Harrisburg University of Science and Technology
Digital Commons at Harrisburg University

Dissertations and Theses

Project Management (PMGT)

8-2016

Scrum Sim - A Simulation Game to Learn the Scrum Agile Framework

Anshuman Bassi
Harrisburg University of Science and Technology

Follow this and additional works at: http://digitalcommons.harrisburgu.edu/pmgt_dandt

Part of the Management Information Systems Commons, and the Management Sciences and Quantitative Methods Commons

Recommended Citation

Bassi, A. (2016). *Scrum Sim - A Simulation Game to Learn the Scrum Agile Framework*. Retrieved from http://digitalcommons.harrisburgu.edu/pmgt_dandt/11

This Thesis is brought to you for free and open access by the Project Management (PMGT) at Digital Commons at Harrisburg University. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of Digital Commons at Harrisburg University. For more information, please contact drunyon@harrisburgu.edu, ereed@harrisburgu.edu.

Scrum Sim - A Simulation Game to Learn the Scrum Agile Framework

Anshuman Bassi

Harrisburg University

GRAD 699-PMGT

Project Supervisor: Dr. Thomas Sheives

Abstract

In recent times, organizations have been attempting to shift from Traditional project management methodologies such as waterfall model to incorporate agile methodologies such as scrum for their project development. The main features of these agile methods include incremental and iterative delivery that allows inclusion of the requirement changes at any stage of the project lifecycle and the most commonly used agile method is scrum. As are result of wide usage of agile mythology in the IT industry, scrum framework is now being taught in various software engineering and project management courses. Even though scrum has been introduced in various academic courses, the constraints of time, scope and facilities restrict students from getting hands on experience on real world scenarios. Many employers believe that students graduating of the universities lack the desired skills and practical experience to adopt and implement agile method like scrum. Researchers have suggested that use of simulations or games if used in complement to the traditional class room setting can enhance students learning. During our previous research on the existing software educational games, it was found that there is availability of many games and simulations based on traditional software methodologies i.e. sequential software process but a lesser number of games or simulations are available on agile methodology especially based on scrum. Also the scrum games that are available are mainly card games, board games or toy games such as "Lego Bricks" which have their own limitations. To address this problem, we have developed Scrum Sim a computer based game that simulates scrum process for a software development project. It provides a glimpse of the scrum lifecycle and represents various aspects of scrum framework in a team for a software development project. We present the game, its rules, game play and discuss its validation or evaluation results.

Keywords: Simulation and Games, Scrum simulation, educational games for software engineering, software project management through simulation and games, scrum through simulations and games.

Table of Contents

Preface

This paper describes the scrum game named Scrum Sim that was developed as a result of our research on existing agile and traditional project management based simulations and games that can be used for educational purpose. The paper also talks about the existing scrum games and the reasons for developing and designing a new agile simulation game.

First of all, I would like to express my gratitude to my supervisor Dr. Thomas Sheives, this game could not have been developed without his guidance. Despite of all difficulties and problem during the course of my research, he was always there to help me with his valuable advice, feedback and suggestions. Secondly I would like to thank Dr. Richard Kordel for his guidance and patience which helped me to develop my research skills during Grad 695 course .Finally I would like to thank all the people who participated in the game validation process and provided their feedback by participating in the survey, without them, I would not have been able to complete this paper.

Introduction

In the recent years, agile is being used as one of the main methodology for software development project in the IT industry and the most commonly used agile methodology in the industry is Scrum (Rodriguez, Soria & Campo, 2015). Traditional methods follow a sequential flow of activities and are thus considered bureaucratic whereas agile methods especially scrum involves little bureaucracy. Scrum is designed for small and self-organizing project teams and allows the development of software on incremental basis i.e. a deliverable software is available at the end of each iteration (Fernandes & Sousa, 2010). Therefore the knowledge of scrum framework is an essential ingredient for the company to ensure successful completion of these software projects. The wide usage of scrum on software projects has also highlighted the gap between the skills required by the software industry vs the skills acquired by the students in an academic setting (Rodriguez, Soria & Campo, 2015).

Because of the growing usage of scrum in the industry, academic institutions started incorporating the agile methodologies in their course curriculums but still the students coming out of these universities still lacks the real world preparedness and industry as a general remains unsatisfied (Fernandes & Sousa, 2010). The root cause of the problem can be related to the way these concepts are taught: students are acquainted to theories and concepts in a classroom setting in a series of lectures and students are required to complete assignments on the subject or at the most complete a small project/case study to put forward their knowledge skills into practice ((Baker, Navarro, & Van Der Hoek, 2005). Therefore due lack of practical experience and understanding, hiring employers are not satisfied even after students pursue these various academic courses as the gap still exists (Fernandes & Sousa, 2010).

