

A SCRUMBAN INTEGRATED GAMIFICATION APPROACH TO GUIDE SOFTWARE PROCESS IMPROVEMENT: A TURKISH CASE STUDY

Murat Yilmaz, Rory V. O'Connor

Original scientific article

In recent years, there has been an increasing interest in tailoring agile development methodologies by combining different agile practices. The adoption of such a balancing approach requires a systematic customization of best practices among agile methodologies. This paper presents an empirical case study for adopting a hybrid Scrumban methodology with an integrated gamification approach, which was conducted in the context of a small-medium enterprise (SME). First, we conducted a focus group to better understand the potential inquiries that might have been useful to improve the development process. Secondly, we employed a cross-sectional survey approach to explore the company personnel's opinions regarding the changes in the process. The survey data was collected from 30 practitioners who were working for the same project in a software development organization in a technology research centre. The descriptive statistics were calculated with paired sample t-tests being used to compare for integration process that contains three stages (i) initial stage, (ii) Scrumban stage, (iii) Scrumban with integrated Gamification stage. The preliminary results of this research support the idea that a set of game elements can be integrated with a hybrid software development methodology to enhance individual and organizational productivity.

Keywords: *game-based agile software development; gamification; Scrumban; software process improvement*

Gamifikacijski pristup s integriranom Scrumban metodologijom u poboljšanju razvoja softvera: turska analiza slučaja

Izvorni znanstveni članak

Zadnjih je godina povećan interes za razvoj agilnih razvojnih metodologija spajanjem raznih agilnih tehnika. Prihvatanje takvog balansirajućeg pristupa zahtijeva sistematsko prilagodavanje najboljih tehnika među agilnim metodologijama. U radu je predstavljena empirijska analiza slučaja za povezivanje hibridne Scrumban metodologije s integriranim pristupom gamifikacije, koji je proveden u okviru malog-srednjeg poduzeća (SME). Najprije smo odabranu grupu upoznali s mogućim upitima, korisnim za poboljšanje razvojnog procesa. Zatim smo jednom preglednom anketom istražili mišljenja osoblja poduzeća o promjenama u postupku. Dobili smo odgovore 30 zaposlenih na istom projektu u organizaciji koja se bavi razvojem softvera u centru za tehnološka istraživanja. Opisni statistički podaci dobiveni su odgovarajućim t-testovima upotrijebljenim kako bi se usporedio postupak integracije koji se sastoji od tri stadija: (i) početni, (ii) Scrumban stadij, (iii) Scrumban s integriranim Gamifikacijskim stadijem. Preliminarni rezultati tog istraživanja podržavaju ideju da se niz elemenata igre može uključiti u hibridnu metodologiju razvoja softvera kako bi se povećala pojedinačna i organizacijska proizvodnost.

Ključne riječi: *agilni razvoj softvera zasnovan na igri; gamifikacija; poboljšanje razvoja softvera; Scrumban*

1 Introduction

Software development is a set of complex socio-technical activities that encompass the life cycle of software production [1]. The planning of such activities, however, is subject to change based on the state and nature of a software project, as well as the software development culture of an organization [2]. Therefore, it is hard to generalize the software development process. While it is agreed that there is no 'one-size-fits-all' process, there have been a number of attempts to customize the software development methodologies regarding project needs [3]. In the past two decades, due to the dynamic and growing business need, there has been a correspondingly significant increase in rapidly changing customer requirements especially in regard to newly emerged software development domains. To cope with such problems, a number of researchers have sought to develop new software development techniques which were mostly collected under the banner of agile development [4]. The vision of agility is based on an immediate response to customer expectations [3], while agile manifesto suggests that people and their interactions should be highlighted over process and tools [5].

Recently, many software small-medium enterprises (SMEs) in Turkey demonstrated an inclination to align themselves with agile development methodologies over traditional approaches. Scrum and Kanban are two of the most frequently used agile methodologies according to an empirical investigation [6]. Many SMEs prefer Scrum to

others, while a few consider Kanban as the primary methodology.

A game is strategic social interaction between two or more people, which can be undertaken for several reasons such as entertainment, education, etc. The concept of games can be considered as an important aspect to explore individuals' interactions, and most importantly a novel way to present social problems [13], which may be based on the actions of individuals who are connected to one other for activities such as software development, deployment, and testing. Consequently, using game thinking should have several benefits for addressing the social issues of software development. We propose that gamifying (i.e. integrating a series of game elements) the software development process would encourage the participants to be socially connected with their software development team, which could motivate them to use their skills and knowledge more effectively.

The goal of this study is to detail the transformation process for adopting Scrumban, (i.e. a hybrid agile methodology), which is additionally improved by integrating a set of game elements. This study therefore aims to address the following research questions: *Do we improve the software development process by using Scrumban integrated gamification framework?*

The paper has been organized in the following way. Section 2 begins by laying out the theoretical dimensions of the research. It covers the background including the details of the software development process that the company used, the definition of gamification, and the idea

of games found in the software engineering literature. Section 3 details the research methodology, which was used to conduct the research. In the next section, we analyse the findings of the research, focusing on the results of interviews and focus group discussions, and the surveys. In the last section, we conclude the paper by discussing our findings and the ideas for further research.

