Improving Scrum Adoption with Gamification

Completed Research

Rita Marques
INOV/IST
rita.marques@tecnico.ulisboa.pt

Gonçalo Costa
INOV/IST
goncalo.m.da.costa@tecnico.ulisboa.pt

Miguel Mira da Silva
INOV/IST
mms@tecnico.ulisboa.pt

Daniel Gonçalves
INESC-ID/IST
daniel.goncalves@inesc-id.pt

Pedro Gonçalves
Xpand IT
pedro.goncalves@xpand-it.com

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 the 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, when compared against the baseline. Quantitative analysis and an interview with the Scrum Master allowed to understand that the proposal should be more challenging and the score system more customized. In the future, the app will be improved based on received feedback.

Keywords

Gamification, software development, software process, agile, Scrum, motivation.

Introduction

According to Standish Group's CHAOS report, most projects run in 2015 were either failed (19%) or challenged (52%) (Standish Group 2015). Thus, software development organizations have been adopting new tools and methodologies (Overhage et al. 2011), proven to improve the 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). Agile teams are intended to produce working software during short iterations. In Scrum, the most adopted of the agile frameworks (One 2017), 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 (Schwaber and Sutherland 2016).

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 (Conboy et al. 2010; Hajjdiab et al. 2012). Such challenges, like improper communication, are partially explained by software development practitioners' lack of motivation to apply agile techniques in practice (Conboy et al. 2010; DeMarco and Lister 2013). 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 and health, 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 (Yilmaz and Connor 2016). However, 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 distinctive characteristics, like the usage of varied game elements, whose selection was strongly rooted on perceptions 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 (Marques et al. 2017)), and an online survey, but it is out of the focus of this paper.

This paper describes the first iteration of a larger research effort following 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 this paper, 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.

Gamification in Software Development

Gamification adds game elements and game design to non-game processes to engage and motivate people to adopt new behaviors (Werbach and Hunter 2012). 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 (Deterding et al. 2011).

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 (Dubois and Tamburrelli 2013; Passos et al. 2011) 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). Closely related, a framework mapping collaboration issues affecting software development teams with target behaviors and game elements to mitigate those issues was proposed (Castro 2016). 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 (Singer and Schneider 2012) and a reputation system to improve the quality of collaboratively written code through documentation (Prause et al. 2012). 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.

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.

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 context.

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. 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).

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, based on Scrum's specification and the analysis of practitioners' perceptions and experiences mentioned in the Introduction, 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.

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 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.

Describe your Players

The target players of this gamification solution are practitioners, who have been identified as a distinct group of workers not probably motivated by the same things as population in general. 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 (Beecham et al. 2008; DeMarco and Lister 2013; França et al. 2011; Mahaney and Lederer 2006; Meyer et al. 2014; Verner et al. 2014).

[1] Jiraffe: http://bugpotion.com/ (Accessed: 13/04/2017)

[2] GetBadges: https://getbadges.io/ (Accessed: 13/04/2017)

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 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. 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) (Fogg 2009). 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*).

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 system. 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. Team achievements can boost cooperation amid 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.

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.

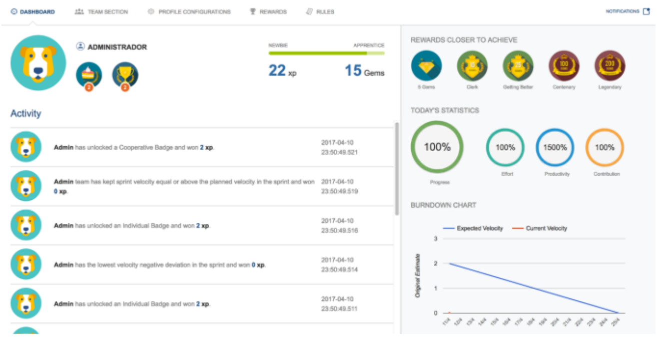
Design and Development

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., problems that need to be solved (e.g. a bug). From now on, we will use the term "issue" instead of "task".

