School of Electronics and Computer Science
Faculty of Engineering, Science and Mathematics
University of Southampton

Scrum Game:
An Agile Software Management Game

by

Aikaterini Gkritsi

23/09/2011

Project supervisor: Dr Ken S Thomas
Second examiner: Professor Michael J Butler

A dissertation submitted in partial fulfilment of the degree of

MSc Software Engineering

by examination and dissertation

Abstract

For the past few years, in their attempt to avoid the heavyweight bureaucracy of traditional project management methods such as the Waterfall model, companies have started incorporating agile methods (e.g. Extreme Programming, Scrum, Crystal) for their project development. These methods are characterised by their incremental and iterative delivery, their ability to incorporate change at any stage of the project lifecycle, as well as their small and co-located teams. Even though these methods are included in the syllabus of many software engineering modules at university level, many students currently feel more confident with traditional, rather than agile methods. Many employers find that recent graduates are not equipped with the desired skills of a software engineer because, even though they are knowledgeable in the different software engineering practices, they lack practical experience of these methods. The combination of these two factors show that the university's approach to teaching software management methods is only theoretical and it does not give students the opportunity to apply them to their projects so they can get a better understanding of their use.

The project developed the prototype of a computer game that simulates the use of the Scrum method within different projects, named Scrum Game. The game is supplementary material for a lecture course, and its purpose is to guide students through the Scrum lifecycle. Students can thereby get a small glimpse of the different phases of Scrum, the way that the different Scrum roles interact with each other, and the way that Scrum is used to implement real projects. In addition, the Scrum Game has an administrator mode enabling lecturers to view a log of the progress of all their students in the game. They can use this mode to create new projects or to alter existing ones by adding new tasks or problems, thereby adjusting the level of difficulty to the level of their students, or so that it fits their teaching.

The web-based system was developed using PHP, MySQL, HTML, CSS, JavaScript, AJAX (jQuery) and Google Charts API. The system was thoroughly tested against the initial requirements and other system tests. The Scrum Game was evaluated by 22 peer colleagues reading for an MSc in Software Engineering at the University of Southampton, to identify whether the system achieved its goal of introducing students to the Scrum methodology and reaching a deeper understanding of its practical use during

project implementation. The results of a questionnaire showed that little prior knowledge was assumed during the game, and that 86% of the participants felt that the game helped them learn more about Scrum. When asked, “Do you think that if this game was part of your Project Management module, would you get a better understanding about Scrum?” an impressive 95% (21 out of 22 participants) agreed that the game would be helpful, and rated the system 8 out of 10 on average.

Acknowledgements

First of all, I would like to say a very big thank you my supervisor Dr Ken S Thomas who, despite all the difficulties and problems that occurred during the project, was always there for me, happy to help, advise and guide me. Also, I would like to thank Mr Quintin Gee for his invaluable help proof-reading this report. Finally, I would like to thank all the people that participated in my survey because, without them, I would not be able to complete this project.

Table of Contents

1 Introduction

Over the past few years, more and more companies have started implementing their projects using agile methods, moving away from traditional project management methods like the Waterfall model (Ambler, 2008). Even though the turn to agile methods started the last few years, the notion of iterative and incremental development was introduced around 1960's (Abrahamsson, 2003). As Larman says " Iterative development is an approach to building software (or anything) in which the overall lifecycle is composed of several iterations in sequence and each iteration can be seen as a self-contained mini-project contained of activities such as requirement analysis, design, programming and test" (Larman, 2004). These are only two of the most important characteristics of agile methods. In particular, agile methods are characterised for their iterative and incremental development, their small and frequent releases and adaptability to change. Unlike traditional project management methods where in order to start the implementation a very precise and detailed plan needs to be conducted and any change to it causes delays and increase of project cost, agile methods have the ability to easily alter their plan and adapt it to the change minimising its effect (Highsmith, 2010).

Although agile methods seem to play such an important role in software engineering, currently, the approach to agile methods from many universities along with the University of Southampton is only theoretical so students do not have the opportunity to understand their actual use and impact in software development. The effect of this is that many employers find that even though graduates have knowledge about the different agile principles, they are not able to apply them in real projects because they lack on practical experience on them (Oh, 2002). To tackle this problem, many courses have started incorporating simulation games to train their students, the same way that pilots get trained (Collofello, 2000). Also, it was found using this kind of games in parallel with lectures, students improve their performance and achieve better results in their final examinations (Clua et al. 2006).

The aim of this project was to create a prototype of a game that simulates the use of Scrum agile method within different projects, to help students learn and understand in depth every stage of its lifecycle. This system was designed to be used as supplementary material in lectures, so it was made fully customisable and each lecturer can change the already existing projects or create their own, so that they fit to their syllabus or focus on specific software engineering concepts.

1.1 Project Description

The Scrum Game is a prototype of a system in the form of a game, which is designed to be used as supplementary material in software engineering related modules at universities. In particular, the game is targeted to students of the IT field, as well as to project management literate students. Through this system, players will be able to understand in depth the Scrum agile method and go through every step of its lifecycle so they can see how this method is used within different projects. Also, this game can be modified from the module leader so that it follows their syllabus. Specifically, the module leader acts as the administrator of the system and they are able to track the progress of all their students within the game and create their own projects or alter the existing ones, adjusting the its difficulty depending on the level of their students. It was decided to focus on the Scrum methodology because it is usually combined with Extreme Programming, which is the most commonly adopted agile method, offering products of better quality, making the teams more productive, giving better customer satisfaction and relatively lower costs compared to other methods (Ambler, 2006; Parsons et al. 2007). Another reason for Scrum is because a survey showed that only 22% of the participants were aware of Scrum, even though it is the second most commonly used method (Ambler, 2006; Gkritsi, 2010).

1.1.1 Detailed Description

The Scrum Game can be divided into 2 different parts, the game and the administrator mode. The game mode is for players to login and start playing the game, while the administrator mode is where lecturers can track the progress of all their students and create their own projects.

As far as the game mode is concerned, to start the game, users are required to register by inserting a username and password. The username is necessary to track the progress of each player as well as to keep their score during, and at the end of, the game. The password is hashed, using the SHA-1 hash function to ensure that it is stored securely in the database. On the first page, players can find information on the purpose of the game, what it tries to achieve, and a general description about their role and responsibilities within the game. On the next page, players are given detailed instructions on how to play the game and are provided with a selection of three (or potentially more) different projects that could be implemented by their teams. Each project has a different duration, which is measured in the number of Sprints (Section 2.3.2). To avoid making the game drawn-out but still represent all the stages of Scrum, it was decided to set the duration of each Sprint as 2 days.

Within the game, users have the role of Scrum Master (Section 2.3.2) and are therefore required to manage, coordinate, and help, a team of people to implement the assigned project. To assemble their team, players can choose their members from a variety of people with different positions, salaries, and levels of experience. Again, for the purposes of this game, to keep it small, simple and interesting for the participants, it was decided to proceed with teams of three. Further, because the game tries to represent the phases of the Scrum lifecycle step-by-step, as soon as the team is built, the game (i.e. the Scrum lifecycle) starts. The next stage is for the players to estimate the duration and the effort of each task represented in the Product Backlog (Section 2.3.2) of the selected project. They then proceed to the development of the Sprint backlog (Section 2.3.2), assigning each task to particular sprints. As soon as this phase is complete, the first Sprint starts.

As previously mentioned, for this game each Sprint lasts two days. As the Scrum lifecycle dictates, at the beginning of each day the team gathers for a Daily Scrum Meeting (Section 2.3.2), where each team member has to answer three questions for the Scrum Master: 1. *What have you done since last meeting?* 2. *What will you do now and for the next meeting?*, and 3. *What problems are you encountering?* If a member has any problems, then the Scrum Master should give them appropriate advice to solve that problem. Adopting the role of Scrum Master, game players have to choose between three different options that correspond to advice that could solve the issue that has occurred. Every option is credited with a specific number of points, which correspond to the quality of the overall product, i.e. the quality of the final product depends on the advice that the Scrum Master will choose to give to their team.

During every Sprint, the Scrum Master can view the Burndown chart (Section 2.3.2) that corresponds to the progress of the team until that point in time. This chart changes every day and shows the amount of work remaining. Its trend changes, depending on the quality of the product at that particular time. As soon a Sprint is finished, the player proceeds to the Review Meeting (Section 2.3.2), where the client reviews and evaluates the completed part of the project. The system evaluates the quality of the product at that point. If the quality is lower than 50%, then the player has to repeat the completed Sprint until they reach the desired quality, otherwise they can proceed to the next Sprint. This procedure is repeated until all the Sprints are complete. When the project finishes, players can see how they did during that game, the history of their previous games and a Top 5 high-score table. Through these features, users will be able to get a complete idea of their progress and can compare it with their colleagues, which may motivate them to perform better on their next try. At the end, players can either logout or start another game.

In the administrator mode, lecturers can see a complete log of their students' progress. In particular, they can view the project, score and number of games each student has attempted and completed. They also have the facility to alter existing projects by inserting additional tasks, employees, employee skills, questions, etc. The system offers them flexibility to create their own projects, inserting the project title, the number of Sprints it will have, its tasks, possibly additional employees and their skills, as well as the different questions and options that the students will have to select during the Daily Scrum Meetings. This feature is very important, because lecturers can adjust the level of difficulty of the game depending on the level of their students, or modify it so that it fits their teaching.

2 Background research and literature search

Before specifying the requirements of the system, it was necessary to perform background research in the literature.

The Scrum Game project was inspired by the Agile Game project that the author undertook for the undergraduate degree (Gkritsi, 2010). In particular, the aim of the Agile Game was the creation of an educational game to be used as additional material during lectures and which introduced players to the most commonly used agile methods: Extreme Programming (XP), Scrum, Crystal Clear, Feature Driven Development (FFD), and Dynamic Systems Development Method (DSDM). Through this game, players could learn the different agile methods and their techniques, while implementing a project divided into 4 different phases, each one corresponding to a phase of the project lifecycle (User Stories/Requirements, Design/Planning, Implementation, and Acceptance Testing). The users played the role of Project Manager and were responsible for deciding the methods and techniques that would be appropriate for the different phases of the project. Every choice was scored, according to the selected method, technique and phase of the project. At the end of the game, users could see their final score and how it compared with the highest scores by other players. The system was evaluated by peer students. The results showed that the game was rated quite high (7 out of 10), while over half of them felt that the game was enjoyable to play. The majority thought that too much prior knowledge was assumed, and 44% thought that the system fulfilled its educational aim.

Having the Agile Game in mind, and taking into consideration the user feedback and potential use of such a game in lectures, it was decided to expand it and create another educational game. This focuses on one agile method (Scrum), clearly representing the different stages of that method, and making it as simple and informative as possible, while at the same time enjoyable and attractive to potential players.

2.1 Research Outline

To start the research for the Scrum Game it was necessary to investigate the Agile Game literature search in order to differentiate it from previous research. The outcome of this procedure was more thorough understanding of the features of some traditional project management methods such as Waterfall and V-Model (Sections 2.2.1 & 2.2.2) and of the most common agile methods (Section 2.3) Extreme Programming and Scrum (Amber, 2006; Lindvall et al, 2002). Although a variety of both traditional and agile project management methods were researched, the author focused on clarifying the Scrum method, as this would be the subject of the system.

Work continued by trying to answer whether creating the system as game would be beneficial to students, and what kinds of game were currently used for educational purposes at university. The question to be answered was, "What characteristics should the system have to ensure it will accomplish its educational aim and help students get a better and deeper practical understanding of Scrum?" Lastly, reviewing existing games allowed their disadvantages to be detected, so identifying the way in which the Scrum Game can be better. The following sections summarise the results of the literature and background research.

2.2 Traditional Project Management Methods

Years ago, when software developers had to implement a project, they did not put much effort into planning and designing the overall product as the systems were small. However, as soon as systems started getting bigger and more complicated, the need to define a specific process of software development became clear and methods like the Waterfall model started to emerge (Suganya & Mary, 2010). The methods characterised as traditional are the ones that put a lot of effort and time into the planning, design and analysis of the final product. A short description of some of these methods now follows.

2.2.1 Waterfall Model

The Waterfall model, which is sometimes referred to as the *classic life cycle*, (Pressman, 2009) is the most intuitively obvious way to implement a project (Mall, 2004). In particular, the Waterfall model takes the processes of specification, development, validation and evolution and considers them as five distinct and linear phases of a project, i.e. requirements, design, implementation, verification and finally maintenance (Figure 2.1) (Somerville, 2007; Huo, 2004).

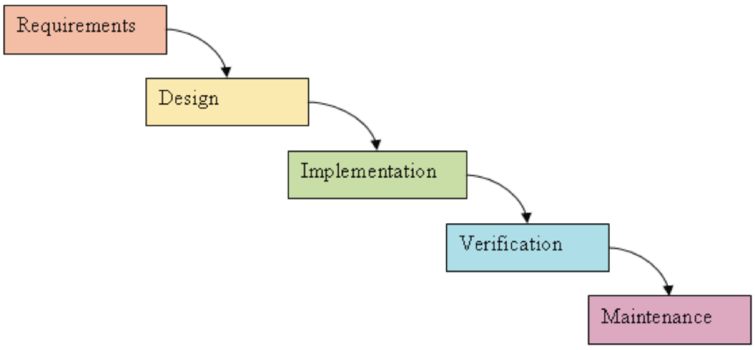


Figure 2.1: Waterfall model diagram (Serena, 2007)

Using this model, teams can produce a very detailed plan and user requirements that can be used throughout the implementation and testing phase, despite the fact that it is time-consuming (Aguanno 2004). The drawback of this model is that it assumes that all the user requirements will be gathered at the beginning of the project and the customer will not change their mind during the implementation. However, this never happens in real life. It is very common for clients not to know clearly what they want from the system, so during the project, changes need to be incorporated in the existing plan, which may cause delay to the final product, or increase the overall cost, or both.

The companies that usually adopt the waterfall model are those responsible for very large, complex and long-lasting projects. When smaller businesses try to implement their projects with Waterfall, they usually are dominated by the whole software development process (Somerville, 2007), which is why they started developing and adopting the agile methods.

2.2.2 V – Model

The V-Model was first introduced by Paul Rook during the 1980's to improve the efficiency and effectiveness of software development (Qian & Zheng, 2009). In particular, the V-Model is "a variation in the representation of the Waterfall model. It depicts the relationship of quality assurance actions to the actions associated with communication, modelling, and early construction activities" (Pressman, 2009).

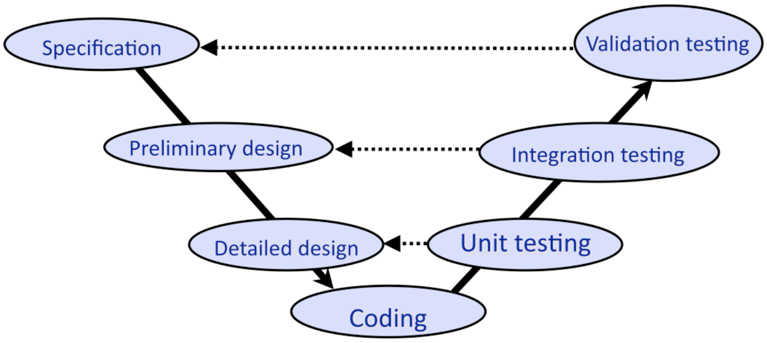


Figure 2.2: V-Model diagram[1]

A characteristic feature of this model is that every task has to be implemented linearly and sequentially. Specifically, to start the execution of one phase, the previous one needs to be finished and tested. Similar to the Waterfall model, the V-Model cannot accommodate change easily as there is a chance that some phases have already been completed and tested. Figure 2.2 shows that V-Model[2] uses unit and integration testing to validate the detailed and the preliminary design respectively. When both the implementation and the design phase of the product are complete, then the overall system is tested against its requirements using validation testing.

2.3 Agile Methodologies

As previously mentioned, agile methods are characterised by their iterative and incremental delivery, as well as their ability to adapt to change at any time during the project lifecycle. In their article, Abbas et al. characteristically say "agile methods are a reaction to the bureaucracy of the engineering methodologies" (Abbas et al 2008). This happened because small to medium sized businesses wanted to avoid the heavyweight implementation approach of traditional project management methods that large businesses were using, and focus more on the software itself rather than solely on its design and documentation (Somerville, 2007). The principles of agile methods are clearly stated in the *Manifesto for Agile Software Development*[3]. Some of the principles are: individuals and interactions should always be more important than processes and tools, software is more important than comprehensive documentation, customer collaboration should be more important than contract negotiation and finally response to change is always more important than following a plan (Beck, 2000).

[1] Available at: https://secure.ecs.soton.ac.uk/noteswiki/w/COMP6038
[2] Available at: https://secure.ecs.soton.ac.uk/noteswiki/w/COMP6038
[3] Available at: http://agilemanifesto.org/

Surveys showed that the most commonly used methods within industry are Extreme Programming (XP), Scrum, Feature Driven Development, and Agile Unified Process (Parsons et al. 2007; Lindvall 2002). A brief description of the features of Scrum and XP now follows.

2.3.1 Extreme Programming (XP)

Extreme Programming (XP) is one of the most popular agile methods (Abrahamsson et al., 2003). In particular, XP focuses more on the implementation and the development of the product. The XP lifecycle starts with the team collecting the client's requirements in the form of **user stories** (Figure 2.3). These user stories are used to conduct the **Release Plan**, where the team splits the tasks into iterations and specifies the order in which these user stories will be implemented, and in which iteration. As soon as the Release Plan is defined, the team proceeds to the **Iteration** phase. Each iteration needs to be as short as possible, usually lasting between 2 and 4 weeks. Following its completion, part of the functionality is delivered (Beck, 2000). Small iterations are characteristic of XP because this way it is easy to adapt to any changes that might occur in the specifications, and get frequent feedback from the customer on the project under development. Each iteration concludes with the **Acceptance Testing** phase. During this phase, the team implements acceptance tests which test the product against the user stories. If the system passes all the necessary tests and satisfies the user requirements, the team releases the developed software (Highsmith 2002; Serena, 2007).

In XP, teams are usually co-located and consist of about 10 people. This allows the team to work more effectively, reinforcing their communication, team spirit and collaboration (Highsmith 2002).

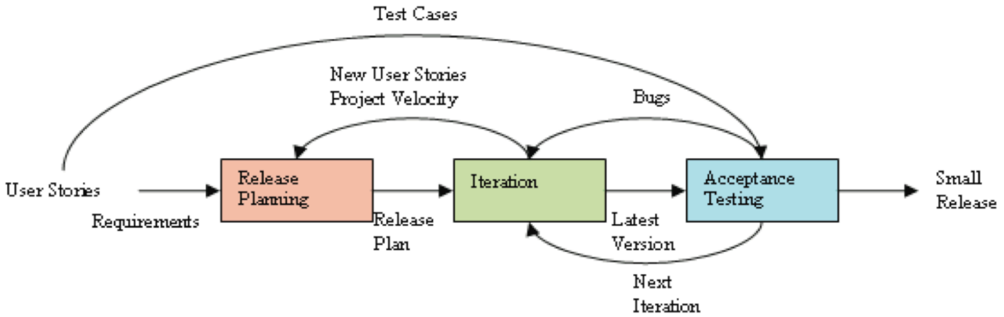


Figure 2.3: XP diagram (Serena, 2007)

“XP is a collection of well-known software engineering practices” (Abrahamsson et al., 2003) and some of its best known practices are pair programming, test-driven development, 40-hour work week, continuous integration and close customer participation (Abrahamsson, 2003; Abrahamsson et al., 2003; Salo & Abrahamsson, 2008).

2.3.2 Scrum

While XP focuses on the product implementation, Scrum emphasises the project management perspective of the project (Highsmith, 2002a). “Scrum is not a process or a technique for building products; rather, it is a framework within which you can employ various processes and techniques. Scrum makes clear the relative efficacy of your product management and development practices so that you can improve” (Schwaber & Sutherland, 2011). Through this method, responsibility and control is moved away from central scheduling, and is assigned to the individual teams which focus on self-organisation, daily team measurements, and avoidance of predefined steps (Schwaber, 2004; Highsmith, 2002 b). Scrum has been used in different projects by many leading companies, such as Google, Yahoo!, Siemens and British Telecoms (Benefield, 2008; Fernandes & Sousa, 2010).

The Scrum lifecycle is fairly straightforward and can be seen in Figure 2.4 below. The lifecycle is divided into small iterations that last between 15-30 calendar days called **Sprints**. Within a Scrum project, people have the role of Product Owner, or Scrum Master, or be part the Development Team.

- **Development Team:**

The Development Team is usually small, co-located, and consists of 5 to 9 people (ideally 7) of different positions: Developers, Architects, QA Engineers, TDD Engineers, UI Designers and Business Analysts. Unusually, it may also include Marketing staff, Technical Writers, and Administrative staff (Playfair, 2008).

- **Scrum Master:**

The Scrum Master is like the Project Manager, but they act more as facilitators and not managers (they are a “servant-leader for their team” (Schwaber & Sutherland, 2011)), as they are responsible of helping the team with any problems they might have during the implementation, for leading meetings, and for finding ways to improve their productivity[4] (Schwaber & Sutherland, 2011; Playfair, 2008; Pressman, 2009).

- **Product Owner:**

The role of the Product Owner is usually held by the client or stakeholders, managers, etc., and they are the people who specify the requirements of the product to be implemented. Apart from that, they are responsible for the production of the **Product Backlog**, a document that contains all the desired tasks of the project prioritised. They inspect and evaluate the team's work at the end of each Sprint (Schwaber, 2004; Playfair, 2008).

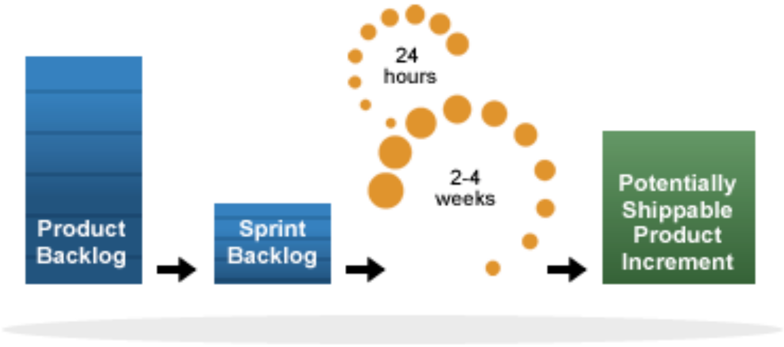


Figure 2.4: Scrum method diagram (Scrum Alliance, 2009)

To start a Sprint, the Development Team and the Scrum Master attend the **Sprint Planning Meeting** that lasts 8 hours for a 1-month Sprint. During the first 4 hours of this meeting, the Product Owner shows to the team the prioritised Product Backlog and explains to the team in more detail what is expected from every task. The team questions them about the content, clarifications, etc. (Schwaber, 2004) In the second half of that meeting, the team selects which project tasks they will be able to implement by the end of the Sprint. They inform the Product Owner of their plan and they then estimate the duration and the effort of each of these tasks. After that, they create the **Sprint Backlog** which specifies which task will be implemented in which Sprint. As soon as this meeting is over, the Sprint starts. During the Sprint, the Scrum Master calls their team together at the same time each day for a 15-30 minute

[4] Available at: http://www.scrumalliance.org/articles/39-glossary-of-scrum-terms

meeting, the **Daily Scrum Meeting**. In this meeting, the Scrum Master asks each member of their team the following three questions:

- What did you do since the last meeting?
- What problems did you encounter?
- What will you do now and for the next meeting?

The purpose of these questions is to ensure that all the team is at the same level, that everyone knows their tasks, and finally to identify any problems. When the Sprint is complete, the **Review Meeting** is held where the team presents the part of the product that has been developed to the Product Owner. In this meeting the Product Owner evaluates the work that has been done and if it meets their standards, they accept it and allow the team to proceed with the next Sprint. If the product is not up to their standards, or the team did not manage to complete all the planned tasks of the Sprint, then the Sprint Backlog is reviewed. After the Review Meeting, the team gathers for a **Retrospective Meeting** to evaluate the progress and performance of the team during the Sprint (Hossain, 2009). Finally, changes in the Product Backlog are not introduced during the Sprint, thus allowing the team to work in a short-term, but stable environment (Schwaber, 2004; Pressman, 2009).

2.4 Games in education

There has been a lot of discussion about whether games should be used as a way to educate students. In particular, there is disbelief about whether these games achieve their goal and teach students the required material, or are just a means of amusement. Despite this, many games have been implemented and used within education and the results were quite promising (Oh, 2002; Basturk, 2005; Moreno-Ger et al., 2007; Amory et al. 1999).

2.4.1 Use of games in education

As previously mentioned, employers believe that students lack practical software engineering skills. Because software project management is complex, and the implementation of a project is expensive and usually takes a long time, the best way to gain practical experience in applying software engineering principles, taught during lectures, is through simulation games (Collofello, 2000; Dahiya, 2010). Collofello argues that the use of simulation games to train future project managers is similar to the use of flight simulators for pilots, as part of their training. "A pilot could not be expected to learn piloting skills from a text or instructor alone, because a pilot needs hands-on, active experience to become effective. Similarly, hands-on, active experience is needed to build effective project management skills" (Collofello, 2000).

Through simulation games, students have the opportunity to "virtually" participate in a real-world project and not just in-class projects. This will let them interact with teams of other people, deal with unexpected results, plan and be responsible for the project budget, things with which students do not have to deal with in in-class projects (Baker et al., 2005; Van Der Duim et al., 2007).

The usual types of simulation games are *role-playing simulations*, which focus on the interpersonal skills of the player and present them day-to-day activities and tasks in their role, *strategy simulations*, where the information and responsibilities of the player are more abstract that in real life so that they have the opportunity to implement long-range strategies and see their results (as in the Scrum Game), and *skill simulations*, whose aim is to train people to learn a specific task or skill, using a variety of skills to complete the desired task (Ferrari et al., 1999).

2.4.2 Desirable characteristics of educational games

Designing an educational game for any discipline is challenging because its aim is to ensure that its users will manage to understand and learn the concepts that the game introduces (Asuncion et al., 2011). To achieve the goals, literature shows that educational games need to have specific characteristics. In particular, the information that students need to learn needs to be spread throughout the game, so that they will see and understand the way it is applied (Gee, 2003). A significant criterion in the success of the game is the complexity of the problem that it presents because, for example, real projects are very complex and long-lasting, something that cannot apply in a game. Furthermore, it needs to be fun and with interesting graphics in order to attract students' attention and engage them in the game (Amory et al. 1999).

The system should be adaptable so it can be adjusted to the student's needs and knowledge level. It must also allow the game instructors to monitor the student's activity to be able to assess their activity and progress (Moreno-Ger et al., 2007; Moreno-Ger et al., 2008).

2.5 Existing educational games

To specify the precise characteristics of the Scrum Game, some research on similar types of game had to be done. The focus was on educational games used at university level. Both traditional and agile project management games were addressed to get a better understanding of the characteristics and the features that an educational game concerning project management methods needed to include. It was thus possible to identify the features that the Scrum Game could adopt to achieve its educational aim.

2.5.1 Overview

Games exemplifying both traditional and agile project management methods were researched. There is a plethora of games on both types of method and of different kinds. A detailed description of these games is now given.

- **The Software Management Game**

The Software Management Game is a system that has been implemented by Dr P W Garratt from the University of Southampton. The game "simulates a project to develop a generic computerised information system" (Garratt, 1995) and is still used in both undergraduate and MSc courses at the University of Southampton as part of the Project Management and Quality Assurance module, since its launch in 1987. The aim of this game is to familiarise students with the notion of teamwork and the different concepts of professional software engineering such as staff scheduling, negotiating, computer-aided software engineering (CASE), and cost-benefit analysis.

The game simulates a 2-year project, which is divided into 10 to 12 sessions of 30-minutes. For the purposes of the game, students form teams of four, where each member plays the role of the Quality Manager, the Technical Manager, the Financial Manager, and the Overall Manager. In each session, players are given forms to record their decisions concerning the number of staff that are necessary to complete a particular task, the methods to apply to implement these tasks, and the tools the employees might need during the implementation.

At the end of each of these sessions, the game administrator (i.e. the lecturer) feeds these decisions into the system. The system then generates a report showing the detailed costs of the team decisions, which tasks are completed, percentage progress, and quality indexes for the Technical, Financial, Technical and Overall Manager. To make the system realistic, randomness is introduced into the game to determine the number of employees that each team can hire at a time (lecturers use darts).

To win, each team seeks for optimum balance between cost, schedule and quality. The team that finishes the project first, with the lowest cost and with the best possible quality, wins. (Garratt, 1995).

Disadvantages
Users do not interact with the actual system.
Not enough feedback on understanding why some tasks where not completed.
Players cannot understand with what criteria the system produces the results shown in the report.
The obligations of each managerial role that players had, where not distinct during the game.

Table 2.1: Software Management Game disadvantages

- **Agile Hour**

The Agile Hour (Lubke & Schneider, 2005) is a game targeted at both Computer Science students and IT professionals who want to learn more about XP methodology. The game lasts 70 minutes and it consists of 7 different phases, each one lasting for 10 minutes (Story Cards & Spike, Estimation of Priorities and Effort, Iteration I planning, Iteration I, Product Presentation & Estimation of Priorities and Effort, Iteration II planning, and Iteration II). Players are required to follow the project requirements and construct a product from Lego bricks instead of actual code. To play this game a team of 5-10 people is required. Some prior knowledge is assumed, so players need be familiar with agile methods and have a clear understanding of the basic principles of XP.

Before starting the game, players are given the goal of the project that they have to implement and some instructions, and then have to choose their roles. Two need to play the role of customers, while the rest play developers. As soon as these choices are made, the game starts. In the first phase, the customers specify the requirements of the product in the form of story cards, while the developers start building a prototype of the product in pairs. The procedure is supervised by the trackers, i.e. the lecturers, who answer any questions and help when problems occur. In the next stage, the developers present the prototypes they have built to the customers, and get "marked" by the trackers. At the same time, the customers explain their story cards to the developers so they can prioritise them and estimate their duration. The duration is estimated in points. These points correspond to the points that the trackers assigned to the prototype. The prototypes are destroyed at the end of this phase. In the third phase, the customers choose the next story card to be implemented. The customers can select any number of story cards to be developed, but the total effort should not exceed the points gathered from the prototype in the previous stage.

In the Iteration I phase, the developers implement the selected story card using pair programming. This technique is applied with one member looking for a particular Lego brick, while the other assembles another brick in the model. When 10 minutes have passed, the team goes to the next phase, while the pair of developers presents the model up to that point to the customer, who checks to see if it meets their expectations based on their story cards. The points of the successfully completed product features are then added, and the customers define new story cards for the next iteration.

For the planning of Iteration II, the developers are divided into new pairs, and plan how to implement the new story cards. Finally, the second iteration proceeds as previously, with one developer trying to find the appropriate brick and the other putting it on the model. When the final 10 minutes expires, the team presents their product to the customers again, who reject or accept it. The team whose product is accepted, win (Lubke & Schneider, 2005).

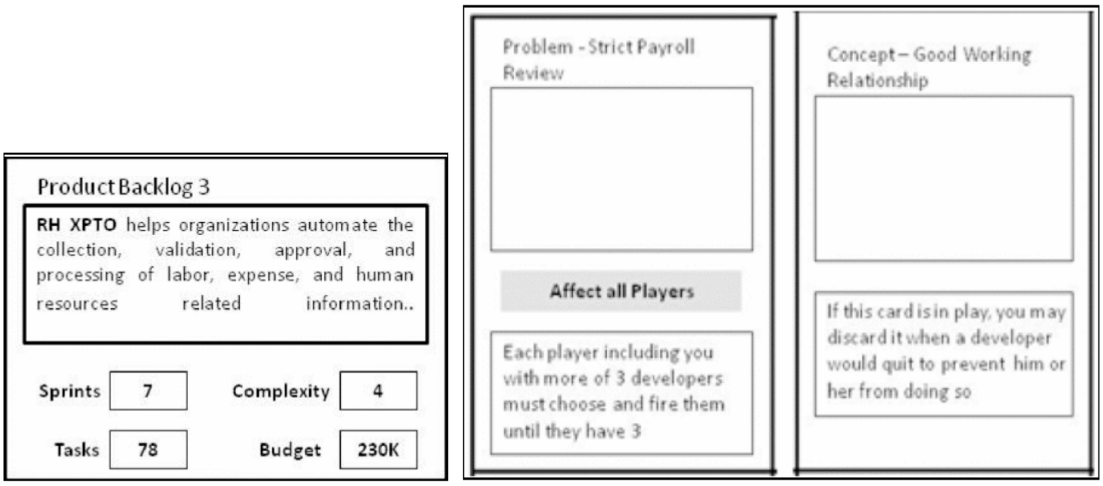
Disadvantages
Duration.
Some players may not have the opportunity to be the ones assembling the final product.
Evaluation showed that the game was tiring and intensive for the participants.
Evaluation found that it was difficult for the players to estimate the effort of the tasks.