2 Background

2.1 Scrum

Scrum is an agile approach based on well-known industrial process control techniques [7]. It is an incremental framework based on small length plans with continuous feedback, and task prioritization [8]. The goal is to reach the target in small, concrete and iterative steps.

The word "Scrum" comes from the game of rugby. This term was first used by Takeuchi and Nonaka [14]. It is a special kind of formation, an "interlock" of rugby teams at their first engagement. Based on feedback loops, Scrum is another form of incremental software development. Its structure supports "empirical process control" techniques for project management [16].

Unlike traditional methodologies where the extent of a delivery is identified by the time needed, scrum uses time-boxed iterations (i.e. usually same length iterations based on scope and quality of working functionality) [17]. In particular, this means that defining the scope in a limited time frame instead of enabling the scope to create a release length mitigates the project risks. Scrum iterations encircle all phases that are required to deliver the product or a part from the product line of development.

Scrum iterations are a series of steps called sprints. Before each sprint there is a gathering called a sprint review meeting, where developers present the requested work (i.e. desired system characteristics/features) to the product owner or to the stakeholders [8]. These meetings aim to organize the next sprint according to the importance of the tasks (change requests or defect's) defined. Sprints usually settle with regard to potential of shippable pieces of software [8]. In other words, in every sprint, the aim is to keep a product in a state near finalization and have the customer see the added value to the product sprint-by-sprint. As a member of a self-organizing team, team members coordinate their work in daily stand-up meetings. There is usually a member who is called a scrum master who is responsible for dealing with team problems [8]. Dingsoyr et al. [16] investigate the characteristics of Scrum by collecting empirical data from a project based on a small cross-organizational development. Their study indicates that Scrum helps to control the development process by increasing motivational factors. On the other hand, by having customer participation in the process there is a slight decrease in-group learning from feedback loops.

Cohn and Ford [15] claim that Scrum is more suitable for web projects. Moreover, it is found suitable for producing software products in turbulent environments with high expectations. On the other hand, Kanban is an information management methodology which is frequently used in industrial development. Instead of a time-boxed approach, it uses a flow-based alternative. In

addition, Kanban uses a board structure to visualize the workflow in which tasks are represented, assigned, and prioritized [9]. Most importantly, tasks shown in a Kanban task-board are assigned by using a pull mechanism that limits the work in process [10], and ultimately the production is constrained with the notion called *minimum marketable functions* [11]. In addition, an increasing number of software organizations recognize that customizing an agile methodology regarding company needs is vitally important [12].

2.2 Scrumban

Based on a set of elements borrowed from Scrum and Kanban, Scrumban is a hybrid agile methodology designed to cope with dynamically changing customer requirements and frequent coding problems. Unlike Scrum, it does not have sprints, which were found problematic by some software practitioners and eventually caused some team members to wait for each other in some cases [9]. However, it possesses some of the best practices of the Scrum development, such as daily stand-up meetings, etc. It is an agile mash-up, which can be considered as a combination of Scrum and Kanban in the form of a pull-based project management methodology [18]. Scrumban borrows the best practices of Scrum such as user-stories, daily stand-up meetings and self-organized team aspects. However, a Scrum task board is not enough to reflect the changes where sprints were replaced with a Kanban style pull-driven coordination mechanism with work-in-progress (WIP) limitations [19]. These limitations control how many units of work can be processed at once. The pull mechanism ensures the workflow is improved as the software teams improve their processes. However, a challenge here is that optimizing the WIP heavily depends on the selection of the right values to achieve an optimal flow [9].

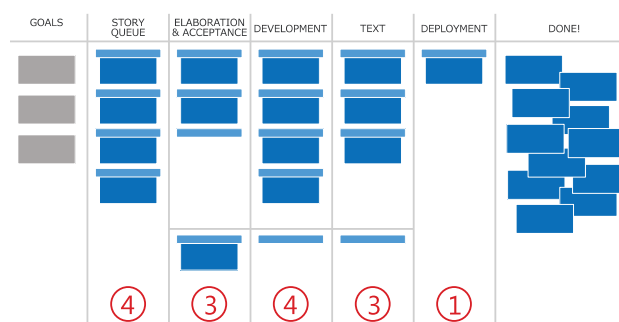


Figure 1 A Conceptual Scrumban Board

In addition, Scrum requires an estimation before each iteration whereas a fixed-size event-driven backlog may be more beneficial for reducing waste, i.e. creation and discussion of too many user stories [19]. Such an approach is necessary for software projects that have ambiguous user stories, and further helps to cope with software projects with frequent programming defects [12]. To deal with problematic issues better, Scrumban requires a good visible representation (see Fig. 1). WIP helps software teams to limit multitasking thereby improving software development productivity.