To bridge the gap, various approaches have suggested to teach software development through capstone projects. A capstone project is cooperative assignment that allows students to apply their skill and knowledge in a pseudo-real work experience (Rodriguez, Soria & Campo, 2015). Mahnic, 2015 suggested teaching scrum through capstone project. In his research he prescribed a detailed course structure, perception of students, and teacher’s observations after the course. Although using approaches like capstone project to apply knowledge of scrum does play a part in providing practical knowledge to the students but they somehow ignore the academic constrains such as limited teachers, physical space, time and multiple teams working at a given point of time (Baker et al, 2005; Rodriguez et al. 2015).

To overcome these constraints an alternate approach of teaching software processes through simulation and games has been suggested by various researchers that enables a more realistic view of practical experience to the students (Rodriguez, Soria & Campo, 2015). The best example of this could be a Pilot who learns flying by using a flight simulator which allows the pilot to get hands on flying experience while attending classroom lectures and listening to the instructors in parallel. Within this context, flight simulators have become critical tools used to supplement pilot training. Similarly games and simulation can become critical tools through which students can participate virtually and experience real world project scenario. This would enable them to enhance their project management skills by allowing them to learn the complexity of real projects by collaborating with other people, dealing with different situations, planning and being responsible (Baker et al 2005).Using games in supplement to the class room lectures not only provides an opportunity to the students to simulate the real project situations in a fun and easy way but also allows them to mimic a role and thereby improve the quality of learning. According to Fernandes and Sousa, 2010, "Various authors have also suggested that, when learning using games, students can acquire what is taught, putting aside their fears and anxieties and devoting themselves to the games".

This paper also presents Scrum Sim a simulation game that mimics the scrum process for a software development project and can be played on any open source tool that supports scrum. The simulation game has been developed to help students to provide a better understanding of the scrum lifecycle. The game basically aims at providing a thorough understanding of the concepts, events and roles of the scrum process to software engineering or project management students who have the basic knowledge of scrum. This simulation game is result of a research on the various existing simulation and games based on traditional and agile software development methodologies. Scrum Sim has been designed to be used in conjunction with class room lectures.

The rest of the paper is organized as follows: The next sections would describe about the Scrum Sim simulation game and game play followed by the literature research section. The method section would describe the basic approach for developing this paper. The result section shows the game validation results after collating the feedback from the students. Finally the last section would describe the conclusion of the paper as well recommendation on the future work.

Literature Review

The whole idea of Scrum Sim simulation game was triggered by the agile course that the author undertook for the graduate degree. During the course, the author simulated a dummy software development project using scrum framework which inspired him to develop a simulation game. To develop a scrum based game, it was necessary to investigate the literature on the existing educational games especially related to software development in order to differentiate it from the earlier research. The research suggested that most of the existing educational games are based on traditional i.e. sequential project management model but lesser number games are available that are based on the agile process (Bassi, 2016).Also the existing agile based games are mainly board games and they too have their own limitations. Some of the other toy games based on scrum are not considered to be useful by teachers for student's skill development (Kurkovsky, 2015).

The research was being performed to understand whether creating a scrum based game would be beneficial for the students, what kind of educational games are being currently used and on what guidelines a new scrum based game should be designed. The main questions that we researched were the following:

a) What are the effects of educational simulations and games?

b) Which are the existing games that are being used for educational purpose with respect to software development?

c) What are the best feature and practices in the creation of existing simulations and games?

d) How they can be incorporated in designing a scrum simulation?

e) What are the critical points that needs to be stressed in such simulations and games?

f) What are the limitation of the existing scrum games and simulations?

To find answer to the above questions, research was performed by analyzing 26 papers that discussed about the studies done to evaluate the efficacy of simulation & games on student's learning, presented existing simulation and games based on tradition project management and agile methods, Presented comparison between the existing games. Summarized results of the literature review is presented in the following sections:

1) Analyzing effects of Simulation and games

Vogel et al. (2006) in their study on computer gaming and simulation for learning found that the working with games and simulation resulted in a significant gain in learning for the students as compared to use of only traditional methods of teaching. They analyzed the data collected from 32 studies on the widely available games and simulations. Most of these studies were evaluated quantitatively but had different methodologies and reporting techniques. Although the study suggested that using games or simulation enhance the learning outcome for the students as their usage increases the understanding and engagement of the students, the only limitation of the study could be the sample of the study i.e. number of papers selected for research were less as lot of the articles were rejected due to pitfalls in reporting and methodology used.

