Empirical Software Engineering
https://doi.org/10.1007/s10664-020-09816-9

A gamification solution for improving Scrum adoption

Rita Marques[1,2] • Gonçalo Costa[1,2] • Miguel Mira da Silva[1,2] •
Daniel Gonçalves[1,3] • Pedro Gonçalves[4]

Published online: 15 April 2020

Abstract

Despite the wide adoption of agile methodologies, software development teams still struggle to meet requirements, partially due to practitioners' lack of motivation to apply agile techniques in practice. We propose a gamification software tool to make Scrum techniques more fun and engaging for practitioners. This paper presents the results of the first iteration of a larger research effort that follows the Design Science Research methodology, in which a prototype was developed as a Jira Software app and evaluated with a real-world Scrum team. The results suggest that gamification can positively impact practitioners' motivation by changing the atmosphere within the team, even if it does not contribute to the improvement of Scrum practices adoption. The metrics corresponding to Scrum techniques slightly increased after using the app, but this result probably cannot be attributed to practitioners' improved motivation. Quantitative analysis and interviews with the team members showed that the gamified experience should be more challenging. We conclude that gamification's potential can be better achieved if more and better studies are conducted based on mixed methods, even if the conditions are not ideal. The app has been improved based on the feedback received, and currently is being evaluated with other Scrum teams.

Keywords Gamification · Software development · Software process · Agile · Scrum · Motivation

Communicated by: Emerson Murphy-Hill

✉ Rita Marques
rita.marques@tecnico.ulisboa.pt

Gonçalo Costa
goncalo.m.da.costa@tecnico.ulisboa.pt

Miguel Mira da Silva
mms@tecnico.ulisboa.pt

Daniel Gonçalves
daniel.goncalves@inesc-id.pt

Pedro Gonçalves
pedro.goncalves@xpand-it.com

Extended author information available on the last page of the article

1 Introduction

The Standish Group's CHAOS Report found that 64% of the projects studied between 2013 and 2017 were either failed or challenged (Standish Group 2018). This reveals that while several practitioners and researchers have been studying for many years how to properly develop software, understanding how to avoid software project failure is still a challenge. Thus, software development organizations have been adopting new tools and methodologies (Overhage et al. 2011), proven to improve overall success of software projects (Riemenschneider et al. 2002).

Agile software development emerged as a flexible, responsive, and team-empowering response to traditional software development and project management (Williams and Cockburn 2003), where teams are intended to produce working software during short iterations. There is evidence that agile projects are twice more likely to succeed and one-third less likely to fail when compared against a traditional methodology (Standish Group 2018). Additionally, agile software development is associated with greater satisfaction than more traditional approaches (Kropp et al. 2018).

While the benefits of agile methods and techniques, such as improved product quality and customer satisfaction, have been demonstrated, agile teams still face challenges, mostly related to human factors (Conboy et al. 2010; Hajjdiab et al. 2012). Such challenges result in an overall poor adoption of agile practices by software development practitioners, which can be partly explained by practitioners' lack of motivation to apply agile techniques in practice (Conboy et al. 2010; DeMarco and Lister 2013).

Motivation is what moves us to do something (Werbach and Hunter 2012), and can be classified either intrinsic or extrinsic, when a person performs an activity because it is inherently interesting or because it leads to a separable outcome (like bonuses or remuneration), respectively (Ryan and Deci 2000b). Motivation is strongly linked to engagement, which refers to the degree of involvement and immersion a person has while performing an activity (Alsawaier 2018), and they are at the core of the Self-Determination Theory (SDT). SDT is based on three psychological needs: competence (sense of controlling the outcome and feeling mastery; related to the concept of flow, a state of mind when one is totally absorbed on an activity (Csikszentmihalyi 1990)), relatedness (sense of interacting and be connected with others), and autonomy (sense of being the source of one's own behaviour) (Ryan and Deci 2000a). This theory also states that extrinsic motivators are effective if they promote these needs. Overall, mostly important is to understand the different motivators and how to foster them.

There is evidence linking that using gamification mechanisms can positively impact motivation and engagement (Alsawaier 2018). Gamification is a recent but popular approach that has been used to understand different motivators and how to foster them to engage and motivate people to adopt new behaviours (Werbach and Hunter 2012).Gamification adds game elements and game design to non-game processes, and aims at making activities related to real-world problems and goals rewarding for themselves, thus creating incentives without incurring into high costs. Despite being related to gaming, gamified systems are not full-fledged games; they use particular parts of games (i.e., game elements) in an already existing process (Deterding et al. 2011). Werbach and Hunter propose a set of games that can be applied in a gamification solution, divided into three categories with different levels of abstraction (Werbach and Hunter 2012). A successful gamified experience need not incorporate all possible game elements, but rather only those that better serve the defined goals.

Besides being successfully applied in fields like education (Barata et al. 2017) and health (Marques et al. 2017c), there is evidence that gamification applied to the workplace can lead to significant improvements in employee engagement (Reeves and Read 2009). Particularly, researchers and practitioners have been applying gamification in software development to increase team motivation and help practitioners to define better goals and to focus on them (Yilmaz and Connor 2016).

Given this evidence, we decided to conduct a longitudinal study to analyse the impact of gamification in the adoption of agile practices. Even though many companies are already implementing agile with success, this topic remains important as some companies are still maturing their adoption. Additionally, not only researchers are trying to understand how to address challenges affecting a proper agile adoption, but also trying to integrate agile with game design since the early 2010s – the same period when gamification started emerging (Hoda et al. 2018).

To narrow down our research, we decided to focus only on Scrum, which is the most adopted of the agile frameworks by companies and practitioners. In the 13th State of Agile report, 64% of the respondents stated that their organizations use Scrum or a hybrid approach (Version One 2019). In the Swiss Agile Study conducted in 2016, the framework is used by 69% of the companies represented in the study (Kropp and Meier 2017).

In Scrum, work is visible to the entire team, so that all practitioners always know what is happening in the project and can adjust their practices to achieve the desired goals (Schwaber 2004). Practitioners following Scrum are organized in small teams, plan and track their work iteration (called "Sprint") based on Scrum artefacts, implement user stories (that represent the requirements), and communicate in Scrum events (Schwaber and Sutherland 2016). Moreover, these teams should be cross-functional, and self-organizing. In the remainder of this paper, the term "practitioner" will be used to refer to any of the roles that a software engineer can have (such as analysts or programmers) within a project. To the best of our knowledge, very few gamification proposals were evaluated with Scrum teams in practice, despite the importance of validating these initiatives' impact on people's motivation.

As a result, we collaborated with a Portuguese company to create a gamification solution aiming at improving practitioners' motivation to adopt Scrum practices. We have been following a Design Science Research Methodology (DSRM), whose iterative process is allowing us to incrementally design, develop, test, and evaluate a solution that is aligned with our end users' needs (Peffers et al. 2008).

In the first iteration of this study, we created a conceptual proposal that is aligned with the Scrum Guide (Schwaber and Sutherland 2016) and that addresses the gaps of previous related works. This proposal was implemented as an app for a popular software management tool and demonstrated and evaluated with a Scrum team from the same company. This case study was mainly exploratory, and we are currently demonstrating the improved proposal with two teams in a different company. In a near future, we will start the third iteration of our research work with other teams.

Overall, this research effort contributes to the knowledge base not only because it is one of the few studies evaluating a gamification solution for Scrum with teams in companies, but also because of its distinctive characteristics, such as the usage of varied game elements whose selection was strongly rooted on perceptions and experiences of real practitioners.

In our previous research work, we presented the initial proposal (Marques et al. 2017b) and discussed some of the results observed (Marques et al. 2018a, b). In this paper, we elaborate and extend the prior research work by:

- Including a more detailed description of the research method;
- Presenting and discussing a review of relevant works studying the challenges affecting a proper Scrum implementation, and possible solutions to address them;
- Presenting and discussing the results of a pre-study questionnaire conducted with practitioners to assess practitioners' motivation in following Scrum practices;
- Presenting the diagram that conceptually describes the proposed solution, along with a more specific description of the different game elements applied;
- Discussing further the functionalities of the software app implemented, complemented with many screenshots that have not been published yet;
- Providing an in-depth statistical analysis of the baseline study data, which allowed to further understand the team's Scrum practices; and
- Including a completely new lessons learned section that goes deeper in the results and discusses them in different perspectives, particularly in the form of lessons learned that can be considered in future gamification initiatives.

Based on a quantitative analysis using the data collected from the software management tool before and after using the app, and a set of interviews with the participants, in this study we address the following research question: *Does Gamification contribute to improving adoption of Scrum practices by practitioners?*

This case study showed that gamification can positively impact practitioners' motivation by changing the atmosphere within the team, even if it does not contribute to the improvement of Scrum practices adoption. The results suggest that the metrics corresponding to Scrum techniques slightly increased after using the app when compared to the baseline, but probably this result cannot be attributed to practitioners' improved motivation. Team members mentioned that the proposal should be more challenging, something that could be easily achieved by allowing organizations to adapt the rewarded behaviours and game elements.

That said, gamification seems to have the potential to motivate people to adopt Scrum practices, but much more research should be conducted with different teams to understand how game elements can impact practitioners' motivation.

We start this paper with a description of the methodology and specific methods used to conduct the work. After presenting a review of relevant works, linked with both the study of challenges affecting Scrum implementation and implementation of gamification in software development, we describe the results of a questionnaire conducted with practitioners. Next, we present the proposal and its design and development, followed by a discussion on the demonstration and evaluation activities conducted with the Portuguese software development company. The paper closes with a discussion of the results and some conclusions and future work.

2 Methods

To accomplish the proposed objective, DSRM was adopted as the methodology to conduct this research, because it is based on an iterative process. This allows to incrementally design, develop, test and evaluate a solution that is aligned with the organization and end users' needs (Peffers et al. 2008).

Hevner et al. have established a set of seven guidelines for a DSRM project (Hevner et al. 2004), which incorporate six iterative phases proposed by Peffers et al. (Peffers et al. 2008). Tasks carried out and correspondent methods used during this study are presented in Table 1 and mapped to the phase where they were performed.

This paper will cover activities 1 and 2, but the focus is to present the methods and results from activities 3, 4, and 5 conducted during the first iteration of the work. This paper itself is part of phase 6, which is not further extended here.

We started by defining the problem and the objectives through the analysis of related works and a survey with software development practitioners during phases 1 and 2. The main output of a DSRM project must be an artefact, which was initially developed in phase 3. Then, it was demonstrated and evaluated iteratively during phases 4 and 5 to obtain information that allows its continuous improvement in next iterations' phase 3. This was done based on a quantitative analysis of the project's data and interviews with the Scrum team members, whose results were evaluated against design and development decisions. Under the scope of this work, one artefact was implemented: an instantiation (i.e., a prototype system).

2.1 Problem Identification and Goals for a Solution

The starting point for the artefact development was recognizing the link between poor adoption of Scrum practices and practitioners' lack of motivation to apply those practices, which is a subject of human behaviour. We then proposed to use gamification as a solution to promote a change of these behaviours in order to achieve the desired outcomes, and studied other works applying gamification in software development to understand their strengths and gaps. With this, we derived more concrete goals based on Scrum's specification and the challenges affecting Scrum implementation identified through an analysis of related work and a questionnaire, whose methods are explained in the following subsections.

2.1.1 Related Work

Works related to this study was searched using Google Scholar. We started by searching for challenges affecting the adoption of this framework. Keywords selected were: ("Scrum" AND

Table 1 DSRM phases and tasks to perform, mapped on this study's iteration

DSRM Phase	Method/Tasks
1. Problem identification and motivation	Analysis of related work and questionnaire with software development practitioners
2. Defining the objectives for a solution	
3. Design and development	Design of the artefact
4. Demonstration	Quantitative analysis based on data extracted from a real-world project and interviews
5. Evaluation	Analysis of activity 4's results, and assessment against design and development decisions
6. Communication	Done during the project, through paper or conference papers such as this

"software development" AND ("adoption" OR "challenges" OR "problems")). Initial criteria for inclusion was the presence and discussion of challenges in Scrum practices implementation in software development organizations. As only four papers were returned, we extended the search to papers addressing agile in general (i.e., switched the keyword "Scrum" with "Agile").

Then, we searched for works applying gamification to the Scrum framework and agile in general, given that such results could also be important for this work. Keywords selected were (("Gamification" OR "Games" OR "Gamifying" OR "Gamified") AND "Scrum" AND "Agile"). Initial criteria for inclusion was the description and discussion of gamification solutions to improve Scrum and agile practices. Because only five papers were returned, the search was then extended to papers applying gamification to software development processes, and then again to software development in general (i.e. we switched "Scrum" and "Agile" keywords with "Software Development").

2.1.2 Questionnaire

The questionnaire used to assess practitioners' motivation in following Scrum practices was prepared with support from a commercial tool, and can be consulted in Appendix 1. It is composed by 20 questions: seven demographic and 13 to assess respondents' compliance and feelings regarding Scrum, including time spent performing Scrum activities, importance given and motivation to perform these activities, and reasons for a lack of motivation in implementing Scrum.

The questionnaire was disseminated by e-mail to some of the contacts of the Portuguese company we cooperated, in a total of 397 practitioners (managers at different levels and developers). Respondents are distributed in several organizations around 30 countries.

2.2 Design and Development

The conceptual part of the artefact was build based on the outputs from DSRM activities 1 and 2, as described in the previous section. Then, the proposal was implemented as an app for a software management tool, Jira Software,[1] so that practitioners do not need to use a separate tool for gamification. This way, we can demonstrate that the proposal can be used to solve the research problem. This tool was chosen not only because it supports most of the Scrum framework, but also due to its flexibility, stability, and strong user base. In fact, 65% of the respondents of the State of the Agile survey in 2018 stated to use Jira to manage their agile projects. Jira Software supports the development process and is based on "issues" i.e., problems that need to be solved (e.g. a bug). From now on, we will use the term "issue" instead of "task".

2.3 Demonstration

In this step, the proposed solution must be validated to show that it can be used to solve the research problem. In this study, this means studying the usage of the gamification solution by a Scrum team in practice. First, a baseline study is conducted on historical data extracted from the team's Jira Software instance. Next, the gamification app is installed, and a field study is

[1] Jira Software: https://www.atlassian.com/software/jira/features (Accessed: 14/11/2019)

conducted for some time. A quantitative analysis is then performed on both datasets, and results of each study are compared against each other. Finally, the analysis is complemented by interviewing team members. Thereby, quantitative results allow to measure behaviours and understand if and how they have changed, but the insights collected through interviews can further explain some of those results.