Table 2.2: Agile Hours disadvantages

- **PlayScrum**

PlayScrum is a card game played with 2-5 players, which teaches Scrum to students at University. The game contains different projects with different durations (Sprints). It includes a board for every player, Product Backlog cards that determine the features of the different projects, Problem Cards to be used against other players, Concept Cards that are used to solve the issues that the Problem Cards cause, Developer Cards that represent the members of the team, Artefacts that act as the tasks implemented by the team during the game, and a dice. The game winner is the one who completes all the tasks without any errors, or the person with the highest percentage of completed tasks without errors at the last phase of the project lifecycle (Fernandes & Sousa, 2010).

To start the game, a Product Backlog card is drawn (Figure 2.5). This card specifies the number of Sprints that this project has, its number of tasks, its complexity, the cost of the red Artefacts (with more than 60% bugs) and blue Artefacts (with more than 20% bugs), and the project budget. Each player then gets 2 Developer cards to build their team, but the sum of their salaries may not exceed the total budget. After that, players assemble their boards with their cards separated into 4 piles (Developer cards, one for the Problem and the Concept cards, blue Artefacts, and red Artefacts,) and then they roll the dice (Figure 2.5).



Product Backlog 3

RH XPTO helps organizations automate the collection, validation, approval, and processing of labor, expense, and human resources related information..

Sprints	7	**Complexity**	4
Tasks	78	**Budget**	230K

Problem - Strict Payroll Review

Affect all Players

Each player including you with more of 3 developers must choose and fire them until they have 3

Concept – Good Working Relationship

If this card is in play, you may discard it when a developer would quit to prevent him or her from doing so

Figure 2.5: Example of PlayScrum game cards (Fernandes & Sousa, 2010)

Players play in turns clockwise. If the dice rolls 1-3, they can pick that many cards from the Problem/Concept stack. If the dice rolls 4-6, they pick only 2 cards from the Problem/Concept stack but also pick from the Developer stack the number on the dice minus 3. During the game, players may hold a maximum 6 cards. When they exceed this, the additional cards need to be disposed of before the next player's turn. During their turn, players are able to hire and fire developers, as well as inspect and correct their Artefacts at any time. Every check or correction of an error within a task costs 1 point.

When players have finished hiring/laying off developers, checking or refactoring the existing Artefacts, etc., they may receive Problem Cards from other players to slow them down and decrease their product's quality. A Sprint reaches its end after 4 rounds. At the end of each Sprint, the game moderator (i.e. the lecturer) inspects the Artefacts of each player and counts the number of errors in them. If the project has more Sprints, then the moderator removes all the artefacts implemented on this Sprint from every player, and the procedure starts again; otherwise, the player with the least errors in their product, wins (Fernandes & Sousa, 2010).

Disadvantages
The game does not represent the notion of the Burndown chart.
The game does not show how the Product Owners prioritise the different tasks in the Product Backlog.

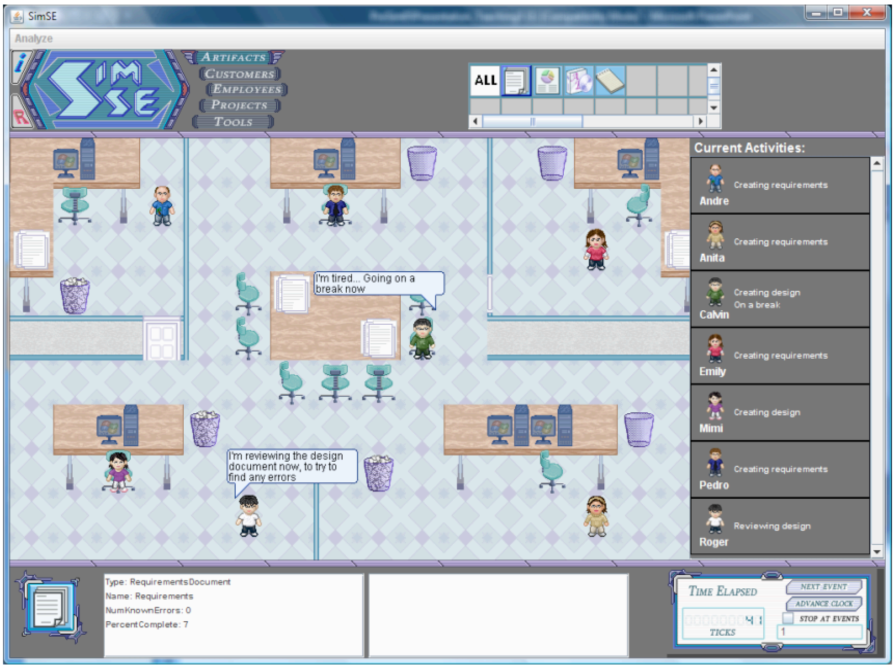
Disadvantages
The idea of Daily Meetings represented by Concept cards does not reflect their real use during the implementation.
Sprint Planning Meeting and Sprint Review Meeting are not covered by the game.
If many people play at the same time, the game takes a long time.
Evaluation showed only an average gain in new knowledge by the participants (3.3 out of 5).
Players with no prior knowledge of Scrum found the game difficult to play.

Table 2.3: PlayScrum disadvantages

- **SimSE**

SimSE is an educational game used at university level, which simulates the application of the Waterfall model within a project. The creators state, "the goal of SimSE is to bridge the gap between the large amount of conceptual software process knowledge given to students in lectures and the comparably small amount of this they actually get to put into practice in the typical class software engineering project" (Navarro & Van der Hoek, 2009). The system is a single player game in which users play the role of Project Managers. They are responsible for a virtual team of developers, monitoring the progress of that team, hiring/firing staff, and making sure that the development team always has the appropriate implementation tools they need (Oh Navarro & Van der Hoek, 2004).

An important feature of this system is that it is graphical and players are able to fully customise the game through the model builder tool (Figure 2.6). The game has many different projects of various size and tasks, allowing the user to customise the model and either create multiple games, or customise them so it focuses on the particular issues of software engineering that players are interested in. This simulation engine consists of two other components, the simulation model that defines all the rules that are necessary to drive the simulation with particular project profiles, and the graphical user interface through which players interact with the system (Oh & Van der Hoek, 2002).



I'm tired... Going on a break now

I'm reviewing the design document now, to try to find any errors

Andre: Creating requirements
Anita: Creating requirements
Calvin: Creating design, On a break
Emily: Creating requirements
Mimi: Creating design
Pedro: Creating requirements
Roger: Reviewing design

Type: RequirementsDocument
Name: Requirements
NumKnownErrors: 0
PercentComplete: 7

Figure 2.6: SimSE interface (Navarro & Van der Hoek, 2009)

The game takes place in an office environment, where the team members communicate with their Project Manager through pop-up messages notifying them about their progress or problems. The player then interacts with their team members through a drop down menu that appears when clicking on a particular employee and selects any of the available actions, e.g. do coding, give a raise, etc. (Oh Navarro & Van der Hoek, 2004). At the end of the game the player receives a score, which corresponds to their performance and how well they managed and advised their team, along with an explanation of why they were given the score they received (Navarro & Van der Hoek, 2005).

Disadvantages
Prior knowledge is assumed.
Lack of guidelines on how to play the game.

Table 2.4: SimSE disadvantages

- **SESAM (Software Engineering Simulation by Animated Models)**

SESAM is an educational game that uses simulation to teach some basic software engineering principles and tasks. The game simulates a project that in real life would last two months to several years, within two hours. Players interact with the system through a text interface, where they insert in natural language the tasks that they want their employees to execute. The virtual team communicates with their manager in the same way. The player’s goal is to successfully complete the project, managing their teams with the optimal way.

At the end of the game, players can see the score they achieved. This score is formed by some variables hidden from the user during the game. When viewing the score, players can analyse their performance using an analysis tool, which shows their progress graphically and the way that these variables changed (Drappa & Ludewig, 2000).

Disadvantages
Prior knowledge was assumed.
Users did not use the analysis tool to analyse their results.
4 out of 9 participants did not improve their performance.

Table 2.5: SESAM disadvantages

- **Problems & Programmers**

Problems and Programmers is an educational software engineering card game for two players targeted at university students, which teaches the principles of the Waterfall model. Players assume the role of Project Manager whose aim is to complete their project as quickly as possible and with the fewest errors in their code.

At the start, both players are given the same project profile, which specifies the budget, its expected quality and complexity, and its duration. The game contains 3 kinds of cards: Concept cards that represent player’s decisions for their project, Programmer cards, which represent the players’ employees, and Problem cards, which are cards that the players’ opponent can play against them that cause problems to their design and slow them down. A player’s opponent can play a Problem card against them if they meet its criteria. They then pick 5 cards from the deck and decide what actions to take, depending on the phase of the Waterfall lifecycle they are in. Players align their cards in different columns from left to right representing the different stages of the Waterfall (Figure 2.7) (Baker et al., 2005).

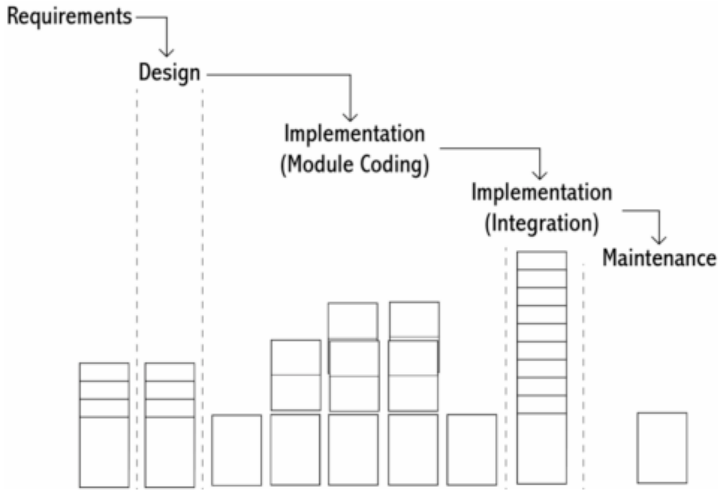


Figure 2.7: Phases of Problems & Programmers game (Baker et al., 2005)

Players are not required to start strictly from the Requirements phase, but if they do not, they will probably encounter problems concerning the progress of their project. At their turn, players draw documentation cards, some of which are marked as “unclear”, to introduce the notion of error or problem with the task under implementation. The players’ programmer cards are put in a row next to the design cards and on each turn that the player works on the implementation, their employees can either create quickly low quality code, or methodically good code, or inspect the existing code, or detect errors.

As soon as the implementation is complete, the player merges all their code cards in a pile, mixes them, and then their opponent cuts the pile in half and selects cards off of the top of the pile equal to the quality of the product indicated in the project profile. The player wins who finishes first and with the minimum numbers of errors (Navarro et al., 2004).

Disadvantages
Evaluation showed the game was moderately (3 out of 5) successful in reinforcing software engineering process.
Evaluation showed that players found the requirements and the design phases boring.
Players felt that the learning curve was too steep.
The game was rated 2.6 out of 5 on teaching new software engineering concepts.
The game received more positive comments from people that had previous work experience.

Table 2.6: Problems & Programmers disadvantages

- **Scrum Master Game**

The Scrum Master Game tries to teach Scrum methodology and its principles through simulation. The system is web-based; it was implemented on the Facebook social platform. It is a game for one player in the role of the Scrum Master, who interacts with their virtual team and Product Owner (Figure 2.8).

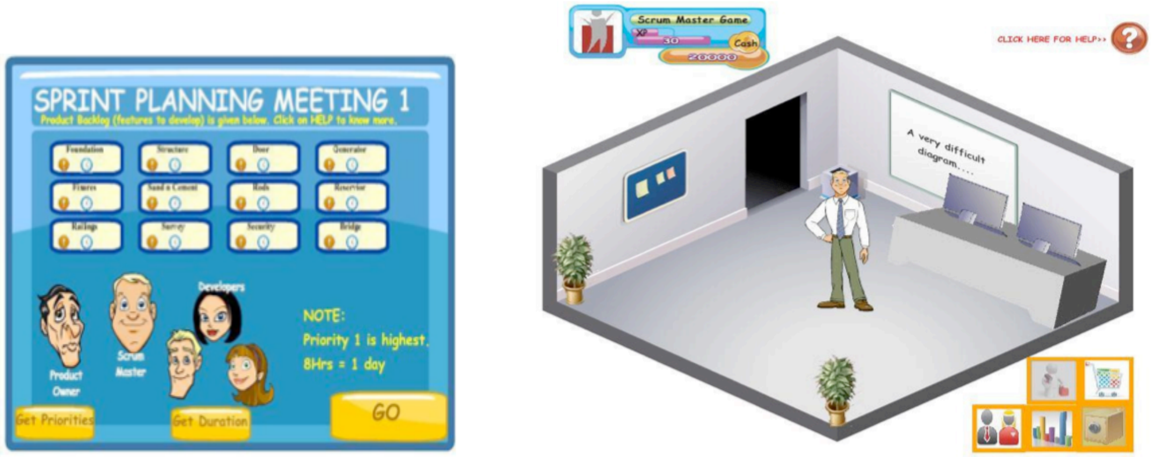


Figure 2.8: Scrum Master screenshots (Panda, 2010; Priyadarshee, 2010)

The player chooses their character, and selects the project they would like to implement. They then pick their team according to skills. The Scrum lifecycle then starts and the player is required to allocate the tasks to be implemented in each Sprint. During the Sprint, the Scrum Master can call for Daily Scrum Meetings as often as they want and ask their employees the 3 questions: what they have done up to that point, what they will do next, and what problems they encountered. Employees reply to these questions and the player has to select the best answer to each problem. At the end of the Sprint, the player calls for a Review Meeting where the Product Owner evaluates the final product. During the game, users are able to view the Burndown chart of their teams' overall progress, and with every task they can either lose or gain experience and cash (Panda, 2010; Priyadarshee, 2010).

Disadvantages
Evaluation showed that 56% of the participants gained new knowledge.
Players felt that there were a lack of guidelines.
The project specifications were not clear.
The Burndown chart does not work.
Players could not understand when a Sprint finishes and another one starts.
Users were not clear at which phase of Scrum they were at any time.
Prior knowledge was assumed.

Table 2.7: Scrum Master Game disadvantages

2.5.2 Comparison of Scrum Game with existing games

Table 2.8 compares the features of the games exposed in the previous section, with the Scrum Game. Even though all of them have the same purpose, the way that they try to accomplish their goal varies. There are games that focus on agile methods (Agile Hour, PlayScrum, Scrum Master Game, Scrum Game), on traditional project management methods, especially the Waterfall model (Problems & Programmers, Software Management Game and SimSE), and on general software engineering issues and decisions. These last cover, for example, what tools to use, or how many people are needed to implement a task, or how to deal with problems during implementation (Software Management Game, SESAM, PlayScrum, Scrum Game). Four out of the eight games are computer-based, so can be played at any time and anywhere, without having to be engaged with them for a long time. An example of this is Agile Hour, where the game lasts for 70 intensive minutes. Finally, several of the games require some prior knowledge of the subject that they cover and, for this reason, players did not perform as well as they expected as occasionally they did not know the different concepts that were presented to them, or how to deal with certain situations.

Game Type	Software Management Game	Agile Hour	PlayScrum	SimSE	SESAM	Problems & Programmers	Scrum Master Game	Scrum Game
Educational game	✓	✓	✓	✓	✓	✓	✓	✓

Game Type	Software Management Game	Agile Hour	PlayScrum	SimSE	SESAM	Problems & Programmers	Scrum Master Game	Scrum Game
Stand-alone system				✓	✓		✓	✓
Traditional methods	✓			✓		✓		
Agile methods		✓	✓				✓	✓
General management decisions	✓		✓		✓			✓
Computer-based game				✓	✓		✓	✓
Interpersonal/ card game	✓	✓	✓			✓		
Requires prior knowledge		✓		✓	✓		✓	

Table 2.8: Game comparison

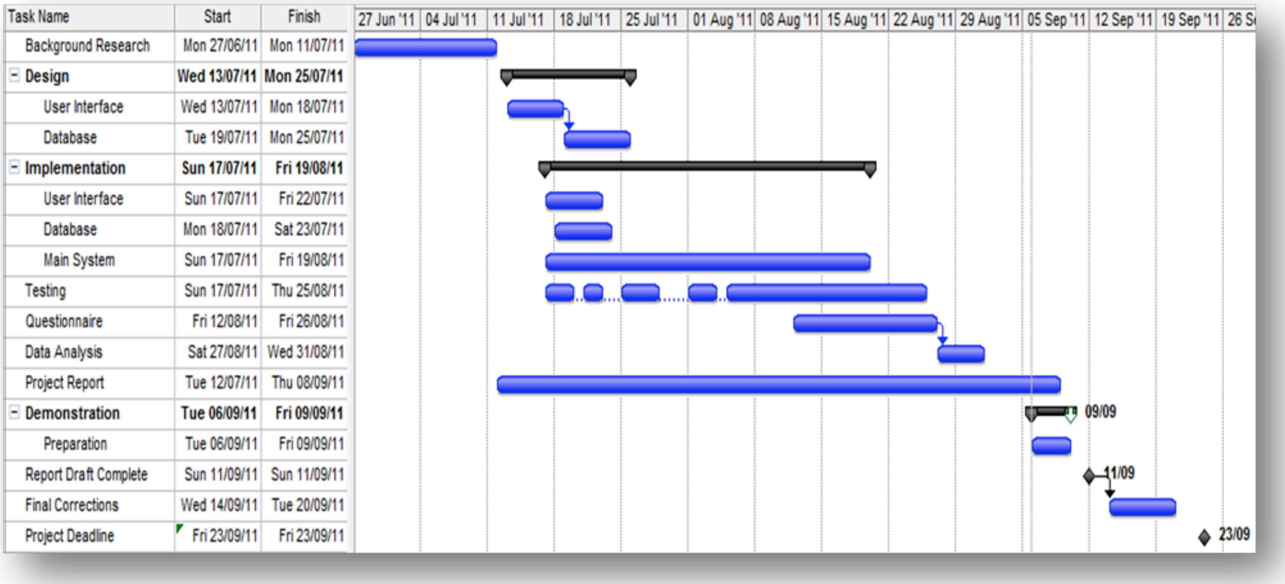
2.5.3 Why Scrum Game?

In Tables 2.1-2.7 show some of the disadvantages of the existing games. This enabled the identification of the weaknesses of current educational games and the creation of a system that would avoid the majority of these problems. Table 2.8 shows that the Scrum Game has many features similar to existing games, in particular with the PlayScrum and Scrum Master Game. All three focus on the Scrum method, but the Scrum Game manages to simulate all the stages and features of the Scrum lifecycle, while clearly showing the application of Scrum in the development of a software project. Further, the Scrum Game users have the opportunity not only to offer advice on technical problems but also to deal with day-to-day problems of their team members (e.g. arguments or illness). This, in combination with the fact that players thought that almost no prior knowledge was assumed (Figure 7.6, helped them learn more about Scrum in an enjoyable and simple way (Figures 7.7 & 7.10). Finally, the Scrum Game can be fully customisable using its Administrator mode, so the game moderator can create their own project or adjust the game to fit their needs.

3 Project Management

3.1 Project planning

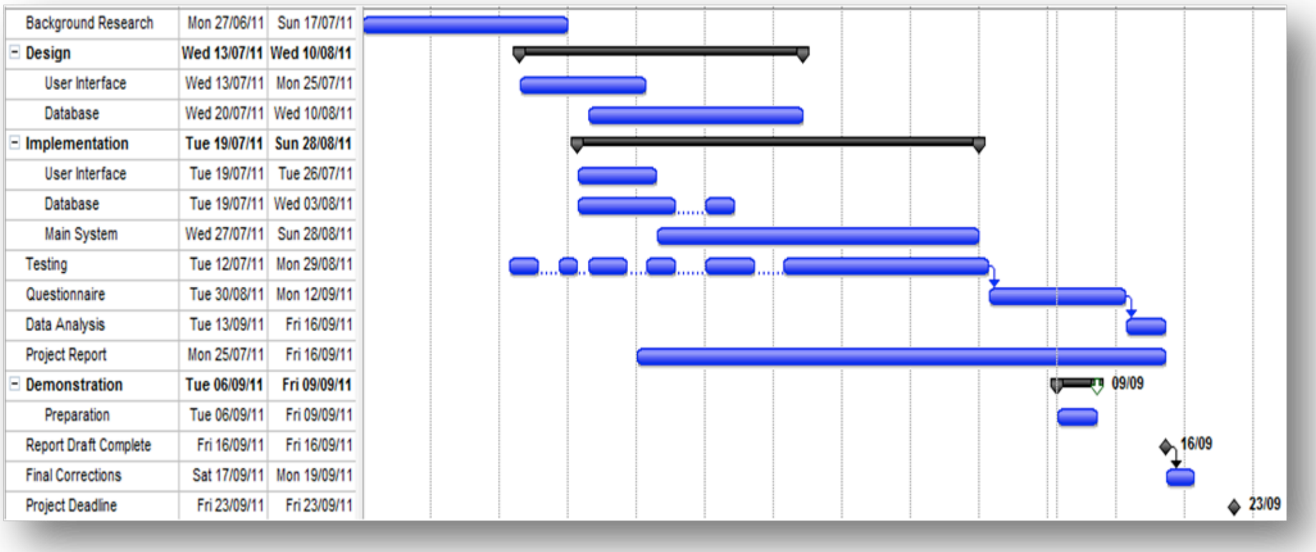
Project planning is a very important stage of project development because developers and managers use it as a guide to track their overall progress, to know what tasks to perform next, to estimate the amount of work that still needs to be completed and finally to keep the project within specified time limits. This project followed the principles of XP programming in terms of planning, design and implementation, which means that, even though a plan was created, more focus was put in planning rather than the plan itself. More specifically, as Cohn characteristically states "agile planning balances the effort and investment in planning with the knowledge that we will revise the plan through the course of the project. An agile plan is one that we are not only willing, but also eager to change" (Cohn, 2007). When using agile methods, this means that change is expected. Therefore an agile project plan, is one that is easily altered when, for example, a change in requirements occurs. Figure 3.1 shows the initial Gantt chart for this project.



Task Name	Start	Finish
Background Research	Mon 27/06/11	Mon 11/07/11
− **Design**	**Wed 13/07/11**	**Mon 25/07/11**
User Interface	Wed 13/07/11	Mon 18/07/11
Database	Tue 19/07/11	Mon 25/07/11
− **Implementation**	**Sun 17/07/11**	**Fri 19/08/11**
User Interface	Sun 17/07/11	Fri 22/07/11
Database	Mon 18/07/11	Sat 23/07/11
Main System	Sun 17/07/11	Fri 19/08/11
Testing	Sun 17/07/11	Thu 25/08/11
Questionnaire	Fri 12/08/11	Fri 26/08/11
Data Analysis	Sat 27/08/11	Wed 31/08/11
Project Report	Tue 12/07/11	Thu 08/09/11
− **Demonstration**	**Tue 06/09/11**	**Fri 09/09/11**
Preparation	Tue 06/09/11	Fri 09/09/11
Report Draft Complete	Sun 11/09/11	Sun 11/09/11
Final Corrections	Wed 14/09/11	Tue 20/09/11
Project Deadline	Fri 23/09/11	Fri 23/09/11

Figure 3.1: Initial Gantt chart

The above Gantt chart was constructed at the beginning of the project, containing only draft estimates of the duration of the different tasks, because not all the requirements of the system were finalised. As soon as the detailed features and requirements were specified and the development process started, an updated Gantt chart was constructed (Figure 3.2).



Background Research	Mon 27/06/11	Sun 17/07/11
⊟ **Design**	**Wed 13/07/11**	**Wed 10/08/11**
User Interface	Wed 13/07/11	Mon 25/07/11
Database	Wed 20/07/11	Wed 10/08/11
⊟ **Implementation**	**Tue 19/07/11**	**Sun 28/08/11**
User Interface	Tue 19/07/11	Tue 26/07/11
Database	Tue 19/07/11	Wed 03/08/11
Main System	Wed 27/07/11	Sun 28/08/11
Testing	Tue 12/07/11	Mon 29/08/11
Questionnaire	Tue 30/08/11	Mon 12/09/11
Data Analysis	Tue 13/09/11	Fri 16/09/11
Project Report	Mon 25/07/11	Fri 16/09/11
⊟ **Demonstration**	**Tue 06/09/11**	**Fri 09/09/11**
Preparation	Tue 06/09/11	Fri 09/09/11
Report Draft Complete	Fri 16/09/11	Fri 16/09/11
Final Corrections	Sat 17/09/11	Mon 19/09/11
Project Deadline	Fri 23/09/11	Fri 23/09/11

Figure 3.2: Final Gantt chart

Figure 3.2 shows that not everything went as planned, but, because the plan was agile, it was easy to incorporate the new project needs in the time available without affecting the final quality or the delivery date of the project.

3.2 Risk Analysis

A project can be affected by many different factors, both expected and unexpected, that may disrupt its development and lead to its failure (Hillson, 2009). These events can either arise from internal sources, e.g. illness or leave of a team member, or external, e.g. hardware failure, change of requirements, lack of funding. Regardless their nature, they can affect the quality, the delivery and the overall cost of the project. Because events like these usually occur unexpectedly, a risk analysis is necessary to identify the possible risks, to estimate the probability of their occurrence and their impact on progress, and to identify ways to avoid them or at least minimise their effect.

Even though risk analysis is usually performed for large and long-lasting projects, it was considered appropriate to perform a risk analysis for this project (Table 3.1), because a high quality outcome had to be achieved in a short time.

		1=Very Unlikely; 2=Unlikely; 3=Moderate; 4=Likely	
Risk	**Impact**	**Probability**	**Strategy to minimise impact**
Illness	Remaining work may be delayed.	3	Adjust the incomplete tasks within the time available.
Database failure	System unable to access information stored in the database.	1	Keep an updated local copy of the database and inform users of the situation.
Computer failure	Part of the work may be lost.	2	Daily backups and save work into online repositories.
Miss deadline	Late project hand in and loss of marks.	2	Remain on schedule or reschedule tasks in the time available.
Missing supervisor meetings	Lose the focus and unable to evaluate progress.	2	Inform Personal Tutor and get advice from them or the second examiner.
ECS filestore failure	Users will not be able to access the system.	3	Keep local copies of the system.
Lack of players	Students may not volunteer evaluate the system.	3	Change the target group of the system and find volunteers to use it.
Insecure system	System will be vulnerable to external attacks.	3	Implement security procedures.
Coding restrictions	Unable to complete desired features.	3	Find alternate ways to implement the feature; ask colleagues to help; advice from online sources.

Table 3.1: Risk Analysis (modified from Gkritsi, 2010)

3.3 Project lifecycle

This project has been developed using the basic principles of XP methodology (Figure 2.3) but with some minor changes. XP was considered the most appropriate method to develop this project because of its flexible and iterative nature. More specifically, due to the short duration of the project, XP lifecycle was ideal because it focuses on the implementation of the final product, rather than the design and documentation. This allowed the user stories to be quickly identified and the implementation phase to be started at the earliest. Moreover, the project followed the XP pattern of small and frequent releases through the weekly meetings, with the project supervisor acting as a stakeholder, evaluating the work that had already been done, and identifying any changes or improvements that were thought necessary. In addition, some of the requirements changed during the implementation phase, therefore the code had to be reviewed and the database to be refactored to accommodate the extra requirements. These new requirements did not affect the progress or the final delivery date, because XP allows the implementation of flexible systems and accommodates any possible changes by delivering pieces of the system in small and simple increments.

The short duration of the project was an important factor that led to the use of XP. This is because change is expected. Therefore, any change in the requirements occurring, or an aspect of the project not operating as expected, results in easy rescheduling of the remaining tasks, aiming for the best outcome within the remaining time.

4 Design

Before proceeding to the development of the final system, it was essential to design it, taking into consideration the system requirements, time limits, and the system and project goals (see sections 4.3 and 4.4).

4.1 System

To enter the Scrum Game, users need to have access to the internet, as it is web based. For the development of the system, the PHP scripting language was adopted because it is widely used, there are plenty of advanced libraries available and is easy to use. PHP is a server-side language, so the users' browser cannot directly display PHP files. Therefore, the web page first needs to access the server where the files are stored, and the latter will then display HTML format. This way the code is more secure compared with other scripting languages such as JavaScript, where the full code is visible to the user (Gkritsi, 2010). All the information necessary for the system is retrieved from the database through queries in the PHP files (Figure *4.1*).

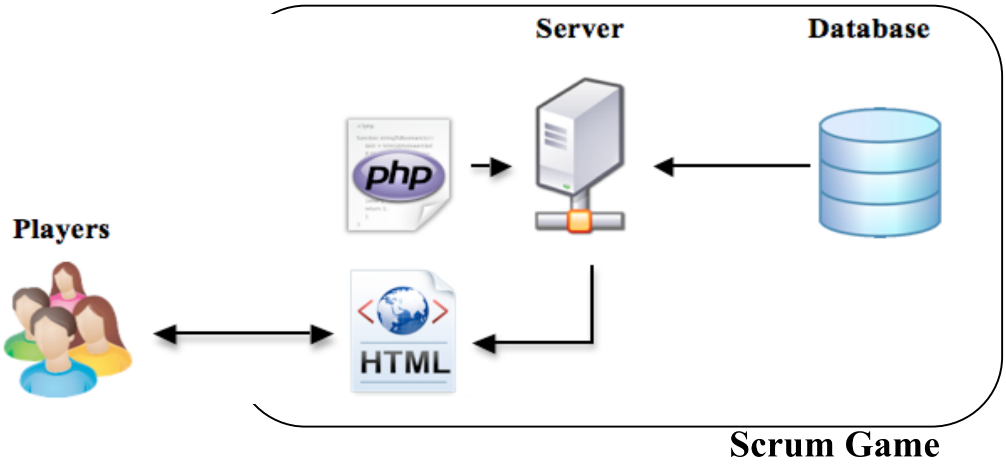


Figure 4.1: **System diagram**

To keep the system and the code concise, simple and manageable, it was broken into different files. It was thus possible to minimise coupling, make sure that each file had a specific purpose, and each component would interact with others optimally. Table 4.1 gives a brief description of the system's PHP files and their contents.

File	Description
index.php	Start page of the game.
authentication.php	Checks if players are registered or are administrators.
registration.php	Registers new players.
connect.php	Connects the system to the database.
change.php	When players forget their password, it replaces their old one with a new one.
welcome.php	Contains the rules of the game and allows users to pick the project to implement.
product_backlog.php	Displays the product backlog of the selected project and users estimate their duration and effort to implement.
sprint_backlog.php	Provides the sprint backlog of the selected project and users assign the different tasks to specific Sprints.
team.php	Shows the different employees available.
sprint.php	Displays the 3 questions of the Daily Scrum Meeting and players select one of the 3 available options to solve their employees' problems.
graphs.php	Displays the Burndown chart.
review_meeting.php	Checks if the quality of the system is above or below the required percentage. If yes, player proceeds to next Sprint. If no, repeat Sprint.
end.php	Summary of the overall progress of the player in the game. Contains their history and a Top 5 table.
admin.php	The main page of the administrator mode, where lecturers can add or preview their submissions.
log.php	Displays the history of all players in the administrator mode.
help.php	Provides additional information about Scrum.
methods.php	Contains commonly used system functions, to avoid coupling.

