Adopting Collaborative Games into Agile Software Development

Mateusz Zakrzewski[1], Dagmara Kotecka[1], Yen Ying Ng[2],
and Adam Przybyłek[1](✉)

[1] Faculty of Electronics, Telecommunications and Informatics,
Gdańsk University of Technology, Gdańsk, Poland
matzak91@gmail.com, dagkotecka@gmail.com,
adam.przybylek@gmail.com

[2] Department of English Studies, Nicolaus Copernicus University,
Toruń, Poland
nyysang@gmail.com

Abstract. Although the emergence of agile methods has triggered a growing awareness that social factors have a crucial impact on the success of software projects, neither the Scrum Guide nor the Agile Manifesto prescribe techniques that aid the human side of software development. To address this challenge, we enriched the Scrum process with a set of collaborative games. Collaborative games refer to techniques inspired by game play, but designed for the purpose of solving practical problems. Our approach was evaluated in two companies. The feedbacks received from Scrum teams indicate that the implementation of collaborative games leads to a variety of measurable societal outcomes. In particular, the adopted games improved participants' communication, involvement, and creativity, make participants more willing to attend Scrum meetings, and produce better results than the standard approach. This paper is an extended version of our previous work [60].

Keywords: Collaborative games · Agile software development · Scrum

1 Introduction

Traditionally, Requirements Engineering (RE) is the process of identifying right stakeholders and eliciting their needs, documenting these needs as explicit requirements, and then, communicating and validating the requirements [51, 54].

Przybyłek [55] enumerated a number of inherent difficulties in the requirements engineering process. Such difficulties, despite being well known, are still encountered in present industrial practice [33]. Customers rarely know what they really need [20] and usually they have only a vague picture of their needs at the beginning of the project [8, 41]. In addition, their needs may be difficult to articulate [14]. Moreover, stakeholders may be numerous and distributed [70]. Their needs may vary and conflict, depending on their perspectives of the environment in which they work and the tasks they wish to accomplish [51]. Furthermore, effective communication among stakeholders may be difficult as a consequence of their different vocabularies and

E. Damiani et al. (Eds.): ENASE 2018, CCIS 1023, pp. 119–136, 2019.
https://doi.org/10.1007/978-3-030-22559-9_6

professional backgrounds [7, 68]. What is more, the ways requirements are documented and communicated may be chosen inappropriately with respect to stakeholders' profiles [34]. Finally, requirements evolve during the project due to exploration in the problem space, and dynamics of a business environment formed and reformed by the interactions of the stakeholders [27, 62]. As a response to some of these problems, agile methods were proposed and over the years have become dominant in the software industry.

In Agile software development, requirements engineering activities are present over the whole life cycle of a project. Thereby, the role of stakeholders is expanded within the entire development process by involving them in writing user stories, discussing product features, prioritizing the product backlog, and providing feedback to the development team on a regular basis [26, 49]. This requires that the customers work with developers as active team members [35, 58, 71]. The idea of having a customer as a member of a development team has grown from a single on-site customer, which was dismissed by Kent Beck himself as "an error of early XP thinking" [12], to a customer team "equal to or larger in size than the programming team" [46]. Since there is a wide range of potential customers, it would be difficult for a single person to represent them all [4]. The representation of stakeholders may be also achieved with the role of a product manager or an entire product management team [73]. Moreover, in agile software development stakeholders are expected to be collaborative [6]. Unfortunately, agile methods do not provide techniques to promote these attitudes. Therefore, inadequate stakeholder participation, inability to obtain consensus among various stakeholders and lack of effective knowledge sharing are still challenges confronting agile RE [8, 11, 13, 26, 49, 53, 61].

In the meantime, many researchers and practitioners have acknowledged and agreed on the importance and the role of creativity in RE [27, 45]. As a result, a substantial body of knowledge has been established, which can be summarized as follows. Requirements are no longer considered to exist in an implicit manner in the mind of stakeholders [39], while the stakeholders are no longer viewed as a passive source of requirements information but rather as active participants in requirements engineering process [50, 58]. Active participation means forward thinking, creating new visions, suggesting IT innovations, and shaping solutions [64]. Thus, finding the "right" requirements is not only about capturing requirements, but is also about helping stakeholders to discover requirements they were not aware of, and solving problems they did not know they had [31]. According to Robertson [64], requirements analysts should invent requirements based on their understanding of the organisation's competitive business goals and context. Such requirements are not often things that requirements analysts directly asked for [44]. Furthermore, Mahaux et al. [42] and Svensson and Taghavianfar [67] suggest that RE is not simply a creative process, but a collaboratively creative process, where interdisciplinary group of stakeholders work together to create ideas, solve conflicts, and reach a consensus on a novel and valuable system they want to build [60]. Thus, traditional requirements gathering techniques such as interviews, questionnaires, focus groups, participant observation, or document analysis [52] are insufficient to elicit the whole range of requirements [14].

Unfortunately, agile methods do not provide new requirements gathering techniques nor they explicitly support creativity [60]. Even though Highsmith and

Cockburn [25] mention that "creativity, not voluminous written rules, is the only way to manage complex software development problems and diverse situations", agile methods make little reference to established creativity techniques [30].

Responding to the above-mentioned challenges, we propose to equip agile teams with a set of collaborative games. A game is collaborative if two or more players must work together to achieve its goal, which is to solve a practical problem [21]. Collaborative games are designed to leverage multiple dimensions of communication that let participants engage the full power of their brains, resulting in richer, deeper, and more meaningful exchanges of information [28]. At the same time, they emphasize the concepts of teamwork and collaboration which are highly valued by agile practices [32]. They can also bring numerous benefits to the requirement elicitation process since they typically provide immediate feedback, activate participants and increase participant's motivation [19, 63].

