Using Gamification for Adopting Scrum in Practice

Rita Marques *rita.marques@tecnico.ulisboa,pt*
INOV/IST
Lisbon, Portugal

Gonçalo Costa *goncalo.m.da.costa@tecnico.ulisboa,pt*
INOV/IST
Lisbon, Portugal

Miguel Mira da Silva *mms@tecnico.ulisboa,pt*
INOV/IST
Lisbon, Portugal

Daniel Gonçalves *daniel.goncalves@inesc-id.pt*
INESC-ID/IST
Lisbon, Portugal

Pedro Gonçalves *pedro.goncalves@xpand-it.com*
Xpand IT
Viana do Castelo, Portugal

Abstract

Despite the wide adoption of agile methodologies, software development teams still struggle to meet time, budget and scope, partially due to practitioners' lack of motivation to apply agile techniques in practice. In this paper, we present a software tool based on gamification to make Scrum techniques more fun and engaging for practitioners. This paper presents results of the first iteration of a larger research effort that follows the Design Science Research methodology, where a prototype was developed as a Jira Software app and evaluated with a Scrum team in practice. Results suggest that the team's Scrum practices slightly improved after using the app. Quantitative analysis and a set of interviews with the team members allowed to understand that the proposal should be more challenging and the score system more customized. Hereafter the app will be improved based on received feedback.
Keywords: Gamification, software development, software process, agile, Scrum, motivation.

1. Introduction

According to Standish Group's CHAOS report [33], most projects run in 2015 were either failed (19%) or challenged (52%). Thus, software development organizations have been adopting new tools and methodologies[24] proven to improve software projects' success [29].

Agile software development emerged as a flexible, responsive, and team-empowering response to traditional software development and project management [36]. Agile teams are intended to produce working software during short iterations. In Scrum, the most adopted of the agile frameworks [23], practitioners are organized in small teams, plan and track their work iteration (called "Sprint") based on Scrum artefacts, implement user stories (which translate the system's requirements), and communicate in Scrum events [30].

While agile methods and techniques' benefits, like improved product quality and customer satisfaction, have been demonstrated, agile teams are still facing challenges, mostly related to human factors [5, 12]. Such challenges, like improper communication, are partially explained by software development practitioners' lack of motivation to apply agile techniques in practice

[5, 6]. Further in this paper, the term "practitioner" will be used to refer to any of the roles a software engineer can have (like analysts or programmers).

Gamification is a recent but popular approach to make processes related to non-gaming contexts more fun, which can boost motivation. Gamification has been successfully applied in fields like education [1] and health [15], but it is still emerging in software development, where evidence exists that this approach can increase team motivation and help practitioners to focus on development tasks and define better goals [38]. However, to the best of our known very few gamification proposals were evaluated with Scrum teams in practice.

As a proper application of Scrum techniques can positively impact software projects' success, this paper explores the potential of gamification to increase practitioners' motivation in adopting Scrum practices by making them more fun and engaging. A software tool based on gamification was developed as an app for a popular software management tool, so that practitioners do not have to use a separate tool for gamification. This work contributes not only by being one of the few evaluating a gamification solution for Scrum with a real team in industry, but also by its distinct traits, like the usage of varied game elements, whose selection was strongly rooted on feelings and experiences of real practitioners. Detailed analysis of these insights was based on relevant literature, a set of interviews (whose analysis can be found in another publication [19]), and an online survey, but it is out of focus of this paper.

This paper describes the first iteration of a larger research effort following Design Science Research Methodology (DSRM), based on an iterative process and applied with the intent of solving problems involving IT and organizations [27]. The paper is organized as follows. We start with a review of works implementing gamification in software development. Next, we present the proposal and its design and development, followed by a discussion on how the proposal was demonstrated and evaluated in a Portuguese software development company. The paper closes with a discussion of the results and some conclusions and future work.

2. Gamification in Software Development

Gamification adds game elements and game design to non-game processes to engage and motivate people to adopt new behaviors [35]. This approach 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; they just use parts of games (i.e., game elements) in an already existing process [7].

