Author(s): Ghanbari, Hadi; Similä, Jouni; Markkula, Jouni

Title: Utilizing online serious games to facilitate distributed requirements elicitation

Year: 2015

Version:

Please cite the original version:

Accepted Manuscript

Utilizing Online Serious Games to Facilitate Distributed Requirements Elicitation

 Hadi Ghanbari ,   Jouni Similä ,   Jouni Markkula

Please cite this article as: Hadi Ghanbari ,   Jouni Similä ,   Jouni Markkula , Utilizing Online Serious Games to Facilitate Distributed Requirements Elicitation, *The Journal of Systems & Software* (2015), doi: 10.1016/j.jss.2015.07.017

Highlights

- Online serious games are used to facilitate distributed requirements elicitation.
- Interactive games enhance collaboration and communication between project members.
- Serious games raise individuals' confidence to engage in requirements elicitation.
- Using serious games can improve both quality and quantity of software requirements.
- Serious games specially enhance the performance of less-experienced stakeholders.

Utilizing Online Serious Games to Facilitate Distributed Requirements Elicitation

Hadi Ghanbari[1]

Department of Information Processing Science, University of Oulu, Oulu, Finland
Email: hadi.ghanbari@jyu.fi
Telephone: +358408053337

Jouni Similä

Department of Information Processing Science, University of Oulu, Oulu, Finland
Email: jouni.simila@oulu.fi

Jouni Markkula

Department of Information Processing Science, University of Oulu, Oulu, Finland
Email: jouni.markkula@oulu.fi

Abstract Requirements elicitation is one of the most important and challenging activities in software development projects. A variety of challenges related to requirements elicitation are reported in the literature, of which the lack of proper communication and knowledge transfer between software stakeholders are among the most important. Communication and knowledge transfer are becoming even bigger challenges with the current increase in globally distributed software development projects due to the temporal, geographic, and sociocultural diversity among software stakeholders. In this study, we propose a new approach to requirements elicitation, which employs online serious games for gathering requirements from distributed software stakeholders. The feasibility and effectiveness of the proposed approach were evaluated in an empirical study with encouraging results. These results especially reveal that our suggested approach enables less-experienced individuals to identify a higher number of requirements. Our results also reveal that for the majority of subjects, especially individuals with less technical experience, this approach was a pleasant and easy way of participating in requirements elicitation. Based on these results we suggest that using online serious games not only enhances innovation and creativity among end-users but also facilitates collaboration and communication among software stakeholders. Implications for both research and practice are considered.

---

[1] Corresponding author. Present address: University of Jyvaskyla, Department of Computer Science and Information Systems, P.O.Box 35, FI- 40014, University of Jyvaskyla, Finland

1. Introduction

Despite all the financial and human resources spent on software development projects, the majority of these projects fail or encounter severe challenges. A study by the Standish Group [1] showed that 24% of software projects failed before completion and 44% of projects faced serious challenges, such as exceeding the budget and delivery time or missing specified features and functions. It has been widely reported by previous studies that inadequate and incomplete requirements elicitation is one of the main reasons for failure in software development projects [2-6]. Indeed, inadequate requirements elicitation may lead to implementing wrong or unnecessary features and missing important or necessary functions, which can cause substantial human and financial damage [1]. Since it is often difficult for end users and customers to articulate their needs [7] and to describe their business processes properly [8], it is almost impossible for development teams to define system boundaries and to produce a consistent and complete set of software requirements to be implemented [2, 3, 8, 9]. Poor communication skills and lack of proper technical and domain knowledge are the main reasons for users not being able to describe their needs and expectations clearly [3, 5, 8, 10, 11]. Additionally customers' expectations are usually in terms of general business needs to be satisfied rather than specific system functionalities. Thus, it is less likely that they think about and ask for non-functional requirements such as performance, security, and maintainability [3, 12].

In Global Software Development (GSD), where software companies use distributed development teams to be more competitive by increasing their productivity and reducing their development costs, requirements elicitation becomes even more problematic [13]. Distributed requirements elicitation is highly challenging, especially due to lack of synchronous communication imposed by temporal and geographic distance [14-16]. Furthermore, in GSD, poor requirements communication caused by sociocultural differences between system stakeholders can cause misunderstandings and requirements conflicts [17].

Traditional software development methods suggest using a complete and formal requirements elicitation process at the beginning of a project. During this process, development teams consult system stakeholders to identify their expectations, system boundaries, and business needs [12, 18]. Following this process, a detailed requirements document is produced that preserves detailed system specifications and, at the same time, is used as the means of communication [9]. Although this approach is still widely in use, it might lead to inadequate requirements elicitation, especially in today's business environment. Usually, customers' requirements are ambiguous at the beginning of the project [8-10], and in later stages of software development when stakeholders and developers have gained more understanding of the system functionality, requirements change is inevitable [18]. In addition, in the modern marketplace software projects often have very tight schedules, and developers are under constant pressure to deliver software as soon as possible. Therefore, they cannot spend sufficient time on a complete requirements elicitation process [18].

In contrast, lightweight and iterative software development methods such as *agile methods* highly rely on effective informal communication between businesspeople and developers during system development cycles [6, 9, 10, 12, 19, 20]. Software stakeholders collaborate to obtain an understanding of

high-level requirements at the beginning of projects [21]. The continuous interaction between customers and the development team throughout the project allows them to identify and add more requirements in the following iterations [9, 21]. Although agile practices alleviate some of the challenges of traditional requirements elicitation challenges—including continuous changes in requirements, heavy documentation, and a tight project schedule [9]—these practices also raise critical issues for agile teams. Lack of an initial estimation of project costs and the increasing rate of requirement change are two of the challenges confronting agile methods [9, 12, 18]. In addition, inadequate customer participation and lack of proper knowledge among agile teams makes requirements communication challenging in agile projects [6, 9, 10, 19, 22].

Despite all the above-mentioned challenges, using agile practices in globally distributed projects has become increasingly popular in the software development community [23-25]. The results of a survey conducted by [25] shows that, out of the 3,501 respondents who participated in this survey, more than 76 percent of them use agile practices in distributed projects. These results show that the number of distributed teams using agile practices is more than doubled that found by same survey conducted in 2012. Although agile methods rely heavily on on-site customer participation, applying agile practices in distributed projects can be beneficial, providing software companies with a combination of the flexibility of agile methods and the cost-effectiveness of GSD [16]. Even though it has become clear that distance makes communication more problematic in distributed projects [15, 23, 26-28], applying *information and communication technology* (ICT)-mediated tools such as *wikis, videoconferencing*, etc., is considered to be a key solution for supporting requirements elicitation in distributed agile projects (e.g., [14, 29]). However, due to challenges associated with communication between distributed stakeholders [14-16], there is a constant need for new and effective methods that encourage customers and end users with different levels of technical and domain knowledge to engage in distributed requirements elicitation. The focus of these methods must be on facilitating proper communication, collaboration, and knowledge transfer between system stakeholders, which enables individuals to bring up their needs and expectations more easily.

Recently, a number of studies have examined the importance of creativity-fostering techniques in improving the quality of the requirements elicitation process (e.g., [30-33]). On the other hand, it has been suggested by a number of studies that playing games is an interesting method of collaboration and an effective way to enhance innovation and creativity among individuals [34-37]. Following these studies, we propose that utilizing online serious games is an effective method of combining creativity-fostering techniques and ICT-mediated communication for facilitating requirements elicitation in distributed projects. We investigate this by answering the following research question:

How does applying online serious games affect the quality of the requirements elicitation process in distributed projects?

To answer this research question empirically, we utilized a set of interactive online serious games for conducting a distributed requirements elicitation process. The results of our experiment, as discussed later in this paper, indicate that using online serious games is an effective method for improving distributed requirements elicitation.

The rest of this paper is structured as follows: Section 2 provides an overview of the impacts of innovation and creativity techniques on requirements elicitation processes. In Section 3, we discuss

5

serious games and briefly describe our suggested requirements elicitation approach. In Section 4, the experimental research settings are described, followed by a report of the research results in Section 5. In Section 6, a discussion of research findings and their academic and managerial implications is provided, along with limitations of the study and suggestions for future research. Finally, Section 7 provides the conclusion of the paper.

2. Related work

In today's highly competitive and turbulent marketplace, the ability to provide innovative products and services is a key competitive factor in organizations' survival [16]. Famous innovation models such as *open innovation* [38] and *user-centered design* [39] promote collaborative knowledge transfer between individuals and companies for the design of innovative products. As it is suggested by [39], system users are valuable sources of innovation, and it is a big mistake to consider them as abstract consumers. Many ICT firms try to promote innovation and gather innovative ideas to provide new value to their customers by making creative changes to their products [40]. Innovation processes take place in the social surroundings of human everyday life, and new information and communication technologies extend the range of people's social lives to a global dimension [41]. Hence, customers are able to easily transfer knowledge about solving a problem, prototyping an idea, or using an existing product in a way that was not originally intended with other consumers all around the world. For example, customer support forums enable individuals from all around the globe to easily ask questions, hold discussions, and transfer knowledge about a product or service, its advantages and weaknesses, or even how it can be improved [41].

Requirements elicitation is a creative process in which software stakeholders collaborate and identify a set of ideas, which are expressed as user requirements [31]. Many studies propose that innovation and creativity-fostering techniques are effective methods for improving the quality of the requirements engineering process (e.g., [30-34]). For instance, Maiden and Robertson [31] described a scenario-driven process called *RESCUE*, which employs collocated creativity workshops to facilitate requirements engineering. In this method, a requirements engineering team first specifies the system boundaries, and then the creativity workshops, which last 2 consecutive days (8 hours), take place to acquire a more precise and complete set of requirements from the system stakeholders. The results from [31] indicate that incorporating creativity workshops encourages creative thinking among software stakeholders. This method helps individuals with different levels of knowledge and understanding participate in the requirements elicitation process and identify, articulate, and negotiate their requirements. In addition, RESCUE helps individuals to understand the requirements identified by other stakeholders [31].

Mich et al. [32] defined an innovative process of requirements elicitation called the *EPM Creative Requirements Engineering Technique* (EPMcreate). This technique can be applied in a creative requirements elicitation session that involves two system stakeholders or two classes of system stakeholders. This technique enables system analysts to generate creative ideas by looking at system requirements from a combination of stakeholders' viewpoints. The effectiveness of this process was investigated through a set of experimental studies [32, 33] by measuring the quantity and quality of the requirements identified by the research subjects. Even though these studies provided descriptive statistics

to indicate that using EPMcreate improved the quantity and quality of requirements identified by the research subjects, this effectiveness was not statistically significant.

Finally, Aaen [34, 35] suggested a new method called *Essence* and a new facility called the *Software Innovation Research Lab* (SIRL) for facilitating software innovation and creativity. The strategy behind Essence is based on activities similar to role-playing games, and therefore this method consists of creative methods, tools, and techniques [34, 35]. In addition, SIRL provides a flexible set of equipment (e.g., interactive screens, desktop computers, a server, idea generation software, ICT-mediated communication mediums, etc.) for the complete software development life cycle [34]. Available infrastructure in the lab enables the development team to use the distributed innovation process through off-site communication. Essence and SIRL were evaluated in an experimental study [34], but the author noted that innovation process support was not fully developed, and further exploratory experiments are needed to provide support for creativity in developing software [34, 35]. According to the above-mentioned techniques, we suggest that there is a need for further research to investigate the impact of innovation and creativity-fostering techniques on requirements elicitation.

Customers' requirements are usually communicated with development teams through interviews, questionnaires, and workshops. Although the method to be used plays a central role in the requirements elicitation process, project failure rates indicate that current requirements communication methods are not sufficient enough for identifying all types of user requirements [7]. This issue is even more problematic in distributed projects, for which the development process takes place across dispersed geographical locations [23]. Usually, multinational and cross-cultural software stakeholders use tools to participate in different development stages, including requirements elicitation, across national boundaries with different time zones [17, 28]. Therefore, new methods that concentrate on improving the quality of interaction and requirements communication between distributed software stakeholders can be beneficial for the software development community.