In our study, we selected 8 games and implemented them in commercial projects. Based on the received feedback, we proposed a framework that specifies how to integrate a set of collaborative games into the Scrum process. There are number of methods and process (i.e. Scrum, Kanban, Scrumban, Spotify model etc.) used in software industry [1, 2, 72]. However, we chose Scrum, since it is one of the most widely adopted in industry [65, 72].

2 Research Method

Our study was conducted as Action Research [5]. Action Research is a partnership of the researchers with the study participants who use an iterative process to initiate improvement and investigate it. The researchers bring their knowledge of action research while the participants bring their practical knowledge and context [5]. Action Research simultaneously assists in practical problem solving, increases participants competencies, and improves scientific knowledge [15, 16]. A precondition for action research is to have a problem owner willing to collaborate to both identify a problem, and engage in an effort to solve it [18]. We conducted two Action Research projects. The problem owner in the first study was an internal software development department of the world's recognized leading food processing company with 150 years of tradition (the company wishes to remain anonymous). The department was experiencing typical challenges faced by Agile teams, such as the inability to gain access to the customer and the lack of customer involvement. The second problem owner was Intel Technology Poland. The company was interested in revising its work practices related to Sprint Retrospective. Authorities of both companies were open to new ideas and willing to deploy our framework in practice.

3 Adapted Games

3.1 Cover Story

In Cover Story [24, 29], customers imagine an ideal future system so spectacular that it gets published on the front page of a newspaper. The customers must pretend as though this future has already taken place. The game encourages people to ignore all limits and "think big". As a result, it uncovers shared goals and can lead to realizing true possibilities that were once unimaginable. To play the game, the customers are divided into teams of four to six and each team is given a template (see [60]) that include six components:

- Cover – states the spectacular success of the software system;
- Headlines – reveal what the cover story is about;
- Sidebars – reveal interesting facets of the cover story;
- Quotes – testimonials about the accomplishment;
- Brainstorm – is used for documenting initial ideas;
- Images – pictures that support the cover story.

After taking 5 min for individuals to silently think over the system, the team should collaborate to fill in each component. Next, each team presents their chart.

3.2 Whole Product

Originally, the game aims to help the team discover new ideas about what can be done to make the product distinct and find ways to gain more customers [40]. However, it can be also useful for prioritizing a product backlog or for defining a product roadmap. The game board comprises four concentric circles that represent different aspects of the product [29]:

- Inner Circle: Generic Product – the fundamental features that define the product;
- Circle 2: Expected Product – the features that customer considers absolutely essential;
- Circle 3: Augmented Product – the features that go beyond customer expectations;
- Outer Circle: Potential Product – everything that might be done to attract and hold customers.

Participants write ideas on sticky notes related to each circle, and then post the ideas on the chart. After all of the ideas are posted, the significance of the resulting chart is discussed. This allows developers to understand what the customers truly want from the product.

3.3 AVAX Storming

AVAX Storming [69] is based on brainstorming. Its aim is to identify the desired functional requirements for the system. The participants write down each functional requirement on a single sticky note and place it on a flipboard. This practice helps customers to figure out the size of their project because soon the flipboard starts to be filled up. There are two note colors. One for “needed” requirements and the other for “desired” requirements. When all notes are posted on the flipboard, each requirement is explained in detail by the author and discussed by the team. Overlapped requirements are merged. Later, the notes are grouped in order to sketch the system modules. The final result is a mind map demonstrating the size of the project.

3.4 Buy a Feature

Buy-a-Feature [28] is a way of choosing the right set of features to be developed in the next Sprint. In this game, customers collaborate to purchase their most desired features. Strictly speaking, they jointly prioritize their desires as a group. Each feature should include a meaningful label, a short description, and an enumeration of benefits. Features are also assigned a price depending on their development costs and a number according to their position in the product backlog. Customers buy features that they want in the subsequent Sprint using game money. Some features may be priced so high that no single player can buy them individually. This motivates negotiations among players because they have to pool their money to buy the feature [28]. Listening to the negotiations improves the understanding of what the customers really need. The total amount of money for all players involved in the exercise should allow them to purchase as many features as the developers are able to implement within a sprint.

3.5 Agile Game Incubator

This game [29] allows participants to teach each other the tangle of factors involved in certain dilemmas while gaining a deeper understanding of the predicament themselves. Its goal is to create a way to explain complex problems so that others will genuinely understand it and be able to form solutions. The game board consists of 5 sections, representing the 5 steps of the game-creation strategy, which conveniently form the acronym PLAID (pronounced “played”). There are also colorful sticky notes that symbolize the ideas for each section:

- Problem – what you want to solve (red notes);
- Lead Objectives – what you hope to gain from solving the problem (green notes);
- Aspects – the different parts of the problem (purple notes);
- Invent – the game created to solve the problem (blue notes);
- Debrief – how the game worked out (yellow notes).

The team should brainstorms ideas related to each of the 5 steps, write them on sticky notes, and then post on the board in the respective sections.

