character in a game is often referred to as an 'avatar', it is a term that originates from Hinduism, and essentially means a physical manifestation of a deity

all of it in one game unless you had virtually unlimited resources and time. By necessity, the developers and clients would have to choose which portions of the Old Testaments were most essential to convey its core teachings and messages, and to cut out the parts that were not as necessary. This is a common dilemma that many serious game projects face: making games is an expensive and time consuming process, and more often than not aspects of a subject matter need to be simplified or excluded in order to make a project realistically executable within a given budget. The compromise here comes down to keeping the core tenet of the subject matter intact, while also making concessions in order to make the development scope more feasible. In the case of Testament, many choices were made based on what would be best for the end user experience. Many of the events taken place in the bible, would, compared to modern day culture be considered old fashioned, obscure, or difficult to relate to. These events, which might not have been appropriate or engaging for a younger target audience, were either excluded from the project or shortened down significantly with a "lighter" description. For the development team, this meant that they would not have to create as many game assets (i.e. creating graphics, models, game mechanics, and stories for characters, enemies, and locations), which significantly cuts down on a development time and costs.

These components, and how they interact are dependent on the project. Some serious games require high level of authenticity and realism while others require a more fun and entertaining learning experience. It all comes down to what is important for the end goal and purpose of the game. Whatever the case might be, these three aspects needs to be working in conjunction with each other, where a give-and-take relationship is established between them. This is a challenging and almost impossible task to assess and foresee in the earlier stages of development and can only be solved once they surface. Therefore, in order to balance these things, an iterative development approach is necessary.

5.2 Iterative development

Before we start talking about the different stages involved in the early phases of a game development project, we feel that it is important to first reiterate one of the most fundamental traits of game development projects: that they are *highly* iterative by nature. This trait can sometimes catch people new to the practice of game development by surprise. As opposed to other types of projects and products, games do not take shape in a linear fashion where each step results in obviously visible and palpable progress and improvement. Games are built in a series of small increments, where ideas are frequently revisited, refined, or revised along the way. This process then repeats itself throughout the whole development phase.

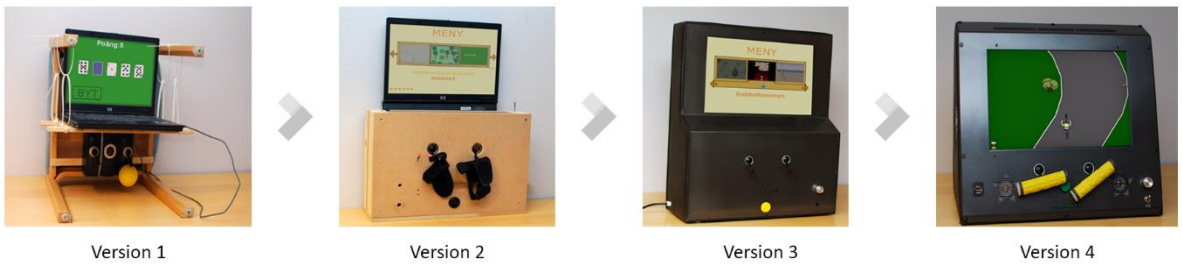

Figure 11. Various iterations from the Elinor project.

You might have an interesting gameplay idea which you want to actualize. This idea, in its infancy, might in fact be of that nature. But, along the way, you might realize that some of the gameplay

mechanics (which you first thought was interesting) are boring, or not properly working. An idea's future, however solid it might seem, lies within the boundaries of uncertainty beyond creativity, testing, and evaluation. This iterative process of nurturing an idea, gameplay mechanic, or a game's design to life is to makes sure that no stone is left unturned and to ensure a successful project.

Let say we want to create a 2D platform serious game specifically targeted towards preschoolers with the purpose of learning more about the cycle of nature. You would start with, (assuming you have completed the pre-production and prototyping phase) implementing the basic core mechanics, the bare minimum of a working game digitally. By playtesting this, you might now first realize (with regard to the target group) the importance of easy-to-learn controllers for instance. Because of this you have to revisit and refine the controllers to better suit the game and its intended audience. With this change, new challenges and opportunities emerge which also need to be addressed, solved, and ultimately evaluated. It is only when your core mechanics are implemented and working as intended that you add the remaining elements, such as sound effects, music, animations, and graphics, etc. When problems arise you deal with the situation and adopt a solution that coincide with your project and game's purpose. In the meantime, you should improve the game's core mechanics, implement new features, and refine the game progressively. The last and final step would be to add finishing touches and tying up loose ends in order to create a cohesive gaming experience.

All these intricate parts of game development are constantly changing and affecting each other on different levels. If some of the graphical elements changes, you might have to adjust some animation or modify the soundscape to keep the game and user experience coherent. Or, you might need to rework one of the gameplay mechanics, resulting in the need for new graphics and better programming. Because of all these unforeseen changes (that will happen, whether you want it or not) it is important to adopt this mindset of an iterative development process so that you can easily assess, discuss, and evaluate your decisions in time and not wasting important resources on "what could have been". Since the game is built progressively you can mitigate risks earlier but also manage changes in requirements more easily. Constantly improving and refining your game will lead to a more robust and thought out gaming experience. Why would you spend countless of hours starting at the end of the development if you were not sure about how the player would interact with the game? Not even a perfectly written game design document can predict the gaming experience. The only way to do so, is to play it.

5.3 Iterative evaluations

This iterative style of development requires an equally amount of evaluations. What is the point of an iterative development if you are not continuously evaluating your decisions? In the best of worlds your evaluation should include the client, end users, be in the right context, and from time to time the people involved in implementation and maintenance. In most cases, such extensive evaluation requires a lot of time, effort, and resources. However, despite these costs and time-consuming processes, it might be wise to include the client during the earlier stages of development to ensure that high quality, relevant, and correct content is added into the game and to push the development in the right direction.

There are many different sampling methods to choose from when you decide to evaluate your game. The most suitable method will depend on your purpose, budget, and timescale. Convenience sampling is (as the name implies) convenient, in that sense that it is quick and cheap. The researcher (or

developers in this case) asks basically anyone to participate in the experiment. However, since this method solely rely on availability rather than reliability it is difficult to know exactly how representative the sample is. In some cases its fine to use this method, but it is preferable to have some form of dimensional sampling in which certain characteristics can be taken into account, such as age, gender, and education, etc. so the evaluation can be assess in a better and hopefully more reliable and meaningful way. Evaluations can be done in multiple ways (focus groups, playtesting, interviews, or expert reviews) and is mainly influenced by the progress of the project. In the earlier stages, evaluations are usually done with paper prototypes, gradually working its way up to a digital version of the game.

The findings should be shared with both developers and clients either email, skype, or face-to-face conversation. Clients might be able to see usage and utility issues that the developers missed, or developers might be able to spot experience issues that the clients would not be able to get.

5.4 Development

There are different ways to create games, but they will always start out as an idea or a concept. The origin of an idea is difficult to pinpoint, but they are, in most cases, adaptations or modifications of previously existing game concepts. For the idea to turn into a game however, it need to be nurtured with different features, stories, settings, gameplay elements. Simultaneously as you think about the target audience, requirements, schedules, staff, and budget. This process, as well as the size of the development team is different from every company and project. A smaller project will obviously require less work. Which, subsequently result in fewer people. A project with a $20 million budget and a staff of 50 people will obviously look a lot different from a startup indie-company with five or six people without a budget. Each project however, requires a certain set of disciplines such as designers, artists, programmers, level designers, sound engineer, and testers. In smaller project, these very strict discipline “lines” get blurred and sound engineers might end up helping testers, or designers end up doing artwork. But in almost all cases, independent of project size and team disciplines; development cycles are used in an iterative manner where ideas, features, and gameplay mechanics, etc. gets tested, evaluated, and refined. Games without a proper development methodology are more likely to surpass their budget, miss deadlines, and deliver a subpar gaming experience. Three major development cycles are usually incorporated when developing serious games; Pre-production, Production, and Post-release/Post-production. One of which we have already discussed in the previous chapter.

5.4.1 The production phase

With a well-planned and carefully implemented pre-production, a solid foundation has been set for your project and it is time to start developing your game. But first, it is important to plan and schedule your work, and the first step to do so, is to accurately estimate the size of the project and the timeframe required to realize it.[31] You can do this by either looking at previously developed games with similar features as yours and the development time required to create it. Or, you can determine the scope and timeframe based on your own previous game development experience. An experienced member of your team can usually make these assumptions better and with higher accuracy than one who is not. However, previous experience will only get you so far since the nature of game development is innovation which most likely forces you to work on something that is completely new to you. This is especially true for serious games development since each game, project, and situation

[31] B. Bates, *Game Design,* 2nd edition, Boston: Cengage Learning PTR, 2004.

is unique. Whatever method you end up using, it is important to understand how the scope and size of your project ultimately affects the time to develop it. The scope of *Tic-Tac-Toe* is obviously a lot smaller than *Grand Theft Auto* and will require less time and effort to develop. But, despite the size of the scope, adding more features to your game will delay your release since you have to properly design, implement, debug, and balance them. By adding, what could be seen as a small feature to a game, can still affect the overall schedule of things. When working with serious game development make sure that both the client and the development team understand how much time it takes to actually create the game, and if new features were to be added, or changes were to be made, make sure you have the time to do so, otherwise, reschedule!

The schedule for development is obviously dependent on the scope and size of the project. Some projects will be finished and realized within as little as a month, while others can require several years to be completed. However, you never have an infinite amount of time to finish your game, so spend it wisely. Do not waste valuable days in the earlier stages just because you feel like you can afford it. Deadlines will sneak up on you faster than you expect.

