TÉCNICO LISBOA

Using Gamification to Increase Scrum Adoption

Sofia Baptista Cerqueira Estêvão Modesto

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Miguel Leitão Bignolas Mira da Silva

Prof. Daniel Jorge Viegas Gonçalves

Examination Committee

Chairperson: Prof. José Manuel da Costa Alves Marques

Supervisor: Prof. Miguel Leitão Bignolas Mira da Silva

Member of the Committee: Prof. Leonel Caseiro Morgado

October 2016

Acknowledgments

Gostaria de deixar aqui os meus agradecimentos a todos aqueles que me apoiaram ao longo do curso, e que contribuíram para a escrita deste documento, para a entrega da dissertação e para a conclusão do Mestrado.

Em primeiro lugar gostaria de agradecer aos meus orientadores, Professor Miguel Mira da Silva e Professor Daniel Gonçalves, por me terem guiado ao longo de todo este projeto, transmitindo-me os conhecimentos necessários, e proporcionando as oportunidades certas que me levaram a conseguir entregar esta dissertação e a aprender tanto ao longo deste processo.

Ao responsável pela empresa onde estive a trabalhar durante a tese, por me ter recebido de forma excecional na empresa, e se ter disponibilizado sempre para me ajudar na realização deste projeto.

Aos alunos de MEIC que estão a fazer a tese com o Professor Miguel Mira da Silva em 2016, por se terem disponibilizado para participar na demonstração desta proposta.

Ao meu namorado, Frederico Franco, por me ter apoiado incondicionalmente em todas as fases da minha vida, incluindo esta. Apoiando-me sempre nos momentos menos bons, e fazendo com que estes parecessem melhores, e apresentando-me sempre possíveis soluções para todos os problemas que encontrei.

Aos meus pais, por me terem dado a oportunidade de tirar este curso, e por basicamente, ao longo de toda a vida me terem educado para ser aquilo que sou hoje e conseguir passar por mais esta fase da minha vida, a faculdade e a entrega da dissertação.

Ao meu tio André um grande obrigado por toda a ajuda que me deu.

Ao resto da minha família, em particular, à minha avó Natália, avô Zé, tia Carolina e avó Salete, por todo o apoio que me deram para que eu seguisse sempre em frente.

À Fátima Santos, por me apoiar, tentando sempre mostrar-me o lado positivo.

E por fim, mas não menos importante, ao meu melhor amigo, Oscar, por ter estado sempre presente e me apoiar incondicionalmente estando sempre ao meu lado.

Resumo

A resistência dos programadores para seguirem os processos de trabalho da sua empresa, tal como o Scrum, é um grande desafio para as empresas de IT. Esta resistência pode levar a diversos problemas, como as falhas no cumprimento dos prazo e projetos que não vão de encontro às necessidades dos clientes ou com dificuldades de se adaptarem a novos requisitos, o que leva a uma possível perda de clientes e aumento dos custos para a empresa. O Scrum permite resolver os problemas acima referidos, e por isso é fundamental que este seja adotado pelos programadores.

Neste documento, propõe-se e testa-se uma solução gamificada, que pretende mudar o comportamento dos programadores, reduzindo a sua resistência à mudança. Uma análise de literatura é feita de modo a validar o problema. E de seguida é apresentada uma solução que pretende motivar os programadores a utilizarem o Scrum do Jira, através da utilização de mecanismos de feedback, recompensas, short-term wins, e, diversão.

Design Science Research é a metodologia usada para guiar este trabalho e para estruturar este documento.

Foi efetuada uma simulação numa empresa de IT que usa o Jira como interface para o Scrum, e de seguida foi feita uma demonstração num ambiente académico, com estudantes de informática que estavam a utilizar o Jira para gerir a sua dissertação com Scrum. Por fim, a avaliação foi feita usando a Strategic DSR Evaluation Framework [1], e recorrendo a uma análise da simulação efetuada e dos dados recolhidos na demonstração.

Palavras-Chave: Scrum, programadores, resistência, gamificação, mudança de comportamento, design science research.

Abstract

Programmers' resistance to follow work processes, including Scrum is a huge challenge for IT companies. This can lead to failure to meet deadlines, projects that do not meet customer needs, projects with difficulties to adapt to change due to new requirements, increasing costs for projects, and, some loss of customers. Using Scrum can be helpful because it promotes teamwork, uses incremental design, reduces costs, and makes it easier to meet customer needs due to the small increments.

In this document, a gamified solution is proposed and tested, in order to change programmers' behavior. A comprehensive search of available literature was performed and analyzed to validate the problem. The proposed solution aims to provide programmers some sense of progression, feedback, rewards, short-term wins, and, fun, if they use Scrum (by using Jira).

Design Science Research Methodology was used as a guide for this work and to write this document.

The simulation used to test our solution was performed in an IT company, that uses Jira as a tool for Scrum, and the demonstration was performed in an academic environment with IT Master Students who are using Jira and managing their thesis with Scrum. The evaluation was performed using Strategic DSR Evaluation Framework [1] by Pries-Heje et al. and by analyzing if the proposed solution achieved the proposed goals, based on the simulation and the data gathered from the demonstration.

Keywords: Scrum, programmers, resistance, gamification, behavior change, design science research.

Table of Contents

List of Figures

List of Tables

List of Acronyms

DSRM	Design Science Research Methodology
IT	Information Technologies
IS	Information Systems
UAE	United Arab Emirates
SDT	Self Determination Theory

1. Introduction

Programmers' resistance to use their organization's work processes, such as Scrum [2] (defined in Section 4.1) is a huge challenge for IT companies [3][4]. This lack of involvement by programmers can lead to various problems that emerge for not using Scrum [2][3][5][6], such as failure to meet deadlines (because project progression is not being controlled), projects that do not meet customer needs (because meetings with customers and respective Sprint meetings are not performed) and projects with difficulty to adapt to new changes in requirements (because incorporating small increments in needs from customers are not possible). Consequently, this increases project costs for both the company and its customers, and also leads to some loss of customers, that are unhappy about delays.

Using Scrum methodology can be helpful because [2][3][5][6] it promotes teamwork, uses incremental design (which leads to meeting customer needs), reduces costs for both the client and the company, and because of the small increments, always make a "last increment" available to be delivered which makes it highly improbable that a deadline is not met. Most interestingly, this, brings no extra costs to the company.

Despite all these clear advantages, some programmers resist to use this methodology [2], because meetings are time consuming, mandatory daily reports are also time consuming, and because people, in general, are resistant to change [7].

Since "games provide motivation to succeed and reduce the sting of failure" [8], this document proposes, demonstrates and evaluates a solution to improve programmers' compliance to Scrum. This solution aims to motivate programmers by creating positive feelings, allowing them to have fun, and by encouraging programmers to be more engaged with their work. Engagement can be the competitive advantage of a company, and, game-design techniques provide the means to achieve it [9]. A gamified solution is proposed to change programmers' behavior, to reduce their resistance to using Jira, a tool that can be used to help programmers follow a Scrum methodology with an easy and intuitive interface, described in Section 5.2.1. This solution aims to provide programmers with adequate feedback, rewards, and fun, when they perform certain behaviors. **The objective of this thesis is to analyze the impact of gamification on Scrum adoption by programmers. It was done by comparing programmers who were using Scrum and were being influenced by the gamified solution developed for this thesis and programmers who were using Scrum but were not being influenced the gamified solution.** To validate our findings, we verified if programmers followed several target behaviors such as, going to every daily meeting, creating a certain number of stores and completing them.

Design Science Research Methodology (DSRM) was used as a guide for this work, and as a model for this thesis' report. In the first half of this report, after this brief introduction, the research problem is defined in Section 2 with an introduction to Scrum, which is the motivation for this research problem. In Section 4, the research methodology used is presented in detail, and in Section 4 "Related Work" a

description of the Scrum is presented with some possible causes of adoption problems and of the concepts proposed as the solution for this thesis problem, such as Change Management and Gamification.

In the second half of this report, a detailed explanation of the research solution is presented in Section 5, including its objectives, its description, the game elements used and a description of the architecture and development of the prototype. Section 6, provides details on how this solution was demonstrated, and Section 7 explains how this work was evaluated, based on Strategic DSR Evaluation Framework [1] by Pries-Heje et al. and includes analysis of the simulation and how well the demonstration validated or not that the proposed solution can solve the proposed goals. Section 8 provides details on how this work and its results were communicated to the scientific community. Finally, the last section, Section 9, presents the conclusions gathered from this thesis.

2. Problem

This section presents briefly the software development work process called Scrum, followed by the problem that motivated this research.

2.1. Problem Description

Scrum is an iterative software development methodology, in which teams collaborate to develop projects [1][2]. This methodology consists of diving project into small stories which need to be completed during the sprint in which they are assigned [10]. In order to make Scrum work, the team should have a product owner, a development team and a Scrum Master [10]. They should work together and attend several mandatory meetings in order to complete stories within the estimated time, and to produce deliverable increments within the estimate time [10], which are deliverable to the customer and allows the team to include new requirements of the customer. A complete description of Scrum is presented in Section 4.1 of this document.

In order to allow readers to understand the importance of addressing this problem, a complete description of the problem is presented in this section, with a simplified version in Figure 1.

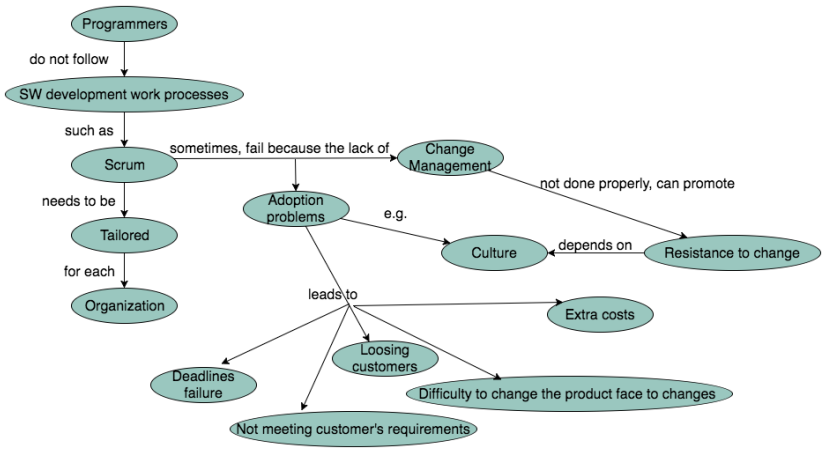


Figure 1 - Research problem.

Since programmers do not normally consider software development as a fun activity [3], the main focus of the research problem of this thesis is that most of the programmers do not follow the software development work processes, such as Scrum, consistently in their work.

“Software projects produce artifacts with an un-satisfactory quality, or exceed budgets in terms of time and cost. This occurs not only because software engineering technologies and methodologies still require further investigations, but also because of human factors” [11].

1 http://www.agilemodeling.com/essays/agileSoftwareDevelopment.htm

Scrum is an incremental and iterative methodology for managing project development. In order to enable this methodology to work, all members of the team need to work together to reach a common goal and the team's company needs to be able to adapt this methodology to the company's culture.

This methodology involves the creation of roles, artifacts and events during the project development, that are explained in Section 4.1. Scrum events, in particular, Daily Meeting (a mandatory daily 15 minute meeting) is considered by many of the individuals involved as a waste of time and only causing delays in their work, even though it only takes very little time to complete [2][12].

Scrum helps teams develop a product with quality, taking into account what the customer wants, with the possibility of an easy adaptation of the product according to changes that the client wants (because it is iterative), however, some individuals, particularly, programmers, resist to use it [7][4][3].

Agile is used by many IT companies, and it is very important to understand, why programmers resist to adopt it. The main reason, besides resistance to change, is them not understanding the need for organizational change [2].

The problem addressed in this thesis is the **poor adoption of Scrum methodology** (or of tools to use that methodology, such as, Jira) **in programmers' daily work routine**. Since programmers do not use the existing methodology, some problems naturally emerge, these include difficulty in following the state of the projects, difficulty in meeting customers' needs to change specific elements, missing deadlines, missing deliveries and lack of documentation, all of which clearly lead to an increase in the company's costs.

In order to lead people to accept the change to Scrum methodologies, proper change management techniques need to be used to deal with the resistance that most people have with change [7]. However, during the literature research we have found a mapping between gamification and change management [13], presented in Section 4.3.5, Table 3, page 22, and for this reason we chose to adopt a strategy to deal with this adoption problem using gamification techniques.

The main focus of this thesis is to increase the adoption of Scrum, which can be measured by several metrics such as, meeting attendance or the number of stories created or completed.

To achieve this, innovative ways were used to make programmers interested and engaged with Jira (an interface for Scrum, explained in Section 5.2.1, in page 28 of this document). This was performed using gamification, which is a technique that uses game elements and applies them to the work environment, in order to make people have fun at work, feel engaged with it, and improve their performance. "Gamification, if applied to software development, may provide several advantages. First, because of its rewarding mechanisms, it may motivate developers to learn new technologies and increase their productivity. Second, it may improve the quality of their work if adopted to encourage best practices. From a management perspective, it may be used as an input to give economic incentives and to support the evaluation of employees as well as of teams" [11].

3. Research Methodology

DSRM was the methodology chosen for approaching this thesis. The purpose of this section is to provide readers with a clear definition of the method in which this work was performed.

Design Science is a problem-solving paradigm that "seeks to create innovations that define the ideas, practices, technical capabilities, and products through which the analysis, design, implementation, management, and use of information systems can be effectively and efficiently accomplished" [14][15]. Its purpose is to solve problems and it is often used in IT organizations or in the IS development.

Design research builds a "to-be" conception and then helps the researchers build a system according to the defined model taking into account its restrictions and limitations [16]. Design science addresses research through the building of artifacts and its evaluation by the degree in which they meet the identified business needs [16]. DSRM is a methodology where an artifact should be created to explain the problem and to create a solution, as explained in Figure 2.

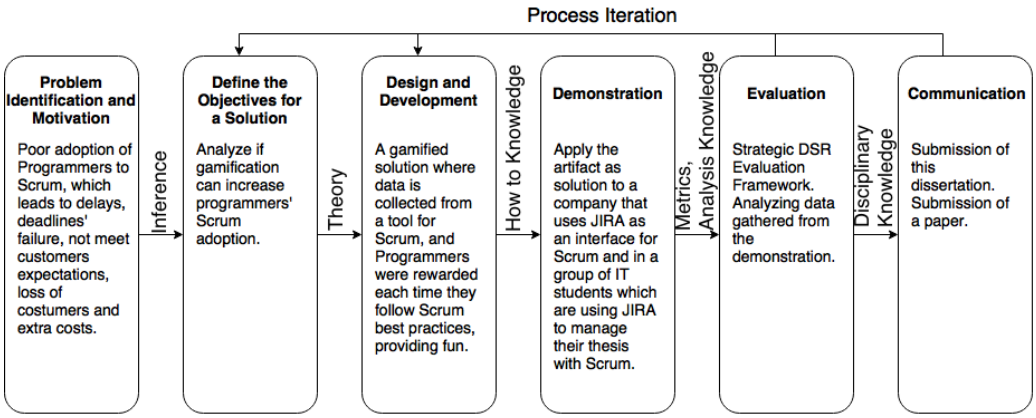


Figure 2 - DSRM applied to this thesis' context, based on [17].