3.6 How-Now-Wow Matrix

When people want to develop new ideas, they most often think out of the box in the creative idea generation phase. However, when it comes to convergence, people often end up picking ideas that are most familiar to them (tastycupcakes.org). The How-Now-Wow Matrix game [60] helps stakeholders select features that make the product unique and distinguish it from the competition. It naturally follows the brainstorming session, where the features that were initially flushed out are now discussed. The features are listed down on a large poster. The game board is a 2 × 2 matrix with

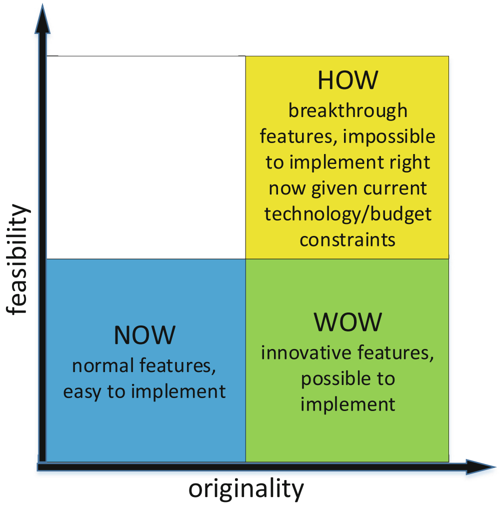
HOW
breakthrough features, impossible to implement right now given current technology/budget constraints
NOW
normal features, easy to implement
WOW
innovative features, possible to implement
feasibility
originality

Fig. 1. How-Now-Wow Matrix [60]. (Color figure online)

"originality" on the x-axis and "feasibility" on the y-axis as shown in Fig. 1. Each player is given 9 colored dot stickers (3 yellow, 3 blue, 3 green) that correspond to the quadrants of the matrix. Then, players place the respective stickers next to the three ideas that they believe are best for each category. After all the dots have been used, the number of dots under each idea are counted. The highest number of dots of a certain color categorizes the idea under that color.

3.7 Speedboat

Speedboat [58] explicitly asks customers to say what they do not like about the product. Nonetheless, it lets the facilitator stay in control of how the complaints are stated. The game starts by drawing a boat. The boat represents the software system. Everyone wants the boat to move fast. Unfortunately, the boat has a few anchors holding it back. Customers write what they don't like on an index card and place it under the boat as an anchor. The lower an anchor is placed, the more significant the issue is. Although most customers have complaints, some of them do not feel comfortable expressing their frustrations verbally, while others complain a lot about the little details. Speedboat creates a relatively safe environment where customers can say what is wrong. By asking people to verbalize their issues in writing, the game motivates them to reflect on what is genuinely most troublesome. In this way, many of them will self-identify trivial issues as just that – trivial issues. When customers are finished posting their anchors, the facilitator reviews each one, carefully confirming the understanding of what they want to see changed in the system.

3.8 Prune the Product Tree

Prune the Product Tree [28, 60] helps to develop a balanced product roadmap by looking at the set of features that compose the product in a holistic manner. In this game, customers collaborate to shape the evolution of the product (i.e., the system to be developed). The product is represented by a large tree on a whiteboard. Branches correspond to major areas of functionality within the software system, while leaves correspond to features. The differently colored canopies stand for various product releases. The oldest features should therefore go near the trunk. Players write a short description for each new feature on an index card, ideally shaped as leaves, and places the card on the tree. This short description generally represents a valued functionality that satisfies customers' needs. Features to add in the next Sprint are attached in the area near the edge considered as the current release. Leaves at the outer edge of the canopy are considered longer term. Participants may group leaves or draw lines between leaves to clarify relationships among features. They may also "prune" features that are not working for them by taking them off the tree.

3.9 Sailboat

The Sailboat game [23, 56, 57] is quite similar to the Speedboat game but instead of engines, there are sails and wind. In addition, there are rocks which represent the risks the team might encounter along the way, an island which represents the team's objectives. Then, the attendances write down their ideas on post-it notes and put them into the different areas according to the picture. After that, they discuss all the categories on the picture.

3.10 Mad/Sad/Glad

Mad/Sad/Glad [17] uncovers the emotional content of the past iteration and helps teams to look for things that:

- drive them mad;
- make them feel sad;
- they are glad of.

The game starts with creating poster divided into three areas labeled Mad, Sad, and Glad. Next, attendees should write one issue per sticky note and put the sticky notes on the corresponding area. Then, the team groups related sticky notes into logical themes. In the end, each theme is discussed, a consensus is found, and corrective actions are proposed [57].

3.11 Starfish

The Starfish game [23, 57] is a technique that helps people to reflect on different degree of things and helps team members to understand each other. Instead of the typical three questions i.e. what worked well, what did not work, and what did we learn, the game board comprises a circle divided into five equal areas:

- Stop Doing – something that has not brought value, or even worse, which hindered the progress;
- Less Of – something that has been done and with added value, but the effort required is bigger than the benefit;
- Keep Doing – something that the team is doing well and wants to continue;
- More Of – something that is useful but could be exploited even more to bring more value;
- Start Doing – something new that the team wants to bring to the game.

To play the game, participants write their ideas on post-it notes, and then proceed in a manner analogous to that for Mad/Sad/Glad.

3.12 5Ls

The 5Ls game [57] handles both the positive and negative aspects of the past iteration but also brings forth the continuous development. Before the game starts, the moderator divides the poster into five categories:

- Liked – what did the team members really like about the iteration?
- Learned – what new things did the team members learn during the iteration?
- Lacked – what things could the team members have done better in the iteration?
- Longed For – what things did the team members wish for but were not present during the iteration?
- Loathed – what things did the team members dislike in the iteration?

Again, the next steps are analogous to those of Mad/Sad/Glad.

4 Proposed Framework

Figure 2 shows the typical Scrum life cycle with collaborative games superimposed. There are three meetings where collaborative games may occur: Product Planning, Sprint Planning, and Sprint Review.

The purpose of Product Planning is to establish the vision of what customers wish to build and accordingly the initial Product Backlog. Three games that can support this phase are Cover Story, AVAX Storming, and Whole Product. Cover Story enables Scrum teams to understand (1) the customer's vision of the system to be developed, (2) the customers' imagination of success, and (3) how the system will create business value.

In turn, Whole Product discovers features at a very high level and categorizes them into four main categories. Features belonging to the two first categories should be implemented first. Lastly, AVAX Storming identifies functional requirements and categorizes each as either needed or desired.

