Improving the Quality of the Software Development Lifecycle with Gamification

Philipp Lombriser, Roald van der Valk

Utrecht University
Department of Information and Computing Sciences

Abstract. Gamification has become a trend in recent years. More and more companies recognize the benefits and have started to apply game elements for their internal and external business processes. Gamification has also been addressed by the different software development phases in order to influence people’s behavior in a positive manner. Different gamified tools exist to support requirements gathering, software development and software testing allowing to improve the overall quality of software engineering. This research paper consists of a state-of-the-art literature study on gamification and how its principles may be successfully applied to the process of software development. Furthermore, the literature research study was compared with two case studies that were conducted with experts from the gamification industry. The results reveal that many game elements are used for solutions to improve the development phases. However, a cross-integrated gamified system over the different phases has not been encountered in practice yet.

1 Introduction

Stating that games are engaging is becoming increasingly redundant. In a report published in April 2014, the Entertainment Software Association claims that 59% of Americans play video games [1]. From the same report, it can be concluded that most notably casual games are making this form of entertainment more and more accessible. Looking at this trend, it is only logical that it would be desirable to take some elements of what is driving this popularity, and implementing it in other situations. This so-called gamification has already been successfully adopted in several large organizations with many positive results, including better long-term collaboration, creativity, productivity, loyalty and learning [2].

On the other side, many software projects nowadays still fail to deliver on time, within cost and with expected scope. One main reason for project failure is unsatisfied or not met requirements, often caused by poor stakeholder collaboration, incomplete understanding of needs, insufficient domain knowledge, fluctuating requirements and others [3]. Therefore, this paper will focus specifically on the software development process and how it can be improved through gamification.

The promises of gamification in this context include a reduction in the amount of time necessary for the project, a decreased amount of errors, redesigns and development dead-ends [2]. Indeed, the focal point of this paper will be on the quality of the process, rather than that of the resulting product. The state-of-the-art of gamification will be thoroughly explored, starting by introducing all the major concepts. Using this basis, adfa, p. 1, 2011.

© Springer-Verlag Berlin Heidelberg 2011
the paper continues by explaining the term of gamification, followed by a discussion of different types of game elements. Furthermore, the use of gamification in software engineering is demonstrated, including real-life examples as well as a conceptual approach for integrating it over the entire software development lifecycle. The second part of the paper consists of two case studies that were conducted with two companies that apply gamification for customer solutions.

2 Context and motivation of study

In order to study gamification in software engineering, the aim and scope of the study needs to be defined. Based on existing literature, it can be concluded that gamification has already been successfully applied in different work situations and has improved the engagement and productivity of employees [4]. In this study, however, the focus is specifically on software development. The aim is to provide a clear overview of exactly how gamification can improve the process quality, for instance by ensuring a stronger degree of engagement with the project. Therefore, the main research question of the study is: in what way does gamification enhance the software engineering process? In order to answer this question, the following sub-questions were formulated:

- RQ1: What is gamification?
- RQ2: What are the game elements that can be used for software engineering?
- RQ3: What are today’s best practices of gamification in software engineering?
- RQ4: How can gamification be applied over the entire development lifecycle?

3 Literature review method

To get a clear picture of the current state of gamification as well as the process of software engineering, a literature study was conducted. For this, different methods were used for multiple sources. The primary source of information was scientific literature. Using Google Scholar, relevant scientific journals and articles were identified. Aside from these, a number of books were found pertaining to either gamification or the software development process. To complete the selection of literature, less formal sources in the form of websites and blogs were used. By using this combination, it was ensured that the research was primarily based on more widely accepted and respected works, while still taking advantage of all the knowledge available elsewhere. Among these less formal sources were statements made by so-called Gamification Gurus, who are experts in the field.

The process of selecting the most relevant source material was done in several steps. Papers found using the selected keywords were first filtered based on title and on relevance to the search terms (i.e. placement in the results list). In case different papers tackled the same topic, a selection was made based on criteria such as content, authors, date of publication and number of citations. At this point, reasons for exclusion were often an apparent lack of relevance for the study, or the paper taking too broad or too vague an approach to the topic. Back references were used to gain a more complete picture of the scientific developments regarding gamification.
For statements made by Gamification Gurus, their status as an expert played an important role. In turn, stories from companies where gamification had been successfully implemented were selected in such a way that they would provide insight into the process, benefits and best practices.

4 State-of-the-art study

The state-of-the-art literature study first clarifies the term “gamification” and its principles. Next, the different game elements are introduced and how they are currently being applied in software engineering. The last part of the literature study examines a cross-integrated gamified solution in the software engineering domain.