This methodology is based on an iterative process with six steps [17], which are described below along with a description of how DSRM was applied to this research.

- Problem identification and motivation: Definition of the specific research problem and justification of the value and importance of a solution. The problem of this research is programmers not using Scrum in Jira in their daily work. This is presented in detail in Section 2 along with the importance of a solution and this was used to develop an artifact to provide a coherent solution.
- Define the objectives for a solution: Consists in inferring the objectives of a solution from the problem specification. The objective of this thesis is briefly presented in Section 2, which is to analyze the impact of gamification on Scrum adoption by programmers, and is described in detail in Section 5.
- Design and development: Determination of the artifact's functionality and architecture and creation of the research's artifact (any designed object in which a research contribution is embedded in the design). Based on the problem defined in Section 2 and in the literature

review performed (Section 4), a solution was designed and developed with the objectives described in Section 5 and with architecture described in Section 5.4.

- Demonstration: Demonstrate the use of the artifact to solve the research problem identified. Our artifact was demonstrated primarily in a simulation (Section 6.1) performed in an IT company with an artificial project and secondly in a demonstration (Section 6.2) in a real project with IT Master Students.
- Evaluation: Observation/Measurement of how well the artifact supports a solution to the problem. This measurement was performed based on the data gathered from the simulation and from the demonstration. This step is explained in Section 7 of this document. At the end of this activity researchers can decide whether to iterate back to step 3 to improve effectiveness or to continue on to step 6, communication. It was decided to not iterate back to step 3 because of time constraints. After this activity we proceed to step 6, communication.
- Communication: Communication of the problem, its importance, the artifact, its utility and novelty, the rigor of its design, and its effectiveness to researchers and relevant audiences.
  The communication was performed by submitting a paper to ICSE 2017 and by submitting this dissertation report, as described in Section 8.

4. Related Work

This section presents the results of the literature review performed regarding this thesis' context, starting with a presentation of the software development work process called Scrum, in Section 4.1 and some common Scrum adoption problems.

Section 4.2, presents Resistance to Change and how to avoid it with Change Management techniques, in order to make programmers adopt Scrum. Section 4.2.2 presents the study of Geert Hofstede about Culture, a useful reference on Resistance to Change and Scrum adoption problems, which explains why Scrum is not being used properly.

Finally, Section 4.3 presents Gamification, which was the technique used to solve the research problem. The following figure presents how this section is organized.

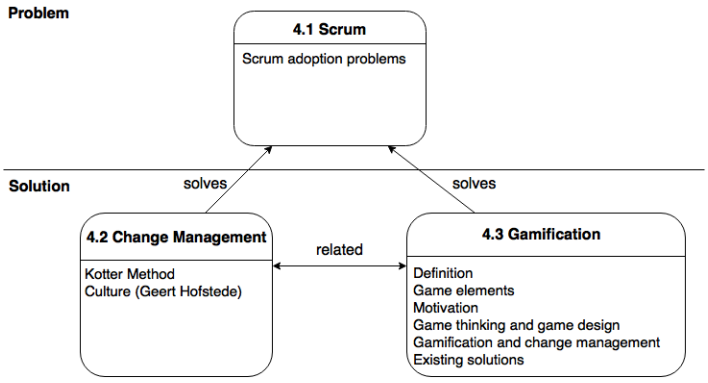


Figure 3 – Related work schematic organization

4.1. Scrum

Scrum is an agile software development methodology, which is composed of an iterative, incremental and collaborative process for teams to work with complex and unpredictable software development [2][2]. Scrum is usually used to deal with large scale projects [2], because it has the ability to include new requirements of the client with a relative easiness.

Scrum helps teams to produce products with quality, respecting time and budget constraints [3][5]. A key principle of Scrum is its recognition that during production processes, customers can change their minds about what they want, and the team can include those new requirements in the following increment. But, despite of all these clear advantages, there are several adoption problems when companies start to use it, as is presented in Section 4.1.1.

2 http://www.agilemodeling.com/essays/agileSoftwareDevelopment.htm

To explain Scrum in detail, one first needs to focus on the Roles, Events and Artifacts that it uses, as represented in Figure 4.

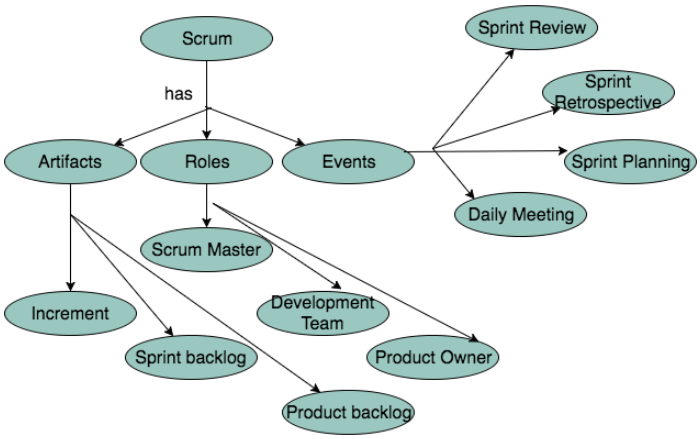


Figure 4 - Scrum roles, events and artifacts (original).

The first role is the product owner, also known as the customer representative. He is responsible for the elaboration of the product [10], writing of the stories, ranking, prioritization, and adding to the product backlog. The second role is the development team, they are responsible for developing increments to the product at the end of each sprint. The third and the last role is the Scrum Master, responsible for removing impediments so the team can deliver the product. The Scrum Master also guides the team through the correct use of Scrum [10], and is not a team leader or a project manager.

Figure 5 presents Scrum events and artifacts, intended to minimize unexpected meetings.

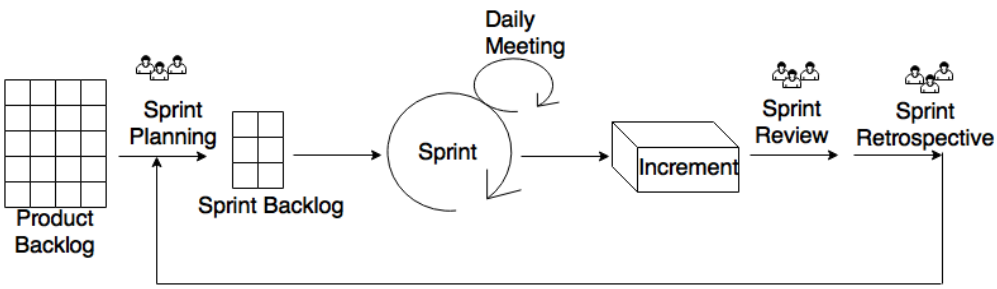


Figure 5 - Scrum events and artifacts (adapted from [61]).

A Sprint is an effort that is restricted to a specific duration (usually 15 days). Each sprint starts with a sprint planning event and ends with a sprint review and a sprint retrospective. Sprint planning is an event that occurs at the beginning of a sprint, where the team communicate the work that will be done during that sprint. There is another event called daily meeting, which is a daily event where all members of the development team come prepared, and should be a short 15 minutes meeting. At the sprint review (at the end of the Sprint), the team reviews the work that was completed and the planned work that was not completed and presents the completed work to the stakeholders. At the sprint retrospective (at the end of the Sprint), the team reflects on the past sprint and identifies and agrees continuous process improvement actions.

Finally, Scrum has artifacts such as the product backlog, a repository of everything that needs to be done during the development of the project, the sprint backlog, which contains the tasks selected by the team to be done in the next Sprint [10], and, the increment, which is the sum of all the product backlog items completed during a sprint and all previous sprints [18].

4.1.1. Scrum Adoption Problems

Agile methodologies, particularly, Scrum, have a lot of advantages for both the companies and the customers, but, unfortunately, they often fail. Some causes were collected from articles in the following websites dedicated to scrum implementation [3,4,5], a UAE case study [2], and a Pakistan Industry case study [12]. The following list is an exemplificative list of possible causes for Scrum adoption problems and not a representation of the causes for this specific research problem. Some causes for Scrum adoption problems are:

- Lack of support from Senior Leadership;
- Existing organizational structures and hierarchies;
- Distributed teams that don't behave as one team;
- Company's culture;
- No organizational learning (when feedback from retrospectives is lost);
- Lack of environment of trust (when people are afraid of negative feedback);
- Product owners with lack of Scrum knowledge;
- Not preparing the organization around a Scrum Project.
- Missing the Scrum Master role;
- The absence of a pilot project: The pilot project is essential to evaluate how the team environment will be able to move from the previous method to a new method. Many organizations do it, such as Yahoo! [6], [19], Amazon [20], and Microsoft [21];
- Scrum Implementation;
- Current work pressure: The already existing firm deadlines lead to the team considering daily Scrum meetings as a waste of time;
- Upper management concerns: The upper management was not easily convinced to invest in a new method.
- Implementing Scrum without mutual agreement;
- Not Empowering Teams: Empowerment enhances decision making and problem solving capabilities that help in Scrum implementation and adoption.
- Wrong prioritization of product features;
- Stressful working environment.

In addition to these items, the resistance that most people have to change, in general, can be a barrier to Scrum adoption too [7].

---

3 http://www.payton-consulting.com/5-common-problems-scrum-adoption/

4 https://www.scrumalliance.org/community/articles/2014/march/tips-for-scrum-adoption

5 http://www.infoq.com/news/2008/12/scrum-adoption-challenges

From the exemplificative problems explained above, one can understand the urgency to adopt mechanisms to engage people in using Scrum. The following Sections 4.2 and 4.3, present two techniques (Change Management and Gamification) which can be used to solve or minimize these problems, for e.g. lack of environment of trust, stressful work environment and current work pressure can be solved with gamification mechanisms such as positive feedback or rewards, and with change management techniques such as communication between all parties involved.

4.2. Change Management

Change management was defined by Moran and Brightman [22] as “the process of continually renewing an organization’s direction, structure, and capabilities to serve the ever-changing needs of external and internal customers”.

People usually respond to the call for change with the affirmation that they are fine [7]. It is important to understand why people resist to change, and do everything possible to avoid it.

As explained in Section 4.3, motivation is an important objective of a gamified system, and, the desired direction can sometimes be achieved by using a system with constant feedback and sense of progression, because people need to understand the direction to follow [7].

In order to make Scrum work, companies that are facing the problem identified in this document need to adopt a management strategy that promotes teamwork where knowledge is shared throughout the organization. Several methods were identified in the literature research, but, the selected one was the John P. Kotter method since it was the one that presented the most similarities with the steps to implement a successful gamified system, described in Section 4.3.5, Table 3, page 22 of this document.

To conduct a successful organizational change, three aspects have to be considered: Strategy, Structure and Culture [23] as presented in Figure 6. The direction has to be defined (strategy), according to the organizational structure, taking into account the existing organizational culture [13].

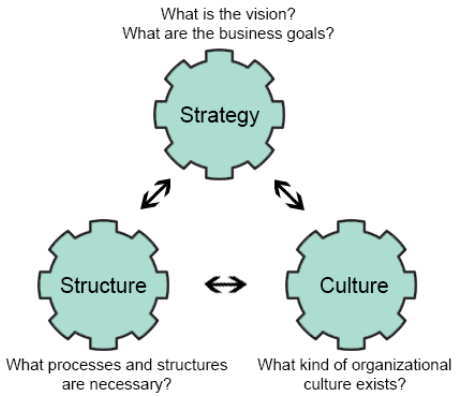


Figure 6 - Strategy-Structure-Culture Triangle (adapted from [23]).

Change Management techniques are important to lead the company to the last phase of the Change Curve (composed by Shock, Denial, Anger, Depression, Acceptance, Integration [24], presented in Figure 7).

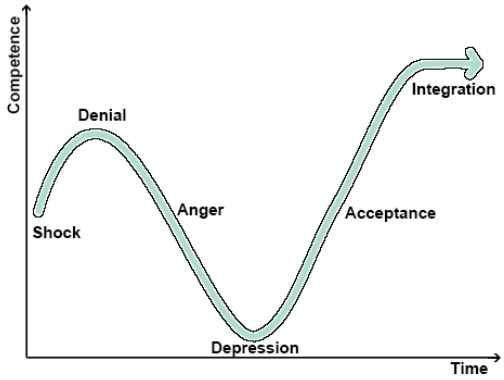


Figure 7 - Change Curve (adapted from [24]).

If change is not dealt properly, employees could be stuck in phase 1, 2, 3 or 4 of the Change Curve. It is important to notice the connection between these steps and gamification steps described later in this document, in Section 4.3.5, Table 3, page 22 of this document.

4.2.1. John P. Kotter Method

In successful change efforts there are eight important steps [25].

1. Increase Urgency

The first step consists in showing people why is change is needed. To achieve this, leaders should provide employees with concrete evidences to ensure everyone is working towards the same goal.

An example of urgency is shown in the case study entitled Gloves on the Boardroom Table by Jon Stegner [7], where a student undertook a project for a company to find how many types of gloves they use in their daily work. She found that they use many kinds of gloves, all with different prices and they all look the same. So they created a scenario where a sample of each glove was tagged with its price and put in their boardroom. All the employees were called to look at the gloves, and were shocked to find out they all had similar gloves but with very different prices. Now, they understood the urgency to change.

2. Build the Guiding Team

The second step consists in creating a team, with leadership capacity and credibility, to guide organizational change.

3. Get the Vision Right

The team created in step 2 should communicate the objectives of the desired change, explaining what need to be changed and what is to stay unaltered.

4. Communicate for Buy-In

The goal of this step is to get as many people as possible acting to achieve the proposed goals, by using proper language to make the goals easy to understand to each person.

5. Empower Action

When people start wanting to change, all obstacles in their path should be removed. It is very important that leaders stay open to communicate with their colleagues and to clarify any doubts they may have, in order to achieve the main goal of this step, which is to make everyone feel they are able to change.

6. Create Short-Term Wins

Short-term wins help to provide feedback to those already working on a particular change, and to attract those who are not yet convinced. Feedback and rewards promote optimism, builds self-confidence and belief in change. As described in Section 4.3, this is used in a gamified system.

Katherine Kane says [7], “managing is fundamentally a people issue. It’s about motivation and influencing behavior, about breaking old habits and attitudes, and about creating an environment that’s conductive to embracing the new. Be less preoccupied with large-scale transformation, and focus instead on small improvements”.

7. Don't Let Up

In this step, although many of the proposed goals have already been achieved, some have not. So urgency should be kept up, without forcing people too much.

8. Make Change Stick

The last step consists in creating a new organizational culture. Culture reflects the predominant behaviors of a group/organization. Culture can be a barrier or an enabler to a change effort, so, culture must be tailored to support the strategy.

The need for change has never been greater than in the current continuously evolving business environment [22]. Management should be used to "minimize operational disruption while making change, understand the change and the associated implementation risk and align individual behavior and skills with the change" [13]. So, change management skills are highly required to complete these eight steps.