Table 4.1: PHP files description

4.2 Game

The primary aim of the Scrum Game was to closely simulate the use of the Scrum method in different projects. As mentioned in Chapter 1, a Sprint may last from 15 to 30 calendar days. Also, a real team consists ideally or more than 7 people, and several meetings take place lasting up to 8 hours each. It can be understood that it would not be appropriate to construct a game that would last this long or involve so many people, as players can lose their motivation or interest in the game during that period.

Consequently, several design decisions had to be taken concerning the structure and timescale of the game. These are detailed below.

1. **Sprint duration**: In the game, the duration of the Sprint was set to be 2 days because this value enables easier representation of the Daily Scrum Meetings, and it would not be tiring for the player.
2. **Duration of a day**: A day in the Scrum Game starts with the Daily Scrum Meeting, where the Scrum Master meets with their team to discuss any problems. As soon as the meeting is over, a progress bar appears for 2 seconds with the message "Executing tasks…" to simulate the team going back to their tasks. The end of that day comes when players get directed to either the next day or to a Review Meeting.
3. **Product Owner**: The administrator plays the role of the Product Owner in this game. At the moment, the different projects, product backlog tasks, answers of the employees and advice options for the Scrum Master during the Daily Scrum Meetings, are inserted by the administrator using the admin.php page.
4. **Scrum Master**: The player plays the role of Scrum Master in this game. Players are responsible for managing a virtual team, monitoring the progress of their team through the Burndown chart, and giving them advice on problems that occur during the Daily Scrum Meetings.
5. **Team**: Real Scrum teams normally comprise 5 to 9 people. For the purposes of this game, it was decided to use teams of 3. This was considered ideal since it is small, and the player will not get bored giving advice to so few, while it is big enough to represent a team. A team of 2 was rejected on the grounds of familiarity with the Pair Programming technique of XP.
6. **Daily Scrum Meeting**: A real Daily Scrum Meeting takes place usually at the beginning of every day and lasts about 15 minutes. The Scrum Master asks each team member three questions about their progress and any problems during implementation (see section 0). To represent it accurately, the system retrieves from the database all the employee answers and the problems that might occur during that specific project, as well as three different advices that the Scrum Master can give to their team (Figure 5.7).
7. **Release Planning Meeting**: In a real meeting, the Scrum Master and team meet with the Product Owner to discuss the prioritised product backlog, and to specify the effort and duration of the different tasks. In the game, the administrator has already inserted the different product backlog tasks, so the

Scrum Master is required to select the effort and the duration (in days) of each task.

8. **Review Meeting**: At the Review Meeting, the team delivers to the Product Owner what has been developed during the previous Sprint, so it can be evaluated. In this game, the evaluation is done by the system itself using the quality of the product as the criterion. The quality of the product depends on the advice that the player gives to their team. If the quality of the project is less than 50%, then players have to redo the previous Sprint, otherwise they can proceed to the next stage of the game.

4.3 Activity Diagrams

The aim of the design was to implement a web-based prototype of an agile project management system in the form of a game, which simulates all the steps of the Scrum lifecycle through various types of projects. In addition, the system needed to have an administrator mode where tutors and lecturers would be able to alter the existing projects or create their own to fit their syllabus. Figure 4.2 represents the activity diagram of the game.

To enter the game, new players first need to register. Returning players have to enter their username and password and will then be returned to the point where they last quit the game. In case they have forgotten their password, players have the option to reset it and then login using their new password. As soon as new users register and login, they can choose the project they would like to implement, and their team. After that, they are required to estimate the effort and the duration of the different tasks in the product backlog. They are then prompted to construct the Sprint backlog in order to proceed to the first Sprint.

During each Sprint, two Daily Scrum Meetings take place. At this stage, the system presents the player with three questions, and the player provides solutions to these questions. This procedure is repeated for the second Daily Scrum Meeting. The player continues to the Review Meeting. Here, the system checks the score (i.e. the quality of the product) of the player. If it is above 50%, players proceed to the next Sprint; otherwise, they repeat the same Sprint until they reach the desired product quality. If the project has more than one Sprint, the procedure is repeated for all the Sprints of the project. Players have completed their game when they deliver a product that meets the Product Owner's quality expectations.

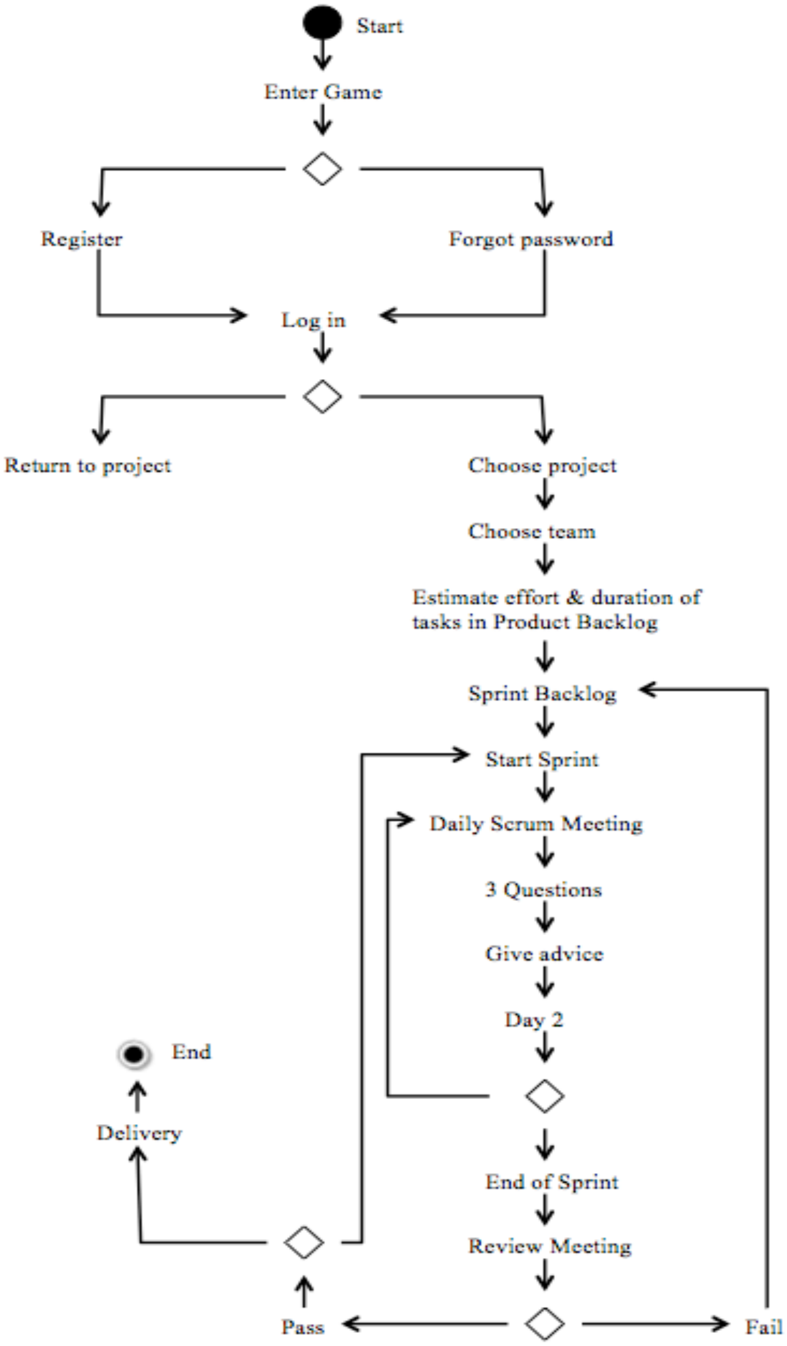


Figure 4.2: Game mode activity diagram

As far as the administrator mode is concerned, lecturers can choose to view a complete log with the progress of all the players within the system, or edit existing projects of the game, or create new projects. For example, when they want to add a project, they insert the information they require using the form provided, which can be previewed before they submit it. After that, they return to the administrator page and they can either alter or proceed with their submission (Figure 4.3).

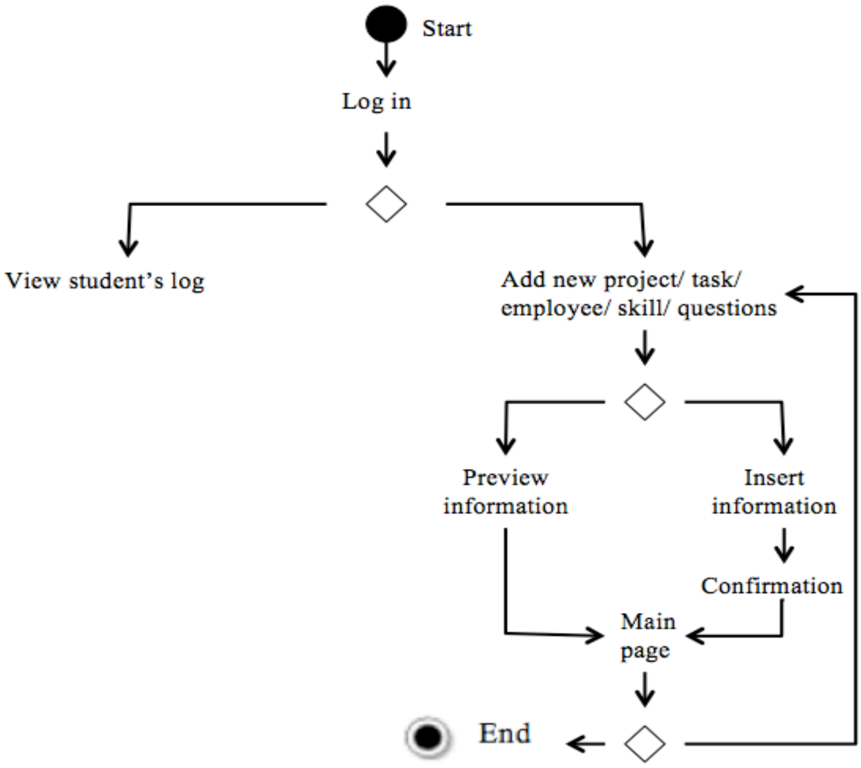


Figure 4.3: Admin mode flow chart

4.4 System Goals

1. Create a prototype of a game that simulates using Scrum for every step that a project has to undergo from requirements to delivery.
2. The system should have two different modes: game and administrator.
3. Create a game where not much prior knowledge on the Scrum methodology is assumed, targeted at students with some IT or Project Management background.
4. Players should have the role of Scrum Master and they will be responsible for managing a team, helping them to estimate the duration and assign the different tasks to the appropriate Sprints.
5. Players should give their team advice on the problems that occur during the Daily Scrum Meetings.
6. Each advice should be credited with a score, which affects the quality of the final product.

7. During each Sprint, players should have a Burndown chart to monitor their progress within the game.
8. At the end of the game, the system must represent the score of the Top 5 players of the game.
9. Administrators should be able to monitor the progress of every player.
10. Administrators should be able to create their own projects, adding new employees, new employee skills, project tasks and Daily Scrum questions and advice.

4.5 Project Goals

1. Meet all the deadlines of the project.
2. Implement a project that conforms to the initial design.
3. Meet all the system goals.
4. Create a fully functional prototype of the Scrum Game and its administrator mode.
5. Implement a game that helps students understand in more depth the Scrum lifecycle.

4.6 Database

Table 4.1 shows the database tables and a brief description concerning the information that is stored in each of them, and Figure 4.4 represents the database model.

Table Name	Description
Player	Contains information about each player.
Project	Information about the different projects a player can choose.
Employees	Holds data for the employees that users pick for their teams.
Skills	Contains the different skills that the above employees have.
Player_has_Employees	Contains the employees that player picked during their game.
Tasks	Stores the tasks of every project.
Player_has_Tasks	Holds the tasks of each player during the game.
Daily_Question	Contains all the questions of the Daily Scrum Meeting.
Daily_Options	Stores the different options of the above questions.
Quality	The score of quality that corresponds to each option.
Player_has_Quality	The score that corresponds to players depending their selections.
Log	Stores a history of the scores and projects that players selected.

Table 4.2: Database tables description

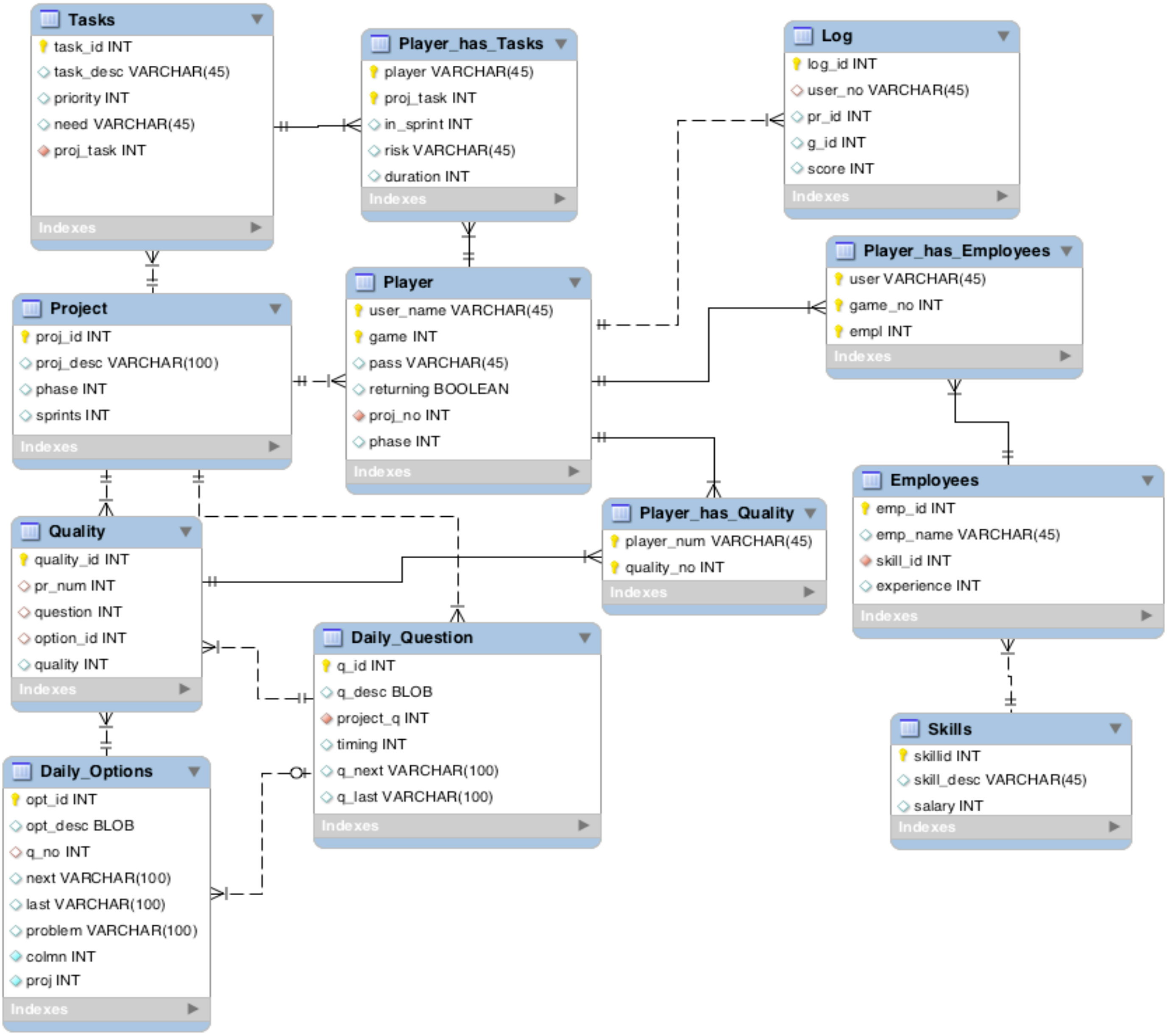


Figure 4.4: Database Model

5 Implementation

5.1 Development languages

The Scrum Game was implemented using many different development languages. It was initially thought that the main system could be built using Java SE, as the author was very familiar with it, its documentation, libraries and syntax. Java SE would lead to the creation of a system that was cross-platform, robust and interactive, and could attract users' attention. However, it was realised during the design stage that the use of Java SE would limit the number of possible users, because students do not have administrative rights on the computers in the School of Electronics and Computer Science. They would not be able to install the system and play it at any time. To overcome this problem, it was decided to make the system web-based.

To implement a web-based system there are many different languages, such as PHP and ASP.NET. A small research study was undertaken to investigate their advantages and disadvantages, to decide which of these languages would be more appropriate for this project.

PHP	
Advantages	**Disadvantages**
Many libraries and frameworks	Few formal training courses
Recommended for small systems	Poor separation of rules
Open source	Low scalability
Cross platform	Limited handling of exceptions
Many books and online communities	
Support for objects and modularity	

Table 5.1: PHP advantages and disadvantages (Gkritsi, 2010)[5]

ASP.NET	
Advantages	**Disadvantages**
Good separation of roles	Complex model as it progresses
Visual development environment	Not recommended for small systems
Good support and training opportunities	Not recommended for non-Windows platforms
Many libraries and frameworks	
Scalability	

Table 5.2: ASP.NET advantages and disadvantages (Gkritsi, 2010)[6]

[5] Available at: https://secure.ecs.soton.ac.uk/notes/comp3018/

[6] Available at: https://secure.ecs.soton.ac.uk/notes/comp3018/

The study showed that both scripting languages have many features that fit in the requirements of the system. However, some of the characteristics of ASP.NET led to the conclusion that it would be better to proceed with PHP. First, the author had no prior experience with ASP.NET and, because of the limited time available for this project, valuable time would be lost in learning its syntax. As a result, some of the desired features of the system would have had to be dropped. Further, the author would have had to use a virtual machine to implement the system under Windows, which would have been time consuming. In addition, ASP.NET is appropriate for large-scale systems, while the Scrum Game is a small prototype of a complete system. The author has had some experience with PHP.

The back-end database of the Scrum Game is a MySQL database. No other technologies where considered (Microsoft Access or Oracle), as the author had previous experience using MySQL, it is supported from the School of Electronics and Computer Science's database, and it interacts very well with PHP. Other implementation languages were used to make the user interface more interactive. These were: HTML, CSS, AJAX and JavaScript. The first two were necessary to modify the structure and the way that the system will be presented by the different browsers, while the latter two were used to add more functionality to the system, and make it user friendly and appealing. Finally, during the implementation phase it was necessary to refer to some online resources such as websites[7], e-books and forums[8], as well as lecture notes[9] and books (Brinzarea-Iamandi, 2009; Castro, 2007; McFarland, 2008; Nixon, 2009; Welling, 2009).

5.2 Implementation tools

During the implementation of this project, numerous development tools were used to build the database, write the project report, develop the system, and keep regular and updated back-up of the overall progress.

Microsoft Project was used to create a draft Gantt chart, as well as a final one, to represent how the project progressed. This program was chosen over Excel because it

[7] http://www.w3schools.com/jquery/

[8] Available at: http://forum.jquery.com/

[9] Available at: https://secure.ecs.soton.ac.uk/notes/comp3018/

produces more professional looking and accurate diagrams. This program is used in real projects by some companies; therefore, becoming familiar with it will be a valuable asset for the future.

The database schema was created using MySQL Query Browser and the database table diagram was produced with MySQL Workbench. The database is hosted on the ECS Linuxproj Apache server, which supports MySQL and PHP, and is securely protected behind the ECS firewall. As mentioned in the previous section, the main programming languages used for the implementation of the system were PHP, MySQL, HTML, JavaScript, CSS and AJAX (jQuery). The tools used for the implementation of the code were Dreamweaver initially, and Coda later. These programs where chosen because they allowed connection directly to the Linuxproj server, making it possible to create and update files from there.

The questionnaire evaluating the system was conducted and circulated through iSurvey[10], a survey tool implemented by the School of Psychology of the University of Southampton, available to all the university's students. This tool, apart from gathering user data anonymously, creates charts of different types with the data collected, facilitating data analysis. For the project brief and the final report, Microsoft Word and Pages were used, and Mendeley[11] was used to keep an archive of all the articles and their references. Finally, the different versions of the system were kept under configuration control with TortoiseSVN, and Dropbox was used as a back up.

5.3 System features

Even though it is a prototype, the Scrum Game still includes the features that are necessary to give students a glimpse of how Scrum is applied within different projects. This section describes some of the most important features of the system and the way they were implemented. All the code written for this project is included on a CD ROM. The Scrum Game can be accessed from:

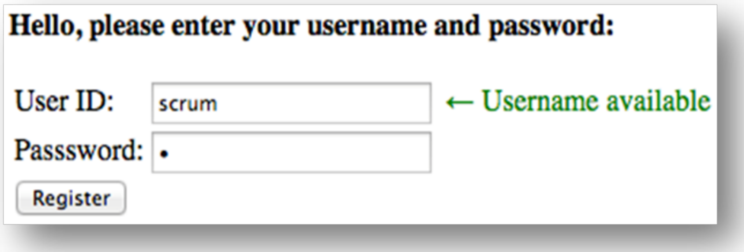
http://users.ecs.soton.ac.uk/ag2006/COMP6029/

[10] Available at: http://www.isurvey.soton.ac.uk/admin/

[11] Available at: http://www.mendeley.com/

5.3.1 Username Availability

When a new player enters the system, before starting their game they need to register first. To do so, they are required to fill in the form below with their username and password (Figure 5.1). While users type their username, the system checks using AJAX and JavaScript whether that username is already in use. Where the username is available, a message appears (Figure 5.1), otherwise another message notifies the user that their selected username is unavailable. This feature avoids users having to insert all their details before being notified that the username they selected is unavailable.



Hello, please enter your username and password:

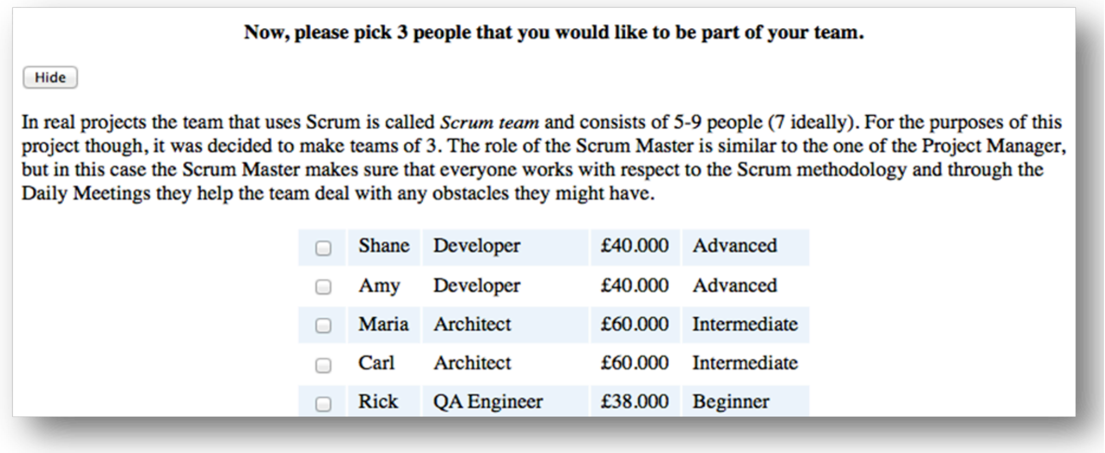
User ID: scrum ← Username available

Passsword: •

Figure 5.1: Username Availability

5.3.2 Team

As soon as players log in the game and select the project that they would like to implement, they are required to assemble their team. Details of all employees shown in Figure 5.2 are stored in the database. During the game, players are given a short explanation on what is happening at that stage and a brief description of all the terms. This display can be hidden whenever players want, by clicking the Hide button. Figure 5.3 shows the code used to implement this feature. Note that the system accesses the database to get the identity (id) of the employee, their name, skill, salary and experience level. With a `for` loop it gets this information for every employee in the database and puts it into a table. Finally, some AJAX was used to validate the number of employees that players select each time.

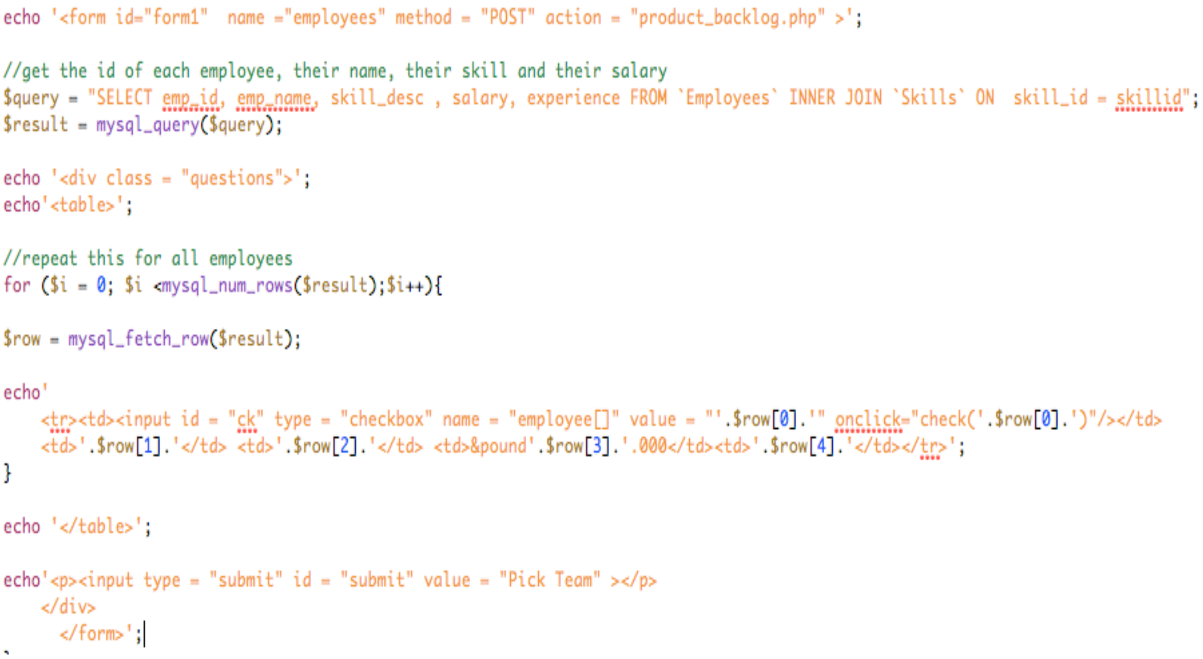


Now, please pick 3 people that you would like to be part of your team.

In real projects the team that uses Scrum is called *Scrum team* and consists of 5-9 people (7 ideally). For the purposes of this project though, it was decided to make teams of 3. The role of the Scrum Master is similar to the one of the Project Manager, but in this case the Scrum Master makes sure that everyone works with respect to the Scrum methodology and through the Daily Meetings they help the team deal with any obstacles they might have.

□	Shane	Developer	£40.000	Advanced
□	Amy	Developer	£40.000	Advanced
□	Maria	Architect	£60.000	Intermediate
□	Carl	Architect	£60.000	Intermediate
□	Rick	QA Engineer	£38.000	Beginner

Figure 5.2: Team

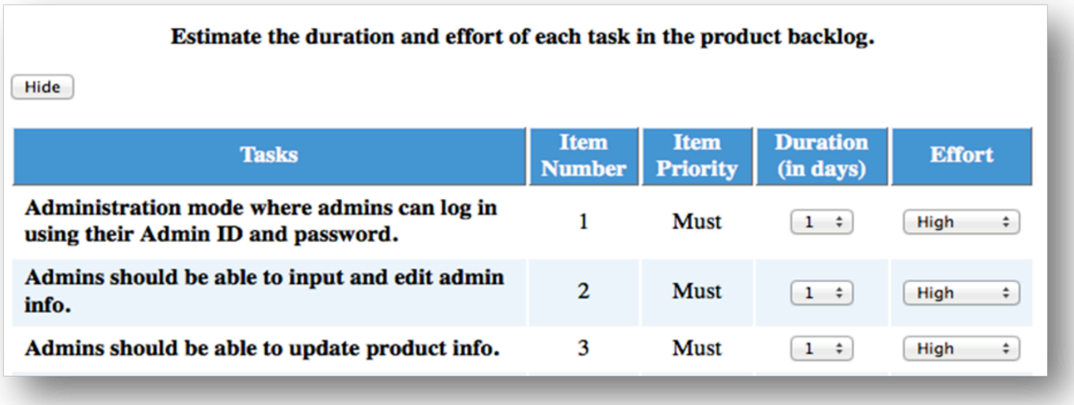


```
echo '<form id="form1"  name ="employees" method = "POST" action = "product_backlog.php" >';

//get the id of each employee, their name, their skill and their salary
$query = "SELECT emp_id, emp_name, skill_desc , salary, experience FROM `Employees` INNER JOIN `Skills` ON  skill_id = skillid";
$result = mysql_query($query);

echo '<div class = "questions">';
echo'<table>';

//repeat this for all employees
for ($i = 0; $i <mysql_num_rows($result);$i++){

$row = mysql_fetch_row($result);

echo'
    <tr><td><input id = "ck" type = "checkbox" name = "employee[]" value = "'.$row[0].'" onclick="check('.$row[0].')"/></td>
    <td>'.$row[1].'</td> <td>'.$row[2].'</td> <td>&pound'.$row[3].'.000</td><td>'.$row[4].'</td></tr>';
}

echo '</table>';

echo'<p><input type = "submit" id = "submit" value = "Pick Team" ></p>
    </div>
      </form>';
```

Figure 5.3: Team code

5.3.3 Product & Sprint Backlog

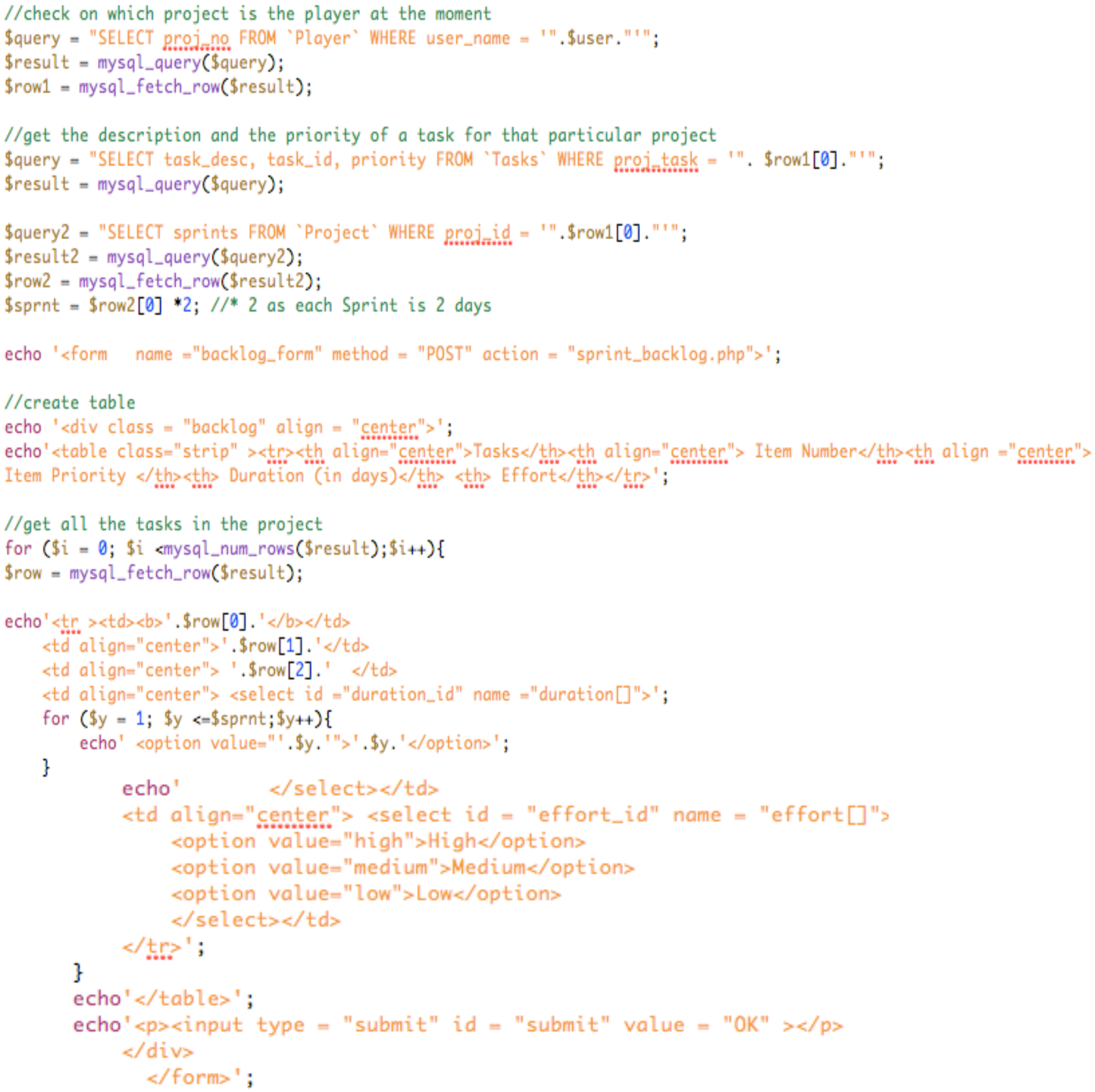
As part of the Release Planning Meeting, players are required to estimate the effort and the duration of each task for the project that they will implement recorded in the product backlog. To create the 5 different columns of the table shown in Figure 5.4, the system first gets the project that player is currently operating, from the **`Player`** table. Then, it requests the id, description and priority of a task for that particular project, as well as its number of Sprints from the **`Project`** table. A **`for`** loop enables the system to get the previous information for each task in that project. The detailed code for this feature is shown in Figure 5.5.