Because software development processes are brain- and collaborative-intensive, comprising some tedious activities, gamification can help making such activities more fun and attractive [26]. 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 [9, 25] and to foster software process improvement initiatives [8, 13, 14]. 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 [16]. Closely related, a framework mapping collaboration issues affecting software development teams with target behaviors and game elements to mitigate those issues was proposed [3]. Moreover, some works are targeted to specific software development processes, like the gamification of a version control system to encourage Computer Science students to commit more frequently [31] and a reputation system to improve the quality of collaboratively written code through documentation [28]. 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 [20]. Yilmaz and O'Connor proposed an integrated gamification approach for Scrumban, where practitioners received points and badges for finishing tasks and helping each other [38]. Loriggio presented a methodology for teaching Scrum, supported by gamification and other theories [17]. Češka prototyped a gamified app to support Scrum development, composed by game elements like points, badges, and progression [4]. Scrum Hero is a gamification framework to support Scrum

software development projects' management, based on game elements like narrative, quests, and rewards [32].

While only the last group of works directly address gamification for agile/Scrum, we can learn from them all, as they target the same players (i.e., practitioners). Overall, 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. All in all, while gamification studies to increase Scrum adoption are emerging, there is much room for growth, namely regarding diversity of game elements and empirical validations in industry.

Apart from research, some commercial tools, like Jiraffe[1] and GetBadges[2], are available, but there are no studies publicly available evaluating the application of these tools in industry.

3. Proposal

To make Scrum techniques more fun and engaging for practitioners, while trying to fill the literature gaps, this study proposes a gamification solution. 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 [35], as it is one of the most mentioned and more complete frameworks to formalize the gamification design process [22].

3.1. 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 improve software projects' success. From here, we derived more concrete goals, based on Scrum's specification and the analysis of practitioners' perceptions and experiences mentioned in Section 1, to build a simple yet complete 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.

3.2. Delineate Target Behaviors and Metrics

Just as for objectives, the behaviours we want players to perform and the metrics for measuring them were defined so that they translated Scrum techniques, thus allowing to understand if and how behaviours changed after gamification. Metrics were defined in sprint and practitioner contexts to understand how behaviors 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 behavior 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. 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

[1] Jiraffe: http://bugpotion.com/ (Accessed: 13/04/2017)
[2] GetBadges: https://getbadges.io/ (Accessed: 13/04/2017)

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 practitioners' motivation, tasks specification's quality, team's cooperation, and project tracking implementation will be given by qualitative means of evaluation, like interviews.

3.3. Describe your Players

The target players of this gamification solution are practitioners, who have been identified as a distinct group of workers not likely motivated by the same things as population in general [2, 11]. Factors likely to motivate these workers include communication between practitioners, management and customer; contribute to the overall success of the project; have feedback on team and individual performance, based on collected data and relevant metrics; and receive rewards and incentives [2, 6, 11, 18, 21, 34].

3.4. 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 feedback, which can be a recognition that (s)he has done a target behavior, or an alert otherwise. Feedback should guide users towards target behaviors and motivate them to take further action. A set of triggers were defined to provide this motivation boost, based on Fogg's Behavior Model [10]. This model proposes three types of triggers to persuade a user to follow a desired behavior in a specific timing: spark (low motivation); facilitator (high motivation but low ability); and signal (maximum motivation and ability). The triggers defined for this proposal are: display reminders when the user is close to reach an achievement or reward, and when an event is close to occur (*signals*); provide immediate feedback after specific behaviors or events, reward a user or the team for performing specific behaviors (e.g. a team resolved all tasks in a sprint) (*sparks*); provide a dashboard with project(s) information (*facilitator*).

3.5. Don't Forget the Fun

Fun elements must be included so that players are likely to engage with the system. As stated in players' description, practitioners like to communicate, so a social component to visualize each other's milestones can promote fun. They should be able to pick the information they want to display here. Team achievements can promote cooperation among practitioners and might help developing a sense of belonging to something greater. A progress bar displaying the user's current experience points (XP) and the XP needed to pass to the next level provide progress information. Achievements will have a visual identification, like badges or gems.

3.6. 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 progress) [35]. This step's outputs are explained in Section 4.

4. Design and Development

The proposal was implemented as an app for a software management tool, Jira Software[3], so that practitioners do not need to use a separate tool for gamification. This tool was chosen due to its flexibility, stability, and strong user base. Jira Software supports the development process and is based on "issues" i.e., problem that needs to be resolved (e.g. a bug). From now on, we will use the term "issue" instead of "task".