5.4.1.1 Key features and priority list

Based on your pre-production and prototyping you can easily identify the key features of the game and establish the core gameplay mechanics which should correlate with your purpose, context, and target audience. Since you have properly prototyped your game beforehand, you know that it is fun and working as intended. If, by any chance you realize that the core of the game is boring, do not randomly add new features in hopes of fixing it. You will end up with a mess of unfinished and unbalanced gameplay mechanics that does not work well together. Instead, go back to the original source of the problem and fix it.

By understanding the size of the project and its importance, you can realistically determine what gameplay features you will have time to develop, implement, test, and balance, and which you cannot. Gameplay mechanics and features are usually prioritized after **must**, **should**, **could**, **won't**. Also referred to as the MoSCoW method.[32]

Must – Gameplay mechanics, features, and requirements labeled as *must* are critical to your project and its success. These are usually the core mechanics or key features of your game. If these requirements are not met before the end of the development, the project should be considered a failure. "The most vital things you cannot live without"

Should – Things labeled as *should* are important to your project, but not necessary. These requirements can be as important as the *must* but are often less time-critical and can be solved later on. "Things you consider as important, but not vital"

Could – Requirements labeled as *could* are desirable but not necessary for the project. They could improve the user and game experience overall but should only be implemented if time and resources permit. "The nice-to-haves"

[32] Clegg, D. and Barker, R. *Case Method Fast-Track: A Rad Approach*, Addison-Wesley Longman Publishing Co., Inc. 1994.

Won't – These are the project's least critical requirements, the serious game has little to gain from implementing these features at all. "Things that provide little to no value"

The reason for using this technique is to avoid "feature creeps"[33] and to help you establish the most critical and important game mechanics and features as soon as possible. You should of course try to deliver what you promised, but if your timetable looks threatened and deadlines are coming up fast, requirements from the *should* and *could* category are the first to be removed or altered. It is better to have a small and well-polished game, than a big game with a promising future with the feeling of "unfinished-ness".

By successfully prioritize and identifying the differences between *must*, *should*, and *could*, from the start of the project has enormous benefits:[34]

- By finishing the high priority tasks first, the team will be motivated to work efficiently to have time to implement features from the *could* category.
- You focus on what is important and does not waste time creating assets that might never get used.
- Knowing what might not end up in the game can help the team focus and do not get overly attached to certain features.
- The game will still feel finished even if only the *must* features are implemented.

This is not to say that everything you established in pre-production and your priority list is set in stone. Once things are implemented and tested, small changes and iterations will happen to further improve them. However, brand new features or directions should not really be considered past this point.

To show how time-consuming some features are to implement, it might be a good idea to involve the client at this time to jointly determine which key elements are most important to achieve the game's goal and educational purpose. There is a difference between idea creation and digital gameplay implementation. New ideas and gameplay features can be realized within minutes while re-creating them digitally requires a lot of hard work and effort. To transform something as abstract as an idea into something that can be truly experienced is a challenging and time-consuming process. Due to this phenomenon it is easy to get carried away with all the possible features you could implement into your game and become too ambitious with your project. You do not have to be innovative in every single aspect of the game. Sometimes it is better to rely on old proven methods and to borrow gameplay mechanics or game design that is known to be efficient and apply it to your own game and problem domain. Already existing similar serious and regular games are a great source of inspiration for your own game development.

In addition to your key features and core mechanics it might be suitable (depending on the purpose and goals of your game) to implement an "observer mode" where the client can observe and sometimes instruct the player to complete certain tasks.

[33] Feature creep essentially means the common phenomenon of new, sometimes not entirely necessary, features slowly creeping into a game during development.
[34] B. Bates, *Game Design,* 2nd edition, Boston: Cengage Learning PTR, 2004.

5.4.1.2 Work procedure

Once you have established the most important features for you project it is time to implement them digitally. There are many different ways of doing this, but a popular framework for managing game development is an "agile" working process called *Scrum*. Scrum is a process wherein the features and workload of a project are divided into different sprints (or, a rapid series of small iterations). The size and duration of these sprints are, as many things, project dependent; some projects will have weekly sprints, while larger and lengthier projects might be better served by having monthly sprints to avoid iteration fatigue.

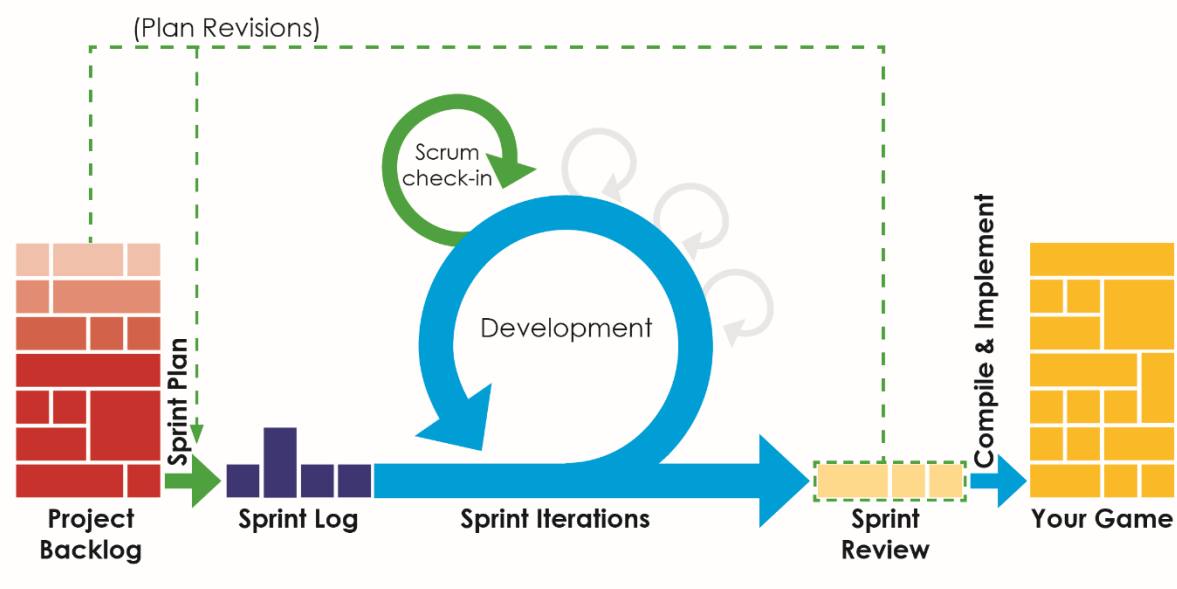

Figure 12. Flowchart illustrating the workflow of SCRUM.

In the beginning of each sprint the developers plan the work intended to be done during the sprint by selecting items from the priority list (in scrum, this is called a product or project backlog) and assess the work needed to finish them in time. At the end of each sprint the work is reviewed and reflected upon, answering questions like "What went well during this sprint?" and "What could be improved?" This review is then used as a guideline for the next sprint. It is recommended to have multiple scrum meetings (maximum of 15 minutes) during the sprint to make sure everything is running smoothly and to foresee and address any impediment that could prevent you from reaching your current sprint goal. This iterative development process makes sure that everyone in the team is working towards the same goal in an effective and efficient way.

Without a proper development methodology and work procedure, problems will occur. Especially if the development team's priorities are not aligned. Programmers, for instance might feel that a certain feature constitutes a fundamental building block for the game, giving it a high priority as it requires a lot of time to make it work properly. The same exact function, however, can be seen as unessential and not be very important to the designer. This unhealthy and unsynchronized work relationship can cause delays in the development. Game development require high levels of collaboration between diverse disciplines. A playable character for instance will only feel complete once the work from several multidisciplinary areas (design, programming, graphics, animation, etc.) are combined and brought together in a seamless way. Which means that everyone involved need to collaborate and focus on what is important and find the best way to realize it. Scrum facilitates this situation within a

multidisciplinary team and points to the importance of team-work and collaboration by synchronizing the disciplines and guide them in an efficient way towards a common goal. Furthermore, this multidisciplinary team might contain (depending on the size of the project) multiple sets of programmers, designers, or graphic artists. It is therefore important to divide these specialized groups into smaller pieces to streamline the production. There is no point (and frankly, quite irresponsible) having two, or three programmers sitting at the same computer, doing the same work. Instead, it is better to divide the work amongst each other in a suitable way (preferably based on experience). E.g., in a development team of three programmers one might work on the enemy AI, while the other two work on different parts of the game's engine. This allocation of work is true for each discipline participating in the development.

Even though this is an iterative process, game development still contains some kind of linearity between the different disciplines as it is difficult to animate something that does not have graphics, or to add sound effects to something that does not exist in any other form than a brief description. During this phase however, it is important to not sit around and wait for things to happen, you can still create concept art, playtest different gameplay mechanics, or create sound effects from relatable work and previous experiences. During a serious game development project, there will always be work to be done. If you experience a low workload, you have to actively seek out things to do to further the progression of the project.

5.4.1.3 Milestones

During game development you must be able to track your progression in a good and meaningful way to make sure everything is on schedule. As with many other things in game development, there is no universal truth to how milestones should be established or planned for, as they are very project dependent. Milestones should more be considered to be general guidelines for what type of deliverables the project should have produced within certain timeframes, as well as various major events during game development and are commonly used to track progression. By breaking down the process into milestone segments, motivation and end user engagement are easier to maintain as it's easier to describe to them where the project is currently at in the development timeframe, and when they should expect the next notable release and which new features they might expect by then. Milestones can also have a positive effect on the project team itself; reaching a milestone is a big accomplishment, and having something to celebrate during the development process can be a good way to maintain team morale. Lastly, milestones help make a project seem more manageable for developers: it is almost always better to work towards multiple smaller goals than to just have one giant goal in the far future to work towards. A game development project usually consist of four primary milestones: **First playable**, **Alpha**, **Beta**, and **Gold**.