Estimate the duration and effort of each task in the product backlog.

Tasks	Item Number	Item Priority	Duration (in days)	Effort
Administration mode where admins can log in using their Admin ID and password.	1	Must	1	High
Admins should be able to input and edit admin info.	2	Must	1	High
Admins should be able to update product info.	3	Must	1	High

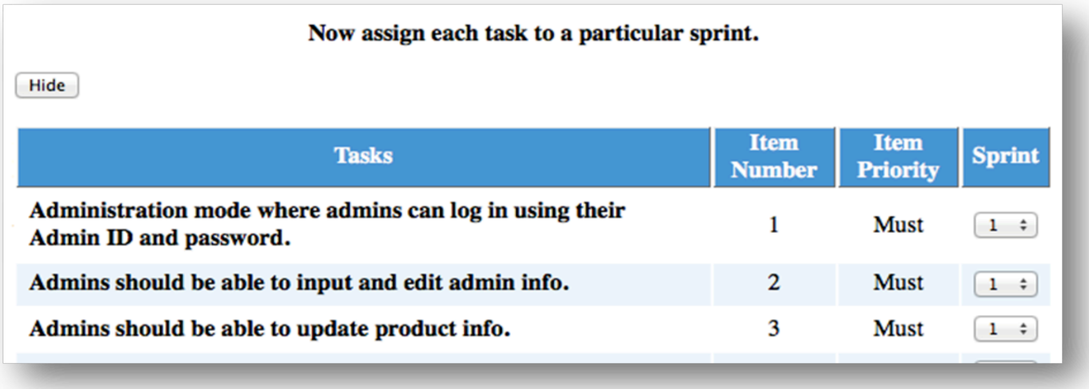
Figure 5.4: Product Backlog



```
//check on which project is the player at the moment
$query = "SELECT proj_no FROM `Player` WHERE user_name = '".$user."'";
$result = mysql_query($query);
$row1 = mysql_fetch_row($result);

//get the description and the priority of a task for that particular project
$query = "SELECT task_desc, task_id, priority FROM `Tasks` WHERE proj_task = '". $row1[0]."'";
$result = mysql_query($query);

$query2 = "SELECT sprints FROM `Project` WHERE proj_id = '".$row1[0]."'";
$result2 = mysql_query($query2);
$row2 = mysql_fetch_row($result2);
$sprnt = $row2[0] *2; //* 2 as each Sprint is 2 days

echo '<form   name ="backlog_form" method = "POST" action = "sprint_backlog.php">';

//create table
echo '<div class = "backlog" align = "center">';
echo'<table class="strip" ><tr><th align="center">Tasks</th><th align="center"> Item Number</th><th align ="center">
Item Priority </th><th> Duration (in days)</th> <th> Effort</th></tr>';

//get all the tasks in the project
for ($i = 0; $i <mysql_num_rows($result);$i++){
$row = mysql_fetch_row($result);

echo'<tr ><td><b>'.$row[0].'</b></td>
    <td align="center">'.$row[1].'</td>
    <td align="center"> '.$row[2].'  </td>
    <td align="center"> <select id ="duration_id" name ="duration[]">';
    for ($y = 1; $y <=$sprnt;$y++){
        echo' <option value="'.$y.'">'.$y.'</option>';
    }
          echo'        </select></td>
          <td align="center"> <select id = "effort_id" name = "effort[]">
              <option value="high">High</option>
              <option value="medium">Medium</option>
              <option value="low">Low</option>
              </select></td>
          </tr>';
      }
      echo'</table>';
      echo'<p><input type = "submit" id = "submit" value = "OK" ></p>
          </div>
            </form>';
```

Figure 5.5: Product Backlog code

A similar approach was adopted for the Sprint backlog shown in Figure 5.6.



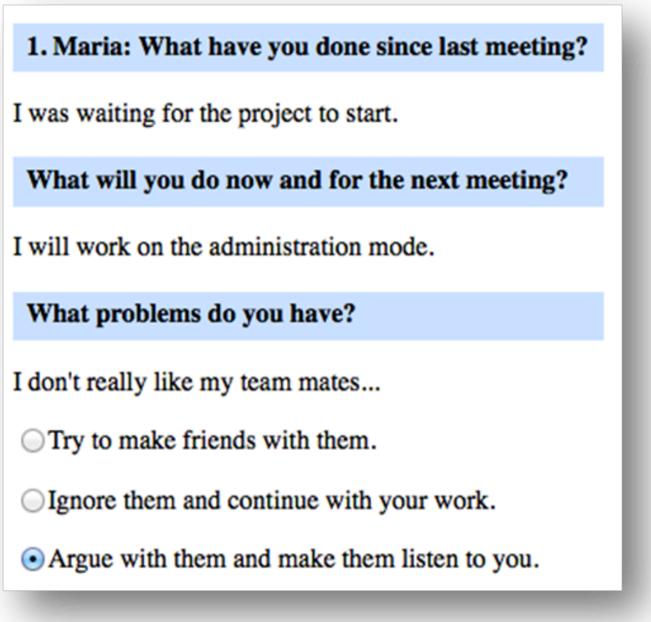
Now assign each task to a particular sprint.

Tasks	Item Number	Item Priority	Sprint
Administration mode where admins can log in using their Admin ID and password.	1	Must	1
Admins should be able to input and edit admin info.	2	Must	1
Admins should be able to update product info.	3	Must	1

Figure 5.6: Sprint Backlog

5.3.4 Daily Scrum Meeting

The Daily Scrum Meeting is one of the most important features of the game. This is the stage where each player interacts with their team, and their decisions will affect the overall quality of the final product. The player has the role of the Scrum Master and asks their team the three questions highlighted in blue in Figure 5.7. To provide all the information that corresponds to each question, the system uses two nested **`for`** loops. The first loop is used to get the different questions, and the other to access the database to print all the answers and options that correspond to these questions.



1. Maria: What have you done since last meeting?

I was waiting for the project to start.

What will you do now and for the next meeting?

I will work on the administration mode.

What problems do you have?

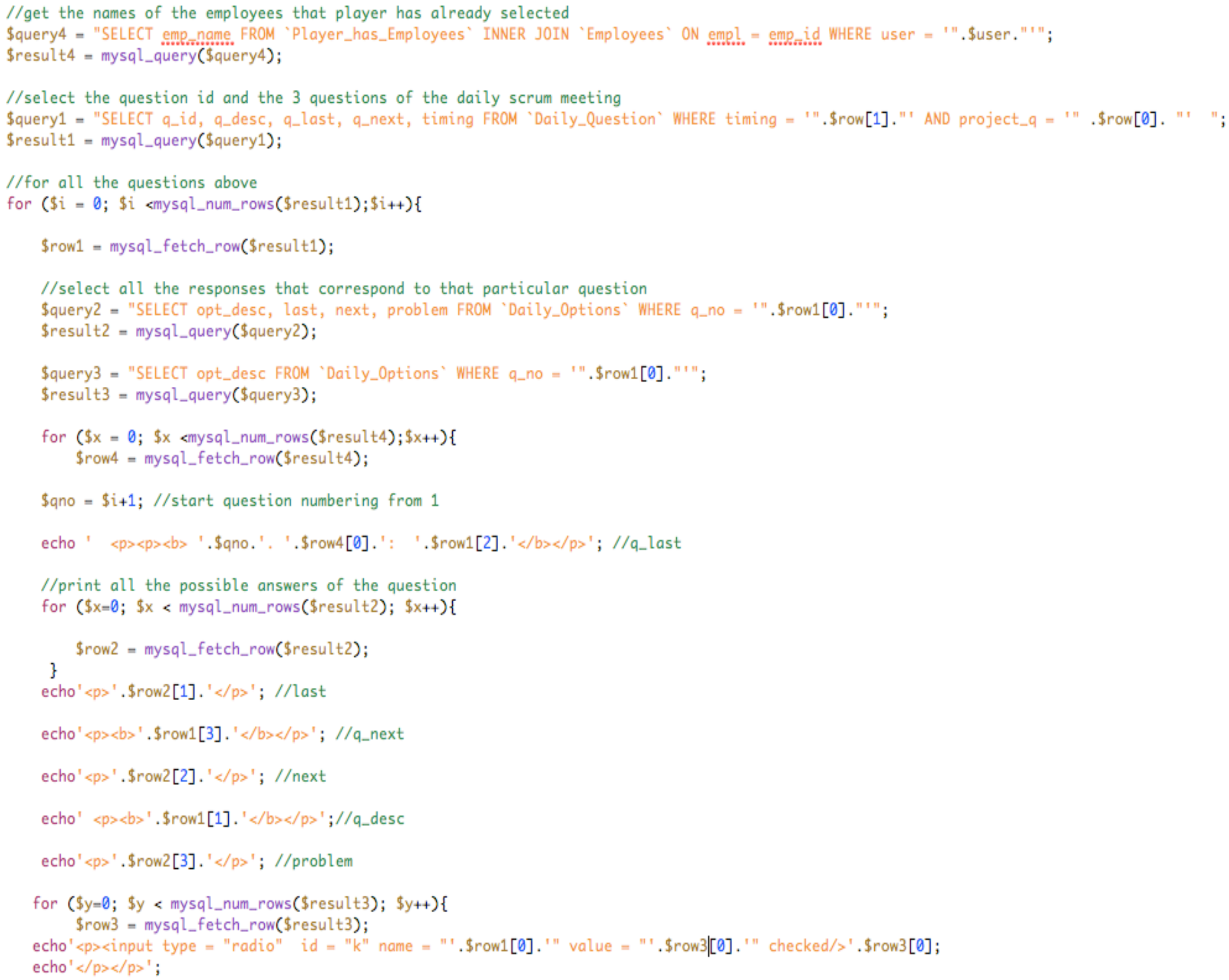
I don't really like my team mates...

○ Try to make friends with them.

○ Ignore them and continue with your work.

● Argue with them and make them listen to you.

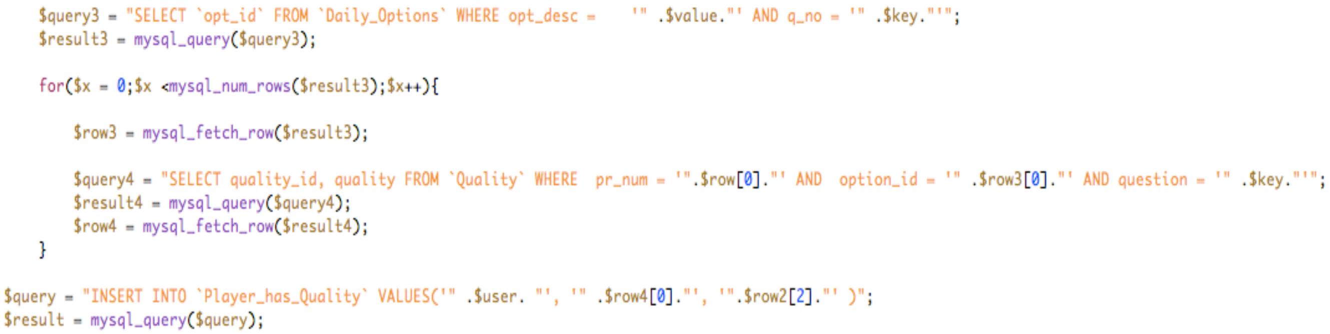
Figure 5.7: Daily Scrum Meeting



```
//get the names of the employees that player has already selected
$query4 = "SELECT emp_name FROM `Player_has_Employees` INNER JOIN `Employees` ON empl = emp_id WHERE user = '".$user."'";
$result4 = mysql_query($query4);

//select the question id and the 3 questions of the daily scrum meeting
$query1 = "SELECT q_id, q_desc, q_last, q_next, timing FROM `Daily_Question` WHERE timing = '".$row[1]."' AND project_q = '" .$row[0]. "'  ";
$result1 = mysql_query($query1);

//for all the questions above
for ($i = 0; $i <mysql_num_rows($result1);$i++){

    $row1 = mysql_fetch_row($result1);

    //select all the responses that correspond to that particular question
    $query2 = "SELECT opt_desc, last, next, problem FROM `Daily_Options` WHERE q_no = '".$row1[0]."'";
    $result2 = mysql_query($query2);

    $query3 = "SELECT opt_desc FROM `Daily_Options` WHERE q_no = '".$row1[0]."'";
    $result3 = mysql_query($query3);

    for ($x = 0; $x <mysql_num_rows($result4);$x++){
        $row4 = mysql_fetch_row($result4);

    $qno = $i+1; //start question numbering from 1

    echo '  <p><p><b> '.$qno.'. '.$row4[0].':  '.$row1[2].'</b></p>'; //q_last

    //print all the possible answers of the question
    for ($x=0; $x < mysql_num_rows($result2); $x++){

        $row2 = mysql_fetch_row($result2);
     }
    echo'<p>'.$row2[1].'</p>'; //last

    echo'<p><b>'.$row1[3].'</b></p>'; //q_next

    echo'<p>'.$row2[2].'</p>'; //next

    echo' <p><b>'.$row1[1].'</b></p>';//q_desc

    echo'<p>'.$row2[3].'</p>'; //problem

   for ($y=0; $y < mysql_num_rows($result3); $y++){
        $row3 = mysql_fetch_row($result3);
   echo'<p><input type = "radio"  id = "k" name = "'.$row1[0].'" value = "'.$row3[0].'" checked/>'.$row3[0];
   echo'</p></p>';
```

Figure 5.8: Daily Scrum Meeting code

All the choices that the player makes during the Daily Scrum Meeting are stored in an array. To add the player's score to the database, all their selections are first retrieved and then these array data are matched against the question number and the id of each selected advice. For each of these selections, using a **`for`** loop, the system gets their quality and inserts the player's score in the **`Player_has_Quality`** table (Figure 5.9).

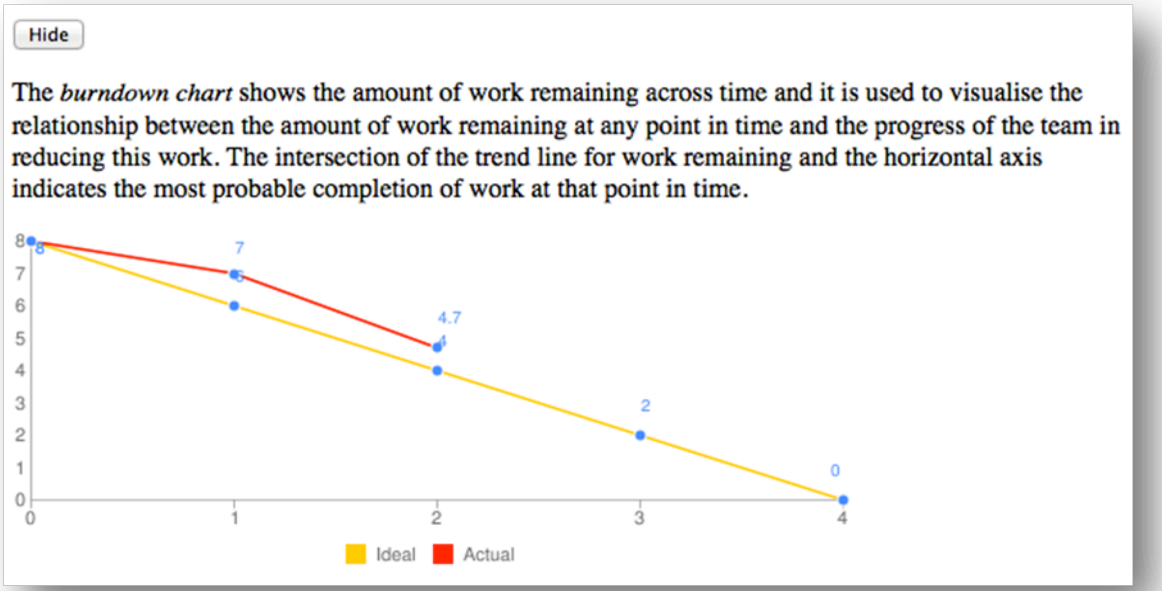


```
    $query3 = "SELECT `opt_id` FROM `Daily_Options` WHERE opt_desc =    '" .$value."' AND q_no = '" .$key."'";
    $result3 = mysql_query($query3);

    for($x = 0;$x <mysql_num_rows($result3);$x++){

        $row3 = mysql_fetch_row($result3);

        $query4 = "SELECT quality_id, quality FROM `Quality` WHERE  pr_num = '".$row[0]."' AND  option_id = '" .$row3[0]."' AND question = '" .$key."'";
        $result4 = mysql_query($query4);
        $row4 = mysql_fetch_row($result4);
    }

$query = "INSERT INTO `Player_has_Quality` VALUES('" .$user. "', '" .$row4[0]."', '".$row2[2]."' )";
$result = mysql_query($query);
```

Figure 5.9: Daily Scrum Meeting code (continued)

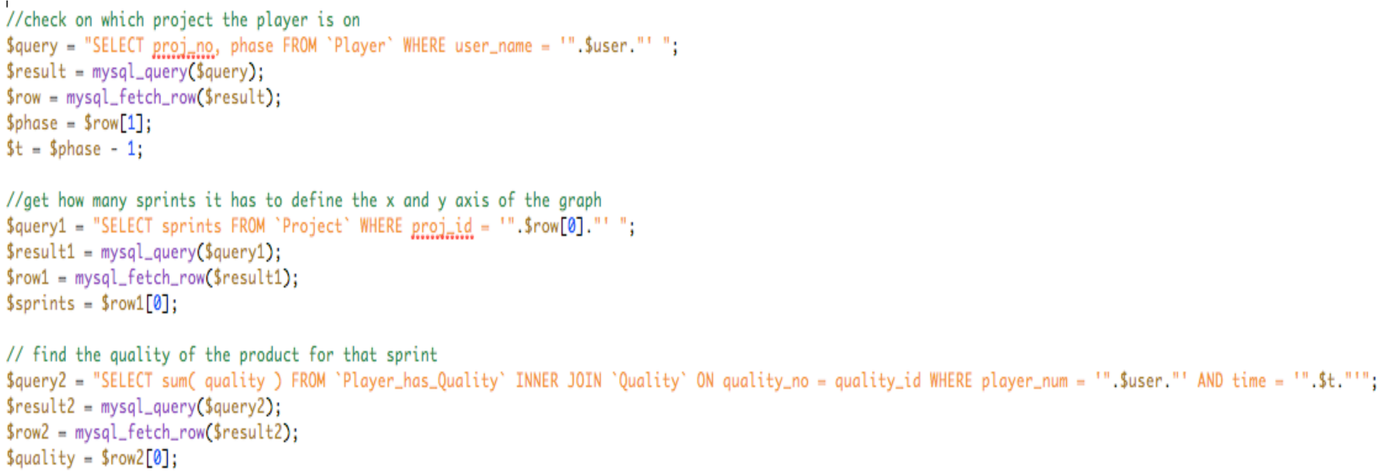
5.3.5 Burndown Chart

Through the Burndown chart, the Scrum Master is able to track the progress of their team at any time and estimate when the final product will be completed. To implement the chart, the author used the Google chart API[12]. The trend line on the chart is a reflection of the quality of the product during that Sprint. The code in Figure 5.11 shows that, to draw the trend line, it is necessary to identify which project the player is implementing, the project's number of sprints to define the values of the x and y axis of the graph, and then to calculate the quality of the product at that point.



The *burndown chart* shows the amount of work remaining across time and it is used to visualise the relationship between the amount of work remaining at any point in time and the progress of the team in reducing this work. The intersection of the trend line for work remaining and the horizontal axis indicates the most probable completion of work at that point in time.

Figure 5.10: Burndown Chart



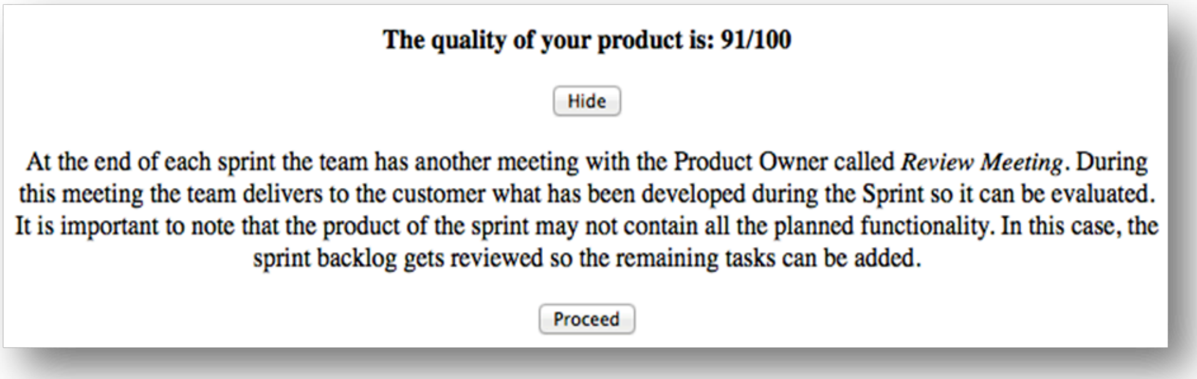
```
//check on which project the player is on
$query = "SELECT proj_no, phase FROM `Player` WHERE user_name = '".$user."' ";
$result = mysql_query($query);
$row = mysql_fetch_row($result);
$phase = $row[1];
$t = $phase - 1;

//get how many sprints it has to define the x and y axis of the graph
$query1 = "SELECT sprints FROM `Project` WHERE proj_id = '".$row[0]."' ";
$result1 = mysql_query($query1);
$row1 = mysql_fetch_row($result1);
$sprints = $row1[0];

// find the quality of the product for that sprint
$query2 = "SELECT sum( quality ) FROM `Player_has_Quality` INNER JOIN `Quality` ON quality_no = quality_id WHERE player_num = '".$user."' AND time = '".$t."'";
$result2 = mysql_query($query2);
$row2 = mysql_fetch_row($result2);
$quality = $row2[0];
```

Figure 5.11: Burndown Chart code

[12] Available from: http://code.google.com/apis/chart/

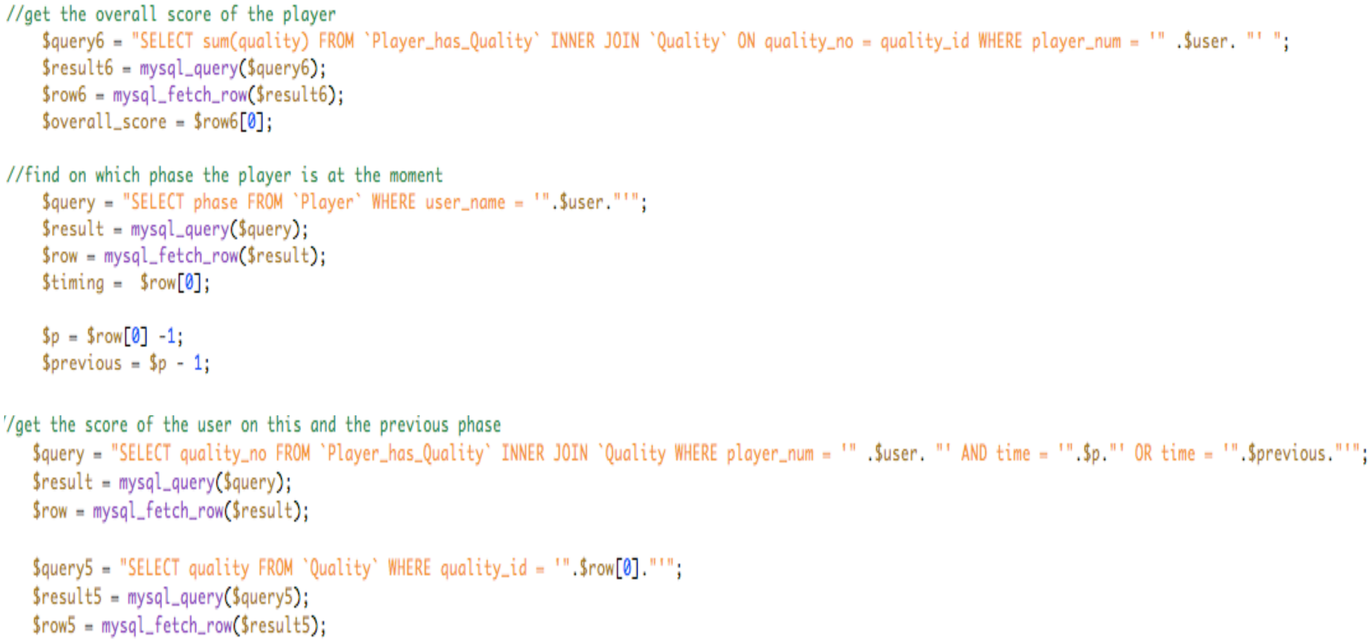
5.3.6 Review Meeting

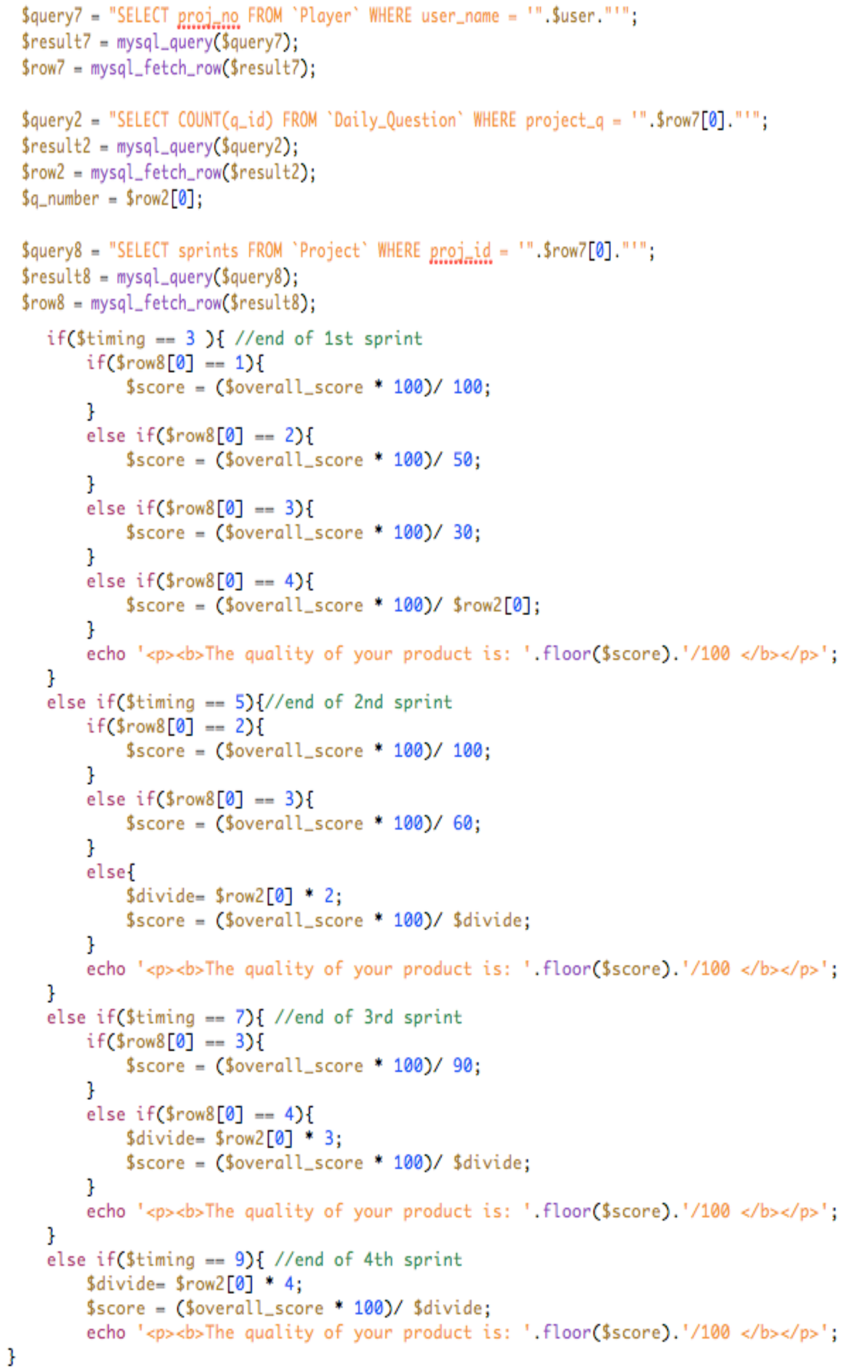
During the Review Meeting in real projects, the team demonstrates to the Product Owner the part of the product that has been implemented so far. The Product Owner then evaluates it towards the agreed system requirements. The same process is simulated in the Scrum Game (Figure 5.12). In particular, the system, in the role of Product Owner, sets the quality above 50% as the criterion of an acceptable product. As mentioned before, the quality of the product depends on the advice that the player (Scrum Master) gives to their team. If the part developed has quality over 50%, then the player can proceed to the next Sprint, otherwise they will have to repeat it (Figure 5.12).


The quality of your product is: 91/100

At the end of each sprint the team has another meeting with the Product Owner called *Review Meeting*. During this meeting the team delivers to the customer what has been developed during the Sprint so it can be evaluated. It is important to note that the product of the sprint may not contain all the planned functionality. In this case, the sprint backlog gets reviewed so the remaining tasks can be added.

