



Student's name and surname: Dagmara Kotecka

ID: 137314

Second cycle studies

Mode of study: Full-time studies

Field of study: Informatics

Specialization: distributed applications and
internet services

MASTER'S THESIS

Title of thesis: Enhancing Scrum with collaborative games

Title of thesis (in Polish): Usprawnienie metodyki Scrum poprzez gry zespołowe

| | |
|-------------------|--------------------|
| Supervisor | Head of Department |
| signature | signature |
| dr Adam Przybyłek | |

Date of thesis submission to faculty office:



STATEMENT

First name and surname: Dagmara Kotecka

Date and place of birth: 12.05.1992, Gdynia

ID: 137314

Faculty: Faculty of Electronics, Telecommunications and Informatics

Field of study: informatics

Cycle of studies: postgraduate studies

Mode of studies: Full-time studies

I, the undersigned, agree/do not agree* that my diploma thesis entitled: Enhancing Scrum with collaborative games
may be used for scientific or didactic purposes.¹

Gdańsk,
.....
signature of the student

Aware of criminal liability for violations of the Act of 4th February 1994 on Copyright and Related Rights (Journal of Laws 2006, No. 90, item 631) and disciplinary actions set out in the Law on Higher Education (Journal of Laws 2012, item 572 with later amendments),² as well as civil liability, I declare that the submitted diploma thesis is my own work.

This diploma thesis has never before been the basis of an official procedure associated with the awarding of a professional title.

All the information contained in the above diploma thesis which is derived from written and electronic sources is documented in a list of relevant literature in accordance with art. 34 of the Copyright and Related Rights Act.

I confirm that this diploma thesis is identical to the attached electronic version.

Gdańsk,
.....
signature of the student

I authorise the Gdańsk University of Technology to include an electronic version of the above diploma thesis in the open, institutional, digital repository of the Gdańsk University of Technology and for it to be submitted to the processes of verification and protection against misappropriation of authorship.

Gdańsk,
.....
signature of the student

*) delete where appropriate

¹ Decree of Rector of Gdańsk University of Technology No. 34/2009 of 9th November 2009, TUG archive instruction addendum No. 8.

² Act of 27th July 2005, Law on Higher Education:

Art. 214, section 4. Should a student be suspected of committing an act which involves the appropriation of the authorship of a major part or other elements of another person's work, the rector shall forthwith order an enquiry.

Art. 214 section 6. If the evidence collected during an enquiry confirms that the act referred to in section 4 has been committed, the rector shall suspend the procedure for the awarding of a professional title pending a judgement of the disciplinary committee and submit formal notice of the committed offence.

STRESZCZENIE

Celem niniejszej pracy magisterskiej jest usprawnienie metodyki Scrum za pomocą gier zespołowych. Głównym elementem badań jest spotkanie Retrospektywa. W tej pracy badawczej szczególnie skupiliśmy się na wyszukiwaniu gier, które usprawniłyby kreatywność, współpracę między członkami zespołu oraz efektywną komunikację. Równolegle, skoncentrowaliśmy się na motywacji oraz opinii uczestników w kwestii ich zrozumienia gier. Co więcej, naszym celem była weryfikacja czy wprowadzone w tej pracy badawczej gry były bardziej efektywne niż standardowe procedury i czy mogłyby stanowić uzupełnienie standardowych procedur. Postawiliśmy sobie za zadanie, aby przekonać członków zespołu, że wprowadzenie gier zespołowych na stałe poprawi relacje międzyludzkie w zespole, jak również przyniesie korzyści dla projektu. Metoda badawcza użyta w tej pracy magisterskiej to Action Research. Wprowadziliśmy gry zespołowe w firmie Intel Technology Poland, aby pomóc im poprawić kreatywność, komunikację oraz współpracę w zespołach deweloperskich. Ponadto zebraliśmy ludzi i odbyliśmy spotkanie używając techniki focus group w celu wydobycia informacji na temat Scruma w praktyce. Dzięki temu podejęciu dowiedzieliśmy się jakie są wady i zalety Scruma z perspektywy praktyków. Oprócz tego udało nam się dowiedzieć za jakimi Scrumowymi praktykami podążają, jakie role i spotkania uważają za potrzebne oraz jakie są największe ich potknięcia związane ze metodykami zwinnymi. Dyskusja na temat metodyki Scrum odbyła się w grupie dwunastu osób. Podczas wprowadzania gier zbieraliśmy dane z ankiet od członków zespołu. Implementacja gier była podzielona na dwie fazy. Pierwsza faza wprowadziła dwie gry i po refleksji z zespołem badawczym i promotorem zmieniliśmy pytania w niej w celu uzyskania bardziej pogłębionej analizy. Druga iteracja implementowała pięć gier, które w rezultacie przyniosły zadowalające wyniki.

W trakcie realizacji niniejszej pracy magisterskiej wprowadziliśmy większość opisanych gier oraz została przeprowadzona ankieta na ich temat. Następnie wyniki zostały przeanalizowane w celu ich jak najlepszej implementacji w kolejnych iteracjach. Niektóre gry głównie w celu zapewnienia większej generyczności wniosków oraz z uwagi na ich wartościowość, były wprowadzane więcej niż raz.

Gry zostały podzielone na dwa rodzaje - podstawowe oraz uzupełniające. Gry podstawowe to takie na których opiera się całe spotkanie Retrospektywa, natomiast uzupełniające to takie, które je wspierają. Retrospektywa może odbyć się używając wyłącznie gier podstawowych, lecz nie ma możliwości przeprowadzania jej bazując tylko na uzupełniających.

W trakcie realizacji opisywanej pracy badawczej, dzięki współpracy z firmą Intel Technology Poland, udało nam się stworzyć jedną zupełnie nową grę oraz usprawnić istniejącą.

Podczas zbierania danych i analizowania ich, postanowiliśmy stworzyć narzędzie, które będzie wspierać i pomoże efektywniej pracować Scrum Masterowi. Zaimplementowaliśmy serwis internetowy, Retrospective Analyzer, bazując na informacjach zwrotnych wyciągniętych z ankiet. Główną funkcjonalnością systemu jest ułatwienie pracy liderów, podczas przeprowadzania spotkania Scrum Retrospective.

Po zakończeniu projektu gry zaimplementowane w firmie Intel Technology Poland są dalej wykorzystywane. Dotyczy to nie tylko zespołów w których badania zostały przeprowadzone, ale również zespołów, które dowiedziały się o proponowanym przez nas podejęciu od członków zespołów badanych.

Słowa kluczowe:

gry zespołowe, Scrum, inżynieria oprogramowania, Retrospekywa, innowacja, usprawnienie

ABSTRACT

The purpose of this master thesis is to enhance the Scrum methodology using collaborative games. The main focus of this work is on the Retrospective meeting. In this work, we tried to find games that would improve creativity, collaboration between team members and effective communication. Simultaneously, we also focused on the motivation and the opinion of the participants in terms of their understanding of the game. Moreover, we tried to verify whether the methods introduced in the research were more effective than the standard procedures or whether they complement the standard procedures. We attempt to convince the team members that using collaborative games permanently will bring improved results in terms of the project and relationship between team members. The methodology used in this work is Action Research. We implemented the games in Intel Technology Poland to aid them with issues related to creativity, communication and team work. Furthermore, we gathered people to perform the focus group to discover the Scrum methodology in practice. We aimed to obtain what the pros and cons of the Scrum were according to the developers opinion. Apart from that, we were able to find out what Scrum practices they are following, which meetings and roles in Scrum they find useful and what the biggest pitfalls or impediments in the case of the Agile Methodologies are. We discussed the Scrum Methodology with a group of twelve people. During the deployment of the games, we gathered data using survey responses from team members. The implementation of the games was divided into two iterations. The first deployed two games and after a reflection we changed the questions in the survey to extract more valuable characteristics. The second stage implemented five games resulting in a satisfactory outcome.

The majority of the games described have been introduced and we have also retrieved an opinion on the subject. Furthermore, the results were analyzed to achieve a better implementation in the next stage. Due to their value and in order to assure a generic results, some of the games were deployed more than once.

The games have been divided into two categories - fundamental and complementary. The Retrospective meetings are based on the fundamental games and the complementary games are those that support it. In our approach, the Retrospective can only happen if the basic games are used, but it is not possible to perform the meeting based only on the complementary games.

During the realisation of the described research work, thanks to the help of Intel Technology Poland, we were able to create a new game and improve the existing one.

During the collection and analysis of the data, we established that creating a tool will make the Scrum Master's work more efficient. We implemented a web service called Retrospective Analyzer based on the feedback from the surveys. The main functionality of the system was to ease the work of the leaders whilst performing the Scrum Retrospective meeting.

After the project has ended the games implemented in Intel Technology Poland are still being used. This includes not only the teams that participated in the research, but also teams which found out about the

Key words:

collaborative games, Scrum, software engineering, Retrospective, innovation, enhancement

TABLE OF CONTENTS

| | |
|---|----|
| List of important symbols and abbreviations | 9 |
| 1. Introduction | 10 |
| 1.1. Context and motivation | 10 |
| 1.2. Problem statement..... | 10 |
| 1.3. Research method | 11 |
| 1.4. Related work..... | 11 |
| 1.5. Outline of the thesis | 12 |
| 2. Scrum overview..... | 13 |
| 2.1. Scrum in theory | 14 |
| 2.1.1. Process phases..... | 14 |
| 2.1.2. Roles in scrum | 15 |
| 2.1.3. Meetings overview..... | 17 |
| 2.2. Scrum in practice | 20 |
| 2.3. Conclusions | 23 |
| 3. Retrospective dedicated games..... | 24 |
| 3.1. Fundamental games | 24 |
| 3.1.1. Speedboat game | 24 |
| 3.1.2. Glad/Mad/Sad game | 25 |
| 3.1.3. Starfish Game | 26 |
| 3.1.4. 360-degrees of appreciation Game..... | 26 |
| 3.1.5. Mood and improvements Game | 27 |
| 3.1.6. 5L's game | 28 |
| 3.1.7. Candy-Love Game | 30 |
| 3.2. Supplementary games..... | 31 |
| 3.2.1. Anonymous | 31 |
| 3.2.2. Safe Room | 32 |
| 3.2.3. One-word-retrospective game..... | 32 |
| 4. Games deployment and evaluation..... | 33 |
| 4.1. First iteration | 34 |
| 4.1.1. Speedboat Game deployment..... | 34 |
| 4.1.2. Starfish Game deployment..... | 36 |
| 4.1.3. Discussion | 36 |
| 4.2. Second iteration..... | 37 |
| 4.2.1. Speedboat Game deployment..... | 37 |
| 4.2.2. Starfish Game deployment..... | 39 |
| 4.2.3. Glad/Mad/Sad Game deployment | 40 |
| 4.2.4. Mood and improvements Game deployment..... | 42 |
| 4.2.5. 5L's Game deployment | 43 |

| | |
|--|----|
| 4.2.6. Discussion..... | 46 |
| 5. Retrospective analyzer system | 52 |
| 5.1. System overview..... | 52 |
| 5.2. Backend architecture | 56 |
| 5.3. Frontend structure..... | 58 |
| 5.4. System evaluation..... | 59 |
| 6. Summary..... | 62 |
| 6.1. Contribution | 62 |
| 6.2. Future work..... | 63 |
| 6.3. Conclusion | 63 |
| Bibliography | 65 |
| List of Figures..... | 67 |
| List of Tables..... | 68 |

LIST OF IMPORTANT SYMBOLS AND ABBREVIATIONS

- | | | |
|--------------------|---|--|
| Standard Procedure | — | A standard procedure in case of this thesis means a standard approach used in Retrospective meetings which contains three questions What was good/bad/to improve in the past iteration? (Good Things, Bad Things, Things to Improve) |
| Timebox | — | Fixed time period to each planned activity |
| Definition of done | — | Consistent acceptance criteria across all User Stories |
| IT | — | Information Technology |

1. INTRODUCTION

1.1. Context and motivation

The Scrum is a methodology that can be traced back to 1986, where it was introduced in the article "The New New Product Development Game"[1, 2], published by the Harvard Business Review. Back in 1986, companies such as Honda, Canon and Fuji-Xerox were producing world-class results using an all-at-once product development method, which was a scalable, team-based technique and one that emphasized the matter of having teams that are self-organized and empowered. What is more, they outlined the role of management in the development process. The name "Scrum" is not an acronym, it is derived from the sport of rugby, "where it refers to the way of restarting a game after an unintended infringement or in case the ball has gone out of play"[3]. In the eighties, large companies creating defense and IT projects were failing due to frequency growth, which led to numerous books explaining how to create a better process. The main issue in the projects back in the 1980s was that traditional methodologies were focused mostly on the documentation and producing secondary artifacts. That approach usually resulted in failure of the product, because it did not cover the business requirements. The growth of the companies and technological progress led to creating the Agile Manifesto in 2001[3, 4]. The created principles set contains rules as follows:

"We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more." [4]

Most of the teams which claim to use Scrum are usually not following described above principles. This action causes aversion to Scrum, because of lack of understanding it. When Scrum is not implemented as designed, major problems may occur, especially when team members do not understand the purpose of the meeting or of the role. In terms of this master thesis a collaborative game can be seen as an activity, which leads to learning new skills and applying them to overcome challenges, getting rewards or punishments. In this research paper, serious games are indicated as those which can be played to achieve a precise goal and to provide entertainment. The serious games might be used as a strategy to address different kind of challenges and to avoid obstacles[5]. The motivation of this master thesis was to verify if collaborative games support Scrum principles[4] such as interactions and collaboration.

1.2. Problem statement

The goal of this research is to propose and implement a set of collaborative games to improve communication, motivation, creativity and involvement amongst team members. Moreover, we tried to confirm whether the collaborative games increase the "continuous attention to technical

excellence" and whether it helps "build projects around motivated individuals", who take part in "face-to-face conversations". In this research, we also tried to investigate whether the effectiveness of the team can be improved by the usage of proposed solutions.

1.3. Research method

The research design is an action research method deployed in Intel Technology Poland in Gdańsk in collaboration with the teams working there. The company was experiencing issues in particular fields of Scrum methodology deployment. The main problem that occurred in Intel teams was a lack of involvement during the Retrospective meeting. Other issues were also observed, such as poor creativity, failure to initiate a discussion and the feeling that the meeting was a waste of time that potentially could have been used for the product development. We cooperated with one of the main Scrum Masters in the corporation, Grzegorz Reglinski. Grzegorz is a certified scrum master (Professional Scrum Master), who has over 10 years of experience in working with Agile teams as a developer, product owner and scrum master. He has gained the practical knowledge, while working on Agile projects. His experience was built in Scrum, Kanban and Scrumban teams. From the agile side, Grzegorz is currently managing and supporting over 50 people working together in a project based on micro-services. The practical aim of our action research is to enhance the efficiency of retrospective meetings. Moreover, the research goal is to examine whether collaborative games can positively influence cooperation and motivation of scrum team members. What is more, we aim to verify whether the creativity of developers would increase using the proposed techniques and whether they are more involved, whilst being entertained. The important factor is also the extraction of valuable results using serious games.

To collect the feedback, we used a focus group and a survey.

A focus group is a discussion led by a moderator within a small group of six to ten people. The aim is to generate a maximum number of different opinions and ideas from various people in the time range between 45 to 90 minutes. The focus group is structured around no more than ten carefully predefined questions. We expect that the participants will stimulate and influence each other's thinking and be more open to discussion[6].

A survey "is a question or a series of questions in order to gather information about what most people do or think about something"[7]. This approach was used to retrieve feedback from team members after collaborative game deployment on the Retrospective meeting.

1.4. Related work

Morales-Trujillo et al. [5] wrote a paper on "Improving Software Projects Inception Phase Using Games" and his work was focused on Agile methodologies and their meetings. This master thesis aims to improve only the Scrum Retrospective meeting. Morales-Trujillo's goal was to help developers understand their customers, market and business opportunities. What is more, he aimed for the improvement of everyday issues in the organisation. Our work is strictly focused on human relations enhancement and retrieving valuable outcomes in the case of the project in the Retrospective meeting.

Another related work is a master thesis written by Mateusz Zakrzewski[8]. He focused on enhanc-

ing the requirements elicitation process using collaborative games. The target of the analysis was to improve the creation of the product backlog, sprint planning and sprint review. In our research we focused on the Retrospective meeting, mainly in the case of creativity, team involvement and collaboration between team members.

There is a strong linkage between Zakrzewski's paper and an article written by David Gelperin [9] regarding the improvement of the communication and cooperation between the developers and customers, while collecting the requirements. He introduces six serious games which help acquire a deep understanding of user and customer needs, through effective communication and cooperation.

We propose an innovative web service named Retrospective Analyzer, which is able to retrieve a game based on simple questions. The participants of the retrospective meeting are obliged to fill a form and based on their answers the system will suggest the most suitable game. According to our knowledge our system is the first of its kind.

1.5. Outline of the thesis

The thesis contains 7 chapters. The first one describes the aim of the work and is an introduction to the topic. The first section of chapter 2 presents the state of the art. It is focused on how does Scrum looks in theory, the process, what kinds of roles do we distinguish in this methodology and what meetings are required. In the second section of chapter two, the practical Scrum is described. The third chapter contains a description of the fundamental and supplementary games together with their requirements and rules. In the following chapter, games implementation and deployment has been elaborated. What is more, in chapter 4, the results of the team meetings and the outcome from the participants' feedback was presented. In addition, the chapter contains interesting aspects of the games implementation process. Chapter five includes a detailed description of the architecture of the created web service, the way the system works and the evaluation of the Scrum Master. The last chapter summarizes all the work that has been done in this master thesis.

2. SCRUM OVERVIEW

The concept of agile development was proposed in 2001. Agile Manifesto was elaborated by 17 developers, the purpose of it was to gather all important rules how to properly produce a good quality product:

"We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more." [4]

The Agile Manifesto is based on following twelve principles:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity—the art of maximizing the amount of work not done—is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.[4]

Scrum is the most popular agile methodology for developing products and services [10]. The Figure 2.1 shows, in a simplified way, how agile development works [2]. The main rule in development using Scrum methodology is that after each iteration (2-4 weeks of implementing planned features) the customer is able to get a usable product. The principle artifact in Scrum is product backlog, which is a list of features based on customer requirements, it should be prioritized from the most important functions to "nice to have" features or just less urgent. The backlog contains user stories, which is a form of expressing business requirement in Scrum, it is created in a way that can be understandable for both sides, business and development. There are a few templates with different structure, but the most popular and commonly used is:

1. Basic User Story structure [2]
 - (a) As <who, a role in system>, I want <what, a need> so that <benefit, goal> e.g.:
 - i. As a *company owner*, I want *the company logo to be visible on the welcome page* so that *customers are able to see it*.
2. Mike Cohn's User Story Structure [11]
 - (a) As <a role>, I want <goal/desire> e.g.:
 - i. As a *user*, I want *the company logo to be visible on the welcome page*.
3. Chriss Matts's User Story Structure [12]
 - (a) In order to <receive benefit> as a <role>, I want <goal/desire>
 - i. In order to *increase the number of sales of our print consumables* as a *marketing manager*, I want *customers to register their e-mail addresses*.

A user story is created to store it in Product Backlog and in the future divide it to tasks, but in order to do it developers should know the requirements, so the main purpose of user story is to start conversation, it is a catalyst to talk about requirements.