This paper presents the demonstration activities performed in the company we partnered with, which provides software products and services, and manages their software development processes with Jira Software. The participants work in ProjectX (anonymous name), one of the company's projects managed with Scrum (which is adapted to their needs, complying with 80–85% of Scrum practices) and where a software product is being developed. The team is composed by six people and follows sprints with variable duration (between one and four weeks). Critical bugs detected in released versions of the product are fixed in special versions with no sprints. Such issues were left out the quantitative analysis but are discussed later in the paper. These workers have experience in using Jira Software on user and/or developer level, which can increase the quality and relevance of their suggestions.

Two field studies were conducted to analyse the proposal. Yet, the first field study was discarded since the app contained a critical bug by then: testers and the Scrum Master progressed way more than other players, as they resolve most issues. The app was refined so that developers receive points when the issues' status changed from "In Progress" to "Waiting for Testing" (i.e., when they finish implementation), and the database was reset before starting the next field study. Thereby, historical data from the baseline study and results of the second field study (referred to as "field study" further in this paper) conducted with the team were compared against each other. In this period the team changed, with some practitioners leaving and others joining in.

2.3.1 Quantitative Analysis

A method was defined to extract, clean, and process data from a Jira Software instance to calculate the target metrics (presented in subsection 5.2). This method was applied to both the baseline and field studies. First, a Python script collects all issues' information from their individual JSON file, filters out irrelevant fields, and exports data to a Comma-Separated Values (CSV) file (CSV1). With this, we get each issue's lifecycle information. Due to Jira Software files' complexity, and as most metrics are not reflected in this raw data, Pentaho Data Integration[2] was used to process data stored in CSV1. A Pentaho process was built based on "steps" (i.e., operations), starting by importing CSV1 and cleaning data (define the right data type for each field and resolve NULL values). Next, "Filter Rows" steps are applied to filter out useless data, and "Select Fields" steps select the values to compute each metric, which are later used by "Group By" and "Row Denormalize" steps to compute metrics on sprint level. All metrics are lastly merged and grouped by sprint in a table with "Merge Join" steps, which is exported as a CSV file (CSV2) later used in the analysis. An analogue process was built for the assignee (i.e., practitioner) context, but computes metrics per assignee.

[2] Pentaho Data Integration: https://community.hitachivantara.com/s/article/data-integration-kettle (Accessed 14 /11/2019).

2.3.2 Interviews

Four team members were interviewed after conducting the field studies (the other two were off work due to medical leave). The interviews were semi-structured which, despite the existence of a guideline (which can be found in Appendix 2), allowed for flexibility and improvisation, clearing the path to explore emerging lines of research. The interview guide was organized in six sections: interviewee's profiling; level of usage and opinion about Jira Software; experience with Scrum; experiences with games/gamification; insights about the proposal; and final considerations to sum up the interview. One specific question was defined for each of the roles Scrum Master and Product Owner.

The interviews were conducted on the same day, at ProjectX's company's office and took around 40 min each. One researcher conducted the interview and took notes using the interview guide as a data collection template. Immediately afterwards, those notes were revised to create a final script.

2.4 Evaluation

In this step, we observe and analyse the results from the demonstration activity to understand to what extend the artefact can be used to solve the research problem. This includes comparing the goals of the solution to the observed results from the use of the artefact in the demonstration.

The statistical analysis of the data extracted from the Jira Software instance is supported by a Jupyter Notebook,[3] using pandas[4] and SciPy[5] python packages. A Shapiro-Wilk normality test was applied to all metrics from the baseline and field studies. Following these tests' results, all statistical differences between groups were checked using a non-parametric Mann-Whitney's U test.

Furthermore, some more in-depth statistical study was conducted on the baseline dataset, for example by finding correlations within the data using the Pearson correlation coefficient for each case. This is a value between −1 and 1 that can represent a positive, negative, or null correlation, which can different strengths: very weak (0–0.19); weak (0.2–0.39), moderate (0.4–0.59); strong (0.6–0.79); or very strong (0.8–1). Nevertheless, this analysis could not be repeated in the field study, given the reduced size of the data samples.

Gamification data was analysed by directly querying the database, using the appropriate management tool.

3 Related Work

In this section, after introducing the Scrum framework, we discuss relevant works whose results should be considered while designing the proposal. Being this paper's focus on the adoption of Scrum in organizations, we studied the specific challenges affecting the adoption of this framework, along with some initiatives that other authors applied to cope with those

[3] Jupyter Notebook: http://jupyter.org/ (Accessed 14/11/2019).
[4] pandas: https://pandas.pydata.org/ (Accessed 14/11/2019).
[5] SciPy: https://www.scipy.org/ (Accessed 14/11/2019).

challenges. Then, we analysed studies where gamification was applied in the field of software development in general, and not only to Scrum and agile, given the reduced number of works directly related to this framework. This way, we were able to understand how other authors have used this approach and learn from their results before designing and developing our artefact.

3.1 Scrum

Scrum is the most widely adopted agile methodology (Kropp and Meier 2017; Version One 2019), where work is visible to the entire team, so that every practitioner always knows what is happening in the project and can adjust their practices to achieve the desired goals (Schwaber 2004). Scrum teams should be self-organizing, thus deciding how to best accomplish their work, rather by being directed by others outside the team. They should also be cross-functional, meaning they are able to perform the work without depending on people or entities external to the team.

When a project starts, the Product Owner creates a Product Backlog with user stories covering all system's requirements, and orders them by priority. Stories must contain all the information needed so that everyone in the team understands how to perform the work. Also, the team must estimate the effort needed to complete each story, using more direct measures like the time estimated to complete the work, or story points. A Sprint starts with the Sprint Planning, where user stories with high priority are selected to build the Sprint Backlog. Stories are broken down into tasks and assigned to each member of the development team. The goal is to complete all stories in the Sprint Backlog by the end of the sprint, to produce a complete increment.

During the sprint, every day a Daily Meeting is conducted to assess project status, and at the end of the sprint an increment must be completed. The sprint finishes with Sprint Review, where the increment is presented and discussed, and Sprint Retrospective, were lessons learned are discussed to improve the following sprint.

In summary, it can be said that Scrum includes a set of roles (product owner, Scrum master and development team), ceremonies (sprint, sprint planning, daily meeting, sprint review, and sprint retrospective) and artefacts (product backlog, sprint backlog, and increment). All of these elements interact during the software development process.

3.2 Challenges Affecting Scrum Implementation

Works analysing agile implementation's challenges focused on specific contexts, either agile principles or software development process' areas, were included, although discarding issues falling in areas other than project management (such as code quality). Papers addressing global development agile were not considered, as they mostly revolve around the fact that teams are geographical dispersed (Hossain et al. 2009; Alzoubi and Gill 2014). This returned more nine papers: four addressing agile practices, in general, and five presenting challenges focused in specific agile principles or software development process' areas.

The authors of the papers that were not available online were contacted to request the documents. The content of the available papers was examined and only those relevant for this study were maintained. References of these papers were analysed to find other relevant papers.

Table 2 Summary of problems in Scrum implementation identified during the analysis of related work, listing problems referenced in at least two papers

	(Overhage et al. 2011)	(Marchenko and Abrahamsson 2008)	(Cho 2008)	(Hajjdiab et al. 2012)	(Turk et al. 2002)	(Dybå and Dingsøyr 2008)	(Conboy et al. 2010)	(Version One 2019)	(Dikert et al. 2016)	(Inayat et al. 2015)	(Stettina and Heijstek 2011)	(Hoda and Murugesan 2016)	(Sekitoleko et al. 2014)
Poor communication or collaboration between teams/customers			X		X	X	X		X				
Insufficient documentation	X		X							X	X		
Lack of senior management support				X				X	X			X	
Difficulty in estimating tasks and tracking process		X							X			X	
Resistance to adopt agile	X							X	X				
Lack of focus on processes and tools		X					X						
Difficulty in integrating non-development functions		X							X				
Ineffective Scrum ceremonies			X	X									
Customer unavailability			X							X			
Role boundaries are diffuse, especially middle managers							X		X				
Personal performance rewarding acts against team-centric agile thinking							X		X				
Company and agile values not aligned								X	X				
Inconsistent implementation of agile or Scrum								X	X				
Dependencies between tasks												X	X
Excessive senior management interference on the process		X										X	

After the search, each paper was analysed and all challenges (i.e., problems hampering correct implementation of Scrum practices) found were listed. Finally, all factors were compared, ordered by frequency, and those referenced at least twice (for simplicity reasons) were summarized in Table 2.

Overhage et al. identified documentation neglecting and resistance to change as inhibitors for developers' Scrum acceptance, which difficult requirements verification after development and affect voluntariness in using Scrum, respectively (Overhage et al. 2011).

Marchenko and Abrahamsson identified challenges affecting Scrum adoption in a company's department working in a multi-team multi-project environment, including lack of discipline for not focusing too much on processes and tools; forcing the methodology too much; difficulty in tracking progress (which could help improving task estimation); difficulty in integrating non-development functions (such as research); and excessive interference of senior management on the process (Marchenko and Abrahamsson 2008).

Based on interviews with practitioners, Cho presents a case study in a company employing Scrum for many projects. Interviews stated that insufficient documentation, solely based on user stories and code; Scrum ceremonies that do not reflect project's complexity; lack of communication between teams and with customers; and a customer not fully involved can negatively impact Scrum implementation (Cho 2008).

A case study of an unsuccessful Scrum introduction in a government entity is analysed by Hajjdiab et al. (Hajjdiab et al. 2012). In addition to challenges particular to Scrum's introduction or government institutions, absence of a Scrum Master; lack of Scrum knowledge; Scrum meetings constituting an overhead for an already pressured team; and lack of support from upper management contributed to the initiative's failure.

Next, we analyse the works addressing agile practices in general in the lenses of Scrum practices' implementation flaws.

Based on a previous analysis of agile processes papers and their experience in agile projects, Turk et al. argue that agile success can be harmed in large teams, because a high number of communication channels have to be maintained (Turk et al. 2002).

Dybå and Dingsøyr present the results of a systematic literature review of empirical studies on agile software development (Dybå and Dingsøyr 2008). The problem of reduced communication inter agile teams is an example used to illustrate the need for an agile implementation to focus on human and social factors. The authors highlight the need for more empirical studied on agile software development, especially in popular methodologies as Scrum, as most analysed studies were based on a single agile methodology.

Senior personnel interviewed by Conboy et al. identified the difficulty in decentralize decision-making; diffuse role boundaries; poor business knowledge of customers; deficiencies in communication, due to practitioners' lack of social skills and fear that transparency might highlight their limitations; practitioners' lack of motivation to use agile methods; lack of focus in agile principles, and the existence of a personal performance rewarding, acting against team-centric agile thinking, as challenges affecting agile software development (Conboy et al. 2010).

In Version One's 13th annual State of Agile report, lack of alignment between company and agile values; resistance to change; lack of management support and Product Owner availability; lack of agile experience and training; and inconsistent agile practices and process were identified as challenges to agile adoption and scale (Version One 2019).

Systematic literature review on challenges of a large scale agile adoption in industry settings, presented in Dikert et al.'s work, uncovers agile poor process customization and different implementations; difficulty in integrating non-development functions; resistance to adopt agile methodologies; difficulty in creating and estimating user stories; lack of support from upper management; diffuse roles for middle managers; lack of communication between agile teams; and keeping a reward system based on personal performance as challenges. Most analysed studies were experience reports, highlighting a need for further research (Dikert et al. 2016).

Inayat et al. present a systematic literature review on challenges faced by agile teams regarding requirements engineering (Inayat et al. 2015). Neglecting non-functional requirements, like security (due to a focus on user stories) and design and architectural issues (which can lead to rework); insufficient documentation based on user stories; requirements' susceptibility to change, which difficult budget and time estimations; and customers' unavailability, which can increase rework, were found to negatively impact agile practices' implementation.

In Stettina et al.'s work, which addressed agile practitioners' perceptions regarding documentation in their projects, it was found that practitioners dislike performing documentation activities, but they recognize their importance (Stettina and Heijstek 2011). They also believe that too little documentation is available, which can cause loss of knowledge.

Project management challenges stemming from self-organization principle in agile teams were identified in Hoda and Murugesan's work (Hoda and Murugesan 2016). These challenges are the constantly changed or delayed requirements, affecting team's ability to estimate effort; difficulty in convincing senior management to adopt agile (who, even after acceptance, might interfere too much on the process); difficulty in performing the time-consuming process of estimation; difficulty in achieving cross-functionality (which forces practitioners to fall out their comfort zone); lack of motivation to assume autonomy; lack of tasks' acceptance criteria; and the existence of dependencies between tasks. Authors argue that challenges are likely to affect each other.

Sekitoleko et al. describe a case study that investigated challenges linked with technical dependences in agile software development (Sekitoleko et al. 2014). Problems in project planning can generate technical dependencies that affect development (like splitting a task between teams instead of allocating it to one team), leading to a need to balance priorities and reschedule work. Also, knowledge sharing can be threatened by practitioners' attitudes: people might not want to share, not be good at explaining, and be too shy to ask or provide help. Moreover, knowledge can be lost if practitioners do not retain what was discussed in a meeting. Finally, authors discuss that challenges affect each other, leading to a domino effect.

These authors also analyse solutions to the discussed challenges, which emerged as lessons learned from past experiences. From the many solutions proposed in these works, we believe that some align better with a gamification implementation than others, as they do not require major management decisions or organizational changes. These include:

- Write more documentation, which can backup communication (Conboy et al. 2010);
- Implement a voting system to empower team decision-making (Conboy et al. 2010);
- Observe and validate agile practices between teams (Conboy et al. 2010);
- Adopt tools to support the development and provide training on how to use them (Hoda and Murugesan 2016);
- Provide continuous feedback, even outside meetings (Dybå and Dingsøyr 2008; Conboy et al. 2010);

- Implement team-based rewards, instead of individual rewards (Conboy et al. 2010);
- Implement peer-review (Conboy et al. 2010).

Despite the evidence that lack of agility in software development processes can hamper their implementation, results presented in this subsection suggest that simply adopting an agile methodology like Scrum does not remove all problems. Also, a comparison between these results against those of a previous study on challenges affecting software development methodologies (Marques et al. 2017a) suggests that some challenges might not be Scrum-specific.

This analysis revealed a need for more and better empirical research on challenges affecting agile practices' proper implementation, particularly on methodologies like Scrum that are widely adopted in industry settings. Literature also suggests that challenges are interrelated, i.e. the occurrence of one challenge might cause another to happen. Although many articles were published more than five years ago, recent works still report some of those challenges, which mainly lead to rework and delays.

3.3 Gamification in Software Development

Because software development processes are brain- and collaborative-intensive, comprising some tedious activities, gamification can help making such activities more fun and attractive (Pedreira et al. 2015). Some research has already been conducted in this field, with different focus. Some authors tackle the main subject by discussing and proposing methodologies to apply gamification in software engineering (Passos et al. 2011; Dubois and Tamburrelli 2013; Dal Sasso et al. 2017) and to foster software process improvement initiatives (Dorling and McCaffery 2012; Herranz et al. 2014, 2015). Additionally, the authors of a literature review about the use of gamification in software development processes concluded that, despite the many gamified tools to support different activities, no tool supports the whole process (Lombriser and van der Valk 2011). In another literature review about the same subject, the authors conclude that gamification is being mainly implemented during coding and testing phases of the software development process (Platonova and Bērziša 2017).