2.3 Defining games and gamification

The broad use of the word game is sometimes equated with the concept of an interactive system that is based on a series of rules and a number of participants who have a type of conflict to achieve an actual or a potential outcome [20]. A game is an activity of play which is composed of players, rules and payoffs. The strategic interactions of the participants usually happen in a setting such as a game board or by using a communication tool such as a computer. A player is a participant in a game who can be considered as a motivated individual for a planned goal. A game may leverage the desire of an employee to compete where researchers believe that such a motivation could be useful for improving the organizational success. Consequently, the term gamification is coined, which refers to the use of game thinking in non-game contexts [21]. Although the idea is not new, the global rise of social networking has expanded the importance of game like structures. The actual implementations of gamification suggest that a process designer should add several mechanisms to business processes such as reputation points, progress bar, leveling system for the participants, achievements and badges. In short, the concept of gamification can be used to produce highly motivated participants that are working on routine activities.

Zichermann and Cunningham [21] point out that gamification may have diverse meanings for different individuals. Deterding et al. [22, pp. 2] define gamification as: "use of game design elements in non-game contexts" [23]. From a marketing viewpoint, Huotari and Hamari [24] identify gamification as an activity in which quality of services are improved by using game-like features, which could potentially be useful for marketing services. This view is not fully supported by Deterding et al. [25] who argue that gamified applications should only employ elements of a game unlike a serious game, which eventually proposes a complete game construction. A potential limitation with this explanation is that first there is no consensus for the exact definitions of game elements [23]. Secondly, it seems that the understanding of the boundary between the game and game element is questionable. For example, a game consists of self-similar elements, which are constructed by using a set of gamified parts; therefore, game itself may be an end product of a gamification process. Thirdly, this researcher argues that initial descriptions fail to take process thinking into account. In fact, one possible outcome of a gamification process might be likely to be a game itself. Even in some cases, a game can be gamified in the actual process (e.g. see [26]). Lastly, Groh [27] discusses the challenges and strategies for facilitating and promoting gamification where he highlights the fact that the concept should be more utilized for conducting rigorous research regarding the pros and cons. In the context of this paper;

Gamification is a transformation process in which interaction patterns, game mechanisms, reusable game components are operationalized to solve problems in an intended environment that is situated within a real world context. [28] (p. 2)

Gamification should be considered as a process based on the two observations. First, we suggest that it is important to define a boundary such as an entry and exit criteria for validating a gamified product. Second, we believe that it is required to measure the difference between the initial state and the final state of a gamified approach. The goal of gamifying a product or a service is to convert "an intrinsic motivation to an extrinsic reward" [21] (p. 27). To gamify a system, game elements should be identified and game mechanics should be tailored and integrated in the selected non-game context. Although using game elements in the software engineering context is recent, we believe that incorporating it with a software development process should (i) increase the development velocity by adding extra motivation, (ii) help software practitioners to focus on more clear goals by shortening the process tasks, and ultimately reveal the goals to make them more achievable.

From a software development perspective, several researchers have suggested using a game thinking approach. For example, Cockburn [29] suggested that the cooperative and iterative nature of software development is somehow similar to a set of invention and communication games. It was claimed that development is a game that is limited with the project resources. Based on the skills of software decision-making, Baskerville et al. [30] point out that a software project can be viewed as a balancing game. To teach CMMI to the software practitioners, Holeman [31] designed a board game for teaching software process improvement. Ogland [32] proposed a solution by utilizing drama theory to form a game playable by software development organizations, and sought ways to improve the software development process.

2.4 Gamification as a design process

Gamification can be considered as a framework where the participants are systematically motivated with targets that have small and quantifiable increments based on continuous feedback and interactions. To keep the participants in the flow [20], the design approach should also offer meaningful choices, which derives challenges and opportunities, letting your participants keep playing. For example, it is important to single out the participants' behaviors which should be rewarded. These rewards can be identified as follows: (i) rewards that a participant can get in any circumstances, (ii) rewards that can be taken as a consequence of interaction, (iii) the rewards that participant can get after finishing a task, and finally (iv) the rewards that are taken based on performance. In addition, the rewards can be presented in (i) a continuous-rewarding, (ii) constant-rewarding, or (iii) variable-rewarding participants [33]. However, similar to any design process, the quality of the process heavily relies on the talents of the designer. The process is evaluated by using continuous feedback loops, which are useful to keep the players motivated, which can be considered as one of the important aspects of any game design. Moreover, these loops are also useful for observing the behaviour patterns of participants.

According to Werbach and Hunter [34], the design process should be implemented in six main steps. First

and the foremost, we should highlight concrete business goals, which should be represented in terms of a ranked-list of business objectives. These items should address the needs of gamifying. To cope with environmental turbulence in business landscapes, these requirements should be subject to change and need to be assessed periodically.

After defining the reasons of gamification in the first step, secondly, the expected behaviours of participants - and the ways to measure them - should be identified. Most importantly, these behaviours should promote the business objectives. The desired behaviours can be quantified using a measurement technique such as points. For example, designer can give several points for the desired actions to keep individuals motivated accordingly.

Thirdly, we should model our participants by predicting gamer characteristics and their relationship with the gamified system. Therefore, the characteristics of the participants should be identified. A well-known personality profiling model used in games are Bartle's player types [35]: achievers, who prefer to level up quickly and earn what the gamified system offers, explorers are the players who are interested in exploring new content, socializers are the type of players who enjoy others company, and prefer to be social and helpful. Killers are the type of players who enjoy dominating others, and mostly prefer competing with other participants.