Both change management techniques and gamification techniques, with positive reinforcements, while taking into account the organizational culture, can increase the probability of success.

4.2.2. Culture

Culture has a huge influence to the success or not, of a change management effort. This section is highlights a study about culture made by Geert Hofstede.

Geert Hofstede is a Dutch management researcher that gathered data from 100.000 IBM employees to determine values in which culture vary. From this study he created the "Hofstede Model" which is explained in this section. This model is helpful in understanding the organizational culture of any organization.

At an organizational level, culture can be defined as "the collective programming of the minds of group members by which one group distinguishes itself from others" [6]. The "Hofstede Model" consists of six dimensions of organizational culture, as explained in Table 1.

6 http://geert-hofstede.com/tl_files/art%20organisational%20culture%20perspective.pdf

Table 1 – Culture Dimensions by Geert Hofstede (based on [7] and [8]).

Dimensions	Description
1. Power-distance index	According to Hofstede, "power distance is the extent to which the less powerful members of organizations and institutions accept and expect that power is distributed unequally.". Low power-distance means that individuals accept that relations are democratic and everybody is considered equal, in terms of power. High power-distance means that individuals accept inequality, such as the existence of a hierarchy.
2. Individualism vs. Collectivism	This dimension refers to how people define themselves and their relationship to others. Cultures that are individualistic place importance on personal goals, believing that themselves are the most important. In collectivist cultures, the goals of the group are valued over the individual ones.
3. Uncertainty-Avoidance	Hofstede defines this dimension as "a society's tolerance for uncertainty and ambiguity". This measures the way individuals deal with unknown situations, and the stress of change. Cultures with an high index are less tolerant of change, in contrast to the ones that score low, which are more open to change and individuals tend to "think outside the box".
4. Masculinity vs. Femininity	This dimension measures the level of importance a culture places on masculine values (competition, ambitious) as well as feminine values (relationship building) such as an emphasis on human relationships.
5. Long-term vs. Short-term Orientation	This dimension describes a society's time horizon. Short-term oriented cultures look for immediate gratification and are less willing to sacrifice. Long-term oriented cultures, looks to future rather than present or past, and usually do anything to achieve a goal.
6. Indulgence vs. Restraint	This dimension measures a culture's ability to satisfy the immediate needs and personal desires of its members. Those that value restraint their behavior based on social rules and the others do whatever they want without thinking about the surrounding environment.

7 http://news.telelangue.com/en/2011/09/cultural-theory

8 http://geert-hofstede.com/tl_files/art%20organisational%20culture%20perspective.pdf

4.3. Gamification

Poor engagement of employees in work processes is a huge challenge to companies. One of the reasons for this problem, pointed in the literature [4][9], is low employee engagement to work, which may be caused by lack of fun, and the feeling that work is boring, repetitive and stressful.

Gamification is an effective solution to this problem because it can motivate and engage users in several contexts, such as work [26]. This technique consists in applying game mechanisms and visual components such as points, badges and leaderboards, in order to achieve fun appealing to some "high level" dynamics such as emotions, constraints, narrative, progression or relationships [9] while providing a new and innovative user experience.

This technique has a great adoption rate because it is based on games, and, everybody on the planet, since the earliest civilizations, is familiar with playing games [9]. According to Werbach and Hunter [9], there are references to the application of game elements to systems since 1980. Professor Richard Bartle made reference to "turning something not a game into a game" [9]. The first use of gamification as it is known today, was in 2003, when Nick Pelling created game-like interfaces for electronic devices. Since then, game designers and researchers started to think about the potential of games in other contexts. In 2010, the term "gamification" became widely adopted [9].

4.3.1. Definition

To understand the concept of gamification, one first needs to understand what is a game. A game can be defined as "an activity that one engages in for amusement" [27], or, more concretely, by having goals, rules to follow, a feedback system, and voluntary participation [10]. Katie Salen and Eric Zimmerman define a game as "A system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome" [28].

Having understood that, some definitions of gamification can be presented. Gamification is defined by Werbach and Hunter as "the use of game elements and game-design techniques in non-game contexts" [9] as presented in Figure 8.

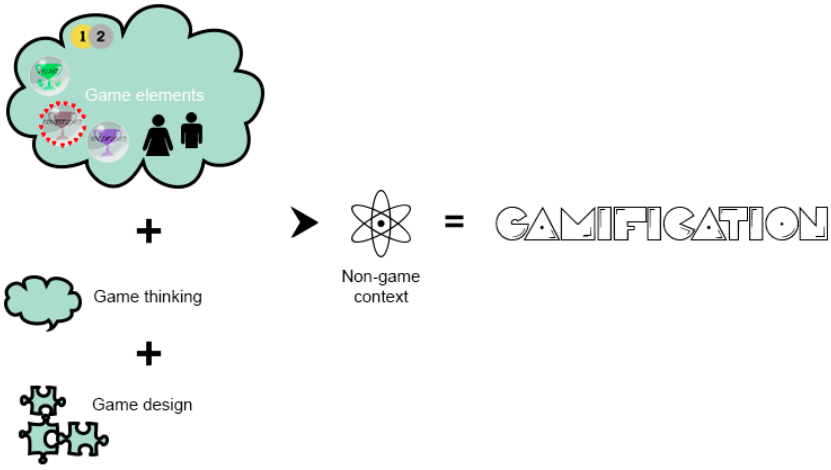


Figure 8 - Gamification based on definition of book [9].

Gabe Zichermman, defines gamification as “the process of using game thinking and mechanics to engage audiences and solve problems” [29]. Amy Jo Kim defines gamification as “using game techniques to make activities more engaging and fun” [30].

As can be understood by the definitions above, to engage people is one of the most important aspects of a gamified system. Engage means “to gain a person’s attention and to involve him/her in the process you created” [8].

Gamification is also used to enrich “products, services, and information systems with game-design elements in order to positively influence motivation, productivity, and behavior of users” [31][32], and it is one of the most important trends in technology, because it can “potentially be applied to any industry and almost anything to create fun and engaging experiences” [9][33].

There are three types of gamification identified by Werbach and Hunter [9]: internal, external and behavior change, as presented in Figure 9. This thesis only focuses on the first and the third type because those were the ones most relevant to the research question.

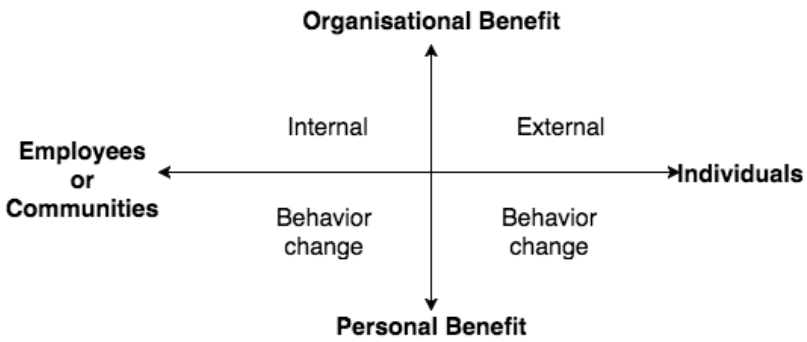


Figure 9 – Types of gamification adapted from [9].

When a company uses internal gamification (also called enterprise gamification), it wants to improve productivity within the organization (for example, foster innovation, enhance camaraderie, positive business results) [9].

When behavior-change gamification is used, it seeks to create new habits among a group of people, which are beneficial to them [9].

9 http://gamification.org/wiki/Game_Mechanics#Achievements

4.3.2. Game Elements

Gamification uses game elements. Game elements are the objects, relationships and rules that exist in a game. For checkers, for example, “they include the pieces and the notion of capturing pieces by jumping” [9]. Game elements can be divided in four categories, Dynamics, Mechanics, Components and Aesthetics, as presented in Figure 10.

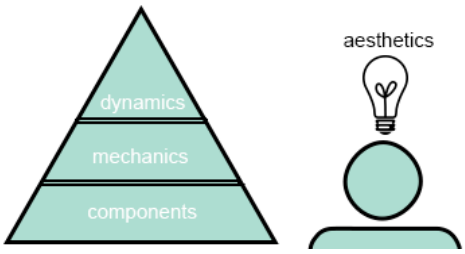


Figure 10 - Game elements, adapted from [9][34].

Dynamics

Dynamics are all the elements at a higher level of abstraction [9]. They have a huge influence on the game, but, they are not directly visible. Some examples are [9]:

- Constraints (limitations);
- Emotions (feelings such as happiness);
- Narrative (a form of teaching lessons);
- Progression (player’s evolution);
- Relationships (social interactions).

Mechanics

Mechanics are used to create engagement, and, can be used to achieve elements of dynamics. “Mechanics describes the particular components of the game, at the level of data representation and algorithms” [34]. “Mechanics can be used to drive almost any user behavior. They have the potential to tap into the full range of human emotions and motivate a wide range of behaviors” [10][33], they include elements such as [9]:

- Challenges (puzzles or other tasks that require effort to solve);
- Chance (elements of randomness);
- Cooperation (various players working together towards the same goal);
- Competition (one player wins and the other looses);
- Feedback (information about the “rightness” or “wrongness” of an action);
- Resource acquisition (obtaining items);
- Rewards (obtained for performing tasks);
- Win states (objectives that makes one player the winner).

10 http://mashable.com/2010/07/13/game-mechanics-business/

Components

Components are the visible and concrete forms that the higher levels of the pyramid takes [9], they include elements such as [9]:

- Achievements (objectives completed);
- Avatars (visual representation of players. The most customizable, the most influence they have on players);
- Badges (visual representation of achievements);
- Collections (set of items gained/collected);
- Leaderboards (visual representation of player's progression in comparison to others);
- Levels (defined steps in player's progression);
- Points (earned by completing tasks);
- Quests (predefined challenges with objectives and rewards);
- Virtual goods (game assets with some value to players);
- Teams (groups that work together towards a common goal).

Aesthetics

The fourth category is Aesthetics, which "describes the desirable emotional responses evoked in the player, when he or she interacts with the game" [34] and it has the potential to create fun.

All these elements can be applied, not only to games, but also to gamified systems.

In addition to all those components, a phenomenon identified by Mihaly Csikszentmihalyi called "Flow" [35] needs to be understood, although this term was not originally defined to be applied to games. Flow is what happens when someone become so involved and engaged that one loses track of time. "Flow" happens between anxiety and boredom, as presented in Figure 11.

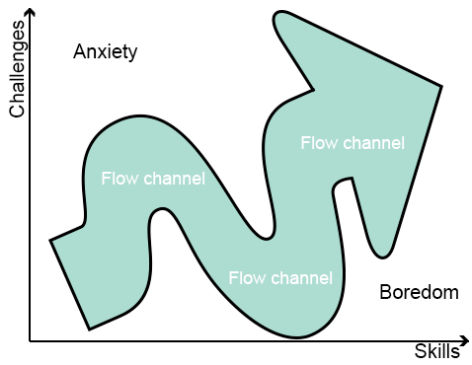


Figure 11 - How to achieve flow, adapted from [9].

To achieve "flow" players must be concentrated, should have understood the goals, should receive immediate feedback, should be involved without effort, should feel in control, and, consequently, concern for self disappears (e.g. Forgetting to eat) leading players to lose sense of time.

Gamification has the potential to engage people, as game designer Raph Koster says “With games, learning is the drug” [36].

Gamification, allows the player to fail with minimal consequences because of the inclusion of game elements, which encourages exploration and curiosity [8] [37].

To conclude, it is important to distinguish gamification from “serious games”, games that help professionals in training for their daily jobs. And also to distinguish it from “alternate reality games” [8], games designed to combine real life and digital game play elements.

4.3.3. Motivation

In order to implement a successful gamified system one should understand game design and some business techniques. It is important to understand that the concept of gamification is deeply connected with motivation.

N. Ahmadi [38], points out the need to integrate results from social and psychological sciences in the software lifecycle. This is where motivation starts to connect with gamification. People feel motivated by well-designed games, so, monetary rewards are not even necessary, because the game itself is a reward. Videogame players invest their resources into acquiring virtual objects that have no tangible value [9].

One important aspect is that games are voluntary. No one can force you to have fun. As James P. Carse said, “Whoever must play, cannot play”. This point can generate a bit of controversy about the application of game elements in the work places, because work is not voluntary.

Motivation is an essential concept that is the basis of gamification. To be motivated is to be moved to do something [9]. There are two types of motivation, intrinsic or extrinsic, and in this solution, both were used. Wanting to do something is called intrinsic motivation because, for the person involved, it lies inside the activity. On the other hand, feeling that you need to do something involves extrinsic motivation, because the motivation lies outside.

Usually, you should provide rewards only for the tasks that players are not intrinsically motivated to perform. When extrinsic incentives are used to motivate people, it often ends up decreasing people’s intrinsic motivation [39].

There are many theories about motivation, such as, Operant Conditioning, Self-Determination Theory (SDT) and Social Learning Theory, which are explained below:

- **Operant Conditioning** - says that players’ motivation is externally driven. It says that they will collect coins, tokens, items, or points because they know they will receive an award [8].
- **SDT** - from Edward Deci et al., it is a theory that explains human motivation to perform a task as being internally driven [40]. SDT suggests that these needs fall into three categories: competence, relatedness, and autonomy. “Competence,” is a need for challenge and a feeling

of mastery [8], “relatedness” involves social connection and desire to interact with others [8]. And finally, “autonomy”, is “the feeling a person has that they are in control of their actions” [8].

- **Social Learning Theory** - created by Robert Bandura 1970 and says that people learn from one another in the context of a social situation through observation.

In parallel to these theories there are several models to follow if you want to motivate players, such as the “ARCS Model” developed by John Keller, and the “Lepper’s Instructional Design Principles”, presented in Table 2.

Table 2 – Models to motivate players [41], [42].

ARCS Model	**Lepper’s Instructional Design Principles**
Attention	Control
Relevance	Challenge
Confidence	Contextualized
Satisfaction	Curiosity

A gamified system has two core processes [9]. The engagement loop is the first, which consists of a loop where players perform a correct action, receives feedback from the system, and this feedback motivates users to keep using the system and performing correct actions, as presented in Figure 12.

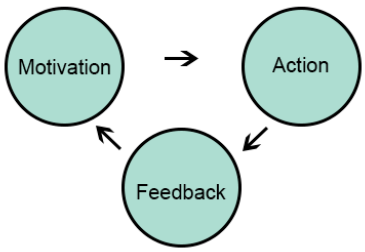


Figure 12 - Engagement loops, adapted from [9].

The second basic process is the progression stairs, presented in Figure 13. This is usually achieved by moving through levels. From the first level to the last, the level of difficulty should escalate, and then slow down [9].

In this solution, a final goal does not exist, since programmers will continue to work in the company for an undefined amount of time, so, this thesis’ “semi-final” goals, for example, end the release or the project.

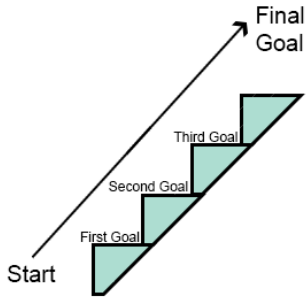


Figure 13 - Progression stairs (original).