[3] Jira Software: https://www.atlassian.com/software/jira/features (Accessed: 24/03/2017)

First, we decided which game elements (displayed in italic) to include to promote defined behaviors. A *score system* awards users with *XP* for certain actions (e.g. resolving an issue), allowing users to progress through *levels*. With levels and XP, we intend to promote a healthy *competition* between users. Greater achievements (e.g. resolving all issues in a sprint) are awarded with both XP 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 behaviors 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. With these features, we want to awake *emotions* (like arousal) on users, allowing them to establish *relationships* and understand their *progression* towards a better adoption of Scrum techniques. As Jira Software does not support Scrum meetings, four issue types were created, one for each meeting type.

Next, these game elements were implemented as features in the app. We decided to leave the marketplace out for now to reduce implementation effort and quickly test the proposal and receive feedback to improve it. The Project Dashboard (Fig. 1**Fig. 1**), displays info and statistics for a project, both general and about the user. On the top left, a small profile provides user information: the profile picture, name, project role, four featured badges, a level progress bar, and XP and gems earned. Right below, an activity feed lists all project's events. On the top right, 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). Below, four project's statistics are displayed to give feedback regarding Scrum practices: Sprint Progress; Effort; Productivity; and Contribution, translating the reasons between resolved/opened issues; effort already completed/assigned to the user in a sprint; current/estimated velocity of a user in a sprint; and number of issues resolved by the user/all users in a sprint, respectively.

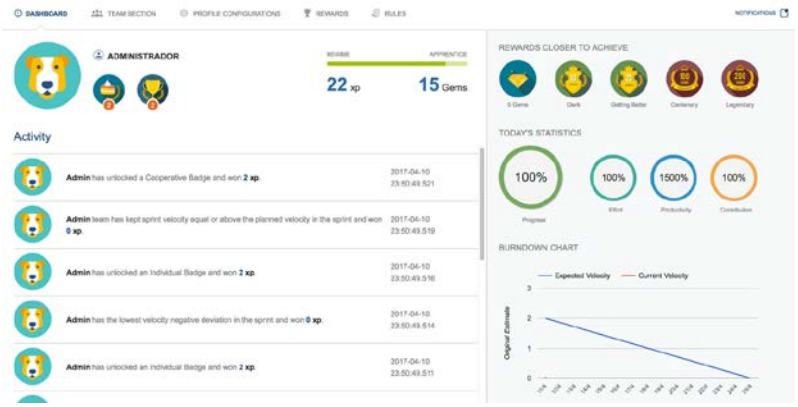

Fig. 1. Project Dashboard.

Profile Configurations is a similar screen that focuses on multiple projects. Users can select at most four of their badges to feature on their profile, and at most four projects to display their statistics. The activity feed displays events from all user's projects. As different teams are assembled per project, the Team screen enables a user to meet the people (s)he works with by seeing their profile, aiming at promoting communication between users. The Rules screen describes the app features' rules. The Rewards' screen displays available rewards, organized by categories. A reward is greyed unless it is awarded to the user, and the number of projects in which it was won is displayed on the bottom.

5. Demonstration

In this DSRM step, the proposed solution must be validated to show that it can be used to solve the research problem. Demonstration tasks are being done in a software provider (products and services), whose software development processes are managed with Jira Software. The participants work in ProjectX (anonymous name), one of the projects managed with Scrum in

the company (which is adapted to their needs, complying with 80-85% of the practices), where a software product is being developed. The team consists of six people and follows sprints with variable duration (between one and four weeks). Critical bugs detected in released versions are fixed in special versions with no sprints. Such issues were left out the quantitative analysis. These workers have experience in using Jira Software on user and/or developer level, which can increase the quality and relevance of their suggestions.

A baseline study and two field studies were conducted to analyze 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 XP 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. Target metrics were calculated based on data extracted from ProjectX's Jira Software instance. Gamification data was analyzed by querying the database with the appropriate management tool, and by studying the activity feed in Dashboard screen, and the Team screen. As only one team was evaluated, global levels are not further analyzed in this paper. The analysis was complemented by interviewing four team members (the other two were off work due to medical leave).

The results of the different demonstration activities are presented separately in the following subsections, where practitioners IDs are displayed in italic.