Figure 5.12: Review Meeting


```
//get the overall score of the player
    $query6 = "SELECT sum(quality) FROM `Player_has_Quality` INNER JOIN `Quality` ON quality_no = quality_id WHERE player_num = '" .$user. "' ";
    $result6 = mysql_query($query6);
    $row6 = mysql_fetch_row($result6);
    $overall_score = $row6[0];

//find on which phase the player is at the moment
    $query = "SELECT phase FROM `Player` WHERE user_name = '".$user."'";
    $result = mysql_query($query);
    $row = mysql_fetch_row($result);
    $timing =  $row[0];

    $p = $row[0] -1;
    $previous = $p - 1;

'/get the score of the user on this and the previous phase
   $query = "SELECT quality_no FROM `Player_has_Quality` INNER JOIN `Quality WHERE player_num = '" .$user. "' AND time = '".$p."' OR time = '".$previous."'";
   $result = mysql_query($query);
   $row = mysql_fetch_row($result);

   $query5 = "SELECT quality FROM `Quality` WHERE quality_id = '".$row[0]."'";
   $result5 = mysql_query($query5);
   $row5 = mysql_fetch_row($result5);
```

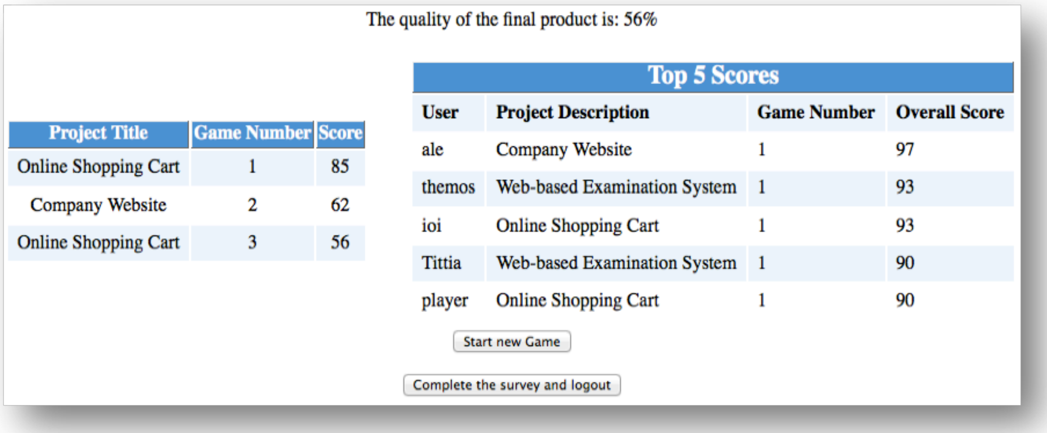


```
$query7 = "SELECT proj_no FROM `Player` WHERE user_name = '".$user."'";
$result7 = mysql_query($query7);
$row7 = mysql_fetch_row($result7);

$query2 = "SELECT COUNT(q_id) FROM `Daily_Question` WHERE project_q = '".$row7[0]."'";
$result2 = mysql_query($query2);
$row2 = mysql_fetch_row($result2);
$q_number = $row2[0];

$query8 = "SELECT sprints FROM `Project` WHERE proj_id = '".$row7[0]."'";
$result8 = mysql_query($query8);
$row8 = mysql_fetch_row($result8);
    if($timing == 3 ){ //end of 1st sprint
        if($row8[0] == 1){
            $score = ($overall_score * 100)/ 100;
        }
        else if($row8[0] == 2){
            $score = ($overall_score * 100)/ 50;
        }
        else if($row8[0] == 3){
            $score = ($overall_score * 100)/ 30;
        }
        else if($row8[0] == 4){
            $score = ($overall_score * 100)/ $row2[0];
        }
        echo '<p><b>The quality of your product is: '.floor($score).'/100 </b></p>';
    }
    else if($timing == 5){//end of 2nd sprint
        if($row8[0] == 2){
            $score = ($overall_score * 100)/ 100;
        }
        else if($row8[0] == 3){
            $score = ($overall_score * 100)/ 60;
        }
        else{
            $divide= $row2[0] * 2;
            $score = ($overall_score * 100)/ $divide;
        }
        echo '<p><b>The quality of your product is: '.floor($score).'/100 </b></p>';
    }
    else if($timing == 7){ //end of 3rd sprint
        if($row8[0] == 3){
            $score = ($overall_score * 100)/ 90;
        }
        else if($row8[0] == 4){
            $divide= $row2[0] * 3;
            $score = ($overall_score * 100)/ $divide;
        }
        echo '<p><b>The quality of your product is: '.floor($score).'/100 </b></p>';
    }
    else if($timing == 9){ //end of 4th sprint
        $divide= $row2[0] * 4;
        $score = ($overall_score * 100)/ $divide;
        echo '<p><b>The quality of your product is: '.floor($score).'/100 </b></p>';
}
```

Figure 5.13: Review Meeting code

5.3.7 Student Log & Top 5 Table

At the end of the game players can view a history of their overall progress within the Scrum Game (Figure 5.14), as well as a score table with the top 5 players of the game. To create the player history table, the system retrieves all the information relevant to the user from the **`Log`** table, whereas for the top 5 score table it takes only the players with the highest scores overall (Figure 5.15).

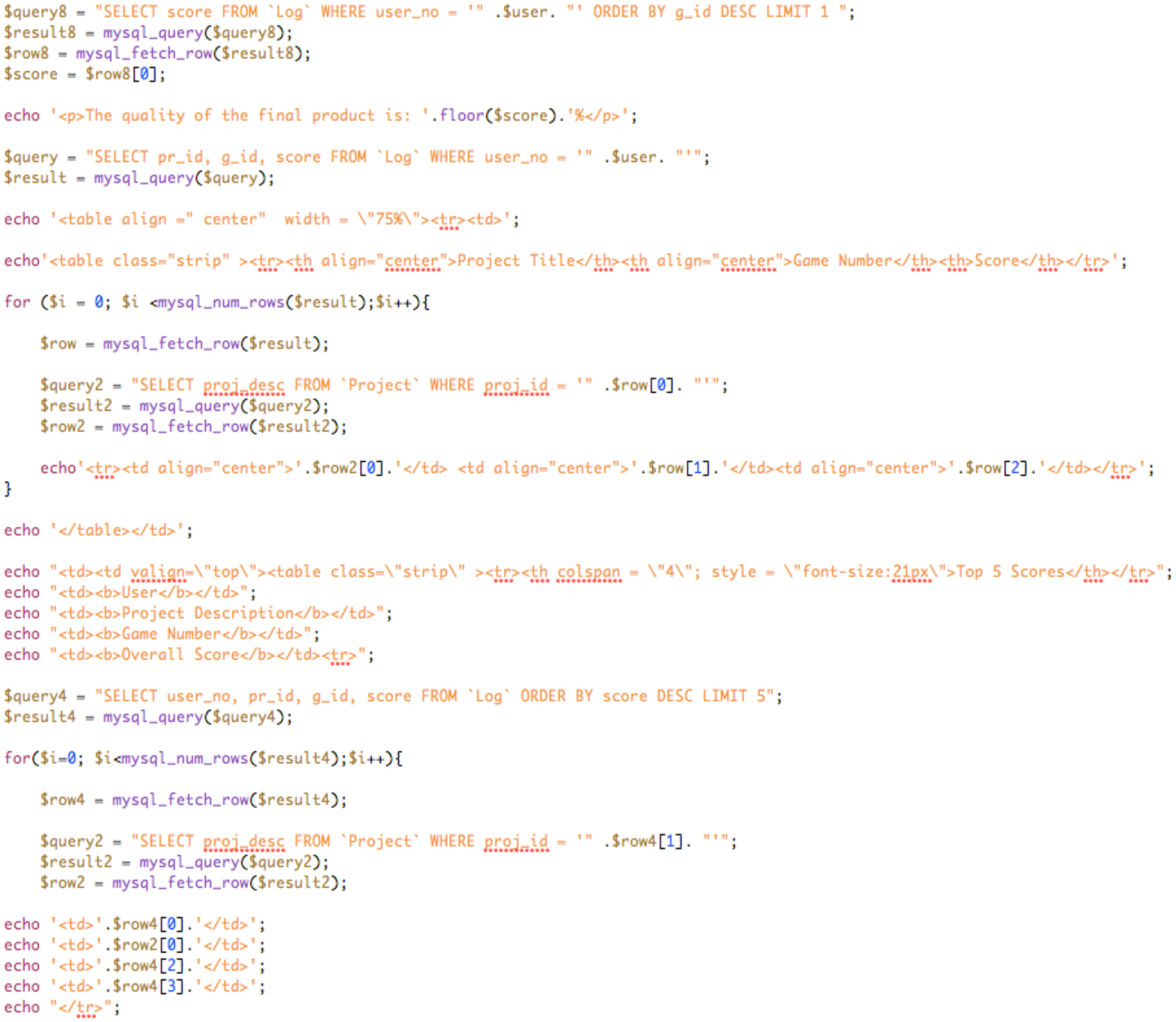


The quality of the final product is: 56%

Project Title	Game Number	Score
Online Shopping Cart	1	85
Company Website	2	62
Online Shopping Cart	3	56

Top 5 Scores			
User	Project Description	Game Number	Overall Score
ale	Company Website	1	97
themos	Web-based Examination System	1	93
ioi	Online Shopping Cart	1	93
Tittia	Web-based Examination System	1	90
player	Online Shopping Cart	1	90

Figure 5.14: Student History & Top 5 Players

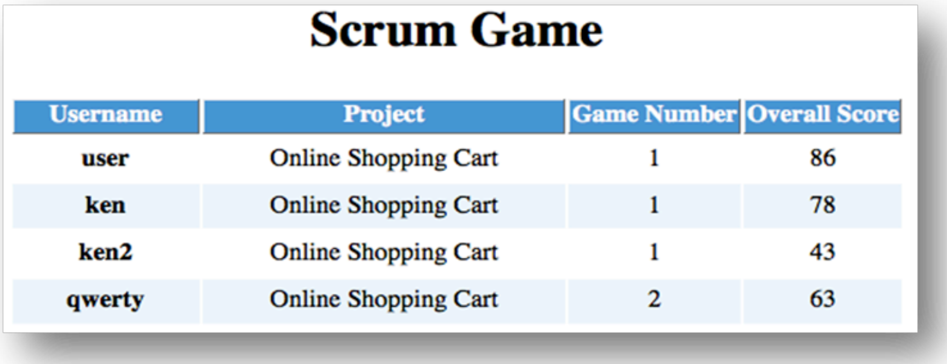


```
$query8 = "SELECT score FROM `Log` WHERE user_no = '" .$user. "' ORDER BY g_id DESC LIMIT 1 ";
$result8 = mysql_query($query8);
$row8 = mysql_fetch_row($result8);
$score = $row8[0];

echo '<p>The quality of the final product is: '.floor($score).'%</p>';

$query = "SELECT pr_id, g_id, score FROM `Log` WHERE user_no = '" .$user. "'";
$result = mysql_query($query);

echo '<table align =" center"  width = \"75%\"><tr><td>';

echo'<table class="strip" ><tr><th align="center">Project Title</th><th align="center">Game Number</th><th>Score</th></tr>';

for ($i = 0; $i <mysql_num_rows($result);$i++){

    $row = mysql_fetch_row($result);

    $query2 = "SELECT proj_desc FROM `Project` WHERE proj_id = '" .$row[0]. "'";
    $result2 = mysql_query($query2);
    $row2 = mysql_fetch_row($result2);

    echo'<tr><td align="center">'.$row2[0].'</td> <td align="center">'.$row[1].'</td><td align="center">'.$row[2].'</td></tr>';
}

echo '</table></td>';

echo "<td><td valign=\"top\"><table class=\"strip\" ><tr><th colspan = \"4\"; style = \"font-size:21px\">Top 5 Scores</th></tr>";
echo "<td><b>User</b></td>";
echo "<td><b>Project Description</b></td>";
echo "<td><b>Game Number</b></td>";
echo "<td><b>Overall Score</b></td><tr>";

$query4 = "SELECT user_no, pr_id, g_id, score FROM `Log` ORDER BY score DESC LIMIT 5";
$result4 = mysql_query($query4);

for($i=0; $i<mysql_num_rows($result4);$i++){

    $row4 = mysql_fetch_row($result4);

    $query2 = "SELECT proj_desc FROM `Project` WHERE proj_id = '" .$row4[1]. "'";
    $result2 = mysql_query($query2);
    $row2 = mysql_fetch_row($result2);

echo '<td>'.$row4[0].'</td>';
echo '<td>'.$row2[0].'</td>';
echo '<td>'.$row4[2].'</td>';
echo '<td>'.$row4[3].'</td>';
echo "</tr>";
```

Figure 5.15: Student History & Top 5 Players code

5.3.8 Administrator Mode

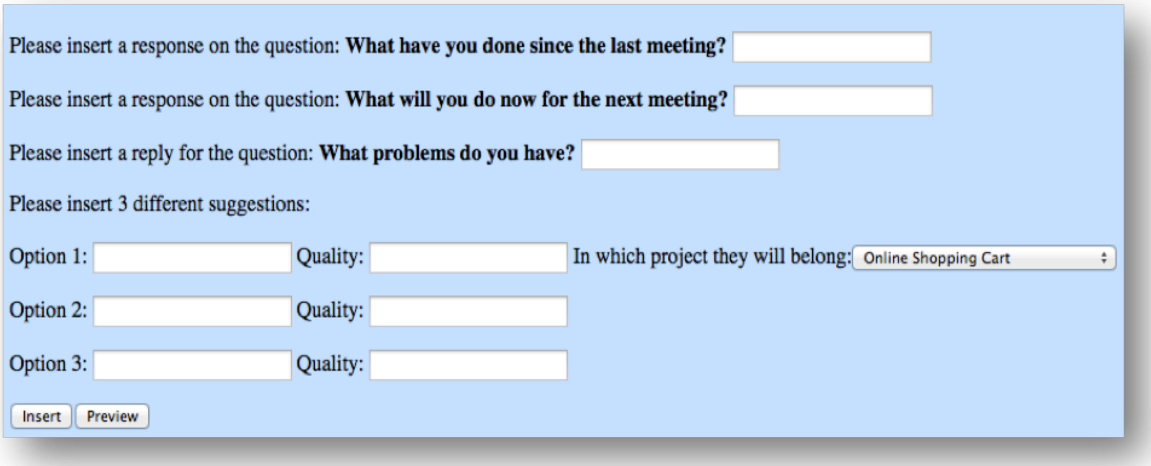
The last important feature of the game is the administrator mode, where tutors and lecturers can view a full log of their students' progress (Figure 5.16). They also have the facility to modify existing projects, or create their own new projects, with employees, project tasks, Daily Scrum Meeting employee responses, etc. (Figure 5.17).



Scrum Game

Username	Project	Game Number	Overall Score
user	Online Shopping Cart	1	86
ken	Online Shopping Cart	1	78
ken2	Online Shopping Cart	1	43
qwerty	Online Shopping Cart	2	63

Figure 5.16: Student Log



Please insert a response on the question: **What have you done since the last meeting?**

Please insert a response on the question: **What will you do now for the next meeting?**

Please insert a reply for the question: **What problems do you have?**

Please insert 3 different suggestions:

Option 1: Quality: In which project they will belong: Online Shopping Cart

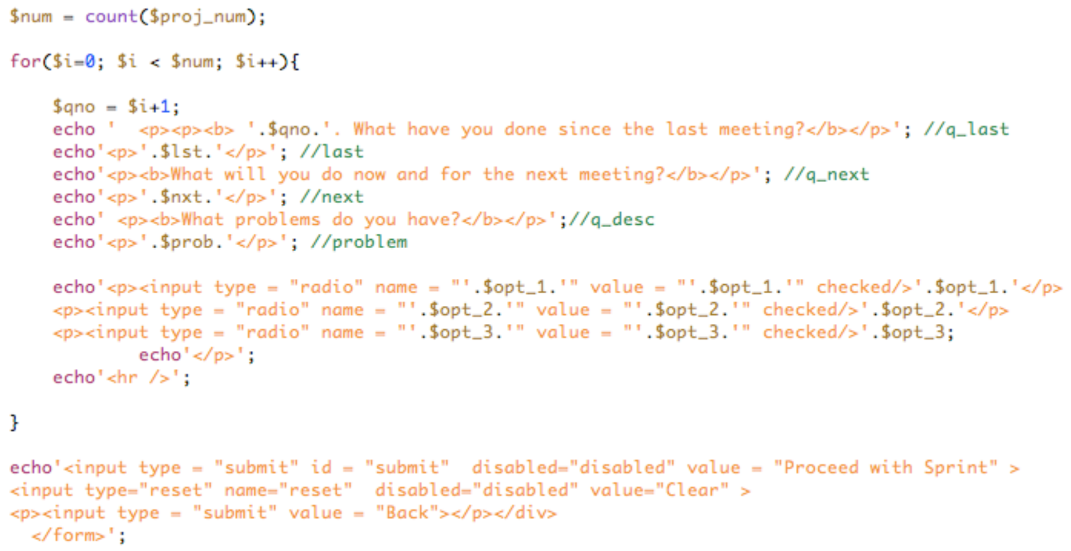
Option 2: Quality:

Option 3: Quality:

Figure 5.17: Insert/Preview Daily Scrum Meeting Questions

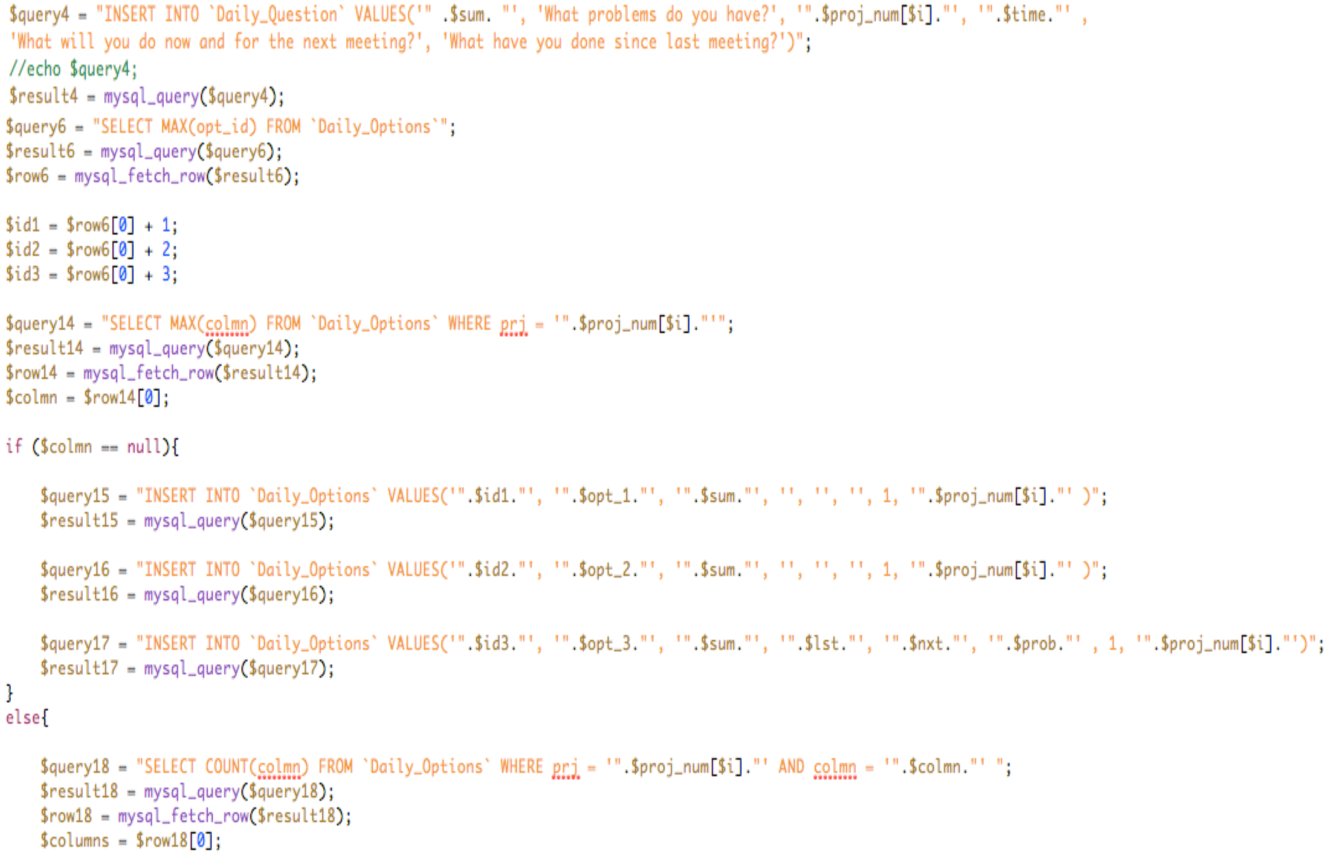
For administrators to insert or preview new questions and answers in a specific game, they need to complete the form shown in Figure 5.17. To preview their query, the system gets the information that the player inserted and displays it as it would normally appear during a Sprint (Figure 5.7). When administrators insert new information, the system checks if there are any previous questions and answers for that particular project, to set the id numbers of the questions, answers and options accordingly, and then add the new information to the database

(Figure 5.18 and Figure 5.19). The same process applies when administrators want to insert or preview new projects, employees, project tasks and employee skills.

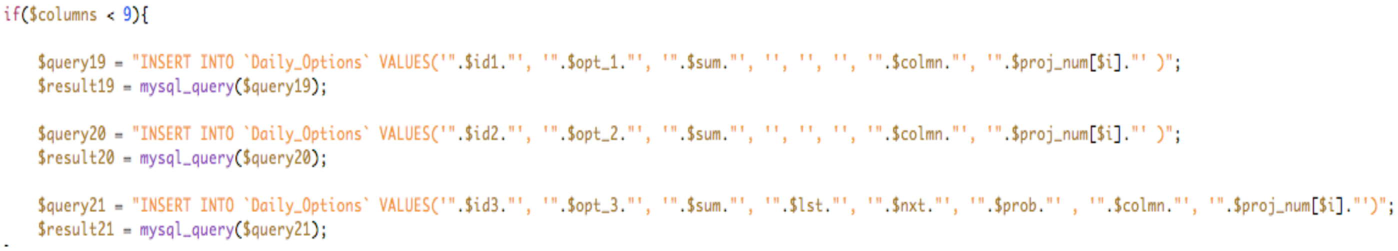


```
$num = count($proj_num);

for($i=0; $i < $num; $i++){

    $qno = $i+1;
    echo '  <p><p><b> '.$qno.'. What have you done since the last meeting?</b></p>'; //q_last
    echo'<p>'.$lst.'</p>'; //last
    echo'<p><b>What will you do now and for the next meeting?</b></p>'; //q_next
    echo'<p>'.$nxt.'</p>'; //next
    echo' <p><b>What problems do you have?</b></p>';//q_desc
    echo'<p>'.$prob.'</p>'; //problem

    echo'<p><input type = "radio" name = "'.$opt_1.'" value = "'.$opt_1.'" checked/>'.$opt_1.'</p>
    <p><input type = "radio" name = "'.$opt_2.'" value = "'.$opt_2.'" checked/>'.$opt_2.'</p>
    <p><input type = "radio" name = "'.$opt_3.'" value = "'.$opt_3.'" checked/>'.$opt_3;
            echo'</p>';
    echo'<hr />';

}

echo'<input type = "submit" id = "submit"  disabled="disabled" value = "Proceed with Sprint" >
<input type="reset" name="reset"  disabled="disabled" value="Clear" >
<p><input type = "submit" value = "Back"></p></div>
  </form>';
```

Figure 5.18: Preview Daily Scrum Meeting Question code



```
$query4 = "INSERT INTO `Daily_Question` VALUES('" .$sum. "', 'What problems do you have?', '".$proj_num[$i]."', '".$time."' ,
'What will you do now and for the next meeting?', 'What have you done since last meeting?')";
//echo $query4;
$result4 = mysql_query($query4);
$query6 = "SELECT MAX(opt_id) FROM `Daily_Options`";
$result6 = mysql_query($query6);
$row6 = mysql_fetch_row($result6);

$id1 = $row6[0] + 1;
$id2 = $row6[0] + 2;
$id3 = $row6[0] + 3;

$query14 = "SELECT MAX(colmn) FROM `Daily_Options` WHERE prj = '".$proj_num[$i]."'";
$result14 = mysql_query($query14);
$row14 = mysql_fetch_row($result14);
$colmn = $row14[0];

if ($colmn == null){

    $query15 = "INSERT INTO `Daily_Options` VALUES('".$id1."', '".$opt_1."', '".$sum."', '', '', '', 1, '".$proj_num[$i]."' )";
    $result15 = mysql_query($query15);

    $query16 = "INSERT INTO `Daily_Options` VALUES('".$id2."', '".$opt_2."', '".$sum."', '', '', '', 1, '".$proj_num[$i]."' )";
    $result16 = mysql_query($query16);

    $query17 = "INSERT INTO `Daily_Options` VALUES('".$id3."', '".$opt_3."', '".$sum."', '".$lst."', '".$nxt."', '".$prob."' , 1, '".$proj_num[$i]."')";
    $result17 = mysql_query($query17);
}
else{

    $query18 = "SELECT COUNT(colmn) FROM `Daily_Options` WHERE prj = '".$proj_num[$i]."' AND colmn = '".$colmn."' ";
    $result18 = mysql_query($query18);
    $row18 = mysql_fetch_row($result18);
    $columns = $row18[0];
```



```
if($columns < 9){

    $query19 = "INSERT INTO `Daily_Options` VALUES('".$id1."', '".$opt_1."', '".$sum."', '', '', '', '".$colmn."', '".$proj_num[$i]."' )";
    $result19 = mysql_query($query19);

    $query20 = "INSERT INTO `Daily_Options` VALUES('".$id2."', '".$opt_2."', '".$sum."', '', '', '', '".$colmn."', '".$proj_num[$i]."' )";
    $result20 = mysql_query($query20);

    $query21 = "INSERT INTO `Daily_Options` VALUES('".$id3."', '".$opt_3."', '".$sum."', '".$lst."', '".$nxt."', '".$prob."' , '".$colmn."', '".$proj_num[$i]."')";
    $result21 = mysql_query($query21);
```

Figure 5.19: Insert Daily Scrum Meeting Questions code

6 Testing

Testing is very important for every project. Because XP method was used for the implementation of this project, the Test Driven Development (TDD) was used to test the different features of the system. The dictum of TDD is “Red, Green, Refactor” (Beck, 2003), which means that all the test cases are written before writing any functionality to be tested. At this stage, all the tests will fail but as soon as some functionality is added, some tests will pass. Then, while the project progresses, more features will be implemented, and all these tests will be re-run to ensure that everything still works as expected. Any previously successful tests that now fail have that part of code refactored, until all tests pass.

6.1 Testing Methods

Various techniques were used to test if all the Scrum Game system features behaved as expected. As XP method dictates, black box testing was performed to check whether the final system conformed to the user stories. Specifically, system testing was performed to ensure that all the system outputs are the expected (B.1-B.67). Then, a white box database test was conducted to check the boundaries and the different types of database values (B.68-B.73). Finally, the system was tested against all its functional (B.74-B.106) and non-functional requirements (B.107-B.118).

6.2 Results of test cases

Test Type	No	Total No of test cases	Passed	Failed
System Testing	B.1-B.67	67	67	0
Database Testing	B.68-B.73	6	6	0
Functional Requirements Testing	B.74-B.106	33	33	0
Non-Functional Requirements Testing	B.107-B.118	12	11	1
Total:		118	117	1

Table 6.1:Result of Test Cases

Table 6.1 shows that 118 test cases were performed, of which only one failed. The test case that failed was number 118 in the non-functional requirements testing, which was “The system should conform to the W3C CSS standards”. This specific feature was of low priority

(*Could*) and it was not implemented because of lack of time. However, the failure of this test does not affect the overall functionality of the system, as it is fully functional with the majority of browsers, therefore the project can be considered as successfully passing testing.

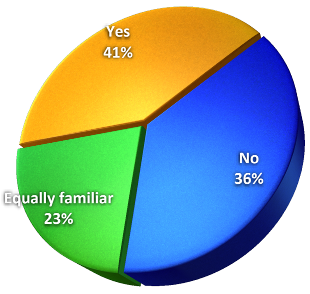
7 Evaluation

The evaluation of the project is important in order to assess whether it fulfilled its aim and delivered the expected outcome. The Scrum Game was evaluated through a questionnaire completed by peer students that was compared against the initial project plan and project goals.

7.1 Questionnaire Results

When the implementation of the game was complete, it had to be evaluated to investigate whether the final system fulfils its initial aim as an educational game, and whether the participants understood in greater depth the use of the Scrum method in real projects. Typically, the evaluation of a system should be undertaken by a large and diverse sample of people. However, due to the limited time available, the questionnaire was distributed to 35 students on the MSc Software Engineering course at the University of Southampton, of different nationalities, ages and educational backgrounds. Some 22 people completed the questionnaire. Participation in the questionnaire was voluntary and all data was gathered anonymously to allow the participants to freely express their opinions. A summary of the questionnaire results follows. More detailed information and graphs can be found in Appendix F.

Even though 77% of the students clearly understand the difference between traditional and agile project management methods, slightly less than a third of them is equally familiar with both project management methods (Figures 7.1 and 7.2).



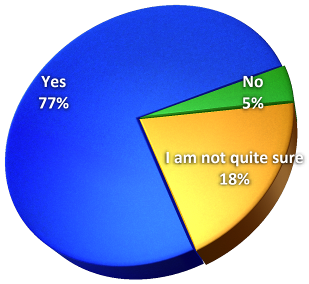
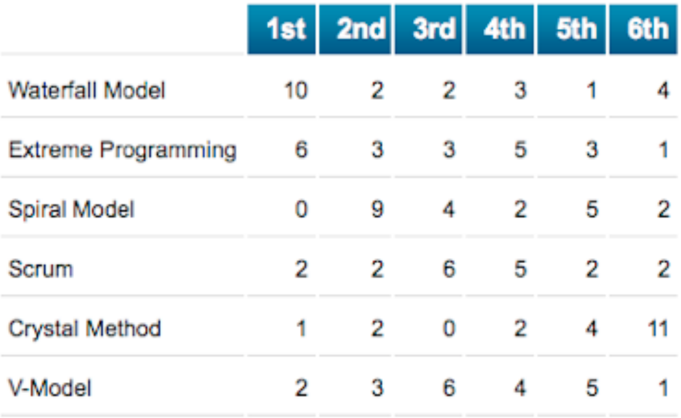


Figure 7.1: Do you believe that you are more
with traditional project management
than agile methodologies?

Figure 7.2: Is it clear the difference between familiar
traditional and agile project management methods rather
methods?

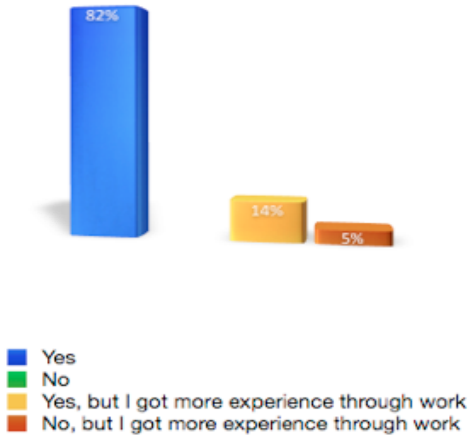
Figure 7.3 shows that indeed participants are more familiar with the Waterfall and Spiral models, followed by the V-Model, Scrum and Extreme Programming.



	1st	2nd	3rd	4th	5th	6th
Waterfall Model	10	2	2	3	1	4
Extreme Programming	6	3	3	5	3	1
Spiral Model	0	9	4	2	5	2
Scrum	2	2	6	5	2	2
Crystal Method	1	2	0	2	4	11
V-Model	2	3	6	4	5	1

Figure 7.3: Please choose with which of the following methods you believe that you are more familiar with (1-most, 6-least).

Even though the majority of players were taught about agile methods (95%), an impressive 32% claimed to be almost completely unaware of the Scrum methodology (Figures 7.4 and 7.5). This shows that, even though university students are being taught about the different agile methods during lectures, they still do not feel familiar and confident enough to apply them in real life.



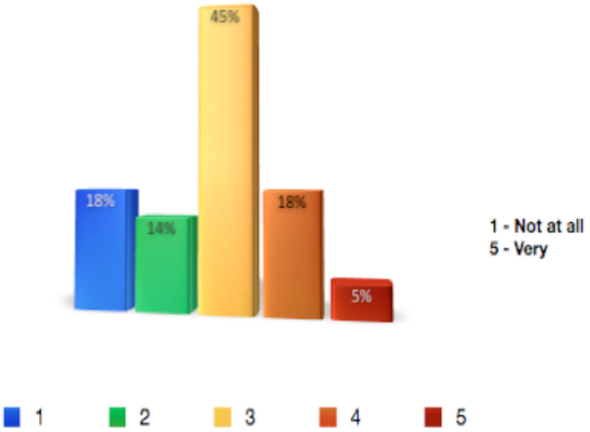


Figure 7.4: Have you been taught about agile methods whilst at University?