On the other hand, recent studies indicate that applying agile practices has become increasingly popular in distributed software development projects [23-25]. This might be due to the fact that by using agile practices in distributed projects, software teams can benefit from both the flexibility of agile methods and the cost-effectiveness of GSD [16]. Therefore, it has become apparent that any new method to be developed must have different characteristics to suit both agile and distributed software development projects. Primarily, it must be applicable online to enable off-site communication between distributed stakeholders [14, 29]. Both methods mentioned earlier in this paper, RESCUE and EPMcreate, incorporate collocated sessions, which means that these methods are not efficient for distributed requirements elicitation. Second, the method must be lightweight and easy to use to make sure that projects do not drop it due to lack of time [34]. In addition, the method should be attractive and pleasant for its users to further their motivation and improve the quality of the results [32, 34]. Finally, the method must enable developers to easily keep track of users' requirements for documentation purposes (e.g., knowledge transfer) and for future purposes (e.g., system maintenance or improvement). Based on the need for better requirements elicitation methods and for their desirable characteristics, we suggest that utilization of online serious games might be a beneficial approach for facilitating and improving the requirements elicitation in distributed context.

7

3. Using online serious games in distributed requirements elicitation

Our suggestion for a new approach to requirements elicitation in the distributed context, which we call "Innovation" requirements elicitation, is based on using a set of interactive online serious games. Using online serious games is suggested for two reasons. First, playing serious games enhances individuals' creativity, and therefore customers are able to provide more innovative ideas about the software to be developed [34, 36, 37]. Secondly, use of interactive online games as a rich ICT-mediated tool facilitates collaboration and off-site communication between software stakeholders, which might lead to effective requirements communication in distributed projects [14, 29].

Playing games is an interesting method of practicing teamwork and improving interaction among individuals because it presents individuals with a new way of thinking, getting and transmitting knowledge, and socializing [37]. Playing games can encourage people to engage in non-entertainment activities that are considered boring. Additionally, playing games provides people with an opportunity to learn new concepts and improve their skills by experiencing and making mistakes.

The term *serious games*, which has become increasingly popular during the last decade, refers to games with a set of cognitive properties [37] that provide individuals with new ways of thinking and transferring knowledge [42]. As has been suggested by Zyda [43], application of "games and simulations technology to non-entertainment domains … results in serious games." Such games have been increasingly used for educational and training purposes in various fields, including management, defense, healthcare, and software engineering [42, 44]. Serious games are mainly used to assist individuals in developing different skills such as communication, collaboration, problem solving, and decision making [44]. In addition to their pedagogical purposes, applying serious games is an interesting and effective method for adopting innovation and applying creativity to software projects (e.g., [34, 36, 37]).

A number of studies use serious games to simulate different stages of software development (e.g., [37, 45, 46]). Integrating serious games in software projects enables teams to evaluate the level of understanding among stakeholders by analyzing their collaboration experience [37]. As mentioned by [37], serious games increase confidence among individuals because their mistakes do not have any serious consequences and they will not be judged according to their actions and decisions.

In addition to these studies, in practice there are a few examples of using collocated and online serious games to identify customers' needs and expectations. However, to the best of our knowledge, no academic research has indicated the effectiveness of serious games in improving the quality of requirements elicitation in a distributed context. To fill this gap, in this study we investigate the effects of serious games on requirements elicitation by applying a set of online serious games in distributed requirements elicitation.

According to our research goals, the serious games to be used must be applicable to distributed agile software development projects. This means that these games must be easy to use, lightweight, cost-effective, and, most importantly, accessible online. In this study, we employ a set of interactive online serious games called Innovation Games®, which refers to a set of serious games introduced by [36]. Hohmann [36] argues that these games are powerful and simple qualitative research and problem-solving techniques. Basically, in these games, a group of system stakeholders participates in collocated or online

sessions and plays a set of directed games to generate ideas or provide feedback about a product or service. Although a considerable number of organizations have been using these games, to the best of our knowledge no academic research has either suggested using these games or explored their effectiveness in distributed requirements elicitation.

A web-based tool designed by Innovation Games® enables geographically dispersed stakeholders to participate in online game sessions. In addition, these distributed players use an integrated chat and whisper tool as the main communication channel during the game sessions. This chat facility also enables the development team to keep track of players' interactions and communication for later purposes. According to the objective of this research, we decided to use the game *Prune the Product Tree* [36] for requirements elicitation to gather user needs and ideas about software to be developed. However, later during the empirical research, we decided to use another game called *Buy a Feature* [36] for requirements prioritization and negotiation. These games are described briefly in the following sections.

3.1. Prune the Product Tree

In real life, gardeners prune trees to increase the quantity and quality of products. The more balanced the tree, the better the fruit. Keeping this in mind, the game *Prune the Product Tree* [36] aims to shape a balanced product roadmap that is organized properly for successful completion of a project in all aspects of the production process. In this game, a small group of 5 to 10 customers collaborate and try to shape their desired product in the form of a tree (i.e., the system to be developed) that consists of different parts, including limbs (i.e., system functionalities) and fruits (i.e., system features). The root system is shown as the trunk, and the main thick limbs represent system functionality areas. The distance between the trunk and the edges of the tree represents the project life cycle, with the area near the trunk considered as the initial stages of the project and the area near the edges considered as the final stages of the project. As can be seen from Appendix A, three different priority levels are defined for the sample tree (i.e., low, medium, and high).

A participant writes a short description for each feature on an index card, which represents a fruit or a leaf, and places the card on the tree. The leaves or fruits closer to the trunk indicate requirements with higher priorities, which should be delivered as soon as possible. For example, imagine a software development project for creating a website. The trunk here represents the website itself, and *User Registration* and *Photo Gallery* can be two main functionalities. If a user asks for *Add New User* as a high-priority requirement, it should be placed on the *User Registration* limb and in the area closer to the trunk. If another user asks for *Photo Sharing* as a low-priority feature, it should be placed on the *Photo Gallery* limb and closer to the edge of the tree.

The online version of the game consists of the game area, a chat and whisper facility, and a palette of items (e.g., fruit, leaves, index cards). To add a new requirement, players can easily drag an item from the palette onto the tree and enter a label and a short description for the item. The interactive online game allows players to see others' decisions and actions in real time and use the chat and whisper facility to negotiate with others (See Appendix A).

In this game, while users are articulating their expectations, other players have the opportunity to ask questions and discuss features and their priorities. Thus, the players can reframe their requirements or define new ideas based on these discussions. At the end of the game, the development team can analyze

9

the requirements of the system to be developed based on the descriptions provided for each feature and the discussions recorded between the participants.

In a way, *Prune the Product Tree* resembles *User Stories,* which is a common agile requirements elicitation practice used as an alternative for a detailed requirements document. In this practice, the customer representative provides functional requirements in the form of short and abstract user stories, usually on index cards or sticky notes [10, 17]. This short description generally represents a valued functionality that satisfies customers' business needs rather than a technical or user interface component.

3.2. Buy a Feature

Choosing the right set of features to be developed in each release of software enables the development team to keep customers satisfied and to avoid expensive changes. Often, it is the case that if you ask customers, they want all the features to be developed in the next release. The game *Buy a Feature* [36] enables the development team to identify and deliver customers' most valuable features in the subsequent product release.

In this game, a small group of 4 to 9 customers collaborate to buy their desired features from a list of potential features. This list includes a set of requirements identified during the requirements elicitation phase and analyzed by the development team. Each feature on the list has a price, which can represent actual development costs, customers' business value, or anything else.

Each customer has his or her own budget in the form of play money. The amount of total money that the players have altogether is one-third to half of the entire system budget. In other words, each group's entire budget is not enough to purchase all the features available on the list. Often some important features are priced so high that it is impossible for a single player to buy them individually. This encourages the players to collaborate and to pool their money in groups in order to buy a set of desired features, especially the more expensive ones [36]. The limited budget facilitates requirements negotiation among customers to make decisions about each feature and its necessity.

Imagine that in the website development case mentioned earlier, each player has €150 and the estimated cost for the *Photo Sharing* feature is €250, which means that the user who asked for this feature does not have enough money to buy it. As a result, the user must negotiate with others and convince them to invest €100 to buy this feature. In this way, players are able to choose the most valuable set of requirements that should be developed in the next release. In contrast, the development team can use these discussions to understand and analyze the reasons behind requesting certain requirements and as a result prioritize customers' needs based on their actual business value.

In the online version of this game, a list of items and prices is available as a table, and players can purchase items by making bids. If players would like to purchase an item, it is enough to click the appropriate cell under the name and enter the amount of money they would like to invest for each feature. Players can change or remove their bids at any time. Once an item is purchased, it is purchased for the entire group. Players can use the chat and whisper facility to negotiate regarding the items that others try to purchase (See Appendix B).

To improve the quality and effectiveness of these two games, it is recommended that only a limited number of players participate in each game session [36]. Therefore, in big projects, a large number of stakeholders can be divided into smaller groups, with a separate game session organized for each group.

10

Later, game facilitators or the development team can merge the results from different games together in order to identify and prioritize customers' requirements.

3.3. The advantages of using online serious games in requirements elicitation

Our suggested Innovation requirement elicitation method using interactive games can be expected to increase stakeholders' motivation to participate in the requirements elicitation process as well as encourage them to be creative and come up with more innovative ideas for the software to be developed.

In addition, our approach has a number of advantages compared to the above-mentioned creativity-based approaches utilized for requirements elicitation. Firstly, using online serious games enables geographically dispersed system stakeholders to collaborate during the requirements elicitation process. In contrast, both RESCUE and EPMcreate incorporate collocated sessions, which means these methods are not efficient for distributed requirements elicitation. Secondly, using online serious games is not considered to be time-consuming since the actual game sessions last less than 90 minutes and no large preparation efforts are needed for these sessions. In contrast, for example, in RESCUE, each creativity workshop lasts for almost 2 full days, which means that the system stakeholders have to spend a considerable amount of time participating in these workshops. Finally, our approach utilizes easy-to-use interactive web-based games that run on all types of Internet browsers and do not require extra hardware or software installation. In contrast, all the other methods discussed above require at least a physical location, furniture, and office supplies for conducting the requirements elicitation sessions.

4. Research design

Our suggested Innovation requirements elicitation method based on using online serious games is expected to facilitate user participation in distributed requirements elicitation and to improve the process and the results of the requirements elicitation. Following evidence-based software engineering principles, the proposed new method requires empirical validation and evaluation of its effectiveness before it can be adopted for professional purposes. Experiments are valuable tools in the software engineering discipline for "evaluating and choosing between different methods, techniques, languages, and tools" [47, 48]. Since experiments are usually conducted in highly controlled environments, they enable researchers to measure the effects of a treatment on the output variable by manipulating one or more input variables directly and systematically [47, 48]. Therefore, we planned and executed an experimental study to evaluate the effectiveness of the Innovation requirements elicitation method, as is discussed in the following sections.

4.1. Experiment design

To conduct our experimental study, we planned and executed a distributed agile software development project at the Department of Information Processing Science, University of Oulu, Finland. In this project, the subjects within two experimental groups performed a series of software development tasks for identifying and prioritizing a set of requirements for the software to be developed. The research design conducted can be considered as a controlled experiment, as the subjects were randomly assigned to

11

different experimental groups and the independent variables were either controlled or manipulated [47, 48].

Following [32 and 33], the effectiveness of the Innovation requirements elicitation method can be evaluated both quantitatively, by measuring the number of requirements identified by system stakeholders, and qualitatively, by measuring the feasibility and novelty of the identified requirements. To make this measurement meaningful, we decided to compare the effectiveness of the Innovation requirements elicitation method with an "Agile" requirements elicitation method that consisted of common agile practices.

The experiment design we used can be characterized as a one factor two treatments design [47, 48]. The factor is the requirements elicitation, while the treatments are the new experimental elicitation method and the commonly used Agile requirements elicitation method. In our design, each requirement was attributed to an individual subject who identified that requirement during the process of requirements elicitation. Thus, the dependent variables used for measuring the effect of the treatment were the number of requirements, number of new requirements, and number of feasible requirements identified by an individual subject.