Before the start of each Sprint two consecutive meetings are held. In the first, stakeholders meet to refine and re-prioritize the Product, and to choose goals for the next iteration, usually driven by highest business value [38]. In the second meeting, the team and Product Owner meet to consider how to achieve the goals, and to create a

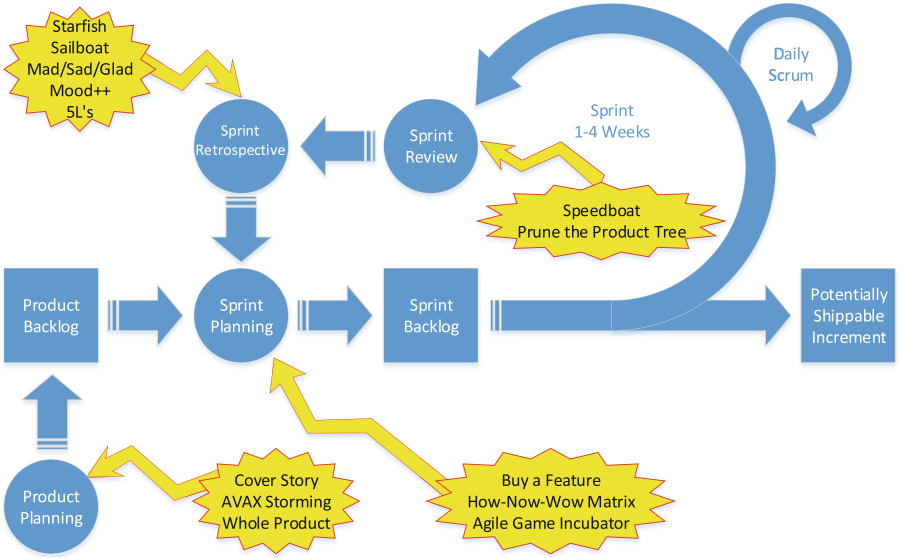

Fig. 2. Scrum life cycle with collaborative games superimposed [60].

Sprint Backlog. The team asks enough questions that they can break down user stories of the product backlog into the more detailed tasks of the sprint backlog.

The essential game to prioritize the Product Backlog enough for the next Sprint is Buy-a-Feature. It identifies the customer's highest-priority features that can be completed within the Sprint period. The game also helps several customer representatives reach a consensus if they have conflicting interests. Likewise, How-Now-Wow Matrix aims at selecting the most valuable features as a group. On the other hand, Agile Game Incubator let the Scrum Team understand complex and unclear requirements.

The Sprint Review is held to inspect the Increment and to adapt the Product Backlog if needed. Typically, after the team demonstrates new features to the Product Owner or to the business stakeholders, all attendees collaborate on what could be done to deliver more business value to the customer. Two games – Speedboat and Prune the Product Tree – may be deploy to elicit the feedback and foster collaboration. Both games give the team the opportunity to identify those features that are simply not meeting customer needs. Speedboat solely focuses on features that need to be addressed, while Prune the Product Tree additionally provides customers with a way to indicate the directions in which to evolve the system. By observing how customers shape the tree's growth, the team has the opportunity to refine the requirements to ensure they maintain cohesion with the business.

At the end of each sprint, the team conducts a retrospective to look back at events that already have taken place, discuss what went right and wrong and decide how to improve these items for the next sprint [57]. We adopted the following games to facilitate this meeting: Sailboat, Starfish, Mad/Sad/Glad, Mood++, 5L's.

5 Evaluation and Results

5.1 Food Processing Company

The evaluation was made on two projects carried out by the same Scrum team, but for different customers. The customers were other departments within the company. The projects were about developing Workflow Management System. Typically, 8 people attended each game session. Among them there were 3 customers, product owner, scrum master and 3 developers.

After each game session, we issued a questionnaire. The attendances were asked to indicate their level of agreement with statements about game-playing activity. The responses were on a Likert scale of: 1 – Strongly disagree; 2 – Disagree; 3 – Neutral;

Table 1. Summary of questionnaire responses (food processing company).

Rating scale: 1 – Strongly disagree 2 – Disagree 3 – Neutral 4 – Agree 5 – Strongly agree	Cover Story	Agile Game Incubator	AVAX Storming	Buy a Feature	How-Now-Wow Matrix	Speedboat	Prune the Product Tree	Whole Product
The game produces better results than the standard approach	3.5	3.9	4.2	4.0	3.5	4.3	3.9	3.6
The game makes participants more willing to attend the meeting	4.3	4.0	4.0	4.0	4.0	4.2	4.0	4.0
The game fosters participants' creativity	2.7	2.9	3.4	3.4	3.0	3.5	2.9	2.4
The game fosters participants' commitment	3.7	3.7	4.2	4.3	4.0	4.4	3.9	3.7
The game is easy to understand	4.0	3.6	4.4	4.4	3.9	4.5	4.1	4.0
All facets together	3.6	3.6	4.0	4.0	3.7	4.2	3.8	3.5

4 – Agree; 5 – Strongly Agree. Table 1 presents average values for each game and statement across both projects and all attendances. The corresponding standard deviation was always less than 1.

At the end of the survey, the attendances were also invited to specify any additional remarks. Several of them reported a high level of enjoyment when using the games, while those who represented the customer side reported that the games were useful and motivated them to contribute to requirements gathering.

Generally all games were evaluated positively, because they achieved the average score between 3.5 and 4.2. The only issue that was not appreciated was the impact on creativity, since four games obtained score below the baseline (neutral). This can be explained by the fact that both projects were designed for internal customers, so the business needs were well known and the requirements gathering process did not require much creative thinking. In addition, the implemented software was a standard Workflow Management System and was not expected to provide any innovative features. On the contrary, willingness to attend the meeting was significantly stimulated by every game.