In another study, De Smale et al. (2015) observed a positive effect of simulations and games on the achievement learning objectives in tertiary education. Tertiary education refers to college education in Australia. The authors in their research could only find 29 articles out 64 that basically talked about the impact of games and simulations on the learning objectives. But out of the 29 articles that were chosen for the study, none of the article established a negative relationship. There were only 3 articles which showed neutral results. De Smale et al. (2015) also talked about the learning effectiveness enabling factors in the context of simulation/games such as role of the instructor, Specificity of the game, and course integration. They also pointed out the need of univocal framework for evaluation of games/simulations based on the classification of learning objectives, clear typology and evaluations i.e. quantitative and qualitative.

Kropp, Meier, Mateescu and Zahn (2014) in their study proposed that simulations and games are the best ways that can be used for teaching and learning agile collaboration. In their study, they introduced an agile simulation game name Scrum Lego city in software engineering course at a bachelor's degree program. During the course the students, the student's built a city of Lego bricks using scrum process. The simulation was part of the six week agile course where classroom sessions were conducted thrice a week and rest of the time students practiced the simulation. The instructor played the role of a scrum coach who observed, provided input and suggestions to the students. The evaluation of the game supported the author's view as students in their feedback reported that they liked participating in the simulation and provided positive feedbacks on the simulation in comparison with the traditional teaching methods.

In the study performed Connolly, Stansfield and Hainey (2007) on game based learning with in software engineering analyzed the following four matured game that are used for educational purpose. :
1) KM QUEST: in the area of knowledge management (Leemkuil and Hoog, 2005).
2) Open Software Solutions – In the area of interactive multimedia simulation of a software house (sharp and Hall, 2000).
3) The Incredible Manager- In the area of software project management (Dantas, Barros & Werner, 2004, 2005).

4) Sim SE – In the area of software engineering simulation (Navarro et al, 2004).

They in their research highlighted 2 major points after analyzing these games:
1) Empirical studies are lacking on game base learning.
2) They found a limited impact of the above mentioned simulation/game on students learning.
Therefore they proposed a participatory approach and still designed a new game based learning application to teach requirements collection, analysis, and design and project management. The proposed game design also included a post-game utility analysis tool to collect empirical data.

Mahnic, (2015) in his study emphasizes on the widespread approach of teaching scrum to students via educational game. Although in his paper, he talks mainly about the framework for designing a practical project work but also states that educational simulations/games can be an alternate to the project work and can be more applicable especially where course curriculums do not have enough time or have other constraints to allow practical project work. Thus team games and simulation could be an ideal possible solution for the students. His study on the educational games was based on following game:
1) LEGO based scrum simulation proposed by Paasivaara, Heikkilä, Lassenius and Toivola, (2014).
2) SCRUMIA- by Wangenheim, Savi, and Borgatto (2013).
3) Play scrum- by Fernandes and Sousa (2010).
4) Plasticine Scrum- by Ramingwong and Ramingwong (2015) "describes a variation of the game developed by (Krivitsky, 2011)
Mahnic, (2015) also made suggestion on the ambiguity faced by the instructors on the issue of defining various scrum roles in creating scrum games and simulations. The same is discussed in one the following sections.

2) Existing educational Simulation and Games

To find out the characteristics of a simulation or a game, we performed a research on the existing educational games and simulation related to software engineering. During the research, games based on both traditional and agile project management were addressed to get a better understanding on the subject matter. This was done to get an insight into the existing educational games in the field of software engineering as well as to identify the features to be adapted in an educational game or simulation in order to achieve its learning objectives. The author found number of games based on traditional project management as well as based on scrum. A description of some of the games is given below:

2.1) Traditional project management based games

a) SESAM- (Software Engineering simulation by Animation Methods) is game based on traditional project management principles was designed by Drappa and Ludewig,(2000).In this game, the player plays the role of a project manager and mimics a real life project for 2 hours by interacting with the system using a text interface. The entire game is based on a quality assurance model as the game mainly emphasizes on the quality side of the software development project. In the game the player needs to take decisions by evaluating the virtual messages received from the employees in the form of statements. The goal of the player is to manage its resources efficiently which can be achieved by asking the team members to complete the tasks as well as by employing or dismissing a team members. The player's score is calculated on the basis of internal variables. Each player can get a feedback on his own performance by using a special tool available in the simulation There game is based on certain important assumptions:

1) The outcome of a software project is highly impacted by the qualification and experience of the developers.

2) Efforts and duration spent on development activities are interdependent.