4.1.1. Project settings

In each of our two experimental groups, 5 distinct teams with different software stakeholder roles and responsibilities were formed (see Table 1). The software engineering task presented for the teams was to identify and prioritize a set of requirements for the software to be developed. Because the participants were students, the software we chose to develop was *UniGuide*, which is an application that can be used as a guide on a university campus to provide useful information about events and locations to students. Since a prototype of this application was previously designed and tested in the same organization and due to the fact that none of the study subjects were aware of the existence of such an application prototype, the original features of this application could be used to evaluate the quality and the quantity of the requirements specified in our experiment [32, 33].

Table 1. Project teams and their responsibilities.

Team	Responsibilities	Work Experience
Headquarters	Controlling and coordinating the development process, interacting with the Customer and Development teams to collect requirements and to deliver the final prototype.	More than 5 years
Development	Participating in requirements elicitation process and providing technical consideration to customers, analyzing user requirements, estimating the effort needed to implement each item, and preparing the software prototype.	Between 1 and 5 years
Customer	Interacting with the Headquarters team, providing requirements, participating in requirements negotiation, and involving User teams in the process.	Less than a year
User 1, User 2	Providing user requirements, participating in requirements negotiation, and evaluating the software prototype.	Less than a year

The project setting was specifically designed to include the main contextual challenges of GSD, including cultural and temporal differences between software stakeholders. The sociocultural distance between participants available in GSD projects was taken into account by recruiting the subjects from a student population consisting of 14 nationalities. In addition, the official language used during the project was English, even though none of the participants were native English speakers.

Additionally, the project teams participating in the experiment were assigned to four imaginary time zones in order to simulate temporal and geographical distance (see Figure 1). According to this arrangement, the simulated geographically and temporally dispersed teams were obliged to use synchronous and asynchronous ICT media to communicate with other teams. Face-to-face communication with other teams was forbidden. Furthermore, as daily work hours vary by country and calendar differences (e.g., daily work hours, lunchtime, and weekends) complicate global coordination between geographically dispersed teams, the teams had to coordinate their communication and interactions with other teams according to their work hours. Figure 1 indicates the work hours in each site using UTC, the absolute reference point of time.
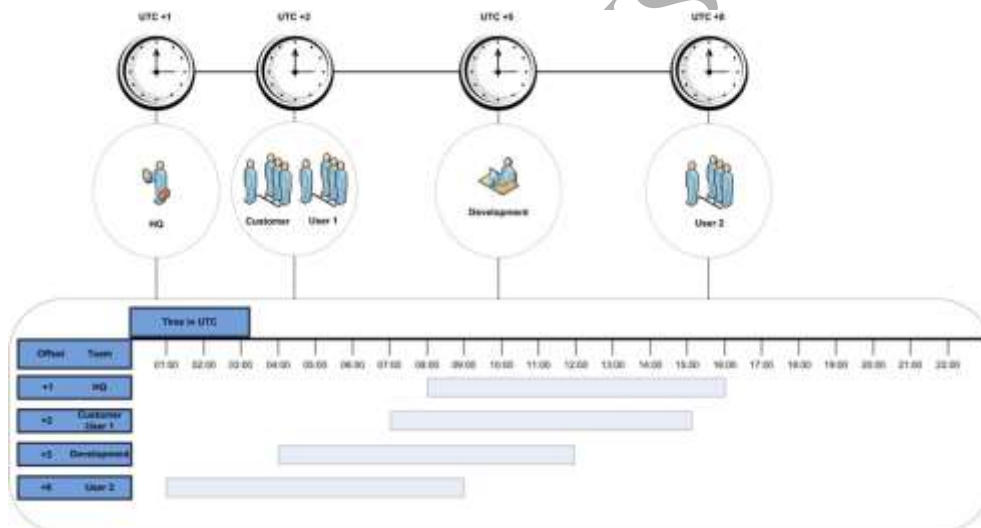

Figure 1. Chart of teams' imaginary time zones based on examples provided by [49].

In this project, we assumed that all teams had the same amount of work time from 9:00 to 17:00 in their imaginary local time. Therefore, before planning online meetings, the teams from different time zones had to decide when would be the best time to have a video or voice conference considering the working hours of both simulated locations[2].

---

[2] For example, the Headquarters team was located in the Coordinated Universal Time (UTC) +1 time zone (e.g., the Czech Republic), and the Development team was located in the UTC +5 time zone (e.g., India). In this case, due to geographical distance, face-to-face communication between teams might be impossible or expensive, which would force teams to communicate via e-mail, phone, videoconferencing, or teleconferencing. However, since there is a 4-hour time difference between these two sites, synchronous communication over ICT channels is problematic, as there are only 4 overlapping work hours between these simulated locations. The time difference was even more problematic, for example,

13

4.1.2. Participants and team division

The subjects of the experiment were recruited from 63 participants registered in the course Software Development in Global Environment, offered for international master's students at the University of Oulu. During this course, the students learn about different aspects of globally distributed software and systems development projects. At the beginning of the course, the goals of the study were introduced to the students. After the introductory session, 47 of the students volunteered to participate in the project and be subjects of the experiment (see Figure 2).

The subject group was relatively homogeneous regarding academic achievement. However, the participants had different levels of work experience in software development. To increase the validity of the experiment and minimize the effect of work experience on the requirements elicitation, we controlled this independent variable of the experiment in the following way. Using the background information provided by the research participants, we grouped them into three subgroups based on their previous work experience in the field of software development and their familiarity with global software development and agile methods: Professionals (i.e., individuals with more than 5 years of relevant work experience), Intermediates (i.e., individuals with more than 1 year and less than 5 years of work experience), and Beginners (i.e., individuals with no work experience or with only academic knowledge of software engineering). The Professionals subgroup included 8 students, the Intermediates subgroup 10 students, and the Beginners subgroup 29 students.

In the next stage, the students from each work experience subgroup were randomly assigned to different teams in the following way:

- Subjects from the Beginners subgroup (i.e. 29 individuals) were assigned randomly to four User teams and two Customer teams. Five of these teams had 5 members each, and one team had 4 members. These teams had a central role in the requirements elicitation phase because they were the actual end users of the software and the ones with knowledge on how to use the system.

- From 10 subjects in the Intermediate subgroups, 9 were assigned randomly to two Development teams and the last subject was added to the Professional subgroup[3]. The Development team is responsible for technical aspects of the project, including analyzing and improving user requirements to feasible software requirements, estimating the effort required for developing each feature, and implementing the software prototype.

- Subjects from the Professional subgroup and one subject from Intermediate subgroup were assigned randomly to 2 Headquarters teams. The Headquarters team is responsible for controlling and coordinating the entire development process, including communicating with customer and developers, planning meetings and deliveries, and so on.

---

when the Customer team (e.g., Finland) and the User 2 team (e.g., China) had to communicate synchronously while there was only 1 hour of overlap between their work hours (see Figure 1).

[3] Since one of the Customer teams had 1 member less than other teams, this has been done in order to balance the total number of individuals who were supposed to participate in the requirements elicitation process (i.e. 19 individuals in each group).

When the above mentioned teams were organized, they were used to form two experimental groups: the Agile Group (AG) and the Innovation Group (IG). As it can be seen in Figure 2 below, each of these groups included 1 Headquarters team, 1 Development team, 1 Customer team, and two User teams. It must be noted that when the experimental groups were formed during the first week of the project, four of the students from the IG decided not to participate in the project (i.e. 1 subject from Development, 1 subject from User 1, and 2 subjects from User 2 teams). For that reason, the number of subjects who participated in the requirements elicitation process was not the same for both groups (see Figure 2).
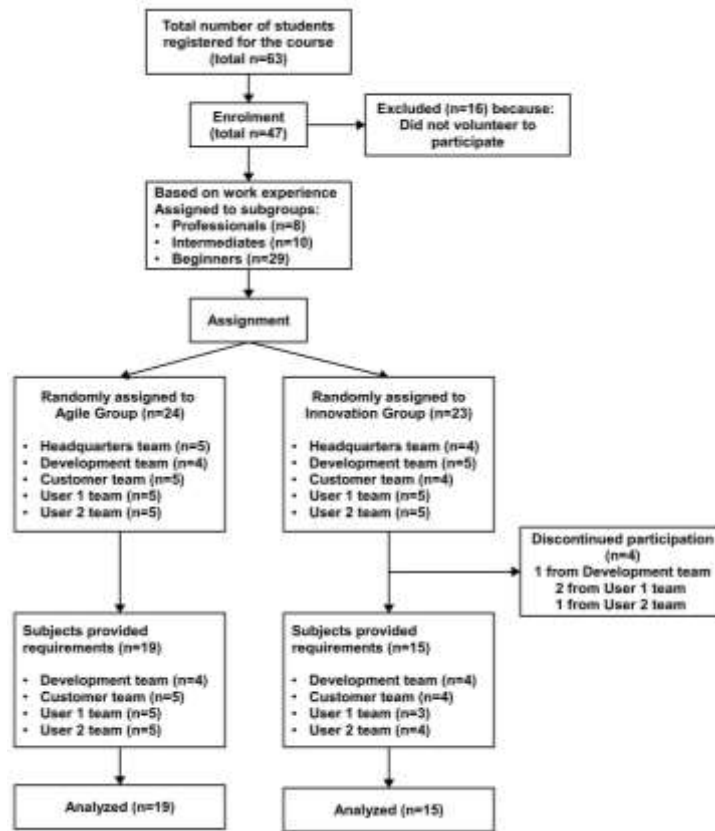

Figure 2. The flow of participants through different stages of the experiment.

As can be seen from Figure 2 above, data were gathered from only 19 subjects from the AG and 15 subjects from the IG for two reasons. First, the subjects on the Headquarters teams did not participate in the requirements elicitation process. Second, when the experimental groups were formed and during the first week of the project, 4 of the 23 students in the IG decided not to participate in the project (i.e. 1 subject from Development, 2 subjects from User 1, and 1 subject from User 2 teams). For that reason, the number of subjects who participated in the requirements elicitation process was not the same for both groups.

4.2. Experiment execution

This project took place over a period of six weeks during the 2012 spring semester (see Table 2). At the beginning of the project, an introduction session was held, and the project settings were described to the participants. Since the participants in this project had different levels of academic knowledge and professional skills, a short training session was conducted to provide an overview of agile methods and GSD. Later, two training sessions lasting approximately 1 hour each were conducted for each group

15

separately: A short introduction to the Agile requirements elicitation method including user stories, product backlog, and effort estimation including planning poker was provided to the AG, and an overview of the Innovation requirements elicitation method was provided to the IG. Each team chose a main contact person from the group who played the role of ambassador [17, 26] and was responsible for coordinating communication between the teams.

The project was divided into four iterations. At the beginning of each iteration, the teams received an email containing their assignment for that iteration and the necessary materials (Appendix C provides an example of the teams' weekly assignments). These assignments were designed according to the teams' roles in such a way that constant interaction and collaboration between the teams would be necessary to produce the final deliverables. At the end of each iteration, the Headquarters and Customer teams of each group had a review meeting. Course assistants participated in these meetings as observers. At the end of the project, a reflection seminar was conducted to discuss and evaluate the project outcomes and to officially conclude the project.

It must be mentioned that the researchers were involved in this project only as instructors and observers without any direct or indirect intervention in team decisions and their activities. To be more specific, the researchers only participated in the training sessions conducted at the beginning of the course and the final project review at the end of the course. In addition to this, the researchers participated in weekly review meetings at the end of each iteration and game sessions only as observers. The whole process of contacting other teams and planning the schedule for the meetings and requirements elicitation sessions was handled by Headquarter teams. Therefore, the Headquarter teams had complete freedom to negotiate and agree with other teams in regard to the day and time of the meetings and requirements elicitation meetings. Only after the decision about the day and time of the meetings had been made by the teams themselves, the researchers were informed by Headquarter teams.