Whole Product, Cover Story and Agile Game Incubator performed the worst, but still above the baseline. Again, the internal customer factor probably prevented Cover Story to demonstrate its full power. In turn, Agile Game Incubator was the most difficult to understand. Indeed, it required the attendances to create their own game to communicate a complex problem. How-Now-Wow Matrix also performed below expectations, probably due to a lack of innovative features in the system.

Prune the Product Tree was considered a bit childish and its output was not perceived as meaningful even though it obtained quite high scores in all facets except creativity. On the other hand, 3 top rated games were Speedboat, AVAX Storming, and Buy-a-Feature. Each of them generated very tangible output that was considered valuable by the attendances.

Note, that some games are substitutes for others, e.g. Speedboat is a substitute for Prune the Product Tree. The attendances preferred AVAX Storming over Whole Product, Buy-a-Feature over How-Now-Wow Matrix, and Speedboat over Prune the Product Tree.

5.2 Intel Technology Poland

The evaluation (for details look at [57]) was conducted by 3 teams, which are characterized in Table 3. We collaborated with the Scrum Master to properly implement the games into the teams. After evaluation in the first company, we reflected that our question set needed to be refined, since it did not captured all essential aspects of game sessions. The new question set is presented in Table 2. For each question, we first took the average per retrospective session, then based on these averages we took the average per team, and finally per game. All games except Mad/Sad/Glad were evaluated positively with respect to all categories. Even if they hardly scored above 3 for one category, they scored around 4 for other categories.

The Sailboat Game. Although the attendances agreed that Sailboat produces better results than the standard approach, they believed it should not be used too often due to

Table 2. Summary of questionnaire responses (Intel Technology Poland).

Rating scale: 1 – Strongly disagree 2 – Disagree 3 – Neutral 4 – Agree 5 – Strongly agree	Sailboat	Starfish	Mad/Sad/Glad	Mood++	5L's
The game produces better results than the standard approach	4.0	4.0	3.0	3.7	4.0
The game should be implemented permanently instead of the standard approach	3.0	4.3	3.2	3.4	4.0
The game may be considered as complementary to the standard approach	3.5	3.6	4.2	4.0	4.2
The game fosters participants' creativity	3.3	3.7	3.7	3.9	4.0
The game fosters participants' involvement	3.8	4.1	3.2	3.9	4.4
The game improves participants' communication	3.5	3.1	2.3	3.7	4.2
The game is easy to understand and play	3.2	4.0	4.2	4.1	4.0
All facets together	3.5	3.8	3.4	3.8	4.1

Table 3. Participating teams.

Team	Description
T1, 9 people	The team had worked on the project for 18 months, when we started our research. Team members had typically 2 years of Scrum development experience
T2, 3 people	The team had just joined a new project, but all team members had over 3 years of experience with Scrum
T3, 8 people	The team had worked on the project for 7 months. All team members had over 3 years of experience with Scrum

three reasons. Firstly, it would be boring to consider the vision and risks every time because they hardly change over the life cycle of a project. Secondly, using a sailboat as a metaphor was too abstract for some attendances, so the game was quite difficult to play. Finally, the attendances claimed that the discussion was not effective on improving the teamwork and the process. In contrast, they were satisfied that the game fostered their participation and built a friendly environment where they were able to express and discuss their frustrations [57].

The Starfish Game. Starfish functioned well in all categories but "communication among team members", which was not affected. Since the game covers all topics of traditional retrospective and fosters attendances' involvement at the same time, the attendances favored the substitution of the game for the standard approach. They also recognized that the game was effective in helping them to understand each other perceived value on the way they worked [57].

Mad/Sad/Glad and Mood++. Mad/Sad/Glad was perceived as the easiest to understand and play, even though it performed the worst overall. In particular, its influence on communication between team members has received negative rate. The cause was likely that the game is too simple and does not cover all topics that are usually discussed during a retrospective. Nonetheless, after improving the game with two new categories, the communication aspect was substantially enhanced, while the new version performed as well as the Starfish game [57].

The 5L's Game. Overall, 5L's received high rate in each category and surpassed all other games. When compared to Starfish and Mood++, it also covers all aspects of the Retrospective, but was perceived as excellent especially in improving attendances' communication. The other strength of the game is that the attendances' involvement was boosted [57].

6 Related Work

Gelperin [21] defined six collaborative games that support requirements understanding by improving communication and cooperation between customers and developers. He also defined a mapping system to help developers choose the best game to play in any situation. His games could be used complementary to the games used in our framework during Sprint Planning.

Trujillo et al. [69] proposed a game-based workshop (ActiveAction) used as an alternative for the software project's Inception phase. ActiveAction combines classical and game-based techniques to foster stakeholders' involvement and a collaborative identification of objectives, constraints and risks.

Ghanbari et al. [22] proposed a new approach for gathering requirements from distributed software stakeholders. Their approach employs two collaborative games (Prune the Product Tree and Buy-a-Feature) provided by a web-based tool designed by Hohmann [29].

Besides, considerable research has been directed at adopting collaborative games to support agile developers. Przybyłek and Olszewski [56] proposed an extension to Open Kanban, which contains 12 collaborative games that help inexperienced teams better understand the principles of Kanban. Ahmad et al. [3] highlighted that collaborative learning and Kanban board play important role in software engineering students learning and professional skills gaining. In turn, Derby and Larsen [17], Gonçalves and Linders [23], Caroli and Caetano [9], and Krivitsky [37] presented collaborative games that can be used to facilitate the Sprint Retrospective. In this study, we evaluated some of these games.

On the other hand, numerous creativity fostering techniques have been proposed to improve the quality of requirements deliverables and to increase customer satisfaction with the final product. The most popular ones are probably brainstorming and Joint Application Development [10]. More recently, Maiden et al. [44] proposed RESCUE, a scenario-driven requirements engineering process that includes workshops that integrate creativity techniques with different types of use case and system context modeling. The process was successful applied to encourage creative thinking about requirements for an air traffic control system [43].

