

ALMA MATER STUDIORUM - UNIVERSITÀ DI
BOLOGNA

Scuola di Scienze
Corso di Laurea in Informatica

**VIRTUAL TEAM BUILDING THROUGH
SERIOUS GAMES: THE SCRUMBLE CASE
STUDY**

Tesi di Laurea

Presentata da:
Roberto Barbone

Relatore:
**Chiar.mo Prof.
Paolo Ciancarini**

Sessione II
Anno Accademico 2019-2020

Contents

1	Introduction	1
1.1	Agile methodology and Scrum	3
1.1.1	Manifesto for Agile Software Development	3
1.1.2	Scrum	5
1.2	Gamification and serious games in software development	7
1.2.1	Serious games & Scrum	7
1.3	GQM: The Goal-Question-Metric approach	12
1.4	Thesis goal	13
2	Methods and technologies for team building	14
2.1	Introduction	14
2.1.1	The meeting platforms	15
2.2	Conditions required by the working ambient	16
2.2.1	Video conference	16
2.2.2	Scheduling meetings	16
2.2.3	Number of people allowed	17
2.2.4	Online streaming	17
2.2.5	Economic availability	18
2.2.6	Working tools	18
2.2.7	Status control	19
2.3	Summary	20
3	Scrumble	21
3.1	The aim of the game	23

3.1.1	The technical debt	23
3.2	Phases of the game	24
3.2.1	Pre-sprint	24
3.2.2	Sprint	25
3.2.3	Sprint Review	27
3.2.4	Retrospective	27
4	Elements of the Game sessions	28
4.1	The subjects	28
4.2	GQM - Metric of evaluation	29
4.2.1	Learning	31
4.2.2	Practicing	31
4.2.3	Cooperation	31
4.2.4	Motivation	31
4.2.5	Problem Solving	32
4.3	Tools and Platforms used	32
4.3.1	Excel	32
4.3.2	Teamviewer	34
4.3.3	Discord	35
4.4	First Session	35
4.4.1	Participants	35
4.4.2	Product to develop: BuyMe	36
4.5	Second Session	38
4.5.1	Participants	38
4.5.2	Product to develop: TrackMyFood	38
4.6	Third Session	41

4.6.1	Participants	41
4.6.2	Product to develop: QuickTrip	41
5	Analysis of the experience	43
5.1	First Session: BuyMe	43
5.1.1	General considerations	43
5.1.2	Analysis with GQM	44
5.2	Second Session: TrackMyFood	47
5.2.1	General considerations	47
5.2.2	Analysis with GQM	48
5.3	Third Session: QuickTrip	50
5.3.1	General considerations	51
5.3.2	Analysis with GQM	52
5.4	Results	54
6	Conclusion and future implications	56
A	GQM table compilation	58

Abstract

Alla luce dell'espansione delle tecnologie di virtualizzazione dei meeting, ancora più propagata in questo periodo di pandemia, nascono nuove occasioni per verificare e studiare la validità di un processo di apprendimento effettuato da remoto.

Inserita in un contesto di sviluppo software agile, questa tesi si pone lo scopo di misurare come le capacità di team building di un gruppo di giovani studenti possano migliorare attraverso la ludicizzazione, propriamente detta Gamification, del modello di sviluppo agile Scrum. Scrumble, strumento della tesi da me impiegato, è un serious game che percorre e simula ogni fase seguita da uno Scrum team per soddisfare i requisiti implementativi necessari al rilascio di un prodotto.

Vertendo quindi sulla ricerca della misurazione di quanto effettivamente lo svolgimento di un processo che coinvolga lo studente su un piano "giocosco" possa influenzare le sue capacità collaborative di gruppo, mi sono avvalso dell'utilizzo dell'approccio Goal-Question-Metric, riempiendo una tabella di valutazione nel corso delle tre sessioni di gioco effettuate.

Ciò che ne risulta dimostra come delle sedute di gioco di Scrumble effettuate da remoto possono ritenersi un valore aggiunto nell'apprendimento e nel rafforzamento delle skills di un team di sviluppo, anche se appena formato, con componenti quindi sconosciuti tra loro e che giocano da remoto.

Chapter 1

Introduction

Learning is an extremely fuzzy concept and is hardly possible to provide an accurate definition. De Houwer, Barnes-Holmes e Moors define learning as "changes in the behavior of an organism that are the result of regularities in the environment of that organism" [1] pointing out the importance of an occurrence which reiterates in an environmental context that influence the way in which this occurrence is stimulated. Moving away from a behavioural perspective and approaching a more technical hypothesis concerning education in terms of "Learning as the processing of information or experience [2]", learning is conceived in terms of "the storage of information in memory as a consequence of any experience the individual might have had" [3]. Again, "Things learn when they change their behaviour in a way that makes them perform better in the future" [4].

Therefore, it is possible to infer a needed process consisting in the repetition of a physical or mental action related to an external event of which we live the experience, and caused to that event we change our behavior so that by repeating the action we execute it in a better manner. Defined the starting basis, it is important that this

operation of personal growth won't be perceived as a negative event, but instead, it seems that a learning procedure placed in a positive context where the knowledge absorption is correlated with a feeling of fun and enjoyment, accelerate the storage of the information, resulting then in a better future expression of these information.

Dorothy Lucardie conducted a study of several adults to prove that the quality of learning is heavily affected by a fun factor inherent in the growth path: have a joy feeling not only encourage the interest and engagement in taking in new data, but let also the chance to establish great interpersonal relationships, enticing students to stay longer in the place dedicated to the learning [5].

Thus, a joyful surrounding can be the key to develop team building skills, but what team building actually is? Micheal Beer defines it as a process by which members of a group diagnose how they work together and plan changes which will improve their effectiveness [6], whereas Boss considers it as “interventions designed to improve effectiveness in working together by confronting and resolving problems” [7]. These interpretations make even more clearer that an entertaining component is fundamental to boost the group performance, and what will be tested is if the insertion of playful mechanics and elements in a non-playful context, can intensify the learning willing and the desire to get involved in a working team. This phenomena, as will be soon explained, is defined Gamification.

In this first chapter, we break through the concepts of Gamification and serious games, Agile and Goal-Question-Metric method of evaluation, which give us an overview of the domain in which I will operate. The second chapter resumes the aim my research wants to obtain, while the third one illustrates the meeting platforms and the condition requires by them to bring a substantial help in the work of improving the team building skills of the subjects of the experiment. The fourth chapter describes the game I played with my team, followed by the chapter which lists all

the elements used to support my evaluations and game sessions. The sixth chapter contains all the results and observations regarding the experience, and the last one sums up including possible implications in the future.

1.1 Agile methodology and Scrum

Agile software development is a group of software development methodologies based on iterative and incremental development (IID) [8], in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. This concept completely detaches from the traditional waterfall model, which principle is driven by a rigid sequential order: once a phase is completed its results are *frozen*, which means that it is not possible to go back and revise; in addition to that, until the results of the current phase are not completed, it is disallowed to start working on a new one [9].

Agile is based on continuous cycles which end with a release of the product developed, persistent interaction of the project team with the user, tiny documentation and great reaction to changes.

1.1.1 Manifesto for Agile Software Development

The realization of the principles on which agile methodologies were born is related to the forming of the Manifesto for Agile Software Development [10]. The document, redacted by the software developers of the Agile Alliance, highlights how directness and constant interaction can get over the old concepts of project management, where tons of work were made in only one go before having any contact whit the actual interested: the customer.

Below we list the main values pointed out by the Manifesto:

- **Individuals and interactions** over processes and tools:

collaboration and team work is the most efficient source to get a work done

- **Working software** over comprehensive documentation:

Considering that agile is based on frequent releases, it is more important the software constantly satisfies the tests requisites.

- **Customer collaboration** over contract negotiation:

The customer is the one who actually knows how is the product he or she desires.

- **Responding to change** over following a plan:

Priority can change many times during the development, and a good team as to be ready to deal with it.

We can therefore deduce that in a situation where we need to encourage the communication, collaboration and trust between team members, an agile approach is way more suitable rather than the waterfall model.

Between the main agile methodologies, we can find Test Driven Development, Extreme Programming, and Scrum: we will focus on the latter since is the engine of the game my study is about.

1.1.2 Scrum

Scrum is defined by Ken Schwaber and Jeff Sutherland as "a framework within people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value". This framework is made by roles, events and artifact, all assembled together to boost the efficacy of the product management and the working techniques [11].

Scrum is composed by sprints, which lasts for two to four weeks and each one brings to a new release of the product. The customer defines the functions to realise and their priorities while the Scrum team decides every day the best way to implement the functions which have the higher priority. These functions are created as *User Stories*:

"As a *< subject >* I want *< objective >* so that *< purpose >*".

Ex. As a *customer* I want *the chart option* so that *I can visualize the saved articles not bought yet*.

All the stories are collected in the Product Backlog, and for each sprint the chosen stories to implement originate the Sprint Backlog. The team release what has been done, following the principle of the *Definition of Done*, namely a shared understanding within the Scrum Team on what it takes to make the Product Increment releasable [12], for example an integrated, working and tested functionality.