3) Reviews are important as they help the team to find errors in the project at an early stage.

4) Verified documents should be rectified immediately.

The evaluation of the game also suggested that player's knowledge improved on the subject matter due to the simulation and the game can be useful tool for simulation of software projects.

b) **SIMSE:** is another single player fully graphical virtual game based on traditional project management i.e. waterfall aimed at university level students. In the game a player simulates the role of project manager and handles a virtual team of developers. The e game allows the player to hire or fire team members, observe the project progress and provide appropriate tools for the team. He can access the all the important information required to manage the project i.e. information related to the employees, task development, Customer satisfaction levels, budget details and deadlines. At the end of game a player is scored on how effectively he is able to manage, supports and handles his team members. The game also has the functionality to provide feedback to the players on the score they received (Baker, Navarro, & Van Der Hoek, 2005; Navarro & van der Hoek, 2004). The game can also be customized by the instructor as it allows instructors to create a customized simulation for a particular software engineering process as well as can provide a comprehensive view. The validation process of the game suggested that the game can complement the traditional methods of teaching and can add value to a software engineering course. The results also indicated that game contributed in a positive manner by allowing the player to practice their knowledge and skills and thus facilitating a better understanding of the software engineering processes.

c) SIMSOFT: is a board game (Figure A&B- Simsoft Board and Dashboard, refer to the appendix **Page 43 &44**) based on traditional software project management. The players are required to complete the project with in the stipulated time and budget constraints along with a team to be managed. The game also has a java based dashboard that facilitates players in analyzing their latest situation by providing them with components such as reports, messages and other information. The board also depicts the game flow and helps the players to determine and plan their further moves accordingly. The other components such as plastic counters depicts the project team members, Poker chips represents the team budget and can be used to hire more staff. During the game play, various events impact the project and decisions are to be taken accordingly (e.g. whether to hire/reduce the manpower or review working hours). Each team plays the games independently and there is no competition but the teams can see the performance of the other team (Caulfield et al, 2011 a, 2011 b). The game mainly concentrates towards the human resource aspect of the project management. The evaluation process of the game again indicated that game succeeded in presenting real world project scenarios to the students. The positive feedback from the students revealed that the students found the game to be interesting and was useful to work in teams provided clear instructions are provided to the students.

d) **Problem and Programmers:** is a card base competitive educational game based on waterfall model where players put the hats of a project manager and are required to complete a project at the earliest with minimum bugs in the code, within the given parameters of budget, quality and customer requirements. During the game, the simulated software project goes through various stages of waterfall model such as analysis, design development, integration and test .All the players are required to work on the same project according to the specified project budget, duration, expect quality and complexity.

The game requires 3 types of cards:

1) Concept cards- represents decision of the players.

2) Programmer cards- represents the employees or the team members.

c) Problem cards: represents the problem in the project and these cards can by used the Players against their opponents if the opponent meets its criteria.

The players can pick 5 cards and can then decide their course of action, based on the stage of the waterfall life cycle the particular player is in. Players also align their cards from left to right showing 5 different stages of waterfall from Requirement analysis to maintenance (Refer to the appendix Figure C: Phases of Problem and programmers, **page 45).** Even if the game follows a sequential software development model but it gives the choice to the players to choose and strategize about fulfilling their goals. In the game, some players will concentrate more on requirements, design and coding phase before staring the integration phase which means they take more time where as some players would rush through these phases and concentrate more on testing and implementation to deliver a project early. Therefore the game allows to identify the ramifications of pursuing different strategies and provides an experience close to the real world scenario (Fernandes & Sousa, 2010; Navarro et al, 2004). The evaluation of this game again revealed that simulations and games can add value to the software engineering course and can provide enriching practical experience to the students (Baker et al., 2003, 2005).

e) **Sim4Projetcs:** is a virtual project simulation based on traditional project management designed for students that are enrolled in the courses that concentrate on developing project management skills. The students can play the simulation individually or in teams. In the simulation students conceptually manage a project. The simulation includes scenarios that emphasize on project management challenges from various domains such as new product development, process conversion, construction and information technology development and implementation. The simulation is designed for students to provide them with real world experiences that are faced in project management and decision making. The simulation also offers students the opportunity to practice various parts of project management—a hands-on, learning-by-doing approach that emphasizes active learning. While playing the simulation, players take decisions regarding

- Employ or dismiss a resource.
- Train a resource to upgrade its skills
- When to start managerial actions to motivate resources.
- Assigning of resources to the tasks.