4.1 Gamification

Games have been around for thousands of years and their main purpose is to entertain people and create pleasure. People nowadays are dedicating a great amount of their time to video games [5]. Statistics show that daily time spent playing video games were on average 23 minutes per capita in the United States in 2013. This number is expected to exceed 28 minutes by the year 2018 [6]. Furthermore, there were more than 157 million active gamers listed in the United States in 2012 [7]. These results not only demonstrate the increase of playing games, but also the high potential and possibilities that games imply.

Salen & Zimmerman [8] define a game as “a system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome”. Unfortunately, most common games are considered to be unproductive, with no valuable outcomes. However, with gamification it is possible to engage users in solving real-world problems, which activities and outcomes are value-adding and not just a waste of time. Furthermore, many research papers have shown that when people are engaged, they are likely to be more productive, resulting in higher quality outcomes [9]. Gamification can be used in environments where it is difficult to influence people’s behavior and improve their motivation and engagement.

Zichermann & Cunningham [10] describe gamification as “the use of game thinking and game mechanics in non-game contexts”. To distinguish games and gamification, a two dimensional model can be used. The vertical axis represents the purpose and fun dimension and the horizontal axis gameplay [11]. The main difference is that games are just for fun and entertaining, whereas gamification has a certain purpose, without any gameplay. Figure 1 shows the different types of game thinking and where gamification is classified.
Gamification does not create games, but rather takes certain game elements from classical video games and applies them to real-world (business) problems. These game elements are typically known as points, badges and leaderboards (PBL) and keeps players engaged and motivated in their activities [12]. Points are responsible to keep track of score, provide feedback, and display progress, whereas badges are representations of achievements. Leaderboards allow users to directly compare themselves in a ranking. More useful game elements that were very often encountered in the literature study are stories, quests, levels, rewards and avatars [13]. Stories are the background narratives and create the dramatics and arouse emotions. A quest is an assignment or a task with a clear goal that must be accomplished by the user. Rewards can exist in form of items or virtual currencies and are presented to the player when a certain task is achieved. Levels help to make tasks more challenging as the user gains new skills and experiences.

From a motivational perspective, levels can help to maintain user engagement for an optimal experience. Csikszentmihalyi [14] describes the theory of ‘flow’ in a manner to keep the level of challenge in alignment with the user’s skills. If the level is too difficult, the player will feel anxious and lose interest in the game. On the other side, if the game is too easy, the player will be bored and stop playing the game as well. Therefore, the balance of these two dimensions is important to keep players engaged. Figure 2 shows the different dimensions and aspects of the flow model.

An avatar allows the user to create an individual virtual character with different types of attributes that can be personalized to his interests. Some attributes of the avatar might enhance while the user levels-up in a game. The usage of avatars can help to increase performance and motivation [15].

Figure 1. Types of Game Thinking
In addition to the aforementioned elements, feedback loops are an essential part of gamification. They provide information about how users are doing in their process. For instance, a score list reflects clear and distinctive information to the player about their current status. The feedback is what creates the motivation for further actions and keeps users engaged. The feedback loop is to be considered less as a game element, but more as a game mechanic, because it ties several game elements together.

In this paper only the most important and common game elements that are potentially relevant for software development are mentioned. For more information, an extensive catalogue of game elements and game mechanics can be retrieved from the Gamification Wiki, which is located on Gamification.org [16].

![Figure 2. The Flow Model](image)

### 4.3 Gamification in software engineering

Now that the potential of the different types of game elements are unveiled, this part will demonstrate how they are applied in software engineering. To incorporate different aspects of gamification into a company’s software development lifecycle (SDLC), it is important to focus on the organization’s internal perspective. A successful system should increase software engineers’ work performance and motivation, resulting in higher software quality, with less changes and defects. Furthermore, projects will more likely be delivered within time and budget [17].

The standard process of a software development project consists of five phases. These are ‘requirements definition’, ‘system and software design’, ‘implementation and unit testing’, ‘integration and system testing’ and ‘operations and maintenance’, as shown in Figure 3. Depending on the methodology used, the software lifecycle might either be straightforward as in Waterfall, or incremental as in Rational Unified Process (RUP) [18].
In the analysis phase, the system’s functionalities, goals and constraints are established in form of requirements. Next, an architecture is constructed based on the identified requirements. In phase three, the system is developed and tested in a set of program units. The separate programs will then be integrated as a complete system and tested against the requirements. Finally, the system will be installed in the customer’s operational environment. This phase involves fixing errors and bugs, which were not detected during testing phases.

Some researchers and companies have already gamified certain parts of the development process. This part will elaborate on some successful practices of the different phases.