4.3.4. Game Thinking and Game Design

"When you're engaged in a game that you care about, you naturally try to succeed. Evolution has wired our brains to be natural game-playing machines. In short, gamers try to win, game designers try to make gamers play" [9]. So, gamification involves game thinking (think like a game designer), which means using all the resources you can master to create an engaging experience that motivates desired behaviors [9], and also using game-design techniques, which are the aspects of games that make them fun, addicting and challenging [9]. "Gamification is a motivational design problem, that can be solved with design thinking and design processes" [42].

Werbach and Hunter suggest a six steps' iterative methodology for designing games [9] which is explained below.

1. **Define Business Objectives**

If a system is being gamified, it should have a reason. We should understand its real goals, and define them carefully, with as much detail as possible along with an explanation of how the company will benefit from them. These goals should be prioritized according to their importance.

2. **Delineate Target Behaviors**

The target behaviors are the ones we want the players to perform, in order to achieve the business objectives defined in the previous item. We should also define metrics to measure the results of performing those behaviors.

3. **Describe Players**

In this step, we should be aware of which types of player our system has, in order to understand what can make them engaged or bored. The system should be designed taking into account different types of players, with different motivations and levels of expertise. These players can also be segmented according to their individual or organizational culture (Section 4.2.2).

4. **Devise Activity Cycles**

Games are not a linear sequence of steps. We should create engagement loops and progression stairs, which have already been explained in this document, in order to keep players motivated. The game should have different activities for inviting new players (novices) to play and to keep masters

motivated, which can be achieved by creating different difficulty levels, and taking into account the different player types and experience level.

5. **Don't Forget the Fun!**

Fun is the main reason to gamify a system, and it should not be forgotten. Fun can be achieved in many different ways, making it difficult to define clearly how to achieve it for a large spectrum of players.

6. **Deploy the appropriate tools**

Finally, the last step of the design requires a strict selection of the tools, dynamics and mechanisms that need to be coded into the system [9].

4.3.5. Gamification and Change Management

Gamification is interconnected and can be viewed as a change management technique that helps in reducing the natural human resistance to change [13]. To increase employees' acceptance of change, gamification or change management techniques already presented in Section 4.2 of this document should be applied. The steps for a successful change management effort along with the steps for a successful game-design are both mapped and presented in Table 3.

Table 3 – Mapping Change Management steps into Gamification steps (based on [13][9]).

Change Management (Kotter)	Gamification (Werbach & Hunter)
Increase Urgency Build the Guiding Team Get the Vision Right	Define Business Objectives Delineate Target Behaviors
Communicate for Buy-In Empower Action	Describe Players
Create Short-Term Wins	Devise Activity Cycles
Don't Let Up Make the Change Stick	Don't Forget the Fun! Deploy the appropriate tools

4.3.6. Existing Solutions

Gamification has been applied in products such as Mozilla Open Badges, Gamify, Badgeville, Userinfuser, Bunchball, Leapfrog, Crowdtwist, Manumatix Bamboo, Big Door, Scvnr and Reputely to increase user's motivation and engagement [4]. Gamification is also being used to motivate people to exercise more with apps like Nike+[11], Fitocracy and Zombies, Run! [8], and even to reduce sedentary behavior by using Piano Stairs [8]. Speed Camera Lottery[12] is a gamified system that intends to reduce the problem of speeding, by reverting fines to a lottery, and giving good drivers tickets to that lottery. Gamification has also been applied to other areas such as medicine[13], hygiene compliance in hospitals [43] and damage control[14].

This section details some of the literature review performed regarding application of gamification.

Gamification Enterprise

Gamification enterprise, refers to gamified workplaces, that are, in other words, "organizations that use gamification to transform some of their work processes into a game-like experience for the employees by applying selected principles of game design and game interaction" [44].

The following examples are tools to gamify workplaces, aiming to improve productivity and to create a positive relation between employees. Dogear is guessing game incorporated within an internal enterprise social system, which aims to make employees aware of their colleagues' interests and increasing their contributions to the system [45]. The last objective is also a goal of IBM's Beehive. Bluegrass explore socialization and collaboration among software engineers, by including them in a 3D virtual world where they can play and learn about each other. Olympus is a tool to explore how employees can present themselves in online meetings, by including game elements in an e-meeting system. "Language Quality Game" is a Microsoft game, that provides enjoyable and addicting review process for employees around the world, awarding them with points per bug found, and allowing them to compete with other regions [9].

The following section presents various scientific studies that have been performed in the area of this research, gamification in information systems development.

Gamification in Information Systems

Many previous works have proposed games as learning tools for software engineering [46]–[51]. Gamification has also been applied to eXtreme Programming [52] and the motivation created by gamified systems was already investigated by Snipes et al. [53]. Bacon et al. [54] investigated the adoption of scoring systems in the context of software development. There are also many examples

11 http://nikeplus.nike.com/plus/

12 http://www.thefuntheory.com/

13 http://www.tid.es/es/Research/Paginas/TIDProjectProfile.aspx?Project=MoviPill

14 http://www.bahnscout.de

where gamification had a great impact on an organization's work processes, in several contexts, like education, enterprise, healthcare, marketing, government and computer programming [9][55].

CollabReview is a web-based reputation system for collaborative source code, proposed by Prause et al [33] in order to improve the quality of source code written in teams, thereby increasing the overall quality of software projects. This system computes an individual score to predict future user behavior and to create peer pressure, based on the number of Javadoc violations in a file. It was verified that each developer's contribution evolved over time, but the system had no impact on improving code quality. It was also noted that a simpler algorithm to calculate scores should have been used, to avoid doubts about the score calculation [33].

Singer and Schneider [51] gamified a version control system, using of social components, "The Teamfeed", to increase the number of developers' commits to repositories. This app uses several elements such as commenting, positive feedback, avatars and leaderboards. Some users liked it, some did not. As expected, one student, tried to game the "game" producing excessive commits, but the overall results were positive, with students increasing the number of commits.

There are some interesting examples of game-like activities that have been used to teach aspects of agile software development [56]. These examples are important for this research, since the problem of this thesis is the poor adoption of programmers to Scrum, which is also an Agile methodology. Agile Technique Hour, was created to explain the difficulties in working concurrently in small iterations [56]. "Scrumhancer" [56] aims to expose the importance of Scrum meetings. "Agile Poker" shows the value of working in pairs and teams [56]. Finally, "An Agile Story" addresses the issue of incorporating a set of requirements into an iteration, and dealing with changing requirements [56].

Scrum Knowsy[15] is an online game that helps people improve their Scrum performance, where players answer questions and compare their results with colleagues or with Scrum experts.

Dubois et al propose a methodology to apply gamification in software development, based on identifying the most adequate gamification mechanisms to be applied, activities to integrate those into the existing software development tools and, finally, evaluation activities to compare developers' performance before and after gamification [11]. Passos et al presented a work where game elements are incorporated in a iterative software development process, transforming the process into a game [3]. Herranz et al presented "Gamiware", a gamification platform to increase employees' motivation and engagement in software initiatives [4]. Unfortunately, Gamiware is not yet integrable in Jira.

Commercial Tools

Some commercial tools to gamify project management have already been created, such as, RedCritter Tracker[16] to gamify agile and PropsToYou[17] to gamify generic project management, but, there are also some add-ons specifically created for Jira, the tool gamified in this work.

15 http://www.scrumknowsy.com/home

16 https://www.redcrittertracker.com/

17 http://www.propstoyou.com/

Jira Hero[18] is an add-on that applies game mechanisms to Jira, which aims to guide Jira users' into using the system, providing a fun and creating an engaging environment in their use of Jira, but, unfortunately, it was archived in 2012.

Jiraffe[19], is a customizable add-on for Jira, which applies game elements such as adventures, points, badges and leaderboards to involve players in a narrative which aims to create a more fun and engaging experience while developing projects. This add-on does not solve this thesis' problem since it is not focused on Agile development.

The last, and most complete tool for Jira that was analyzed, was a commercial tool already integrated with Jira, GetBadges[20]. This is a very complete tool regarding gamification since it works with several components to motivate players. Points and levels give players a sense of progression, increasingly difficult levels challenge players to improve their performance, leaderboards per month, allows users to "start-over", reducing the sting of failure, and let them compete with each other. The monsters' slaying functionality promotes collaboration and fun. However, this platform presents some limitations, such as the poorly explained dashboard and the lack of guidance during the player journey which can reduce the curiosity to start using it.

GetBadges allows integration with Jira through the issue tracker, using a webhook, the same way this thesis' solution was done. This does not solve this thesis' problem because the platform is not optimized for Scrum. This tool only recognizes issues such as stories, tasks and bugs and events such as the creation, deletion, and status change of the issues, while ignoring Scrum events such as meetings. By performing an automatic identification of issues such as features, improvements (or other custom issue types), as tasks, creating several inconsistencies. It is also unable to detect Sprints, which means it cannot gamify Scrum where compliance with each step of the process is relevant.

As can be understood at the end of this section, gamification can be used as a change management technique (Table 6) to improve the adoption of IT work processes [13], and for this reason it was used in this research solution to increase the adoption of Scrum. It is also important to understand that to motivate behaviors for long term, intrinsic motivators should be used [13] and negative achievements should not [9]. The success of a gamified system depends on several factors such as the surrounding environment, involvement of users and player types [55], which proves the importance of tailoring gamification for each organization. Finally, it is important to understand that game elements can have addicting effects, so, after introducing them, if they are removed, positive actions that existed before, could disappear [57].

Despite the existence of a lot of gamified systems, some were created to gamify generic project management and increase users' engagement in software development (such as Gamiware and PropsToYou), some of them in the context of this thesis' problem, gamifying Scrum to increase its

18 https://marketplace.atlassian.com/archive/com.madgnome.jira.plugins.jirachievements

19 https://marketplace.atlassian.com/plugins/com.bugpotion.jiraffe/server/overview

20 http://www.getbadges.io/

adoption by users (such as RedCritter), but none of them is integrated with Jira. As already mentioned, some of the systems studied are even integrated with Jira (such as Jira Hero, Jiraffe, GetBadges), but, none of them solve this research problem because they are not adapted to deal with Scrum. From the problem already explained and the inexistence of a solution to solve it, this research was motivated to continue and the research solution is presented in the next section, which aims to solve this research problem, by using gamification as a change management technique.

5. Solution

This section contains the objectives of this thesis and of our solution based on the problem already defined in Section 2 of this document, followed by a description of our solution, which contains an explanation of Jira, followed by the definition of the target behaviors and metrics, game elements and mechanisms used to reward those behaviors. Lastly, a concrete description of the gamified solution developed is presented. This section focuses on the DSRM's step 2.

5.1. Objectives

The main objective of this thesis **is to analyze if gamification can have a positive impact on programmers' adoption of Scrum.** To achieve it, programmers need to change their behavior and reduce their resistance to change their work methodology.

This document proposes, develops and tests a system that provides constant feedback on whether programmers follow Scrum best practices or not, using a gamified system to make the using of Scrum fun and engaging. These Scrum best practices are the target behaviors described in Section 5.2.2 of this document, and are the ones that will let us analyze if Scrum is being used or not.

This solution was used to try to solve the research problem, **the poor adoption of Scrum methodology** (or of tools to use that methodology, such as, Jira) **in programmers' daily work routine**. The resolution of this primary problem will consequently, bring advantages for both the programmers and the company where they work, by fulfilling client's needs, meeting deadlines, reducing company's costs [58] and increasing programmers' performance.

5.2. Description

The software development methodologies, such as Scrum, in IT companies, are often combined with tools to provide an interface for following that methodology, such as Jira. Jira is a tool that helps programmers follow a Scrum methodology with just a few clicks and with an easy and intuitive interface, as described in Section 5.2.1.

In order to implement this solution, data was extracted from Jira. By combining this data with Scrum best practices (target behaviors defined in Section 5.2.2) and defined metrics (defined in Section 5.2.3), one can deduce if Scrum is being properly followed or not, and, consequently, if programmers will be rewarded or not with the game elements defined in Section 5.3.1.

The problem identified during this research is not only a programmers' problem, but also an organizational problem, for this reason we wanted to perform a demonstration to evaluate our proposal in an organizational environment with a real project and real tasks. Unfortunately, for reasons that were not explained to us, we were not allowed to complete a demonstration with a real project in the IT company that had been assigned for this project.

To demonstrate and evaluate this solution, a simulation was performed in an IT company that uses Jira, in an artificial environment with an artificial project and artificial tasks, and, then, a demonstration was performed in an academic environment with IT Master Students who were using Jira to manage a real project, their thesis, with Scrum. The context where this demonstration occurred brought some limitations to this thesis contributions, because the motivation of students and employees of a company are different, students are motivated by delivering their thesis but no salaries are involved while for employees, the salary could be their motivator. Another difference is that students are developing an individual project while employees of a company are developing projects in teams. Even the deadlines have a different weight for employees of a company that could have customers depending on them, and deadlines of the students that will only be evaluated on submitting their thesis during the final deadline. Having understood all these differences and limitations, we had no other opportunity than to test our solution with IT Master students.

5.2.1. Jira

Jira is a web-based tool for task issue tracking. Jira supports generic software development, such as waterfall. This tool has an embed add-on called Jira Agile, which supports Scrum software development, and Kanban, for flow-based teams.

In this section, Scrum functionalities are explained, because it was software development methodology in which this thesis is focused.

This tool provides an interface for the Product Backlog, where each member of the team can add or remove stories, and prioritize them as being more or less valuable to the customer. In the Backlog, the team collaborate to describe each story and to estimate the story points to attribute to each one.

This tool also provides the ability to create Sprints, and to move stories from the Backlog to the correct Sprint. For each Sprint several columns are available to be easy to track the team project. The Sprint's stories start in the column 'To Do', and can be moved to columns such as 'In progress', 'In Review', 'To Deploy', 'Done' (These names can be changed to meet specific project needs). These changes represent the workflow of a issue. The team should move each story 'status' from the beginning until the end of each Sprint.

At the end of each Sprint, each non-completed task is moved to the Backlog again, and each task in progress is moved to the next Sprint.

As is usual when teams use Scrum, they also use Burndown charts. These charts are automatically generated by Jira, and are useful for the team to track their progress, and to reflect about it in the Sprint Retrospective.

As is understandable by now, Jira allows, with some modifications, for any team to easily use Scrum Agile methodology.

5.2.2. Target Behaviors

In order to use Scrum properly, team members need to perform certain behaviors, which were mapped to certain metrics, and rewarded if they performed the desired behaviors (Section 5.3.1). The behaviors, based on the definition of Scrum, are presented in Table 4.

These target behaviors were defined based on the definition of Scrum presented in Section 4.1 and based on the fact that Jira deals with issues of different types (some by default and some custom defined), and it contains all the artifacts (Product Backlog, Sprint Backlog, and the Release). Based on that, one can deduce that for Scrum to work, there is a need for teamwork and for the existence of Meetings (to be attended), Stories (to be created, assigned and completed), Sprints and Scrum Master interventions (to help the development team to move forward). For some limitations of Jira it cannot guarantee the role of Product Owner or of Scrum Master, it only guarantees the role of Development Team. In order to incorporate the role of Scrum Master, the only elements needed to be checked are his interventions, and for that reason a special type of issue called “Impediment” was created and it is deduced that the assignee of those types of issues is the Scrum Master.