Mich et al. [47] developed and evaluated EPMcreate – a creativity enhancement technique that is based on the Elementary Pragmatic Model. EPMcreate can be applied in any situation in which ideas need to be generated, e.g., at any time one might apply brainstorming [48]. The feasibility of applying EPMcreate to idea generation in requirements gathering was established by two experiments. EPMcreate demonstrated to be very effective in finding requirements that had not been known to the managers of the projects involved. Moreover, EPMcreate proved to generate more ideas and, in particular, more useful ideas than the familiar brainstorming [47]. Furthermore, Mich et al. [48] showed that EPMcreate is also effective when used by individuals.

Sakhnini et al. [66] proposed POEPMcreate, which is an optimization of EPMcreate that requires fewer steps than EPMcreate. The effectiveness of POEPMcreate was demonstrated in two controlled experiments by comparing it to both brainstorming and EPMcreate. The results indicate that POEPMcreate is more effective, by the quantity and quality of the ideas generated, than EPMcreate, which is, in turn, more effective than brainstorming.

Karlsen et al. [36] integrated ART-SCENE, a tool designed to discover more complete requirements with scenarios, with combinFormation, a tool that supports people in creating new ideas while finding and collecting information. As pointed out by the authors [36], their approach was designed to support individual creativity.

Hollis and Maiden [30] extended Ambler's agile process with three creativity techniques: brainstorming, Partners in Creative Learning, and a new technique inspired by Hall-of-Fame. The evaluation shows that requirements generated from the extended process were rated more novel then requirements in the original product backlog.

Svensson and Taghavianfar [67] evaluated four different creativity techniques, namely Hall of Fame, Constraint Removal, Brainstorming, and Idea Box, using creativity workshops. The creativity workshops followed the structure and the design of the creativity workshops in RESCUE [44]. The results indicate that Brainstorming can generate by far the most ideas, while Hall of Fame generates most creative ideas. Idea Box generates the least number of ideas, and the least number of creative ideas. Finally, Hall of Fame is the technique that generates the most practical ideas [67] companies were open to new ideas and willing to deploy our framework in practice.

7 Summary

This paper reports on two Action Research projects carried out in two different companies to explore the ways in which collaborative games could benefit Scrum teams. Collaborative games represent a powerful tool in improving interactions among the development team and between the team and the customer. Their social and entertaining aspects provide an alternative to traditional Scrum meetings. The received feedbacks obtained through questionnaires show that the adopted games improved participants' communication, involvement, and creativity; make participants more willing to attend Scrum meetings; and produce better results than the standard approach. Moreover, the participated teams intended to continue playing the proposed games after the project finished. We also learnt that the issues that teams deals with may be different in each Sprint and project, so the Scrum Master should have a set of possible games to be able to pick the most effective one depending on the situation at hand.

As future work, we will continue implementing collaborative games in other companies. Moreover, we will try to integrate collaborative games with other agile methods. We also plan to add more games to saturate each Scrum meeting in a balanced way. Furthermore, we want to develop a digital version of some games using the framework presented by Przybyłek and Kowalski [58]. Finally, we want to study the effect of collaborative games on social aspects of software development in a controlled experiment with settings similar to [59].

References

1. Ahmad, M.O., Dennehy, D., Conboy, K., Oivo, M.: Kanban in software engineering: a systematic mapping study. J. Syst. Softw. **137**, 96–113 (2018)
2. Ahmad, M.O., Kuvaja, P., Oivo, M., Markkula, J.: Transition of software maintenance teams from Scrum to Kanban. In: 49th Hawaii International Conference on System Sciences (HICSS 2016), Koloa, HI (2016)
3. Ahmad, M.O., Liukkunen, K., Markkula, J.: Student perceptions and attitudes towards the software factory as a learning environment. In: Global Engineering Education Conference (EDUCON 2014), Istanbul, Turkey (2014)
4. Ambler, S.W.: Scaling on-site customer. Dr. Dobbs Journal, 63–66, January 2008
5. Baskerville, R., Myers, M.D.: Special issue on action research in information systems: making IS research relevant to practice—foreward. MIS Q. **28**(3), 329–335 (2004)
6. Boehm, B., Turner, R.: Balancing Agility and Discipline: A Guide for the Perplexed. Addison-Wesley, Boston (2004)
7. Bormane, L., Gržibovska, J., Bērziša, S., Grabis, J.: Impact of requirements gathering processes on success of information system development projects. Inf. Technol. Manag. Sci. **19**(1), 57–64 (2016)
8. Cao, L., Ramesh, B.: Agile requirements engineering practices: an empirical study. IEEE Softw. **25**(1), 60–67 (2008)
9. Caroli, P., Caetano, T.: Fun Retrospectives - Activities and Ideas for Making Agile Retrospectives More Engaging. Leanpub, Layton (2016)
10. Carmel, E., Whitaker, R., George, J.: PD and joint application design: a transatlantic comparison. Commun. ACM **36**(4), 40–48 (1993)
11. Chan, F.K.Y., Thong, J.Y.L.: Acceptance of agile methods: a critical review and conceptual framework. Decis. Support Syst. **46**(4), 803–814 (2009)
12. Conboy, K., Wang, X., Fitzgerald, B.: Creativity in agile systems development: a literature review. In: Dhillon, G., Stahl, B.C., Baskerville, R. (eds.) Information Systems – Creativity and Innovation in Small and Medium-Sized Enterprises. IFIP Advances in Information and Communication Technology, vol. 301, pp. 122–134. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02388-0_9