Fourthly, we design game activity cycles namely engagement and progression loops. Engagement loops can be considered as micro management of participants, which guide the behaviors of participants with instant and continuous feedback while progression loops require a stair-like design, e.g. quests, boss fights, etc. A level in a gamified system may end-up with a major challenge where a player can have some intense activities, later having a resting state or a point.

Fifthly, the gamified system should include elements of fun and ultimately a good design means a participant may join the system voluntarily. Lazzaro [36] suggests that there are four types of fun: (i) easy fun, which consists of lightweight activities for having simple fun, (ii) hard fun that includes complex and challenging activities, (iii) people fun, which includes social activities with lots of interactions, and (iv) serious fun is a type of fun where participants think that activities are beneficial and meaningful.

Finally as the sixth step, in light of the information collected in the previous steps, we deploy the necessary instruments and construct a road map to implement design decisions.

3 Study background

The data composed for this research study was collected over a 12-month period (2012 ÷ 2013) from 30 software development practitioners who were working for the same project in an SME in a technology research centre at the Middle East Technical University.

The company acted as a private development centre that performed only innovative game or simulation projects for museums and local institutions (e.g. cultural heritage games and interactive software for virtual

museums). The project had an expected project delivery time of 14 months. The company had 12 personnel who worked as software developers, 3 software testers, 3 individuals who were systems analysts, 6 individuals who were working as graphical designers and 6 people were recruited as 3D artists. The main organization consists of three sub teams of 10 people equally distributed (i.e. four developers, one software tester, one system analyst, two graphical designers, and two 3D artists).

Although the majority of software practitioners that were hired were familiar with Scrum (e.g. some attend scrum master and scrum product owner trainings), the initial software development process had an uncontrollable lifecycle with intensive scrum sprints, which was followed by some idle time for software practitioners for reasons such as delays in feedback from the client. Many project tasks are complex, based on mostly non-repeatable activities likely to produce complex zones. Due to the complexities on multimedia and game development projects, a significant amount of development tasks require reordering of work items in 2-3 weeks. However, clients naturally expect deliverables on time within a limited budget with high quality standards.

The original intention of the study was to guide the software development organization during its translation stage from one agile development methodology to another. All assessments and software process improvement suggestions were designed and produced regarding the specific company needs. The objective of these improvements was to produce better software development outcomes through the integration of novel techniques. The researcher provided a set of recommendations for best practices that are hybridized from both Kanban and Scrum development. In addition, workflow and plans for education of the staff that were equipped with new technologies and tools were also created.

3.1 Methodology

The process used in this study has two main phases. The first phase comprises the transformation activities of software development organizations from Scrum to Scrumban. Next, a proposed gamification framework was used to improve the software development process. The goal is to increase the levels of interaction among the software practitioners and modelling their motivational activities. Initially, a focus group was conducted to single out the opinions of participants regarding the planned integrations.

To understand the implications of each step in the study, data was gathered in each integration step via a survey instrument. This is the first study to undertake a longitudinal analysis of a complete transformation process of an SME organization. Therefore this part of the study was conducted in the form of a survey. There are two main integration steps of the study, which lasted for 12 months. Firstly, the survey was conducted at the initial stage. Six months after adopting Scrumban initial results were discussed and the potential game elements were highlighted in a retrospective meeting. To explore the changes in individuals' opinions, the survey was conducted again. At the final stage, a set of game

elements were integrated to the software development process. Lastly, six months after that a survey has been conducted once again to observe the opinions of participants (see Fig. 2). This survey investigated the usefulness of the integration process from the viewpoint of company personnel.

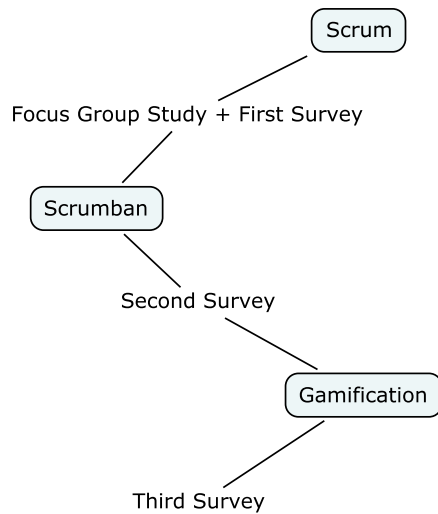


Figure 2 The sequential steps of the research process

The study utilizes a variant of an action research approach [37]. It was designed to explore the perceptions of software practitioners regarding the transformation process and to foster their awareness about the changes. Using the action research method, the problem is to produce practical knowledge where the boundaries of the investigation are constrained by a research team, i.e. a group of participants with the guidance of an expert researcher. Using a collaborative process, preliminary findings are evaluated by the contributors (i.e. research team) and the definition of the problem can be altered whenever necessary [38]. The goal of this approach is to find a solution to a problem by adjusting a set of parameters to alter the outcomes. It certainly requires an expert from the field to deal with change management. The desire for selecting the action research for this study is to contribute to the direction of the research (i.e. take some actions to shape the working process if need be).