SCRUM FRAMEWORK

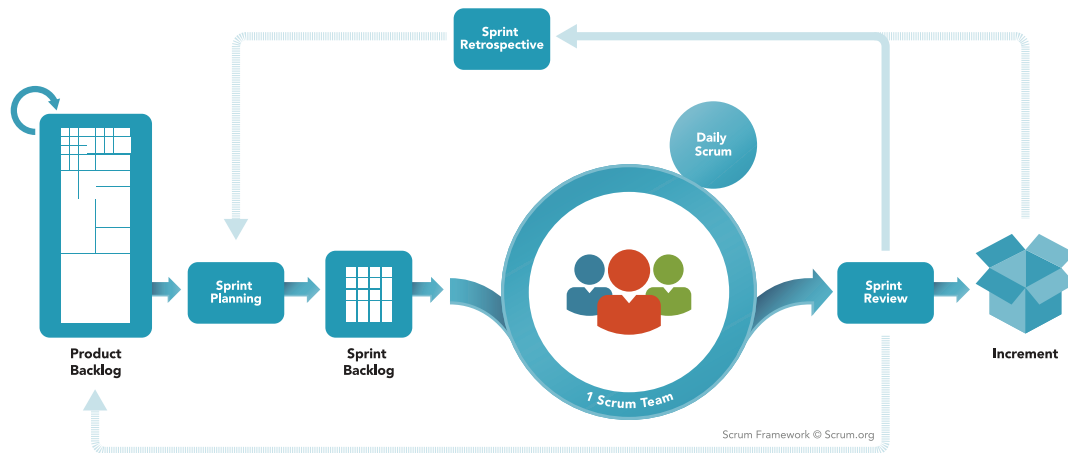


Figure 1.1: Scrum framework, www.scrum.org

The Scrum team is formed by three indispensable roles:

- **Scrum Master:** He helps the team to learn and apply Scrum in order to increase productivity. He is not a boss, but a role model for the team.
- **Product Owner (PO):** He is responsible for the maximization of the return on investment (ROI), identifying the features of the product which can lead to a major success. He handles the User Stories and decides how to compose the Sprint Backlog.
- **Development team:** It is the core of the Scrum team, it includes members

with multiple and different software skills. Their cooperation is the key to let them be capable of fulfill the product requirements.

1.2 Gamification and serious games in software development

One of the pillar between Gamification's definitions is related to Sebastian Deterding, who describes it as the "use of game design elements in non-game contexts" [13], detaching the word "gameful" from the concept of "playful", where the last one applies to behaviours and mindsets as a consequence of the gamification result. Cathie Marache-Francisco emphasizes the need of reversing the meaning of work through gamification, bringing ludic procedures and enjoyable interaction in a context which otherwise could be characterized by conflicts and alienation [14].

How can we relate Gamification and Agile? Agile bases its principles on collaboration and cooperation in order to achieve short-term goals, and this approach can be easily brought to certain game mechanics, where every one plays a role with the aim of winning the game. The "role-playing" idea widely helps agile teams in a better understanding on what are the problems that another member of the team has to deal with: switching roles inside a game highlights the needs and the peculiarities of every component of the group, raising awareness and social skills.

1.2.1 Serious games & Scrum

Whereas Gamification applies to gamified characteristics to enhance a ludic context, serious games relate to the use of complete games for non-entertainment purpose; their aim is to focus the game group on problem solving, learning, simulating real-life

working situation and improving communication and cooperation [15].

Concerning Scrum, it is very easy to relate it with a serious game: reaching a goal in a limited amount of time is an efficiency prerogative for a Scrum team; in addition to that, many phases of the Scrum model can be transposed in a game, even the model itself as we will discuss later. Here there are some examples of serious game applied to Scrum.

Planning Poker

Planning Poker, also know as Scrum Poker, is a game created for estimating the effort and the size of development goals in software development, namely the User Stories which will compose the Sprint Backlog. The goal is to obtain a consensus about the effort needed in order to align the awareness about the work to be done before the start of the sprint.

As researcher Alan McClean pointed out in his study, gamification can be often used as a part of release planning in an agile environment [16]. He showed how the serious game Planning Poker could actually resolve the problem of "analysis paralysis" during the early phases of stories estimation, cutting of useless wastes of time in discussing low level details.



Figure 1.2: A team in a Planning Poker session, ITiCSE '17, July 3-5, 2017, Bologna, Italy

Scrum Lego City

Scrum Lego City is a game which introduces the basic concepts of Scrum development process [17]. The goal consists in building a town with the construction pieces, with many teams creating each one an edifice. The Product Backlog is then composed by many buildings, and the various teams, by playing planning poker, have to estimate the complexity of every construction. After that, they choose what to build in a certain number of sprints, and they start to build the city. Following every sprint, a review is made to understand how qualitatively the city is being built, and if the team is feeling satisfied by the work done. At the end of the game a brief interview

is done to understand how the players felt about the game.



Figure 1.3: A team playing Lego Scrum, <https://antoniopalomaresfernandez.com/lego4scrum/>

Scrumia

Considering that many serious games has been created more for a professional training rather than for a session during a university course, Scrumia purpose is to teach Scrum applications in a 60 minutes game [18]. Students divide in groups of six people, where one is the PO, one the Scrum Master and the other the development

team, the only ones who can actually create something. SCRUMIA is a planet and the team has to satisfy the need of the customers who want to reach it by massively producing two means of transport, planes and boats. Both of them have their costs and profits and they have to be done with simple sheets of paper. The goal throughout the sprint is to make the major number of vehicles according to the needs of the PO: the fun part is that the planes actually have to fly and the boats have to fit the head of the players or the PO.



Figure 1.4: A team playing Scrumia, from "A Model for the Evaluation of Educational Games for Teaching Software Engineering", Rafael Savi

1.3 GQM: The Goal-Question-Metric approach

The Goal Question Metric approach, promoted for the first time by Victor Basili, is based on the idea that a subject willing to measure the quality of some processes or projects must specify the goals for itself and its projects, then trace those goals with the data which define them and finally provide a framework for interpreting the data respecting the goals initially stated [19]. The measurement model has three levels:

- **Conceptual level - GOAL:** A goal is defined for an object for which we want to reach an higher quality level; an object of measurement can be a product, a process or a resource.
- **Operational level - QUESTION:** We use a set of questions to specify how we want the goal to be performed, trying to characterize the object.
- **Quantitative level - METRIC:** A data set is associated with every question in order to answer it in a quantitative way; the data can be objective or subjective.

Using GQM to assess the quality of a process in terms of the impact it has on the feelings and perceptions of human beings is not an easy task. Aleksandr Tarasov attempted to evaluate the job satisfaction of software engineers, posing a range of questions asking how job contentment is dependent on personal activities, passions, relationships with colleagues and work methodologies, like Agile [20]. Indeed, this last one, appeared to positively influence the job satisfaction, together with many personal and emotional factors, as interactions with customers, good relations with colleagues and management and general level of serenity in the working ambient.

Thus, it is possible to denote that in order to evaluate the quality of an agile teaching approach like a Scrum serious games it is necessary to outline concrete

results of the players feelings, which are the most powerful tool to determine the quality of a gamified process. In my study I will extrapolate data from three game sessions of a serious game named Scrumble, which will be completely explained in the third chapter.

1.4 Thesis goal

The thesis goal consists in evaluating the learning growth towards team building skills and the Scrum agile method in a remote environment of a selected team which plays the serious game Scrumble during three virtual meetings.

Through the Goal-Question-Metric approach behaviours and actions will be observed throughout three different sessions in order to understand the efficiency of a remote learning and measure an actual increase from the beginning to the end of a playing path.

Chapter 2

Methods and technologies for team building

2.1 Introduction

During the quarantine caused by the spreading of virus Covid-19, almost every company had to replan its working strategies in order to keep up with production and development.

Beyond work, this condition involves also agile teams, and the way they get trained by agile coaches. Considering that is not a simple and exact discipline teaching team building skills even in a physical meeting, I want to try to give the best remote solutions to let agile coaches fulfil their goals.

It is easy to image the tons of online platforms which were born in these months, trying to catch the majority of users in every country all over the world. Nonetheless, we may be able to narrow down our searching by considering which are the conditions needed the most to enable our team grow up with the values required by agile

methods, such as harmony, practice, cooperation, motivation, problem solving and so on.

2.1.1 The meeting platforms

In the pursuance of reaching our goals, we will take few of these online meeting platforms to analyse a bit deeper. These platforms are:

- Zoom, a popular meeting platform used world-wide, intuitive and adopted by many schools and university to practice frontal courses. It holds a large amount of people talking together, from 100 to 1000.
- Skype, one of the oldest web app utilized for friendly calls and group chats; it updated itself in order to provide many of the new tools needed , such as holding a call between a dozens of people, simplify the usage with browser direct service and allow multiple screen sharing.
- Streamyard, the best web-based platform for streaming, allows the handling of more than one live at the same time, on Facebook, Youtube, Twitch or LinkedIn, while up to 6 people can actively participate to the video meeting.
- TeamViewer, the most famous software for remote controlling, provides many tools to let the participant interact with the meeting, like for example controlling the cursor of the host.
- Teams, a Microsoft platform similar to Zoom which combines group chats, meeting with thousands of people, and sharing of contents. It is widely used for teaching purposes.

- Discord, born to create videogamers communities, it has been widely used also by non-gamers considering the ad-hoc server everyone can create based on the personal needs.

2.2 Conditions required by the working ambient

Once we introduced our chosen platform, we can continue listing the condition required to let a team improve its team building skills.

2.2.1 Video conference

It may seem obvious but having the chance to virtually see the people we are talking to it is fundamental for creating a good team. Visualize our teammates enhances the confidence and lets us sympathizes each other, creating what is called the "office atmosphere" [21]. That is the reason why the best platform we are looking for has to provide a good video quality and the possibility to share multiple webcam videos together.

Luckily this condition is highly carried out by our cavy platforms.

2.2.2 Scheduling meetings

With the goal of obtaining a great team building skilled group, it is highly necessary for a team to easily organize their meetings, better with a tool provided by the conference app itself.

Zoom let the host choose on which calendar publicize the event, in order to make it visible any time someone wants to take a look at it. Skype can create a scheduled meeting inside a chat already created, and only the members of the group chat can

see it. Teamviewer opens Outlook or any other mail app of the host and auto create the invite message, leaving to the coach only the job of adding the recipients. Teams creates an event in which can be invited only the people already linked to the host: for example, the account connected to the university can add people only if belonging to it. Lastly, Streamyard gives the chance to schedule a live stream, making it visible for the followers of the streaming channel.

2.2.3 Number of people allowed

Dependently of the number of people belonging to the team we want to create, it is useful that the platform can host many participants at the same time, in order to guarantee none has the risk to be excluded.

By default plans, Zoom can host up to 100 participants in every meeting, Skype put a maximum of 50 people; Teams provides space for 250 clients in a conference call, TeamViewer up to 25, while Streamyard is the narrowest with 10 people in the call and 6 allowed to show themselves

2.2.4 Online streaming

In case we would like to share our team building learning path with a public who may be interested in our themes, it can be a good practice to live stream the training sessions, not only to provide an educational journey, but also to receive precious feedbacks, considering that remote training is often a field unknown also by the coaches.