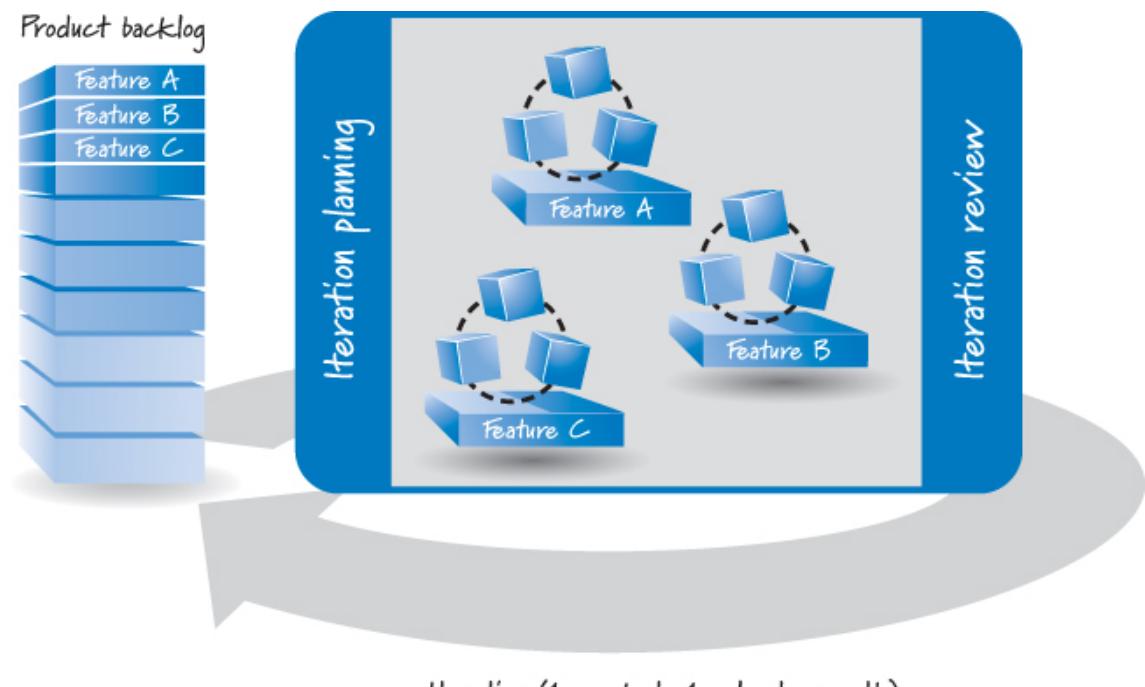


Fig. 2.1. Agile development overview [2]

2.1. Scrum in theory

In this section we will focus on scrum in theory, how it works, what are the required elements - who is required and what meetings. Often scrum in theory differs from scrum in practice and it is said that Scrum is just a tool and it should adjust to the team, not the other way.

2.1.1. Process phases

Scrum is a methodology based on incremental and iterative model of product development cycle as showed on Figure 2.2.

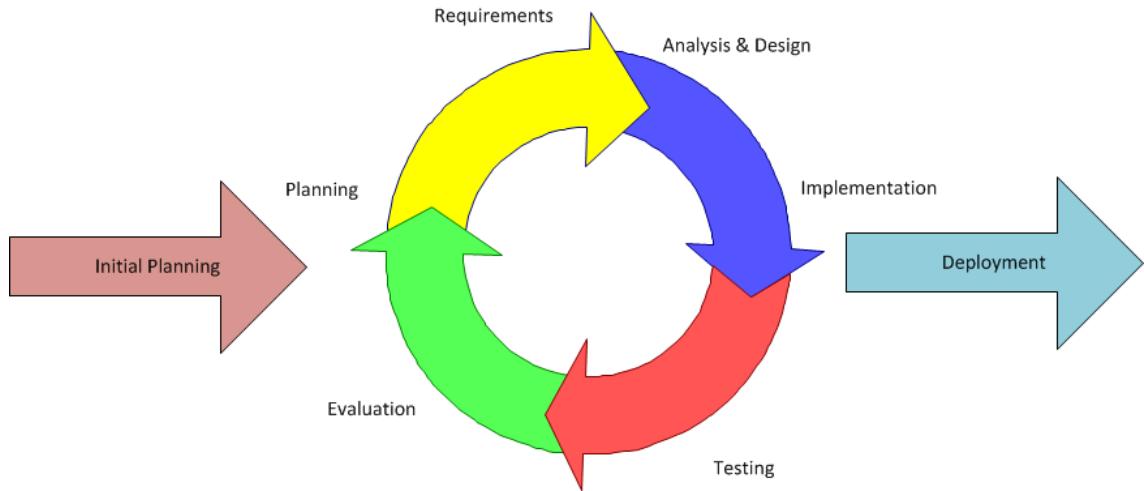


Fig. 2.2. Scrum cycle development

The incremental-iterative model is divided into stages, one whole cycle is called sprint, it is suggested that each sprint should be 2-4 weeks long. Every cycle contains planning, gathering requirements and deciding, what and how should be done, analysis and design, implementing what was planned on the particular iteration, in this moment of cycle we have two paths, we can decide to deploy our product if it is done or continue to next stage which is testing. The last element of the cycle is evaluation, if everything what was planned was actually implemented. The cycle is actually straight-forward and is very effective [2].

2.1.2. Roles in scrum

Scrum has determined each team member included in the project a particular role Figure 2.3, each role has its own responsibilities.

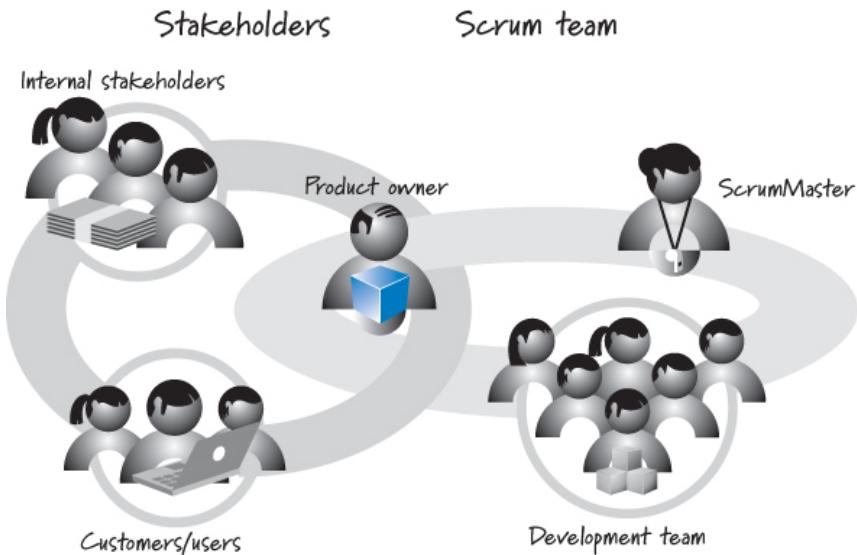


Fig. 2.3. Roles in scrum [2]

Product owner is a person that need to look in at least two directions simultaneously [2]. This role is responsible for communication between stakeholders and the scrum team. The principle obligations are shown in the Figure 2.4.



Fig. 2.4. Product owner responsibilities [2]

Product owner represents scrum team outside and is responsible for product development, decides which feature should be included in a particular sprint, adds user stories and features to the backlog, defines acceptance criteria and verifies whether they are full filled after the sprint.

Another important person in Scrum Team is Scrum Master, whose responsibility is illustrated on Figure 2.5. Mainly his role is to superintend the process and help development team to adapt to the agile methodology. This role is responsible mainly for removing impediments that inhibit team's productivity, protects team from outside interference so that they can remain focused on delivering good quality business value every sprint, servant leader of the Scrum team and team's process authority [2].

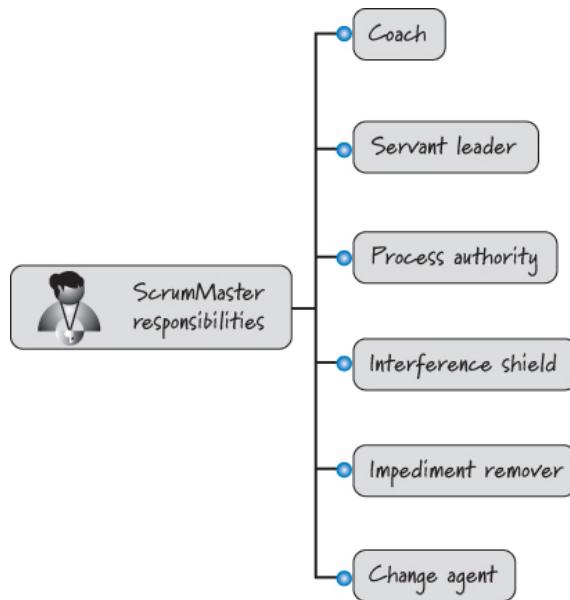


Fig. 2.5. Scrum master responsibilities [2]

The last, but without which there would be no product and which is essential, is development team, also called delivery team, design-build-test team or just team. Types of jobs in a development team for example are: architect, programmer, tester, database administrator, user interface

designer and many more. Development team is responsible for product implementation, testing, integration and design. The team should include people of various specializations and skills, who can fulfill project requirements. Delivery team is obliged to perform sprint execution, which means performing actions that will result with a ready functionality. Each member of the team is expected to participate in scrum meeting such as described in subsection 2.1.3. On this group this work will be mainly focused on [2].

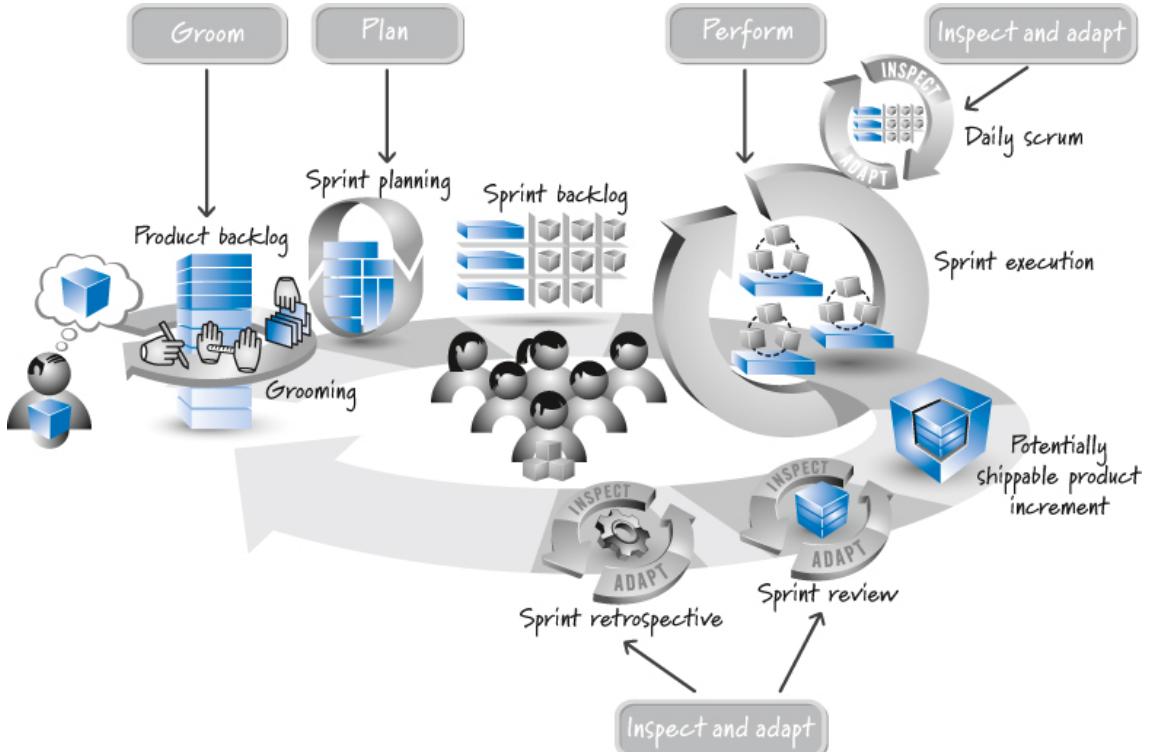


Fig. 2.6. Development Team responsibilities [2]

2.1.3. Meetings overview

A product development is composed of multiple sprints, which can last 2-4 weeks [2] and each iteration should deliver a usable product to a customer, which does not mean ready or fully functional implementation, just a product that can be potentially shippable. As presented on Figure 2.7, before sprint execution there should be a Backlog Grooming meeting, each iteration starts with Sprint planning and ends with Sprint Review and Retrospective, what is more every day begins with Daily Scrum. The next sections will describe all the meetings separately [2].

This Backlog grooming meeting is focused on maintaining the product backlog, it should be executed before next sprint planning. The main aim of this meeting is to [13]:

- remove user stories that are no longer relevant,
- creating new user stories in response to newly discovered needs,
- prioritizing the user stories,
- assigning or correcting estimates to user stories,
- splitting user stories which are high priority but too big to fit in an upcoming iteration.

On the Backlog Grooming it is mandatory that the Product Owner is present, Developments' Team

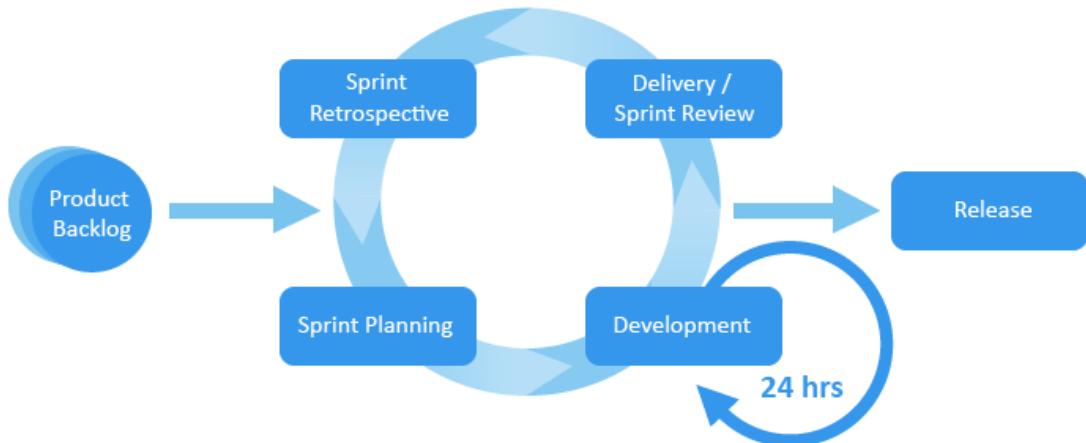


Fig. 2.7. Scrum meetings and process

and Scrum Masters' presence is optional [2].

The Sprint Planning is a meeting on which Product Owner shares initial sprint goal and answers questions regarding product backlog items. The initial requirements should be accomplished in order to start the meeting [14]:

1. Prioritized backlog after backlog grooming.
2. The highest-priority stories should be estimated.
3. Prepared definition of done.
4. The capacity of the team should be known, for example the team has a meeting outside working place that might affect the sprint, this should be included in the capacity field.

Two distinct objectives might be retrieved from the Sprint planning. Firstly, we must ensure that each team member understands what is expected to do in the particular user story and secondly divide user stories into tasks and determine what can they deliver and makes a realistic commitment. Each team member might select one of the tasks, which is willing to own and complete, but it is not a required action, which must be performed. Thereafter, the team estimates, taking all the factors into the consideration, the risk, developers experience, the difficulty of the task and estimate in hours how long it might take to complete a task. From the Sprint planning we have two primary outputs, first is the goal of the iteration, which means, what users stories should be delivered, second is the sprint backlog with list of created tasks [2, 14].

The Review is a meeting in Scrum which takes place at the end of an iteration. The product owner, scrum master and development team should be present, on the list of invited people might customers and managers may be included. The review meeting is a demo of a product that has been delivered after a sprint and also a chance for the customers to provide feedback. Quoting the agile manifesto "Working software is the primary measure of progress." and "Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage." those rules are fulfilled thanks to the review meeting [15].

The Daily Stand-ups is a short meeting which is held every day during the sprint and its purpose is to discuss issues that are preventing work from being done and how is the work progress. The Daily Stand-ups are time boxed to no longer than 15 minutes and the participants are obliged to stand. The team members stand in the circle and answer the following questions [15]:

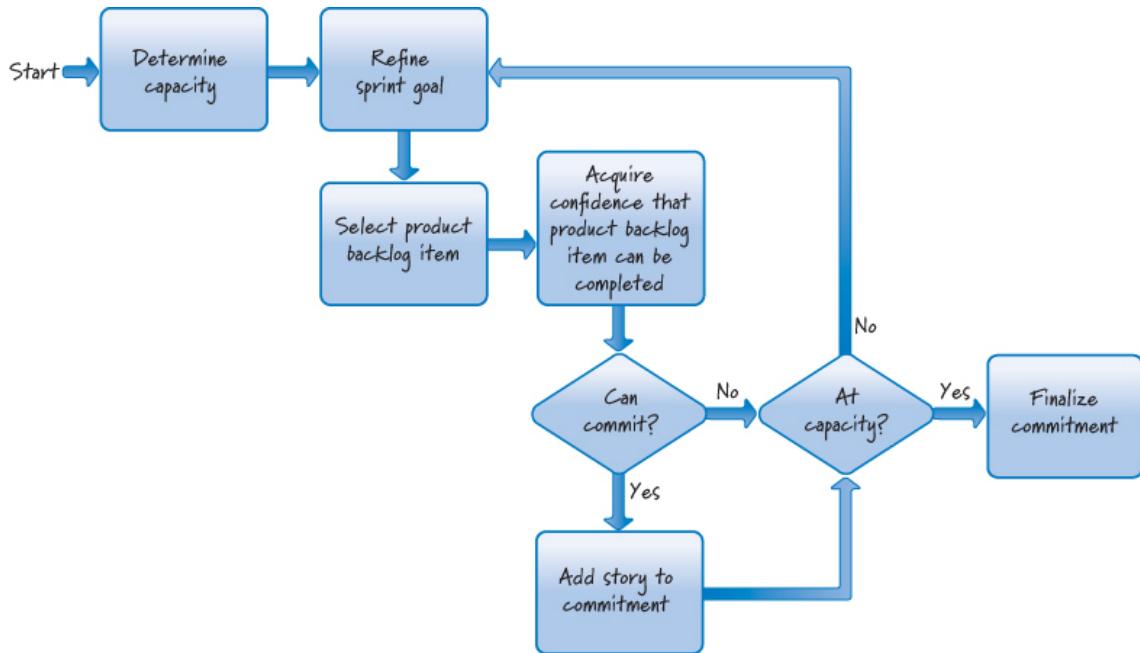


Fig. 2.8. Scrum One-part sprint-planning approach [2]

1. What have you done since yesterday?
2. What are you planning to do today?
3. Do you have any problems preventing you from accomplishing your goal? What progress has been made on existing impediments? Can the blockage be removed or must it be escalated?

It is indicated that in order to keep the meeting short, the major issues which were discovered during the Daily Stand-up will be discussed afterwards [15].

The Retrospective meeting takes place at the end of the meeting and is strictly focused on reflection of team members on the sprint that has passed. The encounter provides the participants a chance to congratulate each other on the things that succeeded and discuss once that went wrong [15]. The purpose of the retrospective was summarized by Norm Kerth, by quoting a fragment from "Winnie the Pooh" [2]:

Here is Edward Bear, coming downstairs now, bump, bump, bump, bump, on the back of his head, behind Christopher Robin. It is, as far as he knows, the only way of coming downstairs, but sometimes he feels that there is another way, if only he could stop bumping for a moment and think of it.

The Sprint Retrospective allows the participants to stop bumping and have a moment to think. The target is to examine what is happening in the team, analyze the way the team is working, identify the problems and ways to improve. The areas that the team should focus on and which are up for discussion are processes, practices, communication, environment, artifacts, tools, internal and external issues within the project and the development group [2].

It is said that the Sprint Retrospective is one of the most important and least appreciated practices in the Scrum and this is because there are people who think that it takes time away from doing "real things" such as developing, testing, designing. Its importance comes from fact that using this approach the team members are able to customize scrum to theirs needs and it is a crucial

contributor, as written in Agile Manifesto [4], to continuous improvement [2].

2.2. Scrum in practice

In order to evaluate how Scrum appears in practice, the best strategy is to ask the people who deal with it every day e.g. developers. The focus group is a form of research which includes people discussing their opinions and attitudes about a particular subject. The focus group has been created to retrieve the opinions about Scrum in practice. They were asked a set of eight questions listed below:

1. Engagement questions:
 - (a) How and from whom did you first learn about the Scrum framework?
2. Exploration questions:
 - (a) What were your biggest pitfalls/impediments in the case of Agile Methodologies in your career?
 - (b) What are the pros and cons of Scrum?
 - (c) What Scrum practices are you following and which are you not? Why? (Meetings, Sprints, Roles, Iterative and incremental practices, Product Backlog, Sprint Backlog)
 - (d) Which meeting do you think is useful and which not? Why?
 - (e) Which Scrum role do you think is useful and which not? Why?
 - (f) If you could change the methodology in your team, what would it be?
3. Exit question:
 - (a) Is there anything else you would like to add to the discussion about Scrum?

The group contained twelve people and the profiles of the developers are presented on the Table 2.1.

Table 2.1. Participants of focus group profiles

| Role | Experience in IT | Quantity |
|------------------|--|----------|
| Scrum master | 10 years | 1 |
| Product Owner | 2 years | 1 |
| Design Lead | 10 years in development, 1 year as design lead | 1 |
| Developer | around 2-3 years on average | 7 |
| Developer | 6 years | 1 |
| Developer Intern | 6 months | 1 |

After the meeting, where the discussion took place, we evaluated the results and they were satisfactory. The goal of the focus group which is "continue the discussion until no more new ideas appear" had been achieved.