Closely related, a framework mapping collaboration issues affecting software development teams with target behaviours and game elements to mitigate those issues was proposed (de Castro 2016).

Moreover, some works are targeted to specific software development processes, like the gamified version control system to encourage Computer Science students to commit more frequently (Singer and Schneider 2012), the reputation system to improve the quality of collaboratively written code through documentation (Prause et al. 2012), and the gamification model to motivate developers to pay more attention to good coding practices (Foucault et al. 2019).

Finally, some studies directly address the application of gamification in agile development and Scrum. McClean added a lottery element to the agile process: practitioners could win a reward, and their chances increased with the number of tasks completed (McClean 2015). Yilmaz and O'Connor proposed an integrated gamification approach for Scrumban, where practitioners received points and badges for finishing tasks and helping each other (Yilmaz and Connor 2016). Loriggio presented a methodology for teaching Scrum, supported by gamification and other theories (Loriggio et al. 2013).

Češka prototyped a gamified app to support Scrum development, composed by game elements like points, badges, and progression (Češka 2016). Scrum Hero is a gamification framework to support Scrum software development projects' management, based on game elements like narrative, quests, and rewards (Souza et al. 2016).

While only the last group of works directly address gamification for agile/Scrum, we can learn from all of them, as they target the same players (i.e., practitioners). In general, these works lack a proper empirical validation in the industry: some are just conceptual proposals; others are only evaluated with qualitative methods; and others are evaluated in alternative contexts (like education) using samples too small and time frames too short to support important conclusions. Furthermore, proposed solutions did not go far beyond the simplest elements (like points and badges) and are not integrated in the tools practitioners use daily.

Apart from research, some commercial tools, like Jiraffe[6] and Gamification for Jira,[7] are available, but there are no studies publicly available evaluating the application of these tools in industry context. Additionally, there are some physical and digital games that support the training and learning of agile frameworks, like ScrumKnowsy,[8] getKanban,[9] or Lego4Scrum.[10]

Summing up, while gamification studies to increase Scrum adoption are emerging, there is much room for improvements, namely regarding diversity of game elements and empirical validations in industry.

4 Questionnaire

The questionnaire had a response ratio of 0.20 and 44 valid responses. Most respondents were Portuguese (89%) male (98%) working in IT (91%). Around 30% of the practitioners are individual contributors. Thus, this questionnaire mostly addresses Portuguese developers' perceptions, and its results are valuable because they can be generalized for the Portuguese reality.

The degree to which teams apply Scrum practices was classified as either sufficient (47%) or good (44%). Respondents identified lack of Scrum knowledge (38%), excessive time-consuming Scrum events and activities (34%), and discomfort of people lacking communication skills (31%) as main reasons jeopardizing motivation in applying Scrum best practices.

All Scrum events were mostly considered important or, classified in the following order: Sprint planning (97%); Daily meeting (84%); Sprint review (75%); and Sprint retrospective (72%). Similarly, motivation to attend those events was rated in the same order: Sprint planning (84%); Daily meeting (75%); Sprint review (66%); and Sprint retrospective (66%). This suggests that practitioners are more motivated to attend Scrum events they perceive as important.

Creating user stories (94%) and sprint backlogs (84%); prioritizing tasks (91%); and estimating task effort (84%) are perceived as important activities, but less respondents stated to be motivated to perform them (in the same order: 66%, 56%, 69%, and 69%). This represents a paradox, as these activities are part of Sprint Planning - a Scrum event respondents

[6] Jiraffe: http://bugpotion.com/ (Accessed: 14/11/2019)

[7] Gamification for Jira: https://marketplace.atlassian.com/apps/1220811/gamification-for-jira/ (Accessed: 14/11/2019)

[8] ScrumKnowsy: http://www.scrumknowsy.com/home/ (Accessed: 23/11/2019)

[9] getKanban: https://getkanban.com/ (Accessed: 23/11/2019)

[10] Lego4Scrum: https://www.lego4scrum.com/ (Accessed: 23/11/2019)

stated to be motivated to perform. A further research might reveal if and how this inconsistency threatens the validity of the questionnaire results.

5 Proposal

This study proposes a gamification solution to make Scrum techniques more fun and engaging for practitioners, while attempting to fill the gaps of existent literature. The solution described in this section was built based on the findings discussed in sections 3 and 4, aiming to be a general-purpose solution that can be applied in many software development organizations.

As gamification design should be supported by some kind of process, this solution was designed by following 6D Framework, an iterative game design process composed by six steps (Werbach and Hunter 2012), as it is one of the most mentioned and more complete frameworks to formalize the gamification design process (Mora et al. 2015).

5.1 Define Objectives

The solution's objectives, which should bring real benefit to the organization, are defined here. The main goal is to increase practitioners' motivation to apply Scrum techniques, which in turn might positively influence software projects' success. From here, we derived more concrete goals, as we discuss in subsection 2.1. These goals are based on Scrum's specification and the analysis of practitioners' perceptions and experiences discussed in sections 3.2 and 4, and allowed to build a simple yet consistent tool without requiring major management decisions or organizational changes: improve tasks specification quality; reduce the percentage of unassigned tasks; reduce the number of uncompleted tasks per sprint; increase the number of effort estimated tasks; increase participation in events (which we will refer to as "meetings" in this paper); increase team cooperation with team-centric goals and rewards; and implement project tracking with continuous feedback based on relevant metrics.

5.2 Delineate Target Behaviours

Just like the objectives, the behaviours we want players to perform and the metrics for measuring them were defined so that they directly corresponded to different Scrum techniques, thus allowing us to understand if and how behaviours changed after gamification. Metrics were defined in sprint and practitioner contexts to understand how behaviours change through sprints and how individual motivation changes through time, respectively.

The number of assigned tasks per sprint and practitioner will reveal if *more tasks are being assigned to a user*, thus leading to *a drop in the rate of unassigned tasks*.

The goal of *reducing the number of uncompleted tasks per sprint* is wholly fulfilled when *all tasks are resolved by the end of each sprint*. This behaviour can be supported by some metrics: number of tasks per sprint and resolved tasks, which are control metrics to calculate the number of completed sprints (where all tasks are resolved before the sprint ends) and sprint velocity (sum of resolved tasks' estimates), and the number of reopened and persistent tasks (created and resolved in different sprints), which translate tasks that are reworked or pulled through sprints.

Table 3 Mapping between this proposal's objectives, behaviours and metrics. Metrics are classified as being defined in a sprint (S), assignee (A), or sprint and assignee (S + A) contexts

Objectives	Behaviours	Metrics
Improve tasks specification quality, which can in turn increase the sharing of knowledge among practitioners and provide documentation to support communication	Elaborate on the task description, providing useful information that clarifies its purpose	–
Reduce percentage of unassigned tasks	Assign more tasks to a user	Number of assigned tasks (S) Number of unassigned tasks (S) Estimated effort (in days) (A)
Increase number of tasks completed per sprint	Resolve all tasks by the end of each sprint	Number of tasks (S + A) Number of resolved tasks (S + A) Number of unresolved tasks (S + A) Number of completed sprints (S) Sprint velocity (S) Number of reopened tasks (S) Number of persistent tasks (S)
Increase the number of effort estimated tasks	Estimate the effort necessary to complete a task, in all tasks	Number of estimated tasks (S + A) Number of not estimated tasks (S + A)
Increase participation in meetings	Check-in to Scrum meetings	Rate of attended Scrum meetings (S + A)
Increase team cooperation with team-centric goals and rewards	–	–
Implement project tracking with continuous feedback based on relevant metrics	–	–

All tasks should be linked with the effort estimated to be necessary to complete them, which is measured through the number of estimated and not estimated tasks. Calculating the effort allocated to a practitioner (given by the sum of all his/her estimations), in days, will provide an overview of team's work allocation.

The rate of attended Scrum meetings will translate if *practitioners' participation in meetings* is increasing. Confirmation and degree of achievement of *increase in tasks specification's quality*, *team's cooperation*, and *project tracking implementation* will be given by qualitative means of evaluation, like interviews. A complete mapping between objectives, behaviours and metrics is presented in Table 3.

5.3 Describe Your Players

The target players for this gamification solution are practitioners, who are a distinct group of workers that usually are not motivated by the same things as population in general. Some factors that likely motivate these workers, which are often mentioned and do not require much management or organizational involvement, include: communication between practitioners, management and customer; contributing to the overall success of the project; having feedback regarding team and individual performance, based on data collected and relevant metrics; and receive rewards and incentives (Mahaney and Lederer 2006; Beecham et al. 2008; França et al. 2011; DeMarco and Lister 2013; Meyer et al. 2014; Verner et al. 2014).

5.4 Devise Activity Cycles

Cycles that will engage players, based on their actions and solution's feedback, are described here. As players perform their daily tasks, they receive positive and constructive feedback, which can be a recognition that (s)he has done something right, or an alert that something needs to be improved. Feedback should guide users towards desired behaviours and motivate them to take further action.

To provide this motivation boost, we applied the Nudge theory (Thaler and Sunstein 2008). A nudge can be defined as a gentle and subtle push to influence individuals' behaviour in a direction that is better for themselves, as judged by them. Such initiatives must be optional, and easy and cheap to avoid. The authors defend that choice architects (i.e. the people responsible for organizing the context in which people make decisions) should follow a libertarian paternalist approach: move people towards directions that will make their lives better, while allowing those who want to follow other paths.

The authors provide examples of several nudging techniques, like:

- **Choosing defaults:** that means, select the option an individual automatically gets if (s)he does not choose actively;
- **Give appropriate incentives**;
- **Give feedback:** as people make better decisions if they are well-informed, providing clear and immediate feedback allows people to learn and choose better;
- **Peer pressure:** achieved by informing individuals what other people are doing;
- **Structure complex choices:** split in smaller steps.

Based on the defined objectives and considering the factors that motivate practitioners, we defined the following nudges for this proposal:

- Provide immediate feedback, in the form of pop-up messages, emerging after specific behaviours or events;
- Display reminders, in the form of pop-up messages, when a relevant event is close to occur;
- Display reminders, in the form of pop-up messages, when the player is close to reach an achievement or reward;
- Give rewards to a player or to each member of a team for performing specific behaviours;
- Provide a dashboard displaying the project(s) information.

5.5 Don't Forget the Fun

In this step we define the fun elements to be included so that players are likely to engage with the solution. As practitioners like to communicate, a social component to visualize each other's accomplishments can promote fun, but they should be able to select the information they want to display. Scrum is oriented to cooperation, thus implementing team achievements can boost cooperation amid practitioners and might help developing a sense of belonging to something greater. Competition can also provide a fun factor, but should be promoted between teams, and not between practitioners. Practitioners like to receive feedback, so a progress bar displaying the user's current experience points and the ones needed to pass to the next level can provide feedback on the player's progression in the game. Given that practitioners like to receive rewards and incentives, these elements must be used. Fun can also be increased by associating the experience with engaging graphics and messages. To achieve this, rewards will be illustrated with appealing images (as badges or gems), and fun messages will be displayed.

5.6 Deploy the Appropriate Tools

The tools used to build the gamification solution, i.e. game elements and software, were defined in this step. Not all game elements are tangible like points, but they all important to understand the game (such as feedback or progress) (Werbach and Hunter 2012). This step's outputs are explained in the next section.

6 Design and Development

The first step was to decide which game elements (displayed in italics) to include to promote defined behaviours. A *score system* awards users with *eXperience points* for certain actions (e.g. resolving an issue). Further in the paper, we will just use "points", as the mostly commonly used acronym (XP) also designates the agile method extreme programming. *Points* allow users to progress through levels, and these two elements together are intended to promote a healthy *competition* between users. Greater *achievements* (e.g. resolving all issues in a sprint) are awarded with both *points* and *rewards* of two types: *badges* and *gems*. *Gems* are a virtual currency that can be traded for real *prizes* in a *marketplace*. *Achievements* can be either individual or collective to promote *cooperation*. For every action, users get positive *feedback* through pop-up notifications. Those will inform users if they are performing the target behaviours or guide them otherwise. Feedback can also be found in "Create Issue" forms, where users receive tips on how to improve an issue's specification. These features offer

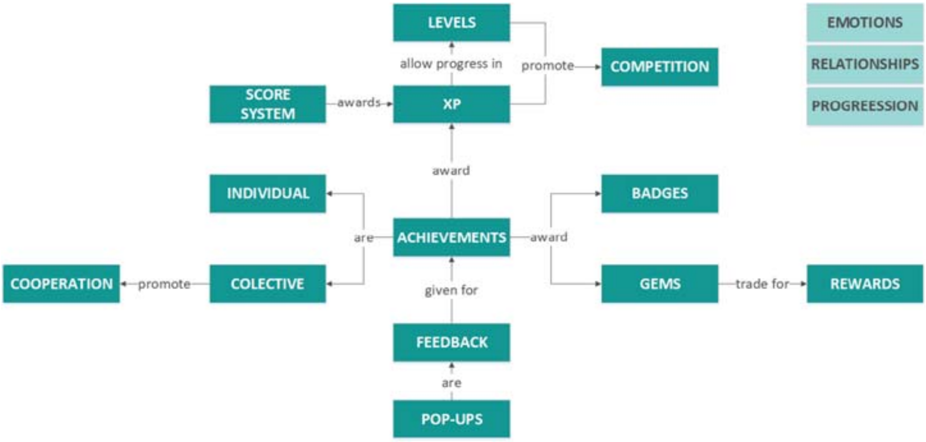
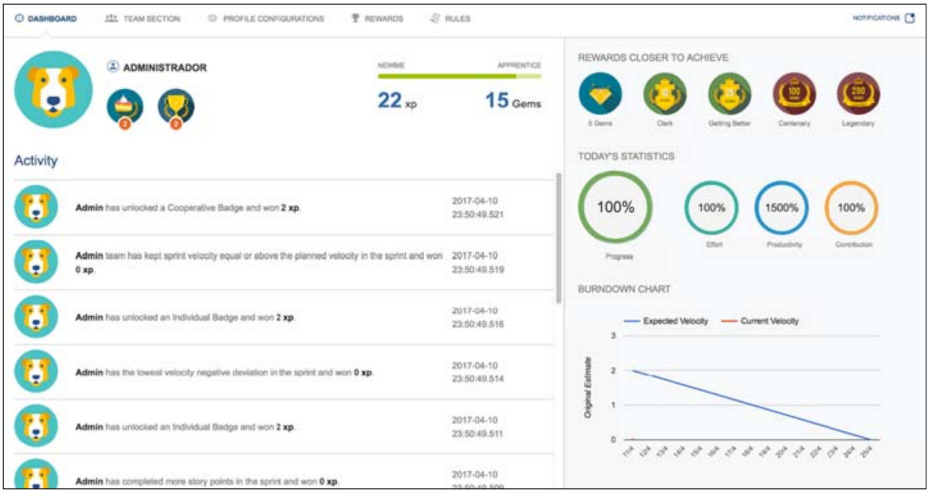


Fig. 1 Game elements used in this proposal



Admin has unlocked a Cooperative Badge and won **2 xp**. 2017-04-10 23:50:49.521

Admin team has kept sprint velocity equal or above the planned velocity in the sprint and won **0 xp**. 2017-04-10 23:50:49.519

Admin has unlocked an Individual Badge and won **2 xp**. 2017-04-10 23:50:49.516

Admin has the lowest velocity negative deviation in the sprint and won **0 xp**. 2017-04-10 23:50:49.514

Admin has unlocked an Individual Badge and won **2 xp**. 2017-04-10 23:50:49.511

Admin has completed more story points in the sprint and won **0 xp**. 2017-04-10

Fig. 2 Project Dashboard, with the profile and activity feed on left, and the rewards close to achieve, statistics, and burndown chart on right

users a wide range of *emotions*, allowing them to establish *relationships* and understand their *progression* towards a better adoption of Scrum techniques.