3.2 Case study

The first phase of the case study started with the inquiry of the researcher regarding the improvement of the development process that was used by the software development organization. To investigate the problems that were encountered in the software development organization, the researcher suggested starting with a focus group meeting where all software practitioners were encouraged to express their opinions to improve the software development process. The focus group discussion was facilitated by one of the researchers who ensured that all participants had spoken enough to express their thoughts about the problems that were experienced using the Scrum methodology. To initialize the discussion, moderator started by asking questions such as "In which part of the process do you think the problem occurs in the communication patterns?" "Have you

observed some missing procedures related to information exchange in teams?", "Are there some visible gaps between the software development teams and design teams?", "Are there any problems related with user stories in Scrum Methodology?", and "Why some teams cannot leave the development process even when a sprint has been finished?"

In light of the focus group discussion, a variety of perspectives were expressed, and several issues were identified. Firstly, there was a communications gap between the product owner and software development teams. This gap had been reported to cause several problems in active project groups especially among designers and developers. It was reported that during several sprints, the software-testing group experienced short but incremental delays on feedback loops. In addition, there was confusion among stakeholders regarding the end of a Scrum sprint. In such cases, performance tests were postponed and ultimately the process of documentation was reported as laborious and slow. The focus group discussion also revealed that many short term solutions which had emerged in the process had a negative impact on long term outcomes. Therefore, some strategic targets were modified. To cope with sprint deadlines, some developers claimed that during a sprint it was not easy to establish the standards in coding quality. We concluded that Scrum is working very well with project planning tasks while there were some serious issues with programming tasks. For example, some software teams finished their assigned task early and waited for other teams to catch up. In the final part of the focus group study, the team discussed the factors that might potentially affect the change process. To measure the implication of the change in the process on software practitioners, we adopted a 12-survey questionnaire from a similar study [39]. To capture the preliminary status, we conducted a survey on 30 personnel of a small-medium software development organization.

Table 1 Process transformation questionnaire first round (adopted from [39])

Survey questions	$N = 30$, Mean = 3,2	Mean
Q1	My tasks are clear to me	3,9
Q2	Changing requirements are affecting my work	3,8
Q3	Team leader describes, prioritizes, and communicates the requirements effectively	3,5
Q4	Obstacles do not impede team progress	3,7
Q5	Group meetings improve efficiency	3,2
Q6	I am satisfied with the software development process	3,2
Q7	The software development process supports innovation	2,9
Q8	I believe I am improving myself	3,4
Q9	The development process positively affects group performance	3,6
Q10	The social interactions during the process improve productivity	2,8
Q11	I got sufficiently rewarded for my individual achievements	2,1
Q12	Social connections in my team are strong enough	2,3

Tab. 1 shows the results of the survey. The questions were analysed based on responses on the 5-point Likert agree-disagree scale. For each statement, there were five choices: strongly disagree (1), disagree (2), neutral (3), agree (4), and strongly agree (5) accordingly.

3.3 Process change: Scrum to Scrumban

After the initial survey, the software process was changed from Scrum to Scrumban. The task board was altered and iterations were replaced with a pull-driven coordination mechanism based on the notion of work-in-progress. The researcher was appointed as the change manager and attended monthly meetings. To investigate the opinions of software practitioners, preliminary survey was conducted after a six month period. The questionnaire required respondents to give information on Scrumban process and its actual implementation. Descriptive statistics were obtained from the two questionnaires.

In the preliminary questionnaire, 12 questions were evaluated by the entire personnel (i.e. conducted for Scrum). The average in the first questionnaire on the Likert scale was found to be 3,2 (see Tab. 1) where as for the second round (i.e. conducted for Scrumban) the average on the Likert scale was 3,7 (see Tab. 2).

Table2 The process transformation questionnaire second round

Survey questions	<i>N</i> = 30, Mean = 3,7	Mean
Q1	My tasks are clear to me	4,1
Q2	Changing requirements are affecting my work	4,3
Q3	Team leader describes, prioritizes, and communicates the requirements effectively	4,3
Q4	Obstacles do not impede team progress	4,5
Q5	Group meetings improve efficiency	3,5
Q6	I am satisfied with the software development process	3,8
Q7	The software development process supports innovation	3,9
Q8	I believe I am improving myself	3,8
Q9	The development process positively affects group performance	3,8
Q10	The social interactions during the process improve productivity	2,9
Q11	I got sufficiently rewarded for my individual achievements	2,8
Q12	Social connections in my team are strong enough	2,7

Table 3 Paired-sample t-test statistics

	Mean	Number of questions	Standard deviation
Scrum (S)	3,2	12	0,58
Scrumban (SB)	3,7	12	0,61

To determine why the differences between the means of the two samples are significant ($p < 0,05$), the data obtained from the two questionnaires was analysed using a paired-sample t-test. This test statistically analyses the difference between the opinions of software practitioners regarding the transition process. In other words, this study investigated the opinion of respondents whether or not the new software development methodology (i.e. Scrumban)

made a better impression than the previous Scrum approach.

As shown in Tab. 4, there is a 0,5 difference between the means of the two questionnaires. The calculated *t* value (6,12) is higher than the critical table value (1,79), and therefore this shows that the new process has a positive effect on the software development company.