4.3.1 Requirements Definition: iThink

iThink is a gamified approach to improve the collaboration and participation in the phase of requirements definition [19]. The authors analyzed several requirement elicitation practices and then developed a system with the appropriate game elements. The tool is based on the theory of the six thinking hats [20]. The idea behind this method is to stimulate parallel thinking in teams in order to be more effective and avoid team conflicts. Each hat has a different style of thinking and is represented by a unique color. This approach can be used to discover new requirements and discuss existing ones.

The main game elements used for iThink are points and progress bars. Points can be acquired through different user actions, which are mapped to one of the hats. For instance, a user can win points by submitting a new requirement, rating a requirement, commenting on a requirement and completing a discussion of a requirement. In Figure 4, an example screen can be seen, in which the user is asked to reflect on a requirement in four different ways, each awarding a certain amount of points. The points are then

![Figure 3. Software development life cycle](attachment:image.png)
reflected on a progress bar. A prototype of this tool was built and tested in two case studies. The results demonstrate that collaboration was improved and people enjoyed using the system. Furthermore, project managers were very satisfied with the generated requirements.

This research is one example of how game elements can be applied to a non-gaming context. Nonetheless, there were only two types of game elements used in this case to increase user participation and motivation. The game still leaves much potential to further gamify requirements elicitation.

Figure 4. Example of iThink

4.3.2 Software development: Code Hunt and Visual Studio Achievements

In the coding domain, many educational web-based platforms exist, targeting students and teachers to learn and advance their programming skills. For instance, Code Hunt was released by Microsoft Research and uses puzzles to encourage players to learn how to code [21]. The system uses the feedback mechanism as a gameplay. Every puzzle must be solved with a piece of written code that is verified by the system. If the code is wrong, the system returns an error. If the code is correct, the player wins the quest and continues to the next level. As the player keeps playing, his skill rating goes up and the quests get more difficult, thereby maintaining ‘flow’. This example of puzzle solving for educational purposes demonstrates the positive effect of learning, and might even be addictive.

Using gamification for learning has many benefits, but it can also be used to solve real business cases in software development. In 2012, Visual Studio introduced a new plug-in, bringing achievement badges to the development phase [22]. With Visual Studio Achievements, developers can unlock badges based on their activity. Unlocking achievements is done in the background during the compilation of the program. New obtained badges pop up in the lower right corner of the development environment to visualize the achievements. Furthermore, the personal player profile is updated and the
player’s position in the leaderboard is re-ranked. The plug-in also allows to tweet about new achievements and share them on Facebook so players can brag about their status.

Overall, there are currently 47 achievements to be unlocked within six categories. Figure 5 shows the badges of the six categories that can be earned. An example for the first badge is to install 5 extensions to Visual Studio. For the last and most advanced badge, an example could be to add 10 items to a sequence diagram.

![Figure 5. Six categories of achievements](image)

Other integrated development environments (IDE), such as Eclipse, also recognized the potential behind gamification and now offer an achievement plug-in as well.

### 4.3.3 Testing: Microsoft Language Quality Game

The behavior organizations want to evoke for testers, is to find as many bugs and defects as possible and report them properly. For many people this task might seem boring. However, game elements can make testing more entertaining, just as with other applications. Much of what testers do can be thought of in gaming terms.

Microsoft is exemplary when it comes to testing. The company has developed a ‘Language Quality Game’ (LQG) to improve the quality of their products and increase productivity at the same time [23].

The standard business process at Microsoft uses professional language translators for the textual content of the Windows operating system. In a second step a quality assurance team is responsible to verify the translation. In some countries it was very difficult to find enough people to translate and review the dialog boxes. To overcome this problem, LQG was established for the review process step. Native speaking citizens around the world can voluntary participate in the game and review Windows’ dialog boxes. In this game, users are awarded points for every mistranslation found and then ranked on a public leaderboard. The scoring system does not just compare individual players, but also aggregates scores on country level. The Language Quality Game created a competitive dynamic environment, followed by a significant increase of reported defects. Over 7,000 defects were reported across all the 36 languages within one month.

Next to points and leaderboards, the system uses levels to keep players challenged. Each level consists of 25 randomly selected images that need to be reviewed. Once a player has completed a set of images, he will move to the next level, where 25 new images are presented. Furthermore, players can also earn badges, known as ‘Markup Pen Colors’, while leveling up.
4.3.4 Cross-integrated gamification solution in SDLC

As demonstrated in previous chapters, certain phases of the software development process have already been gamified. However, these phases are still separate and not yet cross-integrated into one gamified environment in form of an ecosystem. Many papers and articles state how game elements can improve the overall software development lifecycle, but no case studies for an interdisciplinary approach were yet conducted, neither do any best practices exist. Fecher [13] did a first step in this direction, by creating a conceptual proposal of an interconnected gamified process. The author investigated several game elements and analyzed how they can be utilized to make software building more productive and fun. He proposes a multi-layer architecture containing different game elements, as shown in Figure 6.