Next, these game elements were implemented as features in the app. We decided to leave the marketplace aside in a first version to avoid this extra implementation effort and quickly test the proposal and receive feedback to improve it. In Fig. 1 we can see a conceptual map linking all the game elements used in the first version of the proposal.

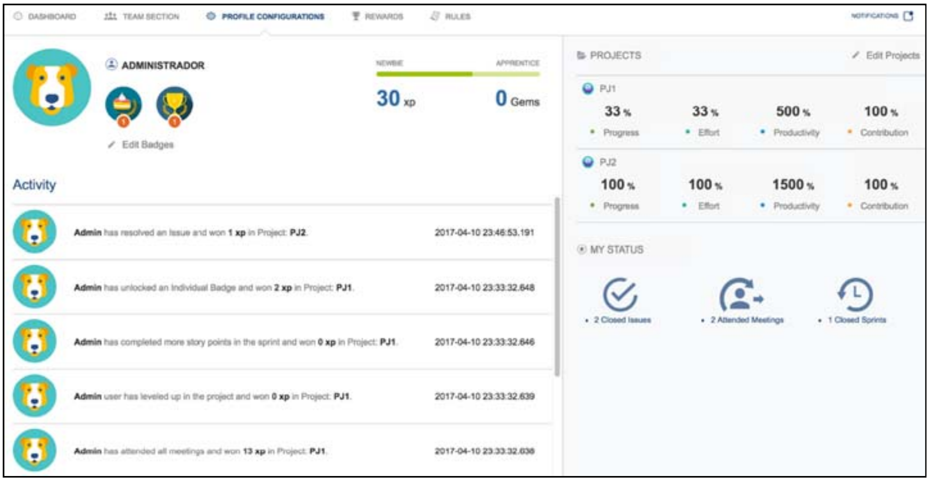
The Project Dashboard (represented in Fig. 2), displays information and statistics for a specific project, both general and concerning the user.

A small profile provides user information: a profile picture, name, project role, four featured badges, level progress bar, and points and gems earned. An activity feed lists all project's events. The user can consult the rewards (s)he is closer to win (e.g. with nine issues resolved (s)he is closer to receive a "Clerk" badge, given when 10 issues are resolved). Four project's statistics are displayed:

- **Sprint Progress:** reason between resolved and opened issues;
- **Effort:** reason between the effort already completed, and the effort assigned to the user in a sprint;
- **Productivity:** reason between current and estimated velocity of a user in a sprint;
- **Contribution:** number of issues resolved by the player and the total number of issues resolved in a sprint.

While "Sprint Progress" provides information about the current general state of the sprint, the other statistics provide information specific to the player. The awareness of the player contribution is important to understand his/her importance in the team.

Profile Configurations is a similar page that focuses on multiple projects (Fig. 3). Here, the player can select at most four of their badges to feature on their profile, and at most four projects (s)he is assigned to, whose statistics are displayed. Other three individual global metrics translate the player's work so far. The activity feed displays events from all user's projects.



	Admin has resolved an issue and won 1 xp in Project: PJ2.	2017-04-10 23:46:53.191
	Admin has unlocked an Individual Badge and won 2 xp in Project: PJ1.	2017-04-10 23:33:32.648
	Admin has completed more story points in the sprint and won 0 xp in Project: PJ1.	2017-04-10 23:33:32.646
	Admin user has leveled up in the project and won 0 xp in Project: PJ1.	2017-04-10 23:33:32.639
	Admin has attended all meetings and won 13 xp in Project: PJ1.	2017-04-10 23:33:32.636

	Progress	Effort	Productivity	Contribution
PJ1	33 %	33 %	500 %	100 %
PJ2	100 %	100 %	1500 %	100 %

- 2 Closed Issues
- 2 Attended Meetings
- 1 Closed Sprints

Fig. 3 Profile Configurations page, with the configuration panel, global activity feed, selected projects, and player's statistics

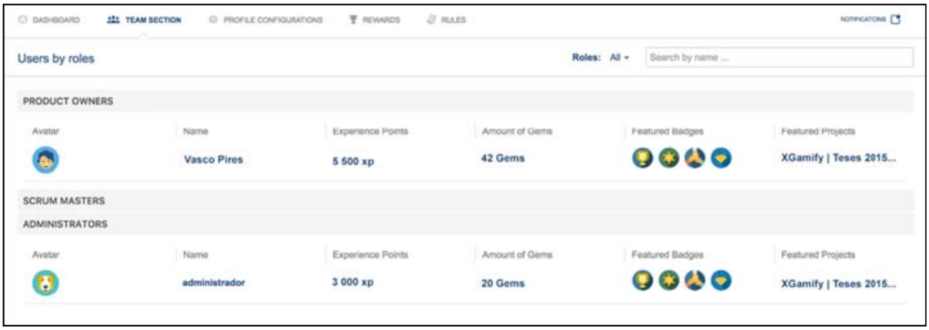
As different teams are assembled per project, the Team page (displayed in Fig. 4) enables a user to meet the people (s)he works with by seeing their profile, aiming at promoting communication between users. There is a collapsible list per role defined in the project and a search box for custom queries.

In Rules page, all rules related to app features are displayed in a collapsible list, divided by type.

All rewards available in the instantiation, both badges and gems, are organized by categories in a separate page, shown in Fig. 5. A reward is greyed unless it is awarded to the player. For badges, a number is placed on the bottom indicating the number of projects for which the player has received that badge. A reward's description is displayed while hovering it with the mouse.

Players receive positive and immediate feedback either through pop-up notifications or in "Create Issue" forms, where they receive tips on how to improve an issue specification (Fig. 6).

As Jira Software does not support Scrum meetings, this functionality was implemented by creating one issue type for each Scrum meeting: Sprint Planning, Daily Meeting, Sprint Review, and Sprint Retrospective, and a new custom field to specify the meeting date. This



PRODUCT OWNERS

Avatar	Name	Experience Points	Amount of Gems	Featured Badges	Featured Projects
	Vasco Pires	5 500 xp	42 Gems		XGamify | Teses 2015...

SCRUM MASTERS

ADMINISTRATORS

Avatar	Name	Experience Points	Amount of Gems	Featured Badges	Featured Projects
	administrador	3 000 xp	20 Gems		XGamify | Teses 2015...

Fig. 4 Team page, displaying the list of players and a search box

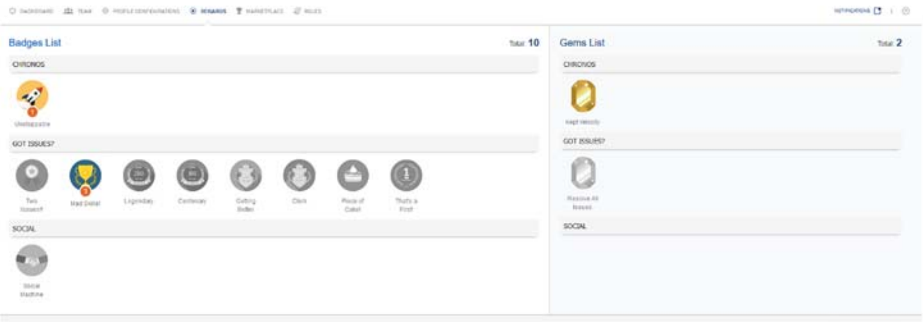


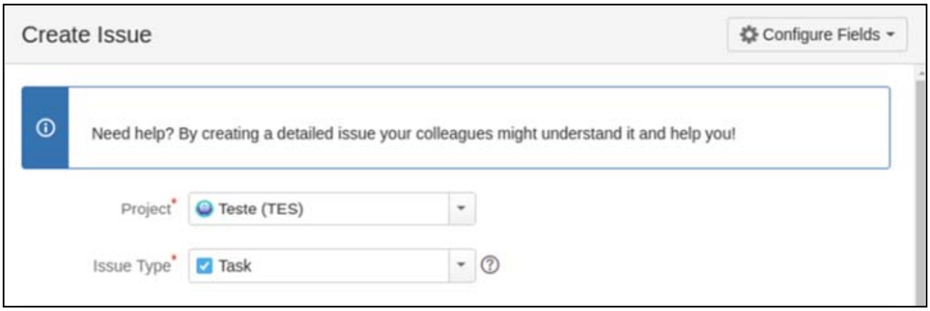
Fig. 5 Rewards page, with badges on the left and gems on the right

way, the Scrum Master can create meetings through these issues, and other team members can access an issue to check-in to a meeting, thus confirming their presence. In Fig. 7 we display an example of an issue of type “Sprint Retrospective Meeting”.

7 Evaluation

Results of the different demonstration activities are presented and analysed in the following subsections, where practitioners’ IDs are displayed in italics. We start with the evaluation of the metrics gathered during the baseline and field studies (described in Table 3), both independently and in comparison against each other. Then, we analyse the insights from the interviews with team members. In section 8, we compare these results against the goals of the solution (discussed in section 2.1) to see if and to what extent the gamification solution can be used to address the research problem.

When applying the Shapiro-Wilk normality test, we had to reject the null hypothesis that samples follow a normal distribution for five metrics in the baseline study ($p < 0.05$). For the field study, even though we cannot reject the null hypothesis for all metrics, data samples are too small to reach a decision of normality. The statistics calculated are presented in Table 4 (sprint context) and in Table 5 (assignee context).

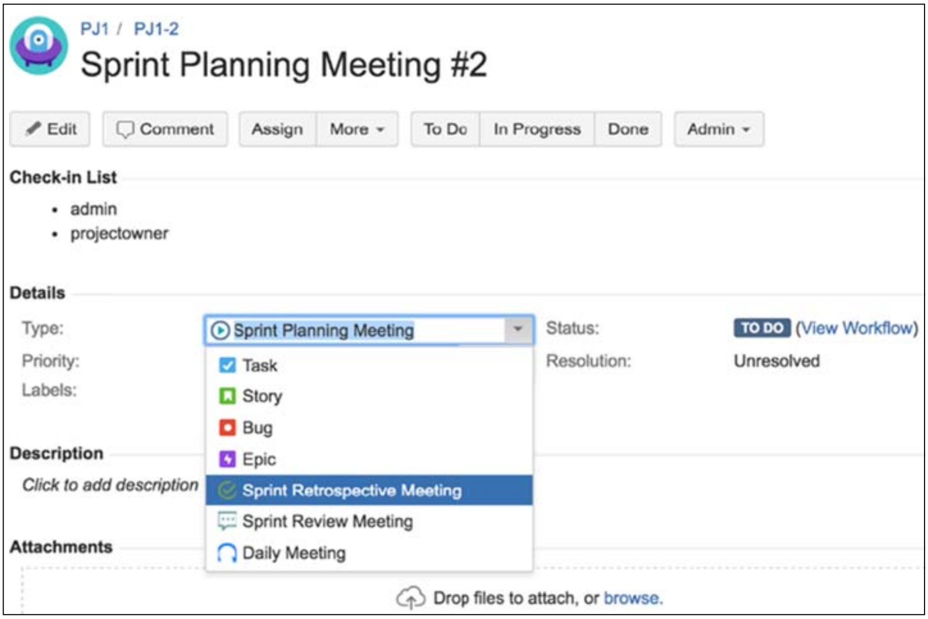


Need help? By creating a detailed issue your colleagues might understand it and help you!

Project* Teste (TES)

Issue Type* ☑ Task

Fig. 6 Form for creating an issue, with example of a tip



Sprint Planning Meeting #2

Check-in List

- admin
- projectowner

Details

Type: Sprint Planning Meeting

Priority:

Labels:

Status: TO DO (View Workflow)

Resolution: Unresolved

Description

Click to add description

Attachments

Fig. 7 Issue of type “Sprint Planning Meeting”. Other issue types are visible in the dropdown list

7.1 Baseline Study

The nine most recent ProjectX’s versions were selected as the baseline sample, as they translate Scrum practices’ current status (which can evolve through time). These versions are composed by 27 sprints, comprising a total of 306 issues over a period of one year and four months.

During data cleaning, we found that 14 issues were resolved several times without being reopened, a phenomenon not allowed in Jira Software’s workflow defined for this project. ProjectX members could not find a justification for this to have happened. Thus, these issues were considered outliers and discarded. Additionally, some issues were not estimated, a fact that impacts sprint velocity metric. However, we decided to keep those issues, and further in this document we discuss the implications of this choice.

At last, 292 issues spread by 27 sprints were analysed, including 29 stories (9.9%), 74 tasks (25.3%), 152 bugs (52.1%), 34 improvements (11.6%), and 3 new features (1%). Overall, 82 issues were persistent. Mean sprint duration was 13.2 days (st-dev = 5.3). Three sprints were complete (11.8%), and 24 contained reopened issues (88.9%). On average, each sprint had

Table 4 Statistics for sprint context’s metrics in baseline and field studies

	Baseline		Field study		Mann-Whitney U test	
Sprint Context Metrics	mean	std	mean	std	U	p value
Issue Total	16.5	10.9	13.8	8.1	58.5	0.813
Estimated Issues (%)	59.0	33.6	89.4	9.9	51.5	0.906
Resolved Issues (%)	63.0	29.1	63.1	37.7	25.5	0.098
Reopened Issues (%)	23.5	15.0	39.0	38.6	37.0	0.331
Velocity (Days)	32	38.9	48.2	30.3	40.0	0.425

Table 5 Statistics for assignee context's metrics in baseline and field studies

	Baseline		Field study		Mann-Whitney U test	
Assignee Context Metrics	mean	std	mean	std	U	p value
Issue Total	41.7	42.8	4.4	2.0	44.5	0.012
Estimated Issues (%)	54.4	22.5	82.1	21.8	9.5	0.061
Persistent Issues (%)	51.6	31.6	66.2	41.4	36.0	0.160
Effort Assigned (Days)	53.4	68.6	12.7	6.1	11.5	0.108

16.5 issues, where 63% were resolved, 59% estimated, and 23.5% reopened. We can see in Fig. 8a) that half of the sprints had at most 66.7% of their issues estimated, and in Fig. 8b) that 10 sprints (37%) had less than half of the issues estimated. Mean velocity was 32 days. Four sprints were completed without persistent or reopened issues.