Table 4 Paired-sample t-test (from Scrum to Scrumban)

	Mean	s.d.	<i>t</i>	<i>df</i>
SB _{mean} - S _{mean}	0,5	0,28	1,79	11

3.3 Improvements with Scrumban

The results of the transformation process suggest several updates. First, the Scrum task board was updated to reflect the new development process, which includes a WIP constraint that limits the work in process within each step. In a traditional time-boxed Scrum process, team velocity is measured in terms of story points where a team will be evaluated by how many stories they realized. However, in the new hybrid methodology, to cope with continuously changing requirement, the work in process is controlled using a pull system. This provides a fixed size backlog where work estimation is not performed in every iteration. To monitor the software development process, the alterations were tracked using a cumulative flow diagram.

In addition, daily stand-up meetings, which were held previously, have been improved. The meetings were performed similarly to the focus group sessions; the agenda was redesigned based on a simple but written protocol similar to semi-structured interviews. Consequently, these stand-up meetings became a place to produce systematic and comparable information. Based on these meetings, the task board was updated. Finally, time-boxed iterations were completely removed from the process. Instead, a task board based information system was utilized to control the software production where the balancing was based on capacity constraints.

3.5 Gamification

To address software practitioners' social concerns (i.e. intrinsic and extrinsic motivation), researchers suggested adding game elements in addition to the Scrumban process. To this end, we used a six-step gamification approach to improve the software development process.

The business objectives of the software development organization were identified as follows:

- Increase the intrinsic and extrinsic motivation of software practitioners using game elements.
- Improve the social interactions of practitioners who were working in the software development process by using interactive elements.
- Increase the awareness of community, and provide continuity of the practice.
- To preserve the perception of progress during the software development process.
- To increase the fun factor during the software development process.

Design of targeted behaviours

- *Finish more tasks*: As frequently seen in games, based on their difficulty level, a set of tasks (i.e. quests) was designed to keep individuals motivated to finish their active tasks. Such quest-based categorization was also beneficial for identifying the problematic pieces that may not be visible upfront.
- *Encourage cooperation*: Software artefacts, which can be developed by collaborative teamwork, were identified. Based on the user stories that were identified in the requirement analysis, the levels of difficulties of tasks were revealed. As participants were recognized as players, they are able to choose a task regarding their player's level and quest difficulty.
- *Share your experience*: Participants who finished their tasks were encouraged to help other practitioners. The goal was to benefit novice practitioners with tacit knowledge of the experience individuals. Such behaviours were highly rewarded, which also promised to increase the participant's system-wide reputation.

Identification of player characteristics

- The software development organization had three main roles of software development: (i) software developer, (ii) 3D and graphical designers, (iii) software testers. The age varied between 20 ÷ 40 and a total of 30 personnel (23 male and 7 female) were actively working in the software development organization.
- During a set of unstructured interviews with company personnel regarding their Bartle's player types: 5 individuals defined themselves as achiever, 8 were explorer, 7 were socializer, 10 were killer types.

Designing activity loops

- In addition to a Scrumban task board, a point system was added where participants earn points regarding their jobs. Finishing a task, sharing information and cooperation were quantified. User stories, which were captured during the requirement analysis, were also ranked and integrated into the points system. Most importantly, some tasks were only available to those with enough points.

Designing the elements of fun

- Achievements were rewarded with badges not only used for keeping the sense of progression but also used for creating a set of surprises for participants.
- All software practitioners were represented by an avatar, which were also added to the task board as a major enhancement. This approach also provides an easy tracking system for tasks versus participants.

Deploying necessary instruments

Using the proposed gamified approach, the workflow was revised regarding selected game design principles. The progression loops in the process were more easily followed. In addition, some feedback was also presented in the task board.

4 Scrumban and Gamification

Six months after adding gamification to the Scrumban process, the survey was conducted once again and its results were compared with the Scrumban survey. The questionnaire required respondents to give information on the enhanced process, which was basically a combination of Scrumban and gamification approaches.

As in the previous state, the software practitioners answered 12 survey questions. The average in the first Likert scale was 3,7 (see Tab. 2) whereas the third round average in Likert scale was 4,09 (see Tab. 5).

To determine if the change in the opinions of software practitioners was significant ($p < 0,05$), a paired-sample t-test was performed on the means of the two questionnaires.

As shown in Tab. 7, there is a 0,39 difference between the means of the two questionnaires. The calculated t value (6,71) is higher than the critical table value (1,79), and therefore we can consider that Scrumban+Gamification integration shows improvement in participants understanding of the software development process.