13. Conboy, K., Coyle, S., Wang, X., Pikkarainen, M.: People over process: key people challenges in agile development. IEEE Softw. **99**, 47–57 (2010)
14. Davis, C.J., Fuller, R.M., Tremblay, M.C., Berndt, D.J.: Communication challenges in requirements gathering and the use of the repertory GRID technique. J. Comput. Inf. Syst. **47**, 78–86 (2006)
15. Davison, R.M., Martinsons, M.G., Kock, N.: Principles of canonical action research. Inf. Syst. J. **14**(1), 65–86 (2004)
16. Dawson, C.: Practical Research Methods: A User-Friendly Guide to Mastering Research Techniques and Projects. How To Books Ltd., Oxford (2002)
17. Derby, E., Larsen, D.: Agile Retrospectives: Making Good Teams Great. Pragmatic Programmers (2006)
18. Easterbrook, S.M., Singer, J., Storey, M.A., Damian, D.: Selecting empirical methods for software engineering research. In: Shull, F., Singer, J., Sjøberg, D.I.K. (eds.) Guide to Advanced Empirical Software Engineering, pp. 285–311. Springer, London (2006). https://doi.org/10.1007/978-1-84800-044-5_11
19. Fernandes, J., Duarte, D., Ribeiro, C., Farinha, C., Pereira, J., da Silva, M.M.: iThink: a game-based approach towards improving collaboration and participation in requirement elicitation. Procedia Comput. Sci. **15**, 66–77 (2012)
20. Faulk, S.: Software requirements: a tutorial. In: Thayer, R., Dorfman, M. (eds.) Software Requirements Engineering. IEEE Computer Society Press, Washington (1997)
21. Gelperin, D.: Increase requirements understanding by playing cooperative games. In: INCOSE International Symposium, Denver, CO (2011)
22. Ghanbari, H., Similä, J., Markkula, J.: Utilizing online serious games to facilitate distributed requirements gathering. J. Syst. Softw. **109**, 32–49 (2015)
23. Gonçalves, L., Linders, B.: Getting Value Out of Agile Retrospectives: A Toolbox of Retrospective Exercises. Leanpub, Layton (2014)
24. Gray, D., Brown, S., Macanufo, J.: Gamestorming: A Playbook for Innovators, Rulebreakers, and Changemakers. O'Reilly Media, Sebastopol (2010)
25. Highsmith, J., Cockburn, A.: Agile software development: the business of innovation. IEEE Comput. **34**(9), 120–122 (2001)
26. Hoda, R., Noble, J., Marshall, S.: The impact of inadequate customer collaboration on self-organizing agile teams. Inf. Softw. Technol. **53**, 521–534 (2011)
27. Hoffmann, O., Cropley, D., Cropley, A., Nguyen, L., Swatman, P.: Creativity, requirements and perspectives. Aust. J. Inf. Syst. **13**(1), 159–175 (2005)
28. Hohmann, L.: Innovation Games: Creating Breakthrough Products Through Collaborative Play. Addison-Wesley Professional, Boston (2006)
29. Hohmann, L.: Innovation Games Website (2017). www.innovationgames.com
30. Hollis, B., Maiden, N.: Extending agile processes with creativity techniques. IEEE Softw. **30**(5), 78–84 (2013)
31. Horkoff, J., Maiden, N.: Creativity and conceptual modeling for requirements engineering. In: 5th International Workshop on Creativity in Requirements Engineering, Essen, Germany (2015)
32. International Institute of Business Analysis (IIBA): Agile Extension to the BABOK®Guide. Toronto, Canada (2011)
33. Jarzębowicz, A., Marciniak, P.: A survey on identifying and addressing business analysis problems. Found. Comput. Decis. Sci. **42**(4), 315–337 (2017)
34. Jarzębowicz, A., Połocka, K.: Selecting requirements documentation techniques for software projects: a survey study. In: 1st International Conference on Lean and Agile Software Development, pp. 1189–1198 (2017). http://dx.doi.org/10.15439/2017F387

35. Jarzębowicz, A., Ślesiński, W.: Assessing effectiveness of recommendations to requirements-related problems through interviews with experts. In: 2018 Federated Conference on Computer Science and Information Systems (FedCSIS 2018), Poznan, Poland (2018). http://dx.doi.org/10.15439/2018F85
36. Karlsen, K., Maiden, N.A.M., Kerne, A.: Inventing requirements with creativity support tools. In: 15th International Working Conference, REFSQ 2009, Amsterdam, The Netherlands (2009)
37. Krivitsky, A.: Agile Retrospective Kickstarter. Leanpub, Layton (2015)
38. Larman, C.: Agile and Iterative Development: A Manager's Guide. Addison Wesley, Boston (2003)
39. Lemos, J., Alves, C., Duboc, L., Rodrigues, G.: A systematic mapping study on creativity in requirements engineering. In: 27th ACM SAC - Requirements Engineering Track, Riva Del Garda, Italy (2012)
40. Levitt, T.: Marketing success through differentiation – of anything. Harvard Bus. Rev. 20–28 (1980) . http://www.confianzys.com/Marketing%20Sucess%20-%20Differentiation%20of%20anything.PDF. The January-February 1980 Issue
41. Maciaszek, L.: Requirements Analysis and Systems Design. Addison-Wesley, Boston (2005)
42. Mahaux, M., Nguyen, L., Gotel, O., Mich, L., Mavin, A., Schmid, K.: Collaborative creativity in requirements engineering: analysis and practical advice. In: 7th IEEE International Conference on Research Challenges in Information Science (RCIS), Paris, France (2013)
43. Maiden, N., Gizikis, A., Robertson, S.: Provoking creativity: imagine what your requirements could be like. IEEE Softw. **21**(5), 68–75 (2004)
44. Maiden, N., Manning, S., Robertson, S., Greenwood, J.: Integrating creativity workshops into structured requirements processes. In: 5th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques, Cambridge, MA (2004)
45. Maiden, N., Jones, S., Karlsen, I.K., Neill, R., Zachos, K., Milne, A.: Requirements engineering as creative problem solving: a research agenda for idea finding. In: 18th IEEE International Conference on Requirements Engineering, Sydney, Australia (2010)
46. McBreen, P.: Questioning Extreme Programming. Addison-Wesley, Boston (2003)
47. Mich, L., Anesi, C., Berry, D.M.: Applying a pragmatics-based creativity-fostering technique to requirements gathering. Requir. Eng. **10**(4), 262–275 (2005)
48. Mich, L., Berry, D.M., Alzetta, A.: Individual and end-user application of the EPMcreate creativity enhancement technique to website requirements gathering. In: Workshop on creativity in requirements engineering at REFSQ 2010, Essen, Germany (2010)
49. Nerur, S., Mahapatra, R., Mangalaraj, G.: Challenges of migrating to agile methods. Commun. ACM **48**, 72–78 (2005)
50. Nguyen, L., Cybulski, J.: Into the future: inspiring and stimulating users' creativity. In: 12th Pacific Asia Conference on Information Systems, Suzhou, China (2008)
51. Nuseibeh, B., Easterbrook, S.: Requirements engineering: a roadmap. In: Conference on the Future of Software Engineering, Limerick, Ireland (2000)
52. Ossowska, K., Szewc, L., Weichbroth, P., Garnik, I., Sikorski, M.: Exploring an ontological approach for user requirements elicitation in the design of online virtual agents. In: Wrycza, S. (ed.) SIGSAND/PLAIS 2016. LNBIP, vol. 264, pp. 40–55. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46642-2_3
53. Owoc, M., Weichbroth, P., Żuralski, K.: Towards better understanding of context-aware knowledge transformation. In: 2017 Federated Conference on Computer Science and Information Systems (FedCSIS 2017), Prague, Czech Republic (2017). http://dx.doi.org/10.15439/2017F383