To start using Scrum, people need to start going to Meetings (such as Daily Meetings, Sprint Planning, Sprint Review and Sprint Retrospective), and to motivate this initial behavior we defined the 1st, 7th, 8th, and 9th behaviors in Table 4. Besides start performing this behaviors, it is wanted to them to continue to perform it consistently during all the project, so we defined the 2nd, 3rd, 4th, 5th, 6th, 10th, 11th, and 12th behaviors in Table 4 based on the recurrence in which a person or a team attend to a certain type of meeting.

“In software engineering, the main goal is to create software by an iterative process composed of several subsequent releases, which is made of a series of iterations” [3]. In order to deliver a deliverable increment at the end of each Sprint, the team need to divide the project in small parts (13th behavior) and complete those stories in the sprint in which they are assigned (14th, 15th, 16th, 17th and 18th behavior).

An important part of Scrum is teamwork. To introduce collaboration and teamwork we created some behaviors to reward the whole team such as if they share knowledge (18th behavior) or if they completed the assigned stories during the sprint or the project (15th, 17th. And 18th behaviors).

In order to help the team move forward, the Scrum Master needs to remove impediments in the development team’s path, reason the 19th behavior was created.

There are other behaviors that could be introduced as future work, in order to reward programmers with a stricter temporal control, such as, estimate the time of a certain number of stories correctly, or completed them in the estimated time. This could also be introduced for the attribution of story points, which can be related to the time in which a person takes to complete a story.

The complete list of the behaviors defined as target behaviors (the Scrum best practices) is presented in the next page in Table 4.

Table 4 – target behaviors (original)

Id	Behavior
1	A person attends one Daily Meeting.
2	A person attends 5 sequential Daily Meetings in a project.
3	A person attends 10 sequential Daily Meetings in a project.
4	A person attends 15 sequential Daily Meetings in a project.
5	A person attends all Daily Meetings during the project.
6	A person attends all Daily Meetings during a sprint of a project.
7	A person attends one Sprint Planning during the project.
8	A person attends one Sprint Review during the project.
9	A person attends one Sprint Retrospective during the project.
10	A person attends all Sprint Planning meetings during the project.
11	A person attends all Sprint Review meetings during the project.
12	A person attends all Sprint Retrospective meetings during the project.
13	The project was divided in small parts.
14	A person completes all stories assigned to him in a sprint of a project.
15	The team completes all stories in a sprint of a project.
16	A person completes all stories assigned to him in a project.
17	The team completes all stories in a project.
18	The team shares knowledge among other team members.
19	Scrum Master intervenes to help the team.

5.2.3. Metrics

After defining the target behaviors, some metrics were defined (Table 5) and were mapped (Table 6) with the behaviors defined in order for our solution to clearly identify when the defined abstract behaviors occurred.

The metrics were defined by analyzing the abstract behaviors and trying to understand how they can be identified by our solution. For example, in order to know if a person went to one daily meeting in a sprint, our solution only needs to verify the existence of one daily meeting in that sprint. But, to know if a person went to 5, 10 or 15 daily meetings, our solution needs to verify the number of daily meetings attended by that person in that sprint. The complete list of metrics is presented in Table 5 and the corresponding mapping is presented in Table 6, the mapping is not one-to-one because for the example given above, it is understandable that several behaviors depend on the same metric.

Table 5 – Metrics (original).

Id	Metric
1	Existence of a cross-functional Development Team.
2	Existence of a Scrum Master.
3	Intervention of the Scrum Master to help the Development Team.
4	Number of Daily Meetings reported per sprint.
5	Consecutive Daily Meetings reported per sprint.
6	Existence of a Sprint Planning, a Sprint Retrospective and a Sprint Review report of the meeting per sprint.
7	Number of stories per sprint.
8	Number of stories per Project.
9	Number of stories completed per sprint.
10	Number of stories completed per project.
11	Number of members to edit a story.
12	Consecutive Sprint Planning, Sprint Retrospective and Sprint Review reports in a project.

5.3. Gamified Solution

This section aims to present a connection between the theoretical concepts explained in the related work and the game elements and mechanisms selected to be used in this gamified solution.

“Gamification isn’t a solution to every business problem” [9], as Werbach and Hunter said. To solve this our problem, the organization’s environment and culture should be taken into account [45]. The elements which motivate people to play have to be identified and tailored to match the solution [4]. For example, if gamification introduces excessive competitiveness it could go against the organization’s culture (as already written in Section 4.2.2, some cultures are competitive, and some are not) and demotivate those already engaged [59]. It is also important to highlight that intrinsic motivators, such as reputation or altruism, are able to motivate players for a longer period of time than extrinsic motivators, such as rewards [60], which can cause dependence from them.

5.3.1. Game Elements

To implement our solution, some game elements were used from the three different levels of the pyramid presented in page 17 of this document.

To influence the game, two **dynamics** were used, but, as with all the dynamics, they are not directly visible [9]. In our solution we used:

- Progression (player’s evolution) – by providing the user with best badges and better points if they accomplish a more difficult task.
- Relationships (social interactions) – by creating collaboration, by rewarding teams, and by creating healthy competition, with the leaderboard.

To engage the users, some **mechanics** were used to achieve elements of dynamics, described above [9]. In our solution we used:

- Cooperation (various players working together towards the same goal) – by rewarding players not only individually, but also in teams.
- Competition (one player wins and the other loses) – by motivating players to compete to be the best and to be on top of the leaderboard.
- Rewards (obtained for performing tasks) – by providing users with badges and points and a corresponding position on the leaderboard if he performs a certain action.

Components are the visible states that the elements described above take [9]. In our solution we used:

- Achievements (objectives completed).
- Badges (visual representation of achievements).
- Leaderboards (visual representation of player’s progression in comparison to others).
- Points (earned by completing tasks).
- Teams (groups that work together towards a common goal).

Elements such as leaderboards were used to foster competition among all members, but without ignoring the demotivating power they can bring [9], because they have a great social impact on people. So, no money/salaries were directly connected to the position on the leaderboard. Motivation is wanted, but conflicts need to be avoided.

Points were used to reward programmers when they performed the defined target behaviors. Those points were given according to the difficulty to complete each task. By performing a simpler task a person earned less points than for performing a more complex task.

We also used badges to represent achievements, we consider each behavior an achievement, and for each behavior a person received a badge. To foster collaboration, our solution not only had individual badges, but also team badges.

A mapping between points and badges and the corresponding behavior is presented in the next page in Table 6.

5.3.2. Motivation

Besides all the elements already described, we used the two loops explained in Section 4.3.3 of this document, the progressions stairs and the engagement loop, to increase motivation of programmers, that was the first focus of the problem. The target activities and the game mechanisms used, need to constantly surprise the people involved, in order to keep them engaged.

The **engagement loop** was implemented by awarding a badge (as feedback) each time a user performed a target behavior, and this will motivate users to keep performing those target behaviors.

The **progression stairs** were used to give users a sense of progression during each project, by providing them with increasing difficulty to accomplish the achievements and badges during the project (e.g. the reward for attending one daily meeting is easier, and less than the reward for attending 15).

5.3.3. Types of gamification

The gamified system created, was based on **internal gamification**, because we wanted to improve productivity inside the company. But, it was also based on **behavior change gamification**, because we also wanted to change programmers' daily habits at work.

The behaviors described can be mapped to metrics that can be collected directly from the system in order to reward users when they perform in a certain way. Table 6, explains how the behaviors are mapped to metrics and also how the system rewards them in each situation. No negative feedback or negative rewards were used, in order not to make them lose motivation and engagement.

It is important to explain that the team badges and points are given to each member of the team. Personal and team badges are visible in each member's profile page, and are kept between projects. Each member can check all the members' badges and points, to foster competition.

Table 6 – Mapping target behaviors in the metrics, with the corresponding game rewards (original).

Behav.	Metrics	Rewards
1	4	+ 10 points
2, 3, 4	5	+ 20 points * number of sequential meetings
5	5	Badge (Project Lover) + 1000 points
6	5	Badge (Sprint Lover) + 400 points
7	6	Badge (Planner) + 20 points
8	6	Badge (Reviewer) + 20 points
9	6	Badge (Retrospecter) + 20 points
10	12	Badge(Planning Project Lover) + 400 points
11	12	Badge(Reviewing Project Lover) + 400 points
12	12	Badge(Retrospective Project Lover) + 400 points
13	7,8	Badge (Teamwork) at the end of the Project.
14	9	Badge (Story Killer) at the end of the Sprint
15	9	Badge (Team - Story Killer) at the end of the Sprint
16	10	Badge (TOP Story Killer) at the end of the Project
17	10	Badge (Team – TOP Story Killer) at the end of the Project
18	11,1	Badge (Sharing) + 100 points
19	2,3	Badge (Helper) + 100 points

The interface of this system is presented in the following section, and a complete list of the screens with all game elements mentioned above is presented in Appendix B of this document.

The following section explains the prototype created as the solution for this research problem. The prototype consists of a system which provides positive rewards when programmers are following

Scrum best practices (defined target behaviors). To identify if the behaviors were being performed, data was extracted from Jira, as explained in the following section.

5.4. Prototype

After choosing the game elements to use in this solution, the development of this thesis' solution started with the configuration of a Jira instance, followed by the creation of a Java app to extract data, and ending with a PHP app to analyze the data extracted and to create the user interface.

5.4.1. Jira Instance Configuration

In order to use Scrum properly, some roles, artifacts and events, should be created and followed. Scrum best practices (programmers' target behaviors) were already defined in Section 5.2.2 of this document, but, this section details the configuration that was performed on a Jira instance in order to be able to extract all the data needed and identify the behaviors performed.

Before starting to develop a prototype, a Jira instance was configured to support Scrum. That instance by default support issues such as bugs, tasks and sub-tasks. To support Scrum by definition, the Jira instance needed to be able to track meetings and scrum master interventions, so, issue types such as Meetings (to control the Daily Meetings, Sprint Planning, Sprint Review, Sprint Retrospective), and Impediments (to control the number of times a Scrum Master intervenes to help the team), needed to be created. Additionally, a custom field called "Meeting Members" was added to each issue of type Meeting, to control user attendance at each meeting. The attendance list is public in order to avoid false information.

5.4.2. Data Extraction

Gamification requires algorithms to measure and respond to actions. Since our simulation and demonstration were done with users who use Jira as an interface for Scrum, data was collected from there and a set of algorithms were defined to track their behavior.

As presented in the Figure 14, in the first iteration, a Java Application was developed to extract data from Jira. Jira provides a mechanism that allows us to define a URL and set it as webhook. A webhook is a user-defined HTTP callback, that is triggered by events, such as creating a story, editing a story or deleting a story, and the corresponding action to be taken is defined by us in a Java Application.

Initially, a temporary requestb.in URL was created to receive callbacks, and those callbacks were read by a JSON Parser in the Java Application. Later, this step was considered unnecessary, because a simple Servlet running on Server could do the job and eliminate that step, so, it was deleted (Figure 15).

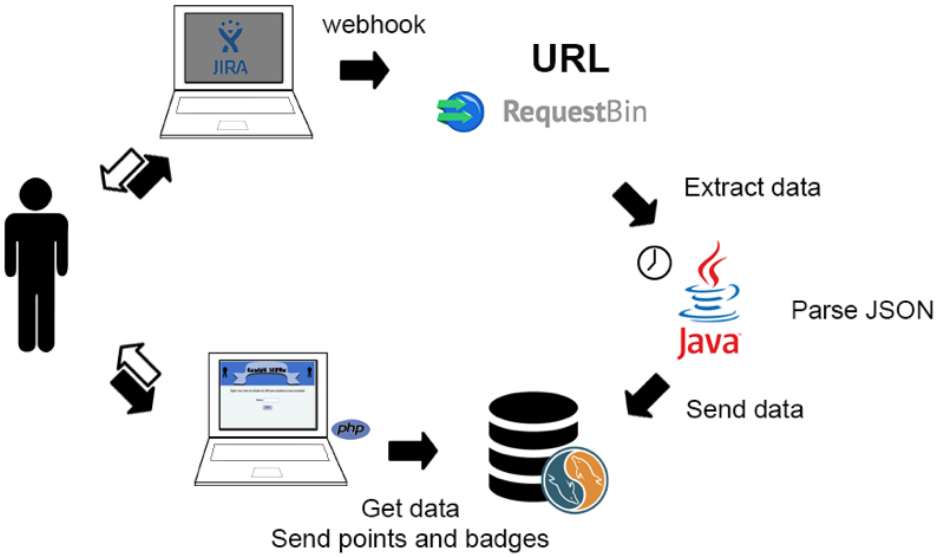


Figure 14 - First architecture of the prototype.

The Java Servlet received the JSON messages instantly when a project, board, user, worklog, issue or comment was created, updated or deleted. And, finally, when a sprint was created, updated, deleted started or closed. That application sent all the data to a MySQL database, created specifically for the project.

The database has tables to record all the important data to be analyzed, such as all data from the Stories, Meetings and Impediments, and some specific data such as all the Assignees/Reporters of a Story, as well as the members of a meeting.

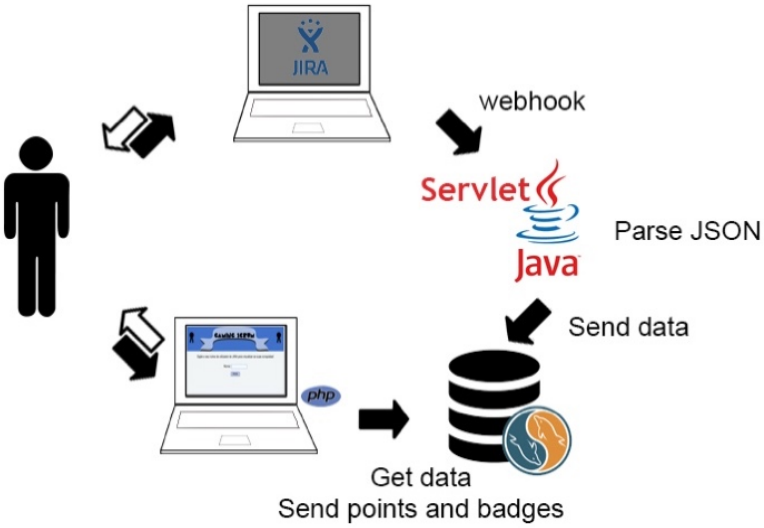


Figure 15 - Second and final architecture of the prototype.

5.4.3. Interface

After this, the first functional prototype in PHP was created. The prototype gave users real time information about their performance in Scrum. In this first prototype every behavior of this document's solution was implemented. To use this prototype, the Jira user had to login on the website with his username, as presented in Figure 16.

Figure 16 - Login screen.