Table 5 Process transformation questionnaire third round

Survey questions	$N = 30$, Mean = 4,09	Mean
Q1	My tasks are clear to me	4,3
Q2	Changing requirements are affecting my work	4,5
Q3	Team leader describes, prioritizes, and communicates the requirements effectively	4,6
Q4	Obstacles do not impede team progress	4,7
Q5	Group meetings improve efficiency	3,8
Q6	I am satisfied with the software development process	4,2
Q7	The software development process supports innovation	4,2
Q8	I believe I am improving myself	4,3
Q9	The development process positively affects group performance	4,4
Q10	The social interactions during the process improve productivity	3,3
Q11	I got sufficiently rewarded for my individual achievements	3,2
Q12	Social connections in my team are strong enough	3,6

Table 6 Paired-sample t-test statistics

	Mean	Number of questions	Standard deviation
Scrumban (SB)	3,7	12	0,61
Scrumban+ Gamification (SBG)	4,09	12	0,50

Table 7 Paired-sample t-test (from Scrumban to Scrumban+Gamification)

	Mean	s.d.	t	df
$SBG_{mean} - SB_{mean}$	0,39	0,20	1,79	11

5 Conclusion and future work

The present study was designed to determine the effectiveness of the transformation process from Scrum to Scrumban and further Scrumban to Scrumban and Gamification. Based on the initial [40], and the current

versions of the study, the following conclusions were obtained. Scrumban solved some of the major issues raised in focus group studies by software practitioners, which were observed using Scrum methodology. In particular, during a sprint, problems regarding a group of user stories were ignored and therefore potential issues were masked for a significant amount of time. Scrumban was also found more compatible with the workflow of software development organization. For example, in Scrum, estimation in the first group of sprints is highly productive. In later iterations, however, task prioritization and proper estimation problems became acute and visible. Our findings suggest that such issues may emerge because of problematic user stories or due to software programming defects. To address these problems, we proposed to add a WIP-based pull mechanism with methods to improve the visualization of software development.

To increase the motivation of software practitioners, we proposed an integrated gamification approach. This study shows its actual implementation with initial results from the field. In general, the additional game elements seemed to increase the motivation and engagement of software practitioners within the process. We found evidence that using Scrumban and gamification provides a systematic performance improvement, which was checked by asking practitioners (in a three step survey method) where paired-sample t-test statistics showed that there was a statistically significant difference ($p < 0,05$) in survey results between initial state with Scrumban, and Scrumban with Scrumban and Gamification. Although the findings of the present study suggest that a Scrumban integrated gamification framework collectively improves the software development process, the results of the study should be interpreted with caution. A large sample from a group of SMEs might be beneficial for better identification of the impact of the transformation process. Finally, our findings provide some support for the conceptual premise that software development organizations benefit from a Scrumban integrated gamification approach. However, further empirical investigations of different software companies are required to validate such a proposition. This study is oriented towards both researchers and practitioners who are interested in the results of methodological transformation efforts in an agile software development organization.

6 References

- [1] Dittrich, Y.; Floyd, C.; Klischewski, R. *Social Thinking-Software Practice*. The MIT Press, Cambridge, MA, 2002.
- [2] DeMarco, T.; Lister, T. *Peopleware: productive projects and teams*. Dorset H. Publishing, New York, NY, 1999
- [3] Boehm, B.; Turner, R. *Balancing Agility and Discipline: A Guide for the Perplexed*. Addison-Wesley Professional, Massachusetts, 2003.
- [4] Abrahamsson, P.; Salo, O.; Ronkainen, J.; Warsta, J. *Agile software development methods: Review and Analysis*. // Volume VTT Publications 478. Technical Research Centre of Finland, Finland, 2002
- [5] Fowler, M.; Highsmith, J. *The agile manifesto*. // *Software Development Magazine*. 9, (2001), pp. 28-35
- [6] Yilmaz, M.; O'Connor, R.; Clarke, P. A systematic approach to the comparison of roles in the software development processes. // *Software Process Improvement and Capability Determination*. Springer-Verlag. 290, (2012), pp. 198-209. DOI: 10.1007/978-3-642-30439-2_18
- [7] Larman, C. *Agile and iterative development: A manager's guide*. Addison-Wesley Professional, Massachusetts, 2004.
- [8] Schwaber, K. *Agile project management with Scrum*. Microsoft Press, USA, 2009
- [9] Ladas, C. *Scrumban-essays on kanban systems for lean software development*. Modus Cooperandi Press, USA, 2009
- [10] Anderson, D.; Concas, G.; Lunesu, M.; Marchesi, M. Studying lean-kanban approach using software process simulation. // *Agile Processes in Software Engineering and Extreme Programming*, 1, (2011), pp. 12-26. DOI: 10.1007/978-3-642-20677-1_2
- [11] Biffi, S.; Aurum, A.; Boehm, B.; Erdogmus, H.; Grunbacher, P. *Value-based software engineering*. Springer, Germany, 2005
- [12] Conboy, K.; Fitzgerald, B. The views of experts on the current state of agile method tailoring. // *Organizational Dynamics of Technology-Based Innovation: Diversifying the Research Agenda*. // IFIP International Federation for Information Processing. Springer, US, 1(2007), pp. 217-234. DOI: 10.1007/978-0-387-72804-9_15
- [13] McGonigal, J. *Reality is broken: Why games make us better and how they can change the world*. Penguin Press, USA, 2011
- [14] Takeuchi, H.; Nonaka, I. The new product development game. // *Harvard Business Review*. 64, (1986), pp. 137-146.
- [15] Cohn, M.; Ford, D. Introducing an agile process to an organization [software development]. // *Computer*. 36, 6(2003), pp. 74-78. DOI: 10.1109/MC.2003.1204378
- [16] Dingsøy, T.; Hanssen, G.; Dyba, T.; Anker, G.; Nygaard, J. Developing software with scrum in a small cross-organizational project. // *Proceedings of the 13th European Software Process Improvement Conference*, Finland, 2006, pp. 5-15. DOI: 10.1007/11908562_2
- [17] Bittner, K.; Spence, I. *Managing iterative software development projects*. Addison-Wesley, USA, 2007.
- [18] Medinilla, A. *Agile Management: Leadership in an Agile Environment*. Springer-Verlag, Berlin, Germany, 2012. DOI: 10.1007/978-3-642-28909-5
- [19] Hughes, R. *Agile Data Warehousing Project Management: Business Intelligence Systems Using Scrum*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2012.
- [20] Salen, K.; Zimmerman, E. *Rules of play: Game design fundamentals*. MIT press, Cambridge, MA, USA, 2003.
- [21] Zichermann, G.; Cunningham, C. *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps*. O'Reilly Media, USA, 2011.
- [22] Deterding, S.; Sicart, M.; Nacke, L.; O'Hara, K.; Dixon, D. Gamification using game-design elements in non-gaming contexts. // *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems*, ACM, 2011, pp. 2425-2428. DOI: 10.1145/1979742.1979575
- [23] Deterding, S.; Khaled, R.; Nacke, L.; Dixon, D. Gamification: Toward a definition. // *Proceedings of the 2011 Annual Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, New York, 2011.
- [24] Huotari, K.; Hamari, J. Gamification from the perspective of service marketing. // *Proceedings of CHI 2011 Gamification Workshop*, 2011.
- [25] Deterding, S.; Dixon, D.; Khaled, R.; Nacke, L. From game design elements to gamefulness: defining gamification. // *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, New