Teams possess its own extension, Teams Live Events, to transmit meeting with a public even more numerous than the default limit; Discord is able to link the two most famous streaming account, Twitch and Youtube, to share on one of them the

streaming; Zoom works similarly for Youtube and Facebook. The real revolution concerning streaming is realised by StreamYard, which gives the chance to stream on more than one streaming channel at the same time, potentially allowing an infinite number of viewers.

2.2.5 Economic availability

We know our goal is to determine the best application to support team building, but it is mandatory to take count of the economic side of our research: if a coach would like to bring his experience to a university course, he has to be aware of the fact that students should not pay the platform to assist at the training, and he himself can obtain the results he wants without wasting a huge amount of money. Indeed, many meeting platforms have a free licence which sadly do not let users talk in video conference with more than one person, schedule meetings, or even share the screen.

TeamViever, except for number of participants limited to 5 excluding the host, has the same functionalities: this is due to the fact that the company gives the free licence for private use. Skype, Teams and Discord as almost the full set of tools even in the free licence, such as Zoom whose only flaw is to hold meetings that last up to 40 minutes. StreamYard instead, on its free version let the host live stream in just one platform at the time.

2.2.6 Working tools

So far as we discussed, we analysed the frame in which our platform should operate: it is now time to consider the main details needed to ensure a better training session experience. First of all, having a wide set of working tools, can make the difference between which software to use, starting from the possibility of sharing the screen.

Then, transform the screen as a sort of virtual blackboard in which pin up every thought or annotation of the session, or even highlight a part of the screen we want to focus on, can be really useful. Moreover, having the chance to record the sessions boosts the learning speed, letting the team review what they have done.

Discord has a good audio management, which lets the participants mute other participants just for them, in case for example that two groups have to work without listening to each other's. Zoom, Skype for Business, Teams and TeamViewer provide an efficient whiteboard functionality for annotation and drawings; the latest one has a really nice tool to encourage team interactions: every participant in the meeting, if allowed by the host, can have the remote control of the host's screen, operating at his place. This is also useful for the agile coach to control and verify the work done by the team, exploring the layouts himself.

2.2.7 Status control

Another feature that can make more efficient the time between each session is the possibility to see whether a member of our team is online and available to get contacted. Indeed, knowing when we can share opinions of doubts with the people we are working with helps the team to grow faster, instead of waiting every session to explain our thoughts or misunderstandings.

In Teams, Zoom, Skype and Discord is provided a set of status icons which are commonly divided between online, offline, absent and busy.

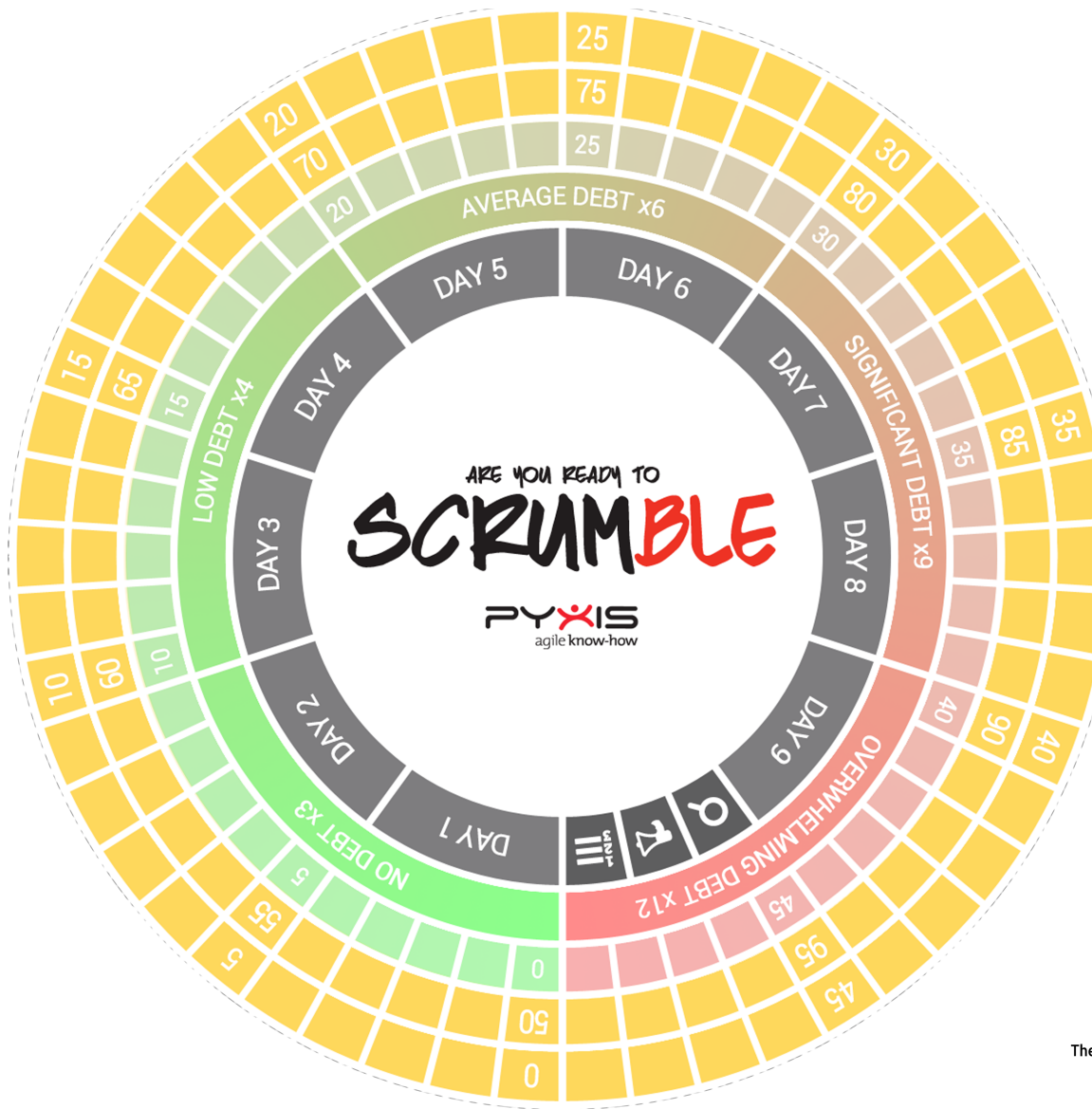
2.3 Summary

To conclude our analysis, we can surely affirm that a good meeting platform is a fundamental ingredient for the team building growth recipe: from the video quality to the functional tools, a software does not have to limit the teamwork in any way. Furthermore, we understood that different platforms have different goals to achieve: whereas StreamYard is finalized to share the contents with a huge public within many streaming platforms, TeamViewer provide all the tools needed for technical reunions, with whiteboard functionalities and remote control direct access. Examined this last one, I would suggest it as a great meeting platform for small agile teams, even more if during their experience they will play some agile games, like for example Scrumble.

Chapter 3

Scrumble

Scrumble is a board game based on the roles, events and artifacts of Scrum. The goal is to show to a team the steps a Scrum team would carry out in order to deploy a new product. The game simulates an experience which passes through some common problems, allowing the game group to test its team building skills facing up many decisions to take [22].



The

Figure 3.1: Game board of Scrumble

3.1 The aim of the game

The aim of Scrumble is to complete every User Story required to develop the product, throughout the gradual accomplishment of its specifications during a chosen number of Sprints, before the technical debt becomes unsustainable.

3.1.1 The technical debt

The **technical debt** represents every complication which can show up in a project and it is related to "immature, incomplete or inadequate artifacts in the software development cycle that cause higher costs and lower quality" [23]: it can be a bug, a testing error, team interaction problems and so on.

At the beginning of the game, the default factor is x4 and the debt marker on 15 points. Then, it can evolve positively or negatively, based on the difficulties encountered, like ending a sprint with uncompleted work, or bad decisions taken from the players, who neglect it. Once the debt reaches 49 point the game is over.

Points of debt	0 to 9 📁	10 to 19 📁	20 to 29 📁	30 to 39 📁	40 to 49 📁
Debt factor	x 3	x 4	x 6	x 9	x 12
Handicap	- 25%	0%	+ 50%	+ 125%	+ 200%

Figure 3.2: Ranges of technical debt factors and handicaps depending of the points of debt

3.2 Phases of the game

3.2.1 Pre-sprint

The first phase of the game consists in the Product Owner presenting to the team the product they will have to create during the game. The better he explains his vision about the product, the more the team will feel involved and interested in playing actively.

He is in charge of sharing the Product Backlog and deciding which User Stories have to be achieved in every sprint. At the first beginning, the PO presents every User Story as clearer as he can, in a way the team is able to truly understand them; when this is done, team members have to subjectively estimate the complexity of each story in terms of actual implementation.

In this step is fundamental that every participant thinks all by himself, so that no one gets influenced by someone else's opinion. On the contrary, after the estimation is a good practice to share the personal thoughts about the complexity of the tasks, in order to grow up within the team a common sense of evaluation, whose ideal aim is to be as similar as possible.

Once the team is done judging the complexity of the tasks, they will have to sort them according to their priority: in this case developers are required to visually figure out the way the product should be deployed to the public, trying to answer to the questions "*Which items provide the most customer value?*", "*Which items provide the most benefit to the business?*", "*What are the dependencies between items?*" [24]

Thus, all these information are allocated in the following table:

By complexity	1 - XS	2 - S	3 - M	4 - L	5 - XL
By priority					
⊕ High					
⊗ Average					
⊕ Low					

Figure 3.3: Sorting table for the User Stories, allocated by complexity and priority

3.2.2 Sprint

Sprints comprise a planning session and ten working days, in which the first nine are development days and the last one is dedicated to the review session and the retrospective.

Sprint Planning

During the sprint planning phase, the Product Owner is in charge of compose the Sprint Backlog, choosing which are the User Stories to be carried out as soon as possible; obviously the team can give its opinion to the PO but he or she has to make the final decision.