In the first question, the warming up and introduction to the subject, most of the participants revealed that they found out and learned scrum at work or university. Two out of twelve participants attended Scrum courses.

On the question "What were your biggest pitfalls/impediments in the case of Agile Methodologies in your career?" the participants retrieved five main ideas:

1. It is hard to work in a big team with multiple small agile teams, when there are many dependencies between those small teams.
2. There is a lack of discipline during daily team scrums - short summaries of everyday problems

tended to evolve into long discussions.

3. Many people have different views of what Agile is, and because of this many companies interpret what Agile must consist of.
4. The inability for a company to change their interpretation of Agile to better suit their business.
5. Bad interpretation of the Scrum leads to chaotic work.

Participants' opinions on the pros of the Scrum are as follows:

1. Regular meetings keep teams focused.
2. Scrum is easy to manage and creates room for improvement.
3. Scrum efficiently organizes the work in development teams and clearly separates "What we do?" from "How will we do it?".
4. Operating on timeboxing, which is the time taken for a task to be completed.
5. Builds good habits like for example, sprints - work in an equal and repeatable manner in the case of time segments.
6. Predictable methodology - provides metrics that can measure the tempo of the work and capacity.
7. Developers like to focus on work without major interruptions.
8. Continuous development thanks to the retrospective meetings.
9. Flexible in case of changes to the market and no need to prepare a documentation before the project starts.
10. Scrum is flexible, allows teams to self-organize themselves.
11. Fast iterations and feedback.
12. Scrum meetings are helpful in terms of relations between people.
13. One of the least restrictive methodologies and due to that, it is an appropriate choice to choose Scrum.
14. Spreading the knowledge about current tasks among the entire project group - everyone is better oriented about all current tasks in the team. Technical knowledge is better spread among team members - good way to know if someone has previously dealt with similar issues.
15. Large projects are divided into easily manageable sprints.
16. Works well for fast-moving development projects.
17. Scrum, being agile, allows to retrieve feedback from customers and stakeholders.

And regarding the answer on the second part of the third question - the cons are as follows:

1. In big projects, there is no need to plan everything straight away.
2. Teams take ownership, so lazy teams mean bad results.
3. It is hard to convince the client to collaborate and participate in the project. They are used to work that is based on a specific model in which they are expecting the work to be done in a defined period of time after the payment.
4. It is hard to understand the simple rules of Scrum for the customer.
5. It is hard to accept the symbiosis of the Development Team and the Product Owner. Unfortunately, very often the business is separated from the developers and they come up with ideas that are not useful, but the developers are forced to implement it.
6. There is no good place for unexpected bug fixing during sprint.
7. Most of the people do not understand Scrum and claim that they are using it.
8. The duration of the meetings should be decreased.
9. Time overspent on daily status reports and planning.

10. Most of the time spent on daily scrums brings little value - it seems like it involves just listening to the team members' thoughts and problems without a perspective to actually help people out.
11. Scrum often leads to scope creep, due to the lack of a definite end-date.
12. Adopting the Scrum framework in large teams is challenging.
13. Based on external research, the best quantity of team members is a team of seven as people then feel like a family.

The pros of the Scrum are extending the cons which is satisfying. In the group there were Scrum "lovers" and Scrum "haters" so it is a success that the number of positive aspects of Scrum was larger than the negative.

The fourth question was about Scrum practices used in the participants' teams. A few of them did not see the point of following any of the meetings, besides Daily Stand-ups and claimed that their team does not do that. A few of the attendants of the focus group did not understand the meaning of these practices in scrum and they felt like they were forced to do them, because of the place they work at. The majority understands and thinks that most of the Scrum Practices are useful and helpful for the project:

1. Daily Stand-ups in the same place and at the same time.
2. The best quantity of team members is a team of seven, based on the external researches, as people then feel like a family.
3. Retrospective is a key to success as it helps to develop the project and the team.
4. Backlog and Sprint Backlog are crucial for development, but unfortunately very often there is too little time for that.

In case of the usefulness of the meetings, the common opinion was that meetings that require discussion over matters that share ownership are important. If an issue has one owner, there should not be a meeting unless completely necessary. The majority of the participants did not see the point of attending all the meetings as they found that only Sprint Planning and the Sprint Stand-ups were useful. Only three out of twelve people agreed on the importance of all the meetings.

The sixth question was about the usefulness of the roles in Scrum. The participants agreed that they are all useful if done correctly and fit the business processes of the company.

The last question of exploration questions was about changing the methodology in the team. Only one of the participants said they would change the methodology to Kanban. The rest of the research attendants, who were previously able to retrieve a number of cons regarding the Scrum, said they would stay with the Scrum as they believed it is the best methodology.

The exit question brought a new topic to the table. The participants agreed that they in general do not understand Scrum. It has been suggested that each member of the Scrum team should read a Scrum Guide a few times and think about what "timebox", "definition of done" and "commitment to deliver a task at the end of the sprint" means to them. What is more, the most common Scrum mistakes were retrieved:

1. Forgetting about Scrum Retrospective meetings.
2. Not following the "definition of done" while ending a task.
3. Not considering the capacity of the team during a Scrum Planning meeting.
4. The Product Owner tells the team how to implement a feature.

5. The Development Team does not provide technical feedback to the Product Owner.
6. The Development Team does not perform Scrum Daily Stand-ups.
7. During the Scrum Daily Stand-up, the participants would talk about how they implemented a task instead of what they did and what they intend to do.
8. The Scrum Review meeting is being treated as an acceptance meeting, while the tasks and user stories have to be closed before the meeting.

To conclude, very often the team members using the Scrum methodology do not understand the purpose of this methodology. A requirement in every company should be a good understanding of the methodology that they are using. The organisation should provide their employees with adequate courses and coaches that will tell them how Scrum works and what its aim is. Moreover, Scrum is very often underestimated, despite it being the most popular used approach in case of developing software.

2.3. Conclusions

Nowadays Scrum is the most popular agile methodology for developing products and services [10], but usually is not implemented as designed. People tend to change the Scrum in order to adjust it for their purpose. For example, following only two meetings, because they do not see the purpose of the other Scrum meetings. Another discovered issue is that team members do not understand Scrum, which creates a major aversion to it. The developers feel that the time spent on the meetings is a waste of time and they like to focus on work without major interruptions. Most of the participants of the focus group claimed that if they could, they would only attend on Scrum Planning meeting and Daily Stand-Ups meeting, but usually they are forced to participate in all of them. The latter creates another issue, forcing team members to attend a meeting that they do not understand may result in aversion to it or in some cases permanent removal of the meeting.

The most underestimated meeting is the Scrum Retrospective meeting, because it mainly focuses on the improvement and on the people. During the focus group we learned that some of the participants know the Retrospective only in theory, but they never experienced it in practice. Moreover, based on the focus group research the aversion to Scrum Retrospective meeting is common, even the participants who experienced the Retrospective meeting feel that it is a time consuming and useless meeting. The lack of understanding and implementing Retrospective not as designed is the reason of the aversion and the cause of the underestimation of the Retrospective meeting. Thanks to the Retrospective meeting we are able to achieve a continuous development in terms of the project and the relations between the team members.

It is impossible to create a perfect methodology and Scrum is not a perfect solution, but it solves plenty of issues, helps the team to self-organize and allows to efficiently develop a product.

3. RETROSPECTIVE DEDICATED GAMES

Communication inside any team can be difficult especially when it involves revealing your own personal opinion which might differ from others. Speaking out loud about problems even in small group might be shameful and most of people do not like to open up in front of others, which might cause layering issues not only in communication between people, but also in developing good quality software. Based on experience that we learned from Scrum Masters and Technical Leaders in company Intel Technology Poland, most people have enormous knowledge about technology, new frameworks or developing with good standards, but they are not used to share it, because they are afraid of criticism, so they rather sit quiet and let other, more brave, sometimes less experience, people talk. The following games are invented for the situations, which were mentioned before to minimize the shyness and fear and to maximize awareness of possible problems and to enforce team to discuss them.

3.1. Fundamental games

To start with, we should be aware of existence of a standard procedure mentioned in subsection 2.1.3. The 3-statement method which contains "Good things", "Bad things" and "Things to improve" is the most popular and the most frequently used by teams, based on Grzegorz Regliński knowledge and experience, technique of conducting Retrospective. Games presented in this chapter are an innovative approach to this meeting, which often changes users' point of view, shows a problem, both inside and outside of the team, from a different angle and entertains, while retrieving interesting and impressive results.

3.1.1. Speedboat game

"Speedboat" (also known as Sailboat) is a game which changes the perspective of looking on retrospective and problems. Team members do not focus what problems they had in terms of ordering them into three columns using 3-statement method, but thanks to change of view they start to think creatively and might retrieve more valuable issues. Figure 3.1 shows what should be drawn on a blackboard. Every item represents different statement [16]:

1. The cloud which is creating the wind symbolizes things that were successful, things that are pushing our team forward and the product. For example, these things could include good code reviews in the previous sprint, team members sharing their knowledge and helping others.
2. The anchor symbolizes things which could have been done better, because the idea is good but the execution is not being done well. For example, these things could include crowding pull request reviews because only one person is doing them and bad communication in the team caused by for example, solving problems that someone already had and consequently did not share the solution with others. The latter leads to time being wasted.
3. The rocks symbolizes things that we wish did not happen again or something that may cause our product major problems now or in the future. These are things that we want to lose. For example, the planning might not have gone well causing many people to be without tasks because of bad division or dependency.

The game should start with describing people what is game about. The main aspect is that people

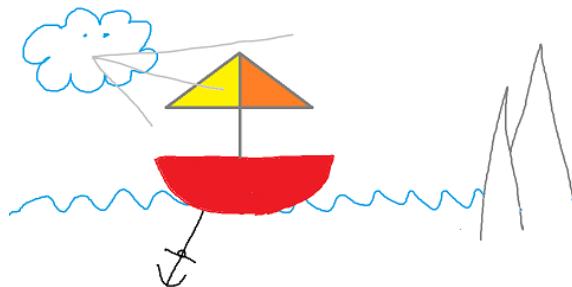


Fig. 3.1. Speedboat

understand what is required from them during “Speedboat” game. The Table 3.1 below shows what is required to start with this improvement game.

Table 3.1. Requirements to start Speedboat game

| | |
|----------|---|
| Target | Improve creativity in the team and maximize solving problems |
| Required | <ol style="list-style-type: none"> 1. Minimum 3-people scrum team 2. Blackboard 3. Sticky Notes 4. Pencil/Pen |
| Time | About 45 minutes |

After description the leader draws the picture, like on Figure 3.1 and team members should stick sticky notes where think they belong. The next step, if everybody already is out of ideas, is to discuss the results displayed on blackboard, maybe a particular issue is for one person positive and for another is negative, they should clarify and talk about it. This approach is not only productive for the team, but also for project, and might retrieve interesting outcome.

3.1.2. Glad/Mad/Sad game

The Glad, Mad, Sad game, also known as Happy, Sad, Angry is a game in which the manager or scrum master is able to observe teams mood, which is definitely a useful factor to retrieve from a retrospective meeting. Figure 3.2 visualizes how the scrum master should draw in order to conduct "Happy/Sad/Angry" game. The board should be divided to three spaces, where each space has its own meaning and purpose [17]:

1. Happy Face symbolizes successful things which occurred in current iteration, accomplishments and valuable effects for the team and project, things that we learned and you are thankful for.
2. Sad Face symbolizes things that made a team member not satisfied of his/hers work, an action which could be done better.
3. Angry Face symbolizes things which caused problems that made team members annoyed, issues which were the source of iteration failure or the work harder than could have been using e.g. other tool, framework etc.

To start with the leader, who is responsible for retrospective should describe the rules of described in this subsection game. Required equipment for this approach is presented in Table 3.2.

The next steps, after team members understands what is the purpose and what are the rules of

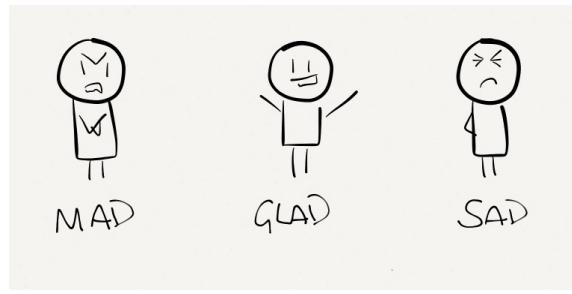


Fig. 3.2. Happy/Sad/Angry

Table 3.2. Requirements to start Happy/Sad/Angry game

| | |
|----------|---|
| Target | Retrieve teams mood and maximize solving problems |
| Required | <ol style="list-style-type: none"> 1. Minimum 3-people scrum team 2. Blackboard 3. Sticky Notes 4. Pencil/Pen |
| Time | About 45 minutes |

described game, are the same as in Speedboat game, filling sticky notes and discussion.

3.1.3. Starfish Game

The next game I would like to describe shows a totally different approach for conducting retrospective. The game is called "Starfish" [18], because this shape divides board to five areas like on Figure 3.3 :

1. Keep doing - this area describes things which are being done well by the team and they should continue doing them because they are valuable for the project and team members.
2. Stop doing - things which are "not bringing value, or even worse, it is getting on the way" [18].
3. Start doing - things which were working for you in the past in other team/project or new ideas, approaches which might help team in delivering product and in general are beneficial for the team.
4. Less of - things which are being done already and they bring value, but should be reduced.
5. More of - things which are being done already and they bring value, but they would bring more value if done more [18].

This game shows a different perspective than is in usual 3-statement method and thanks to that, leaders are able to retrieve problems and issues in nonstandard fields. The requirements are presented in Table 3.3.

3.1.4. 360-degrees of appreciation Game

Another team building game is called 360-degrees of appreciation and is an activity that fosters open appreciation feedback within a team. The game is useful in case of increasing team moral and improving people relationship. The Table 3.4 presents requirements for the game and the rules are as follows [19]:

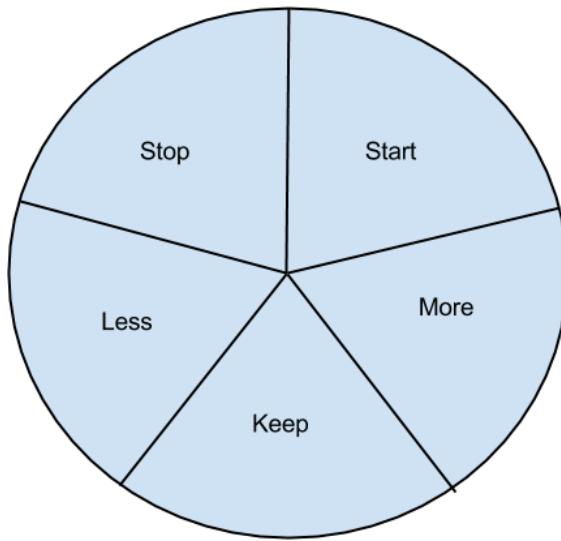


Fig. 3.3. Starfish

Table 3.3. Requirements to start Starfish game

| | |
|----------|---|
| Target | Retrieve problems, increase creativity and maximize solving problems |
| Required | <ul style="list-style-type: none"> 1. Minimum 3-people scrum team 2. Blackboard 3. Sticky Notes 4. Pencil/Pen |
| Time | About 60 minutes |

1. Gather all participants around the table.
2. Write appreciations on your paper for each member of the team gathered around the table.
3. If everyone is ready, ask the participants to sit in a circle.
4. Choose one person that will sit or stand in the middle of the circle.
5. Everyone in the circle should read their appreciations toward the person sitting in the center.
6. Repeat steps four and five for everyone in the circle.

Table 3.4. Requirements to start 360-degrees of appreciation game

| | |
|----------|---|
| Target | Motivates constant feedback, strengthens relationship and trust |
| Required | <ul style="list-style-type: none"> 1. Minimum 3-people scrum team 2. Paper or Sticky Notes 3. Pencil/Pen |
| Time | About 2 minutes for each participant for the item 2 of rules and about 10 minutes for item 5 |

3.1.5. Mood and improvements Game

During the cooperation with Intel Technology Poland teams, the "Mood and improvements" game has been created. The activity is a merge of two approaches, both of them were already described, first in the subsection 3.1.2 and the second one has its roots and has been inspired by the game from subsection 3.1.4. The goal is to retrieve as valuable as possible feedback from the team

about the project and simultaneously increase the trust, strengthen the relationship between team members and retrieve from the team ideas how to improve the project or the team. As presented on Figure 3.4 the game has five areas that need to be filled. The first three fields on the top are described in subsection 3.1.2, because the upper part of the picture is the exact reflection of Glad, Mad, Sad game. The lower area of the Figure 3.4 presents:

1. Flowers - representation of statement "Appreciations", in this section we thank people for things that they have done for us, the team, the project in the past iteration, e.g. quick and upright code review, help in the task etc.
2. Light bulb - is the representation of expression "Ideas", in this area we stick the papers with concepts how to improve the team or the project, e.g. estimate more carefully on the planning, dedicate some time in the iteration to increase code coverage etc.

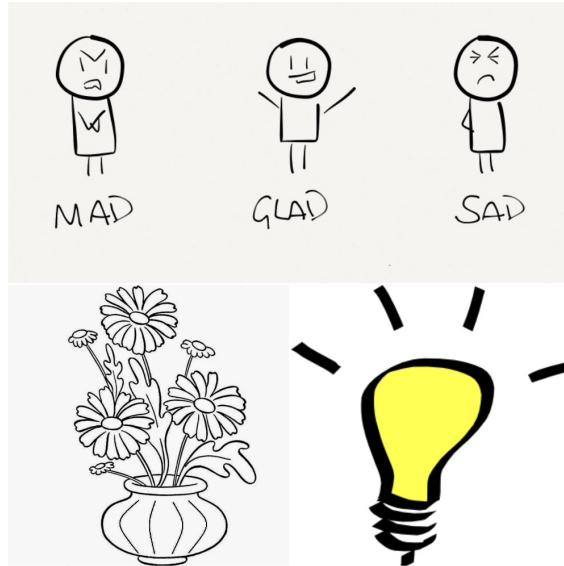


Fig. 3.4. Mood and Improvements

Before implementing the game in the team requirements presented in Table 3.5 must be completed and the team needs to be aware how to play the game.

Table 3.5. Requirements to start Mood and Improvements game

| | |
|----------|---|
| Target | Retrieve teams mood and maximize solving problems |
| Required | <ol style="list-style-type: none"> 1. Minimum 3-people scrum team 2. Blackboard 3. Sticky Notes 4. Pencil/Pen |
| Time | About 60 minutes |

3.1.6. 5L's game

The second game created, thanks to the collaboration with the Intel Technology Poland teams, is 5L's Game. The activity has not been developed from scratch, there is an existing solution called 4L's [20], which we have improved, by adding one additional element. The initial version of the game is shown on Figure 3.5 and it contained four fields:

1. Liked - things that the participants liked about the past iteration, considering project and team.
2. Learned - things that the team members learned in the past iteration, new language, used new framework or tool, this might also be things not connected to the project like for example while being on a integration with the team the participants learned how to windsurf or make sushi.
3. Lacked of - things that have been done in the past iteration and you do not want to stop doing them but you have an idea how to do it better, how to improve the action, or just wish that it could be done, for example more efficiently.
4. Longed for - things that you wish would have been done, for example more attention on code quality, rather than focusing just on the delivery.



Fig. 3.5. 4L's

While implementing the game the team that we were currently working with (team B) suggested that there should be an area where you can describe things that the team disliked in the past iteration. After a discussion we developed the 5th "L" which was "Loathing". The last "L" included in the initial 4L's game was created in order to retrieve negative feedback from the team and, if possible, improve the area where the problem occurred or even dispose it. The Table 3.6 presents the requirements needed in order to start performing the retrospective meeting in using 5L's technique.

Table 3.6. Requirements to start 5L's game

| | |
|----------|---|
| Target | Change perspective in order to learn new things about the team and gather data |
| Required | <ol style="list-style-type: none"> 1. Minimum 3-people scrum team 2. Blackboard 3. Sticky Notes 4. Pencil/Pen |
| Time | About 60 minutes |

The goal of this game is to gather as much as possible information from the team, this technique also changes the perspective. The approach can be implemented in circle version like presented on Figure 3.6 or in the standard column way. What is more it is also recommended to use for each column or area, different color of the sticky notes, this separates and organizes the data.