The practitioner with more issues assigned to him/her was *sdrg*, with 110 issues (37.7%). Other practitioners had fewer than 100 issues each assigned to them. Also, *sdrg* was the one with the most allocated effort (200 days) and persistent issues. The only assignee with all issues estimated was *mfda*, the second with less effort allocated (10 days). Averagely, each assignee had 43.6 issues assigned, where 40.3% were estimated, had 20.4 persistent issues and was allocated with 62.4 days of effort. As shown in Fig. 9, half of the assignees had at least half of their issues estimated, but this value only represents 23 issues (7.5%) and 47.2 days of estimated effort (10.8%) of the whole sample.

These results show that the team complies with Scrum practices by assigning all issues. However, not all issues were estimated, thus influencing velocity, which is based on effort allocation.

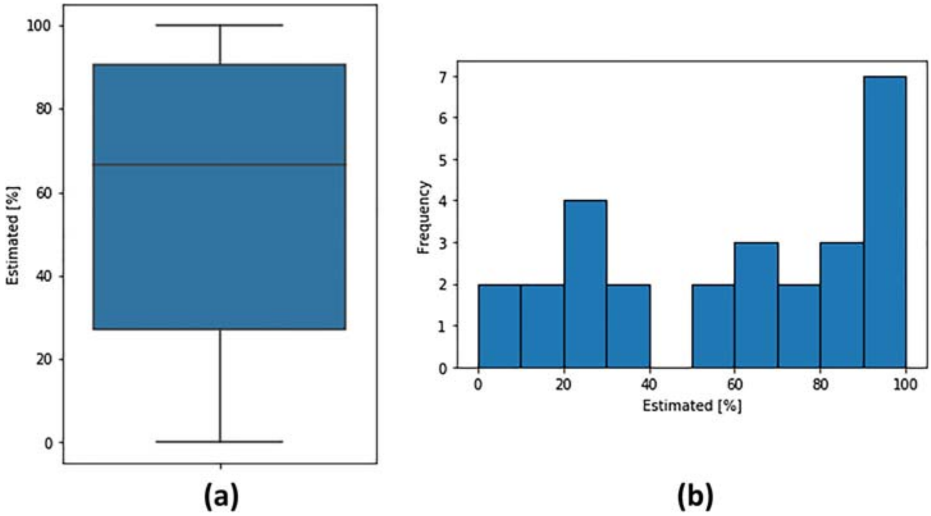


Fig. 8 Issue estimation analysis, where "Estimated [%]" corresponds to the percentage of estimated issues. **a** Boxplot of estimation ratio for all sprints. **b** Histogram representing the sprint frequency for each bin with 10% interval

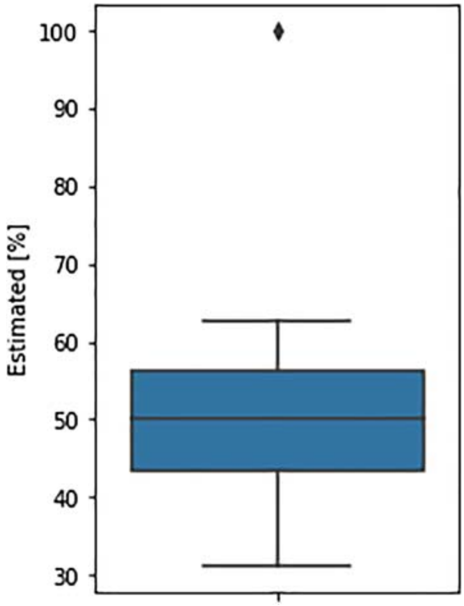


Fig. 9 Boxplot of estimation ratio for all assignees. "Estimated [%]" corresponds to the percentage of estimated issues

Since different issue types can lead to distinct problems, we studied a possible correlation between issue type, existence of estimation, and velocity. As *New Feature* issues are in reduced number, they were discarded from this analysis. Figure 10 presents the heat map with the Pearson correlation coefficient for each case. Issues of type *Bug* and *Task* are moderate and

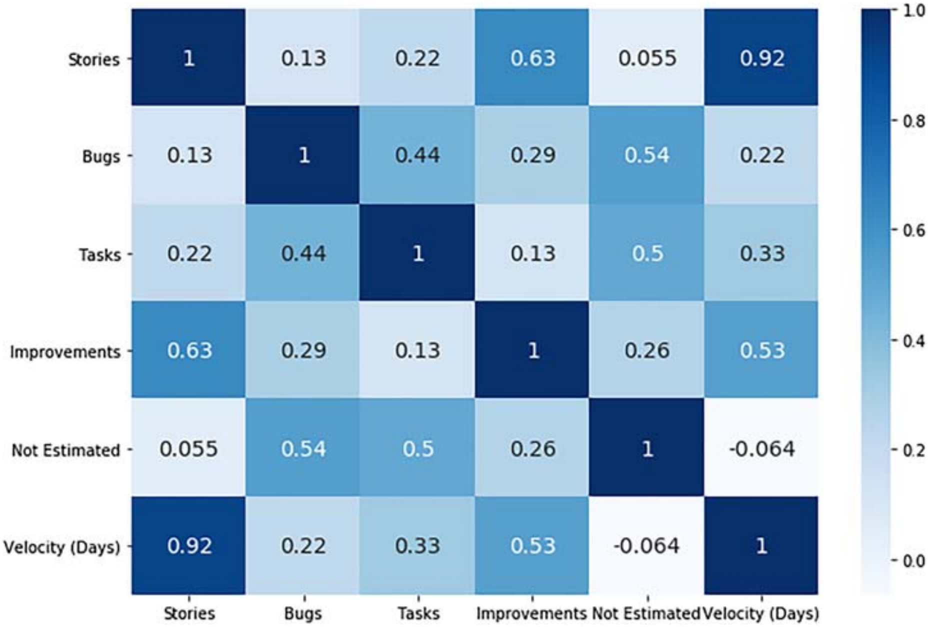


Fig. 10 Pearson correlation coefficients between issue types, not estimated issues and velocity

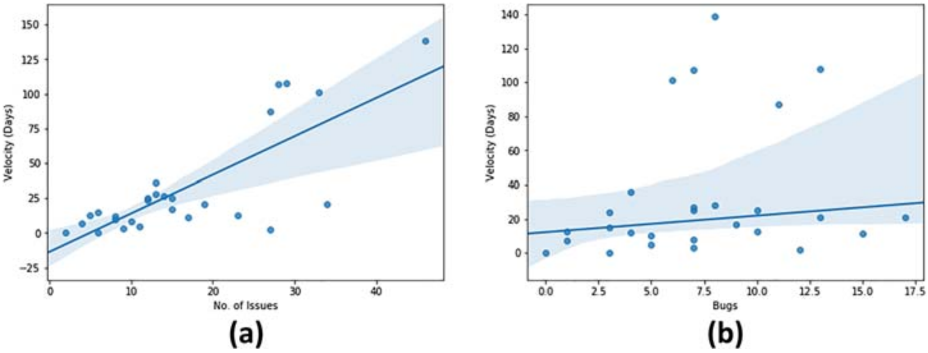
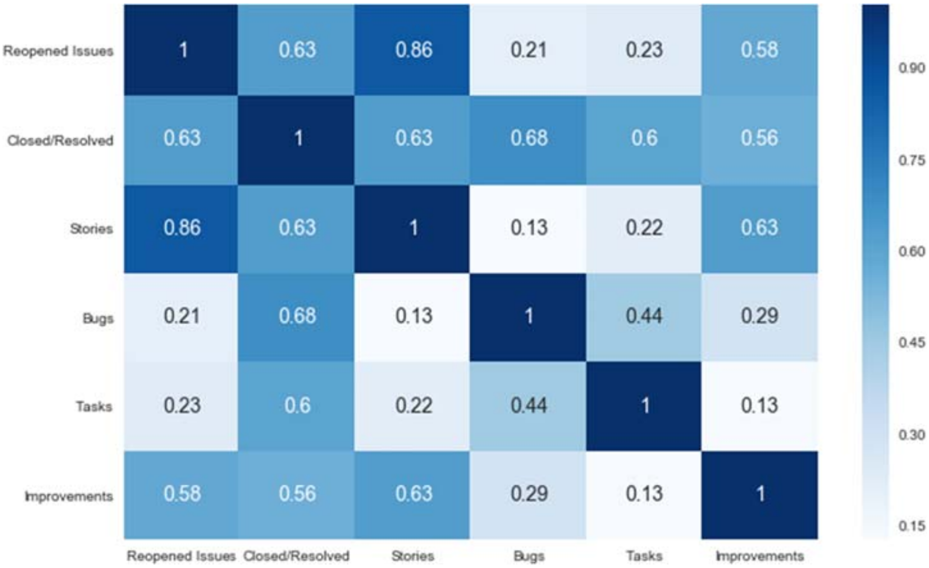


Fig. 11 Scatterplots comparing velocity with (**a**) the number of issues and (**b**) number of issues of type Bug

positively correlated with lack of estimation (0.54 and 0.5, respectively), thus an increase in these issues can lead to a low ratio of estimated issues. In other words, *Bug* and *Task* issues are often not estimated. Oppositely, issues of type *Story* are strong and positively correlated with estimation (0.85) and with velocity (0.92), meaning that *Story* issues are often estimated and might increase the overall velocity.

As most issues are of type *Bug*, which are not often estimated, we further studied these issues' impact on velocity. Scatterplots of Fig. 11 show the relation between velocity and a) the total number of issues and b) number of *Bug* issues.

A comparison of both graphs suggests that velocity increases with both the number of total and *Bug* issues, but growth is slower for *Bug* issues. To further understand these results, we calculated r-squared on both regressions. Around 60% of velocity variation can be explained by the total number of issues, while only 25% is explained by the number of *Bug* issues. This suggests that the number of *Bug* issues has a significant influence on sprint's velocity, thus this slower velocity growth might be explained by the fact that *Bug* issues are often not estimated.



	Reopened Issues	Closed/Resolved	Stories	Bugs	Tasks	Improvements
Reopened Issues	1	0.63	0.86	0.21	0.23	0.58
Closed/Resolved	0.63	1	0.63	0.68	0.6	0.56
Stories	0.86	0.63	1	0.13	0.22	0.63
Bugs	0.21	0.68	0.13	1	0.44	0.29
Tasks	0.23	0.6	0.22	0.44	1	0.13
Improvements	0.58	0.56	0.63	0.29	0.13	1

Fig. 12 Pearson correlation coefficients between issue types, reopened and resolved issues

Most sprints have reopened or persistent issues, showing that not all issues are resolved before the sprint. We analysed how each issue type was likely to be reopened and resolved. In Fig. 12 we can see that Bug issues are weakly correlated with reopened issues, but *Story* issues are usually reopened. All issue types are strongly correlated to resolved issues. Thus, results suggest that *Bug* and *Story* issues are usually resolved, but the former do not tend to reappear, while the latter does.

In the assignee context, a very strong correlation (0.85) was found between allocated effort and number of persistent issues, suggesting that assignees have more persistent issues when they have more effort allocated. We hypothesise that they comply less with the plan when they have more work to do.

Given that insufficient documentation has been identified as a problem affecting Scrum practices' implementation (see section 3.2), we further analysed if providing a description for an issue (i.e., more details and documentation) could influence the number of reopened issues. Because this is a binary metric (1 if an issue has been reopened, 0 otherwise), a special case of the Pearson correlation, the Point Biserial Correlation, was used instead, ranging from −1 to 1. The resulting coefficient of 0.017 suggests that providing a description has no clear impact on the issue reopening.

Overall, these findings suggest that the team has problems in estimating tasks and planning sprints, a result supported in the literature, as described in section 3.2). The results of the questionnaire presented and discussed in section 4 also show that practitioners are not motivated to perform such activities.

No conclusions could be drawn regarding Scrum meetings and roles, as Jira Software does not record that information. Yet, the fact that one assignee has more effort assigned than the rest of the team suggests that (s)he has a special role in the project.

7.2 Field Study

A field study was conducted between middle October of 2017 and January of 2018 (about three and a half months). Two versions were analysed, comprising four sprints and 31 issues, from which 13 were improvements (41.9%), 11 were stories (35.5%), five were bugs (16.1%), and two were tasks (6.5%). Mean sprint duration was 21.8 days (st-dev = 6.7). Overall, 17 issues were persistent (54.8%), 24 estimated (77.4%), and all were resolved and assigned. One sprint was complete (25%) and three contained reopened issues (75%). On average, each sprint contained 13.8 issues and had 63.1% issues resolved, 89.4% estimated, and 39% reopened. Mean velocity was 48.2 days. One sprint was completed without persistent or reopened issues.

The assignee *pamp* was the one with more issues, with seven issues (22.6%), the third with more estimated issues (71.4%), and the only without persistent issues. The assignee with more effort allocated (20 days) and estimated issues (83.3%) was *nmps*. The three assignees who had all their issues estimated (*hjfr*, *tfsi*, and *fcri*) were the ones with fewer issues (four, three, and one, respectively), all persistent. On average, each assignee had 4.4 issues, 2.4 persistent issues, was allocated with 12.7 days of effort, and around 82.1% of his/her issues were estimated.

There were 32 players in the app, but only seven resolved issues. Player with highest score was *rmbr* (91 points), also the one who resolved more issues, achieving level 3. Following, there were *tfsi* (78 points) and *nmps* (55 points), who achieved level 3, and *fcri* (45 points); *hjfr* (34 points); and *pamp* (29 points), who achieved level 2. All other players had either 6 or 0 points. On average, each player scored 13.1 points (st-dev = 23.13). Globally, players unlocked 60 rewards of five types out of 11 (45.5%). Because only one project team was evaluated, global levels are not further analysed in this paper. With five and four rewards, *rmbr* and *tfsi* were the players with more rewards, respectively. Another 17 players received three rewards (the same two badges and one gem), and the others received none. No players turned off notifications. No Scrum roles or events were created.

Regarding gamification, we can see that the player with highest score and more rewards was the one that resolved more issues. The assignee with more issues estimated and allocated effort was the player with the third biggest score, even surpassed by one of the assignees with fewer issues. Apart from the six top players, the remaining scored very low and unlocked few rewards.