In order to distribute the task pawns on the board, it is necessary to calculate the number of tasks which each User Story includes: this operation is done multiplying the current technical debt factor by the complexity of the User Story, all by the number of the developers (excluding then the PO and the Scrum Master). This is well explained in the table below:

Technical debt factor \ User Story's complexity		1 - XS	2 - S	3 - M	4 - L	5 - XL
		No 🎲 x 3	3 x 👤	6 x 👤	9 x 👤	12 x 👤
Low 🎲 x 4	4 x 👤	8 x 👤	12 x 👤	16 x 👤	20 x 👤	
Average 🎲 x 6	6 x 👤	12 x 👤	18 x 👤	24 x 👤	30 x 👤	
Significant 🎲 x 9	9 x 👤	18 x 👤	27 x 👤	36 x 👤	45 x 👤	
Overwhelming 🎲 x 12	12 x 👤	24 x 👤	36 x 👤	48 x 👤	60 x 👤	

Figure 3.4: Operations needed to calculate the number of tasks to be done for each User Story

Once all the task pawns are distributed, the User Story will be done and deliverable when the Players pawn has reached its markers.

Development days

In this phase of the game, every member of the developing team chooses to execute tasks or reduce the technical debt, by rolling a dice and advance or moving backwards according to the number obtained.

The team is highly encouraged to follow a group strategy in order to prevent possible unexpected events. Indeed, in case the player rolls 6, he or she has to pick a **Problem card** and read it out loud to be clear for everyone. A problem card represents any complication a development team could possibly have, and each member has to act together to overcome the difficulty, by thinking at the best following moves.

At the end of every development day, the players pick a **Daily card**, which can be a challenge for the team, a question about the work on progress, or an anecdote

about Agile. After that, a new turn starts and the team goes to the next development day, up to the review.

3.2.3 Sprint Review

At the end of the nine working days, the Product Owner inspects the job done during the sprints, verbally resuming what has been done and encouraging or giving advices to the team whereas they struggled over the game.

The PO then picks as many **Review cards** as the User Stories completely done, which can impact the rest of the game.

In addition to that, the PO gives as many technical debt points for every group of 5 tasks not completed during the sprint. The lack of completion may brutally affect the difficulty of carrying out the User Stories remained.

3.2.4 Retrospective

Lastly, the retrospective is the moment dedicated to the participants. Every player has the chance to express what he liked, disliked, and what phases of the game he would execute differently to have a better impact on the final result.

When the last retrospective comes, it is the best moment to extract all of what the team has learned throughout the experience: the mechanism of Scrum and agiles methods, and the approach to a teamwork *forma mentis*, in order to grow the harmony, the cooperation and the productivity of every group the team members will have to share a project with [25].

Chapter 4

Elements of the Game sessions

4.1 The subjects

The experimentation has been performed throughout three separated meetings in order to mix different combinations of a chosen group of six guys: the first two encounters were composed, excluding me, by four components, while the last one by all six of them, executing a team versus team version of the game.

The background of all of the young people chosen is from a university context: I decided to pick three couples of people who come from different friendly circumstances, so that I could observe better the interactions between them and between members who they have no confidence with.

The participants of the experiments:

- Alice and Federico, they are currently attending the Computer Science degree course in Bologna, I worked with them with some university projects.
- Marco and Fausto, they have been friends of mine for a long time, they recently took a degree in two different Economy degree courses.

- Camilla and Sara, they are studying for the Master degree in Cyber Risk, I am presently working with them on an extracurricular project.

Players	Age	Studies	Sex
Alice	22	Degree in Computer Science	F
Federico	22	Degree in Computer Science	M
Marco	23	Degree in Economics and Management	M
Fausto	24	Master degree in Economics and Food Marketing	M
Camilla	22	Master degree in Cyber Risk	F
Sara	23	Master degree in Cyber Risk	F

Figure 4.1: Basic data of the players

4.2 GQM - Metric of evaluation

In order to evaluate and analyse the growth of the team building skills within the group, I decided to utilize a Goal-Question-Metric approach which could look over many aspects related to team building. These aspects are listed in the table below, provided with their relative questions and metrics. To compare the first two sessions and the last one towards concrete results, both improvements or worsening, I will evaluate the behavior basing myself on a scale from 1 to 5, where 1 means bad and 5

really good, applicable in every field of the GQM table, as explained in the appendix at the end.

GOAL	QUESTION	METRIC
Learn	Do team members understand the Scrum roles?	Knowledge of Scrum roles by questions
	Do team members feel they learned the process?	Opinions from the participants
	Does everyone keep up with the other players?	Check during every sprint retrospective if every one is on point
Practice	Are the game mechanics linear and repeatable?	Opinions from the participants
	Do team success in completing the game?	Number of User Stories completed
	Do team members efficiently estimate during sprint planning?	Uniformity in evaluating the size and the priority of user stories
Cooperation	Do team members know each other better?	Level of players' serenity throughout the game
	Does the game let all players cooperate?	Contribution of every player during the game
	Do team member consult each other about a topic?	Sharing of ideas
Motivation	Do team members encourage colleagues in need?	Players explain something other players don't understand
	Does PO help the team?	Quality of PO's advices to get better in the next sprints
	Does the team come up with good ideas?	Effectiveness of sprint retrospective
Problem Solving	Do team members behave well when facing a problem?	Level of the technical debt at the end of the game
	Does team organize their tasks properly?	Average of tasks left at the end of each sprint
	Does PO plan efficiently the Sprint Backlog?	Average of tasks left at the end of each sprint

Figure 4.2: GQM table related to many skills that team building coaches aim to reach

4.2.1 Learning

The ability of learning is a fundamental skill in a development environment which constantly renovates itself. It is always needed to evolve the personal skills, and working in a team is a great way to efficiently learn from the team mates, only if we are truly willing to do it.

4.2.2 Practicing

'Practice makes perfect', or almost: agile methods highly focus on practice, which lets the team improve their productivity and make better decision in the future sprints. Brief deadlines enable a fast growth in terms of experience and error avoidance.

4.2.3 Cooperation

Cooperation consists in the ability of working with harmony and agreement in a situation where the goal to reach is the same for everyone. Increasing the confidence between the team mates lifts the spirit and forms a stable team. One reason for greater productivity in stable teams is that they don't have to repeatedly go through the team-building stages of forming, storming and performing over and over again [26].

4.2.4 Motivation

The motivation is strongly related to the cooperation of the team: the more the developers are confident with the people they work with, the more they are willing to quickly accomplish a project. Confidence is not necessary linked to excellence: in a team is more important to create the best combination of people based on

their interdependent skills, preferences and personalities rather than take the greatest software developers in the world [26].

4.2.5 Problem Solving

We could define Problem Solving as the fusion of the previous skills: in a team inclined to learn from the mistake and willing to try until the result is done, no problem cannot be solved if everyone puts effort to cooperate in finding its solution. Further more, while so many projects fail nowadays, the motivation to pass through a down period enables the team to stick together and discover a way to fix it up.

4.3 Tools and Platforms used

4.3.1 Excel

Towards the goal of proficiently playing Scrumble in remote meetings, I decided to transpose the game elements in an Excel sheet; fast and simple to use, it made me organize well all the set up needed for the game: the board, the table with the User Stories, the pawns, and the effect cards to pick up during the game. For the last ones, I simply inserted them in a column of cells, and by the time a card is picked up, I just need to delete the row and the next card appears.