Table 2. Project schedule.

Date	Project Milestone
March 13, 2012	Students filled out the online background questionnaire.
March 16, 2012 (Iteration 0)	The project introduction session was held, and a short training session was conducted to provide an understanding of subjects related to the project.
March 23, 2012 (Iteration 1)	The Headquarters team gathered a set of requirements provided by participants from the User teams, the Customer team, and the Development team.
March 30, 2012 (Iteration 2)	The Headquarters team conducted online requirements negotiation meetings between the User teams and the Customer team. The Development team estimated the effort needed for each product backlog item and started to design the software prototype.
April 13, 2012 (Iteration 3)	Headquarters conducted online meetings between the User teams and the Customer team to evaluate the software prototype and to reassess and prioritize the product backlog. The Development team finalized the software prototype based on feedback from the other teams.
April 20, 2012 (Iteration 4)	Students wrote individual reflection reports to provide feedback on the project settings and project challenges and their opinions about the methods used in the project.
April 27, 2012	A reflection seminar was conducted to officially conclude the project.

In general, the project was started by the Customer teams, which were responsible for ordering the software from the Headquarters. According to the project settings, each team was asked to deliver

different types of deliverables during the project[4]. The members of each group performed a series of requirements elicitation tasks under conditions identical in every respect except one: namely, the AG used the Agile requirements elicitation method, while the IG group used the Innovation requirements elicitation method.

4.2.1. Tasks performed by the AG

As mentioned in the previous section, the AG has received training on using common agile practices such as user stories, product backlog, effort estimation, and the prototyping as the main techniques for requirements elicitation. However, due to the distributed nature of the project, following [14, 15, 23] the AG teams were instructed to employ application sharing and ICT-mediated negotiation for distributed requirements elicitation and negotiation. Therefore, in addition to using email, chat, and videoconferencing tools, the members of AG used web-based word processor and spreadsheet provided by Google Inc., as a part of Google Drive service, in order to collect and organize their requirements. It is worth noting that according to [25] in 2012 spreadsheet was the most common tool used by agile teams.

In the first iteration, all subjects from the Customer team, both User teams, and the Development team had 1 week to think about the software to be developed and to provide their requirements in the form of user stories to the Headquarters team. Participants were asked to provide the name of the owner and the priority of the requirement (i.e., low, medium, and high) along with the requirement's description. This enabled the researchers to evaluate individuals' performance in identifying new user requirements and more importantly to analyze the data collected from the subjects. The Headquarters team gathered all the requirements provided by 19 subjects in the initial product backlog for further analysis.

In the next iteration, synchronous and asynchronous technologies were used to analyze, negotiate, and revise the initial requirements. Later, similar requirements were merged or removed, and complementary descriptions were added to the product backlog. The Development team estimated the effort needed to implement each feature. At the same time, the Customer team and the User teams negotiated to set the requirement priorities. Based on the estimated effort and priorities, the Headquarters team updated the Product Backlog and chose a list of features to be developed in the software prototype.

4.2.2. Tasks performed by the IG

The IG used online serious games to identify, prioritize, and negotiate the requirements. In the first iteration, 15 participants from the Customer team, both User teams, and the Development team played online *Prune the Product Tree* games to identify requirements. Since an online game typically takes between 45 and 90 minutes to play [36], we chose a 1-hour time frame for each game. This is especially important due to the time-zone difference between the teams. As it is discussed in Section 4.1.1 and is shown in Figure 1, the Headquarter and User2 teams had only one hour of overlapping work time during

---

[4] Participants were asked to deliver a set of user stories, a prioritized product backlog with effort estimation, a high-level software prototype, an audit trail that fully indicated the team's internal and external communication, and an individual reflection report to provide feedback about the project.

each working day. For that reason, in real life it would not be convenient or even possible for them to attend the game sessions longer than one hour. Indeed, this corresponds to real-life situations where time difference plays a key role in communication issues in GSD.

During each game session, customers and users placed their requirements on different areas of the tree according to their priorities (i.e., low, medium, and high). At the same time, developers had the opportunity to see the users' actions and ask questions by using the integrated chat facility. Each requirement was automatically attributed to its owner by the web-based tool used for playing *Prune the Product Tree* online. After these games, the Headquarters team used the list of features added by the users to prepare the initial product backlog.

In the second iteration, participants from the Customer and User teams played *Buy a Feature* games online to negotiate and prioritize the requirements. In this game, all the product backlog items were assigned a price based on the assessment performed by the Development team. To do so, developers estimated the amount of time and effort needed to develop each feature, and after negotiation with the Headquarters team they tried to choose a price for each feature. The players used the integrated chat tool to discuss, negotiate, and choose a set of features with the highest value. The results of these games were used by the Headquarters team to choose a list of features to be developed in the software prototype. In all games, one of the authors in the role of course assistant participated as an observer without influencing the process.

Both groups used synchronous communication methods such as teleconferencing, videoconferencing, and instant messaging, as well as asynchronous communication methods such as email. All communication among team members and between teams was archived as an audit trail and submitted to the course assistants at the end of the project.

5. Results

Following [32, 33], we analyzed the effectiveness of the suggested Innovation requirements elicitation method both quantitatively and qualitatively. The number of requirements identified by a subject was used as a measure for the quantitative effect and the number of new requirements and the number of feasible requirements for measuring the qualitative effect. As suggested by [33], a creative requirement not only must be new but also must be useful, and for that reason the quality of a requirement can be evaluated by measuring both its novelty and feasibility [32].

To define the novelty of the requirements, we used existing knowledge of the *UniGuide* case application, which was available before the experiment. It is worth noting that initially a set of high-level features have been identified and implemented in the prototype of this application. Since the experiment subjects were not aware of the case application, these high-level features could be used as the baseline for evaluating the novelty of the requirements identified in our experiment [32, 33]. Therefore, a requirement identified by an individual subject was considered to be new if it had not been included in the baseline features identified prior to this experiment. However, it must be mentioned that to make this evaluation more precise, we decided to break down the high-level features of the baseline to 33 atomic requirements in the similar granularity level as the requirements identified by the subjects in our experiment.

On the other hand, to define the feasibility of the requirements all the identified requirements were evaluated on their realism, whether they were out of scope, and whether they had some technical obstacles. This evaluation was done by the authors of this paper. One of the researchers, who had participated in the previous design and development of the *UniGuide*, did the initial evaluation, which was then assessed and verified by a second researcher. An idea was assigned to the infeasible category if it was unrealistic, was considered out of scope, or presented technical obstacles. For example, a member of the IG requested "An application that can be used to regulate temperatures, lighting, etc., remotely." This requirement was considered to be unrealistic due to the fact that usually in public buildings and environments, such as a university campus, the air conditioning systems are managed by facility services and temperature is controlled centrally; obviously, unauthorized individuals such as students do not have access to such systems. An example of a technically infeasible feature is one of the subjects requesting to "Make the system real time by using the CCTV." This feature was considered infeasible because of the fact that providing real-time visual data in a big environment like a university not only requires extensive technical infrastructure but also might not be in line with the security and privacy policies of the organization.

In the experiment, each requirement was attributed to the individual subject who identified it in the requirements elicitation process. The mean numbers of identified requirements were used for comparative analysis of the effectiveness of the two methods used in the experiment. Tables 3 and 4 summarize the results of the experiment.

Table 3. Number of requirements identified by subject.

Group	Subjects	Number of Requirements			Number of New Requirements			Number of Feasible Requirements		
	n	Total	M	SD	Total	M	SD	Total	M	SD
AG	19	70	3.7	1.86	30	1.6	0.96	59	3.1	1.73
IG	15	72	4.8	1.82	36	2.4	0.91	63	4.2	2.34

As can be seen in Table 3, the total number of identified requirements appears to be nearly the same in both groups, but since the IG was smaller in terms of the number of participants, the mean number of requirements was higher in this group. As it can be seen from Table 3 above, the AG identified 70 requirements while IG identified 72 requirements. The IG subjects identified more new requirements (M = 2.4) and feasible requirements (M = 4.2) than the AG subjects (M = 1.6 and M = 3.1). The standard deviation of the number of requirements and number of new requirements were quite similar in both groups. However, the variability between the subjects in identifying feasible requirements appears to be slightly higher in the IG (SD = 2.34) than in the AG (SD = 1.73). Of the 70 requirements identified by AG, 61 were functional, and the rest were non-functional. In contrast, of the IG's 72 requirements, 51 were functional and 21 non-functional (see Table 4).

Table 4. Number of functional and non-functional requirements identified by subject.

Group	Subjects	Number of Functional Requirements			Number of Non-functional Requirements		
	n	Total	M	SD	Total	M	SD
AG	19	61	3.21	1.65	9	0.47	0.697
IG	15	51	3.40	1.76	21	1.40	1.18

As it is indicated in Table 4, the mean number of functional requirements identified by IG (M = 3.4) is slightly higher than AG's (M = 3.21). However the mean number of non-functional requirements identified by IG (M = 1.4) is almost three times bigger than the average number of non-functional requirements identified by AG (M = 0.47).

5.1. Quantity of requirements

The effectiveness of the requirements elicitation methods regarding the quantity of the requirements was measured based on the number of requirements identified by a subject. The results are presented in Table 5.

Table 5. Mean number of requirements identified by subjects.

	Group	N	Mean	Std. Deviation
Number of Requirements	AG	19	3.7	1.857
	IG	15	4.8	1.821

As it is shown in Table 5, the IG subjects identified approximately one more requirement on average (M = 4.8) than the AG subjects (M = 3.7). This indicates that the Innovation requirements elicitation method can have a positive effect on requirements elicitation by increasing the number of identified requirements.

In order to statistically analyze the effect of Innovation requirements elicitation on the number of identified requirements, an independent sample t-test was used. Since we cannot credibly assume theoretically or empirically based on earlier studies that the studied Innovation requirements elicitation method can produce only better results than the compared Agile requirements elicitation method, we used a two-tailed test [50]. The statistical hypotheses for the t-test we used are of the form:

$$H_0: \mu_{IG} = \mu_{AG}$$
$$H_1: \mu_{IG} \neq \mu_{AG.}$$

The results of the t-test for the effect of the Innovation requirements elicitation method on the quantity of the requirements in comparison to the Agile requirements elicitation method are presented in Table 6 below.

Table 6. t-test for the mean number of requirements identified by subjects.

		Levene's Test for Equality of Variances		t-test for Equality of Means			
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference
Number of Requirements	Equal Variances Assumed	0.185	0.670	1.754	32	0.089	1.116

According to the results of the Levene's test, the t-test assumption of the homogeneity of the group variances can be expected to hold. The t-test results reveal that the statistical significance of the difference of the means between the groups ($p = 0.089$) is not significant at the level of $\alpha = 0.05$, but only at level $\alpha = 0.10$. Even if the H_0 hypothesis of equality of mean number of identified requirements between the IG and AG groups cannot be rejected at the significance level $\alpha = 0.05$, it still leaves open the possibility that the Innovation requirements elicitation method could result in identifying a higher number of

20

requirements than using the common Agile requirements elicitation method. Even if a difference in the sample means of the requirements were observable in our experiment and the H_0 hypothesis of equality of the means can be rejected at the level α = 0.10, the sample size of the experiment did not offer sufficient statistical power to reject the H_0 hypothesis with the chosen significance level of α = 0.05. However, the results imply a need for further analysis.

To further explore the effect of the Innovation requirements elicitation method on the number of identified requirements, we decided to examine the potential relation between the effectiveness of the requirements elicitation method and the level of the individual's technical skills. As previously discussed, in software development projects functional requirements are often provided by customers while non-functional requirements are mainly identified by developers [3, 12]. Following this, we decided to further our analysis by investigating the relation between individuals' experience and the numbers of functional and non-functional requirements identified by them. The results from these two sets of analyses are discussed in the following sections.