At this phase, the application showed users a leaderboard with medals for the top 3, and with the score of all Jira users (Figure 17). Next to the score, users could select the "?" option, and were redirected to a screen where all their points were mapped in to the concrete behavior performed. The points and the leaderboard elements were used to foster competition and to create a win state for the user with the highest score in the leaderboard.

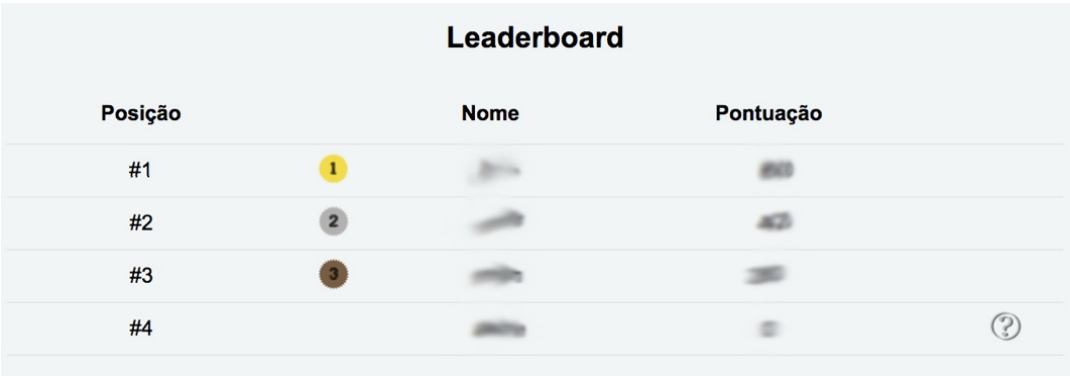


Figure 17 – Leaderboard.

A table with personal and team badges was also created. These badges were designed in order to provide some fun to the users. Also, the team badges aimed to provide cooperation inside the teams. Again, to promote competition, users were given the ability to check other user's badges. A complete list of badges of the system is presented in Appendix A with a brief explanation of the behavior that triggers its attribution.

This prototype was able to be configured according to each company's particular problem, so, it includes an administration screen where the managers can change the number of points attributed to each behavior. Appendix B presents a complete list of all application screens, filled with data gathered from the demonstration.

6. Demonstration

This section explains how this research solution was demonstrated, presenting the results from the simulation and the demonstration performed during this research, which were used to evaluate this thesis.

6.1. Simulation

To prove the artifact is working as expected with all its functionalities, a simulation was performed in an IT company that was facing the problem identified during this research. This simulation was performed with three people, during two weeks, and two sprints were performed with a duration of one week each. This simulation was performed in a non-real project with non-real tasks, with a small number of users (N = 3), so, it could not be used to evaluate any aspect of this document's solution other than proving that all functionalities of the prototype were working as expected.

6.1.1. Meetings

Users failed to report one meeting, since to guarantee that the meetings are performed in the correct day, the application only counts one daily meeting per day, and they reported two meetings in the same day (Table 7), which led the application to ignore one of the meetings. This was explained to users before the simulation.

Several meetings were performed correctly, five daily meetings, in four different days, where all team were present in just three of them. This information shows that they were not meeting on a daily basis, as was supposed to occur in Scrum.

They also performed three sprint reviews and two sprint retrospectives, in the two Sprints of the simulation. Unfortunately, it is obvious from Table 7 that they were not performing sprint planning meetings

From the total of the 11 meetings (counting with the one that was ignored with a duplicate date), only four meetings had all the three team members present, as presented in Table 7. The average of the attendance was 2,36 ≈ 2, which cannot lead us to any generalizable conclusion, because it was a small number of users.

6.1.2. Stories

Aside from not being able to use this simulation to evaluate other metrics, from the analysis of the data collected, one can confirm that users divided the project in small parts, creating 8 stories and all the stories were assigned to a user who completed them in the assigned sprint.

Table 7 - User attendance to meetings.

Meeting type	Date	User attendance	Num. Users
Sprint review	23/05/16	User 1, User 2, User 3	3
Daily meeting	25/05/16	User 1, User 2, User 3	3
Daily meeting	24/05/16	User 1, User 2	2
Sprint review	24/05/16	User 1, User 2	2
Sprint retrospective	24/05/16	User 1, User 3	2
Daily meeting	27/05/16	User 1, User 2, User 3	3
Daily meeting	30/05/16	User 1, User 2, User 3	3
Daily meeting	02/06/16	User 1, User 3	2
Daily meeting	02/06/16	User 1, User 2	2
Sprint review	06/06/16	User 1, User 2	2
Sprint retrospective	06/06/16	User 1, User 3	2

6.1.3. Behaviors

Several target behaviors were identified Table 8, to see the description of each behavior refer to Section 5.2.2.

Table 8 - Behaviors performed per user.

User	Behavior Id
User 1	1, 2, 5, 17, 15, 6, 14, 15, 6, 14, 16, 11, 8, 8, 12, 9, 9
User 2	1, 2, 5, 17, 15, 6, 14, 15, 6, 14, 16, 11, 8, 8
User 3	1, 17, 15, 14, 15, 6, 14, 16, 8, 12, 9, 9

The prototype gathered all the information from Jira with no bugs, and calculated the badges (Figure 18), points and leaderboard positions (Figure 19), as expected.

Figure 18 - Badges of the simulation (User 1).

Posição		Nome	Pontuação	
#1	1	[illegible]	600	?
#2	2	[illegible]	420	
#3	3	[illegible]	280	

Figure 19 - Leaderboard of the simulation.

Section 6.2 of this document explains the demonstration performed in a real environment, to prove validity of this solution.

6.2. Demonstration

The demonstration was performed to evaluate this document's solution. The demonstration was performed in an academic environment, with eleven IT master students, who were performing real-world projects, their Master Thesis.

The context in which this demonstration occurred brought some limitations to the contributions from this thesis. As already described in Section 5.2 of this document, the motivation of students and employees of a company are different. These differences suggest that some conclusions made in this thesis cannot be generalizable to programmers that work in a company, but, the results from this demonstration could be good indicators of the possibility of change in behavior and improvement in the adoption of Scrum.

During a month, those students performed weekly sprints, with weekly meetings, where they reviewed the last sprint tasks and report the tasks they would perform in the next one. These are guided by my supervisor, Professor Miguel Mira da Silva, and I, in each step of this demonstration.

All the tasks and meetings were reported in Jira. Half of them had access to the gamification prototype and the other half did not. The objective was to compare the results of those who used gamification, and those who did not.

Six students were selected to use gamification and the other 5 did not use it, my teacher and I did not belong to any team since we knew the objective of this experiment and did not want to "game the game". The team that used gamification was called Team G, the other team was called Team NG (my teacher and I do not belong to any team even though one or both of us participated in all the meetings performed).

The solution extracted all the information from Jira with no bugs, although, this demonstration has some limitations such as the impossibility of testing all the types of meetings, because in this experiment the daily meetings were not performed because the students had a lot of work to do and could not report progress daily. The sprint planning meetings were also impossible to perform weekly, because most of the students' tasks depend on people outside of the project. This is the reason why only one sprint planning was performed.

Those limitations have already been seen as common Scrum adoption problems, but, in this case, they were defined and agreed between my teacher and I, so we could not take them into account as resistance to Scrum adoption. Furthermore, as the evaluation consisted of a comparison between two teams to see the impact that gamification could have on both, and both had the same limitations, it did not influence the results obtained.

This section is divided in three sub-sections, Meetings, Stories and Behaviors, which are the main categories to analyze in the following section, Evaluation.

6.2.1. Meetings

As already explained, the types of meetings performed in this demonstration were the weekly sprint review, sprint retrospective, and an initial sprint planning meeting.

During this project a total of seven meetings were reported, one sprint planning, three sprint retrospectives, and three sprint reviews. The sprint planning was performed at the beginning of the project, and one sprint review and retrospective were performed at the end of each one of the four sprints.

Table 9 shows that the team which used gamification (Team G), attended 66% of the total number of meetings, and the team which did not use gamification (Team NG), attended 34% of the total number of meetings performed.

Table 9 – Team attendance to meetings.

Meeting type	Date	Num. Users	Team G	%Team G	Team NG	%Team NG
Sprint Planning	01/07/16	3	3	100%	0	0%
Sprint Review	11/07/16	5	4	80%	1	20%
Sprint Retrospective	11/07/16	5	4	80%	1	20%
Sprint Review	18/07/16	11	6	55%	5	45%
Sprint Retrospective	18/07/16	11	6	55%	5	45%
Sprint Review	23/07/16	0	0	0%	0	0%
Sprint Retrospective	23/07/16	0	0	0%	0	0%
	Total	**35**	**23**	**66%**	**12**	**34%**

All the students were invited to all meetings, although, it is understandable that the total attendance varies depending on the academic work the students had to perform in each week, since their Master Thesis was not their only task. The last two meetings were performed only by my teacher and I. Since we were leading this demonstration and our data could not be taken into account, our attendances could not be taken into account either, and for this reason none of the others students were obligated to be present and the total number of attendances were 0. However, that Sprint contained important data for analyzing this demonstration, and for this reason, these two meetings could not be eliminated from this demonstration.

6.2.2. Stories

Regarding the stories that were supposed to be created during the project, the 11 students created 49 and assigned each one according to its owner. Below is a table showing the comparison between the number of stories created and completed by Team G and Team NG.

During the project, 49 stories were created by both teams, but only 27 were completed (Table 10). Team G created 28 stories and Team NG created 21. Team G completed 19 stories and Team NG completed only 8.

Table 10 – Number of stories created and completed total and per team.

Stories Created	Stories Completed	Created by Team G	Created by Team NG	Completed by Team G	Completed by Team NG
49	27	28	21	19	8

Below, Table 11, presents the percentage of stories completed and non-completed from the total number of stories created by each team.

Table 11 - Percentage of stories completed per team.

	Team G	Team NG
Created	28	21
Completed	19	8
% Completed	68%	38%
% Non-Completed	32%	62%

Team G completed 68% of the 28 stories they created and Team NG completed only 38% of the 21 stories they created.

6.2.3. Behaviors

The last topic of this section is the “Behaviors”, which correspond to an important step of this work, which is the analysis to understand if users are performing the target behaviors, already described in Section 5.2.2. Below, is a list of the target behaviors performed by the students during this demonstration.

Table 12 shows us that Team NG performed 21 target behaviors, performing 3 times the 13th behavior, 6 times the 14th behavior, 6 times the 8th behavior and 6 times 9th behavior .

Table 12 - Team NG behaviors performed.

	Times	Behavior Id
	3	13
	6	14
	6	8
	6	9
Total	**21**	

Table 13 shows us that the team which use gamification (Team G) has performed 46 target behaviors, 3 times was performed the 7th target behavior and so on. To see a complete description of each behavior, refer to Section 5.2.2.

Table 13 - Team G behaviors performed.

	Times	Behavior Id
	3	7
	10	9
	6	13
	13	14
	3	10
	10	8
	1	16
Total	**46**	

The analysis of these results is presented in the next section as an evaluation of this solution for the problem identified.

Appendix B presents a complete list of all application screens, filled with data gathered from this demonstration.

7. Evaluation

This section consists of the method used to measure how well this artifact supports our solution to the research problem, which corresponds to the step “Evaluation” of DSRM. This is the most important step of DSRM because it allows us to validate the proposed solution to the research problem.

At the end of this activity, the proposal was to iterate back to step 3 of DSRM to improve the effectiveness of the artifact, but, because of some time constraints, only one iteration was performed, which is evaluated in this section. After this step of DSRM this research was communicated to the scientific community.

This section is composed by a description of the DSRM evaluation framework selected, followed by an analysis of the data gathered from the simulation and from the demonstration with eleven IT Master students, which was already detailed in the previous section. The measurement of how well the solution met the objectives of this solution is divided in 3 important parts. The meetings and the stories, which are main components of Scrum, and the behaviors, which correspond to the Scrum best practices, defined in Section 5.2.2.

7.1. Evaluation Framework

The Evaluation is an important step of DSRM, however, to perform it, there exist more than one framework. In this case, Strategic DSR Evaluation Framework [1] by Pries-Heje et al., was the selected one, because the metrics and behaviors to test and evaluate were already defined.

Table 14 presents the Strategic DSR Evaluation framework composed by the methods chosen to be used in this evaluation.

Table 14 - Strategic DSR Evaluation Framework with methods used to evaluate this solution (adapted from [1]).

	Ex-ante	**Ex-post**
Artificial	-	Simulation
Naturalistic	-	Demonstration

This table’s dimensions correspond to a simplified version of the three answers proposed by the framework [1], which will be answered in the following paragraphs.

1. What was actually being evaluated?

The solution was evaluated as a design product because the artifact consists of an application that incorporates game mechanisms according to users’ performance in Jira.

2. **How was it evaluated?**

The evaluation was performed in two steps. The first step consisted of an artificial evaluation, based on a simulation with a real system (the artifact), in an artificial project with artificial tasks, and a small number of users. This step was used to evaluate if the artifact was working as expected with all its functionalities. The second and last step consisted of a naturalistic evaluation, based on a demonstration with a real system (the artifact), with real users and a real project. This step was used to evaluate if the target behaviors defined previously, were achieved or not.

3. **When was it evaluated?**

An ex-post evaluation (after the artifact was developed) was performed to test the artifact's ability to solve these research problems.

The ex-post evaluation of the simulation and demonstration performed are presented in the following sections.

7.2. Simulation

The simulation performed prior to the demonstration, can only prove that the solution was extracting and processing data correctly, with no bugs. The data gathered from this simulation was not evaluated because it could not give us any generalizable data about the company's programmers, since it was performed in a non-real environment, with non-real tasks and with a small number of users (N=3).

The simulation was used as an ex-post evaluation to prove that the system was working with all its functionalities, as expected.

Besides the limitations presented above, this simulation showed that the company's programmers knew some steps that Scrum consists of, because without any intervention, they performed some target behaviors, attending to meetings and creating and completing several stories during the two Sprints.

During the simulation, programmers performed some Daily Meetings, but not every day. One can verify the inexistence of Sprint Plannings, but, a consistent existence of Sprint Reviews and Retrospective in the two Sprints, although not all members attend to. Regarding the target behaviors, they performed 12 types, which is far from being ideal in representing a flawless following of Scrum.

This data showed the need for further investigation of a gamified solution to cope with IT company's needs, but, for this purpose, a company which is open to perform a demonstration with real projects and a bigger amount of programmers is needed. Since during this research we did not find such company, we proceeded to perform a demonstration with IT Master Students, which changed the contributions of this thesis to the scientific community (already explained in Section 5.2), but, demonstrated that it could be a good indicator of the impact of gamification in adoption of Scrum by programmers.

7.3. Demonstration

This section consists of an analysis of the demonstration performed by eleven IT students. This demonstration was used as an ex-post evaluation. A detailed description of the conditions under which the demonstration was performed, was already provided in Section 6.2. In short, a demonstration was conducted, during one month, among eleven IT students who were managing their Master Thesis with Scrum, using Jira. Those students were divided in two teams, Team G (which used gamification) and Team NG (which did not used gamification).