Figure 7.5: How familiar are you with Scrum Method?

As far as the actual game is concerned, just over 60% of players believe that almost no prior knowledge was assumed in this game, while 86% of participants felt that they learned more about Scrum through the Scrum Game (Figures 7.6 and 7.7). This shows the system did indeed achieve its educational aim and students actually understood the Scrum lifecycle and its application within different projects.

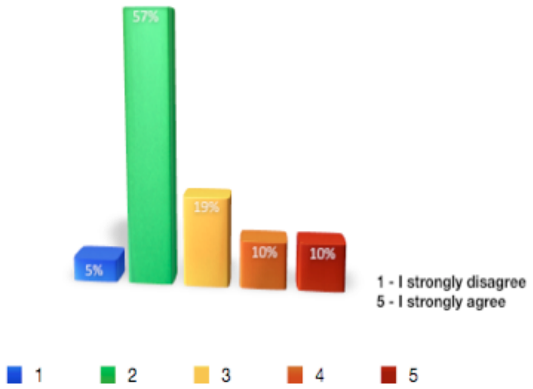


Figure 7.6: Do you think that too much prior knowledge was assumed in this game?

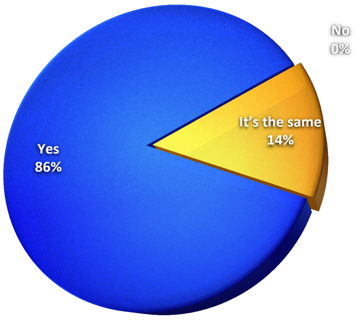


Figure 7.7: Did this game help you learn more about Scrum?

When students where asked if they would like to use such a system as supplementary material, in parallel with their lectures, to get a more practical view of what is being taught during the lecture, the vast majority replied positively (Figure 7.8). The same response was given to the question "Do you think that if this game was part of your Project Management module you would get a better understanding about Scrum?" (Figure 7.9).

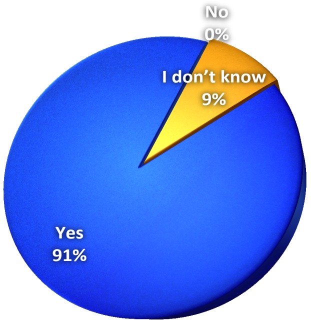


Figure 7.8: Do you think that this game can potentially be used as a supplementary material in parallel with lectures?

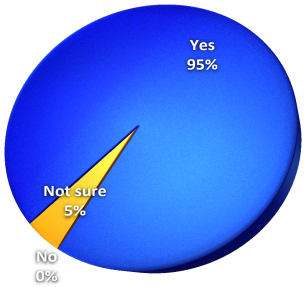
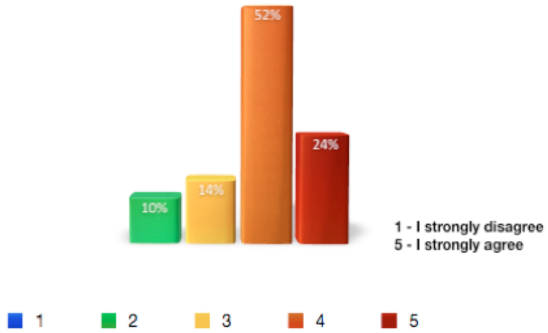


Figure 7.9: Do you think that if this game was part of your Project Management module you would get a better understanding about Scrum?

Finally, over 75% found that the system was enjoyable to play (Figure 7.10) and the overall system was rated 8 out of 10 on average (Figure 7.11) Some positive comments were received, such as "I liked the fact that the game included everyday situations, like arguments with colleagues, or illness. It is important to be able to handle these situations while ensuring the quality of the proposed product" and "Useful and fun way to get experience with Scrum. I would enjoy seeing one as well for XP and Crystal Clear" (Figure D.18).



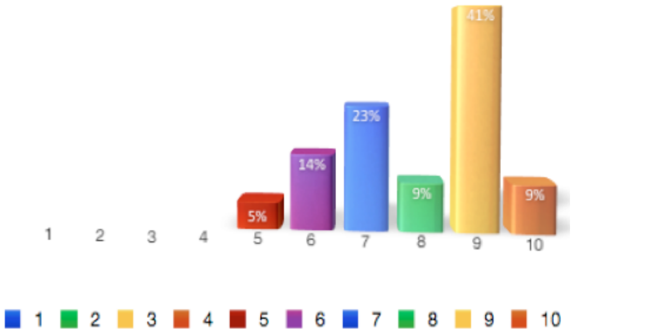


Figure 7.10: Even though the system is a did you find the game enjoyable to play?

Figure 7.11: How would you rate the overall prototype, system?

7.2 Project management

The management of this project has been successful, as both the system and the report of the project were completed and submitted on time. From Figures 3.1 and 3.2 it can be seen that some tasks took longer than expected, which affected the starting dates of the following tasks.

The task that had the biggest impact on progress was the literature review, because the author was researching all available sources to identify the requirements of the system. The extension of that stage occurred because the author underestimated the work required to investigate this field more thoroughly, and in more depth, than had been done for a similar project in the past (the Agile Game). The research for the Agile Game had covered topics concerning the different types of agile method that exist, their main differences from traditional project management methods, as well as different educational games that have been used to teach agile methods in both companies and educational institutions.

For this project, the literature review built on the work that had already been done for the Agile Game, but to differentiate and expand it, more focus was given to the Scrum method, as the target game focused particularly on this method. At the beginning, it was necessary to determine the characteristics that an educational game should have to accomplish its goal and keep players' attention. Because the Scrum Game is an educational system, it was necessary to determine the different kinds of game that are currently used for educational purposes within universities, the academic objectives that these games teach, and to examine how close to reality these games are.

The implementation of the main system took longer than first estimated, because towards the end, some additional functionality had to be added in the administrator mode. As can be seen from Figure 3.2, it required a change of the design of the database. The author also found it challenging to create the Burndown chart in every Sprint, so more time was required to complete this task compared to the initial estimate.

Finally, the use of the XP method to implement the overall project was a very good move as it allowed a very flexible plan that could easily accommodate changes. Specifically, the plan allowed re-estimating the time that had to be spent on the remaining tasks so that they fit the time available, and the resulting impact of the longer-lasting tasks was minimal.

7.3 Project goals

A project is successful if the system achieves the project and system goals that were set at the beginning. Tables 7.1 & 7.2 indicate that indeed all the initial goals have been accomplished.

Project Goals	Outcome	Justification
Meet all the deadlines of the project.	Achieved	As can be seen from Figure 3.2, both the report and the system were delivered on time.
Implement a project that conforms to the initial design.	Achieved	The Scrum Game was implemented following the specified requirements.
Meet all the system goals.	Achieved	As of section 4.4, the final outcome meets all the specified goals.
Create a fully functional prototype of the Scrum Game and its administrator mode.	Achieved	After thorough testing, only less than 1% of the test cases failed which was of low priority.
Implement a game that helps students in more depth the Scrum lifecycle.	Achieved	86% of the questionnaire participants agreed that they learned more about Scrum through this game.

Table 7.1: Evaluation of Project Goals

System Goals	Outcome	Justification
Create a prototype of a game that simulates using Scrum for every step that a project has to undergo from requirements to delivery.	Achieved	The system was rated with 8 out of 10 through a questionnaire and 86% of the players believe that they learned more about Scrum through the game.
The system should have two different modes: game and administrator.	Achieved	The Scrum Game was implemented following the specified requirements.
Create a game where not much prior knowledge on the Scrum methodology is assumed, targeted at students with some IT or Project Management background.	Achieved	The results of the questionnaire showed that just over 60% of the participants found that almost no prior knowledge was assumed during the game.

System Goals	Outcome	Justification
Players should have the role of Scrum Master and they will be responsible for managing a team, helping them to estimate the duration and assign the different tasks to the appropriate Sprints.	Achieved	The system was implemented following the specified requirements. The player is responsible of a team of 3 where they give then advice on problems that occur during each Sprint.
Players should give their team advice on the problems that occur during the Daily Scrum Meetings.	Achieved	Players have the option to select the most appropriate advice among 3 different possible answers.
Each advice should be credited with a score, which affects the quality of the final product.	Achieved	The quality of the developed product depends on the advice that players select during the Daily Meetings.
During each Sprint, players should have a Burndown chart to monitor their progress within the game.	Achieved	Players can view the Burndown chart whose trend line depends on the quality of the project on that stage.
At the end of the game, the system must represent the score of the Top 5 players of the game.	Achieved	Upon completion of the game players can view a top 5-player score table along with a history of all their games.
Administrators should be able to monitor the progress of every player.	Achieved	Administrators are able to see the score of every player through the administrator page.
Administrators should be able to create their own projects, adding new employees, new employee skills, project tasks and Daily Scrum questions and advice.	Achieved	Administrators can customise or create from scratch their own projects, inserting all the necessary information through the administrator page.

Table 7.2: Evaluation of System Goals

8 Conclusion & Future Work

The project was a success, as can be seen from the results from testing, the evaluation questionnaire and the project management. First, it achieved all its system and project aims. This was confirmed by the feedback in the evaluation questionnaire. In particular, it was shown that this prototype game actually helped the participants to familiarise themselves with Scrum methodology. They also thought that it would be useful to have such a system in their Project Management module, in parallel with their lectures in order to get a practical application of this method. The system was quite highly rated (8 out of 10 on average), and the creation of similar games for XP and Crystal method was suggested. Additionally, the system passed over 99% of its test cases, which ensures its correct functionality. The failed test case was of low priority, and concerned only the user interface (which does not affect its functionality); this was not completed due to time constraints. From the project management point of view, though they were some deviations from the initial plan, the use of the XP method expected change during the implementation, so there was space to accommodate them. Thus, their impact was minor and they did not affect the final delivery date or quality of either the system or the report.

This project provided an opportunity to identify the differences between traditional and agile project management methods. To realistically represent the way that Scrum is used within projects, each step of the Scrum lifecycle had to be understood very clearly so that it could be simulated precisely and explained to the players of the Scrum Game. To ensure that the system achieved its educational aim, the types of game currently used for educational purposes during lectures at university level were identified, and their effectiveness. This also covered the characteristics that such games should have to successfully help students understand and apply the subject that they cover. The author also applied her project management skills throughout the project to keep within schedule and deal with changes affecting both delivery and quality of the overall system. Lastly, to complete this project, the author had to make use of various implementation tools, and effectively apply her knowledge of the different languages PHP, MySQL, CSS and HTML, as well as the challenge of using JavaScript and AJAX (jQuery) and Google Charts API.

Even though it is a prototype, the Scrum Game is a system with much potential and many of its aspects can be improved. To make the system better resemble real world situations, additional notions could be introduced such as budget, random factors (e.g. leave of a team member), lack of finance, or even change of requirements and delivery date. Another improvement to the project could be to allow multiple users playing the same game as a team or against each other to see who will deliver their product first with the best quality. Further, to make the system more amusing and interactive, technologies such as Flash could be used to create some graphics and make the user interaction more interesting and smooth. Also, instead of only lecturers being able to edit the existing projects or create new ones, users could play the role of administrator and create their own projects and then play them.

An interesting enhancement of this game would be to allow players to hold different roles throughout each game. In particular, they could play the role of the Product Owner and specify the different project tasks, or be part of the Scrum team so they would have to be responsible for the completion of a specific task and take part in the different Scrum meetings, such as the Daily Scrum Meeting. This way they would have the opportunity to see the application of Scrum through different perspectives and get a good idea of its use within projects. Finally, as suggested by the players of this game, this system could be used as a base to create similar games for other agile methods such as XP, Crystal, FFD and DSDM.

References

Abbas, N., Gravell, A. & Wills, G. (2008) Historical Roots of Agile Methods: Where did "Agile Thinking" Come from? *Methods*, 9, p.94-103. Available at: http://eprints.ecs.soton.ac.uk/16606/.

Abrahamsson P. (2003) Extreme Programming: First Results from a Controlled Case Study *29th Euromicro Conference Proceedings,* 1-6 Sept. 2003

Abrahamsson, P., Warsta, J., Siponen, M. T. & Ronkainen, J. (2003) New directions on agile methods: a comparative analysis. *25th International Conference on Software Engineering 2003 Proceedings*, 6(Proceedings of the 25th International Conference on Software Engineering (ICSE'03)), p.244-254. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1201204.

Aguanno, K., (2004) *Managing Agile Projects* USA: Multi-Media Publications Inc.

Ambler, S.W. (2005-2009) *Feature Driven Development (FDD) and Agile Modelling* Canada: Ambysoft Inc. Available at: http://www.agilemodeling.com/essays/fdd.htm [Accessed: September 2011].

Ambler, S.W. (2006) *Agile Adoption Rate Survey Results: March 2006* Available at: http://www.ambysoft.com/surveys/agileMarch2006.html [Accessed: August 2011].

Ambler, S.W. (2008) *Agile Adoption Rate Survey Results: February 2008* Available at: http://www.ambysoft.com/surveys/agileFebruary2008.html [Accessed: August 2011].

Amory, A. et al. (1999) The use of computer games as an educational tool: identification of appropriate game types and game elements. *British Journal of Educational Technology*, 30(4), p.311-321. Available at: http://www.mackenty.org/images/uploads/3251778.pdf.

Asuncion, H. et al. (2011) Serious Game Development as an Iterative User-Centered Agile Software Project. *Library*, p.44-47.

Baker, A., Navarro, E.O. & Hoek, A.V.D. (2003) An Experimental Card Game for Teaching Software Engineering. In *Software Engineering Education and Training 2003 CSEE T 2003 Proceedings 16th Conference on*. IEEE Computer Society, p. 216. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1191379.

Basturk, R. (2005) The Effectiveness of Computer-Assisted Instruction in Teaching Introductory Statistics. *International Forum of Educational Technology Society*, 8(2), p.170-178.

Beck, K. (2000) *Extreme Programming Explained: Embrace Change*. Boston: Addison-Wesley.

Beck, K. (2003) *Test-Driven Development: by example*. Boston: Addison-Wesley.

Benefield, G. (2008) Rolling Out Agile in a Large Enterprise. *Proceedings of the 41st Annual Hawaii International Conference on System Sciences HICSS 2008*, 0, p.461-461. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4439167.

Brinzarea-Iamandi, B., Darie, C. & Hendrix, A. (2009) *AJAX and PHP: Building Modern Web Applications* (2nd ed.). GBR: Packt Publishing Ltd.

Castro, E. (2007) *HTML, XHTML & CSS* (3rd ed.). USA: Peachpit Press.

Cohn, M. (2007) *Agile Estimating and Planning* (6th ed.). USA: Pearson Education Limited.

Clua, E., Feijó, B., Rocca, J., Schwartz, J., Perlin, K., Tori, R. & Barnes, T. (2006) Game and Interactivity in Computer Science Education In: *SIGGRAPH '06: SIGGRAPH 2006 Educators program*. ACM.

Collofello, J.S. (2000) University/industry collaboration in developing a simulation-based software project management training course. *IEEE Transactions on Education*, 43(4), p.389-393. Available at: http://dx.doi.org/10.1109/13.883347.

Drappa, A. & Ludewig, J. (2000) Simulation in software engineering training. In *Software Engineering 2000 Proceedings of the 2000 International Conference on*. ACM, pp. 199-208. Available at: http://portal.acm.org/citation.cfm?id=337180.337203.

Fernandes, J.M. & Sousa, S.M. (2010) PlayScrum - A Card Game to Learn the Scrum Agile Method. In *2010 Second International Conference on Games and Virtual Worlds for Serious Applications*. IEEE, pp. 52-59. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5460150.

Ferrari, M., Taylor, R. & VanLehn, K. (1999) *Adapting Work Simulations for Schools* J.Educational Computing Research, Vol. 21(1) p.25-53.

Garratt, P. W. (1995) *Simulation & Gaming - The Software Management Game*, Sage Publications.

Gee, J.P. (2003) What video games have to teach us about learning and literacy. *Computers in Entertainment*, 1(1), p.20. Available at: http://portal.acm.org/citation.cfm?doid=950566.950595.

Gkritsi, A. (2010) *Agile Game: A Project Management Game for Agile Methods.* University of Southampton. Available at: http://www.edshare.soton.ac.uk/7538/.

Highsmith, J. (2002 a) *Agile Software Development Ecosystems* Indianapolis: Addison Wesley.

Highsmith, J. (2002 b) *Agile Project Management Principles and Tools* Cutter Consortioum Executive Report 4(2).

Highsmith, J. (2010) *Agile Project Management* USA: Pearson Education, Inc.

Hillson, D. (2009) *Managing Risk in Projects: Fundamentals of Project Management*. England: Gower Publishing Limited.

Hossain, E., Babar, M.A. & Paik, H.Y. (2009) Using Scrum in Global Software Development: A Systematic Literature Review. In *2009 Fourth IEEE International Conference on Global Software Engineering*. IEEE, pp. 175-184. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5196931.

Huo, M., Verner, J. & Babar, M.A. (2004) Software quality and agile methods. *Proceedings of the 28th Annual International Computer Software and Applications Conference 2004 COMPSAC 2004*, 1, p.520-525. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1342889.

Larman, C. (2004) *Iterative development: A Manager's guide* Boston: Pearson Education.

Lindvall, M., Basili, V., Boehm, B., Costa, P., Dangle, K., Shull, F., Tesoriero, R., Williams, L. & Zelkowitz, M. (2002) Empirical Findings in Agile Methods *Computer Science Notes* vol. 2418/2002, Springer, Berlin pp.81-92.

Lubke, D. & Schneider, K. (2005) *Agile Hour: Teaching XP Skills to Students and IT Professionals*, Springer Berlin Heidelberg. Available at: http://www.springerlink.com/content/xh6hu65cwl73embb/?p=6c22318b2e454ed9a3103089ef074d57&pi=39.

Mall, R. (2004) *Fundamentals of Software Engineering* India: PHI Learning Pvt. Ltd.

McFarland, D. S. (2008) *JavaScript: The Missing Manual*. USA: O'Reilly Media, Inc.

Moreno-Ger, P. et al. (2007) Adaptive Units of Learning and Educational Videogames 1 Introduction. *Learning*, (Stahl 2002), p.1-15.

Moreno-Ger, P. et al. (2008) Educational game design for online education. *Computers in Human Behavior*, 24(6), p.2530-2540. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0747563208000617.

Navarro, E. O., Baker, A. & van der Hock, A. (2004) Teaching software engineering using simulation games. In *ICSIE '04: Proceedings of the 2004 International Conference on Simulation in Education*. Available at: http://dl.acm.org/citation.cfm?doid=337180.337203.

Navarro, E.O. & Van Der Hoek, A. (2005) Design and evaluation of an educational software process simulation environment and associated model. In *Software Engineering Education and Training 2005 CSEET 2005 Proceedings 18th Conference on*. IEEE Comput. Soc, pp. 25-32.

Navarro, E. & Van Der Hoek, A. (2009) Multi-site evaluation of SimSE. In *Proceedings of the 40th ACM technical symposium on Computer science education*. ACM, pp. 326-330. Available at: http://doi.acm.org/10.1145/1508865.1508981.

Nixon, R. (2009) *Learning PHP, MySQL and JavaScript*. CA: Pogue Press O'Reilly.

Oh, E. (2002) Teaching Software Engineering Through Simulation. In *Proceedings of the International Conference on Software Engineering*, Orlando, Florida, USA.

Oh, E. & Van Der Hoek, A. (2002) Towards game-based simulation as a method of teaching software engineering. In *Frontiers in Education 2002 FIE 2002 32nd Annual*. IEEE, p. S2G-13 vol.3. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1158674.

Oh Navarro, E. & Van der Hoek, A. (2004) SimSE: an interactive simulation game for software engineering education. In *Proceedings of the 7th IASTED International Conference on Computers and Advanced Technology in Education,* Kauai, Hawaii, August 2004.

Panda, S. (2010) *An interactive Training Game for Agile Project Management* Available at: Electronics & Computer Science Department, University of Southampton.

Parsons, D., Ryu, H. & Lal, R. (2007) *The Impact of Methods and Techniques on Outcomes from Agile Software Development Projects* [online] Available at: http://www.massey.ac.nz/~dpparson/ParsonsRyu.pdf [Accessed: September 2011].

Playfair, K. (2008) *Scrum Mechanics – Introduction to the Basic Scrum Engine* Available at: http://blogs.collab.net/agile/2008/07/14/scrum-mechanics-introduction-to-the-basic-scrum-engine/ [Accessed: July 2011].

Pressman, R. (2009) *Software Engineering: A Practitioner's Approach* 7th ed. NY: McGraw-Hill.

Priyadarshee, D. (2010) *An Educational Game on Agile Methods* University of Southampton Available at: Electronics & Computer Science Department, University of Southampton.

Qian, H. & M. & Zheng, C. (2009) A Embedded Software Testing Process Model. *2009 International Conference on Computational Intelligence and Software Engineering*, (11-13 December 2009), p.1-5. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5366362.

Salo, O. & Abrahamsson, P. (2008) Agile methods in European embedded software development organisations : a survey on the actual use and usefulness of Extreme Programming and Scrum. *Engineering and Technology*, 2(1), p.58- 64. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4460895.

Schwaber, K. (2004) *Agile Project Management with Scrum* USA: Microsoft Press.

Schwaber, K. & Sutherland, J. (2011) *The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game* Available at: www.scrum.org/storage/scrumguides/Scrum%20Guide%20-%202011.pdf [Accessed: August 2011].

Scrum Alliance (2009) *Scrum Alliance: transforming the world to work* [online] Available at: http://www.scrumalliance.org/pages/what_is_scrum [Accessed: September 2011].

Serena (2007) *An Introduction to Agile Software Development* [online] Available at: http://www.serena.com/docs/repository/solutions/intro-to-agile-devel.pdf [Accessed: September 2011].

Suganya, G. & Mary, S.A.S.A. (2010) Progression towards Agility: A comprehensive survey. In *Computing Communication and Networking Technologies ICCCNT 2010 International Conference on*. IEEE, pp. 1-5. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5591769.

Van Der Duim, L., Anderson, J. & Sinnema, M. (2007) Good practices for Educational Software Engineering Projects Louwarnoud van der Duim. *Engineering*, p.698-707. Available at: http://portal.acm.org/citation.cfm?id=1248820.1248900.

Welling, L. & Thomson, L. (2009) *PHP and MySQL Web Development*. USA: Pearson Educational Inc.

Appendix A – System Requirements

Table A.1 represents the functional requirements of the Scrum Game while Table A.2 shows the non-functional requirements. Each table contains prioritised the different features of the system, along with estimate of the effort to implement that particular task (1 being the least and 10 the maximum effort). The tables also illustrate the priority of each task in terms of the MoSCoW prioritisation technique.

Functional Requirements			
No	User Story	Effort	Priority
1	Players will be able to enter the system from the Web.	1	Must
2	Players will have to insert their username and password to log in.	3	Must
3	New players will be required to fill in a registration form in order to add their details to the database.	3	Must
4	Passwords should be stored hashed in the system database.	5	Must
5	Passwords should be viewed by other users.	5	Must
6	No prior knowledge on Scrum should be assumed, so explanatory directions on the Scrum terms and lifecycle should be provided throughout the game.	4	Must
7	A help page should also be provided with more information on Scrum and other agile methods.	4	Must
8	The game should have an administrator mode where only lecturers and tutors can manage the Scrum Game.	8	Must
9	The game should also have a game mode where students login and play the Scrum Game.	8	Must
10	In the admin page, users should be able to alter the already existing projects adding new features e.g. more employees, questions, employee skills, project tasks etc.	9	Must
11	Administrators should be able to add their own projects.	6	Must
12	Administrators should be able to add their own employees.	5	Must
13	Administrators should be able to add their own employee skills.	5	Must
14	Administrators should be able to add their own project tasks.	6	Must
15	Administrators should be able to add their own Daily Scrum options.	9	Must
16	Players should have the option to select among at least 3 different projects.	6	Must
17	For new users, detailed instructions about the game should be provided.	2	Should
18	If players quit the game whilst playing, they should be able to resume from the point they quit.	6	Should
19	Players should have the role of Scrum Masters within the game.	6	Must
20	Players should be able to select at least 3 employees for their teams.	6	Must

Functional Requirements			
No	User Story	Effort	Priority
21	Each of these employees should have different name, skill, salary and experience.	6	Must
22	Players should be presented with a number of tasks where they will have to select the duration and effort to implement them.	7	Must
23	Players should be able to assign each of these tasks to specific sprints.	6	Must
24	During each Sprint the idea of Daily Scrum Meetings should be introduce.	8	Must
25	Players should be able to track their progress using a Burndown chart.	8	Could
26	When players finish their game, they should be able to see a history of all the games they have already played.	6	Could
27	When players finish their game, they should be able to see a table with the top 5 players of the Scrum Game.	6	Could
28	Players should not be able to delete their history.	2	Must
29	During the Daily Scrum Meetings players should have a selection of different advice to solve the problems that their team has.	9	Must
30	The quality of the final product should be affected by the quality of the advice that the Scrum Manager gives to their team.	7	Must
31	Each advice should be credited with a specific number of points that correspond to the quality of the final product.	6	Must
32	The trend line of the Burndown chart should depend on the quality of the product on that particular sprint.	9	Must
33	Administrators should be able to preview their query before adding it to the database.	8	Could

Table A.1: Functional Requirements

Non-Functional Requirements			
No	User Story	Effort	Priority
34	The system should be targeted to students of the IT field, as well as to project management literate students.	-	Must
35	The system should be available from the Web.	9	Could
36	No personal information concerning should be kept.	-	Must
37	The database should be secure.	-	Must
38	The system should be available at all times.	-	Must
39	The overall system should be secure and robust.	-	Must
40	Player passwords should be store encrypted.	-	Must
41	The system should allow users to repeat their game.	5	Must
42	The system should allow users to play with more than one project.	5	Must
43	If users quit the game whilst playing, the next time they login the system should redirect them to the place where they quit the game.	4	Must
44	The system needs to be fully functional with all major browsers.	-	Must
45	The system should conform to the W3C CSS standards.	-	Could

Table A.2: Non-Functional Requirements

Appendix B – Test Cases

The following tables show the tests that took place to check that the system behaves as it was expected.

B.1 System Testing – Black Box Testing

No	Test Case	Expected Outcome	Result
1	Players access the initial page of the system.	Initial page is loaded.	Pass
2	Players click "Log In" button and log in without being registered.	Notification that they are not logged in appears and they get redirected to the homepage.	Pass
3	Players log in without entering a username.	Notification message that user id textbox is blank appears.	Pass
4	Players log in without entering a password.	Notification message that password textbox is blank appears.	Pass
5	Players click on "Register" link.	Page registration.php is loaded.	Pass
6	Players insert valid username and password to register.	Player details inserted in the database and start the game.	Pass
7	Players insert only their username to register.	Notification message that password textbox is blank appears.	Pass
8	Players insert only their username to register.	The system does not record details.	Pass
9	Players insert only their password to register.	Notification message appears notifying that the user id textbox is blank.	Pass
10	Players insert only their password to register.	No details have been recorded by the system.	Pass
11	Players leave all fields of register form blank.	Notification message that details are missing appears.	Pass
12	Players leave all fields of register form blank.	Details have not been recorded by the system.	Pass
13	While players type a username, the system checks username availability.	Message about availability appears while users type next to username textbox.	Pass
14	Players insert usernames fewer than 3 characters long to register.	Pop-up message that username should be at least 3 characters long appears.	Pass
15	Username contains characters different from [^A-Za-z0-9_-] to register.	Pop-up message that only letters, numbers, _, - and ^ characters are allowed appears.	Pass
16	Players insert valid username and password to register.	Players registered and notification that they can now log in.	Pass
17	Players click on "Log In" link.	Users are redirected in the homepage.	Pass

No	Test Case	Expected Outcome	Result
18	Players click on “Log In” button leaving the login form blank.	HTML5 notification appears informing that details are missing.	Pass
19	Players insert only their username to log in.	HTML5 notification that user id textbox is blank appears.	Pass
20	Players insert only their password to log in.	HTML5 notification that password textbox is blank appears.	Pass
21	Players insert usernames fewer than 3 characters long to register.	Pop-up message that username should be at least 3 characters long appears.	Pass
22	Username contains characters different from [^A-Za-z0-9_-] to register.	Pop-up message that only letters, numbers, _, - and ^ characters are allowed appears.	Pass
23	Players insert valid username and password to log in.	Players enter proceed and enter the game.	Pass
24	Display the username of player when they enter the game.	Display “Hello <username>”.	Pass
25	Players click on “Log Out” button to exit the game.	Users exit the game and are redirected in the home page.	Pass
26	Players log in again to the system.	Players are redirected in from the page they last quit the game.	Pass
27	Players click select without selecting any checkboxes in welcome.php.	A pop-up message appears notifying that at least 1 project should be selected.	Pass
28	Players select more than 1 project to implement.	A pop-up message appears notifying that at only 1 project should be selected at a time.	Pass
29	Players click on the “Read Instructions” button for the detailed instructions of the game.	On click a block of text should appear with 10 game instructions.	Pass
30	Players click on a checkbox and press “Select” button on welcome.php.	Players are redirected on team.php to assemble their team.	Pass
31	An explanatory text should give players information about this stage.	An explanatory text appears on the top of the page.	Pass
32	Players should be able to hide the above text.	Players click on the “Hide” button and this text disappears.	Pass
33	Players click select without selecting any checkboxes in team.php.	A pop-up message appears notifying that at 3 employees should be selected.	Pass
34	Players select more than 3 employees for their team.	A pop-up message appears notifying that at only 3 employees should be selected at a time.	Pass
35	Players click on 3 checkboxes and press “Pick team” button on team.php.	Players are redirected on the product_backlog.php.	Pass
36	An explanatory text should appear giving players information about this stage.	An explanatory text appears on the top of the page.	Pass
37	Players should be able to hide the above text.	Players click on the “Hide” button and this text disappears.	Pass

No	Test Case	Expected Outcome	Result
38	Players select the effort and duration of each task in the product backlog and click "Ok".	Players are redirected on sprint_backlog.php.	Pass
39	An explanatory text should give players information about this stage.	An explanatory text appears on the top of the page.	Pass
40	Players should be able to hide the above text.	Players click on the "Hide" button and this text disappears.	Pass
41	Players select the Sprint of each task and click on "Start Sprint".	A loading image appears and users get redirected to sprint.php.	Pass
42	During the sprint players should be able to view the Burndown chart.	When clicking "View Burndown Chart" a new tab opens with the Sprints' Burndown chart.	Pass
43	The Burndown chart should have 2 trend lines, the ideal and the actual.	2 trend lines appear on the Burndown chart.	Pass
44	The trend of the ideal trend line should remain unchanged during all Sprints.	The ideal trend line remains unchanged during all Sprints.	Pass
45	The trend of the actual trend line should depend on the quality of the project until that point.	The actual trend line changes depending on the quality of the project on each Sprint.	Pass
46	On the sprint.php, players should be presented with 3 different questions.	3 questions appear for each team member.	Pass
47	Under the above questions a reply should appear.	A reply appears under each of the questions above.	Pass
48	3 radio buttons should appear under the last reply.	3 radio buttons appear under the last reply of all 3 employees.	Pass
49	Players make their selections and click on the "Proceed with Sprint" button.	A processing image appears and users get redirected to sprint.php.	Pass
50	Users click on "Clear" button in sprint.php.	All radio buttons return into their default position.	Pass
51	When players have completed the 2-day Sprints, they get redirected to the review_meeting.php.	Players get redirected to the review_meeting.php where they can see the quality of their product.	Pass
52	If the product quality on the review meeting is below 50% users are notified that they have to repeat the same Sprint.	Players get notified that the product quality does not meet the Product Owner expectations and they get redirected to the start of the same Sprint.	Pass
53	If the product quality on the review meeting is above 50% players can proceed with the next Sprint of finish the game.	If the project has more Sprints, players continue with the next Sprint, otherwise they finish the game.	Pass
54	In the end.php players should be able to see a history of their progress and a top 5-player score table.	A table with the history of the player in the game and a top 5-player score table appears.	Pass
55	Administrators login using their username and password correctly.	Admins are redirected to the admin.php page.	Pass