5.1.1. Quantity of requirements identified by less-experienced subjects

In the first set of analyses, we evaluated the effects of the method under study on the number of requirements identified by less-experienced participants, i.e., the members of the Customer and User teams. As is discussed earlier, these teams have a central role in the requirements elicitation phase since they are the ones who are supposed to provide their desired requirements for the system to be developed. Tables 7 and 8 provide a summary of the number of requirements identified by only the less-experienced subjects.

Table 7. Mean number of requirements identified by the less-experienced subjects.

	Group	N	Mean	Std. Deviation
Number of Requirements	IG	11	5.5	1.635
	AG	15	3.2	1.521

As it can be seen from Table 7, the mean number of requirements (M = 5.5) identified by less-experienced subjects from the IG is considerably higher than that of the AG subjects (M = 3.2). Additionally as it can be seen in Table 8 below, the mean numbers of both functional and non-functional requirements identified by less-experienced subjects from IG (M = 4.09, M = 1.36) are higher than those of AG (M = 2.93, M = 0.27).

Table 8. Mean number of functional and non-functional requirements identified by the less-experienced subjects.

Group	Less-experienced Subjects	Number of Functional Requirements			Number of Non-functional Requirements		
	N	Total	M	SD	Total	M	SD
AG	15	44	2.93	1.53	4	0.27	0.46
IG	11	45	4.09	1.51	15	1.36	1.20

We have evaluated the statistical significance of these results as it is presented in Tables 9 and 10.

21

Table 9. t-test for the mean number of requirements identified by the less-experienced subjects.

		Levene's Test for Equality of Variances		t-test for Equality of Means			
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference
Number of Requirements	Equal Variances Assumed	0.062	0.805	3.62	24	0.001	2.255

According to the results of the first t-test shown in Table 9, the assumption of the homogeneity of the group variances can be expected to hold. The results of the t-test show a statistically highly significant (p = 0.001) difference between the mean number of requirements identified by less-experienced subjects of IG than of AG.

Table 10. t-test for the mean number of functional and non-functional requirements identified by the less-experienced subjects.

		Levene's Test for Equality of Variances		t-test for Equality of Means			
		F	Sig.	t	df	Sig. (2-ailed)	Mean Difference
Number of Functional Requirements	Equal Variances Assumed	0.449	0.509	1.91	24	0.068	1.16
Number of Non-functional Requirements	Equal Variances Assumed	14.479	0.001	3.23	24	0.004	1.10
	Equal Variances Not Assumed			2.87	12.12	0.014	1.10

According to the results of the second t-test shown in Table 10, the difference between the mean numbers of functional requirements identified by the less-experienced subjects from both groups (p = 0.068) is not statistically significant at the level of α = 0.05. In the case of non-functional requirements, the assumption of homogeneity of the group variances cannot be expected to hold according the Levene's test, and the t-value based on pooled variances is used. The results reveal a statistically significant difference between the mean numbers of non-functional requirements identified by the less-experienced subjects (p = 0.014) from IG and AG. Therefore, regarding the subjects with less professional experience, the Innovation requirements elicitation method appears to improve requirements elicitation by increasing the quantity of the identified requirements, especially the number of non-functional requirements.

5.1.2. Quantity of requirements identified by more-experienced subjects

On the other hand, in the second sets of analyses only the numbers of requirements identified by the more-experienced subjects—i.e., the individuals from the Development teams—were taken into consideration. As noted earlier, the individuals from the Development teams are the ones who must further analyze the user requirements and specify the necessary software requirements. Following this, it could be expected that subjects with a higher level of technical skills provide a higher number of non-technical requirements. The summary of the number of requirements identified by the more-experienced subjects are shown in Tables 11 and 12.

Table 11. Mean number of requirements identified by the more-experienced subjects.

	Group	N	Mean	Std. Deviation
Number of Requirements	AG	4	5.5	0.816
	IG	4	3.0	2.082

As is shown in Table 11, the mean number of requirements (M = 3.0) identified by the more-experienced subjects from the IG is considerably lower than that of the AG subjects (M = 5.5). Table 12 below presents the number of functional and non-functional requirements identified by the more-experienced individuals from each group.

Table 12. Mean number of functional and non-functional requirements identified by the more-experienced subjects.

Group	More-experienced Subjects	Number of Functional Requirements			Number of Non-functional Requirements		
	N	Total	M	SD	Total	M	SD
AG	4	17	4.25	1.89	5	1.25	0.96
IG	4	6	1.5	0.58	6	1.5	1.29

As it is indicated in Table 12, the mean number of functional requirements identified by the more-experienced subjects from AG (M=4.25) is higher than those of IG (M=1.5). However, the mean number of non-functional requirements identified by the more-experienced subjects from IG (1.5) is higher than those of AG (M=1.25). In addition to this, it can also be seen from Table 12 that the mean numbers of functional and non-functional requirements identified by more-experienced individuals from IG are the same. However, the mean number of functional requirements identified by more-experienced individuals from AG is substantially higher than the mean number of non-functional requirements identified by them. This observation suggests that more-experienced subjects from AG have provided a substantially higher number of functional requirements (i.e. user requirements) rather than non-functional requirements. The statistical significance of the differences observed in Tables 11 and 12 are then evaluated through t-tests. The results of the t-test for more-experienced subjects are presented in Table 13 below.

Table 13. t-test for the mean number of requirements identified by the more-experienced subjects.

		Levene's Test for Equality of Variances		t-test for Equality of Means			
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference
Number of Requirements	Equal Variances Assumed	2.4	0.172	2.24	6	0.067	2.5

According to these results, the t-test assumption of the homogeneity of the group variances can be expected to hold. The t-test results reveal that the statistical significance of the difference of the means between the mean number of requirements identified by the more-experienced subjects of IG than AG (p = 0.067) is not significant at the level of $\alpha = 0.05$, but only at level $\alpha = 0.10$. Despite the fact that the t-test result is not statistically significant at the chosen level of $\alpha = 0.05$, the difference in sample means suggests that there could be a possibility that the Innovation requirements elicitation method would result in a lower number of requirements proposed by experienced members of Development teams compared to the Agile method.

23

Table 14 below presents the results from a t-test, which is conducted to evaluate the statistical significance of the differences between the mean numbers of functional and non-functional requirements identified by the more-experienced subjects.

Table 14. t-test for the mean number of functional and non-functional requirements identified by the more-experienced subjects

		Levene's Test for Equality of Variances		t-test for Equality of Means			
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference
Number of Functional Requirements	Equal Variances Assumed	2.882	0.140	2.78	6	0.032	2.75
Number of Non-functional Requirements	Equal Variances Assumed	0.5	0.5	0.31	6	0.76	0.25

As it is indicated in Table 14, the difference between the mean numbers of functional requirements identified by the more-experienced subjects of AG and IG is statistically significant ($p = 0.032$). On the other hand, the difference between the mean numbers of non-functional requirements identified by the more-experienced subjects of IG and AG is not statistically significant ($p = 0.76$). These results indicate that in AG the more-experienced individuals have identified a significantly higher number of user requirements than system requirements. Furthermore, according to Tables 8 and 12, it becomes apparent that while in IG the mean number of functional requirements identified by the less-experienced subjects (M=4.09) is significantly higher than the mean number of functional requirements identified by the more-experienced subjects (M=1.5) the less-experienced subjects from IG identified a slightly higher number of non-functional requirements than the more-experienced subjects. In contrast, in AG the mean numbers of both functional and non-functional requirements identified by the more-experienced subjects (M=4.25, M=1.25) is higher than the mean numbers of functional and non-functional requirements identified by less-experienced subjects (M=2.93, M=0.27).

These observations that seem to contradict the previous observations (see Section 5.1.1) can be explained by the characteristics of the Innovation method. During the game sessions held in IG, while the individuals from Customer and Users teams were providing their requirements, the individuals from the Development team were mainly observing users' actions and trying to clarify their suggested requirements. In other words, while the members of Development team had the ability to suggest potential new features to the end-users and assist them to identify and clarify their requirements, it seems that using the Innovation method hindered the individuals in the Development team from imposing their desired features to the less-experienced individuals. Thus, it allows the end-users to provide those requirements, which were more valuable to them. Therefore, it becomes apparent that the less-experienced subjects from IG had a better contribution in terms of the quantity of functional and non-functional requirements identified during the requirements elicitation process. Overall, it seems that using the Innovation method ultimately increased the relative number of requirements identified by the actual system users from Customer and Users teams in comparison to developers from Development teams.

Considering the quantitative effectiveness of the innovation requirements elicitation method, the results from the analyses presented in this section show that the Innovation method can improve distributed requirements elicitation in comparison to the Agile requirements elicitation in terms of number of user requirements identified by actual system users compared to system developers. Based on these

24

results, we can conclude that utilizing online serious games is an effective method for facilitating the distributed requirements elicitation process with respect to the number of requirements identified by customers and actual system users. Our results reveal that the Innovation method enables less-experienced individuals to identify and provide a higher number of requirements. In addition, while this method enables development teams to evoke more user requirements by assisting end-users in identifying and improving their needs, it seems that using Innovation method also hinders developers from imposing their preferred features to customers. This is in concordance with the principles of agile software development in emphasizing the significance of producing only customers' desired requirements and avoiding unnecessary features, which do not bring any business values to customers [6, 51, 52].

5.2. Quality of requirements

The effectiveness of the methods regarding the quality of requirements was measured with two different variables, number of new requirements and number of feasible requirements. The results of the experiment concerning the identified requirements' novelty are presented in Figure 3, and feasibility is presented in Figure 4.
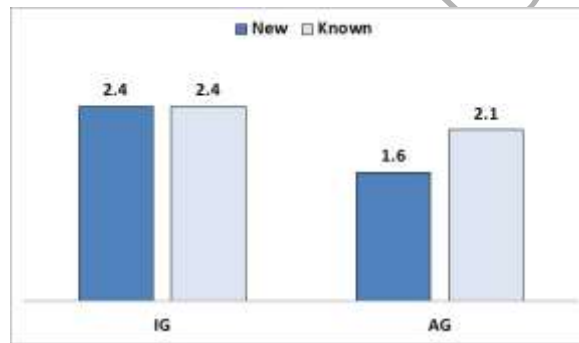
Figure 3. Mean number of new and known requirements by groups.

As is shown in Figure 3 above, the IG subjects identified on average more new requirements (M = 2.4) than the AG subjects (M = 1.6). In addition, the relative number of new requirements compared to known requirements is higher. This indicates that the new Innovation requirements elicitation method would result in more new requirements than the common Agile requirements elicitation method.
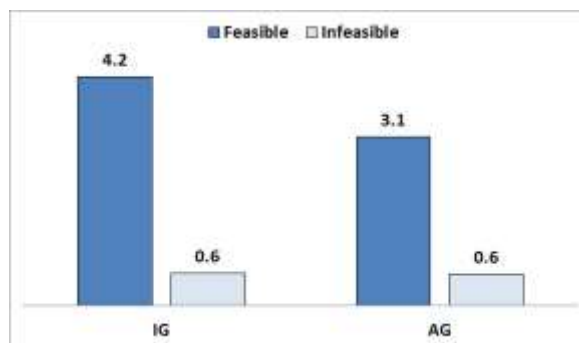
Figure 4. Mean number of feasible and infeasible requirements by groups.

Figure 4 reveals a rather similar effect of the Innovation requirements elicitation method on the number of feasible requirements as on the novelty of the requirements. On average, IG subjects identified approximately one more feasible requirement (M = 4.2) than AG subjects (M = 3.1), even though the mean number of infeasible requirements remained the same for both groups.

25

It must be noted that, despite some similarities between the requirements identified by different subjects in our study, we had to include all those similar requirements in the analyses relevant to the requirements elicitation phase. This is due to the fact that in this study each requirement was attributed to the individual subject who identified that requirement. However, as it is discussed in Section 5.3, those similar requirements were later merged or removed during the requirements negotiation phase.

To analyze the effect of the Innovation requirements elicitation method on the quality of the identified requirements in comparison to that of the Agile requirements elicitation method, we used a similar independent samples t-test as in the previous analysis of quantitative effectiveness. The t-test was done for both requirements' quality aspects separately. Table 15 indicates the results of the t-test.