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*. Levels and XP are intended 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. These features offer users a wide range of *emotions*, 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 aside in a first version to avoid this extra implementation effort and quickly test the proposal and receive feedback to improve it. The Project Dashboard (Figure 1), displays info and statistics for a project, both general and about the user. A small profile provides user information: the profile picture, the name, the project role, four featured badges, a level progress bar, and XP 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 to give feedback regarding Scrum practices: Sprint Progress; Effort; Productivity; and Contribution,

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



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

Figure 1 - Project Dashboard

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. 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.

Demonstration

In this DSRM step, the proposed solution must be validated to show that it can be used to solve the research problem. Demonstration activities are being performed in a company providing software products and services, whose software development processes are managed 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 the 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 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.

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. The analysis was complemented by interviewing the team's Scrum Master.

A method was defined to extract data from ProjectX's Jira Software instance to calculate the target metrics. 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[4] 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.

[4] Pentaho Data Integration: http://community.pentaho.com/projects/data-integration/ (Accessed 12/08/2017).

Gamification data was analyzed by directly querying the database, using the appropriate management tool, and by studying the app's activity feed in Dashboard screen, and the Team screen. Because only one project team was evaluated, global levels are not further analyzed in this paper.

The results of the different demonstration activities are presented separately in the following subsections.

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. 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. 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%). Averagely, 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.

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.

Interview with Scrum Master

A semi-structured interview was conducted with ProjectX's Scrum Master which, despite the existence of a guideline, allowed for flexibility and improvisation, clearing the path to explore emerging lines of research. The interview was remote (by Skype), and took around 45 minutes. One researcher conducted the interview and took notes using the interview guide as data collection template. Instantly afterwards, those notes were revised to create a final script. The interview was organized in six sections: interviewee's characterization; level of usage and opinion about Jira Software; experience with Scrum; past experiences with games/gamification; insights about the proposal; and final considerations to sum up the interview. The interviewee holds an MSc degree in Informatics Engineering, and started working nine years ago, six of them in this company and four in ProjectX. He is not Scrum certified, but works with Scrum since he joined the company. Besides taking the roles of Scrum Master in ProjectX, he also performs tasks of Product Owner, which is an unofficial role. He likes games and plays weekly strategy and action games, seeking to unlock rewards and socialize. Concerning the proposal, he used the app frequently at first, but attention gradually dropped since it is not challenging enough. This could be addressed by converting the Team screen, which he enjoyed, into a leaderboard, or by using real prizes. While the score system improved after refinement, he thinks it is still not fully aligned with the team's needs, and that the administrator should be able to configure the XP earned on each issue status' transition. He said he was not able to read all the several notifications launched on the login, but enjoyed this feedback. Despite finding the tips for improving issues' specification interesting, he thinks they should be linked to a more helpful description. He liked the feedback provided in the activity feed and badges, but did not discern between gems and XP. About the players' differences, he identified himself as the one with highest 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. He suggested to improve the Rules section and to provide more feedback metrics. He believes that having Scrum roles in Jira Software is not important, unless if linked with players' score or profile. They did not use meetings because having each team member checking-in a meeting created an extra burden. Overall, the interviewee liked the app, but thinks there is large room for improvements. He believes that at this point the app is not able to motivate practitioners on the long run (although it can initially impact them), but after implementing the suggested enhancements it can be of value to Scrum teams.

Evaluation

Results presented in previous section regarding demonstration activities are analyzed here. Statistical analysis of the data was supported by a Jupyter Notebook[5], 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 1 (sprint context) and in Table 2 (assignee context).

	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	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

Table 1 – Statistics for Sprint Context's Metrics in 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

Table 2 – Statistics for Assignee Context's Metrics in Baseline and Field Studies

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 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.

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

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.