The story element is located in the bottom layer and serves as the basis for the entire software development lifecycle. Business requirements, project context and organizational culture can be part of the story. Next, epic-quests, quests and achievements are built upon the story to give players a clear goal. It is important to note the difference between quests and epic-quests, as the latter is more challenging and can only be won within a team. Quests should be constructed depending on the particular software phase. The third layer consists of levels, points and rewards, which are dependent on quests and achievements. Solving quests not only increases the level, but also allows to earn more points and rewards. The avatar is positioned on the top of the stack and takes part of the entire development process. The avatar’s characteristics is based on the level, as well as points and rewards achieved during the project lifecycle. For instance, as the player earns more points, the stronger and better his avatar gets.

<table>
<thead>
<tr>
<th>Avatar</th>
<th>Level</th>
<th>Points</th>
<th>Reward</th>
</tr>
</thead>
<tbody>
<tr>
<td>Epic-Quest</td>
<td>Quest</td>
<td>Achievement</td>
<td></td>
</tr>
<tr>
<td>Story</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 6. Conceptual architecture for gamification in SDLC

4.4 Preliminary conclusions

Conclusion 1: In this research the hidden potential and increased business attention of gamification for software engineering was highlighted. Not only can gamification support the enhancement of user motivation and engagement, but also the overall productivity that correlates with motivation [4]. Applying designated game elements to the software development lifecycle can have a positive effect on the software quality as well. Therefore, software projects are more likely to be delivered within time, budget and with fewer defects [24]. The following conceptual model illustrates the relationships between the different concepts.
Conclusion 2: Many game elements are obtained from classical video games and can be incorporated into existing systems. However, there is no complete answer to which elements should be used. The decision depends on the circumstances of the business environment, as well as the players interacting with the system. The two case studies will elaborate more on these types of game elements.

Conclusion 3: Today, many approaches exist to apply gamification to the software development process. The results show positive effects on people's behavior and allow to increase the performance of SDLC [24]. However, there is no proof of concept that demonstrates how a cross-integrated system can be gamified over the entire development lifecycle. The framework introduced by Fecher [13] shows on a conceptual level how this could be achieved in a particular way, but his research still lacks of a practical approach.

In the next step two case studies were conducted with companies that have experiences in the field of gamification. The aim of the case study is to validate the constructed hypothesis and conclusions. The case studies should identify if gamification and its game elements significantly improve work performance and positively influence the quality of the software delivery process.

5 Case Studies

In order to examine and compare the current state of gamification, especially in the field of software development, two case studies were conducted. First, the method of conducting these studies is described. Next, the findings for each of the case is given and compared, followed by a validation and conclusion part.

5.1 Empirical research methods

The case studies were used to find the underlying principles and validate the findings from the state-of-the-art analysis. For both case studies a qualitative research method approach with a semi-structured interview was conducted. All questions were asked in a manner to answer the hypothetical research questions and conclusions. The questions and their relationship to the state-of-the-art findings are listed in the appendix part.

The two participating companies are Subatomic and Enterprise Gamification Consultancy LLC. Both companies have experience with creating gamified solutions for customers. Subatomic is a young start-up company located in Amsterdam. Their experience and expertise were helpful to evaluate more the generic part of gamification. Enterprise Gamification Consultancy LLC was founded in 2013 by four of the top
gamification designers and operates globally. Their experience and expertise was useful to focus more on the software development process. Both interviews shared their insights and allowed us to compare the findings with the research questions.

5.2 Case Study 1: Enterprise Gamification Consultancy LLC

The first case study was conducted with one of the most famous gamification gurus in the world, Dr. Marius Herger. He is currently ranked on the 6th place of the gamification leaderboard \[25\]. Dr. Herger has written several books on gamification and has been a strategist at SAP Labs in Palo Alto, California for 15 years. Furthermore, he is the CEO and founder of Enterprise Gamification Consultancy LLC. He is a true believer in gamification and confirms that gamification works when done properly.

Currently he is working on projects to improve sales departments to acquire new customers in a timely and costly manner, but also to master new sale skills through serious-gaming. With his long-term experience in gamification, he has seen many initiatives and attempts in software engineering. Therefore, this case study was intended to talk about best practices of gamification in the software engineering discipline. The findings were compared and validated with his experience and expertise.

These findings will first examine game elements and motivation. Afterwards, it will continue by looking at the implementation process used. Consequently, different possibilities of using gamification in software development will be discussed.