Table 15. t-test for the quality of the requirements identified by subject.

		Levene's Test for Equality of Variances		t-test for Equality of Means			
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference
Number of New Requirements	Equal Variances Assumed	0.005	0.946	2.531	32	0.016	0.821
Number of Feasible Requirements	Equal Variances Assumed	1.442	0.239	1.571	32	0.126	1.095

According to the Levene's test, the variance homogeneity assumption of the t-test can be expected to hold. The results of the t-test reveal that the difference in the mean number of new requirements is statistically significant ($p = 0.016$) at a significance level of $\alpha = 0.05$. However, the difference in the mean number of feasible requirements is not statistically significant ($p = 0.126$). Based on the results, we can conclude that the Innovation requirements elicitation method can be considered to improve the quality of the requirements with respect to the novelty of the requirements when compared to the Agile method. However, the data do not provide enough evidence to support the hypothesis that the suggested new method would improve the quality aspect of feasibility of the identified requirements.

5.3. Using online serious games in the requirements negotiation process

In addition to the main objective of this study, we evaluated the effectiveness of our suggested approach in requirements negotiation. The main goal of requirements negotiation is to resolve conflicts and to come to an agreement between stakeholders about the most important and valuable requirements. Onsite customer participation enables the development team to negotiate requirements through face-to-face interaction and communication. However, in distributed projects this approach is almost impossible, and teams have to communicate over ICT-mediated mediums.

In our experiment, individuals from the AG used videoconferencing tools for requirements negotiation. During this phase, the Headquarters and Development teams tried to indicate the available inconsistencies and overlaps between requirements suggested by individuals. In addition, teams discussed the priorities of identified requirements based on estimated development efforts and necessity of requirements. Finally, the teams updated the product backlog and selected a set of high-priority requirements to be developed in the software prototype.

In contrast, in the IG individuals negotiated the requirements by applying the *Buy a Feature* game. As mentioned earlier, in this game every software feature in the product backlog has a price, and each individual from the Customer and User teams has a limited amount of play money that can be spent to buy the desired feature. Since the total amount of money that all these individuals have together is only enough to buy one-third of all software features in the product backlog, the individuals have to discuss and choose the most important features to be bought and consequently developed.

Following this phase, in both groups, the Development team delivered an initial version of the software prototype to the Customer and User teams. Later, they collected feedback regarding the usability and appearance of these prototypes, and based on this feedback, the software prototype and the product backlog were improved and updated. To evaluate the effectiveness of each technique, we compared the initial set of requirements provided by each group and the final requirements delivered by the groups at the end of the project.

The final product backlogs delivered by the groups indicate that both groups, to some extent, were able to merge similar ideas and to improve the vague and unclear requirements. However, after evaluating the requirements delivered by each group we realized that there are still some similarities and overlaps among these requirements. Therefore, in order to identify and merge similar requirements and to align the granularity of the requirements with the requirements baseline, all the delivered requirements were analyzed by two of the authors individually. The results from these individual requirements analyses were then discussed with the third author, and the final sets of unique requirements delivered by each group were identified. Based on these results, AG has delivered 34 unique requirements from which 6 were infeasible. In contrast, IG provided a set of 35 unique requirements from which 3 were infeasible (see Table 16).

Table 16. Number of unique feasible requirements after the requirements negotiation phase.

	Group	N	Total	Mean
Number of unique feasible requirements	AG	19	28	1.47
	IG	15	32	2.1

As it can be seen in Table 16 after combining the similar requirements and removing infeasible requirements, the average number of final requirements identified by IG (M = 2.1) is higher than the average number of requirements identified by AG (M = 1.47). Although these results show that IG members were able to deliver a relatively higher number of feasible requirements, we decided to further evaluate the effectiveness of the methods under study based on the relevance of the requirements identified by each of the groups. To do so, we decided to perform a precision and recall analysis that is widely used in the information retrieval context [53]. In the context of information retrieval, precision is measured by calculating the fraction of retrieved items that are relevant while recall is measured as the fraction of all relevant items that are retrieved [53-55]. Higher precision means that from all the retrieved items a relatively greater fraction are relevant, while higher recall means that a relatively greater fraction of all relevant items is retrieved. According to these definitions, for calculating precision and recall measures in the context of requirements engineering, feasible requirements can be considered as relevant items. Therefore, following [53-55], we calculated each of precision and recall measures as follows:

$$Precision = \frac{Total\ number\ of\ feasible\ requirements\ in\ the\ final\ backlog}{Total\ number\ of\ requirements\ in\ the\ final\ backlog}$$

$$Recall = \frac{Total\ number\ of\ feasible\ requirements\ in\ the\ final\ backlog}{Total\ number\ of\ all\ feasible\ requirements\ in\ the\ collection}$$

It must be noted that since it is almost impossible to identify a closed set of "all feasible requirements," we decided to take into account all the feasible requirements that were known at the end of our experiment. Therefore, we formed a collection that consists of all the distinct items from the requirements baseline and all the new and feasible requirements identified by at least one of AG or IG groups during this project as follows:

$$All\ feasible\ requirements\ = \{(\ Requirements\ baseline)\ \cup\ (New\ feasible\ requirements\ from\ AG)\ \cup\ (New\ feasible\ requirements\ from\ IG)\}$$

As previously mentioned, the requirements baselines known prior to this study consisted of 33 unique requirements. In addition to this, after evaluation of the final product backlogs delivered by the groups we identified 10 new feasible requirements delivered only by AG, 14 new feasible requirements delivered only by IG, and 6 new feasible requirements delivered by both groups. Therefore, our collection of "all feasible requirements" after negotiation phase consisted of 63 distinct requirements. Using this collection we calculated the precision and recall measures for both groups. The results from these calculations are presented in Table 17 below.

Table 17. Precision and recall analysis after the requirements negotiation phase.

Group	Precision	Recall
AG	$\frac{28}{34} = 0.823$	$\frac{28}{63} = 0.444$
IG	$\frac{32}{35} = 0.914$	$\frac{32}{63} = 0.507$

As it is indicated in Table 17, both the precision and recall measures from IG are greater than AG's measures. Based on these results, more than 91% of the requirements identified by IG are relevant, while from the requirements identified by AG around 82 % are relevant. IG's precision being greater than AG's means that the individuals from IG were able to identify a relatively greater percentage of relevant requirements. On the other hand, as it can be seen from Table 17, individuals from IG were able to identify more than 50% of all the feasible requirements while individuals from AG were able to identify around 44% of the all feasible requirements. These results show that individuals from IG were able to identify a greater fraction of all feasible requirements known for the *UniGuide* application. These results demonstrate that overall, after the requirements negotiation phase, individuals from IG have a better performance in terms of identifying a relatively higher number of feasible requirements.

When considering the results, it is also essential to assess the amount of time used by each group while identifying the requirements. Although both groups had 1 week to identify and deliver their requirements, it is notable that subjects from the IG identified requirements during 1-hour game sessions. On the other hand, project communication logs indicate that in addition to almost three days of constant

communication via email for identifying and gathering their requirements, AG members spent around 3.5 hours in discussion via chat and online meetings. Even though the exact amount of time spent by AG was not calculated, it seems the IG had a substantial advantage in the amount of time consumed per requirement, which is beneficial, especially in today's fast-moving and competitive market.

6. Discussion

The main objective of this research was to investigate the impacts of online serious games on the quality of requirements elicitation process in distributed software projects. Based on prior research, we suggest that applying online serious games not only is an effective method for encouraging individuals to engage in distributed requirements elicitation but it also enables development teams to facilitate innovation and creativity among software stakeholders. We assessed the effectiveness of our suggested approach with the experimental study. Our experiment demonstrated the effectiveness of the Innovation requirements elicitation method both quantitatively and qualitatively.

The observations indicate that subjects who used online serious games appeared to identify more requirements in general, as well as more new requirements, than those subjects who used common agile practices. In order to test the significance of the difference between the numbers of requirements identified by utilizing each method, we conducted a series of hypothesis tests.

We started our analysis by comparing the difference between the mean numbers of requirements identified by all the subjects participating in the requirements elicitation process. The comparison between the mean numbers of the identified requirements indicates that the suggested Innovation requirements elicitation method results in a higher number of requirements. However, to evaluate the significance of this effectiveness, we conducted a t-test. The results from the t-test showed that the difference between the number of requirements identified by using either of the methods is not statistically significant. For continuing the analysis, we focused on the effect of individuals' experience on the quantity of identified requirements. To do so, we analyzed the number of requirements identified by less-experienced and more-experienced subjects groups separately.

First, we compared the effects of the methods on the total number of requirements and number of functional and non-functional requirements identified by less-experienced individuals. This was done because, based on the literature, it is known that requirements elicitation is especially challenging for customers and end users with low levels of technical and domain skills [3, 5, 10, 11]. In addition, it has been observed in previous studies that using well-structured requirements elicitation methods is more efficient for gathering requirements from less-experienced individuals [56]. The results of two t-tests conducted in this round of analysis reveal that both the mean number of all requirements and the mean number of non-functional requirements identified by the less-experienced individuals using online serious games are significantly higher than the mean numbers of all requirements and non-functional requirements identified by utilizing common agile practices. This is an important observation keeping in mind that the less-experienced subjects have the role of the customers and end users and they are the ones who must provide their desired requirements. Therefore, according to these results, it can be said that using online serious games enabled individuals with less professional experience to better participate in the requirements elicitation process and to identify a higher number of requirements.

We continued our analysis by evaluating the effects of the utilized methods on the total number of requirements and number of functional and non-functional requirements identified by more-experienced individuals from Development teams. This comparison has been done due to the fact that implementing unnecessary features while ignoring important user requirements are often associated with inadequate requirements elicitation [1]. During this round of analysis, we conducted two t-tests in order to evaluate the significance of our findings. Although the mean number of requirements identified by more-experienced individuals from AG is higher than from IG, the results of the first t-test did not confirm this difference to be statistically significant. The results of the second t-test, however, indicate that the difference between the mean numbers of functional requirements identified by the more-experienced subjects of AG than of IG is significantly high. These results in addition to the highly significant results from the t-tests conducted for less-experienced subjects suggest that using online serious games could reduce the number of user requirements proposed by developers and in contrast increase the number of requirements proposed by less-experienced individuals who are the actual system users. This can be considered as a benefit especially from the perspective of agile software development where its highest priority is to deliver customers' most valuable requirements [21] and to avoid consuming project resources on producing unnecessary features, which do not bring any business value to customers [6, 51, 52].

For evaluating the effectiveness of the methods under study regarding the quality of requirements, we conducted corresponding t-tests for number of new requirements and number of feasible requirements provided by the groups. The results of these tests supported the expectation that the requirements identified by subjects within the IG also have clearly more novelty compared to the requirements identified by subjects from the AG. The method also might have some effect on the feasibility of the requirements, but the results from the conducted t-tests did not support this effect clearly. Therefore, we decided to further investigate this effect by conducting a precision and recall analysis [54, 55] to evaluate the overall relevance of the requirements identified by the groups. The results from our precision and recall analysis reveal that, using Innovation method, IG has provided substantially more relevant (i.e. feasible) requirements than irrelevant (i.e. infeasible) requirements compared to AG. Additionally, IG has also returned a slightly greater fraction of all the relevant requirements compared to AG. Overall, our experiment conducted in the context of an academic project gave convincing results for supporting the effectiveness of our proposed Innovation requirements elicitation method.

According to these results, we believe that using online serious games is an effective method for solving some of the main requirements elicitation challenges mentioned at the beginning of this paper. First, like other collaborative games, this technique was interesting to the majority of users. Thus, in alignment with prior studies (e.g., [32-34, 37]), we argue that our proposed method, Innovation requirements elicitation, is an effective method for encouraging system stakeholders to participate in requirements elicitation and to improve their engagement in the process by actively collaborating with other participants. According to our observations during the project and the participants' reflection reports, not only were most of the participants satisfied with using online serious games in requirements elicitations and negotiation sessions, but they also enjoyed participating in these online sessions.