First playable: This stage marks the first playable version of the game with the game's core gameplay elements being functional. The first playable version of the game is often highly dependent on the work the team did on the prototype during the pre-production phase. At this stage you will either see a fun and cohesive core game with the need for some additional features and some polish, or you will notice that you have a gameplay experience with features and mechanics that fit conspicuously poorly together.[35]

[35] E. Bethke, *Game Development and Production,* Plano, Texas: Wordware Publishing, 2003.

Alpha: When the game is playable from start to finish and contains all the game's major features (the *must* and *should* features we described in 5.4.1.1), the game has entered its alpha stage. At the alpha milestone, the game will still contain bugs, ad-hoc workarounds, and some missing assets and placeholder materials. From this point onward, the project team's focus should shift from building the core structure of your game, to finishing it. Gameplay mechanics and features can also be removed or revised based on feedback from various playtests. But as a general rule of thumb, no new major features should be introduced after this point in the development stage. This is simply to make sure that the game will actually be finished within the project's time constraints –

Do not underestimate the importance of feature stops

It can sometimes be difficult to declare a strict feature stop when you feel like you're only half-way or two thirds into your project. In many cases, it can feel like you're just now starting to see new potentials of your game and you start getting new ideas for how it can be improved to become even better. We do, however, *guarantee* that a feature stop will improve your project infinitely more than continuing to add new features to it. Polishing a game to a stage where it feels cohesive and well executed is an incredibly time consuming process. This is why it's immensely important to understand what the core gameplay and goal of your project is. Knowing the core gameplay helps you identify which features are essential, and also helps you realize which features you can cut out without harming your core ambitions.

In the end, end-users will likely appreciate getting a well-executed product more than they might miss that extra small feature you wanted to put in (that the users like won't even know is missing).

Beta: Once the game reaches beta, a complete version of the game should be available. The goal of this stage is to stabilize the project. This means, that all development stops and no more changes to the game are being made. The development team focus solely on eliminating as many bugs as possible and to increase the user experience by increasing the game's usability and stability before shipping. The occasional artwork, text, or sound effect can still be updated if necessary. The game usually suffer from speed or performance issues as well as game crashes and data losses.

Gold: Gold is the final stage of a game, it is a complete user and gaming experience with every feature fully functional. The game is running smoothly without any major performance issues and game breaking bugs. Since you cannot test your game on every single combination of system, hardware, and software out there, this stage is usually followed by patches to increase the usability and the game's stability.

One or two weeks before each milestone, a "code-freeze" is introduced and nothing more is added to the game and the development team enters *crunch mode*, where everyone focuses on finishing the game suitable for the milestone.

5.4.2 Risk management

An iterative approach to game development is not without any risks. Despite meticulous planning and research, things can still go wrong. During software development projects, some problems are more

common than others. Due to this regularity, they are usually referred to as classical mistakes.[36] The following list highlights some of the most important mistakes connected to serious games development and how to avoid them:

5.4.2.1 People related problems

The biggest problems when creating serious games are often tied to basic misunderstandings. The development of serious games require both game development knowledge and domain competence. Both parties need to have a basic understanding of each other's subject matter. This is rarely the case in the beginning of any serious games development. Because of this, friction between developers and customers will arise during your time creating serious games. To avoid misunderstandings and unrealistic expectations it is important that both partners are open-minded and have an understanding of what is doable given the time-frame and resources available. Continuously communication and cooperation between (but also within) developers and clients are essential for a successful serious games project. It is important to discuss the issues and solutions together and take both perspective into account. Poor communication complicates the work and causes uncertainty between the development team and the client.

Another common problem is wishful thinking. Closing your eyes and hoping it will work out in the end is not the right way to develop games (or anything for that matter). Wishful thinking and undermining your plans is not recommended and will eventually backfire. This problem can be avoided by creating a realistic and solid plan.

User-testing, field studies, and feedback is all necessary for a successful serious game project, it is difficult to assess anything at all if you do not understand how the client's organization operates and how your target audience react to your game. The lack of user input can result in time consuming work in the later stages of the project.

5.4.2.2 Process related problems

The scope and size of the project usually determines the scheduling. There is a difference between scheduling for a 3 month project and a one year project. It is however, in both cases detrimental to schedule and plan in an optimistic manner and not to underestimate the time required to realize the game. Working against a short, and unattainable deadline can cause stress which affect morale and productivity within the team. If deadlines has been set beforehand it is important to design the game in such a way that it will be finished in time for the release and not become too ambitious with features and such. It is tricky to avoid this mistake of optimistic scheduling and it usually comes down to previous experience. However, to be on the safe side of things, it is always better to focus on a smaller, well-polished game first. Expending the scope by adding more features should only be allowed if the allotted time and resources permit.

Another problem connected to the process of game development is disorganized work, misdirected focus, and an eagerness to start coding. Most parts included in game development need to be done properly. To create a sloppy design as an excuse to start programming and digitally implementing the game is unacceptable. Unfinished and sloppy work will eventually be thrown out because of unfulfilled requirements. This means that valuable time, resources, and energy has been spent on something that

[36] S. McConnell, *Rapid Development,* Redmond, Washington: Microsoft Press, 1996.

will not make it into the final game or has to be properly reworked. Starting in the right “end” of development is of utmost importance.

Do not underestimate the value of playtesting. It might be time-consuming at the moment. However, most of the game breaking problems will be found and corrected before they can cause any serious harm or delays to your production. Without proper testing, it is likely that your game end up buggy with major performance issues and cannot be released. Skipping the whole testing phase will, without a doubt increase the workload down the road.

5.4.2.3 Product related problems

In most cases, inexperienced game developers aim too high by including way too many unnecessary features and requirements. Sometimes new, cool, and unplanned features are suggested and experimented with way too late into the development process. Added functionality and features increases the game’s complexity, which in turn requires more time to develop, implement, test, evaluate, balance, and debug. It is important to identify your game’s key features and prioritizes them accordingly. In addition to this, is it always better to have a solid, functional core, instead of a big mess of unfinished and unbalanced features. If something were to be added, it is important to understand how that will affect your schedule and what changes need to be made in order to successfully implement them in time. Depending on your target audience, it is almost always better to utilize an inclusive design strategy and to design the game for casual players. This usually goes hand in hand with the complexity of the game. Hardcore gamers will find simple games boring, while people who rarely play anything can find them challenging and entertaining. Finding a middle ground is recommended.

Another problem is that a lot of developers get too attached to their ideas and gameplay mechanics, especially in the earlier stages of development. Your first idea will rarely be the best. Reworking ideas and iterate on gameplay mechanics are an important and necessary part of serious game development. Banging your head against the wall, wasting valuable time and resources trying to keep certain aspects of an idea is not always the best and most appropriate response. Some ideas will not work out.

5.4.2.4 Technology related problems

As mentioned before, switching development tools in the middle of a project is unwise. The learning curve, rework, and the supplied mistakes made with the new tool usually cancels out any benefits. It is better to stick with what you know and assess the strengths and weakness of your development team and choose the development tools and platform that suits your needs. E.g. If your team lack programmers, Game maker might be an option, or if you plan on making a full-fledged adventure game, Unity may be preferable. But remember, the development tool is merely an instrument for realizing the idea.

The game must be easy to operate and maintain once it has been implemented into its right context. The project will usually be supervised by someone with less game and technical experience than the developers. Because of this, establishing a support plan to facilitate continued use and achieve low maintenance requirements is important. Without proper usability and support, the game will not be very useful and it will not be able to provide good educational value nor a good gaming experience.

5.4.3 Playtesting

There are many different ways one could playtest games, and most of the time it's highly connected to the nature of the game and its purpose and practical considerations.[37] Continuously playtesting in symbiosis with development is done to make sure everything is working correctly and that the serious games fulfills its purpose by gathering feedback and identifying potential design flaws and hardware problems. If you never test your game before shipping, you will be in trouble. To playtest using your own computers and tech might be convenient (and in the early stages of playtesting, perfectly fine) but not testing and thinking about how the game is used in its own context can be catastrophic. Establishing a testing environment that represents the final use context is good practice. This could be done by borrowing computers and tech from the application area in which the solution will be used. You could also set up a remote station that emulate their working procedures and programs. Playtesting is always recommended once the game reaches one of the previously discussed milestones.

Early in the development process, quick informal playtesting is recommended. This can be done with people from the development team or people unrelated to the project. There is usually no need to involve the end users at this moment. To organize a play session with the end users require planning both internally with the development team but also externally with the client. This process is very time consuming and takes a lot of effort and resources to realize. To do this on an unfinished game that will probably change a couple of times before its finished is a waste of time and valuable resources. Depending on the size of the project, testing can be done by a special testing team or members of the development team. Despite who is conducting the testing, it is important that the results are evaluated and communicated with the rest of the team (but also with the client) so features can be refined, re-worked, revised, or scrapped.

Before shipping, the game should be tested on the right target audience, in its right user context. This is to guarantee that the serious game fulfill its promises but also to have the chance of correcting any problems that might occur thanks to the new untested territory. This final playtest should be seen as the most important one and should under no circumstances be skipped. The results for this playtest is usually more black and white compared to the other testing occasions. This time it is much easier to understand what is working and what is not. Testing under right conditions can induce unforeseen technological or administrational problems such as firewalls preventing system communication or compatibility issues between different operating system and computers. Therefore, it is important to conduct the final playtest while you still have time to fix the mistakes and correct the problems that might occur.

The most common way of obtaining data from a playtest is to use surveys, conduct interviews, observations, or record players' interactions with the game. It is important to take into account where you are in the development process and how many people will participate in the test. Surveys, for instance, might be more suitable for a lot of people in the later stages, while interviews can be more suitable for fewer people in the earlier stages. The importance of playtesting in the right environment with the right people increases as the development process progresses.