54. Przybyłek, A.: The integration of functional decomposition with UML notation in business process modelling. Adv. Inf. Syst. Dev. **1**, 85–99 (2007)
55. Przybyłek, A.: A business-oriented approach to requirements gathering. In: 9th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2014), Lisbon (2014)
56. Przybyłek, A., Olszewski, M.: Adopting collaborative games into Open Kanban. In: 2016 Federated Conference on Computer Science and Information Systems (FedCSIS 2016), Gdansk, Poland (2016). http://dx.doi.org/10.15439/2016F509
57. Przybyłek, A., Kotecka, D.: Making agile retrospectives more awesome. In: 2017 Federated Conference on Computer Science and Information Systems (FedCSIS 2017), Prague, Czech Republic (2017). http://dx.doi.org/10.15439/2017F423
58. Przybyłek, A., Kowalski, W.: Utilizing online collaborative games to facilitate agile software development. In: 2018 Federated Conference on Computer Science and Information Systems (FedCSIS 2018), Poznan, Poland (2018). http://dx.doi.org/10.15439/2018F347
59. Przybyłek, A.: An empirical study on the impact of AspectJ on software evolvability. Empir. Softw. Eng. **23**(4), 2018–2050 (2018). https://doi.org/10.1007/s10664-017-9580-7,2018
60. Przybyłek, A., Zakrzewski, M.: Adopting collaborative games into agile requirements engineering. In: 13th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2018), Funchal, Madeira, Portugal (2018)
61. Ramesh, B., Cao, L., Baskerville, R.: Agile requirements engineering practices and challenges: an empirical study. Inf. Syst. J. **20**(5), 449–480 (2010)
62. Redlarski, K., Weichbroth, P.: Hard lessons learned: delivering usability in IT projects. In: 2016 Federated Conference on Computer Science and Information Systems (FedCSIS 2016), Gdansk, Poland (2016). http://dx.doi.org/10.15439/2016F20
63. Ribeiro, C., Farinha, C., Pereira, J., da Silva, M.M.: Gamifying requirement elicitation: practical implications and outcomes in improving stakeholders collaboration. Entertain. Comput. **5**(1), 335–345 (2014)
64. Robertson, J.: Requirements analysts must also be inventors. IEEE Softw. **22**(1), 48–50 (2005)
65. Rodriguez, P., Markkula, J., Oivo, M., Turula, K.: Survey on agile and lean usage in Finnish software industry. In: ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, Lund, Sweden (2012)
66. Sakhnini, V., Mich, L., Berry, D.M.: The effectiveness of an optimized EPMcreate as a creativity enhancement technique for website requirements gathering. Requir. Eng. **17**(3), 171–186 (2012)
67. Svensson, R.B., Taghavianfar, M.: Selecting creativity techniques for creative requirements: an evaluation of four techniques using creativity workshops. In: 23rd IEEE International Requirements Engineering Conference, Ottawa, Canada (2015)
68. Taylor-Cummings, A.: Bridging the user-IS gap: a study of major information systems projects. J. Inf. Technol. **13**, 29–54 (1998)
69. Trujillo, M.M., Oktaba, H., González, J.C.: Improving software projects inception phase using games: activeaction workshop. In: 9th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2014), Lisbon, Portugal (2014)
70. Weichbroth, P.: Facing the brainstorming theory. A case of requirements elicitation. Studia Ekonomiczne **296**, 151–162 (2016)
71. Weichbroth, P.: Delivering usability in IT products: empirical lessons from the field. Int. J. Softw. Eng. Know. (2018). https://doi.org/10.1142/S0218194018500298
72. VersionOne, 12th Annual State of Agile Report (2018). https://stateofagile.versionone.com
73. Springer, O., Miler, J.: The role of a software product manager in various business environments. In: 2018 Federated Conference on Computer Science and Information Systems (FedCSIS 2018), Poznan, Poland (2018). http://dx.doi.org/10.15439/2018F100