Based on the decision of the players, the simulation calculates the actual time and cost for each task. The simulation also interacts with MS project to create a file including the results from each round. The project Gantt chart is the basis of this simulation. There is no key to win the simulation successfully, the players need to focus on just analysis and thoughtful decision making assuming as if they are running an actual project.

Sim4Pojects is divided into two parts: pre-play and play. Pre-play involves resource bidding i.e. creation of virtual project team whereas play involves moving through the Gantt chart (from start node to the finish) that represents the project. The teams compete each other, period by period once the paly part begins in the simulation. The length of a period in the simulation is not measured in traditional units such as weeks or months. All teams must complete all of the tasks for each period and must reach the next milestone. In the simulation, the amount of time it takes for each team to accomplish this is a period. The length of time for a period is the amount of time between milestones on the Gantt chart. All teams start each period at the same point on the Gantt chart. It takes the teams differing amounts of time to complete the period tasks and that is the length of a period for that team (Pinto & Parente, 2004, 2009).

2.2) Scrum based games

a) **Play Scrum:** is a competitive card game based on the concept of the traditional project management game programs and programmers but it represents scrum methodology of software development. The game can be played by 2 to 5 players and each player acts as scrum master on a software development project. The game is divided into different rounds called sprints. The main components of play scrum are: (Refer to the appendix Figure d: examples of Play scrum cards, **page 45).**

1) A board

2) Product backlog Cards: illustrate the attributes of a project and includes information like sprints, complexity, budget and tasks, cost of the red and blue artifacts.

3) Problem cards: can be used by the players to slow down the opponent's progression.

4) Concept cards: can be used to counter or resolve the problems raised by their opponents.

5) Developers: depicts the development team.

6) Artifacts cards: represents the team tasks.

7) A dice

To win the game, players are required to complete the given tasks with no or least number of bugs or complete the highest number of given tasks without any errors or bugs at the end of all the sprints. Each player can only have 2 developer cards depicting their team members. The developer salaries cannot exceed the total budget. The players needs to maintain 4 heap of cards that consists of developer cards, concept cards, problem cards and one pile of artifacts. Each sprint in the game has 4 rounds. The game is played with the role of dice and players take clockwise turns. If the dice rolls 1-3, the players can pick same number of cards from the problem/concept deck. If the dice rolls 4-6 then the players can pick 2 cards from problem/concept deck but also pick cards from the developer deck to the tune of the number of dice minus 3. A player cannot hold more than 6 cards in total at a given point of time in the game and any additional cards are disposed of. The players are allowed to employ and lay off developers and can inspect or correct artifacts at any time in the game. At the end of each iteration, the instructor collects and verifies the artifacts and errors are counted. The implemented artifacts are then removed from each player before the start of the second sprint. The same process is repeated till the time all sprints are finished. The player that finishes all or highest number of tasks in the sprints with least numbers of errors is declared the winner. Although the game validation process indicated at some of the limitations of the game but still the process recommended that using the game along with classroom sessions can help students experience real world project scenarios and can enhance their learnings (Fernandes & Sousa, 2010; Gkritsi, 2011).

b) **Lego City:** Lego games have often been used to teach the concepts related to business management. Even the company Lego group officially supports the games for schools and business managers. Krivitsky, (2011) designed a Lego bricks based simulation that could be played in teams and could be adapted to teach scrum process. In this game name Lego City, the players in the each team create a city consisting of various items with certain features such as buildings, airports, hospitals, malls, schools, bus stops etc. by adapting the scrum process. The simulation can be adapted according to the need and size of the players and simulates scrum roles such as product owner: played by the instructor, Scrum master: played by a co-instructor or by a team member chosen by the team, team members: players in the team other than scrum master. The game also portrays the scrum practices like team organization, project chartering, product backlog, estimation, sprint planning and execution, and sprint review. In the game, team communication style, broken processes and behavior of the team members is observed by the instructor and at the end of the game, he/she provides feedback and suggestion to the players. Velic et al. (2012) in their study on teaching scrum process through Lego play advocated the use of Lego bricks as a mode to impart scrum knowledge to the learners from different background and experience levels.

c) **Virtual Scrum:** To overcome the limitations of time, scope, and facilities in the academic arena, Rodriguez et al. (2015) proposed a virtual scrum, a 3D computer simulation game imitating the real world project, allows interaction between the players. In the simulation game, the players can experience the scrum role such as product owner (played by the teacher), scrum master and development team. Virtual scrum reinforces various practices and artifacts of scrum process:

a) Creation and prioritization of user stories in spreadsheet format.

b) Sprint backlog planning and estimating user stories by applying planning poker technique.

c) Sprint execution, controlling and monitoring by applying daily meetings and using task board.

d) Sprint closing by applying sprint review, retrospective meetings and artifacts such as burn down charts.