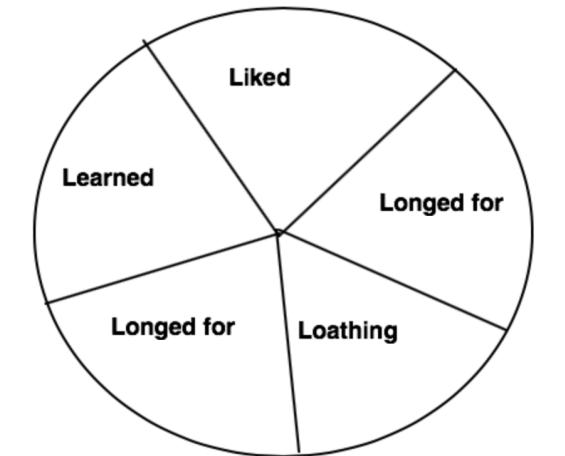


Fig. 3.6. 5L's

3.1.7. Candy-Love Game

Another game, which may be valuable to focus on, is more a team building game rather than solving problems in case of project, but in order to maximize performance and product quality it is advisable to have strong and integrated team [21]. Harvard Business Review claim that "Research shows that when people work with a positive mind-set, performance on nearly every level—productivity, creativity, engagement—improves. Yet happiness is perhaps the most misunderstood driver of performance.". [22] "Candy Love" game integrates team members by letting them speak up and talk about their life beyond work [23]. In order to deploy the game and show it to your team, necessary requirements presented in the Table 3.7 are needed.

Table 3.7. Requirements to start Candy-Love Game

| | |
|----------|---|
| Target | Increase trust in the team, help people to speak up and get to know each other |
| Required | <ol style="list-style-type: none"> 1. Minimum 3-people scrum team 2. Pack of MM's, Skittles or other colourful candies 3. Jar for candies 4. Meanings of colors (rules) |
| Time | About 2-4 minutes per person |

The participants need to sit at the table, without laptops and other distracting things. One person picks the candy out of the jar and shows it to the team, than checks the candy color meaning. The others need to focus on what the person which picked the candy is talking about. The meanings of the colors are necessary to have in order to properly perform the game, because it's hard to remember all the meanings of the colors, which are as follows:

1. Red - share one thing that you like about your job, this color retrieves positive emotions, especially on people that are not satisfied or happy with what they do.
2. Yellow - share your life goal that you are working on, this color illustrates others what is important to you.

- tant for that person and can inspire.
3. Green - share your favorite movie or/and book, this shows person from different angle, maybe this might start a conversation later with people that we usually have nothing to talk about.
 4. Purple - share your favorite way of reviving yourself on a regular workday, this color shows what the person thinking about to decrease stress and might be also a great catalyst for a conversation.
 5. Blue - share one stressful thing in work that you wish you could improve, this color shows what stresses person out, maybe someone has a solution that can share, this can also convert a negative thing into a positive.
 6. Orange - share what is your favorite food, maybe later you can share a meal with someone from the team and it is a topic that everyone likes.

The participants should pass the jar to each team member and end the game when all the candies are gone, when all the team members picked one sweet or when the time of the meeting has ended [23].

3.2. Supplementary games

The techniques that are going to be described in this section are or were probably used by major number of the teams working in information technology, most of them are not aware of its existence and how valuable they are. These approaches are just supplementary games to fundamental ones described in section 3.1. In order to properly and effectively implement a retrospective meeting the team should chose one of the fundamental games and as an additional value add one or more supplementary game, which may increase number of valuable opinions and feedback from the team.

3.2.1. Anonymous

Based on our work in Intel Technology Poland site in Gdańsk and also on Rui Miguel Ferreira [24], who wrote his article based on Hiren Doshi article, which was posted on Practice Agile Software Development blog, we can discover power of anonymous retrospectives. According to the article [24] no matter how supportive, open and transparent the company is, there will be always individuals that are not eager to share their thoughts with others [25]. They suggested the anonymous technique to encourage introvert individuals to also share their point of view on the project. The following model was presented:

1. All the participants of the meeting must be involved and take part in this activity.
2. The data should be anonymous, untraceable to the writer, for example decide that everything will be written using blue pen and capital letters.
3. Collect the papers into an empty container and mix them.

Using this technique all the team members are comfortable to share their thoughts and opinion without the feeling of shame or embarrassment. According to our research we learned that most of team members do not like retrospective meeting, because they do not like to expose and talk about theirs feelings, opinions and thoughts. Using this approach it might change retrospective into a valuable meeting without forcing participants to talk and what we also discovered once something is told by an extrovert its stops being "taboo" and even introverts add something to

discussion [25].

3.2.2. Safe Room

The retrospective is a meeting strictly focused on a team or project and the managers or product owners, if not directly invited to discuss specified problems, should not be present [25]. The safe room techniques is also commonly used in such meetings, but as we mentioned before. The retrospective meeting should be always dedicated to the team and theirs issues and the participants should be comfortable to share their thoughts. We already described one approach how to encourage team to speak up, the second technique is to create a "safe room". To put the "safe room" theory into practice, the person, who is leading the meeting should inform the attendants that everything what is said in the room should stay in the room. This technique must be based on assumption that the team trusts each other that whatever is said during the meeting will not be repeated outside the team, it is a non-formal confidentiality agreement.

3.2.3. One-word-retrospective game

The "One-word-retrospective" is a game that is supposed to help team to deal with feelings. This technique is used to increase the collaboration, respect and improve the understanding in the team [26]. The rules of this supplementary game are as follows:

1. Gather the team and ask each participant to tell how does they feel, using just one word, about the past iteration.
2. Collect all the words, they might be written on sticky-notes using Anonymous technique.
3. Write all the words on the board, so everyone is able to see them.
4. Then start to ask the participants why do they feel that way, use the exact words from the board.
5. List the major issues and confirm them with the team.

The requirement to implement this supplementary game in the team is to have following things [26]:

1. Establish trust and openness.
2. Respect people and their feelings.
3. Be able to deal with the issues.

The most important thing, not only using one-word-retrospective technique, is to ensure that the team trusts each other, especially because you are dealing with peoples' emotions and feelings. The leader should make sure that in order for people to speak openly, he has to introduce, if necessary, depends on the team, rules like for example the anonymous or/and safe room techniques.

4. GAMES DEPLOYMENT AND EVALUATION

We deployed the games presented in chapter 3 in teams working on Intel Technology Poland projects in Gdansk. The research was conducted on three different teams, which are presented in Table 4.1. The teams differed in terms of the number of members, team maturity and whole group experience in scrum. We focused mainly on enhancing the creativity, involvement and communication of team members.

Table 4.1. Groups

| Group | Number of members | Project Age | Scrum experience |
|-------|-------------------|--|-------------------|
| A | 9 | Extremely mature (1.5 years old project) | typically 2 years |
| B | 3 | Immature (2 months old project) | at least 3 years |
| C | 8 | Mature (7 months old project) | at least 3 years |

The process of deploying a game in a team was as follows:

- Describe a particular game to the team
- Conduct the game with the team (as a team member or as a coach)
- Collect the data and discuss the results with the team
- Collect feedback from each team member who participated in a game

Each team was asked for a feedback after game deployment, by asking two different sets of questions. After the first sprint of deploying two games, the Starfish Game and the Speedboat game, the study group and the supervisor reflected on the results, using the research methodology, Action Research. We evaluated that the retrieved results should be improved, in order to retrieve more interesting and less generic characteristics. The first set was as presented below:

1. How would you evaluate the influence of this method on the results?
2. Were the discussions successful?
3. Were any unique features retrieved thanks to the game?
4. Are employees more willing to participate in the project when work is supported with this kind of game?
5. How would you evaluate the work associated with the games compared to standard procedures?
6. Were the results better using games instead of standard procedures?
7. Would you implement the game permanently instead of standard procedures?
8. How would you evaluate the preparation of the game?

The scale was between 1-5, where 5-meant excellent and 1 meant unsatisfactory.

Table 4.2 presents the quantity of feedback retrieved from each team, whilst playing the Speedboat and Starfish Games, in each team. By the time the first sprint was being deployed, team C was not cooperating with us.

A few different aspects were improved in the second set. Firstly and most importantly the questions in the new set are more specific and are focused on not the general feeling of the team, but on the concrete problems. Secondly, members were asked to answer questions about specific

Table 4.2. Number of feedbacks using an old set of questions

| Group | # of plays Speedboat | # of plays Starfish |
|--------|----------------------|---------------------|
| Team A | 1 | 1 |
| Team B | 1 | 1 |
| Team C | 0 | 0 |

characteristics, which are supposed to help the team develop and improve, such as communication, motivation and creativity. Lastly, the questions were more clear and easy to understand, than the old set and they were also presented as a statement on which the respondent was supposed to answer how much he or she agrees with it. The second set was presented as follows:

1. The influence of this method is greater than using standard procedures.
2. The game should be implemented permanently instead of standard procedures.
3. The game might complement standard procedures.
4. Thanks to the game the creativity of team members increased in the retrospective meeting.
5. Thanks to the game the involvement of team members increased in the retrospective meeting.
6. Thanks to the game the communication in the team increased in the retrospective meeting.
7. Thanks to the game the motivation of team members increased in the retrospective meeting.
8. The game is easy to understand and play.

Similarly as in the first set, we projected the answers as numbers using a scale of 1-5, with each number representing the following:

1. Strongly disagree.
2. Disagree
3. Undecided.
4. Agree.
5. Strongly agree.

The Table 4.2 presents the quantity of feedback retrieved from the teams, whilst playing the games listed in the sections 3.1.1, 3.1.2, 3.1.3, 3.1.5 and 3.1.6.

Table 4.3. Number of feedbacks using the new set of questions

| Group | # of plays Speedboat | # of plays Starfish | # of plays Glad/Mad/Sad | # of plays Mood & Improvements | # of plays 5L's |
|--------|----------------------|---------------------|-------------------------|--------------------------------|-----------------|
| Team A | 2 | 1 | 0 | 0 | 0 |
| Team B | 1 | 1 | 1 | 2 | 2 |
| Team C | 2 | 1 | 2 | 1 | 2 |

4.1. First iteration

4.1.1. Speedboat Game deployment

The game was implemented in the scrum team A. While we deployed the game, the team was ending its 31st sprint and was highly integrated. The estimated time of the game was 45 minutes, but the actual time turned out to be 1 hour and 30 minutes. The latter occurred due to unforeseen discussions which was partly caused by the game. The results of retrospective are presented

on Table 4.4 and were quite impressive as the team was able to retrieve 10 things that push them forward, 2 things that could have been done better and 4 “bad things”. Most of the team members were willing to participate in discussion and the main aspects covered were discovering new issues, new points of view and solutions. The second group on which we tested Speedboat implementation was team B (group of 3 people), which, at the time, had their first sprint and retrospective together. The estimated time of the game for this group was 1 hour and by the end, the team was able to retrieve 14 things that push them forward, 7 things that bring them down and 4 things that may cause the project to fail. The team integrated by sharing thoughts with each other. They were asked the same questions as group A and the result was satisfying.

Table 4.4. Results of the Speedboat Game

| Group | The cloud | The rocks | The anchor | Time | Comment |
|--------|-----------|-----------|------------|-------------------|-------------|
| Team A | 10 | 2 | 4 | 1 hour 30 minutes | 31st sprint |
| Team B | 14 | 7 | 4 | 45 minutes | 1st sprint |
| Team C | N/A | N/A | N/A | N/A | N/A |

The chart on the Figure 4.1 represents collected results of this survey using old set of questions.

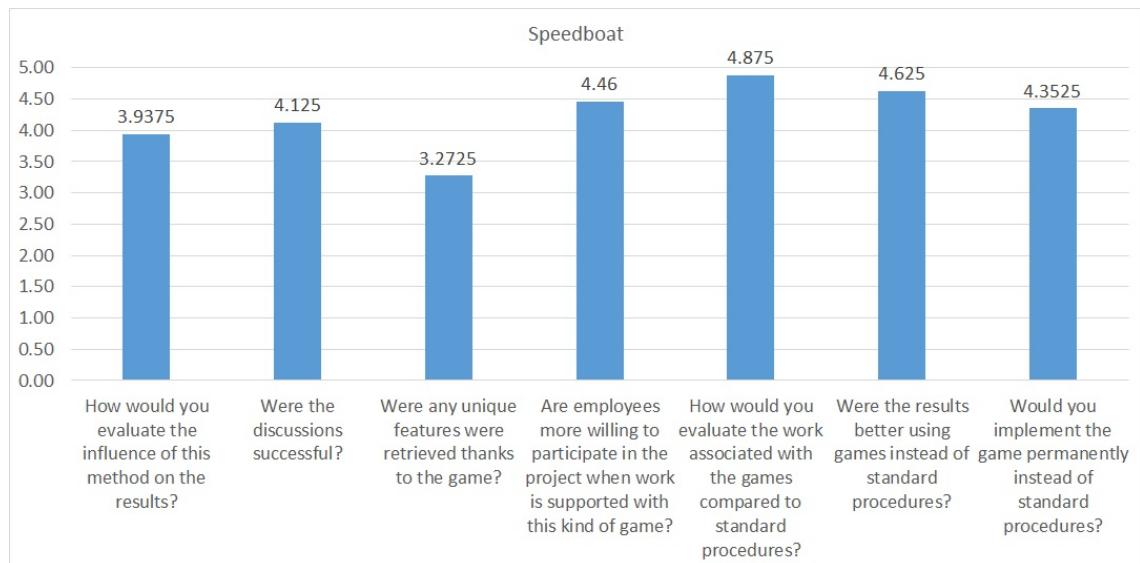


Fig. 4.1. A bar graph depicting the results when deploying the Speedboat game using the old set of questions

As we can see on Figure 4.1, the results are quite similar to each other. Also, the question "How would you evaluate the preparation of the game?" was excluded from the chart, because in the case of this study, it did not give any interesting characteristics. In this case, we are unable to actually retrieve any information, but what we do know is that the participants of the experiment agreed that the game had better results than the standard procedure. What is more, they thought that the discussions were successful. Not only were the employees willing to play the game but the work associated with the game had a positive impact on the team members and most importantly they said they would strongly recommend to permanently implement the game instead of using the standard procedure. On the other hand, some claimed it was hard to say whether they gained any unique features thanks to the Speed boat game.

4.1.2. Starfish Game deployment

The "Starfish" game was also implemented in Team A (while they were evaluating theirs 32nd sprint) and Team B (whilst evaluating their 2nd sprint) using the old set of questions. The results for both teams are presented on Table 4.5. The first team playing this game were able to retrieve 8 things that should be started in order to make the sprint successful and the work easy, 1 thing that they should immediately stop doing, 4 things that they want to continue doing, because they are verified and good for the team and the project. They also found 4 things that should be made less and 3 things that should be done more. Team B retrieved 5 "start doing", 2 "stop doing", 2 "continue", 4 "less of" and 6 "more of". The game changed the perspective, so the participants were more creative and interested in the game.

Table 4.5. Results of the Starfish Game

| Group | Start doing | Stop doing | Continue | Less of | Stop doing | Time | Comment |
|--------|-------------|------------|----------|---------|------------|-------------------|-------------|
| Team A | 8 | 1 | 4 | 4 | 3 | 1 hour 45 minutes | 32nd sprint |
| Team B | 5 | 2 | 2 | 4 | 6 | 1 hour 10 minutes | 2nd sprint |
| Team C | N/A | N/A | N/A | N/A | N/A | N/A | N/A |

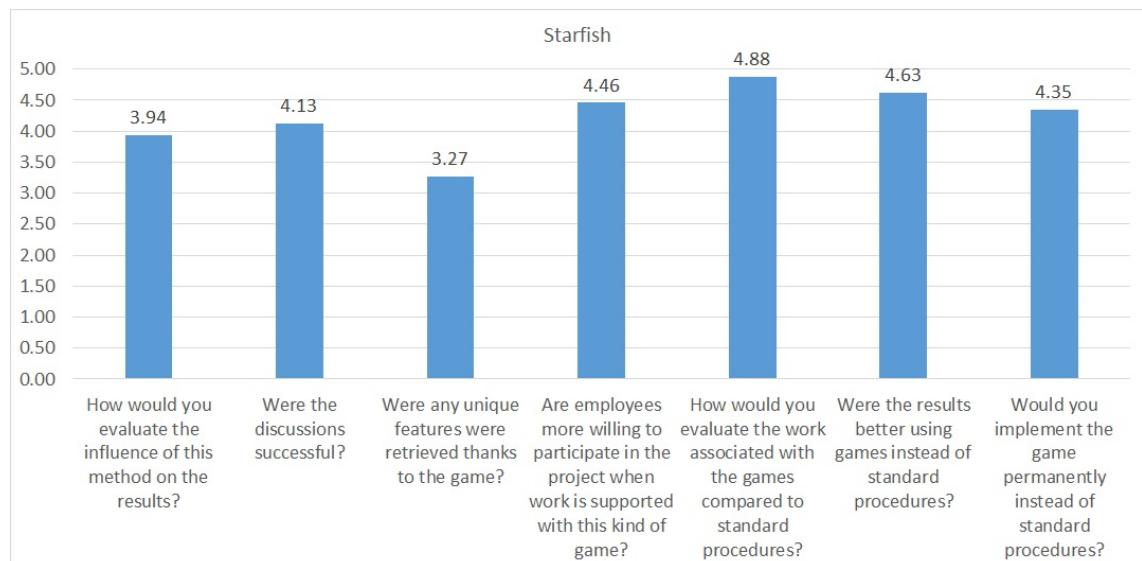


Fig. 4.2. A bar graph depicting the results when deploying the Starfish game using the old set of questions

Figure 4.2 presents the results of the "Starfish" deployment. As we can see, similarly to the "Speedboat" game results using the old question set, it is hard to retrieve useful data from this chart. The uniqueness of the features is hard to establish as in "Speedboat" deployment and the other results are more or less the same.

4.1.3. Discussion

The chart on Figure 4.3 presents a comparison of deploying the "Speedboat" and "Starfish" game. We can see that the results do not differ that much - this was the catalyst of actually changing the set of questions to retrieve more valuable characteristics. As the result we decided to ask more specific questions to retrieve better features.

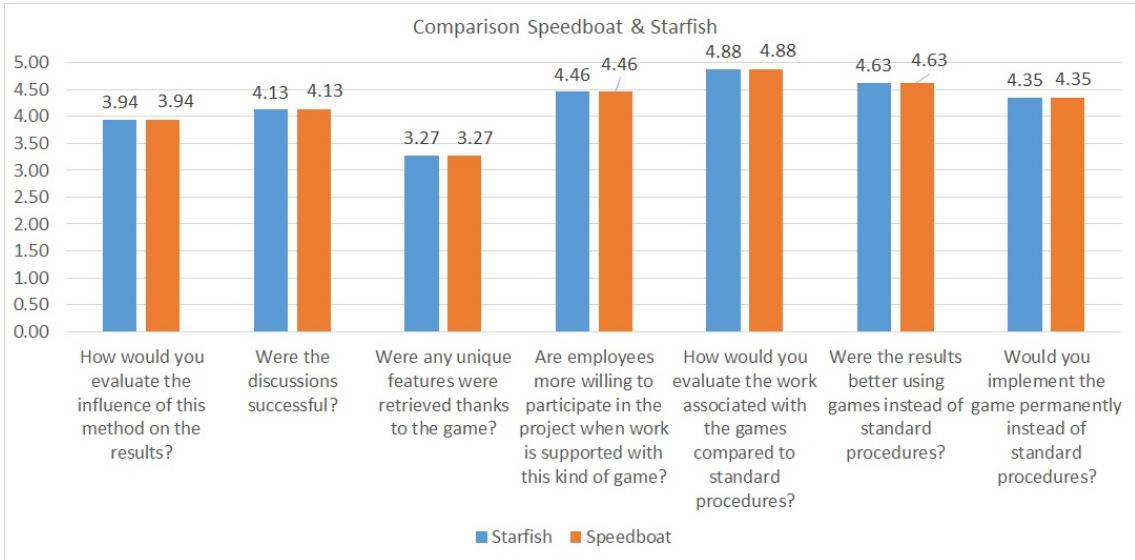


Fig. 4.3. A comparison of deploying starfish and speedboat games using the old set of questions

4.2. Second iteration

4.2.1. Speedboat Game deployment

This test was deployed multiple times on every team that has been listed in Table 4.1. Using this game we are able to see that through a change of perspective, by not using standard procedures, we are able to look at things differently, realise what is pushing our boat forward, what might crush us and what is pulling us down. Thanks to the game, we have observed that the teams were more involved, motivated and eager to participate in retrospective game. Table 4.6 presents the results that were retrieved using the Speedboat technique by the performing the retrospective. In group B, the retrospective was performed for 45 minutes and in team C for 50 minutes. While team A was in their 33rd sprint, they were able to find 8 good things, 5 bad things and 3 to improve. Two sprints later, the results were as follows: 16 good things, 3 bad things and 4 things to improve. The team played the game for 1 hour 15 minutes on average. Team B played the game just once, while having a 3rd sprint, and it took them 45 minutes to successfully get 8 good things, 5 bad things and 6 things that needed improvement. The last team, team C, whilst having their 19th sprint, played the game once for 55 minutes and they retrieved 17 good things, 8 bad things and 5 things to improve.