When it comes to serious games and gamification, playtesting and evaluation is not only about the gameplay, usability and technical performance (more details about this in chapter 6). You also have

[37] M. Eladhari and E. Ollila, 'Design for Research Results: Experimental Prototyping and Play Testing', *Simulation & Gaming,* vol. 43, no. 3, 2012, pp. 391-412.

the purpose and the utility of the game to worry about. If you have developed a game to teach mathematics you will have to show that is actually does. This is another type of testing than traditional playtesting in game development. In fact, this may be one of the unique competencies of a serious games developer as compared to entertainment game design. Remember (as discussed in chapter 2) that a serious game has to fulfil the dual, and possibly conflicting, requirements of being both a good game and a useful solution to an organizational need. We advocate knowledge in scientific methods (experiments as well as interview based methods) to do this type of testing. As an example, when you develop a math game you may have several goals in mind: it should work as an entertaining game; it should raise motivation for learning math; it should be in line with the curriculum; it should improve test results and so on. What aspects that are tested has to do with the goals that you have negotiated with your client. And it is a very important part of a project to show that you can actually deliver on what you sold. Our experience is that clients worry more about the utility of the game than about the quality of the gameplay.

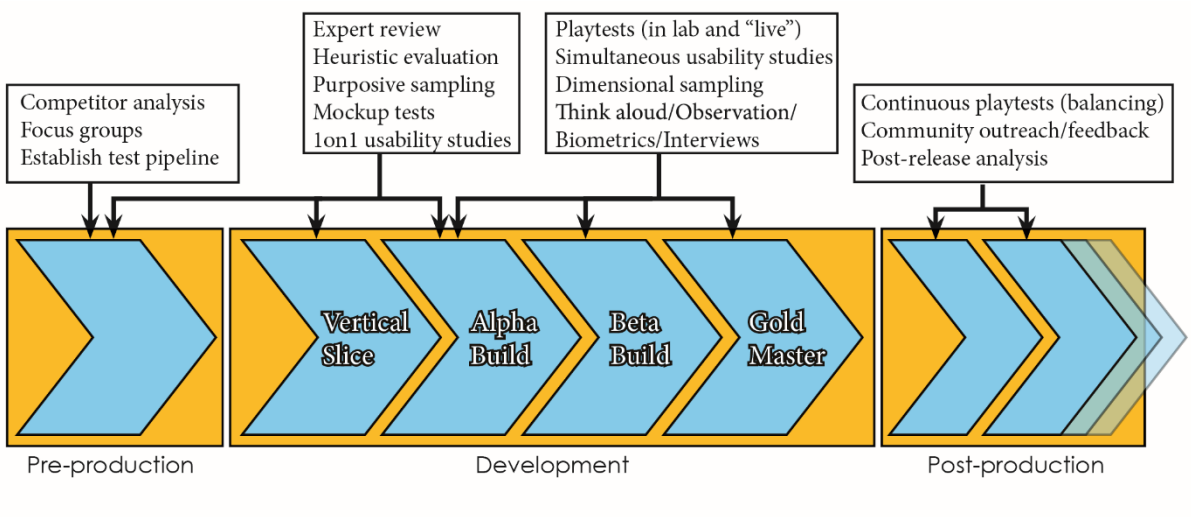

Figure 13. Timeline illustrating when, how and with whom to playtest with during certain points of development

5.5 Participatory design

Participatory design is an approach to design attempting to actively involve all stakeholders in the design process to help ensure the result meets their needs and is usable.[38] In serious games development this cannot be understated. Client inclusion is mainly necessary for a successful serious games project. They are the ones who are going to be using the end product and if they are not involved in the development, you risk creating a game no one wants. The client have, in almost all cases, a better understanding about the subject matter, context, and target audience which can all facilitate with a better and more suitable game design. Participatory design introduces a new, and valuable perspective into the development. This can be used to easier establish what the game should represent but most importantly, how it should be used. In an educational setting this could mean, for instance, that a teacher wants to introduce the core of a subject during regular lectures and other classroom activities and then use the game as an environment where students put their knowledge to the test in interesting ways. The teacher, in this case would probably have a more detailed understanding about the target demographic, namely, the students and could, compared to the

[38] Khaled, R. and Vasalou, A. "Bridging serious games and participatory design", *International Journal of Child-Computer Interaction*, Vol. 2, No. 2, pp 93-100. 2014.

developers better define the game's purpose and how it should be used. Without this extra information it can be quite difficult to develop the game where you are unsure about how the game is meant to be used. With participatory design this can all be avoided or at the very least controllable.

Participatory design with the end users, compared to the client or commissioner is a little bit trickier since the serious game design process is complicated and requires knowledge about either (but preferably both) the application area or game development. Something, which the end users might be lacking. If the end users are, however included into the development it is important to set up project boundaries to assist them in establishing specific and relevant ideas, otherwise they might focus on unnecessary features.[39]

Workshops, observation, and artifact analysis are all methods which can be used for participatory design. Regardless of method, three basic stages are almost always present during participatory design:[40]

1. Initial exploration of work – In this stage, both parties familiarize themselves with work-relatable ways. Technologies used, workflow, routines, and work procedures are all explored and discussed.
2. Discovery processes – This stage helps clarify the goals and values and to agree on a desired outcome.
3. Prototyping – In this stage, both parties iteratively shape the project established in the previous stage.

Participatory design at this stage of the project (development) should not be seen as something revolutionary and changes made to the game should only be minimal. The enactment of collaborative development, will obviously differ from every project, but it's important to understand the value of cooperation since it contributes with relevant and helpful information which guides the project forward.

5.6 Managing end user expectations and acceptance

Depending on the situation you are developing for, starting to manage user expectations early on can be crucial for the success of your project. If you are working with organizational settings, it is important that the end users will actually want to use your game. In Information Systems research, the challenge of real-world uptake of produced solutions in use contexts and among end users is often referred to as the issue of "technology acceptance"[41], and it is highly applicable to the implementation of use of serious games. Shortly summarized, technology acceptance depends on a wide variety of different factors: there is a certain 'performance expectancy' among end users that influence what they expect your serious game to do; they also have an 'effort expectancy' and evaluation of how much extra effort your serious game is going to require from them (e.g. learning a game interface, or learning how to use a new piece of technology); and there are also various 'social influences' and 'facilitating

[39] R. Khaled and A. Vasalou, ' Bridging serious games and participatory design', *International Journal of Child-Computer Interaction,* vol. 2, no. 2, 2014, pp. 93-100.
[40] C. Spinuzzi, ' The Methodology of Participatory Design', *Technical Communication,* vol. 52, no. 2, 2005, pp. 163-174.
[41] Davis, F.D. (1989) "Perceived usefulness, perceived ease of use, and user acceptance of information technology", MIS Q., Vol. 13, No. 3, pp 319-340.

conditions' in the end users' organizations or play contexts that inform their willingness or ability to work with serious games.

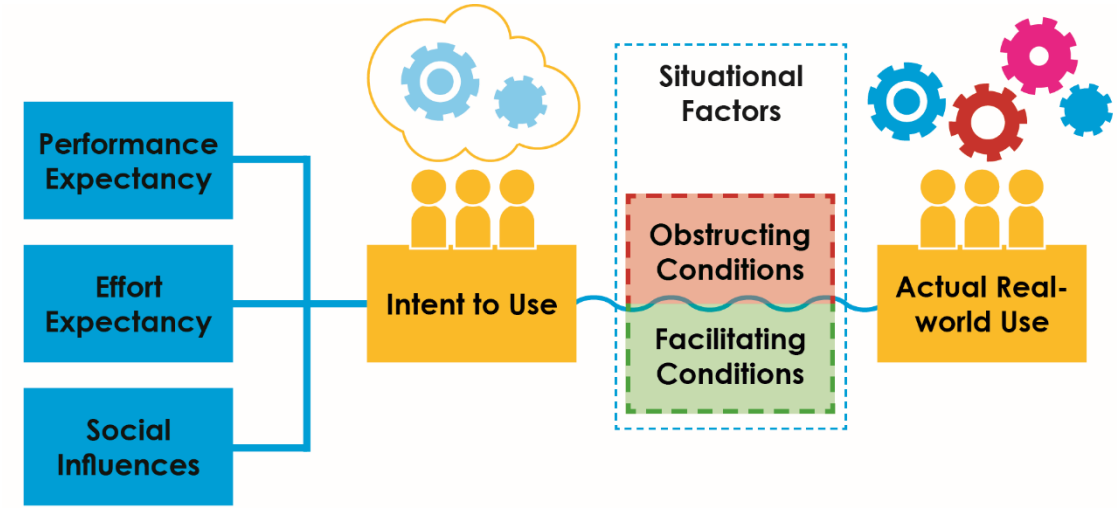

Figure 14 A simplified version of the unified theory of acceptance and use of technology (UTAUT) model[42].

Technology acceptance is an entire science in and of itself, and there is a myriad of small nuances to every single step when it comes to facilitating positive and active end user engagement with software solutions. But, in broad strokes, technology acceptance comes down to managing or accommodating for end users' expectations, understanding the culture of their contexts (whether it's an organizational culture in a business environment or school, or if it's the social culture in their homes or cities), and the conditions in the use context which might facilitate or obstruct end users' interactions with your serious game.

Technology acceptance is crucial when it comes to successfully implementing and integrating a developed serious game solution into any sort of context: if the people who are supposed to be using your serious game don't want to receive it, the serious game project is a failure. Knowing about the general aspects of end users' acceptance is of course a prerequisite to being able to ensure that a project team works in a way that takes them into account in a good way. And even though end user acceptance is often a problem that is most noticeable at the end of a project - many serious game developers learn it the hard way – it is never too early to start thinking about it and incorporating end user-oriented working methods in your development process.