The evaluation study again suggested that simulation was helpful in enhancing student's fundamental knowledge of scrum although the simulation did not outperform the student's expectation.

d) **Scrumble:** is again a board game created by Trocherie, (2014) and is formed on the basis of roles, events and artifacts of scrum process. The game is aimed at presenting the operations and customs of the scrum process in a software development project setting. The game presents a number of problems and opportunities scenarios that are faced by the project team and at the same time tries to provide instant solutions. The objective of the game is to provide a consolidated scrum learning to the players in a fun way. The rounds of the game are different based on the player's decisions and creativity. The game depicts scrum values such as team spirit, creativity, self-organization, transparency, respect, conflict resolution, ability to prioritize and commitment. The game also requires components: The board, pawns, markers, cards, and set of user stories, a dice. The game can be played between 5-11 players including product owner and scrum master and requires 2 hours to complete. The game uses all the three roles specified in the scrum process. To play the game each team can have one scrum master, one product owner and 3-9 people in the development team. The scrum master in the game promotes decision making at critical junctures. Product owner acts a decision maker and intermediary between the users and the development team. The success in the game is computed by the number of value points that can only be earned by completing the tasks sprint after sprint. The value points symbolize client satisfaction level. The team that scores the maximum number of value points after all the sprints is declared as winner.

3) Feature and practices in the creation of existing simulations and games

The research on various articles and literature review indicated, highlighted or proposed some of the common features that were considered while designing the existing simulations and games and should also be taken into consideration while creating or designing new games/simulations in future. Some of these features or characteristics are highlighted below:

1) The game should be enjoyable and should have elements of fun (Baker et al., 2005; Navarro & van der Hoek, 2004).

2) The game/simulation should be complex enough to illustrate real world scenarios.

3) The game/simulation should be able to provide actual project development experience so that player can relate to the context.

4) The game/simulation should be adaptable as per the needs, requirements and knowledge level of the players.

5) The game/simulation should allow instructor to monitor and inspect player's activity, improvement and progress.

6) The game/simulation should allow a player to review his own progress or standing in the game in comparison with his opponents.

7) The game/simulation should provide adequate feedback to the players with reference to their decisions (Baker et al., 2003; Navarro & van der Hoek, 2004).

8) The game/simulation should be simple, easy to learn and play (Baker et al., 2005; Navarro & van der Hoek, 2004).

9) The game/simulation should represent both specific and general software lessons, techniques, artifacts and customs (Baker et al., 2005).

10) The simulation/game should support information exchange, ensure collaboration and fluency in communication among the players (Baker et al., 2003).

4) Critical points to be stressed in simulations and games.

The research on the various paper and articles demonstrated some of the critical points that need attention whenever one creates and implements a new simulation or game. Concentrating on these vital points promote higher efficacy in simulations and games. Some of the points highlighted in the reviewed articles are mentioned below:

1) Studies have suggested that role of instructor/moderator/facilitator in successful implementation of simulation or game and is one of major factor that helps to achieve learning objectives. The studies further advocated that a knowledgeable instructor/facilitator can enhance the learning whereas learning effect is reduced or constrained without a good instructor (De Smale et al., 2015).

2) The learning objectives of the game should be clearly defined and game and simulation should be developed on the basis these learning objectives. Thus learning objectives are critical in important enablers (De Smale et al., 2015).

3) The simulations and games should promote what is being taught in the class lectures. The effectiveness of the simulation depends on its integration with the course curriculum. The other enabling factors are the course guidelines clarity, incentive for interaction and face to face communication with the instructor (De Smale et al., 2015).

4) Evaluation of the effectiveness of the game and simulation is equally important. The evaluation enabling factors could be: classification of learning objectives, using of larger sample database, using both qualitative and quantitative evaluation techniques, spend meaningful time in analyzing sample data (De Smale et al., 2015).

5) Students prefer computer games over card and board games (Caulfield et al., 2011a).
6) The players should be able to learn how to handle complexity and uncertainty in the project through Simulation and games by developing their analytical and problem solving skills (Connolly et al., 2007).
7) Simulations and games can be most effective when they are used in conjunction with the other teaching methods (Navarro & Hoek, 2007).

5) How these best features can be incorporated in designing a scrum simulation?