No	Test Case	Expected Outcome	Result
56	Administrators insert only their username to log in.	HTML5 notification that user id textbox is blank appears.	Pass
57	Administrators insert only their password to log in.	HTML5 notification that password textbox is blank appears.	Pass
58	Administrators click on "Log In" button leaving the login form blank.	HTML5 notification appears informing that details are missing.	Pass
59	Administrators should be able to add a new project description.	A "New Project" button exists where they enter the project description and the number of Sprints.	Pass
60	Administrators should be able to preview their project description before submitting.	A "Preview" button exists where they can preview their query.	Pass
61	Administrators should be able to add a new employee.	A "New Employee" button exists where they enter the employee name, skill, salary and experience.	Pass
62	Administrators should be able to preview their new employee before submitting.	A "Preview" button exists where they can preview their query.	Pass
63	Administrators should be able to add a new skill.	A "New Skill" button exists where they enter the skill description and its appropriate salary.	Pass
64	Administrators should be able to add a new task.	A "New Task" button exists where they enter the task description, number, priority and on which project it belongs.	Pass
65	Admins should be able to preview their new task before submitting.	A "Preview" button exists where they can preview their query.	Pass
66	Administrators should be able to add a new question.	A "New Question" button exists where they enter the answer to the 3 given questions as well as the advice to their team with the corresponding quality.	Pass
67	Administrators should be able to view a complete history of all their students.	A "Get Student Log" button exists where on click administrators can to view a log of their students.	Pass

TableB.1: System testing

B.2 Database Testing

No	Test Case	Expected Outcome	Result
68	Players insert a username that exceeds 45 characters.	The database stores only 45 characters of the username.	Pass
69	Players insert a password that exceeds 45 characters.	The database stores only 45 characters of the password.	Pass

No	Test Case	Expected Outcome	Result
70	Players do not insert any username.	HTML notification appears warning that username is blank.	Pass
71	Players do not insert a password.	HTML notification appears warning that password is blank.	Pass
72	Players leave either username or password fields empty.	If HTML5 is not supported from players' browser, blank fields are not inserted in the database.	Pass
73	Administrators leave text boxes empty on admin mode on submit.	HTML notification appears warning that a textbox is blank.	Pass

Table B.2: Database testing

B.3 Functional Requirements Testing

No	Test Case	Expected Outcome	Result
74	Players will be able to enter the system from the Web.	Initial page is loaded.	Pass
75	Players will have to insert their username and password to log in.	If users insert both their username and password, they enter the game.	Pass
76	New players will be required to fill in a registration form to store their details in the database.	The registration form is loaded, users insert their correct details and the information is stored in the database.	Pass
77	Passwords should be stored hashed in the system database.	Passwords are stored hashed in the database using **`sha1()`** function.	Pass
78	Passwords should be viewed by other users.	Passwords are represented as dots while players type their passwords.	Pass
79	No prior knowledge on Scrum should be assumed, so explanatory directions on the Scrum terms and lifecycle should be provided throughout the game.	Explanatory text is provided at the top of each page.	Pass
80	A help page should also be provided with more information on Scrum and other agile methods.	A help button is provided in every stage of the game at the top left of the page.	Pass
81	The game should have an administrator mode where only lecturers and tutors can manage the Scrum Game.	When inserting the admin username and password, the admin page should be loaded.	Pass
82	The game should also have a game mode where students login and play the Scrum Game.	When players insert their username and password, the welcome page should be loaded.	Pass

No	Test Case	Expected Outcome	Result
83	In the admin page, users should be able to alter the already existing projects adding new features e.g. more employees, questions, employee skills, project tasks etc.	For every feature a review and submit query button should be provided.	Pass
84	Administrators should be able to add their own projects.	Admins fill in the form, submit their query and a confirmation message appears.	Pass
85	Administrators should be able to add their own employees.	Admins fill in the form, submit their query and a confirmation message appears.	Pass
86	Administrators should be able to add their own employee skills.	Admins fill in the form, submit their query and a confirmation message appears.	Pass
87	Administrators should be able to add their own project tasks.	Admins fill in the form, submit their query and a confirmation message appears.	Pass
88	Administrators should be able to add their own Daily Scrum options.	Admins fill in the form, submit their query and a confirmation message appears.	Pass
89	Players should have the option to select among at least 3 different projects.	A table with at least 3 different projects should appear.	Pass
90	For new users, detailed instructions about the game should be provided.	In the welcome page a “Read instructions” appears and when clicked, the instructions appear.	Pass
91	If players quit the game whilst playing, they should be able to resume from the point they quit.	The system checks the phase that the specific player is and redirects them to the appropriate page.	Pass
92	Players should have the role of Scrum Masters within the game.	Users hold the role of the Scrum Master of the game.	Pass
93	Players should be able to select at least 3 employees for their teams.	A table with at least 3 different employees should appear.	Pass
94	Each of these employees should have different name, skill, salary and experience.	A table should appear with the employee name, skill, salary and experience.	Pass
95	Players should be presented with a number of tasks where they will have to select the duration and effort to implement them.	A table with all the tasks of the selected project should appear.	Pass
96	Players should be able to assign each of these tasks to specific sprints.	Through a drop-down box players should be able to select on which sprint to assign each task.	Pass
97	During each Sprint the idea of Daily Scrum Meetings should be introduce.	Have 2 Daily Scrum Meetings for every Sprint; sprint.php is loaded.	Pass

No	Test Case	Expected Outcome	Result
98	Players should be able to track their progress using a Burndown chart.	During the Daily Meeting a "View Burndown Chart" should be available and graph is loaded in a new tab.	Pass
99	When players finish their game, they should be able to see a history of all the games they have already played.	A table with players' full history of games should appear.	Pass
100	When players finish their game, they should be able to see a table with the top 5 players of the Scrum Game.	A table with the top 5 players of the Scrum Game should appear.	Pass
101	Players should not be able to delete their history.	Players are only allowed to view their history.	Pass
102	During the Daily Scrum Meetings players should have a selection of different advice to solve the problems that their team has.	A selection of 3 different options appear.	Pass
103	The quality of the final product should be affected by the quality of the advice that the Scrum Manager gives to their team.	Each advice that the player gives to their team is credited with a specific numbers of points.	Pass
104	Each advice should be credited with a specific number of points that correspond to the quality of the final product.	The quality of the final product depends on the advice that players provide during the Daily Scrum Meetings.	Pass
105	The trend line of the Burndown chart should depend on the quality of the product on that particular sprint.	The trend of the Burndown chart changes depending on the quality of the product on that sprint.	Pass
106	Administrators should be able to preview their query before adding it to the database.	Administrators se a preview of their query and then they can return back to add or modify it.	Pass

Table B.3: Functional requirements testing

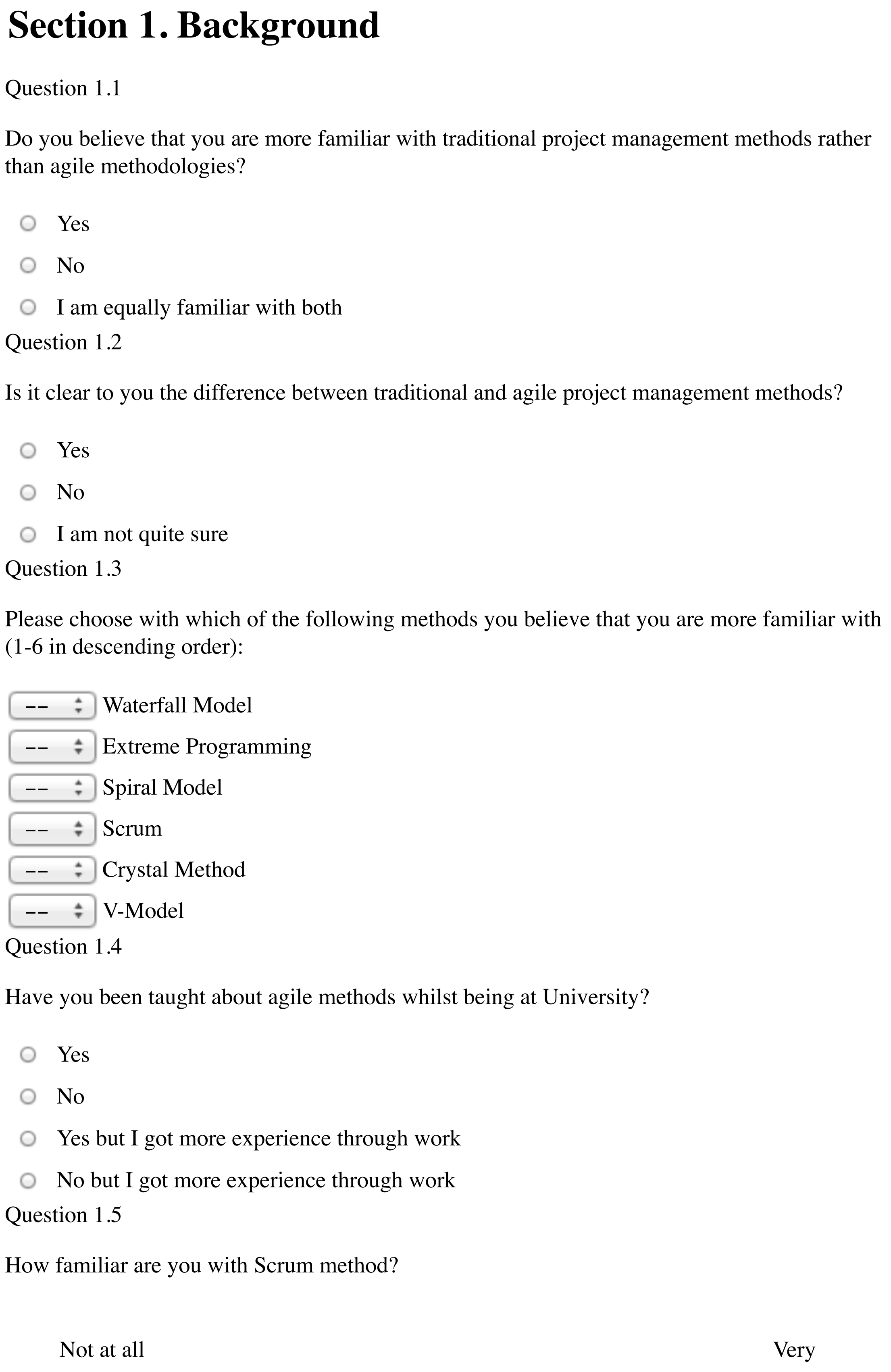
B.4 Non-Functional Requirements Testing

No	Test Case	Expected Outcome	Result
107	The system should be targeted to students of the IT field, as well as to project management literate students.	The system has explanatory text explaining all Scrum definitions.	Pass
108	The system needs to be fully functional with all major browsers.	The system is fully functional in Mozilla Firefox, Chrome, Safari.	Pass
109	No personal information concerning should be kept.	The system requires only a username and a password.	Pass
110	The database should be secure.	Sessions and prevention against SQL injection was used.	Pass
111	The system should be available at all times.	The system is stored on the ECS server.	Pass

No	Test Case	Expected Outcome	Result
112	The overall system should be secure and robust.	Sessions, prevention against SQL injection and functions to avoid coupling were used.	Pass
113	Player passwords should be store encrypted.	Passwords get encrypted through `sha1()` function.	Pass
114	The system should allow users to repeat their game.	A button "Start new game" should appear at the end of the game which redirects players to welcome.php.	Pass
115	The system should allow users to play with more than one project.	A button "Start new game" should appear at the end of the game which redirects players to welcome.php.	Pass
116	If users quit the game whilst playing, the next time they login the system should redirect them to the place where they quit the game.	The system checks the phase that the specific player is and redirects them to the appropriate page.	Pass
117	The system should be available from the Web.	Initial page is loaded.	Pass
118	The system should conform to the W3C CSS standards.	Testing againt CSS standards should pass.	Fail

Table B.4: Non-Functional requirements testing

Appendix C – Evaluation Questionnaire



Section 1. Background

Question 1.1

Do you believe that you are more familiar with traditional project management methods rather than agile methodologies?

○ Yes

○ No

○ I am equally familiar with both

Question 1.2

Is it clear to you the difference between traditional and agile project management methods?

○ Yes

○ No

○ I am not quite sure

Question 1.3

Please choose with which of the following methods you believe that you are more familiar with (1-6 in descending order):

[-- ⬍] Waterfall Model

[-- ⬍] Extreme Programming

[-- ⬍] Spiral Model

[-- ⬍] Scrum

[-- ⬍] Crystal Method

[-- ⬍] V-Model

Question 1.4

Have you been taught about agile methods whilst being at University?

○ Yes

○ No

○ Yes but I got more experience through work

○ No but I got more experience through work

Question 1.5

How familiar are you with Scrum method?

Not at all Very

Question 1.6

Would you be interested to learn more about Scrum method?

○ Yes

○ No

Question 1.7

Do you think that your modules at University focus more on:

○ Agile methods

○ Traditional management methods

○ Both

Section 2. Game Usability

Question 2.1

Do you think that too much prior knowledge was assumed in this game?

I strongly disagree	I strongly agree

Question 2.2

The user interaction with the system was smooth

I strongly disagree	I strongly agree

Question 2.3

Even though the system is a prototype, did you find the game enjoyable to play?

I strongly disagree	I strongly agree

Question 2.4

Do you want to add any comments concerning the user interface of the game?

Question 2.5

Did you encounter any problems whilst playing the game?

Section 3. Learning

Question 3.1

Did this game help you learn more about Scrum?

○ Yes

○ No

○ It's the same

Question 3.2

Did you find the game difficult to play?

○ Yes

○ No

Question 3.3

Do you think that this game can potentially be used as supplementary material in parallel with lectures?

○ Yes

○ No

○ I don't know

Question 3.4

Do you think that if this game was part of your Project Management module you would get a better understanding concerning Scrum method?

○ Yes

○ No

○ Not sure

Section 4. System

Question 4.1

How would you rate the overall system?

Very poor Excellent

Question 4.2

Would you like to add any more comments about the game?

Appendix D – Questionnaire Results

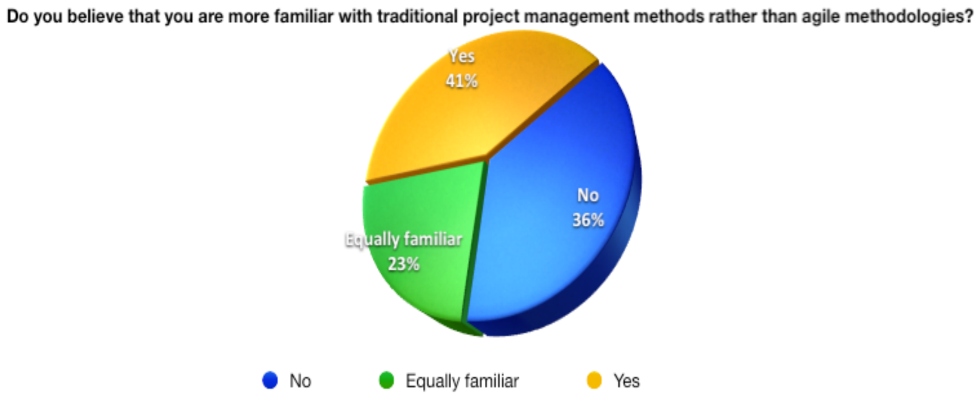

Do you believe that you are more familiar with traditional project management methods rather than agile methodologies?

Figure D.1: Question 1

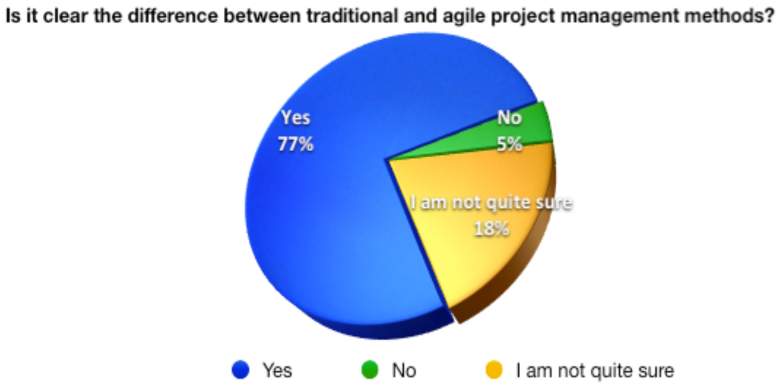
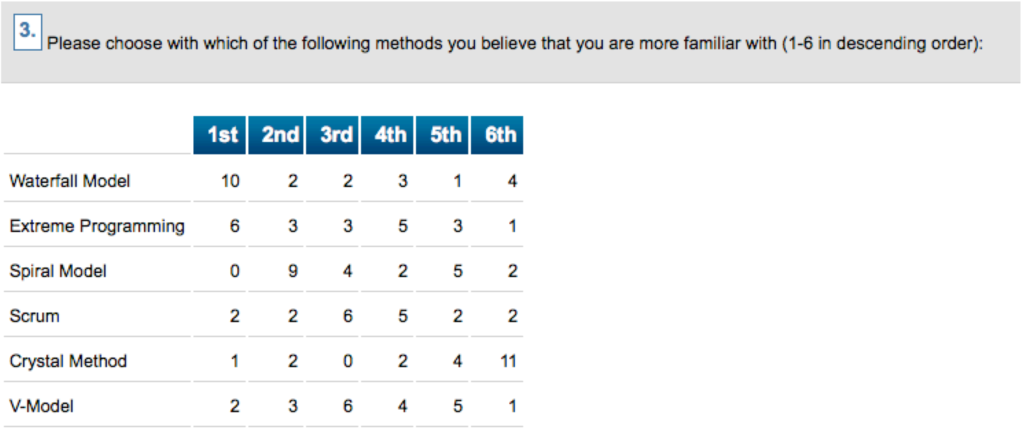

Is it clear the difference between traditional and agile project management methods?

Figure D.2: Question 2


3. Please choose with which of the following methods you believe that you are more familiar with (1-6 in descending order):

	1st	2nd	3rd	4th	5th	6th
Waterfall Model	10	2	2	3	1	4
Extreme Programming	6	3	3	5	3	1
Spiral Model	0	9	4	2	5	2
Scrum	2	2	6	5	2	2
Crystal Method	1	2	0	2	4	11
V-Model	2	3	6	4	5	1

Figure D.3: Question 3

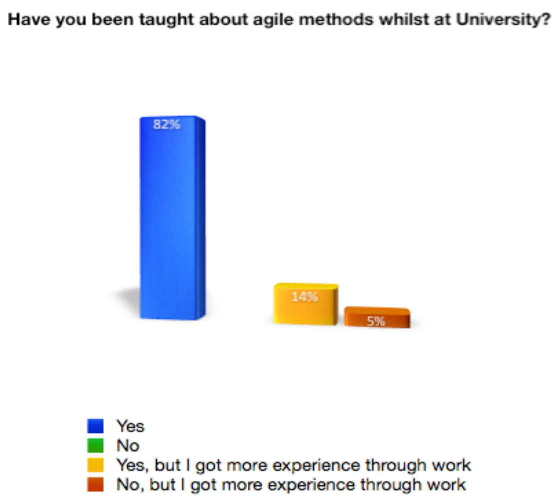


Figure D.4: Question 4

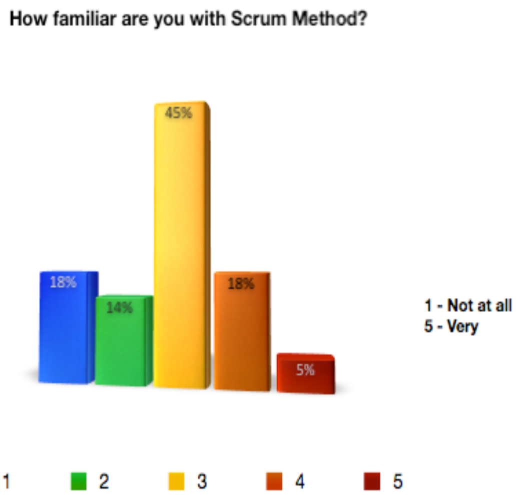


Figure D.5: Question 5

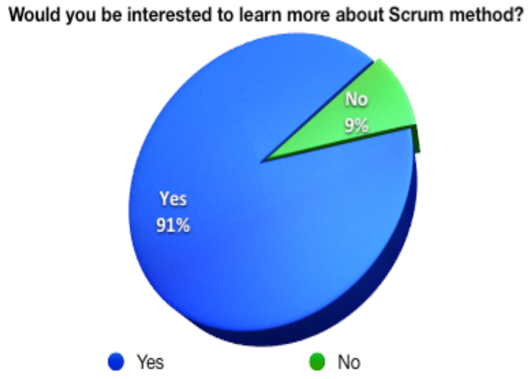


Figure D.6: Question 6

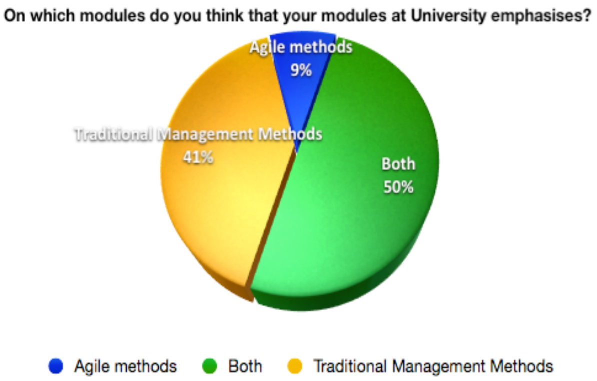


Figure D.7: Question 7

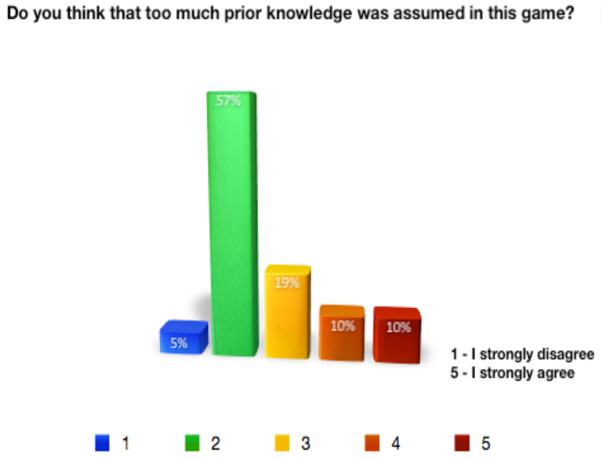


Figure D.8: Question 8

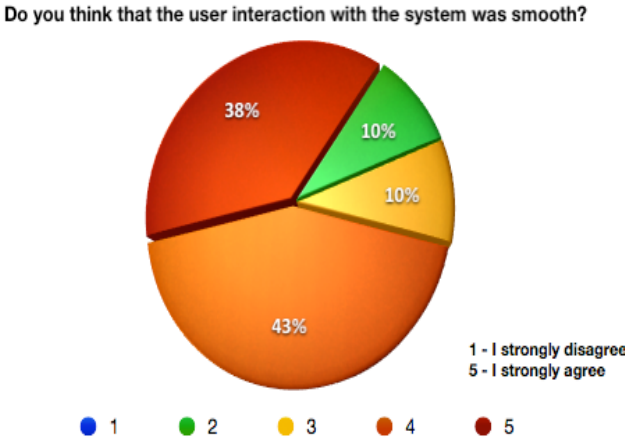


Figure D.9: Question 9

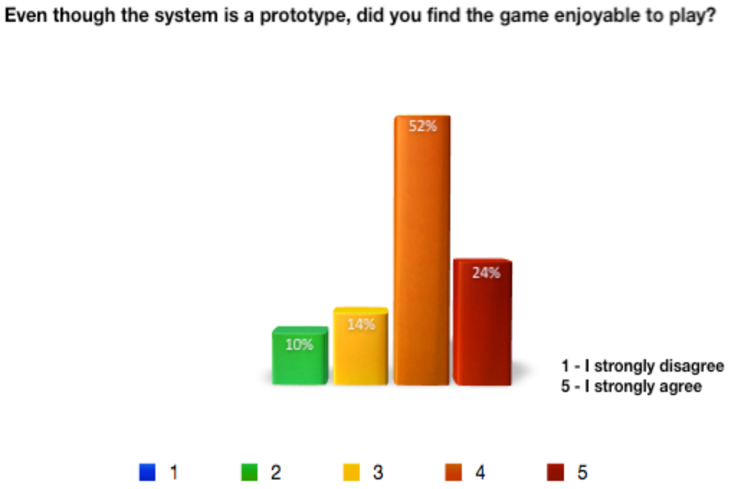
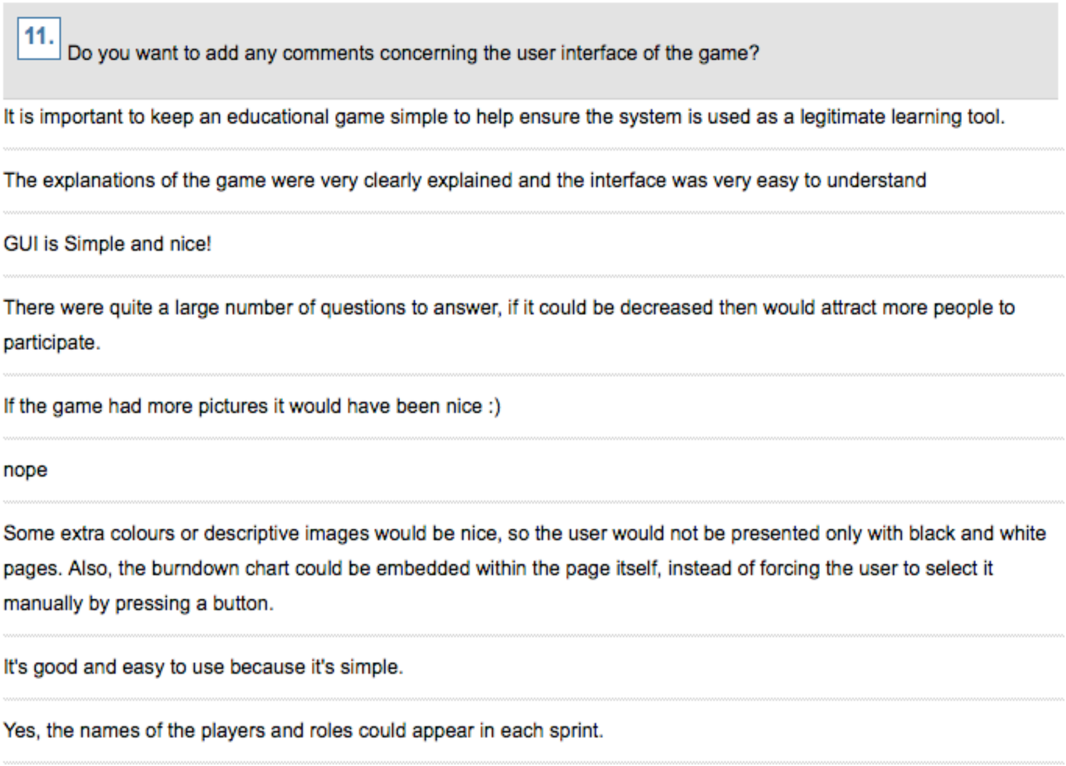


Figure D.10: Question 10



11. Do you want to add any comments concerning the user interface of the game?

It is important to keep an educational game simple to help ensure the system is used as a legitimate learning tool.

The explanations of the game were very clearly explained and the interface was very easy to understand

GUI is Simple and nice!

There were quite a large number of questions to answer, if it could be decreased then would attract more people to participate.

If the game had more pictures it would have been nice :)

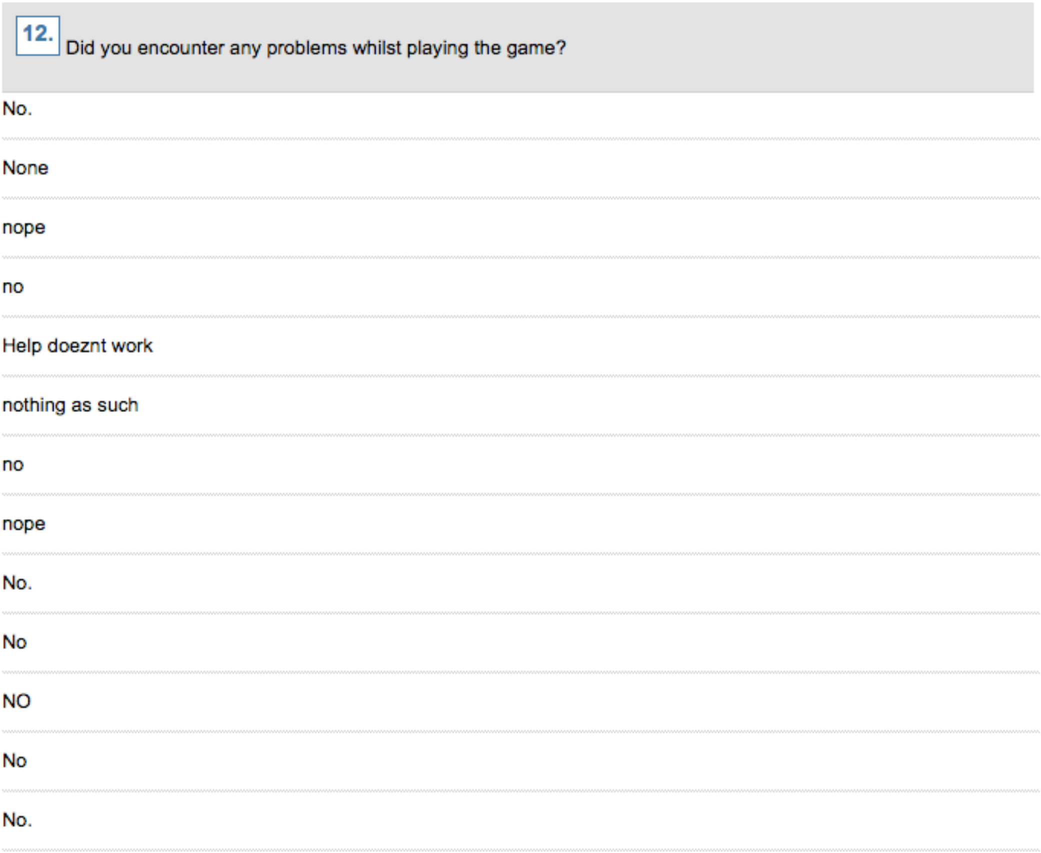
nope

Some extra colours or descriptive images would be nice, so the user would not be presented only with black and white pages. Also, the burndown chart could be embedded within the page itself, instead of forcing the user to select it manually by pressing a button.

It's good and easy to use because it's simple.

Yes, the names of the players and roles could appear in each sprint.

Figure D.11: Question 11



12. Did you encounter any problems whilst playing the game?

No.

None

nope

no

Help doeznt work

nothing as such

no

nope

No.

No

NO

No

No.