Table 4.6. Results of the Speedboat Game

| Group | The cloud | The rocks | The anchor | Time | Comment |
|--------|-----------|-----------|------------|-------------------|-------------|
| Team A | 8 | 5 | 3 | 1 hour 20 minutes | 33rd sprint |
| Team A | 16 | 3 | 4 | 1 hour 10 minutes | 35th sprint |
| Team B | 8 | 5 | 6 | 45 minutes | 3rd sprint |
| Team C | 17 | 8 | 5 | 50 minutes | 19th sprint |

Figure 4.4 presents the "Speedboat" game deployment. Every single game described in this chapter was documented with a picture and the feedback from the team.

Analysing the results from Figure 4.5, we were able to retrieve that through playing the game, the

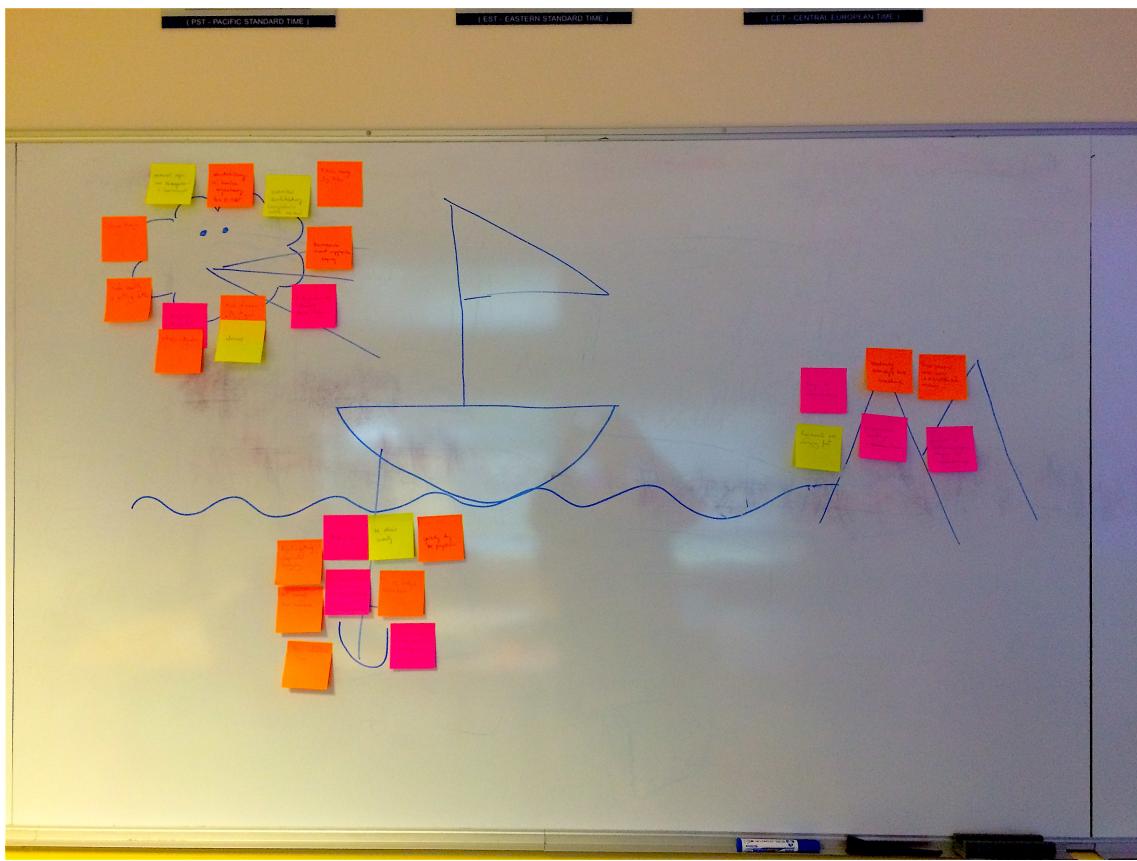


Fig. 4.4. Speedboat deployment

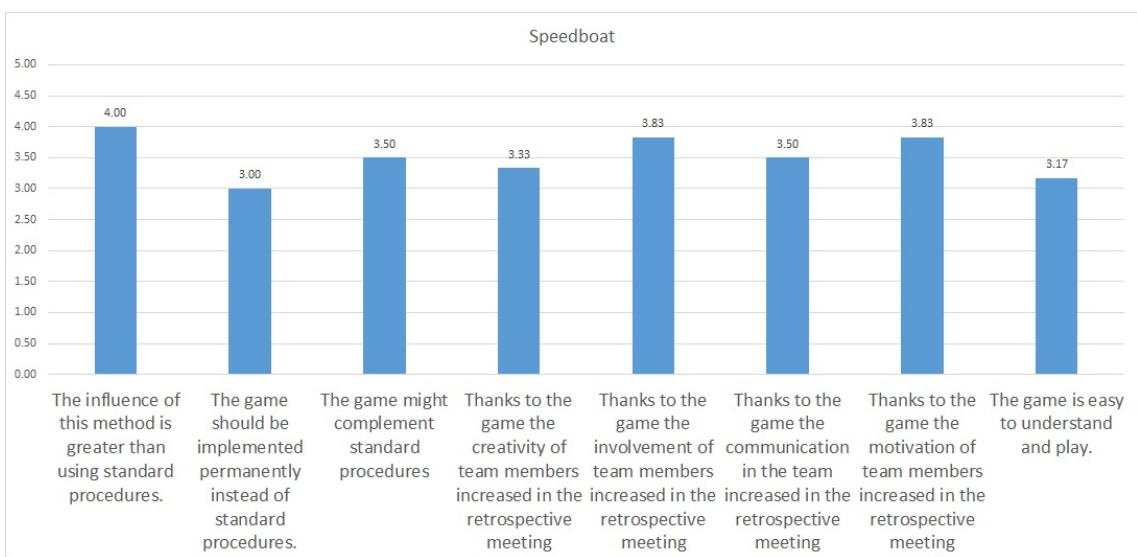


Fig. 4.5. A bar graph depicting the results when deploying the Speedboat game using the new set of questions

team was seeing a better influence on the retrospective meeting, but surprisingly, it was hard for them to decide whether they would like to play this game permanently during the next meetings. The probable cause of the latter is that even though the game is fun and useful, it should not be used on every retrospective. This is due the monotony nature, which we would like to reduce. They could not also decide definitely, but were more convinced than in the previous question, that this game might complement standard procedures. Regarding the characteristics of the game, they were almost definitely convinced that they agree on statements about motivation and involvement of participants while playing "Speedboat". However, they were less convinced about communication being increased and they were also not convinced about the team members' creativity. Moreover, what is actually surprising is that they could not decide on the level of difficulty in terms of understanding the game.

4.2.2. Starfish Game deployment

The "Starfish" game was a big surprise for all the teams due to a drastic change in perspective. The five fields which had to be filled created multiple discussions. The game was implemented in teams A, B and C. None of the participants in these teams were familiar with this game, so because of the complexity in the preparation of the game, explaining the rules and giving the participants the time to familiarize with the new approach, this made the meeting last much longer. The data presented on Table 4.7 shows the results retrieved by the teams in the retrospective meeting using the described game used in this subsection. On average, team A finished the meeting after 1 hour 45minutes. They played the game after their 36th sprint and the game resulted in 9 things that they should start doing, 3 stop doing, 3 continue, 5 less and 2 more. Team B was in their 4th sprint and finished the game in 45minutes. They attained 6 things that may increase their effectiveness and are good for the team, and as a result they should start doing them, 1 that is stopping them and they should dispose of it, 7 things that are being already done in the team and they want to continue performing them, 6 things that the team should do less and 1 thing that should be done more often. While having the 21st sprint, team C retrieved 6 start doing things, 4 stop doing, 2 continue, 3 less of and 2 more of. The game lingered for 1hour 10 minutes. It was an interesting experience to work with teams which are so involved in the retrospective meeting.

Table 4.7. Results of the Starfish Game

| Group | Start doing | Stop doing | Continue | Less of | More of | Time | Comment |
|--------|-------------|------------|----------|---------|---------|-------------------|-------------|
| Team A | 9 | 3 | 3 | 5 | 2 | 1 hour 45 minutes | 36th sprint |
| Team B | 6 | 1 | 7 | 6 | 1 | 45 minutes | 4th sprint |
| Team C | 6 | 4 | 2 | 3 | 2 | 1 hour 10 minutes | 21st sprint |

Figure 4.6 presents live deployment of the "Starfish" game.

The results in Figure 4.7 represent the feedback retrieved from the team members listed in Table 4.1 upon playing the "Starfish" game. The participants decided that they agree that the game should be permanently implemented in retrospective meetings. Moreover, based on their opinion, the results using this approach is greater than using standard procedures. Regarding the ease of the game, after preparation and discussing the rules, they decided that even though the game is complex, they understood it clearly after being presented with a proper introduction. A major-

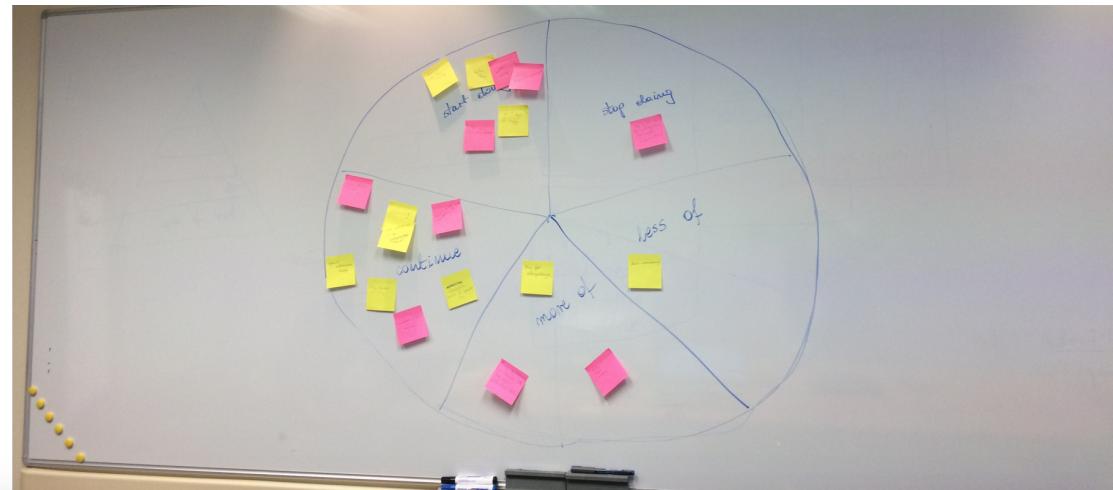


Fig. 4.6. Starfish deployment

ity of the participants answered that involvement is a characteristic that could be retrieved from the "Starfish" game. Regarding the motivation and creativity on the other hand, the features that can also be increased using this approach, probably because of the innovation of this game. The groups would rather use this game permanently than as a supplement of standard procedures. Most of the team members were not convinced though whether communication was actually improved through playing this game.

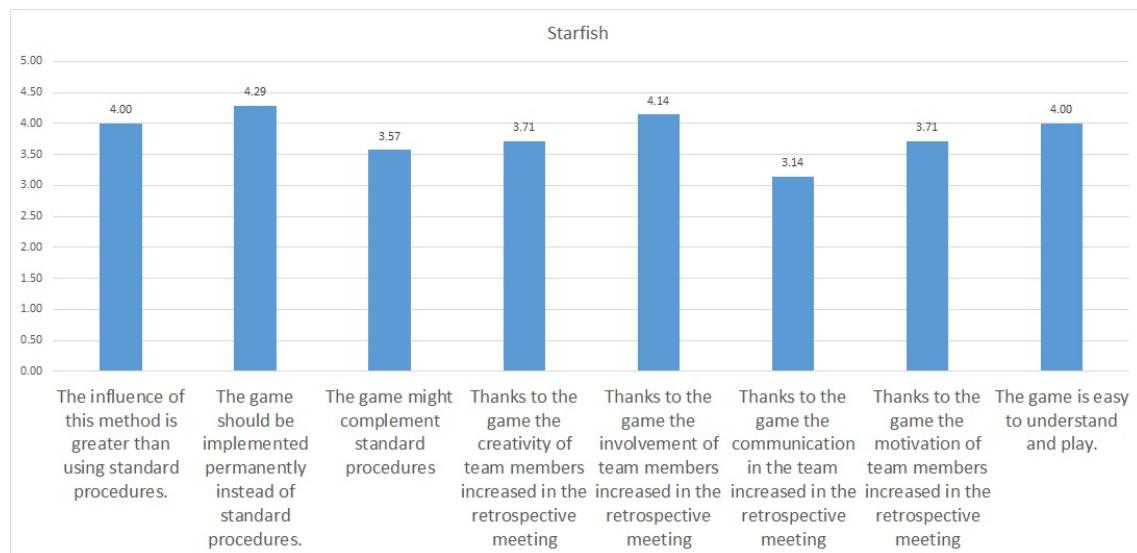


Fig. 4.7. A bar graph depicting the results when deploying the Starfish game using the new set of questions

4.2.3. Glad/Mad/Sad Game deployment

Glad/Mad/Sad, also known as Happy/Sad/Angry game, was deployed in groups B and C, but because internal changes cooperation with team A was not possible anymore. The Glad/Mad/Sad was not a surprise for the teams. The approach of this game is very similar to the standard one, but if you have a choice, it is better to use the Glad/Mad/Sad technique rather than the standard Good Things, Bad Things, Things to Improve technique in order to diversify the retrospective meeting. Both teams played once for approximately 45 minutes with team B having their 5th sprint, and

team C their 20th sprint. Table 4.8 shows the results which were retrieved by the teams. Team A was able to get 8 things that made them glad, 7 mad and 2 sad and team C was able to get 10, 9 and 14 respectively.

Table 4.8. Results of the Glad, Mad, Sad Game

| Group | Glad | Mad | Sad | Time | Comment |
|--------|------|-----|-----|------------|-------------|
| Team A | N/A | N/A | N/A | N/A | N/A |
| Team B | 8 | 7 | 2 | 40 minutes | 5th sprint |
| Team C | 10 | 9 | 14 | 50 minutes | 20th sprint |

On the Figure 4.8 we can see the deployment of the "Glad, Mad, Sad" Game in Intel Technology Poland in Gdańsk.



Fig. 4.8. Glad, Mad, Sad deployment

Figure 4.9 presents the results of the Glad/Mad/Sad game deployment. The highest result was retrieved in the third and eighth question and indicates that participants agree on the fact that this technique might complement standard procedures and is easy to understand. The members of the experiment are also rather convinced that the game had increased the teams' creativity, which proves a previously presented thesis that it is better to use the Glad/Mad/Sad approach, rather than the standard technique. In terms of the influence of game on improving the results, the participants were undecided on whether the game should be implemented permanently and whether the involvement of team members is increased in the retrospective meeting. The attendees of the experiments agreed that communication and motivation of team members was not influenced by this technique.

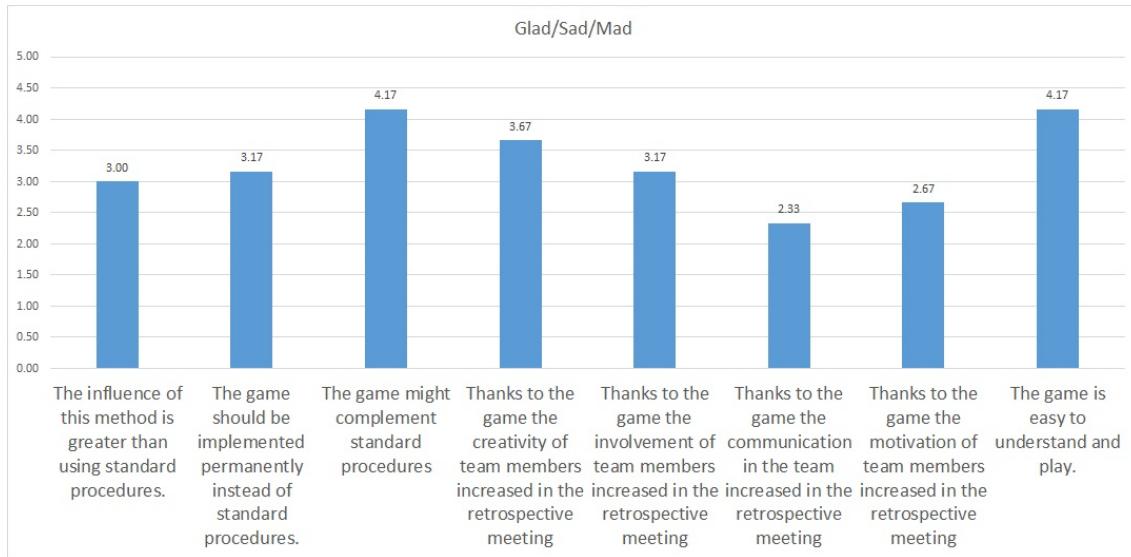


Fig. 4.9. A bar graph depicting the results when deploying the Glad/Mad/Sad game using the new set of questions

4.2.4. Mood and improvements Game deployment

Figure 4.10 presents the deployment of the "Mood and improvements" game that have been created in collaboration with Intel Technology Teams. We merged the "Glad, Mad, Sad" game with two fields to indicate ideas for improvements and appreciations amongst team members.



Fig. 4.10. Mood and improvements game deployment

Table 4.9 presents the results of the Mood and Improvements Game implementation. On average, team B played the game twice for 45 minutes and in their 6th sprint retrieved 6 things that made them glad, 3 mad and 6 sad. They also had 3 ideas for improvement and 4 appreciations. In the 9th sprint, they found 13 things that made them glad, 2 mad and 2 sad, 3 ideas for improvement and 1 appreciation. The second team that participated in the research was team C and in their 22nd sprint, they were able to get 12 good things, 16 bad things, 4 things to improve, 4 ideas for enhancement and 1 appreciation. This took them 50 minutes to finish the game.

Table 4.9. Results of the Mood and Improvements Game

| Group | Glad | Mad | Sad | Ideas for improvement | The Appreciations | Time | Comments |
|--------|------|-----|-----|-----------------------|-------------------|------------|-------------|
| Team A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| Team B | 6 | 3 | 6 | 4 | 4 | 35 minutes | 6th sprint |
| Team B | 13 | 2 | 2 | 3 | 1 | 55 minutes | 9th sprint |
| Team C | 12 | 16 | 4 | 4 | 1 | 50 minutes | 22nd sprint |

Figure 4.11 presents the results of feedback received from the participants. The chart indicates that the game was easy to understand and might complement standard procedures. The participants also agreed on the fact that the game increases creativity, communication and involvement. Moreover, they agreed that the game had a better influence on the results compared to standard procedures and are willing to permanently implement it into the retrospective meeting. The only characteristic that attendants of the experiment cannot decide on is, whether the technique increases motivation.