The research also exhibits and reveals the ways the features and critical points can be fused into the simulations and games. Some of the ways suggested in the various papers are compiled below:
1) By fusing the different roles related to software process methodologies into games/simulations and allowing players to play these roles during the game play (Connolly et al., 2007).
2) By fusing authentic business model in games such as clients, budget, time etc. (Connolly et al., 2007).
3) By fusing AI (Artificial Intelligence) to play the role of a competitive player when the player is unavailable especially in a team-based activity (Connolly et al., 2007).
4) By recording the gameplay to examine postgame analysis and evaluation (Connolly et al., 2007).
5) By clearly mentioning the assumptions
6) By providing clear guidelines to the players on how to play the game and simulation.
7) By assessing the knowledge of the players by giving them quizzes.
8) By assigning the software process roles to the players and instructors (Mahnic.V, 2015).
9) By asking players to break the user stories to an appropriate level.
10) By providing clarity on product backlog, rules for sprint backlog, appropriate administrative workload, and good co-operation with the Scrum Master and the Product Owner (Mahnic.V, 2015).
11) By providing clear definition of done especially in case of scrum simulations and games (Mahnic.V, 2015).

6) Limitation of existing scrum simulation and games

Even though evaluation on the existing scrum based simulation and games provided evidence to prove their potential in enhancing the students learning but these evaluations of the various paper also highlighted certain limitation of the existing games and simulations based on scrum. The list of these limitations is compiled below:
1) Some of the existing games require players to have prior knowledge of scrum before they play the game (Fernandes & Sousa, 2010; Gkristi, 2011).

2) Existing games do not cover all the aspects and practices of the scrum process (Fernandes & Sousa, 2010; Gkristi, 2011).

3) Some of the existing games/simulations take longer time to complete in case of more participants (Fernandes & Sousa, 2010; Gkristi, 2011).

4) Some of the existing game allow limited flow of communication (Rodriguez et al. 2015).

5) Lack of tracking with respect to user stories and team progress (Rodriguez et al. 2015).

6) Lack of scrum artifacts (Rodriguez et al. 2015, Fernandes & Sousa, 2010, Gkristi, 2011).

7) Some of the games are considered as toy games and instructors doubt their usefulness in improving player's skill and knowledge level.

Methodology

This sections describes an outline of the methods that and approaches that were taken to develop this paper. Likewise this section also discusses about the procedure and techniques that will be used for evaluation of Scrum Sim.

Work Plan: Initially a work plan was created to identify the key action steps that would be required to develop a scrum based game. The identification of key action steps further provided an insight to identify the expected outcomes, people responsible at each step, resources required and various deliverables. As the timelines were set for completing each key action step, thus it provided the direction and set pace of the entire project.

Literature review: Before designing a scrum game, there were certain key questions that were needed to be answered. Also there was a need to know more about the existing games in detail. Therefore a primary research was conducted in HU's library database and google scholar and the literature on the existing educational games was performed. In all 26 papers were reviewed and were bifurcated on the following criteria:

a) Papers based on the studies evaluating the usage of educational simulations games

b) Papers representing existing games and simulations based on traditional software project management.

c) Papers representing existing games and simulations based on scrum software project development.

d) Papers on comparisons between the existing games.

Scrum Simulation Designing: In the design phase, the outline and the concept of the scrum simulation was created. This included identification of a dummy software project which formed the base for simulation game. The literature on existing project management games was further reviewed to identify the real world project scenarios and to learn how the same can be incorporated in the dummy software development project. The application on which the simulation game can be played was also identified during scrum designing and scrummage (an open source tool) was finalized for playing the scrum simulation in the university's context.

Game prototype and testing: This is one of the important step in the scrum simulation game development process. At this step, the intricacies of the simulation game were defined, the structure of the game was created which included fusing the project scenarios in the scrum game, creating rules, game play and testing the simulation game (Unit testing) before it was presented for further evaluation.

Evaluation: At this step, the Scrum Sim simulation game will be validated. To validate the game following further steps were/will be taken:

Identification of the participants: To validate the game, students who were pursuing the agile course in the university will be invited. To ensure no biasness and encourage students to participate, extra credit was given to the students who participates in the process.

Validation Design: To perform the initial evaluation of the game, it was decided to design a simple experiment where the students were taught to play the game and then they were asked to provide their written feedbacks by answering a structured questionnaire. While the proposed evaluation method is more subject that some other methods, it is believed that this method is appropriate for the initial evaluation of the game as it allows us flexibility in the information we gather.

Data Collection: The data was collected by using an online survey from the students who participated in the experiment. For the online survey, a questionnaire was prepared which had mainly closed ended questions but a few open ended questions. The participants completed the questionnaire by expressing their opinions on the effectiveness of the simulation game. Some of the questions were asked for a numerical answer on one to five scale while others allowed participants to express their feedback and response. The questions that were asked were specific and were adopted from the previous studies.