Interview with Scrum Master

The interviewee liked the Team screen and feedback given in the dashboard and notifications. Conversely, the many pop-ups launched in the login, the absence of leaderboards, and the burden caused by individual check-in to Scrum meetings were identified as aspects to improve. He suggested that score should be based on workflow transitions (and tailored by the administrator) and login pop-ups should be merged into a single one. Still, this analysis lacks the perspective of other team members. Also, we understood that the Scrum Master and the Product Owner are the same person, against what is advocated by Scrum.

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 functionality. 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, concerning 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. Nevertheless, this study presents some limitations. The reduced size of the field study's sample hinders the realization of a more in-depth statistical study of the dataset (e.g. correlations between issue type and resolution rate). The study of differences between studies is limited as there is no evidence that these they are statistically significant. A new study, with a larger sample size, could help verify the trends found in this one. According to the Scrum Master, only the team's growth between studies could influence the results. Yet, this and several other factors can influence projects' success, and would be crucial to study and understand their impact on the results.

Conclusion

Addressing practitioners' lack of motivation to adopt agile practices remains a challenge. Many authors 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, with the help of the team's Scrum Master. 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.

Currently, we are interviewing other team members to gather insights from the whole team. 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 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

Beecham, S., Baddoo, N., Hall, T., Robinson, H., and Sharp, H. 2008. "Motivation in Software Engineering: A Systematc Literature Review," *Information and Software Technology* (50:9), Elsevier.

Castro, F. S. de. 2016. "Using Gamification as a Collaboration Motivator for Software Development Teams: A Preliminary Framework," *SBSC 2015 Brazilian Symposium on Collaborative Systems*, Pontifícia Universidade Católica do Rio Grande do Sul.

Češka, B. M. 2016. "Gamification in the SCRUM Software Development Framework," *Technology*, Masaryk University, Faculty of Informatics. (https://doi.org/10.1111/j.1748-5991.2008.00039.x).

Conboy, K., Coyle, S., Wang, X., and Pikkarainen, M. 2010. "People over Process: Key Challenges in Agile Development," *IEEE Software* (28:4), pp. 48–57. (https://doi.org/10.1109/MS.2010.132).

DeMarco, T., and Lister, T. 2013. *Peopleware: Productive Projects and Teams*, Addison-Wesley.

Deterding, S., Dixon, D., Khaled, R., and Nacke, L. 2011. "From Game Design Elements to Gamefulness," *Proceedings of the 15th International Academic MindTrek Conference on Envisioning Future Media Environments - MindTrek '11*, pp. 9–11. (https://doi.org/10.1145/2181037.2181040).

Dorling, A., and McCaffery, F. 2012. "The Gamification of SPICE," in *Communications in Computer and Information Science* (Vol. 290), Springer, pp. 295–301.

Dubois, D. J., and Tamburrelli, G. 2013. "Understanding Gamification Mechanisms for Software Development," in *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*, New York: ACM, p. 659. (https://doi.org/10.1145/2491411.2494589).

Fogg, B. J. 2009. "A Behavior Model for Persuasive Design," in *Proceedings of the 4th International Conference on Persuasive Technology - Persuasive '09*, ACM, p. 1.

França, A. C. C., Gouveia, T. B., Santos, P. C. F., Santana, C. A., and Silva, F. Q. B. 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, A. S., and Ali, J. 2012. "An Industrial Case Study for Scrum Adoption," *Journal of Software* (7:1), pp. 237–242. (https://doi.org/10.4304/jsw.7.1.237-242).

Herranz, E., Colomo-Palacios, R., Amescua Seco, A., and Yilmaz, M. 2014. “Gamification as a Disruptive Factor in Software Process Improvement Initiatives,” *Journal of Universal Computer Science* (20:6).

Herranz, E., Colomo-Palacios, R., and Seco, A. de A. 2015. “Gamiware: A Gamification Platform for Software Process Improvement,” in *Communications in Computer and Information Science* (Vol. 543), Springer, pp. 127–139. (https://doi.org/10.1007/978-3-319-24647-5_11).

Lombriser, P., and van der Valk, R. 2011. *Improving the Quality of the Software Development Lifecycle with Gamification*, Springer: Berlin Heidelberg.