5.2.1 Game elements

When talking about the different types of game elements (Dr. Herger rather prefers the term “game design elements”) a clear distinction between intrinsic and extrinsic motivation must be made. According to the interviewee, certain game elements, such as leaderboards, scores and badges are considered to be more extrinsically motivating, whereas stories and quests provoke more intrinsic motivation of players. The focus of the scope was mainly on the classical game elements such as points, badges and leaderboards.

Motivation

Extrinsic motivation: With extrinsic motivation the human behavior is driven by external rewards coming from the outside of an individual. An example of external incentives from a business perspective would be bonuses. Money very often stimulates people to achieve their goals, but it can be very ineffective after a long period of time. However, it still helps to influence people’s behavior, but for a long-term success, the emphasis should be on intrinsic motivation \[26\].

Intrinsic motivation: Intrinsic motivation originates on the inside of an individual and is driven by internal rewards. Coon and Mitterer \[27\] describe intrinsic motivation when people simply enjoy an activity or see it as an opportunity to explore, learn, and actualize their potentials without any external rewards. People simply want, for their own personal reasons, to perform an action, because they enjoy the activity itself.
Examples of game elements to evoke motivation more from the inside could be a story. Since our childhood we enjoy reading books or watching good movies with an appealing story. This is one reason why in the United States storytelling is very often used in management practice to persuade people.

“Zombies, Run!” is a very good illustration of an app with a compelling story and gameplay behind it. The app can be used for running, during which players listen through headphones to a zombie apocalypse narration. The players of the game have to rebuild civilization while escaping from zombies.

Nike also launched an app, called Nike+, to gather information about runner’s habits and performance. However, the software is mainly data-based and only uses external incentives, such as progression, scores and leaderboards.

Overall, it is crucial to have a proper balance between intrinsic and extrinsic motivation to successfully change people’s behavior through gamification. A thorough analysis of game elements and their possible combination should always be considered when implementing gamification.

**Leaderboard**

The choice of using a leaderboard for an internal business process has to be carefully made. The idea behind a leaderboard is to generate competition among players. While this works well for Nike+ and “Zombies, Run!” to keep people competing against each other, it can become very conflicting within an organization. Most companies prefer to have a collaborative and harmonized culture, rather than a competitive workplace environment. Workplace competition can create unhealthy rivalries that result in workers resenting one another and in undue stress. In the worst case it even causes unethical behavior among employers.

To sum up, leaderboards should only be used in the software development process where competition is necessary and considered to be supportive. The goal of an organization must be to encourage collaborative behavior and not competition.

**Players**

Selecting the right game elements also depends on the players interacting with the system. The first step is to get a clear understanding of the existing core problem. If the underlying issue is a human behavioral problem, then gamification is an option to make a change in order to obtain the desired behavior. If the problem is more a technical or business process problem, then gamification might not be the right choice.

Enterprise Gamification Consultancy LLC tries to avoid the term “users” and instead prefers “players”. A player is defined as “a person who plays the game or interacts with the gamified system” [2]. Gamification designers should always try to think from a player’s perspective and make the system entertaining. If designers think from a user-centered design, then the main focus is on usability, visibility and legibility, but not on enjoyment.

Furthermore, the consultancy company uses personas to identify players interacting with the game. Personas are visual stories with a detailed description of the actual
players. While personas are very common in user-centric design, they are also applied in game design. Defining the players and selecting the right game elements requires much empathy from the designers to understand the motivations and behavior [2].

5.2.2 Gamification in Software Engineering

The third research question tackles the problem of current innovative tools in practice of the software engineering discipline. Many more gamification solutions were mentioned by the CEO of Enterprise Gamification Consultancy and are also represented in the Enterprise Gamification Wiki [28]. The following software phases list more examples and benefits of existing approaches.

Requirements Definition: The requirements definition phase requires a great amount of creativity and innovation by the community (stakeholders) to find new solutions or distribute existing ideas for a software product [18]. One way to enhance the flow of creativity is with an innovation management tool. Besides iThink, which is specific for requirements engineering, several more tools exist that motivate players to share creative ideas. Be-novative is a solution where players can read and evaluate the ideas based on their impact and feasibility and track the implementation. The main goal of the tool is to inspire creative and anonymous idea-generation within large enterprises. Similar tools are SpigitEngage and Venture Spirit that also motivate employees to spread their ideas.

Software Development: Dr. Herger confirmed that many online tools exist to learn and teach programming with great success. Besides to Code Hunt from Microsoft several other learning platforms exist, such as Codecademy, CheckIO, Ruby Warrior, Try Objective C and many more. There is even a game called ScratchJr that teaches children above 8 years programming. ScratchJr has developed an own programming language that allows children to program their own games and animations.