7.3 Differences between Studies

We looked at the differences between metrics in baseline and field studies. Nevertheless, the Mann Whitney's U tests applied on these differences revealed no statistical evidence that these metrics differ between studies (p value>0.05). Yet, the p value for the difference between mean percentages of estimated issues per assignee is 0.06, which is just above the 5% confidence level. Because the p value is very close to the defined threshold, there is some evidence that these means can indeed be different (suggesting that assignees are estimating more issues), even though for a strict p value of 0.05 this hypothesis should be rejected.

7.4 Interviews

Interviewees were asked to answer some questions using points in a Likert scale, which varied with the question. Results achieved in such questions can be found in Table 6 and in Fig. 13.

The characterization of the four ProjectX team members interviewed is presented in Table 7. The Scrum Master (who also performs tasks of Product Owner, which is an unofficial role) and three developers, responsible for developing and provide support to the product, were interviewed. The Scrum Master was the one working for the longest time at the company, starting six years ago on a different project, while developers started working there less than one year and a half ago in ProjectX. This was the first professional experience for two developers. All interviewees holded a degree in IT Engineering.

Table 6 Results of the Likert scale questions

Question ID	Short description	Median
10	Satisfaction with Jira Software	5.0
12	Compliance with Scrum practices	4.0
12.1	Compliance with adapted Scrum practices	5.0
20.1	Relevance of the information in the Dashboard page	3.5
20.3	Importance of project metrics' feedback in improving work	4.0
21.1	Relevance of the information in the Profile Configurations page	3.5
21.3	Importance of the individual statistics' feedback in Profile Configurations	3.5
23.1	Usefulness of the information presented in pop-ups.	4.0
23.2	Usefulness of the tips to improve issues' specification	4.0
25.1	Clarity of the score system	4.0
25.2	Balance of the score system	3.5
26	Importance of supporting Scrum roles and meetings in Jira Software	2.0

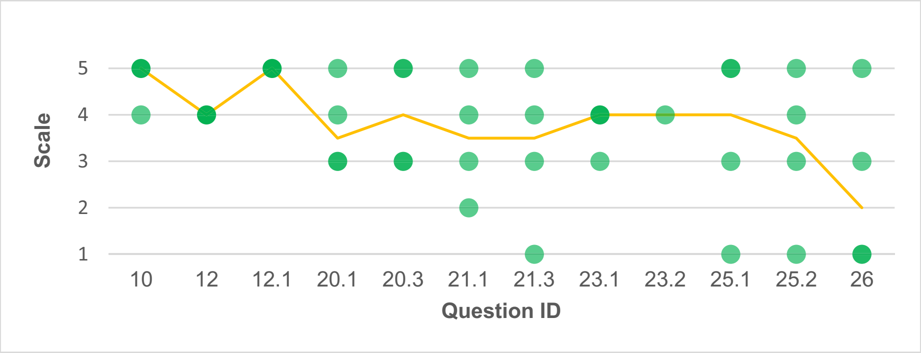


Fig. 13 Results of the Likert scale questions, where the points represent individual answers, and the line represents the median

Interviewees said they use Jira Software daily and are very satisfied in working with this tool, for which the developers received some training when they joined the company. None of them was Scrum certified but worked with the framework since they joined the company. All of them rated the team's compliance with Scrum with 4 (stating they comply with 80–85% of the practices) and with their adapted version of Scrum with 5.

Except for one developer, all interviewees liked games and played regularly (at least weekly). Overall, they liked strategy, action, RPG, and sports games, mostly seeking to solve challenges and unlock rewards. Two developers did not know what gamification is, and the Scrum Master was the only one with previous experience using a gamification app (a badge app integrated with the code repository). Concerning the proposal, interviewees said they first used the app frequently, but attention gradually dropped since it is not challenging enough. Currently, they only consulted the app when a sprint is closed, excepting for one developer that stated to use it whenever he closes an issue.

Although the score system improved after refinement, two of the four interviewees still found it hard to understand and thought it was still not fully aligned with the team's needs. They thought a clearer distinction should be drawn between developers and testers, which could be achieved if the administrator could configure the points earned on each issue status' transition. Regarding players with residual number of points and rewards, the Scrum Master said that these are clients of the company, who can also create issues.

Information displayed in the dashboard was considered relevant, but not essential. This includes the project metrics, whose feedback was deemed somehow important. Interviewees enjoyed seeing their points and the rewards closer to achieve, as well as the activity feed. Since the team resolved all issues right before closing the sprint, they do not use the burndown chart artefact, thus this element was neutral.

In Profile Configurations, interviewees liked the individual statistics, but not all found them important. They would like to have more statistics available, particularly regarding project-related events, like the number of commented issues. They suggested to show here the issues assigned to team members in all projects, with a direct link to the issues' page. Despite enjoying the possibility of selecting featured badges and projects, some interviewees forgot to select them, and thought there should be a default selection (e.g., by order of assignment).

Interviewees did not discern between Dashboard and Profile Configurations pages. This might be because information shown was very similar, as the study was conducted in only one project. As workers in this company are usually assigned to only one project at a time for long

Table 7 Interviewees' profiling

Questions		I1	I2	I3	I4
Position/Role		Product Manager Scrum Master Product Owner	Developer Support	Developer Support	Developer Support
Education and Training		MSc in IT Engineering	MSc in IT Engineering	MSc in IT Engineering	BSc in IT Engineering
Experience	Time at the company	6 years	15 months	8 months	10 months
	Time at the project	4 years	15 months	8 months	10 months
	Past experiences	Software developer for three years in another IT company	None	Internship as software developer during 9 months	Internship in system maintenance and network installation
Has Scrum Certification?		No	No	No	No

periods (for example, the Scrum Master has been in this project for four years), having both project and global contexts might not be very relevant for this company. Maybe due to this, there was no consensus regarding the relevance of the information displayed in this page.

The Team page was the most popular one, as interviews were very competitive, but they think it should be converted into a leaderboard with the possibility of ordering team members by earned points. They would also like to know how each player won his/her points, which can be known by displaying a list of the events that increased a player's score. Additionally, they would like to see more information per player, like statistics (especially the number of resolved issues), and the description of featured badges, which currently can only be consulted on Rewards page. The existence of roles without users assigned was a drawback that created empty sections in this page. These should be omitted, or the role should become a simple profile field.

This team liked to win and showcase badges, but because they are only awarded for those who resolve issues, interviewees believed that more and diverse badges should be available, including some of easy access, so that everyone can unlock these rewards. None of the interviewees understood what gems represent, and thought they were linked with the score system, and not with rewards.

While some interviewees did not like the many notifications launched on the login, they enjoyed this functionality for the useful feedback it provides on the work and the game. One interviewee suggested that pop-ups should be distinct from Jira Software ones, so that they could be easily differentiated. Only the Scrum Master noticed the tips for improving issues' specification, so the median of this functionality's usefulness rating corresponds to the only answer received. He found this functionality interesting but suggested that it should be linked to a more explanatory description.

Most interviewees believe that having Scrum roles in Jira Software is not important, unless if linked with players' score or profile. They did not use the functionality for registering meetings because they already had a well-defined process to register them, which they considered better. Nevertheless, they believed this method could be valuable to teams without some structured method, concerning the presences were recorded by the Scrum Master.

In general, interviewees liked the app, which they considered as an informative tool that brings some challenge and simultaneously promote a healthy competition and team spirit. However, they thought there was large room for improvements. They believed few was information available, and that the app should be improved with real awards and a marketplace to manage them. The functionalities that positively impacted their experience were the rewards and the Dashboard page (specially the project statistics). Inversely, the misfit score system was considered to have a negative impact on the team. Their experience using the app was positive (yet not outstanding), as it softened the process formality. Yet, they believed that at this point the app is not able to motivate practitioners on the long run, but after implementing the suggested enhancements it can be of value to Scrum teams.

In sum, these interviews revealed that this team is not cross-functional, as some practitioners are fully committed with testing activities, and that the Scrum Master and the Product Owner are the same person, against what is advocated by Scrum. These aspects highly impacted the score system, as the app currently awards players mostly for resolving issues, but in this team the person assigned to an issue rarely is the one resolving it. Interviewees liked the Team page, rewards, and feedback given in the Dashboard and notifications, but identified the lack of global leaderboards, the reduced number of rewards (and the type of behaviours that unlock them), the many pop-ups launched in the login, and the burden caused by individual check-in to Scrum meetings as aspects to improve. They suggested that score should be based on workflow transitions (and tailored by the administrator) and login pop-ups merged in one comprising the most important information. They think that the events awarding players with points and rewards should be more clearly presented on the app, and that the constant feedback can impact the team's work, given identified limitations are fixed.

8 Lessons Learned

In this section, we answer the research question (if gamification contributes to improving adoption of Scrum practices by practitioners) by analysing the impact of the gamification proposal in the team based on the results presented in the previous section.

Although we found no strong evidences that differences between studies are statistically significant, there seems to be a slight improvement in the metrics representing the behaviours and objectives in Table 3– in particular, assignees are estimating more of their issues, which are always assigned.

However, this team still faces problems in implementing the entire sprint backlog, as all issues are rarely resolved by the end of the sprint. Also, the team works on issues in versions without sprints to resolve critical bugs, and some practices are inconsistently implemented, such as having sprints with variable duration, despite being compliant with Scrum recommendations. On the other hand, we could not assess whether some of the problems were addressed after using the gamification solution. For example, the specification quality of the issues does not seem to have been influenced by provided tips, as practitioners were not aware of its presence in the app. As the mechanism to register meetings was not used, no conclusions can be taken regarding this goal.

We must note that this solution might not be addressing all challenges affecting this team. Some of the challenges identified in sections 3.2 and 4 were left aside while designing the gamification solution, as it would not be feasible to consider them all. For example, we did not implement gamification elements to mitigate the discomfort that some practitioners might experience while communicating with the team, as we found in the questionnaire (see section 4). Nevertheless, this team seems to suffer from challenges that were not even previously identified, like having the same person taking Scrum Master and Product Owner tasks, or not being cross-functional, as some practitioners are fully committed with testing activities. Therefore, there is a risk that we might not be influencing practitioners' motivation if we are not gamifying behaviours linked with the challenges this team is really facing.

Therefore, it is important to emphasize that not all challenges affecting Scrum implementation can be easily addressed with gamification. For instance, if senior management does not support or excessively interfere with gamification, the impact is likely to be limited. Likewise, promoting customer involvement might be hard if customers are not willing to use the gamification solution.

After validating the quantitative results with the Scrum team members during interviews, we realized that players enjoyed unlocking badges and seeing the information displayed in the Dashboard (such as the points) despite the limitations mentioned above. Moreover, they believed that the continuous feedback could impact the team's work, but right now too many notifications are being launched simultaneously. This aspect must be fixed in the future, because unclear messages might be considered as "noise" and consequently the feedback would lose its meaningfulness (Foucault et al. 2019).

An interesting finding was the identification of three clusters of players during the metrics and gamification's analysis: Scrum Master, testers, and developers. The Scrum Master is the player with higher score, more rewards, and more issues resolved. Testers come second regarding score and rewards, as they also resolve many issues. Developers rarely resolve issues and, because the app currently awards players for resolving issues, they received very few points and rewards. Additionally, clients are a special group of users: they are shown in the app but, because they are not part of the Scrum team, they do not participate in the gamification, thus are not considered a player's cluster. Despite only one cluster being Scrum specific, in the future we might consider creating clusters by Scrum roles.

Overall, and based on the interviews' outputs, the small improvements in Scrum metrics do not seem to be explained by an increase in practitioners' motivation. However, participants did enjoy the experience of using a gamification solution, as it promoted a healthy competition environment that softened the project formality.

We know that extrinsic motivators have a limited effect on engagement, that usually fades away if not combined with intrinsic motivators aligned with SDT principles (Nicholson 2015). This could explain our results, as the proposed gamification solution is mostly based on extrinsic motivators. However, and especially since this research was mainly exploratory, we should "not wait for the birth of intrinsic motivation" (Zichermann and Cunningham 2011) but instead use extrinsic motivators to start the process. More research is needed to understand which game elements could create the conditions for this intrinsic motivation to occur (Alsawaier 2018).

Additionally, we further need to understand how our gamification solution impacts motivation on the long term, as participants stated not to believe that this proposal could motivate them for a long time. Conducting longitudinal studies during larger periods of time can provide insights on the long-term effects of gamification, particularly on the individual impact of each game element on participants' motivation (Platonova and Bērziša 2017; Alsawaier 2018).

Still, we believe that every gamification study provides a contribution, even if smaller. The goal of this exploratory study was to test gamification's acceptance and impact on the team, without focusing too much on its long-term effect. Collecting feedback early in the process was crucial to understand the importance of each element and to detect improvements. For example, we understood that the gamification data recorded in the database did not support a deeper data analysis, such as the evolution of points during the sprints. Thereby, in the future we could focus on extending gamification effects in time, knowing which are the best elements.

Another important aspect to consider is the alignment between the characteristics of participants and the game design techniques employed. In this study, some gamification and design choices were not aligned with the team's characteristics:

- The burndown chart was of no use for this team, as issues are resolved right before closing (i.e. not during) the sprint;
- Because in this team practitioners are assigned to only one project at a time, there is no need to have both the Dashboard and Profile Configurations pages, as the information presented is similar;
- This team was very competitive, thus cooperative challenges and rewards did not seem to motivate these practitioners;
- Game elements were selected assuming the team was cross-functional, which is not true and had impact for example on the score system.

These issues might have impacted participants' motivation. For example, even the Scrum Master, who was privileged due to the score system bias, eventually became demotivated to use the app. In other contexts, such as a team that was just starting to use Scrum, the positive impact on practitioners' motivation would probably be small as well since other challenges (such as learning the Scrum basics) would have to be addressed first. All in all, understanding and designing for the context is essential (Foucault et al. 2019) but a balance must be found between creating gamification solutions that can indeed impact the team but that are not too specific in order to be validated with other teams.

Many of the remaining challenges are difficult to overcome. There is a need for more empirical and longitudinal studies with different real-world Scrum teams (Pedreira et al. 2015) but we have found difficult to find organizations and teams to carry out experiments. Still, and based on our results and those of similar works discussed in this paper, gamification can always be useful, at least for changing the atmosphere that lightens the formality of software development projects. By conducting more and better studies based on mixed methods design, which allows the collection of data and its validation with qualitative methods (Alsawaier 2018), researchers and practitioners can create better gamification solutions that can greatly impact practitioners' motivation and promote behaviour change.

9 Threats to Validity

This section discusses the threats to the validity of this study's results, categorized into the four types proposed by Wohlin et al.: internal, external, construct, and conclusion (Wohlin et al. 2012).

Internal validity assesses the causal relationship between treatment and outcome. All instruments used in this study, like the gamification solution and the interview guide, were validated by all authors to prevent problems in the study's design. Occasionally, other experts (e.g. designers) were consulted to validate some decisions. The critical bug detected in the proposal might have affected the participants' behaviour during the field study, although no evidence was found during the interviews to support this. The team's satisfaction in following Scrum practices and its relationship with one of the authors might have affected their behaviour, thus this group might not be representative.