Performance expectancy is the individual end user's expectations of what type of values the serious game is going to bring to their work processes and contexts. Essentially, this is where the user either thinks "I can't wait to see how great this serious game is going to be, I bet it's going to be the new World of Warcraft!" or "I really don't have high hopes for this game they're bringing in, it looks pretty mediocre…" Everyone always has some sort of expectations on things they are going to be interacting with, and it is important that these expectations are managed well. It is important to realize that the hyperbolic first example of expectations can be just as damaging as the more pessimistic second one.

[42] Venkatesh, V., Morris, M.G., Davis, G.B. and Davis, F.D. (2003) "User acceptance of information technology: toward a unified view", MIS Q., Vol. 27, No. 3, pp 425-478.

When your serious game fails to live up to over-hyped expectations, the end users might quickly turn against it.

Effort expectancy is the end user's expectations of how the new serious game is practically going to impact their working situation. Essentially, it is the end users' impression of whether this new serious game is going to be a huge extra burden to them since they might have to learn a brand new game system and work with all manners of complicated technical devices. Effort expectancy is best met with communication, and by the project team being very explicit about their understanding of pre-existing organizational working processes and individual end users' situations.

Social influences are unavoidable when working with serious games, and end users often have some sort of social influences surrounding them that informs the way they think about serious games. Some organizations might have a social culture where everyone is eager to try out new things, or where individuals will jump at the opportunity to be able to pioneer a new solution. In other organizations, an opposite culture might be present where people are more careful when approaching new solutions. Maybe they get new experimental solutions pushed upon them too frequently, and are experiencing some fatigue because of it. Understanding the social influences in the among end users in the use context is important, and there is no universal solution for how to best foster a positive social atmosphere around your serious game as every organization's culture is unique. The best thing to do is to take a thorough inventory of what the organization is like, what previous solutions they've already tried using, and in which ways previous interventions might have fallen short of the end users' expectations.

Intent to use is a combination of all these expectations of influences. End users' individual and shared experiences that are present in their social environment, as well as their expectations of performance and required effort, dictates how your end users start forming their intentions and plans of how they'll be working with your game. These intentions might be outspoken, or they might be subconscious; end users might have started forming clear plans for what they'll do the first day they get their serious game, or they might have some un-specified gut feeling about it. Intent to use can be either positive or negative, some users might have already decided that they don't want to use your serious game based on their expectations and the feelings of their coworkers, or they might be extremely motivated and eager to start. Capturing these feelings is important, and can often be done rather early in a project through testing early versions of the serious game together with end users or, as previously mentioned, inviting them to participate in the design process. Ensuring that your serious game creates good intent is crucial, as it increases the chances that your serious game can actually see positive real-world use.

The intent of use, regardless if it's positive or negative, is always ultimately filtered through a series of **Situational Factors**, or **Facilitating and Obstructing Conditions**. There always comes a time when intentions and planning needs to face the brutality of real-world practicalities. Computers can crash, operative systems can need mandatory updates that breeds compatibility issues with your serious game, internet service providers might have untimely interruptions of service, and so forth. If end users feel confident that your project team is aware of these practicalities, and have contingency plans for when they happen, it can be hugely beneficial for their view and acceptance of your serious game. This is where many serious games fail - project team that create serious games that they themselves think are amazing in terms of content and gameplay experience often fall into disuse or distrust quickly when

they turn out to be unprepared for obstacles caused by the practical situational factors that the end users need to deal with.

Real-world Use is the end result of all these processes of expectations, socializing and experience sharing, deliberation, planning, and first encounters with practical reality. This is where, if a project team has been able to manage and accommodate for all the previous steps well, the serious game is able to have a positive impact in its target use context. This means that the serious game lives up to end users' expectations, has a solid anchoring in the social culture surrounding the use context, and manages to anticipate and work well with the practical challenges the end users' face.

Our game is great, the end users just don't *get it*!

It is of crucial importance to understand that end users are not to blame if they find your serious game insufficiently useful and don't want to use it. Having created a product doesn't automatically entitle anyone to a positive and receptive audience. Historically, a common rhetoric among serious game developers and pundits has been to claim that a perceived "stubbornness" or "short-sightedness" of clients or end users is the main reason for serious games not managing to have a stronger impact in various sectors. This is not a rhetoric you should subscribe to if you want to enact positive change through serious games – if a product fails to deliver, it is the project team's fault and not the fault of the target end users.

Sometimes the factors surrounding a project's failure can feel unfair, and sometimes they might feel like they're outside of the project team's control, but ultimately that does not matter. Ultimately, the serious game either didn't accommodate for, or work together with, real-world conditions well enough to be useful, or there wasn't enough work done on end user buy-in for them to want to make the serious game work in spite of unfavorable real-world conditions.

Blaming the end users is extraordinarily easy, perhaps even cathartic, and it is also an excellent way for a project team to save face. But, it is also fundamentally unfair. It is analogous to a chef not bothering to take dietary restrictions, allergies, or food hygiene into account when preparing food at their restaurant, and then blaming their customers for not wanting to eat, or for making a scene when they leave the restaurant in an ambulance.

One of the best ways to encourage end user technology acceptance, is to use participatory design processes that include end users and to have frequent play tests that involve end users. As previously mentioned, it is also useful for developers to be able to see and experience the end users' situation first-hand to whatever extent it is possible. If the serious game is developed for physical rehabilitation, for example, the developers would be well served by both having in-depth interviews with end users as well as trying to use the products that the end users normally use to directly see how they might be improved upon and in what ways they already work well. When working with education, seeing the physical setup of a classroom, the computers they have available, as well as the systems and processes the teachers are already adept and comfortable with using, will make developers better at creating something that meets their needs and works in a way that teachers find to actually be useful.

6 Implementing, Using, and Evaluating Serious Games

This chapter will focus on the implementation, use, and evaluation of serious games. This is also the last, and final step of our journey in serious games development. A good example to highlight the fundamental parts of this definitive process would be our collaboration with Högskolan i Borås and Västra Götalandsregionen and our contribution to the project SAREK, where our gaming experience, knowledge of organizational introduction, and evaluation models has been put to the test. SAREK is, as previously mentioned a flexible training environment for prehospital care. The purpose and vision of the project is to make it possible to role play realistic and extensive scenarios covering all phases and aspects regarding prehospital care work, ranging from the call out and all the way to handing over the patient to the hospital. The system of enactment of patient care in live role-playing scenarios and simulated prehospital environment is physically installed and currently in use to facilitate training and certification at a regional ambulance center in Skövde, Sweden. We have been a part of, and studied this project since 2014. We have also been working with another serious games installment for a school in a municipality in Sweden, where we used a modified version of *Minecraft*, called MinecraftEdu[43]. The modified version was developed to make the original game more accessible in an educational setting by adding server management tools, easier installment, and "teacher commands" to facilitate, setup, and manage various exercises. The process of setting up this formal game-based learning environment required preparation of three primary resources:

- **Reliable technical infrastructures** – By acquiring and prepare various hardware and software, setting up reliable systems, and taking inventory and reviewing the school's resources.
- **Organizational work structures and praxis** – Examine the curriculum goals and make sure the game reflects them in a good and educational way and also adapt the gaming experience to scheduling constraints and available resources.
- **Human factors** – Establishing a dialogue between teachers, students, and stakeholders to facilitate the implementation but also to survey their attitudes towards, and experiences with educational games.

Both these projects have given us valuable information and experiences of what to do once the digital solution has been implemented and helped us establish important guidelines regarding the last step of finalizing your project and successfully maintaining it.

6.1 Formal and informal implementations

As we mentioned in chapter 2.4, there is a fundamental difference between developing and using games for formal contexts as opposed to doing it for informal contexts.

The most central, and biggest difference between games used in formal and informal settings is the circumstances in which players participate. Formal settings are, by nature, structured, compulsory, and have a stronger adherence to time and space restrictions during play sessions. Games under these controlled forms are usually administered by a supervisor or instructor who must comply with specific requirements regarding learning objectives and allotted time. One common example of this could be a teacher that leads classroom play sessions: in these types of situations, the teacher needs to ensure that curriculum goals and learning requirements for the class are achieved, but s/he also supervises the play sessions to ensure that the technology is working well and helps students that might struggle

[43] A modification of Minecraft created by TeacherGaming LLC

with the content or game interfaces. In formal settings and applications, the audience are captive and participation is mandatory compared to the "opt-in" situation usually found in informal settings. This significantly alters the requirements that are placed on the games, and it introduces requirements on what they *should* do, but also constraints on what they are *allowed* to do. For example, a game in a formal setting should take great care to represent subject matters accurately and authentically, but primarily to provide in-roads for instructors or supervisors to contextualize play activities and anchor them in broader curriculums. But games in formal settings need to ensure that the game is accessible and appropriate for everyone in its target audience - since we are now dealing with a captive audience, it is essential that members of this audience are comfortable and can participate as equally as possible in the product we have created. To, once again, use educational games as an example: a game that even just a few of the kids in a classroom are not able to play or feel comfortable with, be it due to skill requirements or to inappropriate content, a teacher cannot in good conscience use it for classroom sessions.