Testing: Primarily, gamification is currently in place to enhance quality assurance. One famous software code bug fixing game mentioned is “Bugathlon”. Bugathlon is used for Odnoklassniki, the largest Russian social network and usually lasts for five days every few months. Within these days, bugs are selected and must be fixed by the developers. The winner is the player who improved his or her skills the most, and not who solved most bugs. The results of the event demonstrate a significant improvement of bug fixing. Within the five days half of the chosen bugs were fixed by the developers. SAP uses a similar approach called “hackathon”. A hackathon is an event in which software developers collaborate intensively on software projects between a day and a week. The main purpose of the event is for education or social interaction. In many cases the events are aimed at creating useful software through players’ innovation and creativity.

Maintenance: The focus in the state-of-the-art research part was mainly on requirements definition, software development and software testing. However, as soon as a software is released and installed in the customer's business environment, it needs to be maintained and supported. In this phase, IT supporters can be incentivized with game elements to effectively process and solve incidents or events. An example of a successful gamified IT operation management system is offered by Hewlett Packard...
(HP). Players obtain certain achievements for their work and see their progress on a timeline to get instant feedback [29].

**Knowledge Sharing:** Not only programming and quality assurance are important in software projects, but also documentation and knowledge sharing. SAP built a community called SAP Community Network that was gamified to enable SAP engineers, developers and other experts to share ideas, learn, innovate and connect with others. Players of the community network are encouraged through points to stay active and support their fellow colleagues within the enterprise [30]. StackOverflow is another good example of an online community network intended for software developers to ask and answer questions. Based on user voting, questions get higher or lower ranked and players earn reputation points and badges.

**Cross-integrated solution:** Dr. Herger is not aware of a fully gamified system over the entire software development process, nor has he ever heard of a business that used an interdisciplinary gamified approach. He claims that only certain pieces of the development lifecycle have been gamified up until now. It might be possible that certain companies have all phases gamified, but not cross-integrated.

## 5.3 Case Study 2: Subatomic

The second case study was done as a semi-structured interview with Joanna Ioannidou of Subatomic, a company that advises on and designs user experience improvements. The Amsterdam-based company has worked on many successful projects, including the game Speel Je Toekomst for Aegon, which has won the “Grand Prix 2013” at the Red Dot Award.

Subatomic does not concern itself solely with gamification, but with user experience improvement more broadly. For this reason, they prefer the term “user” over “player”, as it is broader and less connected to preconceptions on gaming. For each project, clear goals and expectations are defined in advance in order to ensure the proposed solution will be in line with what is expected. Figure 8 shows this as being the first step of the entire process. It then continues with understanding the audience and understanding the need for change, which corresponds with what the literature described in section 3. These first three steps are crucial, according to the interviewee, as the result of these phases combined define what it is that should be developed.

![Figure 8. Implementation Process](image)

In this process, it is important to emphasize that the result of each of the phases should correspond with the rest. For instance, a company may want to implement a leaderboard, as was also described in the first case, however with the aim to improve
cooperation. Obviously, this would not achieve this goal. It is crucial that the effect of each game element and mechanic are explained clearly and applied appropriately.

Another important point of attention is the audience, as this defines what game elements may or may not work. This has to do with what motivates different kinds of people. For example, older people will be less likely to pay attention to receiving a special digital badge than younger people, making this element less effective. In this case, it is a good idea to apply gamification without computers. Another factor that can change how people experience different game elements, or even simply how they cooperate, is culture. The example that was mentioned was the difference in working culture between Western and Asian countries. When designing a proposal for the system, this should definitely be taken into account. Dr. Herger also talked about this aspect of gamification when discussing the different kinds of motivation.

Apart from the leaderboard that was named earlier, other possibly relevant game elements are quite numerous. A number of elements were discussed as part of the interview, as examples of how they were implemented in other projects and what their intended effects were. Awards and badges that show certain achievements or skills could, for instance, increase the prestige of certain aspects of work by making it visible in a desirable manner.

Another example of using game elements to improve cooperation and efficiency that was discussed is setting up a system that will allow employees to experience what it is like to have a different role within the company, or even as a customer. Since every role comes with different priorities and desires, this will create a lot of insight and understanding for each other's work. As a result, conflicts may be avoided and expectations may be managed better.

Unfortunately, most projects at Subatomic are confidential. One of their projects is “Speel Je Toekomst”, which is a web-based game that aims to give users insight into their financial future. To start the game, the user creates an account with information about their current financial situation and where they expect their future to be like. Subsequently, the game shows how to get to the possible future state. The player gets to experience three events during the game, each of which is user-selectable. Examples of these events include getting a baby or getting hit by a car. The user then sees how this affects the future situation. This is called “gamifying information”. The game elements used were achievements, using a character and a (sarcastic) narrator.