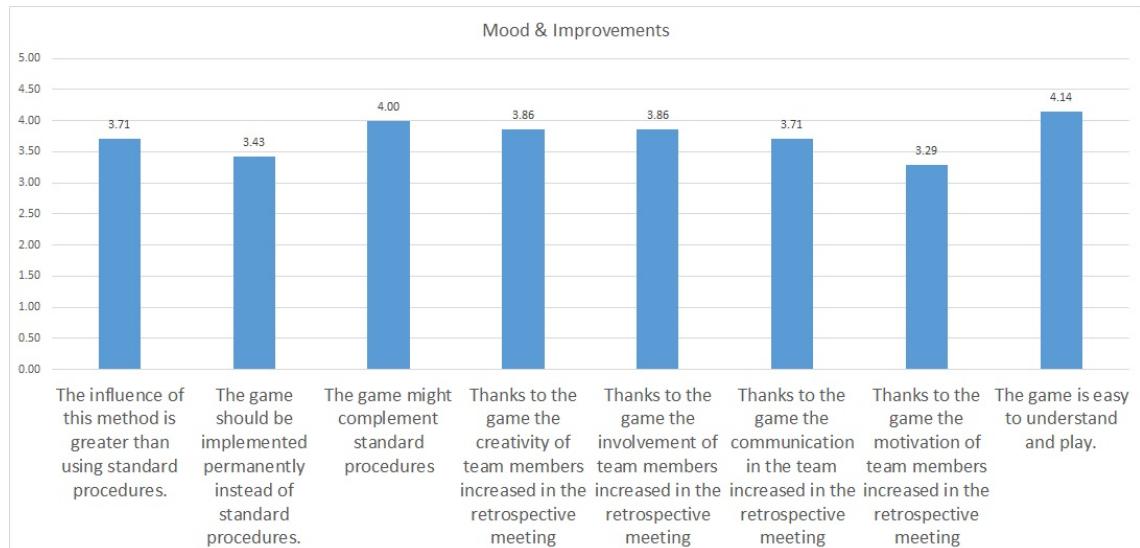


Fig. 4.11. A bar graph depicting the results when deploying the Mood and Improvements game using the new set of questions

4.2.5. 5L's Game deployment

The "5L: Liked, Learned, Lacked, Longed for, Loathed" has been created in collaboration with Intel Technology Poland teams, using as a archetype game "4 L's". This approach changes the perspective and shows the participants a different point of view. It also enables them to learn

more about other team members. For example, this could include what they learned and what was important for them in the past iteration. There are two ways of playing the game that were deployed in the teams. The first one is shown on the areas arranged in a circle in Figure 4.12.



Fig. 4.12. 5 L's game deployment

The second one is presented on Figure 4.13, we arranged the fields in columns. Furthermore, we assigned colored sticky-notes to particular areas this way, we were able to see a clearer picture, as the sticky notes appeared more organized.

Using the 5L's game, the participants were involved in the retrospective meeting and particularly when using the technique with colors, they even become competitive. For example, they would compete over the number of pink sticky-notes and in some cases over who had the most diverse pile of paper. In teams, B and C, the game was deployed twice and was played responsively for 50 and 65 minutes on average respectively, as presented on Table 4.10. After finishing their 7th sprint, team B retrieved 18 things they liked, 4 learned, 2 lacked, 2 longed for and 6 they loathed, and upon the 8th iteration they discovered 8 likes, 6 learns, 7 lacks, 3 things that they longed for and 8 that they loathed. In contrast, team C retrieved 15 things they liked, 4 learned, 3 lacked, 7 longed for and 10 loathing in the 24th iteration and 14, 4, 2, 5 and 6 on the 25th iteration, respectively.

Figure 4.14 presents the results retrieved from the participants' feedback. The results were positively surprising, because all the bars are equal or above the "Agree" statement. The participants particularly felt that their involvement was increased by this technique. What is more, they also agreed on the fact that communication and motivation was improved by this technique and they also that the game should complement the standard procedure in the retrospective meeting. The

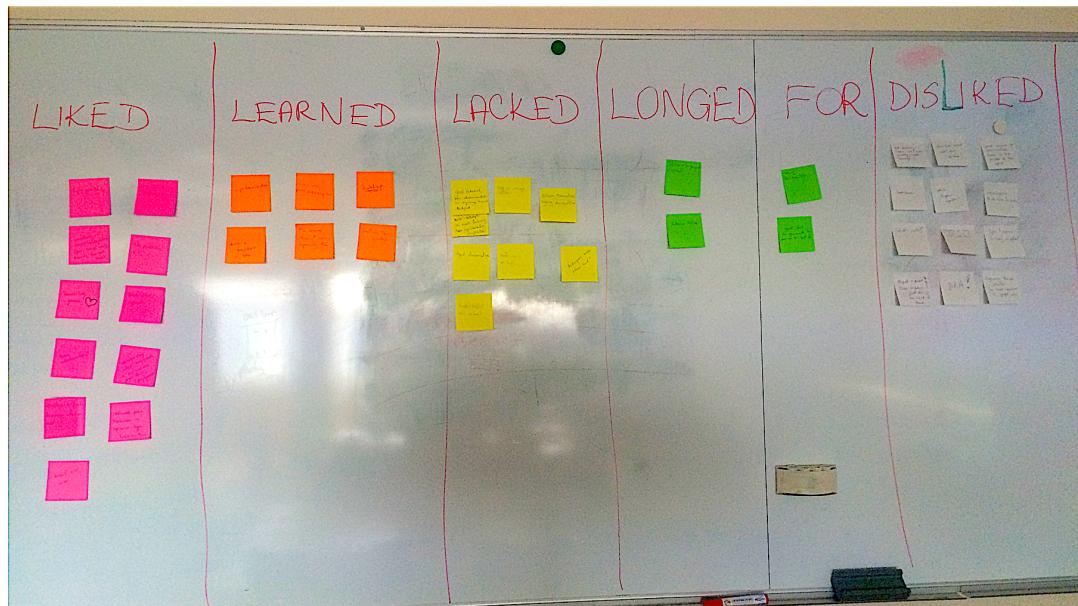


Fig. 4.13. A different representation of the 5 L's game deployment

Table 4.10. Results of the 5L's Game

| Group | Liked | Learned | Lacked | Longed for | Loathing | Time | Comment |
|--------|-------|---------|--------|------------|----------|------------------|----------------|
| Team A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| Team B | 18 | 4 | 2 | 2 | 6 | 55 minutes | 7th iteration |
| Team B | 8 | 6 | 7 | 3 | 8 | 45 minutes | 8th iteration |
| Team C | 15 | 4 | 3 | 7 | 10 | 50 minutes | 24th iteration |
| Team C | 14 | 4 | 2 | 5 | 6 | 1hour 20 minutes | 25th iteration |

lowest result, was still very high overall, because the participants assigned an "Agree" statement, regarding whether the influence of the 5L's method is better than using the standard approach and whether they would also permanently implement the 5L's to the retrospective meeting. Moreover, they discovered that there was an increase in creativity and they thought that the game was easy to understand.

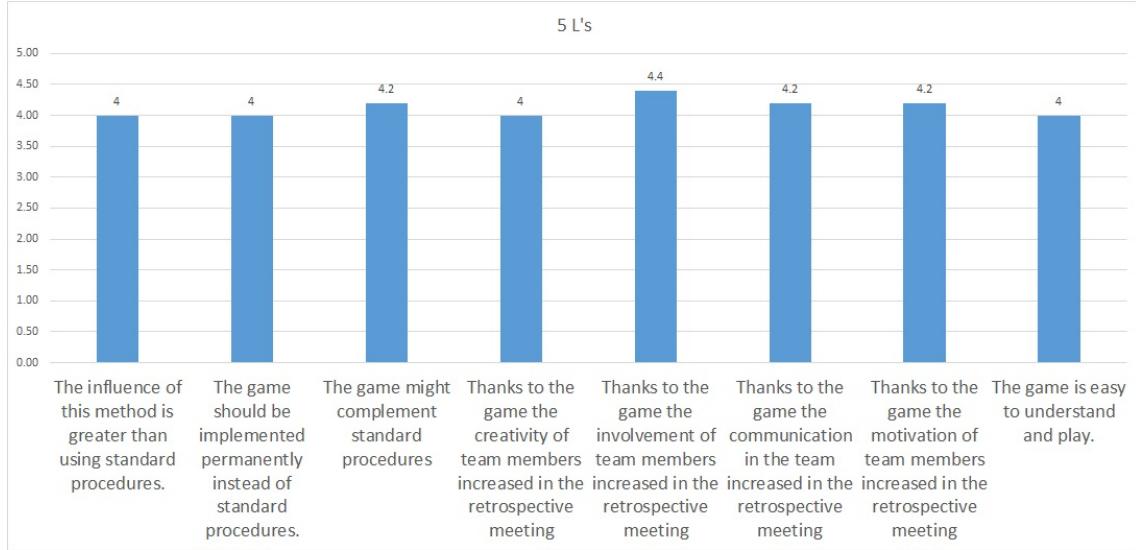


Fig. 4.14. Results of deploying 5 L's game using new set of questions

4.2.6. Discussion

In section Second iteration, a number of games were deployed in Intel Technology Poland in Gdansk in collaboration with teams listed in Table 4.1. Thanks to Intel, these two games were either created or improved. Moreover, we cooperated with the Certified Scrum Master, Grzegorz Regliński, who helped us properly implement the games into the teams.

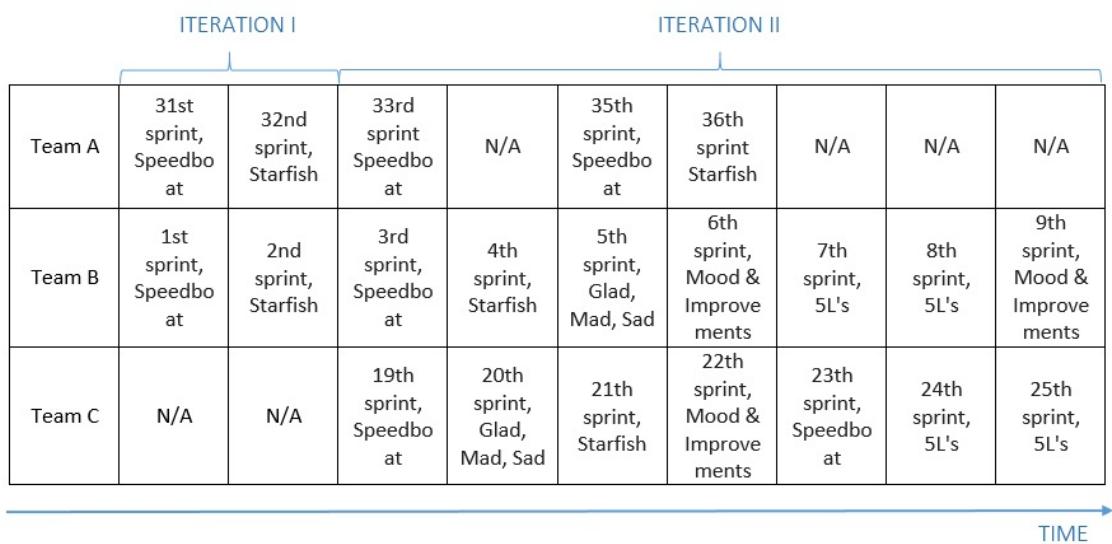


Fig. 4.15. Overall timeline of the games

The overall description of the deployment is presented in Figure 4.15. The figure points out a given

time when the game was introduced, which team whilst performing a sprint executed a particular game and whether it was the first or second sprint of the question set. A summary of the second iteration of game deployment is as follows:

1. The influence of this method is greater than using standard procedures.
 - The participants agreed that the following games had a positive impact on the results: Speedboat, 5L's and Starfish.
 - Although not as strongly, but still closer to the "Agree" statement, the participants thought that Mood and Improvements influenced the outcome of retrospective compared to standard procedures.
 - It was hard for the members of the experiment to decide in the case of Glad/Mad/Sad approach.
 - The best games were as follows: Speedboat, 5L's and Starfish.
 - The worst game was: Glad/Mad/Sad.
2. The game should be implemented permanently instead of standard procedures.
 - The participants definitely agreed on the idea of implementing Starfish Game permanently.
 - They also had a positive feeling in terms of 5L's game in this field.
 - The attendants of the experiment rather bowed to the statement "Agree" in the case of Mood and Improvements.
 - It was hard for them to decide in the case of Glad/Mad/Sad.
 - The best game was: Starfish.
 - The worst game was: Speedboat.
3. The game might complement standard procedures.
 - In the case of complementing the standard procedures with the introduced new approaches, the Glad/Mad/Sad and 5L's had almost the same results.
 - Directly assigning the "Agree" statement was mostly retrieved in the Mood and Improvements approach.
 - In terms of Starfish, participants rather bowed to the "Agree" statement in the case of complementing the standard procedure with the presented technique.
 - In this category, Speedboat has the worse result as it was hard for the participants to decide whether the approach should complement the standard procedures.
 - The best game was: 5L's
 - The worst game was: Speedboat
4. Thanks to the game the creativity of team members increased in the retrospective meeting.
 - This statement achieved the exact value of "Agree" statement and what is more, the best result in this category was assigned to 5L's technique.
 - The game creativity in Starfish, Mood and Improvements and Glad/Mad/Sad games is also influenced, but the results are below the "Agree" statement, but heavily above the "Hard to say" opinion.
 - The worse result was assigned to Speedboat, for the participants it was hard to determine if the influence in the creativity happened.
 - The best game was: 5L's
 - The worst game was: Speedboat
5. Thanks to the game the involvement of team members increased in the retrospective meeting.
 - Based on data collected from the teams feedback from the teams ,the 5L's game is the best in terms of the involvement of participants'. The results were only slightly worse for

the Starfish technique.

- Both Speedboat and Mood and Improvements are below the "Agree" statement but significantly above the "Undecided" opinion, so it is reasonable to claim that they both lead to an increase in involvement.
- According to the opinions of the team members, the Glad/Mad/Sad approach has no impact on the involvement.
- The best game was: 5L's
- The worst game was: Glad/Mad/Sad

6. Thanks to the game the communication in the team increased in the retrospective meeting.

- The 5L's game achieved the best results in terms of increasing the communication in the team.
- The "Mood and Improvements" and "Speedboat" games are slightly below the "Agree" statement. Moreover, we can assume a positive influence in on the team communication using these approaches.
- Regarding the Starfish technique, it has been decided by the attendants of the research that is hard to indicate whether communication has been improved.
- In case of the Glad/Mad/Sad game, it has been established that it does not have any influence on increasing communication.
- The best game was: 5L's
- The worst game was: Glad/Mad/Sad

7. Thanks to the game the motivation of team members increased in the retrospective meeting.

- The team was motivated and eager to participate particularly when using the 5L's method.
- In the case of Starfish and Speedboat it has been decided that, even though the bar is slightly below the "Agree" statement, we can still establish that these methods increase the teams' motivation.
- For the Mood & Improvements game, the participants determined that it is hard for them to decide whether motivation was increased using this technique.
- In the case of the Glad/Mad/Sad game, motivating the team to be involved in the retrospective. The game rather failed, but it is closer to the "Undecided" statement, rather than the "Disagree" opinion.
- The best game was: 5L's
- The worst game was: Glad/Mad/Sad

8. The game is easy to understand and play.

- In this category most of the games are above the "Agree" statement, which gives rise to the conclusion that the Starfish, 5L's, Mood & Improvements and Glad/Mad/Sad games were easy to understand by the attendants.
- Only in one out of five approaches was it hard for the participants to define whether it was easy or difficult to understand - Speedboat.
- The best game was: Glad/Mad/Sad
- The worst game was: Speedboat

To conclude, the feedback obtained from the participants as presented Table 4.11 indicates, how many times a game transpired to be the best and the worse. In summary, the best game for all the scenarios, on the basis of the attendants' experiment opinion is 5L's. As indicated on Table 4.11, it is the best game in six out of eight categories. The worse technique turned out to be the Glad/Mad/Sad, probably because of its major similarity to the standard procedures. This does not suggest that the Glad/Mad/Sad approach should not be used or that the 5L's method should

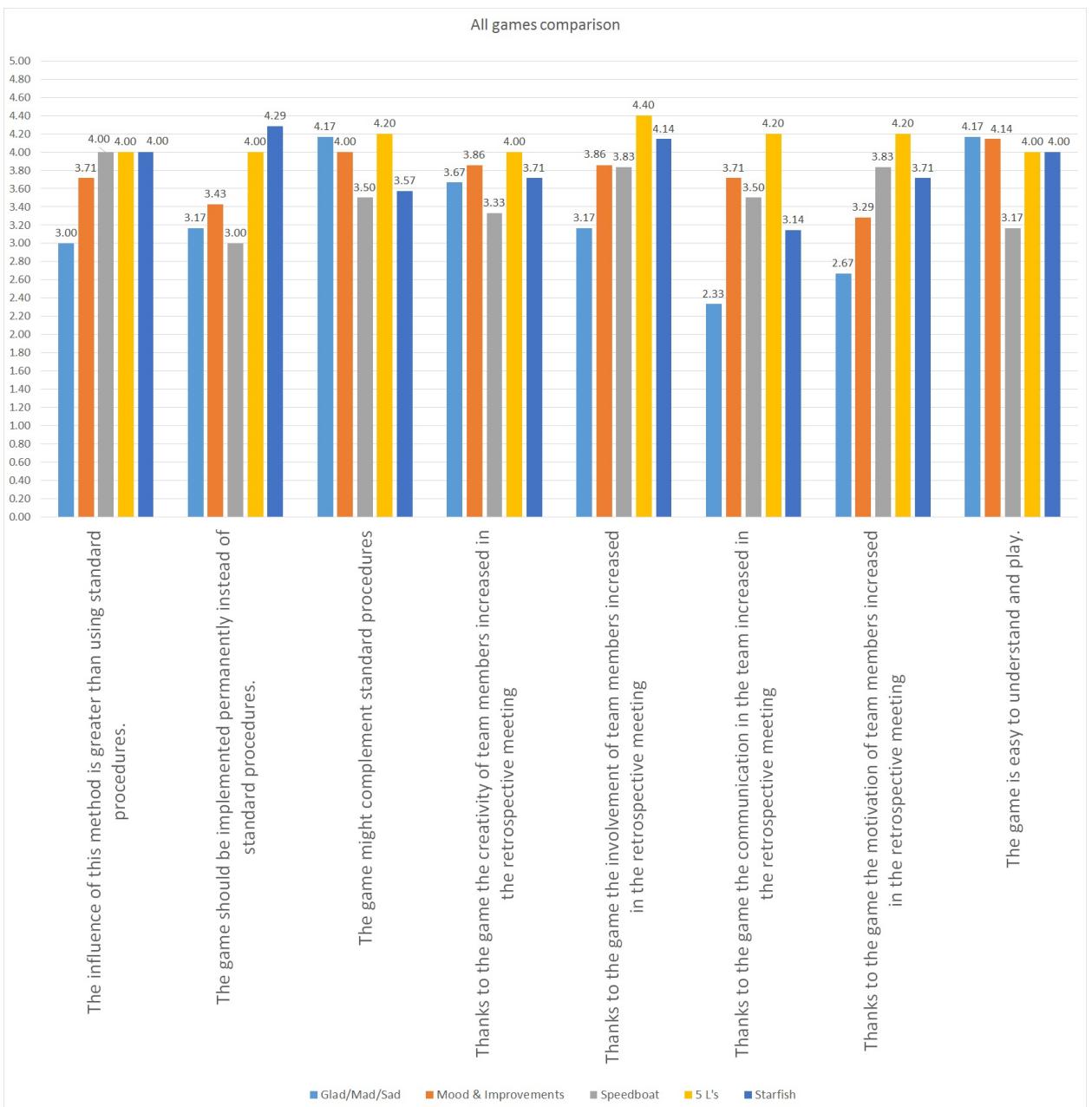


Fig. 4.16. A comparison of deploying all the games using the new set of questions

be used for all the retrospective meetings. Based on our research, it is advised that the way the retrospective meeting is led should vary as different techniques being deployed provides more valuable results.