- York, NY, USA, ACM, 2011, pp. 9-15. DOI: 10.1145/2181037.2181040
- [26] Hamari, J.; Eranti, V. Framework for designing and evaluating game achievements. // Proceedings of DiGRA 2011: Think Design Play, 2011.
- [27] Groh, F. Gamification: State of the art definition and utilization. // Institute of Media Informatics Ulm University, 2012, 39.
- [28] Yilmaz, M. A software process engineering approach to understanding software productivity and team personality characteristics: an empirical investigation. // PhD thesis, Dublin City University, 2013.
- [29] Cockburn, A. The end of software engineering and the start of economic- cooperative gaming.// Computer Science and Information Systems (COMSIS), 1(2004), pp. 1-32. DOI: 10.2298/CSIS0401001C
- [30] Baskerville, R. L.; Levine, L.; Ramesh, B.; Pries-Heje, J. The high speed balancing game: How software companies cope with Internet speed. // Scandinavian Journal of Information Systems, 16(2004), pp. 11-54
- [31] Holeman, R. The software process improvement game. In Ibrahim, R., ed. // Software Engineering Education. Volume 895 of Lecture Notes in Computer Science. Springer, 1(1995), pp. 259-261. DOI: 10.1007/3-540-58951-1_109
- [32] Ogland, P. The game of software process improvement: Some reflections on players, strategies and payoff. Norsk konferanse for organisasjoners bruk av informasjons-teknologi (NOKOBIT-16), 1(2009), pp. 209-223.
- [33] Kapp, K.M. The gamification of learning and instruction: game-based methods and strategies for training and education. John Wiley & Sons, New York, NY, 2012.
- [34] Werbach, K.; Hunter, D. For the Win: How Game Thinking Can Revolutionize Your Business. Wharton Digital Press, Pennsylvania, USA, 2012.
- [35] Schell, J. The Art of Game Design: A book of lenses. CRC Press, USA, 2008
- [36] Lazzaro, N. Why we play games: Four keys to more emotion without story. Technical report Oakland, CA: XEO Design Inc. 2004.
- [37] Given, L. The Sage encyclopedia of qualitative research methods. Sage Publications, USA, 2008. DOI: 10.4135/9781412963909
- [38] Tomal, D. Action research for educators. Rowman & Littlefield Pub Incorporated, USA, 2010.
- [39] Nikitina, N.; Kajko-Mattsson, M.; Strale, M. From scrum to Scrumban: A case study of a process transition. // IEEE International Conference on Software and System Process (ICSSP), 2012, 1(2012), pp. 140-149. DOI: 10.1109/icssp.2012.6225959
- [40] Yilmaz, M.; O'Connor, R. Application of Scrumban and Gamification approaches to a small and medium-sized software development organization [in Turkish]. // Workshop on Agile Approaches to Software Development (CEYA'13), 7th Turkish National Software Engineering Conference (UYMS 2013), Izmir, Turkey, 2013.

Authors' addresses**Murat Yilmaz, Assist. Prof. Dr.**

Cankaya University
Department of Computer Engineering
Ankara, Turkey
+903122331358
E-mail: myilmaz@cankaya.edu.tr

Rory O'Connor, Prof. Dr.

School of Computing
Dublin City University
Glasnevin, Dublin 9, Ireland
E-mail: roconnor@computing.dcu.ie