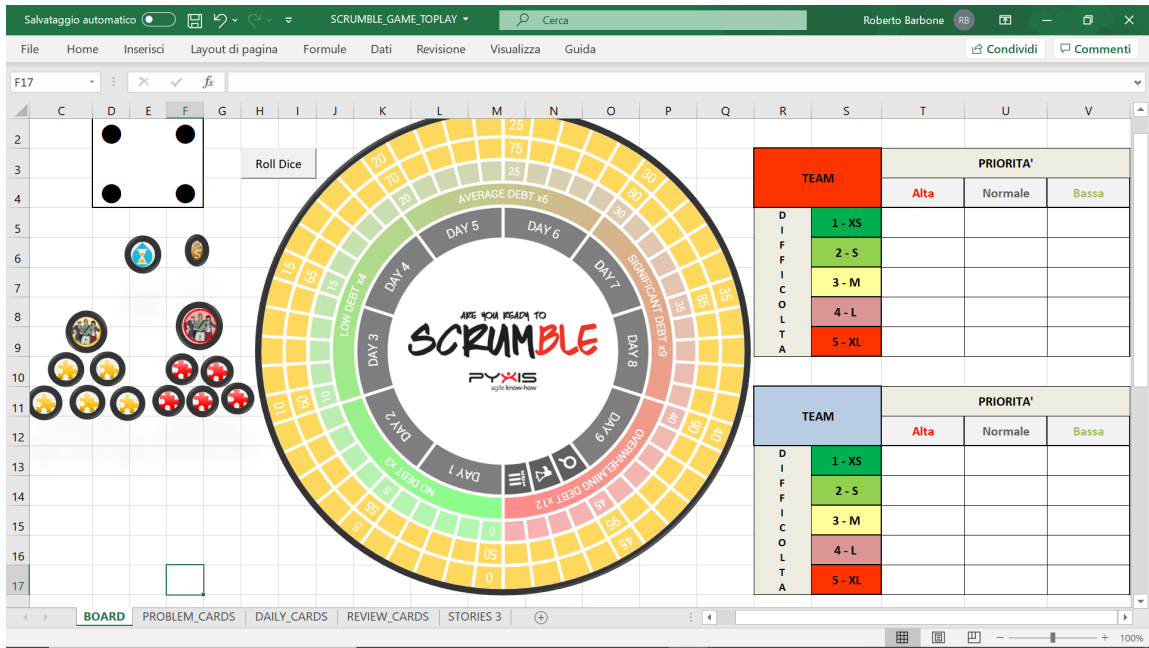


Figure 4.3: Transposition of Scrumble in an Excel sheet

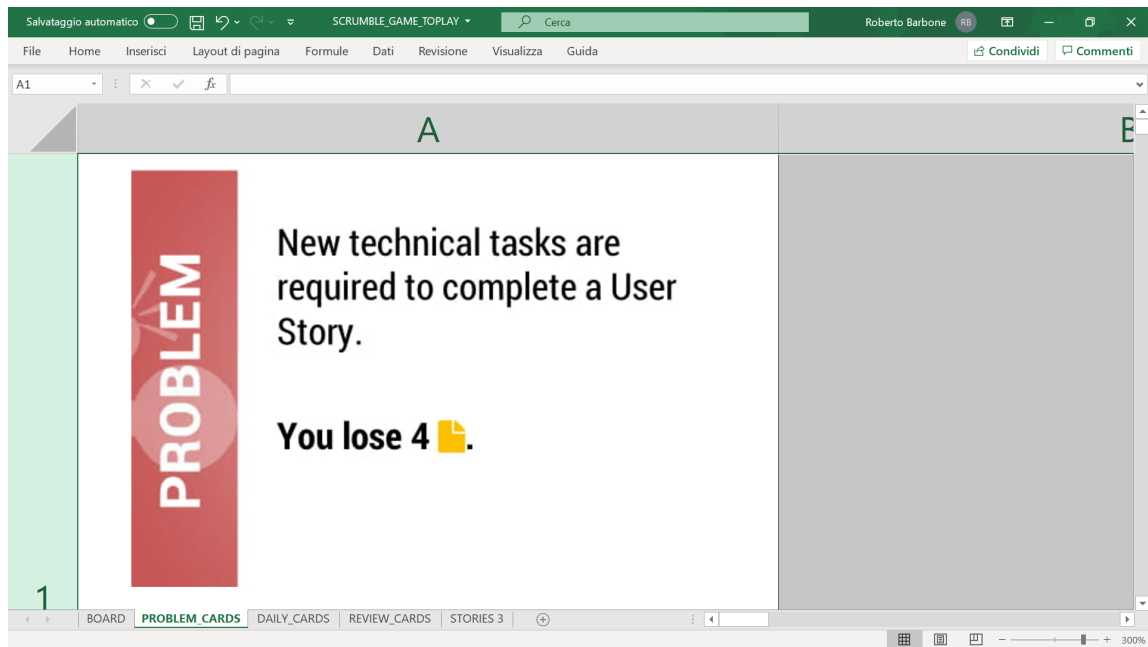


Figure 4.4: Example of effect card to "pick up"

4.3.2 Teamviewer

For all the experimental session, I decided to use TeamViewer as the platform for video conferencing: considering that Scrumble is first of all a board game, I felt essential giving the participants the chance to personally interact with the game. Indeed, TeamViewer during a video conference meeting lets the member use the host cursor to perform every task: In this way, the players could virtually roll the dice and move the pawns like they were physically present in the host room.

4.3.3 Discord

In the last session, I took also advantage of the great audio management system of Discord: seen that the game consisted in a team versus team game, in many phases the platform let a team mute the other and vice versa, allowing me to simultaneously analysing the thinking process of both teams.

4.4 First Session

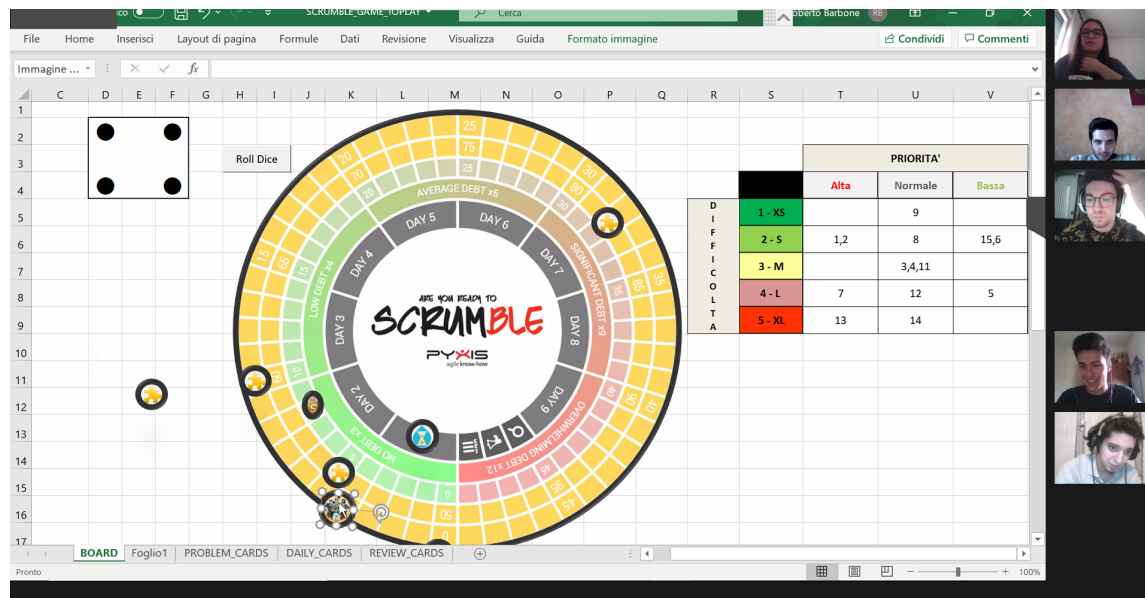


Figure 4.5: The team playing Scrumble

4.4.1 Participants

In the first session four guys participated: Alice, Federico, Fausto and Marco. Before the date of the meeting, they have been provided with a brief explanation of Scrum

and a simplified version of Scrumble's rules. Federico impersonated the Product Owner, choosing a product which fitted with the given set of User Stories illustrated in the next subsection.

4.4.2 Product to develop: BuyMe

BuyMe is a service of food delivery where every supermarket involved has the chance to insert its products on the related site; the users shop online and the corresponding supermarket sends the purchase through a courier working for BuyMe.

Shopkeepers can add, edit and remove products from the shop, decide when their service is not open to the public, track visitors statistics and secure their page, in order to prevent erroneous or fraudulent orders.

On the other hand, visitors can create an account filling it with personal details, filter the list of product by many indexes, decide the payment methods and give suggestions to the shops.




















<p> 1 Requires  13</p> <p>As a shopkeeper, I want to add and edit new products for my shop.</p> <p>Value: ★★★★★ Size:</p>	<p> 2 Requires  13</p> <p>As a shopkeeper, I want to prevent new orders during my holidays.</p> <p>Value: ★ Size:</p>	<p> 3</p> <p>As a visitor, I want to see the list of top products.</p> <p>Value: ★ Size:</p>
<p> 4 Requires  11</p> <p>As a customer, I want to edit my banking and contact information.</p> <p>Value: ★★★★★ Size:</p>	<p> 5</p> <p>As a regular customer, I want to receive frequent gift suggestions by email.</p> <p>Value: ★★ Size:</p>	<p> 6</p> <p>As a visitor, I want to share my wishes on the social networks.</p> <p>Value: ★★★★★ Size:</p>
<p> 7</p> <p>As a customer, I want to order using Paypal, Visa, transfer or check.</p> <p>Value: ★★★★★ Size:</p>	<p> 8</p> <p>As a visitor, I want to search for gifts, filtering and sorting them by price.</p> <p>Value: ★★ Size:</p>	<p> 9</p> <p>As a visitor, I want to search for gifts according to themes, colors and materials.</p> <p>Value: ★★★★★ Size:</p>
<p> 10</p> <p>As a customer, I want to get a receipt and the tracking of my order.</p> <p>Value: ★★★ Size:</p>	<p> 11</p> <p>As a regular visitor or buyer, I want to create a customer account.</p> <p>Value: ★★ Size:</p>	<p> 12</p> <p>As a shopkeeper, I want key indicators about the visitors, their interests and behaviors.</p> <p>Value: ★★★★★ Size:</p>
<p> 13</p> <p>As a shopkeeper, I want a secure administration area to manage my website.</p> <p>Value: ★★ Size:</p>	<p> 14 Requires  13</p> <p>As a shopkeeper, I want to cancel any erroneous or fraudulent order from the secure administration area.</p> <p>Value: ★★★★★ Size:</p>	<p> 15</p> <p>As a visitor, I want to suggest my own gift ideas to the shop.</p> <p>Value: ★ Size:</p>

Figure 4.6: List of User Stories of BuyMe

4.5 Second Session

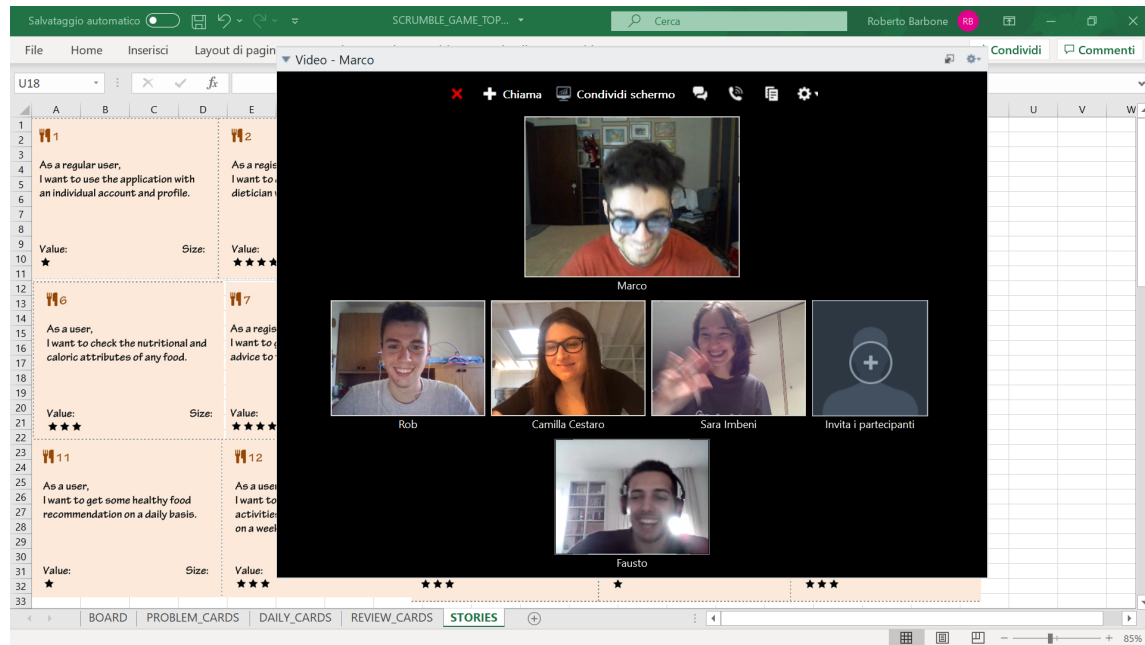


Figure 4.7: The team playing Scrumble

4.5.1 Participants

In the second session I decided to keep two participants, Fausto and Marco, and add Camilla and Sara as new players. This choice was made in order to analyse the impact of two new player assisted by others with a bit more of experience.