Looking specifically at the software development process, Subatomic does not have experience with gamifying these kinds of projects. Nevertheless, the approach would be the same as with other projects, and can be used to examine what game elements would bring about the desired change. Looking at the process as shown in Figure 8, the key question would be what behaviour should be encouraged. If it is competition, for example, then the badges or awards mentioned earlier can be an option. According to the interviewee, there is no reason to deviate from this process.

The biggest difficulty of gamification, according to the interviewee, is a lack of understanding of what gamification is. Many people have prefixed conceptions about what it is, based on their ideas of games in general. In explaining what the process entails, it is important to emphasize that what is asked for, is not necessarily what is
needed to achieve the desired result. Furthermore, many people have set associations when confronted with the term “gamification”, based on their own experiences. These associations may differ greatly from some avid gamers expecting elements from their favorite games, to skeptical non-gamers who do not associate the term with anything serious or potentially interesting. It should be noted, in contrast to the process, that this pitfall is more focused on working with a client, rather than the implementation phase of gamification that was the case in the earlier study.

5.4 Case study comparison

The following table lists the summaries of the different outcomes from the two case studies. The information shown between the two cases are not mutually exclusive, but more highlight the different emphasis of the companies.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Subatomic</th>
<th>Enterprise Gamification Consultancy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case Study Focus</td>
<td>• More on the general principles behind gamification and its benefits</td>
<td>• More on gamification within software engineering</td>
</tr>
<tr>
<td>Company information</td>
<td>• Focusses on User Experience Improvement</td>
<td>• Was founded in 2013 by 4 top gamification gurus</td>
</tr>
<tr>
<td></td>
<td>• Has worked on award-winning projects such as Speel Je Toekomst</td>
<td>• Was the first gamification consultancy for business</td>
</tr>
<tr>
<td>Game Elements</td>
<td>• Emphasizes ties between elements and resulting behavior</td>
<td>• Operates globally</td>
</tr>
<tr>
<td></td>
<td>• Should not be limited to elements on screens</td>
<td>• Helps organizations to apply gamification</td>
</tr>
<tr>
<td>Gamifying a system</td>
<td>• First understand the audience and the problem</td>
<td>• Makes a clear distinction between extrinsic and intrinsic motivation.</td>
</tr>
<tr>
<td></td>
<td>• Elements requested by client may not be what is needed to achieve the desired effect.</td>
<td>• Avoids leaderboards where collaborative behavior in organizational teams is required.</td>
</tr>
<tr>
<td>Users of the system</td>
<td>• Can vary greatly</td>
<td>• The core problem first needs to be identified</td>
</tr>
<tr>
<td></td>
<td>• Prefers the term users instead of players.</td>
<td>• Only if the core problem is a behavioral problem, gamification can help</td>
</tr>
<tr>
<td></td>
<td>• Important to understand the users in order to find correct game elements.</td>
<td></td>
</tr>
<tr>
<td>Gamification in SDLC</td>
<td>• Do not have direct experience with gamifying SDLC.</td>
<td>• Has many experience with different kinds of tools and</td>
</tr>
</tbody>
</table>
Would approach it in the same way as other projects, no reason to change this.

Demonstrated more existing solutions that are in SDLC related phases, such as knowledge sharing and documentation

Is not aware of any cross-integrated business solutions

5.5 Threats to validity

In order to test the validity of this study, the factors that may affect is need to be examined. These factors have been split into two categories. The first category looks at internal validity, which examines whether the methods used and reasoning based upon the discovered information is sound. The second category explores the external validity, which concerns whether the conclusions drawn can be generalized [31].

5.5.1 Internal validity

There are two big factors in assessing threats to internal validity in this case, namely instrumentation and selection of interview subjects. These two factors form the foundation of how the information was gathered for this paper and need to be closely examined in order to determine what factors may detract from the correctness of this information.

In terms of instrumentation, both the selected method and how it was employed should be examined. As described earlier, the research was performed as a semi-structured interview, due to the descriptive nature of the study. While the questions were formulated in a neutral manner and were based upon the research questions, the information gathered should form a proper basis to answer the research questions. However, it may still be the case that some information was not reported, or that the interview itself may have influenced the interviewees.

The selection of interview subjects may have affected the validity of the study. Since in both cases the subject worked in the field of gamification, their views on this area of study are bound to be positive. However, since the focus of this study was not on the suitability or success of gamification in a general sense, but rather about what methods could be employed to extend gamification into the field of software development, this factor should not be too problematic.

5.5.2 External validity

As described earlier, external validity examines to what extent the conclusions drawn from this study can be generalized. It should be kept in mind that the conclusions drawn are not based on a (statistically sound) group of individual cases, but rather based on an in depth examination of a small amount of instances. Since only 2 cases were
examined, the conclusions of this study may not be extended to other situations. In order to generalize the results, a larger study would be needed with a larger sample. The findings presented here can serve as a basis for such a study.