The online serious game was considered especially helpful by some participants since they were able to ask questions and discuss suggested features and ideas with other players via the chat facility. One subject mentioned:

> *"I was really nervous about the games, and I was sure that I wouldn't come up with any*
> *ideas. But instead it went totally different. I used the chat box at the play, and I really was*
> *involved in the game."*

Second, we noticed that during the game sessions most of the participants were confident enough to interact with other players and participate in discussions, and as a result they were highly engaged in the requirements elicitations and negotiation sessions. Some of the subjects mentioned that it was easier for them to interact and communicate with other players and to describe and discuss their requirements. In addition, the majority of participants said that playing interactive games allowed them to easily see and follow other participants' actions and create new ideas accordingly. For instance, one of the participants from IG said:

> *"It was interesting to see what kind of requirements others added to the tree, and it gave*
> *me the opportunity to invent new requirements based on others' requirements."*

This might be due to the fact that interactive serious games provide a rich visual image of the software under development to the participants, which makes it easier for them to understand the relations between different user requirements and how these requirements must be developed over time. In addition to this, due to the interactive nature of these games, collecting user requirements from several stakeholders and providing feedback to them happens quickly and simultaneously. In comparison, when using common requirements elicitation techniques, the requirements are mainly represented in the written format, and therefore it might be more difficult and time-consuming to follow and compare all the user requirements and understand the necessity and priority of those requirements. Therefore, it might be challenging for individuals to identify innovative ideas based on requirements suggested by others. It is worth noting that one of the participants from the AG brought up this issue:

> *"It was even hard for me to figure out one new [idea] after I saw everybody's ideas."*

Furthermore, many participants considered the *Buy a Feature* game a helpful technique for negotiating and prioritizing requirements. The diversity between users' expectations and requirements is a main factor that leads to conflict among software stakeholders and makes requirements negotiation challenging [9, 18]. The winning factor in this game was that participants did not have enough money to buy all their desired features. Therefore, the teammates had to discuss and reach an agreement about the most important and necessary feature to buy. Our observations during and after the game sessions indicate that this technique is effective for prioritizing customers' requirements based on business value and the cost and effort needed to develop each feature. A participant observed:

"Everyone has a different priority. At some point, I wanted to buy a feature that seemed really important to me, but I didn't have enough money. So I suggested to others to buy that. Unfortunately, it was not a high priority for everyone."

Another factor that might have improved the quality of the process of requirements elicitation in the IG is that the development team constantly followed the users' actions and discussions. Obtaining a common understanding among different stakeholders is difficult [4], and extra effort is needed by software developers and system analysts to gain a shared understanding of system features [9]. In this study, when there was a misunderstanding about an idea or requirements, the team could immediately ask for more clarification and description. This technique enables the development team to gather more detailed information about users' requirements. The available chat history from the game sessions indicates that in several cases the developers discussed issues with users and immediately changed, merged, or improved requirements.

In addition, in our case, using a web-based tool was a good solution for the problem of documentation, one of the challenges of requirements engineering in software projects [9, 18]. The web-based tool we used for organizing online requirements elicitation sessions automatically produces a complete list of players, their requirements, and their actions during the game. This information can be used by the development team for further analysis in later stages of the projects.

Finally, in all game sessions participants used only the chat tool provided by the game environment as the communication medium. This method enables players to be more comfortable participating in discussions and to collaborate with others as they can overcome their low self-conception of their foreign language skills. Since English is usually used as the working language in GSD projects, using instant messaging enables cross-cultural teams to mitigate issues related to language barriers such as the diversity of dialects and the variants of English [57]. In addition, [58] found that the efficiency of discussion between software stakeholders via instant messaging is higher than that of face-to-face discussion, especially in cross-cultural teams. Furthermore, the available chat history enables the development team to document discussions between stakeholders without spending extra time and money to transcribe face-to-face conversations between software stakeholders.

6.1. Academic and managerial implications

When summarized, the contributions of this study are both academic and practical. From an academic perspective, our results reveal that serious games can be used not only for pedagogical purposes but also as a driver for enhancing creativity and facilitating innovation among individuals. In this study we focused on using serious games only in distributed requirements elicitation and negotiation processes. However, further research may enhance our understanding about the impacts of serious games on different stages of software development projects. In addition, here our results only indicate that utilizing online serious games is significantly effective for improving the performance of less-experienced stakeholders such as end users in the requirements elicitation process. However, it may be valuable to further investigate how this approach can be beneficial to improve the performance of more-experienced stakeholders like development teams.

32

Our results are also providing valuable implications for practice. The results of our data analysis revealed that playing games encourages individuals to collaborate in team projects and enables them to be more confident in communicating with others. In particular, using interactive online serious games provides a clear and understandable image of the project to the customers while it enables participants to observe and follow others' actions easily. In addition, integrating text-based communication alongside online games is an effective method for improving the quality of communication in multicultural teams and mitigating language barriers. Finally, our suggested approach enables development teams to mainly concentrate on understanding user expectations and needs during the requirements elicitation process without spending extra time and effort on the documentation process. Although in this study our main goal was to study the impacts of iterative serious games on distributed requirements elicitation, it is worth noting that utilizing such games might be beneficial to facilitate communication and collaboration between software stakeholders in different types of collocated projects.

6.2. Threats to validity

It must be mentioned that there are some limitations of this study that may present threats to validity of the results [48]. First, the subjects of our experiment were chosen among students (i.e. threat to external validity). Although using students is common in conducting empirical research in Software Engineering discipline, it is suggested by previous studies that using students as subjects must be under certain circumstances [59]. Even though following [60] we compensated this threat to external validity by choosing an application for which students are the end users, the question remains whether it would be fun and pleasant for other user types with different age ranges to play online games to provide their requirements. Another threat to external validity [48] is that we conducted our empirical research in an environment that was a simulation of distributed software development. Although the activities performed by teams were similar to those of real GSD projects, we had to assign teams to different time zones to simulate temporal distance. In addition to simulating the geographical distance, we also forced teams to use ICT-mediated tools as their main medium of communication. Thus, the geographical and temporal distance might not have been completely sensible for participants. However, most of the subjects mentioned the difficulty of communication and collaboration in cross-cultural teams and over ICT-mediated tools in their personal diaries delivered at the end of the project. The last identified threat to external validity [48] is that in this research we used *UniGuide* as the software under development, which is considered to be a small and simple application with a limited number of features to be developed. The question here is whether it would be beneficial to use this method in order to identify and prioritize a large number of requirements in more complex and bigger industrial projects.

One major threat to conclusion validity [48] is the low number of subjects in our experiment. Since our evaluation is based on data collected from 2 experimental groups this may reduce the ability to reveal patterns in the collected data. Another threat to conclusion validity is the quality of the data collected from the subjects participated in our study. Since in this study the subjects were asked to prepare different types of deliverables, including requirements, there is a risk that they have not paid enough attention to identifying and providing their requirements. However we have tried to eliminate this threat by balancing the workload of the subjects in different teams based on their assignments and activities.

A major threat to the construct validity [48] is that we used the number of identified requirements for measuring the effectiveness of requirements elicitation methods under study. Since the delivery of requirements was a part of the course assignments there is a risk that students might have thought that the higher number of requirements might lead to higher grades and for that reason they just paid attention to the quantity of requirements. This threat however has been mitigated in several ways. First of all at the beginning of the project we have emphasized that the main objective of the course is to understand the issues associated with communication and collaboration in distributed projects and not to identify a higher number of requirements. Secondly we have tried to measure the effectiveness of the treatments in terms of the quality of the requirements (i.e. novelty and feasibility) provided by subjects. Thirdly we have conducted a precision and recall analysis at the end of the project to evaluate the relevance of the requirements identified by each group.

Finally our experiment also presents a threat to internal validity [48] because the numbers of subjects in two experimental groups were not the same. In order to avoid this threat we have tried to keep the number of subjects in both experimental groups balanced. However, unfortunately the unexpected departure of participants from group IG made the number of individuals in the groups imbalanced. Even though the higher number of subjects in the AG is in favor of the control group, this difference could have an effect on the experimental results especially on the quantity of the requirements identified by each group.

6.3. Future work

Several aspects of our suggested approach require further study. First, although the positive results of this study are very encouraging, more empirical studies are needed to obtain highly significant and more generalizable results. In addition, in this study we mainly aimed to evaluate the effectiveness of online serious games in the requirements elicitation process. However, there is some evidence that this method is also effective in requirements negotiation between software stakeholders. Therefore, more research investigating the effectiveness of online serious games in requirements negotiation is needed. Finally, in this study we used specific types of commercial serious games designed by Innovation Games®. Today, many serious games can be used in different stages of requirements engineering. Each game has its own rules and goals, and for that reason individuals may have different reactions to these games. Thus, studying the impacts of various types of games on customers' participation and collaboration in software projects would be interesting.

7. Conclusions

In this research, we suggested a novel approach called Innovation requirements elicitation in which a group of distributed stakeholders identify and prioritize their requirements through participation in online serious game sessions. The results from our analyses show that utilizing online serious games can be expected to increase the quantity of user requirements. Our findings specifically reveal that while our suggested approach enables less-experienced individuals to identify and provide a higher number of requirements, it also hinders developers from imposing their preferred features to customers. Based on these findings, we believe that utilizing online serious games is an effective method for increasing

motivation among customers to actively participate and engage in the requirements elicitation process. These interactive games provide a visual image of products to customers and as a result increase their understanding of the project and its boundaries. By this, customers are able to actively engage in requirements elicitation processes and play a central role in identifying and communicating their needs and expectation. Furthermore, online serious games enable the development teams to collaborate with a geographically diverse set of customers and users and to easily collect new ideas and quick feedback for creating precise project roadmaps. Finally, using serious games can be a catalyst for discussion and negotiation between software stakeholders, which can be an effective method for resolving requirements conflicts.

Appendix A. Web-based *Prune the Product Tree*

Appendix B. Web-based *Buy a Feature*

Appendix C. A sample of course weekly assignments

SDGE-course 2012, University of Oulu

Assignment for HQ-team: Week 1

You are to simulate the role of a "headquarters"-team of a global software development effort. You have already received the information and training necessary for this simulated exercise. You are urged to act as closely as possible like it would be a "real thing" and act according to your role. All discussions, communications etc. affecting your decisions must be accountable, that is to say: No discussions "off-channel" with other teams. Every piece of communication and action must leave an electronic/paper audit trail for the purposes of the simulation.(Face-to-face meetings with other teams might in reality require lengthy air travels, jet lags, etc. which are not in the interests of this exercise, whether it be in reality or in simulation.)

For this exercise, the simulated physical location of your team will be in the timezone UTC +1 (=CET) hrs. Please direct your actions accordingly.

Task

Your task for this week is:

- Elicit user stories / requirements from the client, user and subcontractor teams
- Co-ordinate the process of collecting the user stories / requirements between different teams.
- Prepare those stories / requirements as a product backlog format.
- Arrange a steering group meeting at the end of the sprint
  - Include representatives from the client team and the teaching staff

Materials

- Templates for user stories and product backlog
- Reading material from the methodology tutorial

Deliverables

Deliverables

This week you are to deliver for the steering group meeting (at the end of the sprint) the following

items:

- Full list of user stories / requirements gathered
- Current version of the product backlog
- Short description of project status and events of the previous week.

References

[1] The Standish Group International, Inc., "'The Chaos Summary 2009,'", 2009.

[2] B. Davey and C. Cope, '"Requirements Elicitation: What is Missing?" *Issues in Informing Science and Information Technology*, vol. 5, no. 1, 2008, pp. 543-551.

[3] A. Eberlein and Leite, Julio, Cesar, Sampaio, do Prado, '"Agile Requirements Definition : A View from Requirements Engineering," *International Workshop on Time-Constrained Requirements Engineering. (TCRE'02)*, 2002, pp. 4-8.

[4] H.K. Edwards and V. Sridhar, '"Analysis of software requirements engineering exercises in a global virtual team setup," *Journal of Global Information Management (JGIM)*, vol. 13, no. 2, 2005, pp. 21-41.