Figure D.12: Question 12

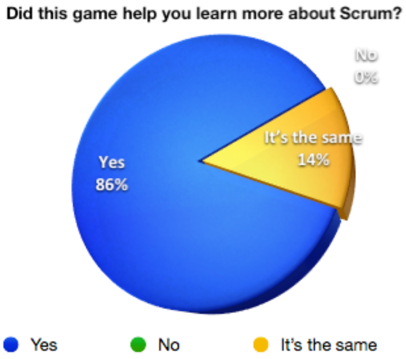


Figure D.13: Question 13

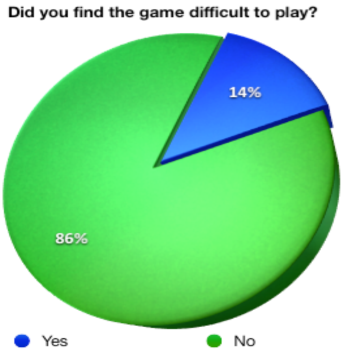


Figure D.14: Question 14

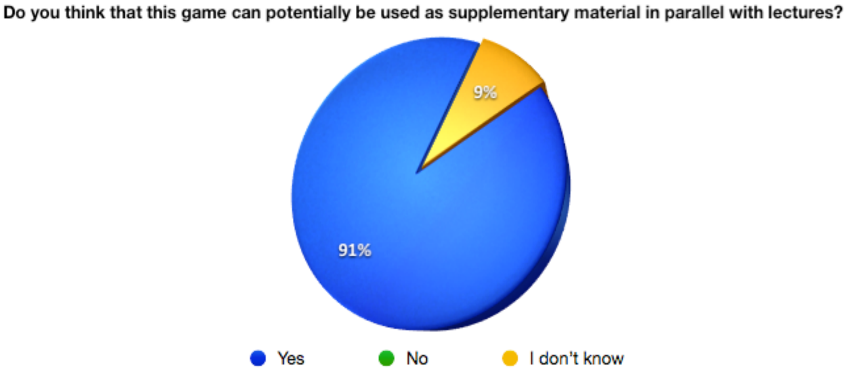


Figure D.15: Question 15

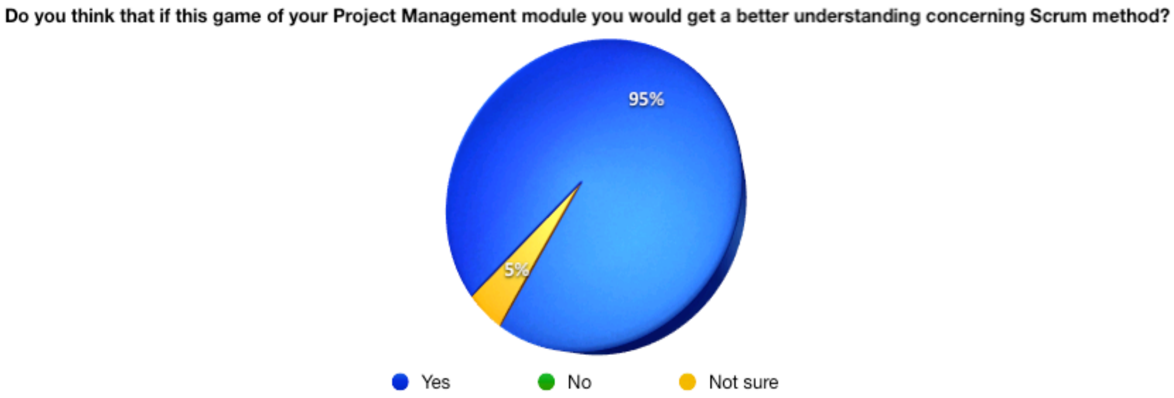


Figure D.16: Question 16

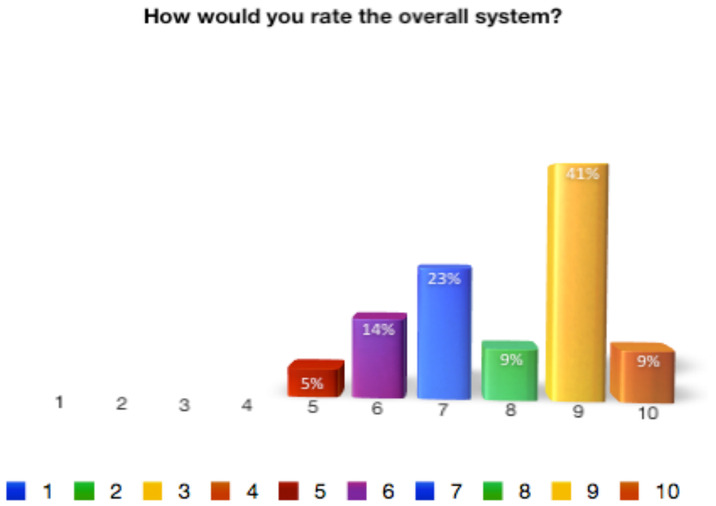
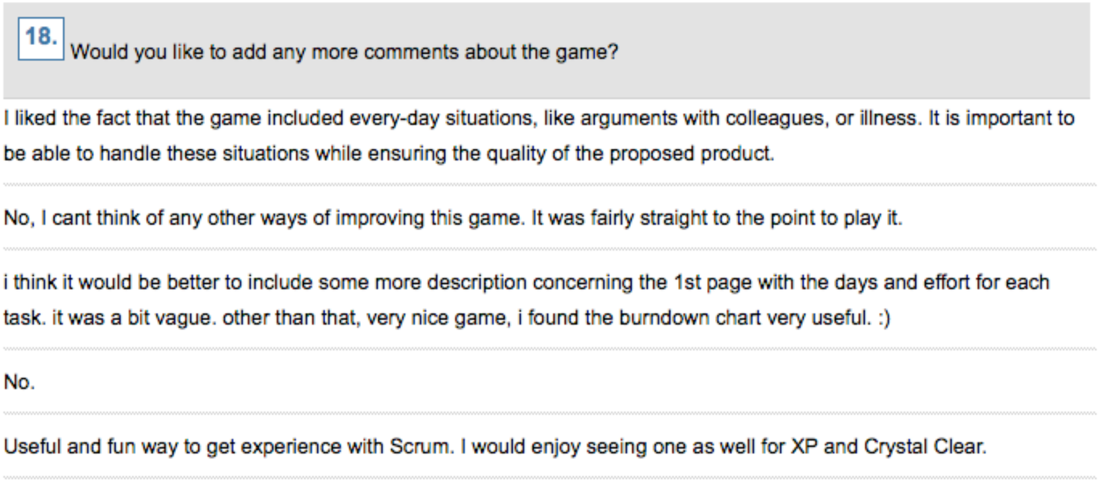


Figure D.17: Question 17



18. Would you like to add any more comments about the game?

I liked the fact that the game included every-day situations, like arguments with colleagues, or illness. It is important to be able to handle these situations while ensuring the quality of the proposed product.

No, I cant think of any other ways of improving this game. It was fairly straight to the point to play it.

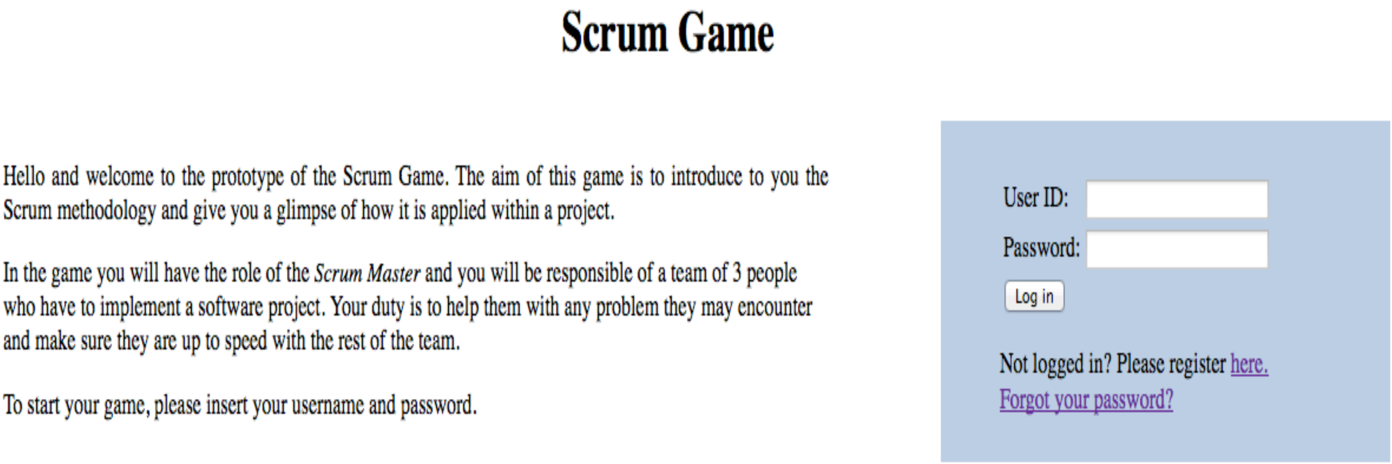
i think it would be better to include some more description concerning the 1st page with the days and effort for each task. it was a bit vague. other than that, very nice game, i found the burndown chart very useful. :)

No.

Useful and fun way to get experience with Scrum. I would enjoy seeing one as well for XP and Crystal Clear.

Figure D.18: Question 18

Appendix E – System Screen shots



Scrum Game

Hello and welcome to the prototype of the Scrum Game. The aim of this game is to introduce to you the Scrum methodology and give you a glimpse of how it is applied within a project.

In the game you will have the role of the *Scrum Master* and you will be responsible of a team of 3 people who have to implement a software project. Your duty is to help them with any problem they may encounter and make sure they are up to speed with the rest of the team.

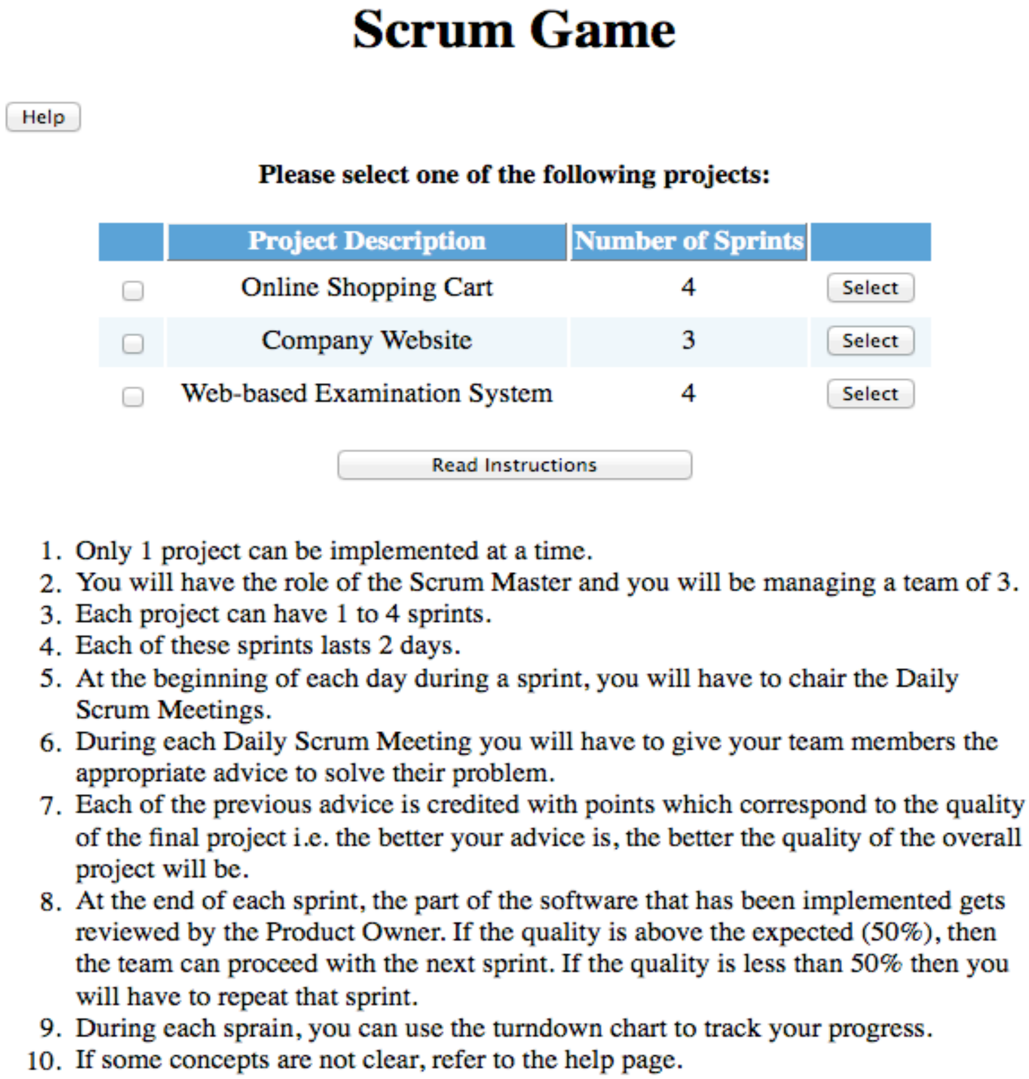
To start your game, please insert your username and password.

User ID:

Password:

Not logged in? Please register here.
Forgot your password?

Figure E.1: index.php



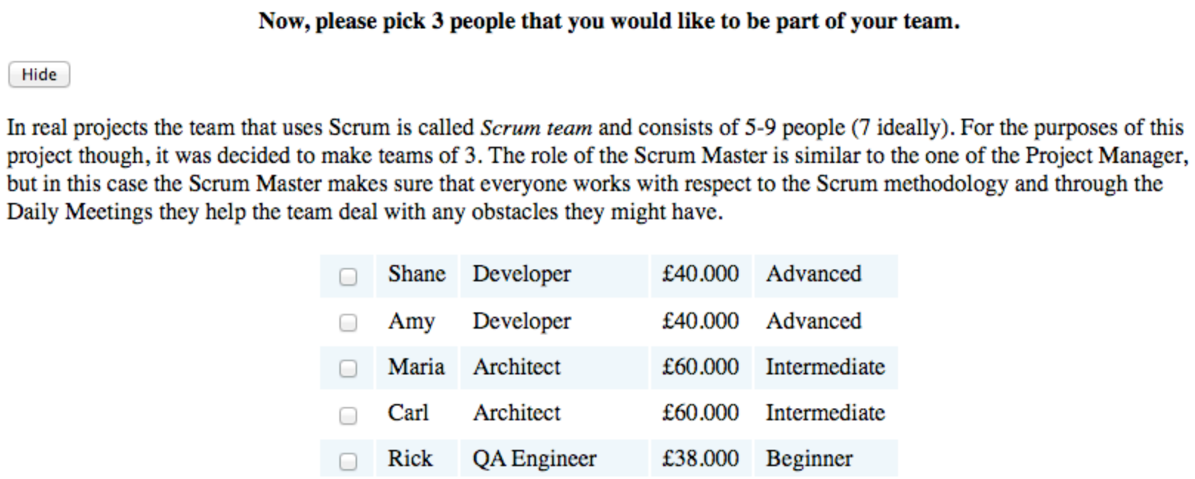
Scrum Game

Please select one of the following projects:

	Project Description	Number of Sprints	
□	Online Shopping Cart	4	Select
□	Company Website	3	Select
□	Web-based Examination System	4	Select

1. Only 1 project can be implemented at a time.
2. You will have the role of the Scrum Master and you will be managing a team of 3.
3. Each project can have 1 to 4 sprints.
4. Each of these sprints lasts 2 days.
5. At the beginning of each day during a sprint, you will have to chair the Daily Scrum Meetings.
6. During each Daily Scrum Meeting you will have to give your team members the appropriate advice to solve their problem.
7. Each of the previous advice is credited with points which correspond to the quality of the final project i.e. the better your advice is, the better the quality of the overall project will be.
8. At the end of each sprint, the part of the software that has been implemented gets reviewed by the Product Owner. If the quality is above the expected (50%), then the team can proceed with the next sprint. If the quality is less than 50% then you will have to repeat that sprint.
9. During each sprain, you can use the turndown chart to track your progress.
10. If some concepts are not clear, refer to the help page.

Figure E.2: welcome.php

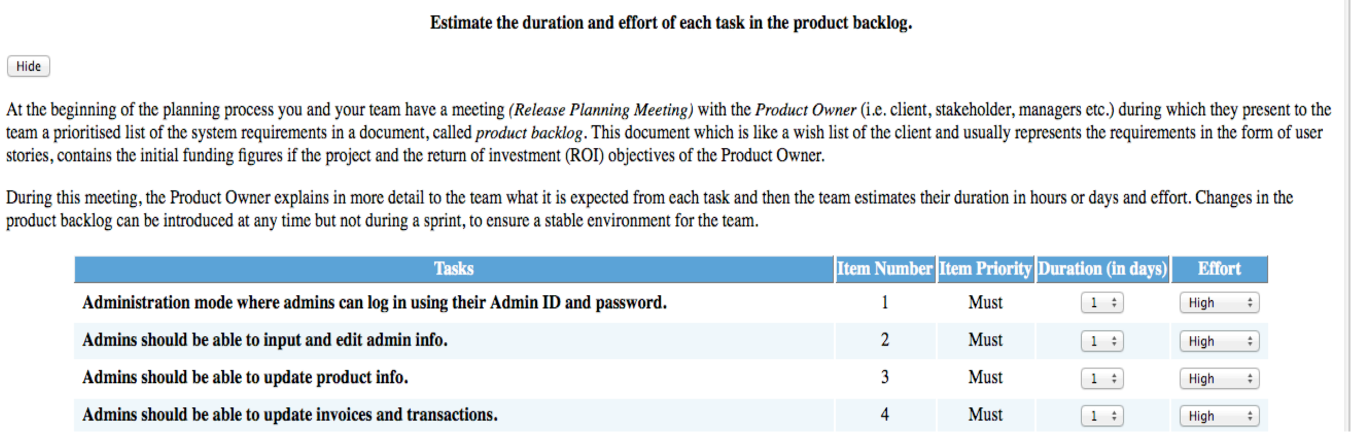


Now, please pick 3 people that you would like to be part of your team.

In real projects the team that uses Scrum is called *Scrum team* and consists of 5-9 people (7 ideally). For the purposes of this project though, it was decided to make teams of 3. The role of the Scrum Master is similar to the one of the Project Manager, but in this case the Scrum Master makes sure that everyone works with respect to the Scrum methodology and through the Daily Meetings they help the team deal with any obstacles they might have.

□	Shane	Developer	£40.000	Advanced
□	Amy	Developer	£40.000	Advanced
□	Maria	Architect	£60.000	Intermediate
□	Carl	Architect	£60.000	Intermediate
□	Rick	QA Engineer	£38.000	Beginner

Figure E.3: team.php



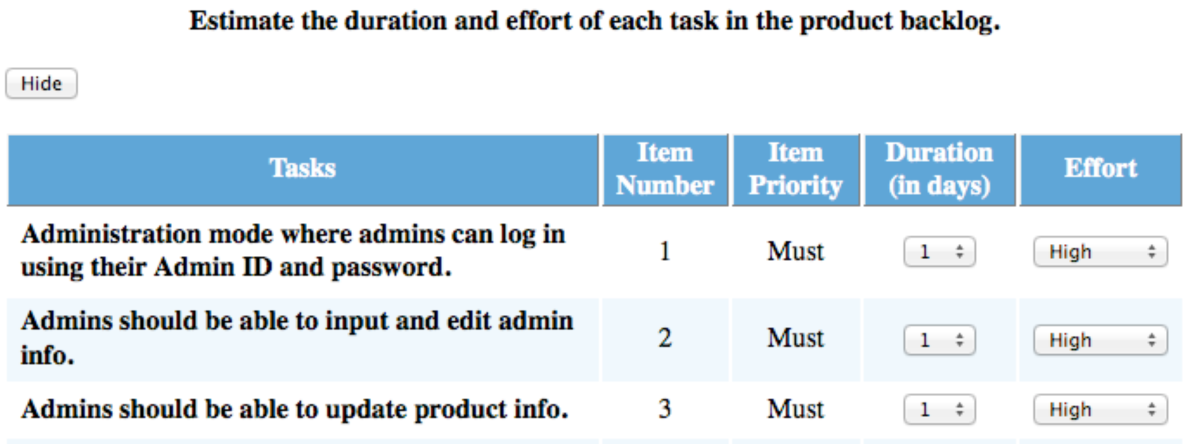
Estimate the duration and effort of each task in the product backlog.

At the beginning of the planning process you and your team have a meeting *(Release Planning Meeting)* with the *Product Owner* (i.e. client, stakeholder, managers etc.) during which they present to the team a prioritised list of the system requirements in a document, called *product backlog*. This document which is like a wish list of the client and usually represents the requirements in the form of user stories, contains the initial funding figures if the project and the return of investment (ROI) objectives of the Product Owner.

During this meeting, the Product Owner explains in more detail to the team what it is expected from each task and then the team estimates their duration in hours or days and effort. Changes in the product backlog can be introduced at any time but not during a sprint, to ensure a stable environment for the team.

Tasks	Item Number	Item Priority	Duration (in days)	Effort
Administration mode where admins can log in using their Admin ID and password.	1	Must	1	High
Admins should be able to input and edit admin info.	2	Must	1	High
Admins should be able to update product info.	3	Must	1	High
Admins should be able to update invoices and transactions.	4	Must	1	High

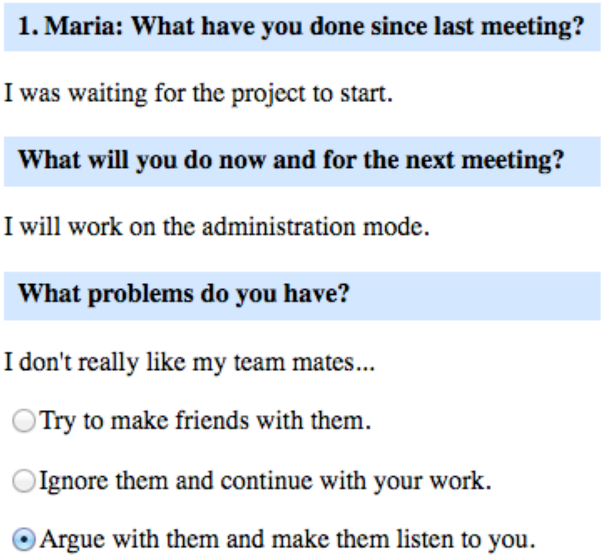
Figure E.4: product_backlog.php



Estimate the duration and effort of each task in the product backlog.

Tasks	Item Number	Item Priority	Duration (in days)	Effort
Administration mode where admins can log in using their Admin ID and password.	1	Must	1	High
Admins should be able to input and edit admin info.	2	Must	1	High
Admins should be able to update product info.	3	Must	1	High

Figure E.5: sprint_backlog.php



1. Maria: What have you done since last meeting?

I was waiting for the project to start.

What will you do now and for the next meeting?

I will work on the administration mode.

What problems do you have?

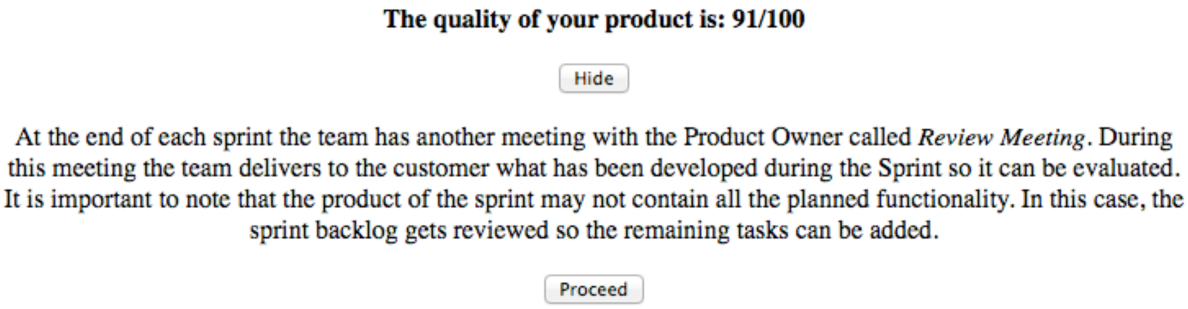
I don't really like my team mates...

○ Try to make friends with them.

○ Ignore them and continue with your work.

● Argue with them and make them listen to you.

Figure E.6: sprint.php



The quality of your product is: 91/100

At the end of each sprint the team has another meeting with the Product Owner called *Review Meeting*. During this meeting the team delivers to the customer what has been developed during the Sprint so it can be evaluated. It is important to note that the product of the sprint may not contain all the planned functionality. In this case, the sprint backlog gets reviewed so the remaining tasks can be added.

Figure E.7: review_meeting.php

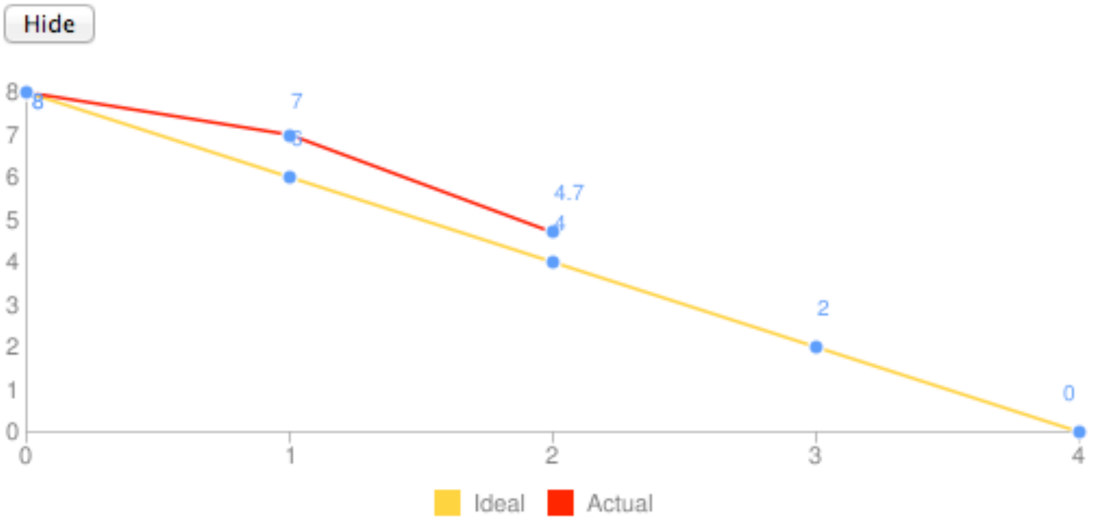
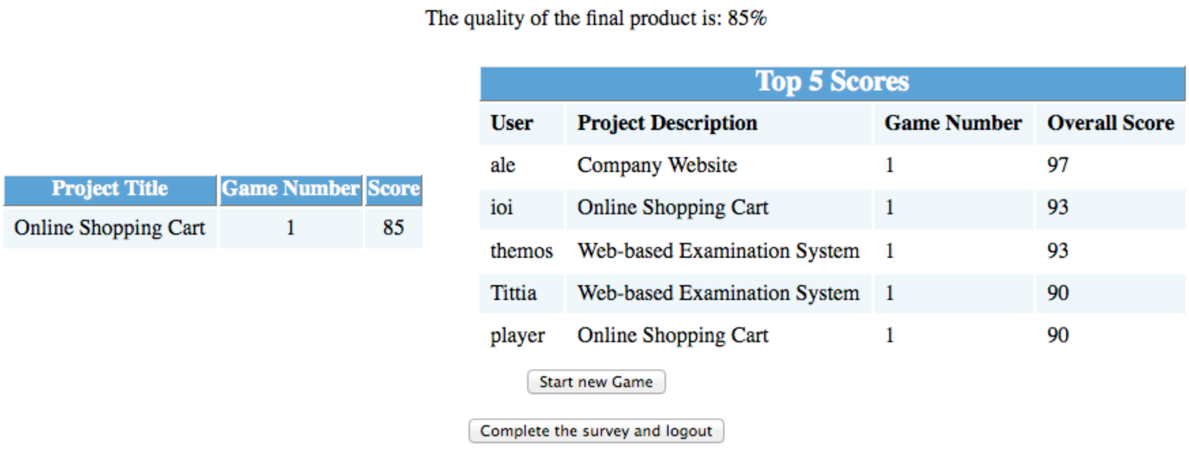


Figure E.8: Burndown chart

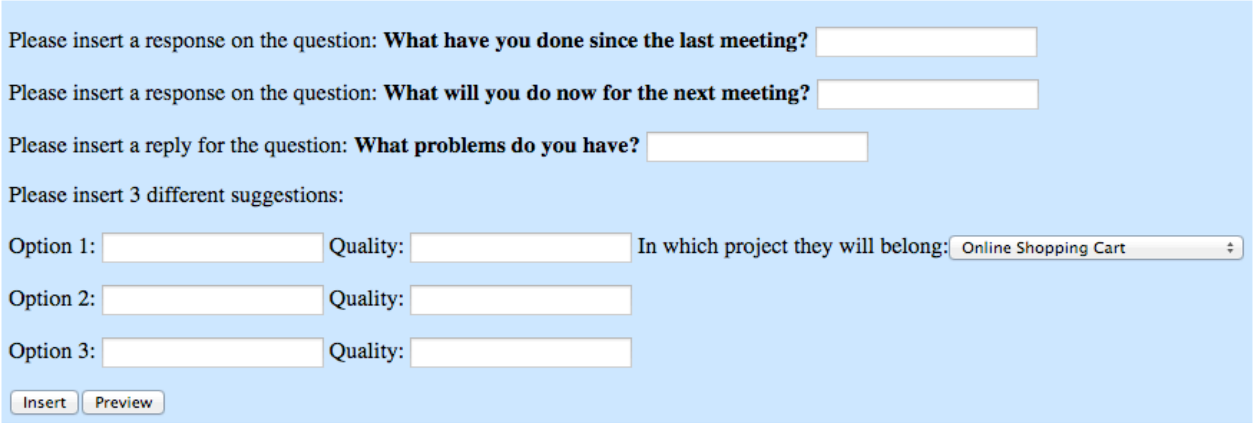


The quality of the final product is: 85%

Project Title	Game Number	Score
Online Shopping Cart	1	85

Top 5 Scores			
User	**Project Description**	**Game Number**	**Overall Score**
ale	Company Website	1	97
ioi	Online Shopping Cart	1	93
themos	Web-based Examination System	1	93
Tittia	Web-based Examination System	1	90
player	Online Shopping Cart	1	90

Figure E.9: end.php



Please insert a response on the question: **What have you done since the last meeting?**

Please insert a response on the question: **What will you do now for the next meeting?**

Please insert a reply for the question: **What problems do you have?**

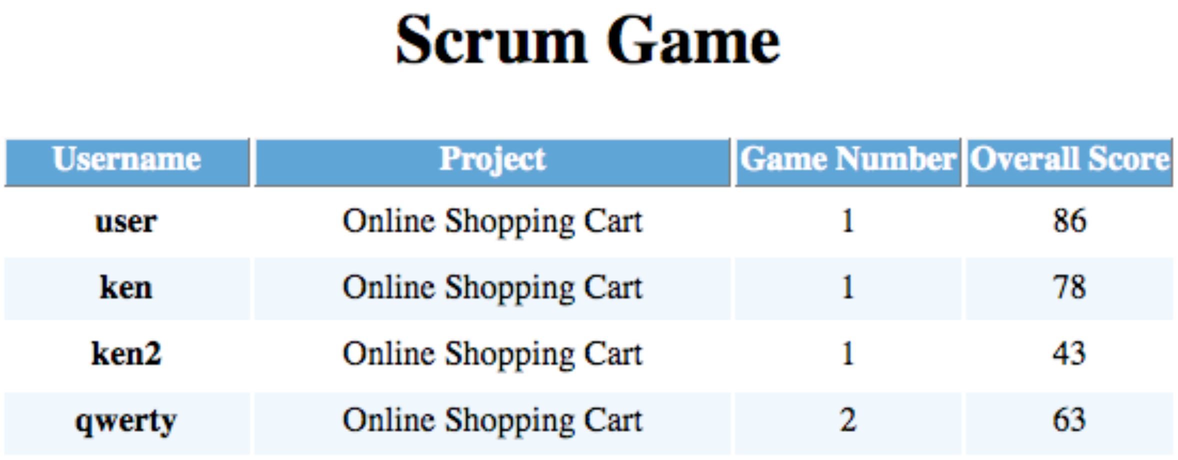
Please insert 3 different suggestions:

Option 1: Quality: In which project they will belong: Online Shopping Cart

Option 2: Quality:

Option 3: Quality:

Figure E.10: admin.php



Scrum Game

Username	Project	Game Number	Overall Score
user	Online Shopping Cart	1	86
ken	Online Shopping Cart	1	78
ken2	Online Shopping Cart	1	43
qwerty	Online Shopping Cart	2	63

Figure E.11: log.php