4.5.2 Product to develop: TrackMyFood

TrackMyFood is an app which let the user checks all the nutritional and caloric attributes of any food, providing him a way to easily follow a diet.

Every registered user is followed by a dietician, who has access to all the information about the people he coaches; in addition to that, he can give suggestions and plans to successfully observe the diet.

Normal users can receive suggestions about sports and activities recommended, obtain information about the best dietetics in the area, read some famous diets and get motivational pills. When they create an account they get the chance to choose a dietician, set the own goals, receive a complete nutrition program and write a diary containing meals and activities' history.
























<p> 1</p> <p>As a regular user, I want to use the application with an individual account and profile.</p> <p>Value: ★ Size:</p>	<p> 2 Requires  1</p> <p>As a registered user, I want to designate my own dietician who will be my coach.</p> <p>Value: ★★★★★ Size:</p>	<p> 3 Requires  2</p> <p>As a registered dietician, I want to access to all information about someone I am the coach.</p> <p>Value: ★★★★★ Size:</p>
<p> 4 Requires  1</p> <p>As a registered user, I want to set my own goals, to reach at a given time.</p> <p>Value: ★★★★★ Size:</p>	<p> 5</p> <p>As a user, I want to read about the most efficient known diets.</p> <p>Value: ★★★★★ Size:</p>	<p> 6</p> <p>As a user, I want to check the nutritional and caloric attributes of any food.</p> <p>Value: ★★★ Size:</p>
<p> 7 Requires  2</p> <p>As a registered dietician, I want to give some objectives advice to the users I am the coach.</p> <p>Value: ★★★★★★ Size:</p>	<p> 8 Requires  1</p> <p>As a registered user, I want to provide useful personal information for my coaching.</p> <p>Value: ★★★ Size:</p>	<p> 9 Requires  1</p> <p>As a registered user, I want to receive frequent and personalized offers to improve my health.</p> <p>Value: ★★★ Size:</p>
<p> 10</p> <p>As a user, I want to receive a few automatic incentives to motivate me.</p> <p>Value: ★★ Size:</p>	<p> 11</p> <p>As a user, I want to get some healthy food recommendation on a daily basis.</p> <p>Value: ★ Size:</p>	<p> 12</p> <p>As a user, I want to get some sports and activities recommendation nearby on a weekly basis.</p> <p>Value: ★★★ Size:</p>
<p> 13 Requires  1</p> <p>As a registered user, I want a complete nutrition program based on my profile and personal goals.</p> <p>Value: ★★★ Size:</p>	<p> 14</p> <p>As a user, I want to find the best addresses in the area regarding dietetics.</p> <p>Value: ★ Size:</p>	<p> 15 Requires  1</p> <p>As a registered user, I want to write a meals and activities diary.</p> <p>Value: ★★★ Size:</p>

Figure 4.8: List of User Stories of TrackMyFood

4.6 Third Session

4.6.1 Participants

In the last session, I asked all of the previous participants to play in a team versus team version of the game, later better explained. Marco, Camilla and Alice played together in the team "Aliroulette", against the team "Ferrovie Svizzere" composed by Sara, Fausto and Federico.

4.6.2 Product to develop: QuickTrip

QuickTrip is a travel app which lets the user create personalized trips, keep track of the voyages already done and receive offers and suggestions for the future ones.

The common visitors can explore the travel diaries published by other to take inspiration, look at the travel offers, generate random trip with some filters and see the most popular locations.

The registered users can create their own travel diary where they can add photos and notes, personalize the trips generated by the app, receive suggestions based on previous experiences and create an entire travel path all by himself.

<p>1 Requires 4</p> <p>As a registered user, I want to write a travel diary and list all visited places to it.</p> <p>Value: ★★ Size:</p>	<p>2 Requires 4</p> <p>As a registered user, I want to add photos and notes to my travel diary for each visited place.</p> <p>Value: ★★★ Size:</p>	<p>3 Requires 12</p> <p>As a visitor, I want to browse the publicly published travel diaries and paths.</p> <p>Value: ★★★★★ Size:</p>
<p>4</p> <p>As a regular user, I want to manage a minimal QuickTrip account to store information.</p> <p>Value: ★★ Size:</p>	<p>5</p> <p>As a visitor, I want to see some travel offers in the region.</p> <p>Value: ★★ Size:</p>	<p>6</p> <p>As a visitor, I want to generate one-day travels in a limited range of 200km and for a cost of less than \$500.</p> <p>Value: ★★ Size:</p>
<p>7 Requires 4</p> <p>As a registered user, I want to generate travels lasting up to two months.</p> <p>Value: ★★★ Size:</p>	<p>8 Requires 4</p> <p>As a registered user, I want to generate travels with some personalized stops around the world.</p> <p>Value: ★★★★★ Size:</p>	<p>9 Requires 4</p> <p>As a registered user, I want to restrict my budget for the generated travels.</p> <p>Value: ★★★★★ Size:</p>
<p>10 Requires 4</p> <p>As a registered user, I want to create travel paths.</p> <p>Value: ★★★★★ Size:</p>	<p>11 Requires 4</p> <p>As a registered user, I want to extract the path of a generated travel to edit it as my own.</p> <p>Value: ★★ Size:</p>	<p>12 Requires 4</p> <p>As a registered user, I want to share publicly some of my travel diaries and paths.</p> <p>Value: ★★ Size:</p>
<p>13 Requires 4</p> <p>As a registered user, I want to receive new travel paths after one month of inactivity</p> <p>Value: ★★★ Size:</p>	<p>14 Requires 4</p> <p>As a registered user, I want to be assisted by QuickTrip when paying expenses for a whole travel.</p> <p>42</p> <p>Value: ★★★ Size:</p>	<p>15</p> <p>As a visitor, I want to see the most popular locations within a chosen area.</p> <p>Value: ★★ Size:</p>

Figure 4.9: List of User Stories of QuickTrip

Chapter 5

Analysis of the experience

During our route aimed to improve the team building skills of the chosen group, the approaches and the decisions have been analysed as well as the qualities fundamental for a stable team, the ones listed in our GQM table.

Therefore, for every session played, I will describe and express a general consideration of the attitude shown, and then I will try to answer the questions created in the GQM plan, with the support of the results of the GQM evaluation.

5.1 First Session: BuyMe

5.1.1 General considerations

In the first session, I took the role of the Scrum Master, in order to assist the path of the team during the game, while Federico proposed himself as the Product Owner, leaving to Alice, Fausto and Marco the roles of the developers.

The first impact I felt as soon as we started playing Scrumble concerned the professionalism every one immediately displayed, even though all of them are firstly

friends rather than colleagues or co-workers. They took Scrumble for a real agile and team building training, not just for a board game to play for fun.

Final retrospective

At the end of the first session, I interviewed the players relatively to the game, and summed it up.

Alice: *"The game gifts a simplified vision of Scrum, every role and development phase are easy and intuitive, if well handled is a fast track for a great team work"*.

Federico: *"Scrumble lets understand what is the idea behind all the path from choosing the priority to complete the final product. It enhances team importance and cooperation in making decisions"*.

Fausto: *"I felt it is very important to give the right value to User Stories and to prioritize them, essential phase for the product development and the success with the customers"*.

Marco: *"I think the Sprint planning phase is very important , where we want to make sure that the Sprint Backlog is well balanced for every Sprint, in order to complete all the tasks and minimize the technical debt"*.

5.1.2 Analysis with GQM

Learning

Generally the objective of transmitting the idea of the Scrum process has been reached: on one side players who have never attended a computer science university course believed the game is a valid tool to enter the world of agile development; on the other side players with a computer science education had the chance to clarify and simplify the base aspects of the operating procedures of a Scrum team.

Thanks to the straightforward learning approach, nobody was left behind, and every doubt born during the game has been made clear during the review phases, and very important, not only by me but also by other players.

Moreover, thanks to the frequent recalls of the game to the theory of Scrum, through the **Daily** and **Review** cards, I often could consult the gamers on which are the Scrum roles and their functionalities.

Practice

Even though players showed a genuine interest in determining priority during the estimation phase, they tended to underestimate the importance of evaluating the User Stories relatively to the completion of the product: at the beginning they didn't follow the priorities they assigned themselves and they didn't balance the amount of User Stories by dividing it between easy ones and hard ones, increasing in this way the technical debt caused by the lack of completed tasks.

During the game they realized the necessity of this balancing and they adjusted their decision, which let them complete five User Stories in the last Sprint instead of three in the first two Sprints.

Cooperation

Even if at the beginning team members appeared not so comfortable, considering it was an environment never experimented before, they rapidly created a feeling between them, sharing thoughts and impressions on what they were doing.

Cooperation indeed proved to be fundamental in evaluating User Stories priority, and let the whole team members give their contribution to the main decisions: the marketing, cyber security and scientific competences of the group made enormously

realistic the feasibility evaluation of the User Stories and their logic queuing.

Motivation

The increasing harmony inside the squad permitted to the players to feel free of suggest and encourage other players ideas towards the choices related to the production methods during the development days, the order of task execution or the importance of the reduction of technical debt.

The PO played his role very wisely, leaving freedom of speech to the players and intervening regularly whereas the team messed with the order of User Stories execution or their complexity; furthermore, during the reviews at the end of each sprint, he tried to confront with the team to balance the amount of work throughout the three Sprints.

Problem Solving

What made me really satisfied with the game session was my marginal presence during the entire game (excluding the beginning for obvious reasons). The team always attempted to analyse and understand what could be the right choice to take before making any decision, without leaving things to chance.

Concerning the priority estimation of the User Stories, every player appeared to be a lot involved and interested in evaluating the customer needs which in a real world context have the priority: they expressed their doubts and perplexities and they discussed sharing opinions based on their personal experience.