External validity translates to what extent the study's results can be generalized to other settings and are of interest to people outside the study. As most of the questionnaire responses are Portuguese developers, results cannot be generalized to all practitioners. Similarly, only one team participated in the study, which is very homogeneous and was sampled by convenience (i.e., the nearest and most convenient people were selected). Because this group might not represent all practitioners, any conclusion drawn from this study cannot be generalizable outside the company, or even the project. Therefore, this study should be replicated with other teams in different companies. All the study's settings were reported, so that other researchers can frame and evaluate the results in this specific context. Despite these issues, this study represents an important contribution as it was conducted in a real setting with real people.

Construct validity shows how the study settings reflect the properties we really intend to capture. To improve the construct validity of the solution, we built a strong theoretical base using many sources. However, the outputs of the analysis of related work present some limitations that can have threatened the validity of the results. Most papers available are not so recent, and may not translate the current situation, and despite considering Scrum are more focused on agile in general, thus identified challenges might not be Scrum-specific. Finally, challenges were analysed as being independent and equally relevant, but they can be interrelated and their impact on the project can be different. Nevertheless, results showed that the solution might have been build considering the wrong challenges, which might threat construct validity. We also tried to avoid some bias by using two methods to test and evaluate the proposal, which allowed us not only to identify bugs but also to further understand results like the lack of cross-functionality in the team. Most team members are junior developers with a similar background in gaming. Results might variate with changes in seniority or gaming habits. The extent to which the team applies Scrum practices can influence results as well. Because they are being monitored, participants are subject to Hawthorne Effect, i.e. might try to improve their results to bias the outcome (Buchanan and Huczynski 2016). Moreover, the fact that their company was involved in the research project might have influenced their personal opinions during the study. All researchers validated each step of the research to reduce researcher bias.

Conclusion validity relates to factors that can affect the ability to draw conclusions about relations between treatment and outcome. A research protocol was defined to ensure results' reliability, clarifying how each step was run. The inconsistency found during questionnaire analysis might have threatened the validity of these results. While objective metrics were defined to evaluate the solution, some features needed to be measured more flexibly, thus we accepted this threat. Similarities between practitioners limited the heterogeneity of the studied team, but the data samples were too small. No statistically significant differences were found between studies but given the low statistical power of this tests there is a high risk that conclusions drawn can be wrong. This can be explained by the reduced size of the field study's sample. Data needed to be collected during several months to increase statistical power, but a gamification solution should be evaluated more often. The issues discarded from the analysis could have influenced results like the effort allocated to a practitioner. Because their specificities do not align with most metrics, this threat was embraced. According to the Scrum Master, no influence other than the growth of the team occurred during the studies. Yet, this and several other factors can influence projects' success, and would be crucial to evaluate their impact on the results.

10 Conclusion

Addressing practitioners' lack of motivation to adopt agile practices remains a challenge. Many authors have proposed gamification solutions, but research still lacks empirical validation. We developed a gamification solution as a Jira Software app to increase practitioners' motivation in adopting Scrum practices, which was demonstrated with a real-world Scrum team from the company we cooperated with for creating the gamification solution.

A comparison of data from a baseline and a field studies, extracted based on metrics defined in the proposal, suggest that results slightly improved after using the proposal. Gamification results revealed three different players' clusters: Scrum Master, testers, and developers. Except for the Scrum Master cluster, these clusters are not Scrum specific, but in the future three clusters could be created based on Scrum roles, for which specific game elements should be defined. Some improvement opportunities have been identified in this study during the interviews. After implemented, these enhancements can increase the potential of the proposal.

Based on these results, we learned some important lessons regarding the implementation of gamification initiatives, which are supported by most of the works previously analysed in section 3.3:

- Gamification can be useful in creating a positive atmosphere within Scrum teams, even if the impact on practitioners' motivation is reduced.
- A gamification solution must promote behaviours that address the real problems, otherwise practitioners' motivation can be hampered if their needs are not addressed. These problems should be studied not only by analysing prior works, but also by conducting empirical studies with the target team(s).
- A balance should be achieved between building a gamification solution that is adapted to the context where it is being applied but can also be validated with more teams.
- A gamification initiative should be run in a more agile way, where researchers should not only be in contact with management, but also with the participants. This would ensure that issues (such as bugs) are detected early, which can improve the success of the initiative.
- Gamification's potential can be harvested if more and better studies are conducted based on mixed methods, even if the conditions are not the ideal. Solutions that are built on extrinsic motivators and run during a smaller period can provide learnings and findings that can add to the research area. Plus, starting to demonstrate a solution earlier in time allows researchers to find problems in the proposal and the research method.

This exploratory case study represents the first iteration of our larger research effort, but in the future, we intend to replicate this work with more Scrum teams. Currently, we are demonstrating the improved proposal with two different teams from a different company.

These results can be of interest not only to researchers in this field, but also to organizations who are looking forward to increasing their workers' motivation in applying Scrum techniques. After being improved to be more customizable, the tool can be used by any software development organization to motivate practitioners.

Acknowledgements This work was partially funded by IAPMEI through project XGamify/P2020/SII&DT/11497.

Appendix 1

Questionnaire

We are a research team interested in improving the adoption of Scrum practices software development practitioners, and for that reason we are contacting you and would appreciate if you could help us by filling out a questionnaire.

The main goal is to understand the level of Scrum practices' implementation in your team, as well as your opinion regarding your team's motivation to adopt such practices.

Your participation in this study is voluntary. By completing and submitting it, you are voluntarily agreeing to participate. You are free to decline to answer any particular question and to abandon the questionnaire at any moment.

This questionnaire is anonymous, and data collected will only be used under the scope of the research project.

Demographic

1) Select your age range:

□ 17 or younger
□ 18 – 30
□ 31 – 39
□ 40 – 49
□ 50 – 59
□ 60 or older

2) Select your gender:

□ Male
□ Female

3) Select the country you are originally from.

Extensive list of options provided in Survey Monkey.

4) Select the highest degree of school you have completed.

□ High School and Lower
□ Bachelor
□ MSc
□ PhD
□ Post-Graduate Degree

5) Select the option that best matches the sector your organization operates in.

Extensive list of options provided in Survey Monkey.

6) Select the option that best matches the time you have been working for the company.

□ < 1 year
□ 1 - 5 years
□ 6 - 10 years
□ 11 - 20 years
□ > 20 years

7) Select the option that best matches your job's main function.

□ Individual Contributor
□ Team Lead
□ Manager
□ Senior Manager
□ Regional Manager
□ Vice President
□ Management / C-Level
□ Partner
□ Owner
□ Volunteer
□ Intern
□ Other

Scrum

1) Do your team use Scrum in their daily work activities?

If you don't selected “Yes” as an answer, please skip the remaining questions.

- □ Yes
- □ No
- □ No, but we plan to use it in the future
- □ Don’t know/ Don’t want to answer

2) Which is your main role in your Scrum team?

If you choose the option “Scrum Master”, please skip the following question.

- □ Product Owner
- □ Scrum Master
- □ Member of the development team
- □ Don’t know / Don’t want to answer

3) How supportive is your team’s Scrum master?

- □ Not at all
- □ Fairly supportive
- □ Very supportive
- □ Don’t know / Don’t want to answer

4) How much time per day do your team spend performing Scrum activities?

E.g., attending meetings, organizing the sprint board, dividing the project in small stories, etc.

- □ < 5 minutes
- □ 5 - 30 minutes
- □ 30 minutes – 1 hour
- □ 1 – 2 hours
- □ > 2 hours

5) How many daily meetings has your team performed during the last week?

- □ None
- □ 1 – 2
- □ 3 – 4
- □ 5 (All of them)

6) Classify the following Scrum events according to the level of importance your team believe they have.

1 – Waste of time, 2 – Not important, 3 – Indifferent, 4 – Important, 5 – Crucial

- □ Daily meeting
- □ Sprint planning
- □ Sprint retrospective
- □ Sprint review

7) Classify the following Scrum events and events according to the level of motivation your team has to perform them.

1 – I have no motivation, 2 – I’m not that motivated, 3 – Indifferent, 4 – I am somehow motivated, 5 – I am absolutely motivated

- □ Daily meeting
- □ Sprint planning
- □ Sprint retrospective
- □ Sprint review

8) Classify the following Scrum activities according to the level of importance your team believe they have.

1 – Waste of time, 2 – Not important, 3 – Indifferent, 4 – Important, 5 – Crucial

- □ Creating user stories
- □ Creating sprint backlogs
- □ Prioritizing tasks
- □ Effort estimation for each task

9) Classify the following Scrum activities according to the level of motivation your team has to perform them.

1 – We have no motivation, 2 – We are not that motivated, 3 – Indifferent, 4 – We are somehow motivated, 5 – We are absolutely motivated

- □ Creating user stories
- □ Creating sprint backlogs
- □ Prioritizing tasks
- □ Effort estimation for each task

10) Which do you believe are the main reasons for the lack of motivation your team members have in attending Scrum events and perform Scrum activities?

Select all that applies.

- □ Scrum events and activities are boring.
- □ Scrum events and activities are excessively time-consuming.
- □ People lack deep Scrum knowledge.
- □ Because Scrum implies more team communications, people lacking communication skills feel uncomfortable.
- □ Because Scrum roles' boundary is very weak, people feel more pressured into being competent in more areas.
- □ Other (please specify): _______

11) How do you evaluate your team's compliance while using Scrum best practices?

- □ Residual
- □ Bad
- □ Sufficient
- □ Good
- □ Great

12) Imagine that you move to a different organization, and management allows you to choose which development process to follow. Would you choose Scrum?

- □ Yes
- □ No
- □ Don't know/ Don't want to answer

13) How many more years do you believe your team will be using Scrum?

- □ < 1 year
- □ 1 – 5 years
- □ 6 – 10 years
- □ 11-20 years
- □ > 20 years

Filling the questionnaire will take you no more than 10 minutes.

Thank you!

Appendix 2

Interview Guide

Duration: 20 - 40min

Goals of the study

Understand the impact that the gamification app had in the performance and motivation of the studied Scrum team.

Goals of the interview

- Figure out which type of work practitioners do using Jira Software;
- Understand how important practitioners think Scrum is, and to what extent they are motivated to use the framework;
- Gather practitioners' opinions about initiatives that use gamification to increase motivation in the workplace;
- Identify the strengths and weaknesses of the gamification app in improving the adoption of Scrum practices;
- Gather insights on how the gamification app can be improved to better align with the team's needs.

Questions

Warm up

1. Briefly describe your main responsibilities and tasks in your team.
2. For how long have you been working in the organization? And in this team, in particular?
3. Describe your career path (including years of experience).
4. Describe your academic background.
5. Are you Scrum certified?

Jira Software

1. Did you receive Jira Software training?
2. How frequently do you use Jira Software (daily, weekly, etc.)?
3. Which kind of work/tasks do you perform in Jira Software?
4. Which are the major advantages and drawbacks of the tool, in your opinion?
5. Classify your level of satisfaction in working with Jira Software in a scale from 1 to 5, where 1 corresponds to "Strongly Dissatisfied" and 5 to "Strongly Satisfied"

Scrum

Please answer these questions based on your experience before the usage of the gamification app.

1. For how many years have you been using Scrum?

2. How do you classify the compliance with Scrum practices in your team in a scale from 1 to 5, where 1 corresponds to “Very Weak” and 5 to “Excellent”?

2.1 Using the same scale, how do you classify the compliance with the adapted Scrum practices in your team?

3. How important your team believes is to follow Scrum? And how motivated are they?

3.1 And to conduct meetings, in particular?

4. List the main reasons you believe that explain employees’ lack of motivation to follow Scrum.

Gamification

1. Do you play games? How frequently?
2. Which type of games do you like to play (adventure, strategy, social, casual, war, educational, sandbox, etc.)?
3. What do you like to do the most in a game (explore the game space, socialize, create new contents, conquer rewards/spaces/etc., risk, solving puzzles, etc.)?
4. Have you heard about gamification before this project?

4.1 If your answer is “yes”, which applications did you heard about? Did you use any of them?

4.2 If your answer is “no”, what did you believe this approach was about?

Gamification App

1. How much time per day did you spent using the gamification app?
2. About the “Dashboard” screen:

2.1 Classify the relevance of the information presented in this screen in a scale from 1 to 5, where 1 corresponds to “Inexistent” and 5 to “Crucial”.

2.2 Is there any useful information missing from this screen?

2.3 What is your level of agreement with the statement “Project metrics provide important feedback to improve mine and my team’s work” in a scale from 1 to 5, where 1 corresponds to “Strongly Disagree” and 5 to “Strongly Agree”?

2.4 What do you like the most in this screen? And the least?

3. About the “Profile Configurations” screen:

3.1 Classify the relevance of the information presented in this screen in a scale from 1 to 5, where 1 corresponds to “Inexistent” and 5 to “Crucial”.

3.2 Is there any useful information missing from this screen?

3.3 Classify the importance of the feedback provided by individual statistics in a scale from 1 to 5, where 1 corresponds to “Not Important” and 5 to “Very Important”.

3.4 What do you like the most in this screen? And the least?

4. About the “Team” screen:

4.1 Do you consider the information presented in this screen as relevant or interesting?

4.2 If your answer is “yes”, is there anything important you believe that should be here?

4.3 If your answer is “no”, what could be different (removed or added)?

4.4 What do you like the most in this screen? And the least?

5. About the notifications:

5.1 Classify the usefulness of the information presented in pop-ups in a scale from 1 to 5, where 1 corresponds to “Not Useful” and 5 to “Very Useful”.

5.2 Classify the usefulness of the tips to improve issues' specification in a scale from 1 to 5, where 1 corresponds to "Not Useful" and 5 to "Very Useful".
5.3 What do you like the most in this functionality? And the least?
6. About the rewards:
6.1 Do you believe these rewards are meaningful for your team as a whole and for each member?
6.2 What do you like the most in this functionality? And the least?
7. About the score system:
7.1 Classify the clarity of the score system (i.e., if it is easy to understand) in a scale from 1 to 5, where 1 corresponds to "Very Weak" and 5 to "Excellent".
7.2 Classify the balance of the score system (i.e., if it is fair) in a scale from 1 to 5, where 1 corresponds to "Very Weak" and 5 to "Excellent".
7.3 What do you like the most in this functionality? And less?

The gamification app implements two Scrum elements that Jira does not natively support: Roles and Meetings.

1. Classify the importance of supporting Scrum roles and meetings in Jira Software in a scale from 1 to 5, where 1 corresponds to "Not Important" and 5 to "Very Important".
2. Did these elements benefit the project?
3. How could Scrum roles definition in Jira Software benefit your projects?
4. Do you consider the procedure to register meetings as simple and perceptible?