Another defining trait of formal use contexts for games, is that they often have pre-existing and well established organizational structures that the game needs to be able to fit into and accommodate for. This can problematize the implementation process due to added requirements put on the game and specific and loosely defined organizational cultures and working processes that are hard to foresee and develop for. But, it can also be complicated for the organization itself as games can introduce requirements that organizations need to adapt to and fulfil. When games are used in educational environments such as schools, processes might need to be put in place to increase game and technology literacy amongst teachers, adaptation to the syllabus, expand IT-departments' working duties, and to update or replace obsolete hardware to make it more reliably compatible with these new game-based teaching tools. To successfully implement and maintain the game in a formal context, these accompanying requirements (which rarely exist in informal contexts) must be met. Advergames (which could be argued to be more informal than games used in schools), for instance, require less (but equally important) attention to the application area, target audience, and context compared to games for educational purposes. This difference regarding the amount of influential factors and requirements influence, and usually complicate, the implementation of the serious game. A simple thing as installing the game on multiple computers can be proven to be an intricate and time consuming process, where end users often encounter many technological obstacles. These types of issues with practicality and requirements of use is something we often encounter in our own research, and one piece of general advice we have for developers and customers of serious games and gamification is: "When developing or attempting to use an educational game, the strengths and shortcomings of the recipient organization need to be used as a starting point. The organizational structures, culture, working processes, and technological infrastructure need to be thoroughly inventoried before the integration of an educational game can be initiated."[44]

6.2 Processes and challenges of implementation

6.2.1 Different organizational settings

As previously discussed in chapter 2, almost every organization and application area in which a gamified solution or a serious game can been implemented and used is uniquely different from each

[44] B. Berg Marklund, *Unpacking Digital Game-Based Learning: The complexities of developing and using educational games,* Doctoral Thesis, University of Skövde, Sweden, 2015. p, 197

other. A serious game that has been proven efficient and effective in healthcare might not work at all in schools or educational environments. This is because, in most cases these areas are so unique and different from each other. Both in the way that they operate and run their business but also to whom they are catering to. Which only make sense. Why would a game, developed for kindergarten be useful in a military setting? The same can be said for the context in which the game operates. E.g. our project with Karlsborgs Tourism required a novelty solution for it to feel natural because of the unique environment. Sure, one could have implemented an ordinary game with regular controllers. However, the experience and the outcome of the game would definitely not have been the same. We would argue that the "wow-factor" of games is sometimes enough to instill learning and to increase motivation, especially in public places, such as museums and art installations. Therefore, the context is equally important as the game itself. Furthermore, just as organizations are different, so are the people within them, and games can be stimulating in many different ways which often only address a certain type of players. Some people might prefer simple puzzle games while others only play adventure games with well written narratives. All these various concepts are unique to each organization, context, and discipline. Making it an important part of serious game design and development and should be treated with care and consideration.

Not only do the different application areas differ from one another in unique ways, there are often differences within different use contexts of application areas. For example, two hospitals might differ greatly from one another when it comes to working practices and organizational cultures, even though they are both within the same area of application (healthcare). It is for this reason we often repeat ourselves when talking with developers or people who want to start using games in their organizations: make sure you understand what makes a specific use context (be it a hospital, a classroom, or a corporation) unique, as it is difficult for a game to provide a general solution that is directly applicable to a broad range of use settings.

In short, each application area and use context comes with different prerequisites, serious games developed and used for educational purposes such as teaching tools in classroom environments requires not only an understanding of context and target audience but also the significance of the curriculum for it to be useful and beneficial. What is included in the curriculum is mandatory and its content is problematic to change. Teachers, in particular need to relate to this if games were to be introduced into the classroom. There is also a different need between teachers and students. Including tools which can assist the teacher in different tasks throughout the training cycle, mainly to assess and control the gaming session by keeping track of student progression is almost always necessary. In addition to this, schools operate on a tightly strict budget, which is most likely limited when it comes to alternative learning methods such as games. It is therefore important as a developer to understand these principles and respect for the customer's prerequisites by pricing and developing your game to meet these requirements.

6.2.2 Putting the solution to use

As mentioned before, serious games should assist, not replace whomever planned on using them. Putting the solution to use however, might be easier said than done, and as we pointed out in chapter 4, one of the most important and crucial part of serious games development is to: understand the **context of use**, and to understand your **target end users**.

6.2.2.1 Context of use

Context of use are, as we also insist on repeating, incredibly varied and they can be very unpredictable if developers and clients do not co-operate to create a thorough inventory of the different situational factors, organizational processes, and cultures in place at the context of use. When implementing a game-based solution in a formal use context, situational factors such as technological infrastructure (e.g., the availability and performance of computers and other devices and hardware), organizational support structures (e.g., IT-department's work processes and availability and training in managing new software), and pre-existing working processes all need to be accommodated for. Informal use contexts are, of course, slightly different, but they can pose their own challenges as they might be harder to fully understand and are, to a further extent, out of the developer's control. When creating an advergame, or an educational game that is to be encountered or used at home, it is important to understand how and when people will interact with the game.

Informal use contexts across cultures

In an invited lecture given at Sweden Game Conference 2016, Shailesh Prabhu, an independent game developer working in India, spoke about the mismatches between informal use contexts of mobile games in India as opposed to western countries. According to Shailesh, many of the use contexts of mobile games that people are accustomed to in the west, and which western game developers often create games for, do not really exist in the same way in India. For example, some of the more common use context of mobile games in the west are: when riding the bus or public transit, or when visiting the bathroom. Shailesh mentioned that public transit in India is significantly more crowded than in many western cities, and that there's often simply not enough space to hold one's hands in a "gaming" position, or maybe uneven terrain which makes the ride more bumpy which makes touch interfaces harder to interact with, both of which makes the "public transit" use context less prevalent. For the bathroom use context, a large portion of the population and in many places in India the bathrooms do not have seats - which has a very obvious impact on the ability to lazily play mobile games.

These examples might be a little bit off kilter or out of the ordinary, and are probably not something most people reading this report will have to deal with, but it highlights how simple misunderstandings or a lack of knowledge about simple practicalities of use contexts can significantly change the usefulness of a game or have an impact on a game's marketability. Not understanding the technologies available at a hospital or school, or not understanding how and when a target audience might interact with your game-based solution, can both render an otherwise excellent product useless or unwanted.

6.2.2.2 Target end users

As mentioned previously, end users can also be a very complicated thing to handle, and it is very important not to underestimate how incredibly varied most peoples' technological abilities and gaming literacies are. Never underestimate the heterogeneity of your end user group; even in classrooms, where students are often thought of as "digital natives" with a similar proficiency and interest in technology and games, the division between user's game literacy is often enormous. There will always be some people who are more comfortable than others using new technology thanks to previous experiences or how tech-oriented their households are, and compounding that people will often have

widely different preferences or requirements of what they want and expect from a game. This exact phenomenon appeared, and was observed several times, during our serious games project with the previously discussed MinecraftEdu. Where the range of the students' gaming literacy, gaming preferences, subject matter knowledge, motivations to play and learn, general interests and proficiencies were vast. Some students had played the game before, some even had their own YouTube channel where they showed off their play sessions and private dedicated servers they paid for to play with friends, while other students struggled to even start the game or do rudimentary things in the game. A thing that might seem intuitive and simple to some, such as moving your in-game character around or looking around in the game world, can be quite challenging for someone who has never played a game before. Regardless of application area and context, this phenomenon will most likely always recur. This is something that developers in particular can have a hard time managing. As developers themselves are often proficient with technology and games to the point where they understand and grasp most game interfaces almost intuitively, and when you are that proficient it is almost hard to imagine a mindset and skill-level where something as basic as "walking" is difficult to understand. Accommodating for the heterogeneity of the target audience is of course something that needs to be pre-planned, designed, and executed in the early stages of development, otherwise you will have a game that only a few people can play, experience and learn from. It is challenging, but important to find a balance in your game design were both novice and proficient players are equally included in order for the game to be considered useful.

There are usually many different people involved (developers, end users, SMEs, and stakeholders) in a serious game project with different expectations and benefits, both during and after development but also throughout the play sessions. The expectations and roles between novice and proficient players are usually vastly different. Proficient players can sometimes feel constrained and limited compared to what they are used to and that the serious game fails to measure up to their expectations because of their previous gaming experiences whereas novice players tend to have lower expectations due to the fact that they do not really play games that often. This will often reflect upon the player's behavior were proficient players tend to get more easily distracted and therefore devote their attention to non-assignment relevant gaming and focus less on the subject matter, learning objectives, and the task at hand.

Using games for educational purposes requires continuous management of tools that make gaming sessions possible, these tools are often complex pieces of software and setting them up and using them requires a significant time investment and high levels of technological proficiency.[45] For educational games in formal settings, a teacher, trainer, or instructor need to be present during the play sessions. The person who is responsible and handles the situation need to be versatile. They need to be able to prepare necessary gaming exercises and integrate them into their existing teaching methods in correlation with the subject matter, but also act as a game administrator during the play session. Furthermore, this person also need to be able to tutor in an authoritative manner in both subject and gameplay related matters. In order for serious games to function efficiently (especially in an educational environment, such as a school), the person who administer the process need to possess various skills including technological know-how, gaming literacy, subject matter expertise and a strong

[45] B. Berg Marklund and A-S. Taylor, 'Educational Games in Practice: The challenges involved in conducting a game-based curriculum', *The electronic Journal of e-Learning,* vol. 14, no. 2, 2016, pp. 122-135.

pedagogical foundation.[46] This approach and versatility is also appropriate for other application areas as well. Organizations who want to implement serious games into their respective application area need to understand the work and effort behind it to make it work. As a customer you have to have the capacity to integrate and use the solution. This includes technical as well as usage capacity. Moreover, try to avoid that the game is strongly attached to a specific individual, making him or her a crucial part of the experience or solution, it may be unwise. Creating user guides and examples of how your game can be used (especially if you are aiming for an "open market") might be a good idea to cope with this problem and facilitate inexperienced users.

Because of this lack of technical experience (in tandem with all previously discussed problems with serious games), creating serious games which requires low maintenance is always a good idea. The same can be said about major complicated patches and/or expansions. It is most likely that you, as a developer has to get to the site and install everything yourself. Because of this, it is important to establish a plan of support in case anything goes wrong or something unexpected were to happen, so you, as a developer easily can assist the customer in an efficient and cost effective way. We have to ensure that it is easy and understandable to install, play, and evaluate the serious games, because as soon as something stops working, it is guaranteed that they will go back to previous, more efficient solutions. Remember, we are (in most cases) competing against books!