This section is divided into three sub-sections, Meetings, Stories and Behaviors, which are the main categories to analyze in this evaluation. At the end of this section it should be clear if gamification can improve the students' motivation to perform meetings, to create stories, and to perform certain target behaviors.

7.3.1. Meetings

This section contains an analysis of the meeting attendance by both teams, although, a complete description of the types of meetings performed and limitations was already presented in Section 6.2.1.

Table 9, in Section 6.2.1, presents the data extracted from Jira regarding meeting attendance by Team G and Team NG. The Sprint Planning performed had only been attended for members of Team G and no members of Team NG. The first Sprint Review and Sprint Retrospective had 80% of the attendances by Team G and only 20% of Team NG. The second Sprint Review and Sprint Retrospective had 55% of the attendances by Team G and only 45% of Team NG. Finally, the last two meetings had only been attended by my teacher and I who are not being taken into account for this demonstration, which lead to 0 presences of members of both teams. Based on this data, it can be concluded that in the first five meetings, Team G had always more attendances than Team NG. Although, in the last two meetings, the absence of attendances is equal for both teams.

By analyzing Table 9, can be concluded that the attendance of users who are using the gamified solution in each meeting and in the average (line "total") of all meetings is greater than the one of those who were not using it.

Table 9 is a good indicator, that in this context, with students, a gamified platform can be used to motivate people to use a certain tool or perform a certain task, since every meeting is more attended by students who are using it than by those who are not, and the average percentage of the Team G attendance is 66% versus only 34% of the Team NG, who did not use gamification.

7.3.2. Stories

This section contains an analysis of the number of stories created and completed by both teams, although, a complete description of this topic, including the demonstration and correspondent results were already presented in Section 6.2.2.

By comparing Team G and Team NG it is clear that besides the number of stories created by both being similar (28 vs 21), the number of stories completed by the students of the team which used gamification is much higher than the only eight stories completed by Team NG (19 vs 8).

Students who had access to our gamified solution, not only performed more meetings, but also created more stories and divided the project into smaller part. By analyzing the percentage of stories created and completed by one team compared to the other was analyzed (Table 11), it can be verified that Team G completed 68% of the stories that they had created, leaving just 32% incomplete, while Team NG almost inverted these results, completing only 38% of the stories created by them, and leaving 62% incomplete.

These results let us conclude again that, in this context, with students, gamification could positively lead people to perform certain tasks, such as following Scrum, such as attend and perform more meetings, and to divide the project into smaller parts by creating more stories and completing them.

7.3.3. Behaviors

In Section 5.2.2 of this document the target behaviors were defined, and those are the goals of the solution in order to achieve the main goal, which is to increase Scrum adoption among programmers. In this section an analysis of what goals (target behaviors) were achieved or not, will be presented. The behaviors performed by both teams are already present in Section 6.2.3, in Table 12 and Table 13.

It can be concluded that the team that used gamification performed more target behaviors in terms of number and types. Table 12 and Table 13 show that the team which did not use gamification performed only 21 behaviors (from 3 types), and, on the other hand, the team which used gamification performed a much higher number of behaviors, 46 (from 7 types). By performing further analysis, it can also be concluded that there is no type of behavior that had been performed by Team NG and had not been performed by Team G. Although Team G has 6 members and Team NG has only 5, that was not the reason which led to this great difference in all the results.

By performing a more detailed analysis, it can be verified that not all the 19 target behaviors defined were performed, and it must be noted that, some limitations of this demonstration were the cause of this. The behaviors from 1st to 6th did not occur because daily meetings were not performed in this demonstration. Behaviors 11 and 12 did not occur because the last two meetings were only attended by me and the teacher, and the other students were allowed to miss those meetings. The 18th behavior, which refers to more than one user working in the same story, was also impossible to be performed because all students were working on different tasks. The 19th behavior was not tracked since that behavior corresponds to an intervention of the Scrum Master, and in this case that was me. Since I was leading this demonstration and my data could not be taken into account, the stories I created could not be taken into account either, which contain my interventions as Scrum Master.

Finally, there were two behaviors which did not happen for some reason, 15th and 17th behaviors were not performed since there was not a moment where all members of one team completed all the stories in a sprint or in the whole project.

From the 19 behaviors identified, only 7 were performed, and this was due to the limitations described above. From the 12 behaviors not performed, only two were possible to be performed, the 15th and the 17th. Which makes us conclude that gamification had influenced the students to use Scrum.

Since our evaluation was based on comparing results from the two teams, from that comparison it can be confirmed that gamification influenced positively the users who were using it, by increasing the number and types of target behaviors performed.

7.3.4. Discussion

The results analyzed during the demonstration are good indicators, that in this context, with students, a gamified platform can be used to increase Scrum adoption, since every meeting is more attended by students who are using it than by those who are not, and the division of the project into smaller stories and completing them in the assigned sprint, were more performed by the team who used gamification.

At the end, although this cannot be generalizable to programmers that work in a company, because they have different motivators (described in Section 5.2), we can demonstrate that gamification can have a positive impact in the adoption of Scrum among IT Master Students. This could be a good indicator that this work needs to be tested in an IT company with real projects and real tasks, to validate if in that context, the results are the same.

The following DSRM's step to be performed is Communication, and it is described in the following section.

8. Communication

This section corresponds to the last step of DSRM methodology, which describes how this work was communicated to reveal, to the relevant public, its relevance, and the artifacts created to solve the research problem, explaining also why this research is so important and unique.

During the research, a paper was submitted, in order to present this work to the scientific community, to a conference about Software Engineering in Practice, ICSE, to be held in 2017.

This work was also presented as this thesis' report, which will be presented, discussed and evaluated by a qualified jury.

9. Conclusion

As a conclusion for this document, this section presents a summary of all the work performed during this thesis, the lessons learned, limitations of the solution and, finally, a description of future work suggested.

As guide to this work, DSRM methodology was used, starting by defining the problem, defining the objectives for the solution, design the artifact, evaluate it and communicate it.

This thesis was motivated by a real problem, which is that programmers do not follow flawlessly the work processes, such as Scrum, in their work. This can lead to several other problems such as delays in projects, and projects that do not meet customer requirements or have a low capacity to adapt to new requirements. These problems can potentially increase the project costs for both the company and its customers, and can make the company lose some customers.

Gamification was used to solve to this problem because it can motivate and engage users in several contexts, such as work [26]. By applying game mechanisms and visual components such as points, badges and leaderboards, in order to achieve fun, users may increase their motivation to perform certain tasks.

The main objective of this thesis is to analyze if gamification can have a positive impact on programmers' adoption of Scrum.

To be able to perform that analysis, a prototype was created using gamification, with several game elements and mechanisms. That prototype extracted data from Jira to reward users when they performed the target behaviors defined earlier, which are the behaviors that compose Scrum.

The evaluation was performed based on Strategic DSR Evaluation Framework [1], proposed by Pries-Heje et al., in the simulation, and the data gathered from the demonstration explained below.

To test the prototype functionalities, a simulation was performed in an IT company. This data showed that the prototype was working as expected, and, although the data could not be generalized because of the artificial nature of the simulation, it showed the need for further investigation of a gamified solution to cope with the IT company's needs. However, during the research, a company open to perform a demonstration with real project and a bigger amount of users, was not found, so, we performed a demonstration with IT Master Students.

The context of the demonstration, changed the contributions of this thesis, as explained in Section 5.2., because the motivation of students and employees of a company are different, while some were motivated by delivering their thesis, the other were motivated by their salary and keeping their job, and, some were performing an individual work while others were performing work in teams. The deadlines also had different weights for different groups, because for one it was only the final thesis deadline, and for the other the deadlines could have customers depending on them.

This demonstration was used to evaluate if the solution was solving the proposed goals, by comparing the behaviors of a team which was using the gamified solution to that of another which was not. Notwithstanding the differences in context, with the data gathered, we can observe good indicators of the positive impact of gamification in programmers' adoption of Scrum, by increasing the number of target behaviors performed.

9.1. Lessons Learned

There are several lessons learned during this thesis which are explained below.

During this work we were able to understand, both from the literature review performed and from the contact performed with several IT companies, the huge problem that companies are facing nowadays with programmers that resist to change, and that do not follow work processes, such as Scrum, flawlessly. Solving these issues is what motivated is this work and is what will drive research in the same area in the future.

From the literature review we also understand that the element of surprise is essential to keep people engaged during long periods of time, which would be a problem we could have faced if the demonstration was performed during a longer period of time.

With the solution developed, we can demonstrate how gamification can have a positive impact on programmers' adoption of Scrum, by making them perform more Scrum best practices than if they had not been in contact with gamification.

9.2. Limitations

The simulation performed had several limitations already described, such as being performed in an artificial environment, with an artificial project with artificial tasks, and a small number of users. These limitations increase the time needed to gather usable data, making it impossible to perform a second iteration because the first simulation could not give us valid data from the behaviors. The fact that we only performed one iteration was a limitation of this work.

As already explained in this report (Section 5.2), the context of the demonstration, changed the contributions of this thesis, because students and employees of a company have different motivators. So, the data gathered and our conclusions, are reliable, but, are not generalizable for an organizational environment of an IT company.

The demonstration performed also had some limitations such as not testing daily meetings, which led us to not test six of the target behaviors. Another limitation was that each user was working on a different project, their individual thesis, which also led to not testing one more target behavior. Finally, in this project, the author of this thesis was also the Scrum Master, which meant that the stories of the author were not considered in this work, making it impossible to test the behavior regarding scrum master interventions.

The solution developed had an additional limitation which needs to be highlighted, as it could help to improve Scrum adoption even more, which is, that it was not developed inside Jira, it was actually developed independently. The data extracted from Jira by our program is stored in a database, from where it will be possible in the future to get that data and correspondent behaviors and use it in the development of an add-on to Jira.

9.3. Future Work

Using this thesis as a foundation for future work, with more time and knowledge, the most important recommendation is to transform this solution into an add-on for Jira and to demonstrate it and evaluate it in an IT company. All the important data extracted from JIRA is stored in a database, which makes it easier to create an add-on to Jira, by reusing the extraction performed in this work and the target behaviors defined. This step would be very important to improve the adoption of Scrum, because a programmer who works with Jira all day, could check his progress, points, and badges, directly in his work environment.

To be effective in the long run, we will have to find a way to keep programmers interested when the novelty of the artifact disappears. This could include new behaviors to be rewarded, more customization, new rewards and more surprises. A more complete description of the badges would also be important for users to understand their progress.

A diagram with points per week and per month, would also be an important functionality, in order to create the possibility for users to change their leaderboard position according to the time-frame.

The possibility of customizing the add-on for each company, would also be an interesting new feature, allowing each company to gamify their work processes, such as, waterfall and quality management, which also exist in Jira Software and in Jira Service Desk, respectively.

From the promising results gathered with the IT Master Students, we believe this work is just the first iteration of a great solution to solve programmers' motivation problems, and other students will be able to build on this work by continuing to perform more iterations.

Bibliography

[1] J. Pries-Heje, R. Baskerville, and J. R. Venable, “Strategies for Design Science Research Evaluation,” *Eur. Conf. Inf. Syst.*, no. 2004, p. 87, 2008.

[2] H. Hajjdiab and A. S. Taleb, “Agile adoption experience: A case study in the U.A.E,” *ICSESS 2011 - Proc. 2011 IEEE 2nd Int. Conf. Softw. Eng. Serv. Sci.*, pp. 31–34, 2011.

[3] E. B. Passos, D. B. Medeiros, P. A. S. Neto, and E. W. G. Clua, “Turning real-world software development into a game,” in *Brazilian Symposium on Games and Digital Entertainment, SBGAMES*, 2011, pp. 260–269.

[4] E. Herranz, R. Colomo-Palacios, and A. Amescua Seco, “Gamiware: A Gamification Platform for Software Process Improvement,” in *Communications in Computer and Information Science*, vol. 543, Springer, 2015, pp. 127–139.

[5] W. S. Humphrey, *A Discipline for Software Engineering*, vol. 640. 1995.

[6] K. Scotland and A. Boutin, “Integrating scrum with the process framework at Yahoo! Europe,” in *Proceedings - Agile 2008 Conference*, 2008, pp. 191–195.

[7] H. Press, *Managing Change to Reduce Resistance*. Boston, Harvard Busiess Book Press, 2005.

[8] K. M. Kapp, *The gamification of learning and instruction: game-based methods and strategies for training and education*. John Wiley & Sons, 2012.

[9] K. Werbach and D. Hunter, *For the win: How game thinking can revolutionize your business*. Wharton Digital Press, 2012.

[10] A. Loriggio, V. Farias, and P. Mustaro, “Aplicações de gamificação e técnicas de motivação à aprendizagem da metodologia ágil scrum,” in *VIII International Conference on Engineering and Computer Education*, 2013, vol. 8, pp. 326–330.

[11] D. J. Dubois and G. Tamburrelli, “Understanding gamification mechanisms for software development,” *Proc. 2013 9th Jt. Meet. Found. Softw. Eng.*, p. 659, 2013.

[12] M. J. Akhtar, A. Ahsan, and W. Z. Sadiq, “Scrum adoption, acceptance and implementation (A case study of Barriers in Pakistan’s IT Industry and Mandatory Improvements),” *Proc. - 2010 IEEE 17th Int. Conf. Ind. Eng. Eng. Manag. IE EM2010*, pp. 458–461, 2010.

[13] S. Raphael, “Gamification in Change Management processes An empirical research by means of qualitative methods to analyze relevance , implications and selected use cases,” University of Applied Sciences Munich, 2014.

[14] P. J. Denning, “A new social contract for research,” *Commun. ACM*, vol. 40, no. 2, pp. 132–134, 1997.

[15] P. J. Denning and R. M. Metcalfe, *Beyond Calculation: The Next Fifty Years of Computing*. Springer Science & Business Media, 1998.

[16] H. Österle *et al.*, “Memorandum on design-oriented information systems research,” *Eur. J. Inf. Syst.*, vol. 20, no. 1, pp. 7–10, 2011.

[17] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, “A design science research methodology for information systems research,” *J. Manag. Inf. Syst.*, vol. 24, no. 3, pp. 45–77, 2008.

[18] K. Schwaber and J. Sutherland, “The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game,” *Scrum. org, Oct.*, vol. 2, no. October, p. 17, 2011.

[19] G. Cloke, “Get your agile freak on! Agile adoption at Yahoo! Music,” in *Proceedings - AGILE 2007*, 2007, pp. 240–248.

[20] A. Atlas, “Accidental adoption: The story of scrum at amazon.com,” in *Proceedings - 2009 Agile Conference, AGILE 2009*, 2009, pp. 135–140.

[21] A. Begel and N. Nagappan, “Usage and perceptions of Agile software development in an industrial context: An exploratory study,” in *Proceedings - 1st International Symposium on Empirical Software Engineering and Measurement, ESEM 2007*, 2007, pp. 255–264.

[22] R. T. Todnem, “Organisational Change Management: A Critical Review,” *J. Chang. Manag.*, vol. 5, no. 4, pp. 369–380, 2005.