4.1 If your answer is "no", do you have any improvement suggestion?

For the Scrum Master only:

1. Do you think the gamification app motivated your team to follow Scrum?

For the Product Owner only:

1. Do you think the gamification app has led to an alignment between the increment and the sprint's user stories (e.g. having all user stories fully implemented, without bugs, etc.)?

Final

1. Altogether, what do you believe is the best and the worst in the gamification app?
2. Do you believe this gamification app motivated you to follow Scrum best practices?
3. From all functionalities, which do you consider that positively impacts your motivation in using Scrum the most? And the least?
4. Do you regard the experience of using the gamification app as positive?
5. For how long would you use the gamification app?

5.1 If the period is low, what could be done to prevent your motivation to decline?

Thank you for your time and the information you shared with us. We hope this study's results can help improving the gamification app and, consequently, your work.

References

Alsawaier RS (2018) The effect of gamification on motivation and engagement. Int J Inf Learn Technol 35:56–79

Alzoubi YI, Gill AQ (2014) Agile global software development communication challenges: a systematic review. Proceedings of Pacific Asia Conference on Information Systems. AIS, In, pp 1–13

Barata G, Gama S, Jorge J, Gonçalves D (2017) Studying student differentiation in gamified education: a long-term study. Comput Human Behav 71:550–585. https://doi.org/10.1016/j.chb.2016.08.049

Beecham S, Baddoo N, Hall T et al (2008) Motivation in software engineering: a Systematc literature review. Inf Softw Technol 50:860–878. https://doi.org/10.1016/j.infsof.2007.09.004

Buchanan DA, Huczynski AA (2016) Organizational behaviour. Ninth Edit, Pearson

Češka BM (2016) Gamification in the SCRUM software development framework. Masaryk University, Faculty of Informatics

Cho J (2008) Issues and challenges of agile software development with scrum. Issues Inf Syst 9:188–195

Conboy K, Coyle S, Wang X, Pikkarainen M (2010) People over process: key challenges in agile development. IEEE Softw 28:48–57. https://doi.org/10.1109/MS.2010.132

Csikszentmihalyi M (1990) Flow: the psychology of optimal experience. Harper and Row, New York

Dal Sasso T, Mocci A, Lanza M, Mastrodicasa E (2017) How to gamify software engineering. SANER 2017 - 24th IEEE Int Conf Softw Anal Evol Reengineering 261–271. https://doi.org/10.1109/SANER.2017.7884627

de Castro FS (2016) Using Gamification as a collaboration motivator for software development teams : a preliminary framework. Pontifícia Universidade Católica do Rio Grande do Sul

DeMarco T, Lister T (2013) Peopleware: productive projects and teams. Addison-Wesley

Deterding S, Dixon D, Khaled R, Nacke L (2011) From game design elements to gamefulness. Proc 15th Int Acad MindTrek Conf Envisioning Futur Media Environ - MindTrek '11 9–11. https://doi.org/10.1145/2181037.2181040

Dikert K, Paasivaara M, Lassenius C (2016) Challenges and success factors for large-scale agile transformations: a systematic literature review. J Syst Softw 119:87–108. https://doi.org/10.1016/j.jss.2016.06.013

Dorling A, McCaffery F (2012) The gamification of SPICE. Communications in Computer and Information Science. Springer, In, pp 295–301

Dubois DJ, Tamburrelli G (2013) Understanding gamification mechanisms for software development. In: proceedings of the 2013 9th joint meeting on foundations of software engineering. ACM, New York, p 659

Dybå T, Dingsøyr T (2008) Empirical studies of agile software development: a systematic review. Inf Softw Technol 50:833–859. https://doi.org/10.1016/j.infsof.2008.01.006

Foucault M, Blanc X, Falleri J-R, Storey M-A (2019) Fostering good coding practices through individual feedback and gamification: an industrial case study. Empir Softw Eng:1–24

França, ACC, Gouveia, TB, Santos, PCF, et al (2011) Motivation in software engineering : A Systematic Review Update. In: Proceedings on 15th Annual Conference on Evaluation & Assessment in Software Engineering (EASE 2011). IET, pp 154–163

Hajjdiab H, Taleb AS, Ali J (2012) An industrial case study for scrum adoption. J Softw 7:237–242. https://doi.org/10.4304/jsw.7.1.237-242

Herranz E, Colomo-Palacios R, Amescua Seco A, Yilmaz M (2014) Gamification as a disruptive factor in software process improvement initiatives. J Univers Comput Sci 20:885–906

Herranz E, Colomo-Palacios R, de Seco A (2015) Gamiware: a gamification platform for software process improvement. Communications in Computer and Information Science. Springer, In, pp 127–139

Hevner AR, March ST, Park J, Ram S (2004) Design science in information systems research. MIS Q 28:75–105

Hoda R, Murugesan LK (2016) Multi-level agile project management challenges: a self-organizing team perspective. J Syst Softw 117:245–257. https://doi.org/10.1016/j.jss.2016.02.049

Hoda R, Salleh N, Grundy J (2018) The rise and evolution of agile software development. IEEE Softw 35:58–63

Hossain E, Babar MAM, Paik H (2009) Using scrum in global software development: a systematic literature review. In: Fourth IEEE International Conference on Global Software Engineering, 2009. ICGSE 2009. IEEE, pp 175–184

Inayat I, Salim SS, Marczak S et al (2015) A systematic literature review on agile requirements engineering practices and challenges. Comput Human Behav 51:915–929. https://doi.org/10.1016/j.chb.2014.10.046

Kropp M, Meier A (2017) Swiss agile study 2016

Kropp M, Meier A, Anslow C, Biddle R (2018) Satisfaction, practices, and influences in agile software development. In: Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018. pp 112–121

Lombriser P, van der Valk R (2011) Improving the quality of the software development lifecycle with Gamification

Loriggio A, Farias V, Mustaro P (2013) Aplicações de gamificação e técnicas de motivação à aprendizagem da metodologia ágil scrum. VIII International Conference on Engineering and Computer Education, In, pp 326–330

Mahaney RC, Lederer AL (2006) The effect of intrinsic and extrinsic rewards for developers on information systems project success. Proj Manag J 37:42

Marchenko A, Abrahamsson P (2008) Scrum in a multiproject environment: An ethnographically-inspired case study on the adoption challenges. In: Agile 2008 Conference. IEEE, pp 15–26

Marques R, Costa G, Mira da Silva M, Gonçalves P (2017a) A survey on failures in the software development process. In: Proceedings of the 25th European Conference on Information Systems (ECIS), Guimarães, Portugal, June 5–10, 2017. AIS, pp 2445–2459

Marques R, Costa G, Mira da Silva M, Gonçalves P (2017b) Gamifying software development scrum projects. In: 2017 9th International Conference on Virtual Worlds and Games for Serious Applications, VS-Games 2017 - Proceedings. IEEE, pp 141–144

Marques R, Gregório J, Pinheiro F et al (2017c) How can information systems provide support to nurses' hand hygiene performance? Using gamification and indoor location to improve hand hygiene awareness and reduce hospital infections. BMC Med Inform Decis Mak 17:1–16. https://doi.org/10.1186/s12911-017-0410-z

Marques R, Costa G, Mira da Silva M, et al (2018a) Using Gamification for adopting scrum in practice. In: 27th International Conference on Information Systems Development

Marques R, Costa G, Mira Da Silva M, et al (2018b) Improving scrum adoption with Gamification. In: 24th Americas Conference on Information Systems (AMCIS 2018). AIS, New Orleans

McClean A (2015) An exploration of the use of Gamification in agile software development. Dublin Institute of Technology

Meyer AN, Fritz T, Murphy GC, Zimmermann T (2014) Software developers' perceptions of productivity. In: proceedings of the 22nd ACM SIGSOFT international symposium on foundations of software engineering. ACM, pp 19–29

Mora A, Riera D, Gonzalez C, Arnedo-Moreno J (2015) A literature review of gamification design frameworks. In: 7th International Conference on Games and Virtual Worlds for Serious Applications (VS-Games). IEEE

Nicholson S (2015) A recipe for meaningful gamification. Gamification in education and business. Springer, In, pp 1–20

Overhage S, Schlauderer S, Birkmeier D, Miller J (2011) What makes IT personnel adopt scrum? A framework of drivers and inhibitors to developer acceptance. In: proceedings of the 44th annual Hawaii international conference on system sciences. IEEE, pp 1–10

Passos EB, Medeiros DB, Neto PAS, Clua EWG (2011) Turning real-world software development into a game. In: Brazilian symposium on games and digital entertainment. SBGAMES, IEEE, pp 260–269

Pedreira O, García F, Brisaboa N, Piattini M (2015) Gamification in software engineering - a systematic mapping. Inf Softw Technol 57:157–168. https://doi.org/10.1016/j.infsof.2014.08.007

Peffers K, Tuunanen T, Rothenberger MA, Chatterjee S (2008) A design science research methodology for information systems research. J Manag Inf Syst 24:45–77. https://doi.org/10.2753/MIS0742-1222240302

Platonova V, Bērziša S (2017) Gamification in software development projects. Inf Technol Manag Sci 20:58–63. https://doi.org/10.1515/itms-2017-0010

Prause CR, Nonnen J, Vinkovits M (2012) A field experiment on Gamification of code quality in agile development. Psychol Program Interes Gr Annu Conf 64:175–186

Reeves B, Read JL (2009) Total engagement: how games and virtual worlds are changing the way people work and businesses compete. Harvard Business Press

Riemenschneider CK, Hardgrave BC, Davis FD (2002) Explaining software developer acceptance of methodologies: a comparison of five theoretical models. IEEE Trans Softw Eng 28:1135–1145. https://doi.org/10.1109/TSE.2002.1158287

Ryan RM, Deci EL (2000a) Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. Am Psychol 55:68–78. https://doi.org/10.1037/0003-066X.55.1.68

Ryan RM, Deci EL (2000b) Intrinsic and extrinsic motivations: classic definitions and new directions. Contemp Educ Psychol 25:54–67. https://doi.org/10.1006/ceps.1999.1020

Schwaber K (2004) Agile project management with scrum. Microsoft Press

Schwaber K, Sutherland J (2016) The scrum guide

Sekitoleko N, Evbota F, Knauss E et al (2014) Technical dependency in large-scale agile software development. International Conference on Agile Software Development. Springer, In, pp 46–61

Singer L, Schneider K (2012) It was a bit of a race: Gamification of version control. In: 2nd international workshop on games and software engineering: realizing user engagement with game engineering techniques. IEEE, pp 5–8

Souza JP, Zavan AR, Flôr DE (2016) Scrum Hero: Gamifying the scrum framework. Brazilian Workshop on Agile Methods. Springer, In, pp 131–135

Standish Group (2018) CHAOS report

Stettina CJ, Heijstek W (2011) Necessary and neglected ? An empirical study of internal documentation in agile software development teams. In: proceedings of the 29th ACM international conference on Design of Communication (SIGDOC 2011). ACM, pp 159–166

Thaler R, Sunstein C (2008) Nudge: improving decisions about health, wealth, and happiness. Yale University Press, New Haven & London

Turk D, France R, Rumpe B (2002) Limitations of agile software processes. In: Proceedings of the Third International Conference on eXtreme Programming and Agile Processes in Software Engineering (XP 2002)

Verner JM, Babar MA, Cerpa N et al (2014) Factors that motivate software engineering teams: a four country empirical study. J Syst Softw 92:115–127. https://doi.org/10.1016/j.jss.2014.01.008

Version One (2019) 13th State of agile report

Werbach K, Hunter D (2012) For the win: how game thinking can revolutionize your business. Wharton Digital Press

Williams L, Cockburn A (2003) Agile software development: its about feedback and change. Computer (Long Beach Calif) 36:39–43

Wohlin C, Runeson P, Höst M, et al (2012) Experimentation in software engineering. Springer Science & Business Media

Yilmaz M, Connor RVO (2016) A Scrumban integrated gamification approach to guide software process improvement: a Turkish case study. Teh Vjesn - Tech Gaz 23:237–245. https://doi.org/10.17559/TV-20140922220409

Zichermann G, Cunningham C (2011) Gamification by design. O'Reilly Media, Inc

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Rita Marques is a gamification researcher at INOV INESC Inovação and a PhD student at Instituto Superior Técnico, University of Lisbon, Portugal. Her thesis consists in the application of gamification to improve workers adherence to software development processes and methods, such as Scrum. Rita holds a MSc in Computer Science from the same University. Her Master thesis was about the application of gamification to reduce hospital-acquired infections, and was based on her work as a researcher on Instituto de Higiene e Medicina Tropical.

Gonçalo Costa holds a MSc in Computer Science from Instituto Superior Técnico, University of Lisbon. His master thesis was developed at INOV INESC Inovação and consisted in the application of gamification to increase software development practitioners' motivation to implement Scrum techniques in practice.

Miguel Mira da Silva is currently Associate Professor of Information Systems at Instituto Superior Técnico, leader of the research group "Digital Transformation" at INOV, and coordinator of the MISE online MSc degree. Miguel has a PhD in Computing Science from the University of Glasgow and an MSc in Management from the London Business School. Miguel created five companies, published four books and 200 research papers, managed dozens of research and consulting projects, graduated 9 PhDs and 150 MSc students, and created a MOOC about digital transformation. His current interests include digital transformation, IT governance, and online learning.

Daniel Gonçalves is a researcher at the Visualization and Multimodal Interfaces Group of INESC-ID and Professor of Computer Science at Instituto Superior Técnico (IST/UL His research encompasses Information Visualization, Personal Information Management and Gamification, having published over 160 peer-reviewed papers. He managed and participated in several national and European projects and has served as Program Chair of several conferences. Since 2015 he is part of the Editorial Board of the UAIS journal. He is a senior member of ACM and member or the Board of Directors of the Portuguese Computer Graphics Group (the national Eurographics chapter).

Pedro Gonçalves obtained a degree in Information Systems and Computer Engineering at Instituto Superior Técnico in Portugal. He is co-founder and CTO of Xpand IT, a fast growing software company based in Portugal that offers software products and services around the world, in countries such as Germany, United States, United Kingdom and France. During his career, he has assumed the role of systems implementation consultant, software developer, systems architect and product manager.

Affiliations

Rita Marques[1,2] • **Gonçalo Costa**[1,2] • **Miguel Mira da Silva**[1,2] • **Daniel Gonçalves**[1,3] • **Pedro Gonçalves**[4]

[1] Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais 1, 1049-001 Lisbon, Portugal

[2] INOV Inesc Inovação, Av. Duque de Ávila 23, 1000-005 Lisbon, Portugal

[3] INESC-ID, Rua Alves Redol n°9, 1000-029 Lisbon, Portugal

[4] Xpand IT, Rua de Fornelos 77, 4900-709 Viana do Castelo, Portugal