The goal of serious games (learning games) is to facilitate learning in particular subjects and for this to be a reality the games need to be able to show proof of that. Therefore, the ability to log progress, review, and customize the experience is important. Including tools and features specifically designed for the instructor, like “observer-mode”, the ability to pause or replay certain scenarios, and to change the outcome of the game can massively affect the user experience, both during and after the play session, ultimately affecting the learning outcome in a more positive way. Once the game is implemented into its right context, it might be good to demonstrate the practical usefulness and benefits of your serious games or digital solution to help instructors “unlock” its full potential. By fully understanding the game and its mechanics, a better learning environment can be set up.

6.2.3 Keeping your game-based solution alive?

When we talk about organizational use and support in this report it is because it is very important to keep in mind both as a client and as a developer. What happens when you are installing the game, and more importantly, what happens afterwards? What happens if there is an unexpected technical issue? The short answer to this is that there has to be someone available that can help when problems occur. Many of the examples we have mentioned in this report comes from research projects, and in those situations can be slightly different and more lenient. Many research projects are engaged together with different client organizations for purposes of R&D, and both us and the clients have wanted to explore new grounds and new potential solutions. These projects usually have a definitive ending point, a set time after which we will leave a prototype or product for the client to try out after the project is finished. We cannot, however, promise extensive support outside the actual research project since most of the researchers involved in the project will need to start devoting their time elsewhere. We have had experiences where we have collaborated with clients and end users for several months to create a serious game with good subject-matter appropriate content, only to have it fall into disuse

[46] B. Marklund, *Unpacking Digital Game-Based Learning: The complexities of developing and using educational games,* Doctoral Thesis, University of Skövde, Sweden, 2015.

very shortly after we have left the project and thus have taken a lot of the technical know-how with us out of the use context.

This situation is, as far as we have been able to discern, similar for most research-based projects. There is a set time window to produce things, and afterwards it is difficult to set up a system in which the majority of the people involved in the development team can be available for tech support and maintenance. In the SAREK projects, and others, we have tried to mitigate this by ensuring that we have a lot of time allotted in the project to be able to train people within the client organization to be able to use the serious game well and troubleshoot technical issues without our help.

And that is, essentially, our main recommendation for both clients and developers: ensure that you can, during your project, spread out the competencies regarding the technical back-bone of your serious game as much as possible before your development process is over. If only people within the developer team, or only one individual at the client organization, knows how to troubleshoot and maintain the functionality of the serious game, the entire use process of the game will be very frail. If the developers move on to a new project, the time they can spend doing tech support becomes very limited. And if the one knowledgeable individual at the organization needs to shift their working duties, the serious game might lose all its on-site support. Ensuring that as many people as possible have a familiarity with the serious game, as well as the technology it runs on, is crucial for keeping it alive for an extended amount of time.

6.3 Evaluating the solution

Serious games evaluation and assessment is a huge subject and only some of the fundamental parts will be discussed throughout this part of the chapter. Earlier we highlighted some of the benefits of gamification and serious games by show-casing the effects in various application areas. However, as of today, due to the complexity of serious games, no universal model or framework for evaluating serious games have been established. But the overall goal and sole purpose of evaluating serious games is to prove that the game meets its requirements in an efficient, effective, and reliable way. If this is not the case, it is going to be very difficult to persuade someone to implement it into their organization. Solid evidence and sound assessments is required to increase the acceptance of serious games and to establish them as valid tools for learning. But without proper evaluation methods in which objective goals and learning outcomes are analyzed, the industry will never grow and the untapped potential of serious games will be forgotten.

The evaluation of serious games are quite different from regular entertainment games because, in addition to entertainment, both immersion and effectiveness has to be taken into account. These are quite difficult to successfully measure due to their, in this case, subjective nature. And as we have discussed throughout this report, games can contain millions of different elements configured in various ways, be played on different platforms aimed at specific people developed for different contexts. The challenges lies within gathering useful, quality data for a broad range of serious games on different topics with various objectives, used in particular institutional contexts under uncontrolled circumstances.[47] Meaning that, the results and data should be characterized by generalizability and

[47] I. Mayer et.al., 'The research and evaluation of serious games: Toward a comprehensive methodology', *British Journal of Educational Technology,* vol. 45, no. 3, 2014, pp. 502-527.

validity.[48] So the big question is, how do one measure the effectiveness of serious games and can you be sure that the end users actually learn anything?

6.3.1 Important aspects and considerations

Four aspects need to be taken into considerations when evaluating serious games; **theoretical grounding**, **technical grounding**, **empirical grounding**, and **external grounding**.[49]

Theoretical grounding: the circumstance and context in which your game operates usually dictates the method of evaluation. Serious games are typically developed and designed to meet certain requirements and to achieve a specific purpose. Therefore, we "only" have to measure the effect of the intended purpose. The most common way to evaluate this (due to its simplicity) is to perform a qualitative or quantitative pre-test/post-test with groups of randomized people and compared it against a control group either by conducting surveys, questionnaires, observations, or interviews. The serious game's effectiveness can either be self-reported by the player, tested by subject matter related questions, or observed by the researcher themselves. With a pre and posttest we are able to establish a baseline of the participants in which the results can be compared against each other. This comparison between a before-and-after state of knowledge (which sometimes is perfectly fine) is however limited to the scope of the testing instrument. Because of this serious games should also be evaluated against the traditional form of training.[50]

Technical grounding: the importance of an iterative development process for serious games cannot be stressed enough, the same can be said about the evaluation of serious games. An iterative method facilitate the process of identifying strengths and weaknesses within a system or in this case, the game's design. Because of these changes '[a]n effective evaluation strategy will thus be a flexible one in which evaluation requirements are not held constant, but are adjusted according to findings from early iterations in game development'.[51] Since the model you are examine is not set in stone (until it is finished), the method of evaluation should not be either.

Empirical grounding: in order to properly assess the effectiveness of serious games it is important to base your conclusions on empirical findings and reliable data. Therefore, conducting your experiments in a controlled environment in which the game and its mechanics have been properly isolated to better observe the expected training outcome and to ensure that valuable data is gathered is important. In practice this means conducting several different experiments with different scientific methods to avoid errors and bias related problems but also to increase the results validity and reliability. Furthermore, the tests should involve people from the population of interest. I.e. a serious game specifically designed for teenager should not be tested and evaluated using adults.

External grounding: evaluating and stating that a game is effective under controlled circumstance is a good start. However, for the game-based solution to be really useful, the game need to be evaluated in its right context, in a real world setting and be tested under circumstances that are as close as

[48] M. Shapiro and J. Peña, 'Generalizability and Validity in Digital Game Research', in U. Ritterfeld, M. Cody, P. Vorderer, in *Serious Games: Mechanisms and Effects,* Routldge, 2009.
[49] D. Wilson, et.al., 'Serious Games: An Evaluation Framework and Case Study', *Proceedings of the 49th Hawaii International Conference on System Sciences,* 2016.
[50] D. Wilson, et.al., 'Serious Games: An Evaluation Framework and Case Study'.
[51] ibid, p.640

possible to its intended use.[52] To properly assess the effectiveness of your serious game, retentive learning should be evaluated by conducting longitudinal studies. However, these studies are far more time-consuming and expensive.

6.3.2 Challenges and problems

The process of evaluating serious games are usually riddled with challenges and problems, some of which are more common than others.[53] During our work with serious games we have found and experienced these problems first hand:

Recruitment - Finding people who can and want to participate in your studies is a time-consuming process. To organize and properly set up serious game experiment is difficult, especially if you investigate a sensitive subject targeted towards a specific audience. E.g., the games used in our Elinor-project were played with simple and straightforward controllers, but because of the patient's condition, even the simplest tasks meant an incredible effort on their part. Problems like this, in addition to the skepticism of serious games can cause low numbers of participants followed by high level of drop outs. Furthermore, a niche area such as stroke rehabilitation also meant a very small sample size to begin with.

You also have to take into considerations the different prerequisites within your test group. Some people might have little knowledge about the subject but play a lot of games, while other understand the subject perfectly fine but does not play any games at all. This could ultimately affect your results and outcome of the study. It is therefore important that each participant is tested under the same circumstances and that they represents the end users in a good and equitable way.

Context and environment - Evaluating your game under the exact circumstances it will be used is often a challenge. It is, however, very important to account for the usage context on the feasibility of the game. When you make such evaluations, for example in the form of a field experiment or case study, you end up in situations with many parameters out of your control. As opposed to an experiment under more controlled forms you may have people around who affect what happens. You should also be aware that you are intruding and there are cost for the receiving organization, at least in the form of idealistic nonprofit work on someone's behalf. As an example, if you test something in a school, you have to make sure that you have all necessary permissions and that your intervention does not have negative effects on the pupils.

Operationalization - What should be measured, how do you do it, and how can you ensure the findings are valid? Depending on your situation, context, and purpose of your game it might be difficult to pinpoint exactly what to measure and how to do it. To measure a game's effectiveness on basic mathematics might not be that difficult. But sometimes, serious games are used for more abstract problems such as motivational issues or behavioral changes which makes the process of evaluation a little bit trickier. How do you measure abstract skills such as teamwork or leadership and how can you be sure you are truly measuring it? If the purpose of your serious game is to inform the player about obesity related risks and problems, how do you evaluate that? Do you use questionnaires? Or do you sit down and interview the players? Or do you observe the players while they play? How can you make

[52] Ibid. pp. 638-647.
[53] K. Emmerich and M. Bockholt, 'Serious games evaluation: Processes, models, and concepts', *Computer Science,* vol. 9970, 2016, pp. 265-283.

sure that the knowledge is transferred from the game and applicable in the real world? All these options have different strengths and weaknesses which need to be carefully considered before you decided on how to evaluate your serious game.