[23] G. Kraus, C. Becker-Kolle, and T. Fischer, *Change-Management: Steuerung von Veranderungsprozessen in Organisationen; Einflussfaktoren und Beteiligte; Konzepte, Instrumente und Methoden*. Cornelsen, 2010.

[24] Kübler-Ross, *On death and dying: What the dying have to teach doctors, nurses, clergy and their own families*. Taylor & Francis, 2009.

[25] J. P. Kotter and D. S. Cohen, *The heart of change: Real-life stories of how people change their organizations*. Harvard Business Press, 2012.

[26] B. Burke, *Gamify: How Gamification motivates people to do extraordinary things*. Bibliomotion, Inc., 2014.

[27] Oxford English Dictionary, *Paperback Oxford English Dictionary*. 2002.

[28] K. Salen and E. Zimmerman, *Rules of Play: Game Design Fundamentals*. MIT press, 2004.

[29] G. Zichermann, “Fun is the Future: Mastering Gamification,” *GoogleTechTalks*, 2010. [Online]. Available: http://www.youtube.com/watch?v=6O1gNVeaE4g.

[30] A. J. Kim, “Gamification 101: Designing the player journey.” GoogleTechTalks, 2011.

[31] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, “From Game Design Elements to

Gamefulness: Defining Gamification," in *Proceedings of the 15th International Academic MindTrek Conference on Envisioning Future Media Environments - MindTrek '11*, 2011, no. January, pp. 9–15.

[32] K. Huotari, "Defining Gamification - A Service Marketing Perspective," *Proc. 15th Int. Acad. MindTrek Conf. Envisioning Futur. Media Environ.*, pp. 17–22, 2012.

[33] A. Dorling and F. McCaffery, "The gamification of SPICE," in *Communications in Computer and Information Science*, vol. 290, Springer, 2012, pp. 295–301.

[34] R. Hunicke, M. LeBlanc, and R. Zubek, "MDA: A Formal Approach to Game Design and Game Research," *Work. Challenges Game AI*, pp. 1–4, 2004.

[35] M. Csikszentmihalyi, "Beyond Boredom and Anxiety: Experiencing Flow in Work and Play," *Jossey-Bass Behav. Sci. Ser.*, 1975.

[36] R. Koster, "Theory of Fun for Game Design," *A Theory Fun Game Des.*, p. 256, 2005.

[37] K. M. Kapp, L. Blair, and R. Mesch, "The Gamification of Learning and Instruction Fieldbook," *Princ. Leadersh.*, vol. 12, no. December, pp. 56–59, 2014.

[38] N. Ahmadi, M. Jazayeri, F. Lelli, and S. Nesic, "A survey of social software engineering," in *Automated Software Engineering - Workshops, 2008. ASE Workshops 2008. 23rd IEEE/ACM International Conference on*, 2008, pp. 1–12.

[39] J. Pries-Heje and J. Johansen, *Improve IT: A book for improving software projects*. DELTA, 2007.

[40] M. E. Gredler, *Learning & Instruction Theory into Practice*, 3rd ed. Prentice Hall, 1996.

[41] J. M. Keller, "Development and use of the ARCS model of motivational design," *J. Instr. Dev.*, vol. 10, no. 1932, pp. 2–10, 1987.

[42] S. Deterding, "Gamification: designing for motivation," *Interactions*, vol. 18, no. 4, pp. 14–17, 2012.

[43] R. Marques, J. Gregório, F. Pinheiro, P. Póvoa, M. M. Da Silva, and L. V. Lapão, "Improving nurses' hand hygiene compliance using gamification," *CEUR Workshop Proc.*, vol. 1574, no. Pahi 2015, pp. 33–42, 2016.

[44] F. Petrillo, M. Pimenta, F. Trindade, and C. Dietrich, "Houston, we have a problem...: A survey of actual problems in computer games development.," in *ACM symposium on Applied computing*, 2008, pp. 700–711.

[45] L.-T. Cheng *et al.*, "Finding Moments of Play at Work," *CHI 2011 Work. Gamification Using Game Des. Elem. Non-Game Context.*, pp. 2–5, 2011.

[46] E. Navarro, "SimSE: a software engineering simulation environment for software process

education," *Vasa*, p. 321, 2006.

[47] A. Baker, E. O. Navarro, and A. Van Der Hoek, "An experimental card game for teaching software engineering processes," *J. Syst. Softw.*, vol. 75, no. 1–2, pp. 3–16, 2005.

[48] K. Claypool and M. Claypool, "Teaching software engineering through game design," *ACM SIGCSE Bull.*, vol. 37, no. 3, pp. 123–127, 2005.

[49] E. Sweedyk and R. M. Keller, "Fun and Games : A New Software Engineering Course," *Sci. Educ.*, pp. 138–142, 2005.

[50] S. Sheth, J. Bell, and G. Kaiser, "Halo (highly addictive, socially optimized) software engineering," in *Proceedings of the 1st International Workshop on Games and Software Engineering*, 2011, pp. 29–32.

[51] L. Singer and K. Schneider, "It was a bit of a race: Gamification of version control," in *2012 2nd International Workshop on Games and Software Engineering: Realizing User Engagement with Game Engineering Techniques, GAS 2012 - Proceedings*, 2012, pp. 5–8.

[52] C. Andres and K. Beck, "Extreme Programming Explained: Embrace Change," *Read. Addison-Wesley Prof.*, 2004.

[53] W. Snipes, V. Augustine, A. R. Nair, and E. Murphy-Hill, "Towards recognizing and rewarding efficient developer work patterns," in *Proceedings - International Conference on Software Engineering*, 2013, pp. 1277–1280.

[54] D. F. Bacon, Y. Chen, D. Parkes, and M. Rao, "A market-based approach to software evolution," *Proceeding 24th ACM SIGPLAN Conf. companion Object oriented Program. Syst. Lang. Appl. - OOPSLA '09*, p. 973, 2009.

[55] J. Hamari and H. Sarsa, "Does Gamification Work ? — A Literature Review of Empirical Studies on Gamification," in *47th Hawaii International Conference on System Science*, 2014, pp. 3025–3034.

[56] D. Parsons, "Creating game-like activities in agile software engineering education," in *Proceedings of the Australasian Software Engineering Conference*, 2014.

[57] J. Thom, D. R. Millen, J. DiMicco, and R. Street, "Removing gamification from an enterprise SNS," *Proc. ACM 2012 Conf. …*, pp. 1067–1070, 2012.

[58] C. R. Prause, J. Nonnen, and M. Vinkovits, "A Field Experiment on Gamification of Code Quality in Agile Development," in *Psychology of Programming Interest Group (PPIG) Annual Conference 2012*, 2012, vol. 64, pp. 175–186.

[59] E. Herranz, R. Colomo-Palacios, A. Amescua Seco, and M. Yilmaz, "Gamification as a Disruptive Factor in Software Process Improvement Initiatives," *J. Univers. Comput. Sci.*, vol. 20, no. 6, pp. 885–906, 2014.

[60] R. Farzan and P. Brusilovsky, “Encouraging user participation in a course recommender system : an impact on user Computers in Human Behavior user behavior,” in *Computers in Human Behavior*, 2011, vol. 27, pp. 276–284.

[61] K. S. Rubin, *Essential Scrum: A practical guide to the most popular Agile process*. Addsion-Wesley, 2012.

Appendixes

Appendix A – List of Badges

A complete list of badges used in this solution is presented below.

Helper

Badge received when the Scrum Master intervene to help the team.

Planner

Badge received when a user attends to his first Sprint Planning meeting in a project.

Retrospecter

Badge received when a user attends to his first Sprint Retrospective in a project.

Reviewer

Badge received when a user attends to his first Sprint Review in a project.

Planning Project Lover

Badge received when a user attends to all Sprint Planning meetings during a project.

Project Lover

Badge received when a user attends to all Daily Meetings during a project.

Retrospective Project Lover

Badge received when a user attends to all Sprint Retrospective during a project.

Reviewing Project Lover

Badge received when a user attends to all Sprint Review during a project.

Team Planning Project Lover

Badge received when all team attend to all Sprint Plannings during a project.

Team Project Lover

Badge received when all team attend to all Daily Meetings during a project.

Team Retrospective Project Lover

Badge received when all team attend to all Sprint Retrospectives during a project.

Team Reviewing Project Lover

Badge received when all team attend to all Sprint Reviews during a project.

Sprint Lover

Badge received when a user attends to all Daily Meetings during sprint of a project.

Team Sprint Lover

Badge received when all team attend to all Daily Meetings during sprint of a project.

Story Killer

Badge received when a user completes all stories assigned to him in a sprint of a project.

Team Story Killer

Badge received when all team complete all stories in a sprint of a project.

Teamwork

Badge received by the team when the project is divided in small parts.

Team Sharing

Badge received when the team shares knowledge among members by several members contributing to complete the same story.

TOP Story Killer

Badge received when a user completes all stories assigned to him in a project.

Team TOP Story Killer

Badge received when the team complete all stories in a project.

Appendix B – Application Screens

This Appendix presents all application screens filled with the data gathered from the demonstration.

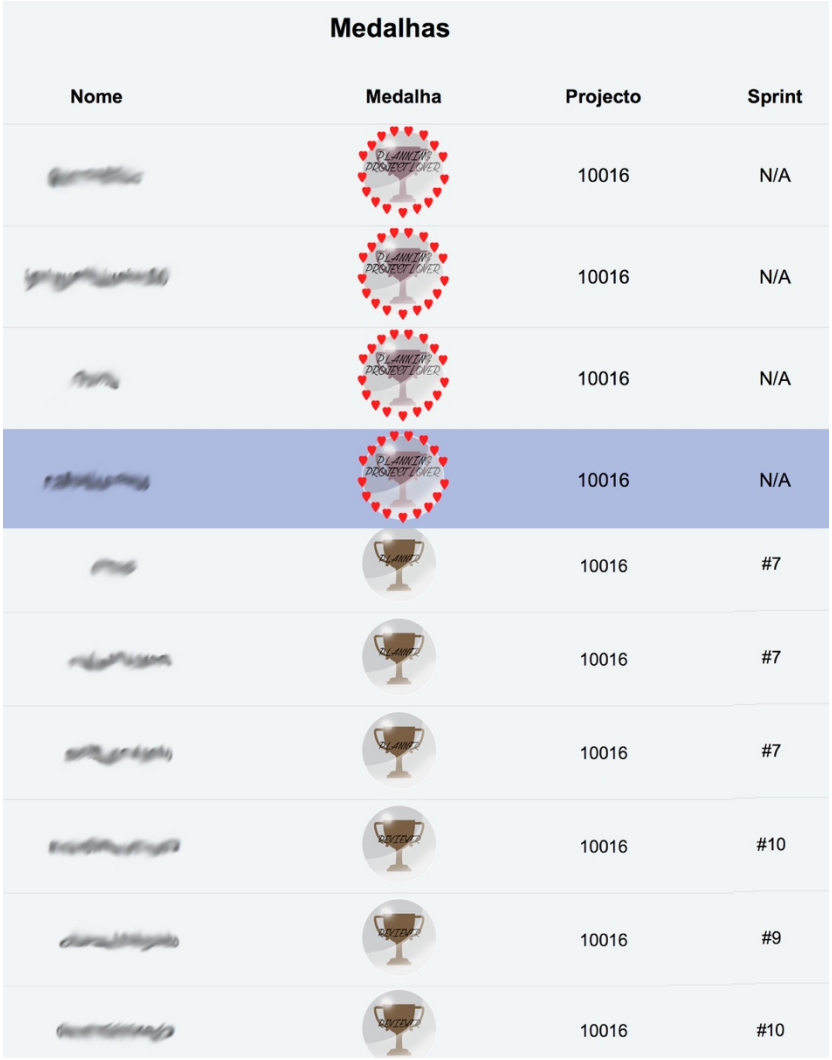
Login screen where users can login with their username to check their progress, points and badges.

Page to consult the leaderboard and personal badges.

Page where the user goes after clicking “?” in the page presented above, where the user is presented with a detailed explanation of his or her score.

ID	Descrição	Pontos
10	Compareceste em todos os Sprint Planning de um projeto.	40
7	Compareceste num Sprint Planning num sprint de um projeto.	40
11	Compareceste em todos os Sprint Review de um projeto.	100
8	Compareceste num Sprint Review num sprint de um projeto	40
8	Compareceste num Sprint Review num sprint de um projeto	40
8	Compareceste num Sprint Review num sprint de um projeto	40
12	Compareceste em todos os Sprint Retrospective de um projeto.	100
9	Compareceste num Sprint Retrospective num sprint de um projeto.	40
9	Compareceste num Sprint Retrospective num sprint de um projeto.	40
9	Compareceste num Sprint Retrospective num sprint de um projeto.	40

Page to visualize all users' badges. The user is presented with this page after clicking "Ver todos" in the page presented 2 pages before.



Medalhas

Nome	Medalha	Projecto	Sprint
[illegible]		10016	N/A
[illegible]		10016	N/A
[illegible]		10016	N/A
[illegible]		10016	N/A
[illegible]		10016	#7
[illegible]		10016	#7
[illegible]		10016	#7
[illegible]		10016	#10
[illegible]		10016	#9
[illegible]		10016	#10

Page where the administrator can login with a password to edit the points given for each behavior.

Page where the administrator can edit the points attributed to users for each behavior.

Bem-vindo ao painel de administração do jogo!

Escolha o comportamento que quer editar, escrevendo o número de pontos que quer oferecer por comportamento nas caixas de texto à direita.

(Para ver os pontos atribuidos anteriormente, veja o valor à esquerda da caixa de texto.)

ID	Comportamento	Pontos
1	Uma pessoa participar num Daily Meeting num projecto.	20
2	Uma pessoa ir a 5 Daily Meetings (pontos/meeting).	20
3	Uma pessoa ir a 10 Daily Meetings (pontos/meeting).	20
4	Uma pessoa ir a 15 Daily Meetings (pontos/meeting).	20
5	Uma pessoa ir a todos os Daily Meetings durante um projeto	40
6	Uma pessoa ir a todos os Daily Meetings durante um sprint.	40
7	Uma pessoa ir a 1 Sprint Planning num sprint de um projeto.	40
8	Uma pessoa ir a 1 Sprint Review num sprint de um projeto	40
9	Uma pessoa ir a 1 Sprint Retrospective num sprint de um projeto.	40
10	Uma pessoa ir a todos os Sprint Planning de um projeto.	40
11	Uma pessoa ir a todos os Sprint Review de um projeto.	100
12	Uma pessoa ir a todos os Sprint Retrospective de um projeto.	100
13	Uma equipa dividir o projeto em pequenas partes.	100
14	Uma pessoa terminar todas as histórias num sprint de um projecto.	100
15	Uma equipa terminar todas as histórias num sprint de um projecto.	200
16	Uma pessoa terminar todas as histórias num projecto.	100
17	Uma equipa terminar todas as histórias num projecto.	200
18	Uma equipa partilhar conhecimento durante um projecto.	100
19	O Scrum Master ajudar a equipa a superar um impedimento.	100

ID: Pontos:

Submeter

Sair