Also the approach towards technical debt varied throughout the game: initially the team thought only to reduce the technical debt without considering the productive side of the game, but after they noticed that at the end of the nine development

days some tasks were missing, they changed the plan thinking at first to the development of the User Stories, by intervening on the technical debt only in case it would have become higher than a prefixed threshold.

Not so obvious and interesting was the uniformity in deciding to keep the technical debt in the lowest range, which metaphorically represents the special attention in reducing every possible bug and problem born during the programming phase.

At the hint of a problem, represented by the **Problem** cards, team wavered a bit: I deemed it was more than common, since for who doesn't have a solid experience in programming and team work, the sudden born of a complication, both an implementation (bug) or an organizational one (client dissatisfaction), can destabilize and slow down the initial game strategies.

In this regard, a good but non so easy solution could be to add in the retrospective a small stage of risk forecasting, with the consequent mental preparation to its immediate resolution.

5.2 Second Session: TrackMyFood

In the second session I kept my role of Scrum Master considering there were two new entries in the development team, Camilla and Sara. Marco took the role of Product Owner, while Fausto kept his one as developer.

5.2.1 General considerations

What clearly appeared was that every beginning process took much fewer time than the first session: even if two members didn't play yet, the other supported me transmitting the concept of Scrum and Scrumble.

Final retrospective

Camilla: *"I like a lot the idea of giving priority to the problems, in some ways it is what we should do in everyday life; moreover, only one person choosing what to do can destabilize the solidity of the team, working and deciding in group is fundamental"*.

Fausto: *"In this second session I had the chance to focus more on the details, roles, actions, estimations... In addition to it, I think it is really important the PO constantly gets involved to support us: we are a team and every one has to give his opinion"*.

Marco: *"As a PO i felt the need of give advice to the new entries and also to the whole team, when I passively watch them play, I can notice much easier when they make a doubtful decision"*.

Sara: *"The game offers an adequate base to approach the agile world; maybe adding some competition we could engage ourselves even more"*.

5.2.2 Analysis with GQM

Learning

Having a couple of new members playing with another one a little more experienced gave me the chance to analyse the different approach in learning new stuff: while the newest ones focused on understand the concepts of the game, the others concentrated more on small details to get in order to deal with the game in a more conscious approach.

Despite the different knowledge level, nobody felt to be left behind except for short moments, also thanks to Fausto and Marco who had no problem in explaining some concepts when they thought it was needed to proceed in the game.

Practice

Conscious of the errors made the last time, the team spent more time in pondering how many User Stories they could do in every Sprint, in order to avoid increasing the technical debt at the end of it.

Much pressure has been made by the "old" players to focus on the task execution instead of wasting the initial development days in reducing the technical debt: this decision was crucial because at the end of the third Sprint they have been able to complete the product.

Cooperation

Surprisingly, cooperation goals have been widely reached: the sharing of a common goal to achieve and the comfortable situation I tried to create in order to make them feel free to express their opinion, highly promoted the desire of completing the game with commitment; new developers asked support in some decisions, identifying their selves in a real Scrum working scene.

Furthermore, who already knew how to play, owned a stronger feel of leadership, inviting his fellows to reason on the User Stories and their priority as realistically as possible.

Motivation

Motivation has been increasing throughout the game: at first the new gamers were obviously a little disoriented while the others already got to experiment the idea of the game, not brand new anymore; secondly, as with every good-made board game, the engagement took hold and the desire of enjoying the episode turned out in effort of winning the game and deploy the product.

Problem Solving

This second session was a great opportunity to adjust the mindset towards the resolution of the game: at the end of the first Sprint, the team comprehended that in every circle of working days, they were approximately reaching the same amount of cells on the board; in this way, they started programming the next sprint summing up the number of tasks of every User Story according to the number of cells they could possibly achieve.

Thanks to this strategy, it was also easier to handle the growth of the technical debt, because relatively to the development they were currently at, they could understand if executing a task had the priority over the debt or not.

5.3 Third Session: QuickTrip

In the last session I tried to explore new ways to play Scrumble, testing a team versus team version, in which I played the role of the Product Owner and the other six were all developers. I decided to give them total freedom in the construction of the Sprint Backlog, without interfering with their decisions.

Additionally, instead of giving the two teams all of the fifteen User Stories in the pre-set, in the first Sprint planning I shared with them just the first twelve. At the end of it I pretended that from the interaction with the customers, the desires about the product changed and I challenged the teams to create three brand new User Stories each, which fitted the clients needs both in contents and priority.

In order to win the game, I stated that the final result was made from the subtraction between the sum of the complexity value of the User Stories completed and the amount of technical debt left at the end of the game; the team with the highest

value is the winner. In this way, developers were dared to come up with a strategy which could let them execute as more tasks as possible without forgetting about the debt.

5.3.1 General considerations

Testing both the actual version with only one team playing and the other with two squads racing to develop first the product made me observe the resulting reactions and attitudes towards the game.

In the first two sessions, where one team had to deploy the product, the behaviour of the gamers was focused in learning and understanding all the process symbolized by Scrumble, which made them pay attention more on *why* they were doing something.

In the third and last session, the approach changed from theoretical to practical: the skills learned during the previous plays have been used to actually execute and simulate a product development, where the focus switched to the search of *how* they could do something in a more efficient way.

Final retrospective

Camilla: *"The team versus team version showed us that cooperation is fundamental when the time available to make decisions is limited"*.

Fausto: *"Much better the second version, you better understand what is cooperation and team is freer to act in a situation where challenge makes everything more dynamic. In addition to it, modify the product backlog has been very formative but also mentally hard"*.

Marco: *"This way of playing revolutionizes the concept of priority in a dynamic working ambient: what at first may seem necessary can be passed at any time by new"*

desires and trends”.

Alice: *”I liked how the competitiveness made us engage much more in solving blocking situations and last time problems, the possible card effects are a true example of the fact that anything can happen while working on a project”.*

Federico: *”Besides the sense of competitiveness, I found really interesting those in-game quizzes in the way in which they kept us concentrated reminding us something we may forget about Scrum”.*

Sara: *”Thanks to this version I felt much more the idea of playing an involving game, which forced us to plan a good strategy and put an effort to win over the other team”.*

5.3.2 Analysis with GQM

Learning

Looking over the learning process, I retain that its goals has been fully achieved: in the last session players rarely asked for help, they truly identified their selves in a team of developers, forgetting they were just playing a board game.

The concepts of Scrum work, team and artifacts have been assimilated, providing them a good knowledge basis in a fun and engaging way.

Practice

Summing up all the three sessions, I saw that by practicing together, the team was increasingly uniform in evaluating the complexity of the stories and their priority. The competition made the team be constantly aware of the technical debt, even though also in the previous session has been efficiently domesticated.

Cooperation

The members in both teams worked together without any unpleasant discussion, creating a solid cooperative confidence which made them play the game serenely and with fun.

I felt satisfied watching and listening to them while they were creating the User Stories: nobody was imposing his idea to the others, every thought was argued and discussed in order to evaluate its effectiveness; every player understood that there was not only one way to play the game, so no idea was the absolute best idea.

Motivation

The competitive version involved much more the players, determined to win the Scrumble challenge. During the retrospectives they actually reasoned about what could be improved in their way of acting, showing me that they weren't just passively playing a game I proposed to them, but actively engaged to overcome the opposing team.

Problem Solving

Following the experience from last sessions, players made an estimation of a feasible number of tasks, giving them the chance to calibrate the amount of User Stories composing the Sprint Backlog.

Technical debt was minimal at the beginning meaning that the team was efficient in letting almost no task to do at the end of the development day. This way to operate made the teams foresee and be prepared for possible unexpected events, which were represented by the card effect, that could increase the debt and turn it enormous if it was previously already high.

5.4 Results

After the end of the three game sessions, I summed up all the evaluations according to the GQM table and illustrated them in the following charts. The evaluation for each question of the GQM was on a scale from 1 to 5, where 1 meant really bad, according to the context of the question, and 5 excellent.

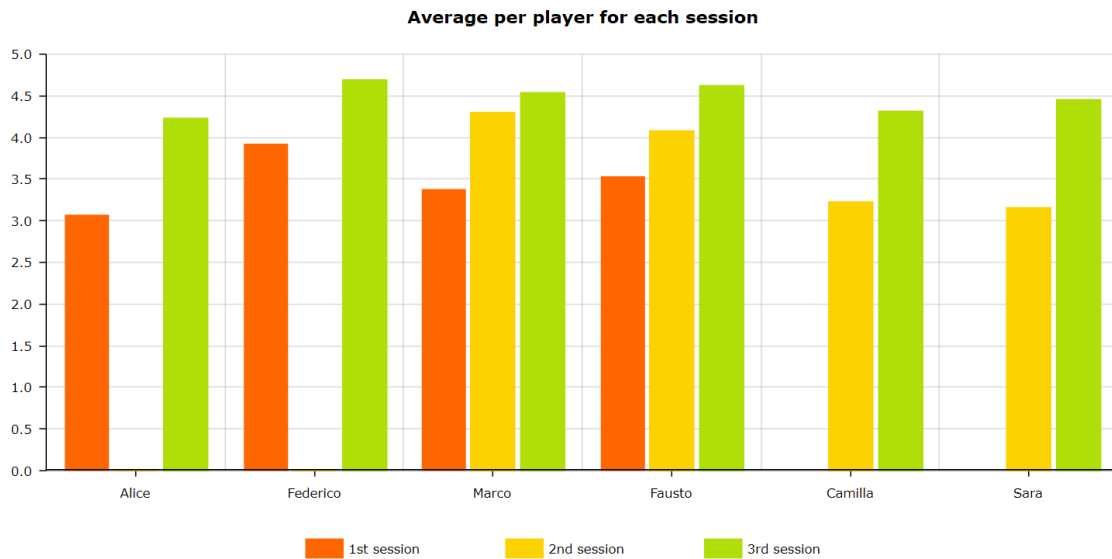


Figure 5.1: Average value of every player for each session

The first one shows the personal growth of each team member through the sessions, considering that the majority of the team played two sessions and the rest all of them. As we can clearly see, concerning the guys who played in only two meetings, the heightening between the sessions is evident: this could mean that the new players aren't affected so much by the old ones, even if the latter ones keep improving their team building skills, raising then the quality of the game performance by leading it.