Short and long term effects - Games are (in many application areas) novelties, but there is a difference between short term and long term results and studies. Games that show promise and short term positive effects does not necessarily mean a suitable solution for long term use. It might be the novelty and attractiveness of games, compared to traditional teaching methods, and not the content that captivate the users. Which means that even poorly designed serious games can show preliminary positive results. However, bad games, played over a period of time will become boring, which can decrease subject motivation and interfere with the learning process.

There are different ways to practically evaluate your serious games. Three of the most common methods are; completion assessment, in-process assessment, and teacher evaluation.[54] With completion assessment the end result or destination is all that matters. Did the player successfully finish the game? This could be an indicator that the player understands the subject. However, there is one big flaw connected to this method. Game completion does not necessarily mean that the player learned anything, he might have just been good at the game for all we know. In-process assessment on the other hand is more about the journey than the destination. With this method you track the player's progress by analyzing the mistakes and corrections made along the way. The last and final method could be seen as a combination of both completion and in-process assessment. This method requires assistive tools built into the game in which the tutor can track and control subject-related data to better evaluate the students. The previously discussed “observer-mode” is a great tool for this type of assessment. For instance, with this mode, the observer (e.g. teacher) could adjust the game’s difficulty level and content to better suit a particular type of player.

Whatever method you end up using, evaluation should be an integral part throughout the whole development process. By continuously evaluating your choices and understand how they ultimately will affect the player and building serious games with these guidelines and challenges in mind, will help you assess and create efficient and more effective serious games.

[54] D. Michael and S. Chen, ‘Proof of Learning: Assessment in Serious Games’, *Gamasutra,* 2005, https://www.gamasutra.com/view/feature/130843/proof_of_learning_assessment_in_.php (accessed 21 September 2017)

7 References

Altizer, R., and Zagal, J., 'Designing inside the box or pitching practices in industry and education', *Proceedings of the 2014 DiGRA International Conference,* vol. 8, 2014.

Backlund, P., and Hendrix, M., *The Educational Value of Serious Games: Summary and Overview of Current Research,* Lifelong Learning Programme Education and Culture DG, University of Skövde, 2010.

Bates, B., *Game Design,* 2nd edition, Boston: Cengage Learning PTR, 2004.

Bellotti, F., Berta, R., and De Gloria, A., 'Designing Effective Serious Games: Opportunities and Challenges for Research', *International Journal of Emerging Technologies in Learning,* vol. 5 no. SI3, 2010, pp. 22-35.

Berg Marklund, B., and Taylor, A. S., 'Educational Games in Practice: The challenges involved in conducting a game-based curriculum', *The electronic Journal of e-Learning,* vol. 14, no. 2, 2016, pp. 122-135.

Berg Marklund, B., *Unpacking Digital Game-Based Learning: The complexities of developing and using educational games,* Doctoral Thesis, University of Skövde, Sweden, 2015.

Berg Marklund, B., *Working with Educational Games: Fundamental guidelines for developers and educators interested in working with educational games,* University of Skövde, School of informatics, 2014.

Bethke, E., *Game Development and Production,* Plano, Texas: Wordware Publishing, 2003.

Boyle, E. A., et al., 'An update to the systematic literature review of empirical evidence of the impacts and outcomes of computer games and serious games', *Computers and Education,* vol. 94, 2016, pp. 178-192.

Calvo, A., et al. 'Tools and approaches for simplifying serious games development in educational settings', *IEEE Global Engineering Education* Conference, vol. 10, 2016, pp. 1188-1197.

Chen, J., and Ringel, M., 'Can advergaming be the future of interactive advertising', 2001.

Clegg, D. and Barker, R. *Case Method Fast-Track: A Rad Approach*, Addison-Wesley Longman Publishing Co., Inc. 1994.

Davis, F.D. (1989) "Perceived usefulness, perceived ease of use, and user acceptance of information technology", MIS Q., Vol. 13, No. 3, pp 319-340.

Eladhari, M., and Ollila, E., 'Design for Research Results: Experimental Prototyping and Play Testing', *Simulation & Gaming,* vol. 43, no. 3, 2012, pp. 391-412.

Emmerich, K., and Bockholt, M., 'Serious games evaluation: Processes, models, and concepts', *Computer Science,* vol. 9970, 2016, pp. 265-283.

Granic, I., Lobel, A., and Engels, R., 'The benefits of playing video games', *American Psychologist*, vol. 69, no. 1, 2014, pp. 66-78.

Griffiths, M., 'Video games and health', *British Medical Journal*, vol. 331, no. 7509, 2005, pp. 122–123.

Guthrie, C.H. (2014) "Who Are We Teaching? The Learning Expectations of "Digital Tribes" in the Classroom", International Journal of e-Education, e-Business, e-Management and e-Learning, Vol. 4, No. 2.

Guy, S., Ratzki-Leewing, A., Gwadry-Sridhar, F., 'Moving Beyond the Stigma: Systematic Review of Video Games and Their Potential to Combat Obesity', *International Journal of Hypertension,* vol. 2011, 2011, pp. 1–13.

Hamari, J., Koivisto, J., and Sarsa, H., 'Does gamificaion work? – A literature review of empirical studies on gamification', *Proceedings of the Annual Hawaii International Conference on System Sciences*, IEEE Computer Society, 2014, pp. 3025-3034.

Hanus, M., and Jox, J., 'Assessing the effects of gamification in the classroom: A longitudinal study on intrinsic motivation, social comparison, satisfaction, effort, and academic performance', *Computers and Education,* vol. 80, 2015, pp. 152-161.

Hellkvist, M., *Braille Hero: Feedback modalities and their effectiveness on alphabetic braille learning,* Master Thesis, University of Skövde, Sweden, 2017.

Jones, C., Ramanau, R., Cross, S., and Healing, G., 'Net generation or Digital Natives: Is there a distinct new generation entering university?', *Computers and Education,* vol. 54, no. 3, 2010, pp. 722-732.

Khaled, R., and Vasalou, A., ' Bridging serious games and participatory design', *International Journal of Child-Computer Interaction,* vol. 2, no. 2, 2014, pp. 93-100. 2014

Klimmt, C., 'Serious Games and Social Change: Why They (Should) Work', in U. Ritterfeld, M. Cody, P. Vorderer, in *Serious Games: Mechanisms and Effects,* Routledge, 2009.

Marsh, T., 'Serious games continuum: Between games for purpose and experiential environments for purpose', *Entertainment Computing,* vol. 2, no. 2, 2011, pp. 61-68.

Marti-Parreño, J., et.al, 'Factors contributing brand attitude in advergames: Entertainment and irritation', *Journal of Brand Management,* vol. 20, no. 5, 2012, pp. 374-388.

Martí-Parreño, J., Seguí-Mas, D., and Seguí-Mas, E., 'Teacher's Attitude towards and Actual Use of Gamification', *Procedia – Social and Behavioral Sciences,* vol. 228, 2016, pp. 682-688.

Mayer, I., et.al., 'The research and evaluation of serious games: Toward a comprehensive methodology', *British Journal of Educational Technology,* vol. 45, no. 3, 2014, pp. 502-527.

McConnell, S., *Rapid Development,* Redmond, Washington: Microsoft Press, 1996.

Michael, D., and Chen, S., 'Proof of Learning: Assessment in Serious Games', *Gamasutra,* 2005, https://www.gamasutra.com/view/feature/130843/proof_of_learning_assessment_in_.php (accessed 21 September 2017)

Michael, D., and Chen, S., *Serious Games: Games that Educate, Train, and Inform,* Boston, Thomson Course Technology PTR, 2005.

Rogers, E. M., *Diffusion of Innovations*, 4th edition, New York: The Free Press, 1995.

Salen, K., and Zimmerman, E., *Rules of Play – Game Design Fundamentals.* Cambridge, Massachusetts Institute of Technology, London: The MIT Press, 2004.

Schell, J., *The Art of Game Design: A book of Lenses,* Burlington, Elsevier, 2008.

Seaborn, K., and Fels, D., 'Gamification in theory and action: A survey', *International Journal of Human Computer Studies,* vol. 75, 2015, pp. 14-31.

Shapiro, M., and Peña, J., 'Generalizability and Validity in Digital Game Research', in U. Ritterfeld, M. Cody, P. Vorderer, in *Serious Games: Mechanisms and Effects,* Routldge, 2009.

Spinuzzi, C., 'The Methodology of Participatory Design', *Technical Communication,* vol. 52, no. 2, 2005, pp. 163-174.

Stapleton, A., 'Serious Games: Serious Opportunities', *Health Care,* vol. 1, 2004, pp. 1-6.

Taylor, A. S., Backlund, P., and Niklasson, L., 'The Coaching Cycle: A Coaching-by-Gaming Approach in Serious Games', *Simulation & Gaming,* vol. 43, no. 5, 2012, pp. 648-672

Venkatesh, V., Morris, M.G., Davis, G.B. and Davis, F.D. (2003) "User acceptance of information technology: toward a unified view", MIS Q., Vol. 27, No. 3, pp 425-478.

Volkswagen, 'Thefuntheory', *Thefuntheory,* 2009, http://www.thefuntheory.com/piano-staircase (accessed 21 September 2017)

Wang, H., and Singhal, A., 'Entertainment-Education Through Digital Games', in U. Ritterfeld, M. Cody, P. Vorderer, in *Serious Games: Mechanisms and Effects,* Routldge, 2009, pp.271-292.

Wilson, D., et.al., 'Serious Games: An Evaluation Framework and Case Study', *Proceedings of the 49th Hawaii International Conference on System Sciences,* 2016.