Table 4.11. Overall results of the games from the feedback

| Game | # of best results | # of worse results |
|---------------------|-------------------|--------------------|
| Speedboat | 1 | 4 |
| Starfish | 2 | None |
| Glad/Mad/Sad | 1 | 4 |
| 5L's | 6 | None |
| Mood & Improvements | None | None |

The additional Figure 4.17 presents a table of grouped results. The green color indicates whether the threshold is equal to 4 or above "Agree" statement. The cells appointed with a red color denote a result value of less than 3 which indicates a "Disagree" statement. The yellow color presents results above and equal to a 3.5 threshold, which is also classed as "Agree" and the blue which once which are above 3 and below 3.5 threshold is classified as a "Undecided" opinion. As presented in the table, the 5L's is the only game, that achieved all the cells marked as above 4. All but beside one game, Starfish, retrieved the most positive results with a threshold higher than 3.5. On the other hand, Glad/Mad/Sad is the only approach in which a threshold less than 3 has been obtained. In terms of the other games, they were found to give rise to successful outcomes in a particular situation and as a result, should be used as a solution to a dedicated problem. For example, for problems regarding communication in the team, you, the Scrum Master might try the "Mood & Improvements" approach. We have observed that the major number of results are yellow and green, which confirms the thesis, that the games increase the number of positive factors in the retrospective meeting. The blue and the red cells are in a minority. What can also be classified as a success is the fact that in the "All games together" column, all the values are yellow colored, which means that in general, the games increase the results. In the "All questions summary" column, the situation is less satisfying, especially for "Glad/Mad/Sad", but in the majority of these games, a threshold higher than 3.5 was still reached.

| | Glad/Mad/Sad | Mood and Improvements | Speedboat | 5L | Starfish | All the games together |
|---|--------------|-----------------------|-----------|------|----------|------------------------|
| The influence of this method is greater than using standard procedures. | 3.00 | 3.71 | 4.00 | 4.00 | 4.00 | 3.74 |
| The game should be implemented permanently instead of standard procedures. | 3.17 | 3.43 | 3.00 | 4.00 | 4.29 | 3.58 |
| The game might complement standard procedures. | 4.17 | 4.00 | 3.50 | 4.20 | 3.57 | 3.89 |
| Thanks to the game, the creativity of team members increased in the retrospective meeting. | 3.67 | 3.86 | 3.33 | 4.00 | 3.71 | 3.71 |
| Thanks to the game, the involvement of team members increased in the retrospective meeting. | 3.17 | 3.86 | 3.83 | 4.40 | 4.14 | 3.88 |
| Thanks to the game, the communication in the team increased in the retrospective meeting. | 2.33 | 3.71 | 3.50 | 4.20 | 3.14 | 3.38 |
| Thanks to the game, the motivation of team members increased in the retrospective meeting. | 2.67 | 3.29 | 3.83 | 4.20 | 3.71 | 3.54 |
| The game is easy to understand and play. | 4.17 | 4.14 | 3.17 | 4.00 | 4.00 | 3.90 |
| All the questions summary | 3.29 | 3.75 | 3.52 | 4.13 | 3.82 | |

Fig. 4.17. A table showing grouped results of all the games

5. RETROSPECTIVE ANALYZER SYSTEM

This chapter is focused on the complementary software that has been added to this master thesis. Building on the approaches introduced chapter 3 and the ones that were actually implemented in Intel Technology Poland presented in the chapter chapter 4, we created a web service, which helps scrum masters to choose a game suitable for the team. The Retrospective analyzer system was created especially for Scrum Masters, Team Leaders and Managers in order to help them find a suitable game for the retrospective meeting. Leading a team is a difficult assignment, but thanks to Scrum, this job is being simplified. In order to make it more easy, we created a system that chooses a retrospective game for the team. What is more, we created a base of verified games with descriptions, in order to present a user with the different retrospective techniques our system provides.

5.1. System overview

The Retrospective analyzer application is a web service. On Figure 5.1, the welcome page is presented with an introduction what the system is offering.

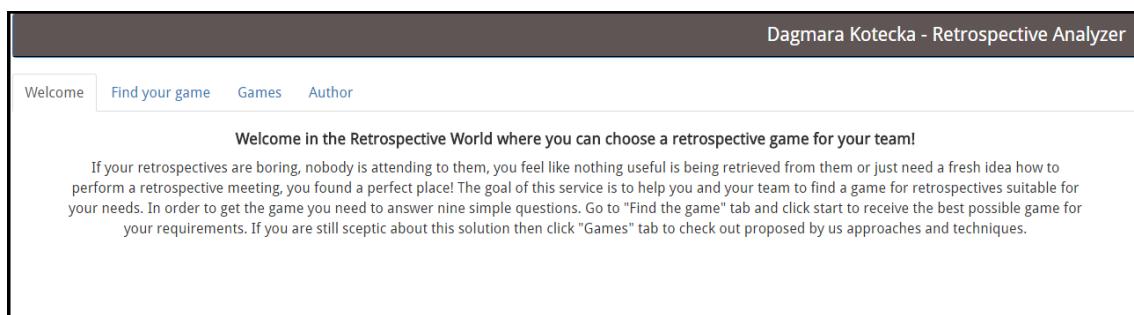


Fig. 5.1. Welcome page screenshot

The Games tab is presented on the Figure 5.2 and it contains all the games included in this system.

Another subpage and the most important one is the "Find The Game" tab, where the main functionality of the system is introduced. From this subpage, after clicking the start button shown on Figure 5.3, the user is redirected to the page with questions presented on Figure 5.4. The question page corresponds to the main issues which may occur in the Scrum team. Based on users answers, the system chooses the most suitable retrospective approach.

The questions in the set were created based on Figure 5.5, which shows how many points the game reached in a particular factor using a scale from 1 to 10, where 1 is the lowest value and 10 the highest. The table has been created in association with Grzegorz Reglinski, the Intel Technology Scrum Master. We evaluated the table using feedback from the teams, our scrum experience and on the data collected from the survey.

Overall, "360-degrees of appreciation" was granted the highest sum whereas "Glad, Mad, Sad" received the lowest amount. The rest of the games' summary points fluctuates around a value of 50.

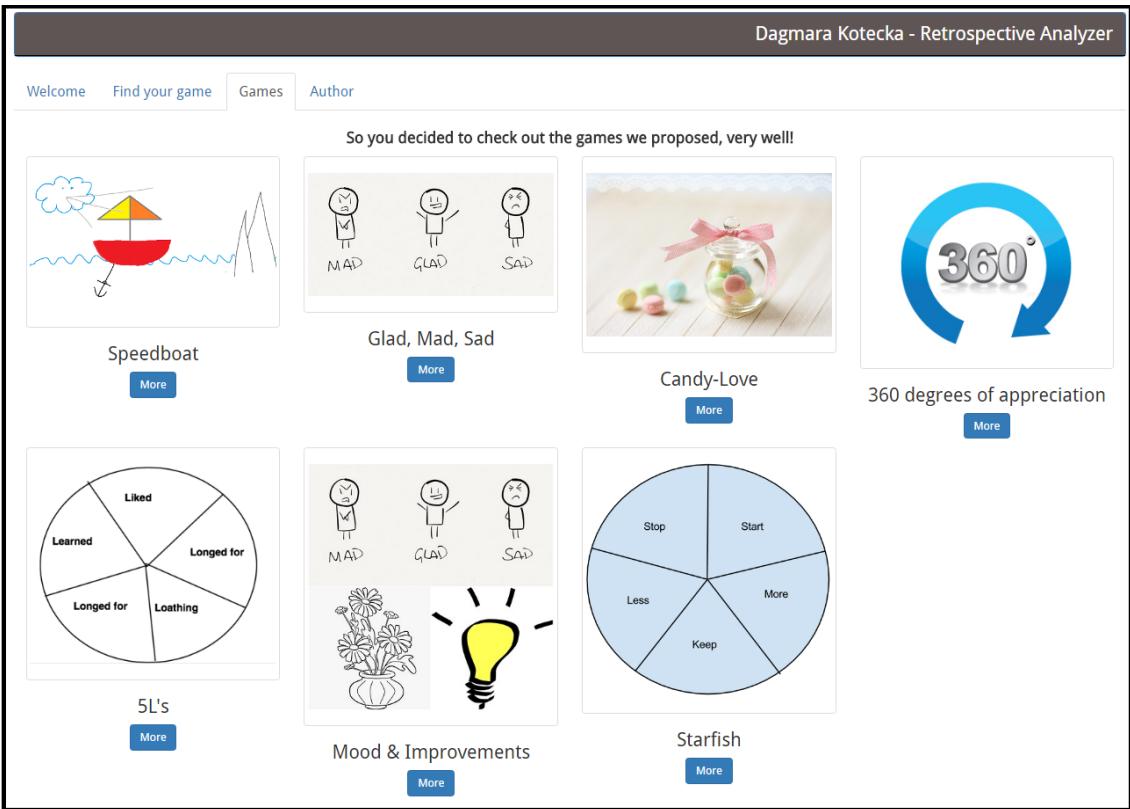


Fig. 5.2. Games page screenshot

The chosen factors included in the table presented on Figure 5.5 are as follows:

1. Creativity - how the game influences the creativity of the team members.
2. Team mood - this factor is useful for the team leaders and managers and thanks to this, they are able to notice what team mood is and react accordingly.
3. Collaboration - this characteristic refers to how the game affects the cooperation of the team members while performing the retrospective meeting.
4. Communication - this element indicates whether the proposed approach enhances the discussions during the meeting.
5. Degree of involvement - shows whether the proposed game increases the involvement of the participants.
6. Change of perspective - indicates how different the approach is compared to the standard procedures and how big an impact the proposed game imposes on the change of the perspective.
7. Honesty - this indicates whether we can observe an increase in honesty amongst the participants whilst they are playing the game.
8. Entertainment - this factor presents how much amusement the game brought to the retrospective meeting.
9. Improvement - indicates how much the game supports the team in terms of proposing ideas for improvements.

Moreover, the nine factors above indicates which game will be retrieved after answering a set questions using a scale 1-5, where 1 is the user strongly disagrees and 5 is strongly agree. The nine questions are listed as follows (also listed on Figure 5.4):

1. Does the creativity in your team need to increase?

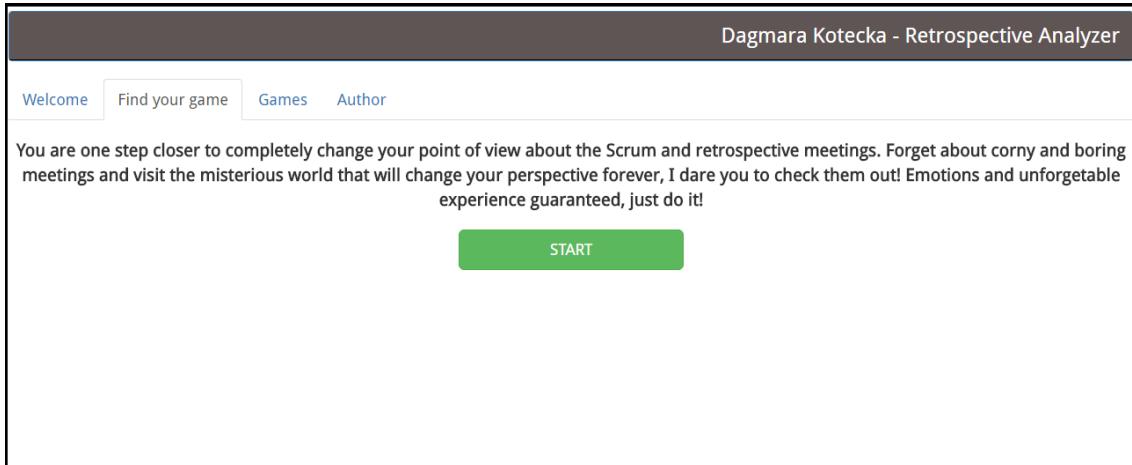


Fig. 5.3. Find game screenshot

Answer the below questions in order to retrieve your retrospective game (1-5)

Does the creativity in your team need to increase:
 1 2 3 4 5

Is your team motivated during the retrospective meetings:
 1 2 3 4 5

Whilst playing the game, is it important for you to observe a change in the team mood:
 1 2 3 4 5

Is your team communicative during the meetings:
 1 2 3 4 5

How involved are the team members during the retrospective meeting:
 1 2 3 4 5

Do you feel that honesty should be improved in your team:
 1 2 3 4 5

How much would you like to change the perspective of the retrospective meetings:
 1 2 3 4 5

Does your team have problems with evaluating possible improvements during or after the retrospective:
 1 2 3 4 5

How important for you is it to be entertained during the retrospective meeting:
 1 2 3 4 5

Fig. 5.4. Questions page screenshot

2. Is your team motivated during the retrospective meetings?
3. Whilst playing the game, is it important for you to observe a change in the team mood?
4. Is your team communicative during the meetings?
5. How involved are the team members during the retrospective meeting?
6. Do you feel that honesty should be improved in your team?
7. How much would you like to change the perspective of the retrospective meetings?
8. Does your team have problems with evaluating possible improvements during or after the retrospective?
9. How important for you is it to be entertained during the retrospective meeting?

The implemented algorithm of choosing the game is written in JavaScript and is presented below in Listing 5.1:

The server receives the data and parses it into the temporary array, *arr*. Afterwards, it is passed into the while loop which is continuously executed until the array stops containing values between 1-5. The first function, *getMaxIndexes* returns indexes of elements from the array that currently

```

while (arr.indexOf("1") !== -1 || arr.indexOf("2") !== -1 ||  

       arr.indexOf("3") !== -1 || arr.indexOf("4") !== -1 ||  

       arr.indexOf("5") !== -1) {  

    getMaxIndexes(arr, function () {  

        changeMaxToZero(function () {  

            getGames(function () {  

                reduceGames();  

            });
        });
    });
}

```

Listing 5.1. The algorithm of choosing the game in JavaScript

have a maximum value. The *changeMaxToZero* procedure sets the values of the returned elements from *getMaxIndexes* to 0. Upon retrieving taking these indexes, the latter is mapped to the factors defined in Figure 5.5 in the *getGames* function and the ones with a

$$factorsValue \geq (MAX * 2) - 1$$

are saved and stored in a global *tmpArray* along with the *max* value which ultimately determines the weight of the game. If the *returnGames* array is empty then the *reduceGames* function assigns the *tmpArray* to the *returnGames*. When the *returnGames* array is not empty, it makes a sum of the *max* with the already stored in it value

$$value[i] = value[i] + max$$

If the array does not contain values of 1-5 the game with the highest weight is being returned and:

1. If the *returnGames* array has more than 1 element, then all of the games are viewed to the user.
2. If the *returnGames* array contains 1 element, the game is shown to the user.

The source of this system has been open-sourced and can be found under this link <https://github.com/dagkotecka/Retrospective-Analyzer>.

After the user answers the questions, the system returns the game and a reference (as a button) to its description. This page is presented on the Figure 5.6.

There are two ways of accessing the game description page shown on Figure 5.7. The first way is to access it via the Games Page (Figure 5.2). The second way, after filling the form on the questions page and retrieving the game from the system, is to click the description button on the Retrieve Game page (Figure 5.6). The game description contains the game name, the rules and the expected goal after playing the game.

The use cases of the software are presented on Figure 5.8. The retrospective meetings are led by developers which include the Scum Master, manager, team leaders and the developers. There are two use cases, where the first one is about viewing games and the second is about answering

| Game Name | Creativity | Team mood | Collaboration | Communication | Degree of involvement | Change of perspective | Honesty | Entertainment | Improvement | Sum |
|-----------------------------|-------------------|------------------|----------------------|----------------------|------------------------------|------------------------------|----------------|----------------------|--------------------|------------|
| Speedboat | 8 | 1 | 1 | 5 | 5 | 10 | 1 | 10 | 8 | 49 |
| Mood and Improvements | 8 | 10 | 1 | 4 | 5 | 5 | 1 | 5 | 10 | 49 |
| Glad, Mad, Sad | 3 | 10 | 1 | 4 | 2 | 1 | 1 | 5 | 7 | 34 |
| Starfish | 10 | 1 | 1 | 5 | 4 | 10 | 1 | 5 | 10 | 47 |
| Candy-love | 5 | 5 | 1 | 8 | 6 | 8 | 7 | 10 | 1 | 51 |
| 360-degrees of appreciation | 8 | 5 | 8 | 10 | 10 | 7 | 8 | 7 | 7 | 70 |
| 5L's | 10 | 10 | 1 | 5 | 5 | 6 | 1 | 10 | 8 | 56 |

Fig. 5.5. Games points table

Dagmara Kotecka - Retrospective Analyzer

Well done! Your retrieved game is: Starfish! Congratulations!

[Get detailed Starfish description](#)

Fig. 5.6. Retrieved Game page screenshot

questions.

Both the backend and frontend of the software has been written using Visual Studio 2015 with a NodeJS extension on the Windows 8.1 platform. The tests have been executed on Google Chrome version 51, Internet Explorer version 11 and Mozilla Firefox version 37 browsers. The results of the evaluation were successful.

5.2. Backend architecture

The backend is written in the NodeJS, which is an asynchronous event driven by JavaScript runtime and it has been designed in order to build scalable network applications. The Google's V8 JavaScript engine interprets JavaScript in the runtime environment. NodeJS enables the creation of servers without using threading and it uses the model of event-driven programming that uses callbacks in order to signal the completion of the task.

The backend framework was chosen from the a variety of available possibilities for a number of reasons. Firstly, both the backend and the frontend can be written in JavaScript, which means there is no switching between languages and also, this means the build process can be simplified. What is more, NodeJS is distinctly faster than for example Java or PHP, as there are no threads and no overheads that slow down the service.

Candy Love

The Candy-Love is team building game rather than solving problems in case of project, but in order to maximize performance and product quality it is advisable to have strong and integrated team. "Candy Love" game integrates team members by letting them speak up and talk about their life beyond work. The participants need to sit at the table, without laptops and other distracting things. One person picks the candy out of the jar and shows it to the team, than checks the candy color meaning. The others need to focus on what the person which picked the candy is talking about. The meanings of the colors are necessary to have in order to properly perform the game, because it's hard to remember all the meanings of the colors, which are as follows:

Red - share one thing that you like about your job, this color retrieves positive emotions, especially on people that are not satisfied or happy with what they do.

Yellow - share your life goal that you are working on, this color illustrates others what is important for that person and can inspire.

Green - share your favorite movie or/and book, this shows person from different angle, maybe this might start a conversion later with people that we usually have nothing to talk about.

Purple - share your favorite way of reviving yourself on a regular workday, this color shows what the person thinking about to decrease stress and might be also a great catalyst for a conversation.

Blue - share one stressful thing in work that you wish you could improve, this color shows what stresses person out, maybe someone has a solution that can share, this can also convert a negative thing into a positive.

Orange - share what is your favorite food, maybe later you can share a meal with someone from the team and it is a topic that everyone likes.

The participants should pass the jar to each team member and end the game when all the candies are gone, when all the team members picked one sweet or when the time of the meeting has ended.



Fig. 5.7. Game description page screenshot

The data is stored in Random Access Memory (RAM), because there is no need to keep it in any database. The game descriptions are rendered by being redirecting to the endpoint with the game name and the table presented on Figure 5.5. The factors are stored in a JSON object, which is a natural part and the foundation of JavaScript and can be used without including any libraries. The other reason to not detain the data in the database is the time-consuming communication between the system and the storage. Moreover, the data was so small that it would have been pointless to store it in a database.

Table 5.1 shows the API Rest calls made to the NodeJS server. The first call was made in order to retrieve the game from the server, using the listed factors in Figure 5.5 factors as parameters. The second call as listed in Table 5.1 is to render the page with the game description. This does not take any parameters.

For the purpose of maintaining the pages Express has been used, which is minimal and flexible NodeJS web application framework. Thanks to Express, it is easy to create a server using just a few lines. The declaration of endpoints and REST Api calls is very simple and compared to the main competitors of the Express framework, the Koa and the Hapi, it has the biggest community and is the most mature and reliable framework. What is more, it promotes code reuse with its built in router.

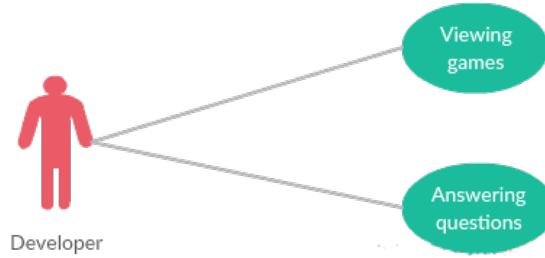


Fig. 5.8. System use cases

Table 5.1. Rest API calls

| Type of HTTP Request | URL | Parameters | On Success | Description |
|----------------------|-----------|---|----------------|---|
| /POST | /gameName | creativity, collaboration, teammood, communication, involvement, honesty, perspective, improvement, entertainment | Returns a game | Call to server to retrieve the game after answering on a set of questions |
| /GET | /getGame | None | Renders Page | Rendering a description page of the games. |

5.3. Frontend structure

While implementing the frontend of the retrospective, the main goal was to create a simple, intuitive and responsive design. In order to simplify the interface, a bootstrap framework has been used. Thanks to the bootstrap framework, the website is able to scale and is properly viewed on phones, tablets and desktops with the CSS media queries.

The choice for the framework that would ease and allow developers to create a responsive design of the responsive website has been made between two of the most popular and well documented solutions. The retrieved candidates were Bootstrap and Foundation. The main reason why the Bootstrap framework has been chosen, was that we have already implemented services using the framework and also, we have experience in that field. The comparison presented in Table 5.2 shows that both frameworks have very similar specifications and without the advantage of our experience in Bootstrap, it would not make difference in terms of choosing which framework to use. The main advantages of Bootstrap are the support for all the modern browsers and the bigger community support. What is more, the code is clean and readable when using the Bootstrap framework. Moreover, it is easy to find the examples and tutorials on how to implement a particular element. The components of the Bootstrap are well documented.

Even though HTML has been a foundation of the web we decided to make use of a more modern approach. Accordingly, Jade was designed at the beginning for server-side templating in NodeJS, but in our solution it has been used as a short hand for HTML. Using Jade has made the code cleaner and there was no need to use ending tags. Jade uses whitespaces and indentation as part of its language, so in order to properly render a page, it is crucial that you are careful and follow the proper syntax. What is more, Jade offers an easy way to write conditions and iterators, which as a result, gives rise to shorter codes. An example is presented on Figure 5.9.