Taking a look at the last session, it is possible to see that the average value of

each player tends to flatten almost at the same point: the experience now owned by every one, enabled the team to grow up together; the gamers discussed a lot, acted more natural, they felt more confident and they were more aware of the big view of the game.

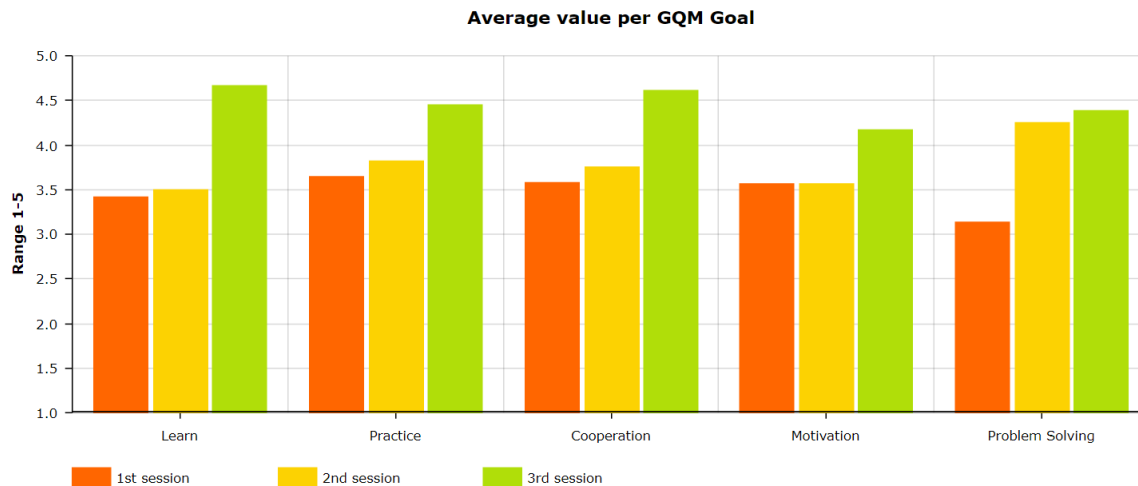


Figure 5.2: Average value of every GQM goal for each session

The second chart represents the observation and variations of the GQM goals I wanted the team to improve, namely the team building skills necessary to deal with the agile development world. What it seems for all the goals (except problem solving), is that the introduction of new members in the second game brought it to a sort of "starting level", as we could refer to Brooks' law, which states that "adding manpower to a late software project makes it later" [27]. On the opposite, regarding problem solving skills, the influence of more expert team members, enormously supported the new ones: playing the game the first time and understand the mistakes that led to a non completion of the game helped the players develop a concrete strategy that could be easily shared and assimilated by the newbies.

Chapter 6

Conclusion and future implications

Agile development is widely affirmed as a software projecting model, a model in which team work qualities outperformed the traditional and tight attachment to processes and detailed plans. In this scenario, the support of Gamification and serious games let agile coaches fulfill their tasks making their teams actually enjoy what they are learning and experimenting.

We had the chance to see how powerful a game can be to support the growth of team building skills, providing a complete working process within a playful and enjoyable surrounding. Indeed, Scrumble is a serious game which explores Scrum dynamics and permits a team to build its own product in the manner a real Scrum team does. Eventually, Scrumble could be a possible teaching instrument to adopt by university professors in a course which includes software engineer concepts and methods.

All these processes can be efficiently done in a remote environment, with the support of great technologies and softwares created to minimize the differences between physical and virtual meetings. This opportunity allows people from disparate

region, or even more countries, to follow a solid and satisfying path of team building learning, also during difficult periods as the one we are living in right now.

In this perspective, a new study could relate to this one by analysing the distinctions and similarities between a path in which Scrumble sessions are played remotely, and another one where the board game is played face-to-face in the same room.

Appendix A

GQM table compilation

In order to make the study repeatable and, eventually, expandable, I provide the table I used to evaluate the players throughout the three sessions, with some hints useful to balance the estimation. Every question receives a value from 1 to 5, where 1 means really bad, and 5 excellent: the correct value is up to the Scrum Master or whoever decides to keep track of the gamers actions.

GOAL	QUESTIONS	EVALUATION	Player 1	Player 2	Player 3	Player 4	Player 5	Player 6
Learn	Q1	1 = no idea of the Scrum roles 5 = perfect knowledge of the roles and their jobs						
	Q2	1 = couldn't repeat the game 5 = could play the game as a Scrum Master by himself						
	Q3	1 = totally lost 5 = leads the game driving the other players						
Practice	Q4	1 = feels the game is unrepeatable 5 = feels the game could be played in any situation						
	Q5	1 = 0 to 3 stories 2 = 4 to 6 3 = 7 to 9 4 = 10 to 12 5 = 13 to 15						
	Q6 ONLY DEV TEAM	1 = abnormal difference from the other players 5 = coherent and uniform with the group most of the time						
Cooperation	Q7	1 = never speaks with the other players 5 = talks friendly to anyone in every situation						
	Q8	1 = never puts effort in doing something 5 = every time is willing to understand what is going on						
	Q9	1 = never asks for an opinion 5 = wants to discuss about every topic						
Motivation	Q10	1 = not involved by the game 5 = always makes sure everyone is on point						
	Q11 ONLY FOR PO	1 = poor/absent advices 5 = wise and helpful suggestions when is required						
	Q12	1 = doesn't express opinions during retrospective 5 = feels the retrospective fundamental to express opinions						
Problem Solving	Q13	On the game board, if the debt pawn is on the lowest stage, the evaluation is 5, for every higher stage it decreases by 1						
	Q14 ONLY DEV TEAM	Calculate the average of tasks left for each sprint: 1 = 21+ 2 = 16-20 3 = 11-15 4 = 6-10 5 = 0-5						
	Q15 ONLY FOR PO	Same evaluation as G14 for the PO						

Figure A.1: Sheet of evaluation for the Scrumble GQM table

As shown on the figure, some evaluations are specific for the product owner and others for the development team, considering the different roles they cover as members of a Scrum team.

Bibliography

- [1] Agnes Moors Jan De Houwer Dermot Barnes-Holmes. “What is learning? On the nature and merits of a functional definition of learning”. In: *Psychon Bull Rev* (2013), pp. 631–642.
- [2] Andrew B. Barron - Eileen A. Hebets - Thomas A. Cleland - Courtney L. Fitzpatrick - Mark E. Hauber and Jeffrey R. Stevens. “Embracing multiple definitions of learning”. In: *Trends in Neurosciences, Vol N.38* (2015), pp. 405–407.
- [3] Arthur B. Markman Douglas L. Medin Brian H. Ross. *Cognitive Psychology*. Harcourt College Publishers, 2001.
- [4] Ian H. Witten. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kauffman, 2005.
- [5] Dorothy Lucardie. “The impact of fun and enjoyment on adult’s learning.” In: *Procedia - Social and Behavioral Sciences 142* (2014), pp. 439–446.
- [6] Micheal Beer. *Organisational Change and Development : A systems view*. Goodyear, 1980.

- [7] R. Wayne Boss. “Team building and the problem of regression: the Personal Management Interview as an intervention”. In: *Journal of Applied Behavioral Science* (1983), pp. 67–83.
- [8] M. Lindvall D. Cohen and P. Costa. “Agile Software Development”. In: *Data & Analysis Center for Software* (2003).
- [9] Conrad Weisert. “Waterfall methodology: there’s no such thing.” In: (2003).
- [10] Kent Beck et al. “Manifesto for agile software development”. In: *Agile Alliance* (2001).
- [11] Jeff Sutherland Ken Schwaber. *The Scrum guide*. Scrum Alliance, 2001.
- [12] Sumeet Madan. *Understanding Of The Definition Of Done*. 2019. URL: <https://www.scrum.org/resources/blog/done-understanding-definition-done>.
- [13] R. Khaled S. Deterding. “Gamification: Toward a Definition”. In: *CHI* (2011).
- [14] B. Eric M. Cathie. “Redefining Gamification”. In: *IADIS Interfaces and Human Computer Interaction 2012* (2012).
- [15] P. Backlund T. Susi M. Johannesson. “Serious Games - An Overview”. In: *Technical Report HS- IKI -TR-07-001* (2007).
- [16] Alan McClean. “An Exploration of the Use of Gamification in Agile Software Development”. In: *Masters Dissertation* (2015).
- [17] Alexey Krivitsky. *Scrum Simulation with LEGO Bricks*. 2012.
- [18] A. F. Borgatto C. G. Wangenheim R. Savi. “SCRUMIA—An educational game for teaching SCRUM in computing courses”. In: *The Journal of Systems and Software* (2013), pp. 2675–2687.

- [19] H. D. Rombach V. R. Basili G. Caldiera. “The Goal Question Matric approach”. In: *Encyclopedia of Software Engineering* (2002).
- [20] Aleksandr Tarasov. “Assessing Job Satisfaction of Software Engineers Using GQM Approach”. In: *Software Technology: Methods and Tools* (2019), pp. 121–135.
- [21] Thomas Buergi Margaret Oertig. “The challenges of managing cross-cultural virtual project teams.” In: *Team Performance Management 01/2006* (2006), pp. 23–30.
- [22] Pyxis Technologies. *Scrumble Game manual*. 2014.
- [23] Viviane Santos Frederico Oliveira Alfredo Goldman. “Managing Technical Debt in Software Projects Using Scrum: An Action Research”. In: *2015 Agile Conference* (2015).
- [24] Zhi Lee. In *Sprint Planning, How Do You Prioritize the Requirements in the Backlog?* 2017. URL: <https://dzone.com/articles/in-sprint-planning-how-do-you-prioritize-the-requi>.
- [25] Gaurav Dhooper. *Understanding the Importance of Retrospectives*. 2019. URL: <https://www.projectmanagement.com/articles/544911/Understanding-the-Importance-of-Retrospectives>.
- [26] David Mole Sandy Mamoli. *Creating Great Teams: How Self-Selection lets people excel*. 2015.
- [27] Jr. Frederick P. Brooks. *The Mythical Man-Month*. Addison-Wesley, 1975.