5.1. Baseline Study

The nine most recent 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, containing 306 issues in total. During data cleaning, we found that 14 issues were resolved several times without being reopened, something not allowed in Jira Software's workflow defined for this project. Thus, these issues were considered outliers and discarded. At last, 292 issues spread by 27 sprints were analyzed, including 82 persistent issues. Mean sprint duration is 13.2 days (st-dev=5.3). Three sprints were complete (11.8%), and 24 contain reopened issues (88.9%). On average, each sprint has 16.5 issues, where 63% are resolved, 59% estimated, and 23.5% reopened. Mean velocity is 32 days. Four sprints were completed without persistent or reopened issues. The practitioner with more issues is *sdrg*, with 110 issues (37.7%). Other practitioners have less than 100 issues each. Also, *sdrg* is the one with more allocated effort (200 days) and persistent issues. The only assignee with all issues estimated is *mfda*, the second with less effort allocated (10 days). Averagely, each assignee has 43.6 issues assigned, where 40.3% are estimated, has 20.4 persistent issues and is allocated with 62.4 days of effort.

5.2. 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 analyzed, comprising four sprints and 31 issues. Mean sprint duration is 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 (25.0%) and three contain reopened issues (75.0%). On average, each sprint contains 13.8 issues and has 63.1% issues resolved, 89.4% estimated, and 39.0% reopened. Mean velocity is 48.2 days. One sprint was completed without persistent or reopened issues. The assignee *pamp* is 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%) is *nmps*. The three assignees who have all their issues estimated (*hjfr*, *tfsi*, and *fcri*) are the ones with fewer issues (four, three, and one, respectively), all persistent. On average, each assignee has 4.4 issues, 2.4 persistent issues, are allocated with 12.7 days of

effort, and around 82.1% of their issues are estimated. There were 32 players in the app, but only seven resolved issues. Player with highest score is *rmbr* (91XP), also the one who resolved more issues, achieving level 3. Following, there are *tfsi* (78XP) and *nmps* (55XP), who achieved level 3, and *fcri* (45XP); *hjfr* (34XP); and *pamp* (29XP), who achieved level 2. All other players had either 6XP or 0XP. On average, each player scored 13.1XP (st-dev=23.13). Globally, players unlocked 60 rewards of five types out of 11 (45.5%). 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.

5.3. Interviews

Four ProjectX's team members (characterized in Table 1), were interviewed, including 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. Interviews were semi-structured, which allowed for flexibility and improvisation, clearing the path to explore emerging lines of research. Interviews were conducted on the same day in ProjectX's company's office and took around 40 minutes. One researcher conducted the interview and took notes using the guide as data collection template. Right afterwards, notes were revised to create a final script. The interview comprised six sections: interviewee's profiling; level of usage and opinion about Jira Software; experience with Scrum; experience with games and gamification; insights about the proposal; and final considerations to sum up the interview.

Table 1. Interviewees' characterization

Questions		**I1**	**I2**	**I3**	**I4**
Position/Role		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 in an IT company (3 years)	None	Software developer internship (9 months)	System maintenance and networks internship
Scrum Certified?		No	No	No	No

The Scrum Master is the one working the longest for the company (six years), while developers started working there less than one year and a half ago. All interviewees hold a degree in IT Engineering. None of them is Scrum certified, but work with Scrum since joining the company. Except for one developer, all interviewees like games and play regularly (at least weekly). Overall, they like strategy, action, and sports games, mostly seeking to solve challenges and unlock rewards. Two developers were not aware of what gamification was, and only the Scrum Master had used a gamification app before.

Concerning the proposal, interviewees said they first used the app frequently, but attention gradually dropped since it is not challenging enough. While the score system improved after refinement, they think it is still not fully aligned with the team's needs, and that admins should be able to configure the XP earned on each issue status' transition. About the players' differences, the Scrum Master was the one with top score (as he is responsible for closing all issues when a version is released), followed by the two testers (who resolve most issues), and then by the three developers (who rarely resolve issues). The players with few XPs and rewards are clients, who can also create issues.

The Team page was the most popular one, as interviewees are very competitive, but they think it should be converted into a leaderboard with more personal information, like the feedback metrics. They considered the information displayed in the Dashboard (like the XPs)

useful, but not essential. They said they forgot to select featured badges and projects in Profile Configurations and think there should be a default selection (e.g., by order of assignment). They did not discern between gems and XP, nor between Dashboard and Profile Configurations pages. The latter might be because information shown was very similar, as the app was active in only one project. They liked to win and showcase badges, but they think the app should have more and diverse badges, so that everyone can unlock these rewards, and not only those resolving issues. Some interviewees are not able to read all the notifications launched on login, but they enjoyed this feedback.