[5] S. Pekkola, N. Kaarilahti and P. Pohjola, '"Towards formalised end-user participation in information systems development process: bridging the gap between participatory design and ISD methodologies," *Conference on Participatory Design: Expanding Boundaries in Design*, vol. 1, 2006, pp. 21-30.

[6] A. Sillitti and G. Succi, '"14 Requirements Engineering for Agile Methods," *Engineering and Managing Software Requirements*, 2005, pp. 309-326.

[7] C.J. Davis, R.M. Fuller, M.C. Tremblay and D.J. Berndt, '"Communication Challenges in Requirements Elicitation and the Use of the Repertory GRID Technique," *Journal of Computer Information Systems*, vol. 47, 2006.

[8] J. Weedman, '"Client as designer in collaborative design science research projects: what does social science design theory tell us?" *European Journal of Information Systems*, vol. 17, no. 5, 2008, pp. 476-488.

[9] B. Ramesh, L. Cao and R. Baskerville, '"Agile requirements engineering practices and challenges: an empirical study," *Information Systems Journal*, vol. 20, no. 5, 2010, pp. 449-480.

[10] L. Cao and B. Ramesh, '"Agile requirements engineering practices: An empirical study," *IEEE Software*, vol. 25, no. 1, 2008, pp. 60-67.

[11] L. Mathiassen, T. Saarinen, T. Tuunanen and M. Rossi, '"A Contigency Model for Requirements Development." *Journal of the Association for Information Systems*, vol. 8, no. 11, 2007, pp. 569-597.

[12] F. Paetsch, A. Eberlein and F. Maurer, '"Requirements engineering and agile software development," *IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises. WET ICE*, 2003, pp. 308-313.

[13] G. Walsham, '"Cross-cultural software production and use: a structurational analysis," *MIS Quarterly*, vol. 26, no. 4, 2002, pp. 359-380.

[14] E. Hossain, M.A. Babar and H. Paik, '"Using scrum in global software development: a systematic literature review," *International Conference on Global Software Engineering (ICGSE)*, 2009, pp. 175-184.

[15] M. Paasivaara and C. Lassenius, '"Could global software development benefit from agile methods?" *International Conference on Global Software Engineering (ICGSE)*, 2006, pp. 109-113.

[16] J. Sauer, '"Agile practices in offshore outsourcing–an analysis of published experiences," *Information Systems Research Seminar in Scandinavia, IRIS*, vol. 29, 2006, pp. 12-15.

[17] L. Layman, L. Williams, D. Damian and H. Bures, '"Essential communication practices for Extreme Programming in a global software development team," *Information and Software Technology*, vol. 48, no. 9, 2006, pp. 781-794.

[18] I. Sommervile, '"Integrated Requirements Engineering: A Tutorial," *IEEE Software*, vol. 22, no. 1, 2005, pp. 16-23.

[19] M. Korkala, P. Abrahamsson and P. Kyllonen, "'A case study on the impact of customer communication on defects in agile software development," *Agile Conference*, 2006, pp. 77-88.

[20] T. Tuunanen, K. Peffers, C.E. Gengler, W. Hui and V. Virtanen, "'Developing feature sets for geographically diverse external end users: a call for value-based preference modeling," *Journal of Information Technology Theory and Application (JITTA)*, vol. 8, no. 2, 2006, pp. 41-56.

[21] M. Fowler and J. Highsmith, "'The agile manifesto," *Software Development*, vol. 9, no. 8, 2001, pp. 28-35.

[22] B. Boehm, "'Get Ready for Agile Methods, with Care," *IEEE Computer*, vol. 35, no. 1, 200, pp. 64-69.

[23] E. Hossain, P.L. Bannerman and D.R. Jeffery, "'Scrum practices in global software development: a research framework," *Product-focused software process improvement*, 2011, pp. 88-102.

[24] J. Sutherland, G. Schoonheim, E. Rustenburg and M. Rijk, "'Fully distributed scrum: The secret sauce for hyperproductive offshored development teams," *Agile Conference*, 2008, pp. 339-344.

[25] VersionOne, "'Annual State of Agile Survey,", vol. 8, 2013.

[26] E. Carmel and R. Agarwal, "'Tactical approaches for alleviating distance in global software development," *IEEE Software*, vol. 18, no. 2, 2001, pp. 22-29.

[27] E.Ó Conchúir, P.J. Ågerfalk, H.H. Olsson and B. Fitzgerald, "'Global software development: where are the benefits?" *Communications of the ACM*, vol. 52, no. 8, 2009, pp. 127-131.

[28] H. Holmstrom, E.Ó Conchúir, P.J. Agerfalk and B. Fitzgerald, "'Global software development challenges: A case study on temporal, geographical and socio-cultural distance," *International Conference on Global Software Engineering (ICGSE)*, 2006, pp. 3-11.

[29] J. Sutherland, A. Viktorov, J. Blount and N. Puntikov, "'Distributed scrum: Agile project management with outsourced development teams," *Annual Hawaii International Conference On System Sciences (HICSS)*, 2007, pp. 274a-274a.

[30] M. Kauppinen, J. Savolainen and T. Mannisto, "'Requirements Engineering as a Driver for Innovations," *IEEE International Requirements Engineering Conference*, 2007, pp. 15-20.

[31] N. Maiden and S. Robertson, "'Integrating creativity into requirements processes: Experiences with an air traffic management system," *International Conference on Requirements Engineering*, 2005, pp. 105-114.

[32] L. Mich, C. Anesi and D.M. Berry, "'Applying a pragmatics-based creativity-fostering technique to requirements elicitation," *Requirements Engineering*, vol. 10, no. 4, 2005, pp. 262-275.

[33] V. Sakhnini, L. Mich and D.M. Berry, "'The effectiveness of an optimized EPMcreate as a creativity enhancement technique for Web site requirements elicitation," *Requirements Engineering*, vol. 17, no. 3, 2012, pp. 171-186.

[34] I. Aaen, "'Essence: facilitating software innovation," *European Journal of Information Systems*, vol. 17, no. 5, 2008, pp. 543-553.

[35] I. Aaen, "'Roles in Innovative Software Teams: A Design Experiment," *Human benefit through the diffusion of information systems design science research*, 2010, pp. 73-88.

[36] L. Hohmann, "'Innovation games: creating breakthrough products through collaborative play,", 2006.

[37] C. Le Marc, J. Mathieu, M. Pallot and S. Richir, "'Serious Gaming: From Learning Experience towards User Experience," *International Conference on Concurrent Engineering, ICE*, 2010, pp. 21-23.

[38] H.W. Chesbrough, "'Open innovation: The new imperative for creating and profiting from technology,", 2003.

[39] E. Von Hippel, "'Democratizing innovation,", 2005.

[40] M. Sawhney, R.C. Wolcott and I. Arroniz, "'The 12 different ways for companies to innovate," *MIT Sloan Management Review*, 2011, pp. 28-34.

[41] A. Botero, S. Vihavainen and K. Karhu, "'From closed to open to what?: an exploration on community innovation principles," *International MindTrek Conference: Everyday Life in the Ubiquitous Era*, 2009, pp. 198-201.

[42] C. Caulfield, J. Xia, D. Veal and S.P. Maj, "'A systematic survey of games used for software engineering education," *Modern Applied Science*, vol. 5, no. 6, 2011, pp. 28-43.

[43] M. Zyda, "'From visual simulation to virtual reality to games," *Computer*, vol. 38, no. 9, 2005, pp. 25-32.

[44] M.Z.J. Carlos and G. Awad-Aubad, "'Requirements Game: Teaching Software Project Management," *CLEI ELECTRONIC JOURNAL*, vol. 10, no. 1, 2007.

[45] E. Knauss, K. Schneider, K. Stapel and K.S. De, "'A Game for Taking Requirements Engineering More Seriously," *International Workshop on Multimedia and Enjoyable Requirements Engineering-Beyond Mere Descriptions and with More Fun and Games*, 2008, pp. 22-26.

[46] R. Smith and O. Gotel, "'RE-O-Poly: A Customizable Game to Introduce and Reinforce Requirements Engineering Good Practices," *Department of Computer Science, Pace University, New York*, 2008.

[47] C. Wohlin, P. Runeson, M. Host, C. Ohlsson, B. Regnell and A. Wesslén, "'Experimentation in Software Engineering: an Introduction,", 2000.

[48] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell and A. Wesslén, "'Experimentation in software engineering,", 2012.

[49] E. Carmel and J.A. Espinosa, "'I'm working while they're sleeping: Time zone separation challenges and solutions,", 2011.

[50] S.W. Huck, W.H. Cormier and W.G. Bounds, "'Reading statistics and research,", ed. 6th Edition, 2011.

[51] P. Abrahamsson, O. Salo, J. Ronkainen and J. Warsta, "'Agile Software Development Methods: Review and Analysis,", 2002.

[52] P. Abrahamsson, J. Warsta, M.T. Siponen and J. Ronkainen, "'New Directions on Agile Methods: A Comparative Analysis," *International Conference on Software Engineering*, vol. 6, 2003, pp. 244-254.

[53] K. Ting, "'Precision and Recall," *Encyclopedia of machine learning*, 2010, pp. 781-781.

[54] S. Robertson, "'Evaluation in information retrieval," *Lectures on information retrieval*, 2001, pp. 81-92.

[55] V. Raghavan, P. Bollmann and G.S. Jung, "'A critical investigation of recall and precision as measures of retrieval system performance," *ACM Transactions on Information Systems (TOIS)*, vol. 7, no. 3, 1989, pp. 205-229.

[56] M. kelanti and S. Saukkonen, "'Comparison of Stakeholder Identification Methods – The Effect of Practitioners Experience," *International Conference on Software Engineering Advances*, 2014, pp. 90-96.

[57] T. Niinimaki, A. Piri and C. Lassenius, "'Factors affecting audio and text-based communication media choice in global software development projects," *International Conference on Global Software Engineering (ICGSE)*, 2009, pp. 153-162.

[58] L.D. Setlock, S.R. Fussell and C. Neuwirth, "'Taking it out of context: collaborating within and across cultures in face-to-face settings and via instant messaging," *ACM Conference on Computer Supported Cooperative Work*, 2004, pp. 604-613.

[59] M. Svahnberg, A. Aurum and C. Wohlin, "'Using students as subjects-an empirical evaluation," *Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, pp. 288-290.

[60] P. Berander, "'Using students as subjects in requirements prioritization," *Proceedings of International Symposium on Empirical Software Engineering ISESE'04*, pp. 167-176.

Hadi Ghanbari is pursuing his Ph.D. at the Department of Computer Science and Information Systems, University of Jyväskylä, Finland. Simultaneously, Ghanbari is involved in international industry-driven R&D projects in the area of Software-intensive Systems and Services. His research interests lie in the area of software development methods and processes with an emphasis on agile methods and global software development, empirical software engineering and innovative requirements engineering. He received his Bachelor's degree in Software Engineering from Azad University, Iran, in 2005 and his Master's degree in Information Systems from University of Oulu, Finland in 2012. In addition to his academic career, he has worked in IT industry for several years.

Jouni Similä is full professor of Software Engineering at the Department of Information Processing Science, University of Oulu and recently until 2010 a Visiting Scholar at International Computer Science Institute and Center for Open Innovation, UC Berkeley. His present research interests include software process assessment and improvement, empirical software engineering, product road mapping, agile software development, requirements engineering, global software engineering and open innovation. He has authored more than 80 research publications in the Information Systems and Software Engineering fields. Prior to assuming an academic position, he worked in software industry close to 20 years.

Dr. Jouni Markkula is Senior Research Fellow at the University of Oulu, Finland. He received his Ph.D. in Computer Science from the University of Jyväskylä in 2003. His main research areas are knowledge management, decision making, privacy, software engineering, and service design. He has lead and managed several research projects in these fields, in co-operation with industry. Before the University of Oulu, Dr. Markkula was working at the

Information Technology Research Institute of the University of Jyväskylä as a Research Director.