Loriggio, A., Farias, V., and Mustaro, P. 2013. “Aplicações de Gamificação e Técnicas de Motivação à Aprendizagem Da Metodologia Ágil Scrum,” in *VIII International Conference on Engineering and Computer Education* (Vol. 8), pp. 326–330. (https://doi.org/10.14684/ICECE.8.2013.328-332).

Mahaney, R. C., and Lederer, A. L. 2006. “The Effect of Intrinsic and Extrinsic Rewards for Developers on Information Systems Project Success,” *Project Management Journal* (37:4).

Marques, R., Costa, G., Mira da Silva, M., and Gonçalves, P. 2017. “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*.

McClean, A. 2015. “An Exploration of the Use of Gamification in Agile Software Development,” Dublin Institute of Technology.

Meyer, A. N., Fritz, T., Murphy, G. C., and 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. (https://doi.org/10.1145/2635868.2635892).

Mora, A., Riera, D., Gonzalez, C., and 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.

One, V. 2017. “11th Annual State of Agile Report.”

Overhage, S., Schlauderer, S., Birkmeier, D., and 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, E. B., Medeiros, D. B., Neto, P. A. S., and Clua, E. W. G. 2011. “Turning Real-World Software Development into a Game,” in *Brazilian Symposium on Games and Digital Entertainment, SBGAMES*, IEEE, pp. 260–269. (https://doi.org/10.1109/SBGAMES.2011.32).

Pedreira, O., García, F., Brisaboa, N., and Piattini, M. 2015. “Gamification in Software Engineering - A Systematic Mapping,” *Information and Software Technology* (57:1), Elsevier, pp. 157–168.

Peffers, K., Tuunanen, T., Rothenberger, M. a., and Chatterjee, S. 2008. “A Design Science Research Methodology for Information Systems Research,” *Journal of Management Information Systems* (24:3), pp. 45–77. (https://doi.org/10.2753/MIS0742-1222240302).

Prause, C. R., Nonnen, J., and Vinkovits, M. 2012. “A Field Experiment on Gamification of Code Quality in Agile Development,” *Psychology of Programming Interest Group (PPIG) Annual Conference* (64).

Riemenschneider, C. K., Hardgrave, B. C., and Davis, F. D. 2002. “Explaining Software Developer Acceptance of Methodologies: A Comparison of Five Theoretical Models,” *IEEE Transactions on Software Engineering* (28:12), IEEE, pp. 1135–1145. (https://doi.org/10.1109/TSE.2002.1158287).

Schwaber, K., and Sutherland, J. 2016. “The Scrum Guide.” (https://doi.org/10.1053/j.jrn.2009.08.012).

Singer, L., and 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. (https://doi.org/10.1109/GAS.2012.6225927).

Souza, J. P., Zavan, A. R., and Flôr, D. E. 2016. “Scrum Hero: Gamifying the Scrum Framework,” in *Brazilian Workshop on Agile Methods*, Springer, pp. 131–135.

Standish Group. 2015. “CHAOS Report.”

Verner, J. M., Babar, M. A., Cerpa, N., Hall, T., and Beecham, S. 2014. “Factors That Motivate Software Engineering Teams: A Four Country Empirical Study,” *Journal of Systems and Software* (92:1), Elsevier Inc., pp. 115–127. (https://doi.org/10.1016/j.jss.2014.01.008).

Werbach, K., and Hunter, D. 2012. *For the Win: How Game Thinking Can Revolutionize Your Business*, Wharton Digital Press.

Williams, L., and Cockburn, A. 2003. “Agile Software Development: Its About Feedback and Change,” *Computer* (36:6), IEEE Computer Society, pp. 39–43.

Yilmaz, M., and Connor, R. V. O. 2016. “A Scrumban Integrated Gamification Approach to Guide Software Process Improvement: A Turkish Case Study,” *Tehnicki Vjesnik - Technical Gazette* (23:1).