Only the Scrum Master noticed the tips for improving issues' specification, a functionality he found interesting, but thinks it should be linked to a more explanatory description. They believe that having Scrum roles in Jira Software is not important, unless if linked with players' score or profile. They did not use Scrum meetings functionality because they already have a structured process to register them, which they prefer. Yet, they believe this method could be valuable to teams without some structured method.

Overall, interviewees liked the app, which gives useful information and promotes healthy competition, but they think there is large room for improvements. Examples are providing more information and giving real awards in a marketplace. Their experience using the app was positive, as it softened the process formality. Yet, they believe 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.

6. Evaluation

Results presented in previous section are analyzed here. Statistical analysis of the data was supported by a Jupyter Notebook[4], using *pandas* and *SciPy* python packages. A Shapiro-Wilk normality test was applied to all metrics from the baseline and field studies. For five metrics in the baseline study, we had to reject the null hypothesis that samples follow a normal distribution ($p<0.05$). 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. Following these results, all statistical differences between groups were checked using a non-parametric Mann-Whitney's U test. The statistics calculated are presented in Table 2.

Table 2. Statistics for calculated metrics in baseline and field studies

	Baseline		**Field Study**		**Mann-Whitney U test**	
	mean	**std**	**mean**	**std**	**U**	**p-value**
Sprint Context Metrics						
Total of Issues (#)	16.5	10.9	13.8	8.1	58,5	0,813
Estimated Issues (%)	59	33.6	89.4	9.9	51,5	0,906
Resolved Issues (%)	63	29.1	63.1	37.7	25,5	0,098
Reopened Issues (%)	23.5	15	39.0	38.6	37,0	0,331
Velocity (Days)	32	38.9	48.2	30.3	40,0	0,425
Assignee Context Metrics						
Total of Issues (#)	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

6.1. Baseline Study

Results show that the team assigns all issues, thus complying with Scrum practices. Instead, not all issues were estimated, thus influencing velocity, which is based on effort allocation. Most sprints have reopened or persistent issues, showing that not all issues are resolved before the

[4] Jupyter Notebook: http://jupyter.org/ (Accessed 30/04/2018).

sprint. This suggests that the team has problems in estimating tasks and planning sprints. 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.

6.2. Field Study

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 weak evidence that these means can indeed be different (suggesting that assignees are estimating more issues), even though this hypothesis should be rejected.

Regarding gamification, 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 third player with more 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.

6.3. Interviews

Interviewees liked the Team page, the rewards, and the feedback given in the Dashboard and notifications. Inversely, the many pop-ups launched in the login, the lack of leaderboards, and the small number of rewards (and the behaviors that unlock them) were identified as aspects to improve. They suggested that score should be based on workflow transitions (and tailored by the admin) and login pop-ups merged in a single one. Also, we found that the Scrum Master and the Product Owner are the same person, against what is advocated by Scrum.

7. Lessons Learned

Although there is no evidence that differences between studies are statistically different, there seems to be a slight improvement in results. Specially, assignees seem to be estimating more of their issues. Oppositely to what Scrum advocates, the same person takes Scrum Master and Product Owner's tasks in this team, which is not cross-functional, as some practitioners are fully committed with testing activities. Issues are always assigned, but rarely resolved by the end of the sprint. Conversely, sprints' duration, despite variable, complies with Scrum recommendations. However, the team works on issues in versions without sprints to resolve critical bugs, which does not fit Scrum practices.

Based on metrics and gamification's analysis, three clusters of players were identified. 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 several issues. Developers rarely resolve issues, and because the app currently awards players for resolving issues, they received very few XP and rewards. 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.