Table 5.2. A comparison between Bootstrap and Foundation

| Comparison area | Bootstrap | Foundation |
|-------------------|------------------------------|--|
| Browser support | all modern web browsers | lack of Internet Explorer 8 support |
| Community support | bigger than Foundation | smaller than Bootstrap |
| Performance | similar to Foundation | similar to Bootstrap |
| Layout definition | easy | easier |
| Sizing units | pixels | rems (equals to the font size of root element) |
| Form validation | not so easy at the beginning | effortless (<i>Abide</i>) |

```
- var languages = ['php', 'node', 'ruby']
div
  each value, index in languages
    p= index + ". " + value

<div>
  <p>0. php</p>
  <p>1. node</p>
  <p>2. ruby</p>
</div>
```

Fig. 5.9. The same layout defined in Jade and HTML

5.4. System evaluation

The web service evaluation has been made in Intel Technology Poland in Gdansk. The Scrum Master, Grzegorz Reglinski, after testing the software, has made a few suggestions. In our company, the Retrospective has been the least appreciated meeting in Scrum and the task of encouraging team members to actively participate is a difficult job. Grzegorz thinks that the Retrospective Analyzer is very useful in terms of supporting the leader, who performs the activity. He found the software is very easy to use, the interface is intuitive, the layout is simple and it does not contain any useless content. What is more, he tested the main functionality of the web-service and the game retrieval, and as a result was satisfied with the outcome. The answers on the input questions suited the game he wanted to get. He also noticed that the retrospective approaches are easy to understand and have an interesting description. Grzegorz found the pictures and the description very important in order to properly perform the game. He suggested a few improvements and two out three of them have been added to the software:

1. The welcome page should contain a picture of the flow and how the game retrieval functionality works. This is because pictures tend to express more than words. The suggested improvement is presented on Figure 5.10.
2. The questions should have bigger spaces between each other and instead of a number describing the opinion, a statement should be used. For example, instead of 5, there could be "the team is very creative, has plenty of ideas and is eager to share them". The suggested improvement was added to the backlog.
3. Each game description should contain the evaluated points presented in Figure 5.5. The suggested improvement is presented on Figure 5.11.

What is more, this complementary Retrospective Analyzer web service has implemented the

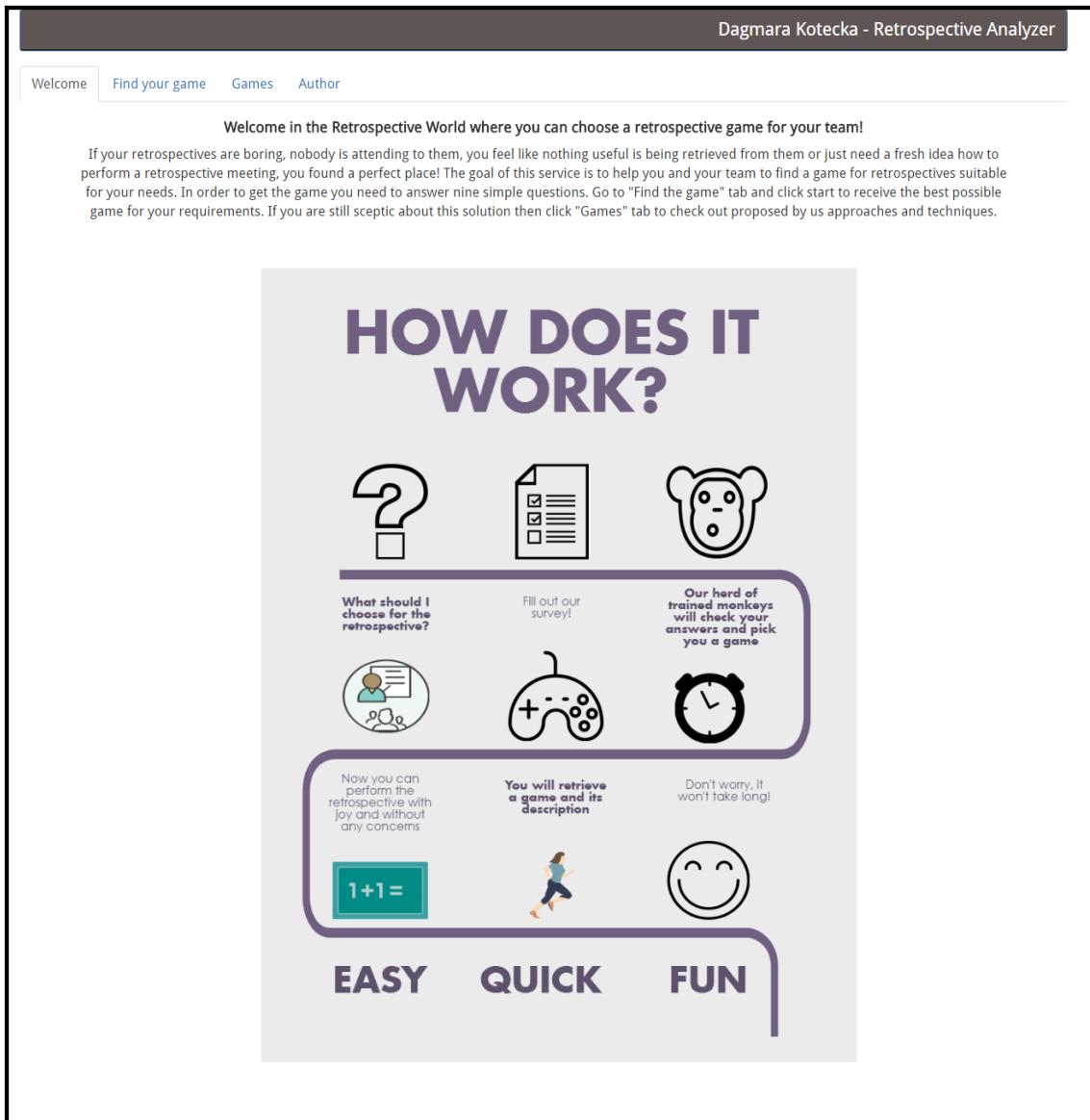


Fig. 5.10. The improved Welcome Page

"must have" features in order to show how many possibilities there are to help the developers to effectively perform the Scrum Retrospective meeting. There are plenty of planned features that would make the system more functional, but as it is with software,, the system must be completed within a given deadline. This is because the development of the service would last forever until someone would eventually say "Stop, that is enough". The possible features that might be included in the software in the future have been retrieved through valuable discussion with Grzegorz Reglinski and these are as follows:

1. As a scrum master, I would like the system to allow the result to be affected by multiple users.
2. As a scrum master, I would like to have the option to create separate groups for the team in order to maintain the results of each group.
3. As a scrum master, I would like to have an admin panel where I can track the results of the game.
4. As a user, I would like the questions to have bigger spaces between each other and instead of a number describing the opinion a statement. For example, instead of 5, there could be "the

Candy Love

Creativity - 5, Team Mood - 5, Collaboration - 1, Communication - 8, Involvement - 6, Change of perspective - 8, Honesty - 7, Entertainment - 10, Improvement - 1

The Candy-Love is team building game rather than solving problems in case of project, but in order to maximize performance and product quality it is advisable to have strong and integrated team. "Candy Love" game integrates team members by letting them speak up and talk about their life beyond work. The participants need to sit at the table, without laptops and other distracting things. One person picks the candy out of the jar and shows it to the team, than checks the candy color meaning. The others need to focus on what the person which picked the candy is talking about. The meanings of the colors are necessary to have in order to properly perform the game, because it's hard to remember all the meanings of the colors, which are as follows:

Red - share one thing that you like about your job, this color retrieves positive emotions, especially on people that are not satisfied or happy with what they do.

Yellow - share your life goal that you are working on, this color illustrates others what is important for that person and can inspire.

Green - share your favorite movie or/and book, this shows person from different angle, maybe this might start a conversation later with people that we usually have nothing to talk about.

Purple - share your favorite way of reviving yourself on a regular workday, this color shows what the person thinking about to decrease stress and might be also a great catalyst for a conversation.

Blue - share one stressful thing in work that you wish you could improve, this color shows what stresses person out, maybe someone has a solution that can share, this can also convert a negative thing into a positive.

Orange - share what is your favorite food, maybe later you can share a meal with someone from the team and it is a topic that everyone likes.

The participants should pass the jar to each team member and end the game when all the candies are gone, when all the team members picked one sweet or when the time of the meeting has ended.



Fig. 5.11. The improved Game Page with the Game Points included

team is very creative, has plenty of ideas and is eager to share them".

6. SUMMARY

This chapter contains the summary of this master thesis, the contribution that has been made and the future work that might be done.

6.1. Contribution

The goal of this master thesis was to propose and implement a set of collaborative games in order to improve involvement of team member, creativity and communication. We also tried to simultaneously to increase the team members' motivation. Moreover, we asked the participants about what their opinion is regarding the game. We required what the influence of the game was compared to standard procedures, whether they would implement it permanently, whether it would complement standard procedures and whether or not the game is easy to understand. In the analysis of the retrieved results, we tried to respond to the hypothesis raised in the first chapter.

Firstly, we gathered all the data required to start the collaborative games implementation. We determined which were the most interesting and as a result, deployed them in the Intel Technology Poland site. While searching the games, we had to analyze many retrospective approaches in order to find the most suitable for our case. Two of the approaches presented in this master thesis have been created in collaboration with Intel and its Scrum Master.

Secondly, we deployed the games in two iterations using different sets of questions. The first iteration of the implementation contained two games which were implemented in two separate teams containing 9 and 3 people, respectively. The question set had been changed though, because after reflection with the supervisor and the study group, we decided it needed improvement in order to retrieve more interesting and less generic characteristics. The second iteration has been implemented in three teams with 9, 3 and 8 participants. Most of the games had been implemented twice, but there were cases when some of them had not been deployed or were deployed only once. This was due to internal organisation issues.

After the meeting we asked the participants to fill out a the survey. We collected and analysed, and in the most cases the results were satisfactory and proved that the hypothesis has been raised in the introduction of this master thesis. In just one game, which is very similar to standard procedures, the team members admitted that in major fields, it does not enhance Retrospective meetings. Besides that, we can claim with high confidence that the collaborative games increase both the creativity and involvement of the team members. Furthermore, we found that the participants of the Retrospective meeting were willing to discuss issues both in the team and in the project while playing the collaborative games. What is more, the additional characteristics, in most of the games were enhanced. The study showed that using sticky-notes and the a blackboard increased the team collaboration and brought more and better results, compared to just writing down the Good Things, Bad Things and Things to Improve.

Furthermore, the focus group had been performed on a group of twelve people. The conclusions have been a major part in describing Scrum in practice. We gathered developers in one room, and using the aforementioned prepared set of questions, we discussed what they thought about this methodology. The team members willingly answered the questions and as a result, interesting

results have been retrieved.

While collaborating with the Scrum Master in Intel Technology Poland in Gdansk, we decided to create a tool that is based on the study presented in this master thesis. The web service created for the purpose of this study has been open-sourced and will be expanded in the future. The main functionality has been implemented, which is game retrieval based on a set of questions. The Retrospective Analyzer is already supporting the Scrum Master to find a suitable game, for a particular team, which we found to be the most important success of this system.

6.2. Future work

This Master Thesis focused only on the Scrum Retrospective and the study about whether the collaborative games enhance the meeting has been fully covered. In the future, it would be useful to research other Scrum practices. There are plenty of games that could be implemented for the Planning meeting, Backlog grooming and Daily Stand-ups. It would be interesting to verify whether the collaborative games are able to enhance better estimation of the tasks or prioritization of the user stories. What is more, an intriguing piece of research research would be the improvement of Daily Stand-Ups. In this case, it would be useful to focus on fitting into the Daily Stand-Up "timebox" or directing team members to talk only about "what you did yesterday" and "what are you going to do today" instead of digging into the implementation details.

Another piece of future work should be to expand the software described in this master thesis. The Retrospective Analyzer would be an even more useful tool if it had a "creating groups" functionality, an admin panel and the possibility of allowing multiple team members to have an effect on the resulting game.

6.3. Conclusion

To sum up, the collaborative games enhance the Scrum Retrospective meeting and has shown to have a positive impact on the team members. The Scrum Master from Intel Technology Poland has been pleased with the functionality that the complementary software offers, and as a result has been added to this thesis. The Retrospective Analyzer is being used in the real organization and it has been shown to fulfill its primary purpose.

BIBLIOGRAPHY

- [1] I. N. Takeuchi Hirotaka, "The new new product development game." pp. 137–146, January 1986.
- [2] K. S. Rubin, "Essential scrum: A practical guide to the most popular agile process," pp. 1–3, 165–171, 185–187, 196–198, 375, July 2012.
- [3] C. Keith, "Agile game development with scrum," p. 13, May 2010.
- [4] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, R. C. Martin, S. Mellor, D. Thomas, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, K. Schwaber, and J. Sutherland, "Agile manifesto."
- [5] J. C. G. Miguel Ehécatl Morales-Trujillo, Hanna Oktaba, ""improving software projects inception phase using games activeaction workshop", 2014.
- [6] E. . Associates, "Guidelines for conducting a focus group," p. 1, 2005.
- [7] M.-W. L. Dictionary, "Survey in merriam-webster dictionary." [Online]. Available: <http://www.merriam-webster.com/dictionary/survey>
- [8] M. Zakrzewski, "Enhancing requirements elicitation process through collaborative play," 2015.
- [9] D. Gelperin, "Improve requirements understanding by playing serious games," 2011.
- [10] VersionOne, "Versionone: 9th annual state of agile survey. technical report," 2015. [Online]. Available: <https://www.versionone.com/pdf/state-of-agile-development-survey-ninth.pdf>
- [11] M. Cohn, "Advantages of the "as a user, i want" user story template." April 2008. [Online]. Available: <http://www.mountaingoatsoftware.com/blog/advantages-of-the-as-a-user-i-want-user-story-template>
- [12] A. Marcano, "Old favourite: Feature injection user stories on a business value theme," March 2011. [Online]. Available: http://antonymarcano.com/blog/2011/03/fi_stories/
- [13] K. Nielsen, "Mcgraw-hill education acp agile certified practitioner exam," p. 138, March 2016.
- [14] K. R. Sondra Ashmore Ph.D, "Introduction to agile methods," pp. 165–167, June 2014.
- [15] S. M. Jerrel Blankenship, Matthew Bussa, "Pro agile .net development with scrum," p. 25, October 2011.
- [16] J. Benton, "A guide for retrospectives," July 2013. [Online]. Available: <http://www.slideshare.net/jasonfbenton/a-guide-toretrospectives>
- [17] S. Thomas, "Glad sad mad retrospectives," September 2013. [Online]. Available: <http://itsadeliverything.com/glad-sad-mad-retrospectives>
- [18] T. T. C. Paulo Caroli, "Starfish," August 2012. [Online]. Available: <http://www.funretrospectives.com/starfish/>
- [19] ——, "360 degrees appreciation." [Online]. Available: <http://www.funretrospectives.com/360-degrees-appreciation/>
- [20] ——, ""4 ls: Liked – learned – lacked – longed for". [Online]. Available: <http://www.funretrospectives.com/the-4-ls-liked-learned-lacked-longed-for/>
- [21] M. Zwilling, "How to increase productivity by employee happiness," December 2014. [Online]. Available: <http://www.forbes.com/sites/martinzwilling/2014/12/02/how-to-squeeze-productivity-from-employee-happiness/#6a0860b01de5>
- [22] S. Achor, "Positive intelligence," January-February 2012. [Online]. Available: <https://hbr.org/2012/01/positive-intelligence>
- [23] T. T. C. Paulo Caroli, "Candy love." [Online]. Available: <http://www.funretrospectives.com/candy-love/>

- [24] R. M. Ferreira, "The power of anonymous retrospectives," June 2014. [Online]. Available: <https://www.infoq.com/news/2014/06/power-anonymous-retrospectives>
- [25] H. Amin, "Do's and don'ts of agile retrospectives," July 2014. [Online]. Available: <https://www.scrumalliance.org/community/articles/2014/july/dos-and-don-ts-of-agile-retrospectives>
- [26] B. L. Luis Goncalves, "Getting value out of agile retrospectives: A toolbox of retrospective excercises," pp. 18–20, 23–24, July 2014. [Online]. Available: <https://www.infoq.com/news/2014/06/power-anonymous-retrospectives>

LIST OF FIGURES

| | |
|---|----|
| 2.1. Agile development overview [2]..... | 14 |
| 2.2. Scrum cycle development | 15 |
| 2.3. Roles in scrum [2] | 15 |
| 2.4. Product owner responsibilities [2]..... | 16 |
| 2.5. Scrum master responsibilities [2] | 16 |
| 2.6. Development Team responsibilities [2]..... | 17 |
| 2.7. Scrum meetings and process..... | 18 |
| 2.8. Scrum One-part sprint-planning approach [2]..... | 19 |
| 3.1. Speedboat..... | 25 |
| 3.2. Happy/Sad/Angry | 26 |
| 3.3. Starfish | 27 |
| 3.4. Mood and Improvements..... | 28 |
| 3.5. 4L's | 29 |
| 3.6. 5L's | 30 |
| 4.1. A bar graph depicting the results when deploying the Speedboat game using the old set of questions | 35 |
| 4.2. A bar graph depicting the results when deploying the Starfish game using the old set of questions..... | 36 |
| 4.3. A comparison of deploying starfish and speedboat games using the old set of questions | 37 |
| 4.4. Speedboat deployment | 38 |
| 4.5. A bar graph depicting the results when deploying the Speedboat game using the new set of questions | 38 |
| 4.6. Starfish deployment..... | 40 |
| 4.7. A bar graph depicting the results when deploying the Starfish game using the new set of questions..... | 40 |
| 4.8. Glad, Mad, Sad deployment..... | 41 |
| 4.9. A bar graph depicting the results when deploying the Glad/Mad/Sad game using the new set of questions..... | 42 |
| 4.10.Mood and improvements game deployment | 42 |
| 4.11.A bar graph depicting the results when deploying the Mood and Improvements game using the new set of questions | 43 |
| 4.12.5 L's game deployment..... | 44 |
| 4.13.A different representation of the 5 L's game deployment..... | 45 |
| 4.14.Results of deploying 5 L's game using new set of questions..... | 46 |
| 4.15.Overall timeline of the games..... | 46 |
| 4.16.A comparison of deploying all the games using the new set of questions | 49 |
| 4.17.A table showing grouped results of all the games..... | 51 |
| 5.1. Welcome page screenshot..... | 52 |

| | |
|--|----|
| 5.2. Games page screenshot | 53 |
| 5.3. Find game screenshot..... | 54 |
| 5.4. Questions page screenshot | 54 |
| 5.5. Games points table | 56 |
| 5.6. Retrieved Game page screenshot..... | 56 |
| 5.7. Game description page screenshot | 57 |
| 5.8. System use cases | 58 |
| 5.9. The same layout defined in Jade and HTML..... | 59 |
| 5.10.The improved Welcome Page | 60 |
| 5.11.The improved Game Page with the Game Points included..... | 61 |

LIST OF TABLES

| | |
|---|----|
| 2.1. Participants of focus group profiles | 20 |
| 3.1. Requirements to start Speedboat game | 25 |
| 3.2. Requirements to start Happy/Sad/Angry game..... | 26 |
| 3.3. Requirements to start Starfish game..... | 27 |
| 3.4. Requirements to start 360-degrees of appreciation game | 27 |
| 3.5. Requirements to start Mood and Improvements game | 28 |
| 3.6. Requirements to start 5L's game..... | 29 |
| 3.7. Requirements to start Candy-Love Game..... | 30 |
| 4.1. Groups | 33 |
| 4.2. Number of feedbacks using an old set of questions..... | 34 |
| 4.3. Number of feedbacks using the new set of questions..... | 34 |
| 4.4. Results of the Speedboat Game | 35 |
| 4.5. Results of the Starfish Game | 36 |
| 4.6. Results of the Speedboat Game | 37 |
| 4.7. Results of the Starfish Game | 39 |
| 4.8. Results of the Glad, Mad, Sad Game | 41 |
| 4.9. Results of the Mood and Improvements Game..... | 43 |
| 4.10. Results of the 5L's Game..... | 45 |
| 4.11.Overall results of the games from the feedback..... | 50 |
| 5.1. Rest API calls..... | 58 |
| 5.2. A comparison between Bootstrap and Foundation | 59 |