5.6 Conclusions and future directions

Gamification is all about fine-tuning and selecting the right concepts and technologies. People’s behavior must be continuously measured and improved when necessary. However, gamification is only one element of an enterprise system and cannot stand on its own and solve all problems. This research aimed at defining the term gamification and its principles together with the different types of game elements. Furthermore, an examination of literature was conducted to evaluate the current status of gamification in software engineering.

The phenomenon has been successfully used in the last few years with an upward trend to change people’s behavior and improve work performance [4]. Applying game design techniques can motivate people to improve the complex software development process across the lifecycle. It is already used in all different phases of software engineering and some companies might even have gamified all phases in a certain way. Nonetheless, it is still unknown if a cross-integrated solution exists that combines different phases into one gamified environment. This is still an open question to be answered by future research.

Furthermore, derived from the case studies and the literature review, it is important to identify the business goals when gamifying a certain business problem. Understanding the core problem and people’s behavior is also very essential to successfully adapt gamification. Once the problem and players have been studied the proper game elements must be chosen to enhance intrinsic and extrinsic motivation. Both types of motivation must be well balanced to give the player the experience of flow [14].

Overall, the results from the case study encouraged the reliability of the literature study, since there were no significant differences found to the three preliminary conclusions. Most of the discussion from the interviews added value, but no issues or discrepancies were identified. Nevertheless, since only two companies were studied, there is still the risk that some developments in this area of study were not discovered.

The need for future studies has been made clear at several points in this study. An integrated approach for implementing gamification throughout the software development process is still lacking, and would provide for an interesting study as to the effectiveness of such an approach. An integration process as such could be done by first integrating 2 of the current phases into one game. Next, the other phases could be added step by step until the game covers the entire process.

Another area of study that could add robustness to the arguments on whether or not to implement gamification is a large study focusing on delivering more statistical evidence of the effectiveness of different game elements. While there has already been some studies in this field, the promising results should be cause for greater interest in examining gamification.
6 References


Appendix

7.1 Interview Questions

All questions acted as a guideline to structure the interview and were applied to both case studies. The right column shows the relationship to conclusions from the preliminary conclusion part in chapter 4.4. The questions were all chosen in order to validate our preliminary findings. All questions were created by the researchers themselves and were chosen in order to answer our hypothetical research questions.

<table>
<thead>
<tr>
<th>Nr</th>
<th>Question</th>
<th>Question Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>What is your company's business?</td>
<td>General information</td>
</tr>
<tr>
<td>2</td>
<td>Which projects are you currently working on?</td>
<td>General information</td>
</tr>
<tr>
<td>3</td>
<td>How do you measure the success of your implementations?</td>
<td>To validate conclusion 1</td>
</tr>
<tr>
<td></td>
<td>Question</td>
<td>To validate</td>
</tr>
<tr>
<td>---</td>
<td>--------------------------------------------------------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>4</td>
<td>Do you have any experience with gamifying software development process?</td>
<td>conclusion 1</td>
</tr>
<tr>
<td>5</td>
<td>Do you know, if a case study exists, where the entire SDLC is gamified into one system?</td>
<td>conclusion 1</td>
</tr>
<tr>
<td>6</td>
<td>How can gamification be implemented in a business process/software development process?</td>
<td>conclusion 3</td>
</tr>
<tr>
<td>7</td>
<td>Have you worked on projects in which, after delivery, the team was still involved in product support? How can gamification be applied in this setting?</td>
<td>conclusion 3</td>
</tr>
<tr>
<td>8</td>
<td>How can gamification be implemented in the different stages of SDLC?</td>
<td>conclusion 3</td>
</tr>
<tr>
<td>9</td>
<td>Are there significant difference in game elements? Which elements could support the SDLC more and which less?</td>
<td>conclusion 2 and 3</td>
</tr>
<tr>
<td>10</td>
<td>What are today’s best practices of gamification (in software engineering)?</td>
<td>conclusion 2 and 3</td>
</tr>
<tr>
<td>11</td>
<td>What are the pitfalls of gamification?</td>
<td>conclusion 1</td>
</tr>
<tr>
<td>12</td>
<td>How would you tackle the implementation of gamification in the SDLC?</td>
<td>conclusion 2 and 3</td>
</tr>
<tr>
<td>13</td>
<td>Do you have any quantifiable data on the effects of gamification?</td>
<td>conclusion 1</td>
</tr>
<tr>
<td>14</td>
<td>Can you confirm that gamification adds to engagement?</td>
<td>conclusion 1</td>
</tr>
</tbody>
</table>