In Jira Software, and particularly in this team, people assigned to issues might not be the ones resolving them. Because the score system is biased, probably developers are not very motivated now. Even the Scrum Master, who is privileged, eventually became demotivated to use the app. Hence, the score system must cover the whole workflow and rewards should not only be oriented to issue resolution, so that no player is privileged. After this, different elements should be used to motivate each specific cluster. An idea is to create player leagues and define specific rewards for them, which could leverage the roles' feature. Cooperative challenges and rewards did not seem to motivate this competitive team. We believe that the marketplace element could motivate by boosting competition. Issues' specification quality does not seem to have been influenced by provided tips. Still, the constant feedback can impact the team's work,

given identified limitations are fixed. As the mechanism to register meetings was not used, no conclusions can be driven regarding this goal.

In this first iteration, the aim was to test gamification's acceptance and impact on the team, without focusing on its long-term effect. Despite the relevance of this aspect to gamification, collecting feedback early in the process was crucial to understand the importance of each element and to detect improvements, like the score system bug. Thereby, in next steps we can focus on extending gamification effects in time, knowing which the best elements to use are. Future work includes testing the app with different teams to find out if some results, like the need for competition and non-use of the mechanism to register meetings, are generalized or specific to this team.

8. 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 [37].

Internal validity assesses the causal relationship between treatment and outcome. All instruments, like the interview guide, were validated by all authors to prevent issues in the study's design. The critical bug detected in the proposal might have affected the team's behaviour during the field study, although no evidence was found that support this.

External validity translates to what extent the study's results can be generalized to other settings. Because only one team participated in the study, any drawn conclusion 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.

Construct validity shows how the study settings reflect the properties we really intend to capture. To improve validity of the solution, a strong theoretical base was built using many sources and two methods were used to test and evaluate the proposal. Most team members are junior developers with similar game habits, but results might variate with these factors. The extent to which the team applies Scrum practices can influence results as well. 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. 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. 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.

9. 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 is being demonstrated in a company that manages its software development processes using this tool.

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, 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, which do not seem to be challenging enough. Additionally, the long-term effect of gamification should be addressed.

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. In the future, the proposal will be improved based on the results presented in this paper and tested again with this and other teams. An in-depth statistical study of the baseline dataset (e.g., finding correlations between the issue type and resolution rate) could reveal important insights about this team's Scrum practices.

Acknowledgements

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

References

1. Barata, G., Gama, S., Jorge, J., Gonçalves, D.: Studying student differentiation in gamified education: A long-term study. Comput. Human Behav. 71 550–585 (2017)
2. Beecham, S., Baddoo, N., Hall, T., Robinson, H., Sharp, H.: Motivation in Software Engineering: A Systematc Literature Review. Inf. Softw. Technol. 50 (9) (2008)
3. Castro, F.S. de: Using Gamification as a Collaboration Motivator for Software Development Teams : A Preliminary Framework. Pontifícia Universidade Católica do Rio Grande do Sul (2016)
4. Češka, B.M.: Gamification in the SCRUM Software Development Framework. Masaryk University, Faculty of Informatics (2016)
5. Conboy, K., Coyle, S., Wang, X., Pikkarainen, M.: People over process: Key challenges in agile development. IEEE Softw. 28 (4), 48–57 (2010)
6. DeMarco, T., Lister, T.: Peopleware: productive projects and teams. Addison-Wesley (2013)
7. Deterding, S., Dixon, D., Khaled, R., Nacke, L.: From game design elements to gamefulness. Proc. 15th Int. Acad. MindTrek Conf. Envisioning Futur. Media Environ. - MindTrek '11. 9–11 (2011)
8. Dorling, A., McCaffery, F.: The gamification of SPICE. In: Communications in Computer and Information Science. pp. 295–301. Springer (2012)
9. Dubois, D.J., Tamburrelli, G.: Understanding gamification mechanisms for software development. In: Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering. p. 659. ACM, New York (2013)
10. Fogg, B.J.: A behavior model for persuasive design. In: Proceedings of the 4th International Conference on Persuasive Technology - Persuasive '09. ACM (2009)
11. França, A.C.C., Gouveia, T.B., Santos, P.C.F., Santana, C.A., Silva, F.Q.B.: Motivation in Software Engineering : A Systematic Review Update. In: Proceedings on 15th Annual Conference on Evaluation & Assessment in Software Engineering (EASE 2011). pp. 154–163. IET (2011)
12. Hajjdiab, H., Taleb, A.S., Ali, J.: An industrial case study for Scrum adoption. J. Softw. 7 (1), 237–242 (2012)
13. Herranz, E., Colomo-Palacios, R., Amescua Seco, A., Yilmaz, M.: Gamification as a Disruptive Factor in Software Process Improvement Initiatives. J. Univers. Comput. Sci. 20 (6), 885–906 (2014)
14. Herranz, E., Colomo-Palacios, R., Seco, A. de A.: Gamiware: A gamification platform for software process improvement. In: Communications in Computer and Information Science. pp. 127–139. Springer (2015)
15. Johnson, D., Deterding, S., Kuhn, K.-A., Staneva, A., Stoyanov, S., Hides, L.: Gamification for health and wellbeing: A systematic review of the literature. Internet Interv. 6 89–106 (2016)
16. Lombriser, P., van der Valk, R.: Improving the Quality of the Software Development Lifecycle with Gamification. (2011)

17. Loriggio, A., Farias, V., Mustaro, P.: Aplicações de gamificação e técnicas de motivação à aprendizagem da metodologia ágil scrum. In: VIII International Conference on Engineering and Computer Education. pp. 326–330. (2013)
18. Mahaney, R.C., Lederer, A.L.: The effect of intrinsic and extrinsic rewards for developers on information systems project success. Proj. Manag. J. 37 (4), 42 (2006)
19. Marques, R., Costa, G., Mira da Silva, M., Gonçalves, P.: 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. (2017)
20. McClean, A.: An Exploration of the Use of Gamification in Agile Software Development. Dublin Institute of Technology (2015)
21. Meyer, A.N., Fritz, T., Murphy, G.C., Zimmermann, T.: Software developers' perceptions of productivity. In: Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering. ACM (2014)
22. Mora, A., Riera, D., Gonzalez, C., Arnedo-Moreno, J.: A literature review of gamification design frameworks. In: 7th International Conference on Games and Virtual Worlds for Serious Applications (VS-Games). IEEE (2015)
23. One, V.: 11th annual State of Agile Report. (2017)
24. Overhage, S., Schlauderer, S., Birkmeier, D., Miller, J.: 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. pp. 1–10. IEEE (2011)
25. Passos, E.B., Medeiros, D.B., Neto, P.A.S., Clua, E.W.G.: Turning real-world software development into a game. In: Brazilian Symposium on Games and Digital Entertainment, SBGAMES. pp. 260–269. IEEE (2011)
26. Pedreira, O., García, F., Brisaboa, N., Piattini, M.: Gamification in software engineering - A systematic mapping. Inf. Softw. Technol. 57 (1), 157–168 (2015)
27. Peffers, K., Tuunanen, T., Rothenberger, M. a., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. J. Manag. Inf. Syst. 24 (3), 45–77 (2008)
28. Prause, C.R., Nonnen, J., Vinkovits, M.: A Field Experiment on Gamification of Code Quality in Agile Development. Psychol. Program. Interes. Gr. Annu. Conf. 64 175–186 (2012)
29. Riemenschneider, C.K., Hardgrave, B.C., Davis, F.D.: Explaining software developer acceptance of methodologies: A comparison of five theoretical models. IEEE Trans. Softw. Eng. 28 (12), 1135–1145 (2002)
30. Schwaber, K., Sutherland, J.: The Scrum Guide. (2016)
31. Singer, L., Schneider, K.: 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. pp. 5–8. IEEE (2012)
32. Souza, J.P., Zavan, A.R., Flôr, D.E.: Scrum Hero: Gamifying the Scrum Framework. In: Brazilian Workshop on Agile Methods. pp. 131–135. Springer (2016)
33. Standish Group: CHAOS Report. (2015)
34. Verner, J.M., Babar, M.A., Cerpa, N., Hall, T., Beecham, S.: Factors that motivate software engineering teams: A four country empirical study. J. Syst. Softw. 92 (1), 115–127 (2014)
35. Werbach, K., Hunter, D.: For the Win: How Game Thinking Can Revolutionize Your Business. Wharton Digital Press (2012)
36. Williams, L., Cockburn, A.: Agile Software Development: Its About Feedback and Change. Computer (Long. Beach. Calif). 36 (6), 39–43 (2003)
37. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering. Springer Science & Business Media (2012)
38. Yilmaz, M., Connor, R.V.O.: A Scrumban integrated gamification approach to guide software process improvement: a Turkish case study. Teh. Vjesn. - Tech. Gaz. 23 (1), 237–245 (2016)