Data Analysis: To analyze of the data collected via questionnaire, both quantitative and qualitative analysis was performed. To analyze the questions that had numerical answers, quantitative analysis was chosen whereas for the open ended questions for which the students provided their general feedback, qualitative analysis was used.

Writing up Results: The results and findings of the quantitative analysis were explained and were supports by tables and graphs whereas the findings from the qualitative analysis were supported by verbatim quotes from the participants.

Scrum Sim- Game Description

The simulation game Scrum Sim, based on a combination of board games like play scrum and Scrumble is a competitive game in which different scrum teams compete in a software development project that follows scrum process. The simulation game can be played by 4-5 different teams consisting of maximum of 6 members in each team. Each team completes the number of tasks which are defined at the staring of the game.

Scrum Sim is a game designed on the roles, events are artifacts of the scrum framework. The aim of this game is represent various aspects of scrum framework in a team for a software development project. The game basically aims at providing a thorough understanding of the concepts, events and roles of the scrum process to software engineering or project management students who have the basic knowledge of scrum. The game can be played in an open source software which facilitates scrum process. The game would highlight a number of problematic and opportunity scenarios that are faced by a teams during the game play. These situations are designed to familiarize the students to a real software project scenario. The game is an attempt to teach students by simulating a scrum software development project from inception to delivery. By the end of the game, players learn about the various concepts, roles and artefacts of the scrum framework. The objective of the game is to be enable the students to learn the concepts of scrum process in a fun way. In the game the participants work in team and play the role of scrum master and developers. The role of a product owner will be played by the game moderator or the instructor. Each team competes again the other team to complete a software development using scrum framework by delivering maximum value with least cost. The players will face new opportunities, constraints and problems that will enhance their knowledge of scrum development.

a) Learning Objectives:

Students after playing the game will understand/do the following:

- Enter user stories and create a product backlog.
- Prioritize user stores.
- Analyze the Interdependencies between the user stories.
- Experience the activities, responsibilities and interactions associated with the various roles in the scrum framework.
- Create and organize sprint backlog, based on prioritization, value and interdependencies between the user stories.
- Learn to estimate the size of the user stories based on experience.
- Discover the concept of task board, learn to plan and organize tasks during the sprint.

- Measure the team velocity and based on that make decisions.
- Experience self –organizing teams.
- Learn to measure performance and improvements.
- Discover the concept of sprints, which includes sprint planning, sprint execution.
- Discover the concept, activities of daily meetings, review and retrospective.
- Analyze the impact of budget on the project.
- Analyze the impact of skill level and personality of the team on the project.
- Discover and analyze the outcome of unexpected events on a project and make decisions accordingly.
- Discover and interpret the artifacts of the scrum framework.
- Use a scrum tool.
- Assist the players to develop and cultivate agile mindset by enabling the players to experience complexity, vagueness and ambiguity during the project.
- Analyze the importance of effective communication.

b) Overview of the game

Players: As defined in the scrum framework, Scrum Sim has three roles in the game: i.e. product owner, scrum master and the development team. To play the game, every team needs to have once scrum master, 4 to 5 persons will be take the role of development team and product owner will be the moderator for all the teams.

Scrum Master: will not directly play the game as a player but their role will be an important one as this will person will coordinate all the activities and facilitates decision making to ensure smooth running of the game. The scrum master should be familiar with the rules of scrum.

Product Owner: is the one who is responsible for the product. He/she focuses on delivering the value and is a communicator who has the business or the domain knowledge about the field for which software is intended. In the game, product owner acts as decision maker and intermediary between the users, project sponsors and development team.

Development team: consists of players performing the role of junior or senior developers and tester. All the team members will be equal in a team. These teams are self-organizing and they participate in the development process and are responsible for incremental delivery of the software during all phases. The gameplay and the end of the game depends entirely on the development team according to their salaries, experience, qualities, skill and interactions.

c) Objective and winning condition

The prime goal of the game is to provide players a better understanding of the scrum development process. Teams will eventually figure out the whole point of the simulation game. There is no specific condition for wining. The real success lies in learning achieved during the game and its application in the real project context. The failure would be if the players give up without learning any lessons in the game play.

In the simulation game, the team success is evaluated by the completion of user stories according to their priority, value or satisfactions point's i.e. delivered by the team in term of completed user stories by performing various tasks by using minimum allotted budget. The number of value points must be as large as possible at the end of the game with the usage of least possible money. Success would mean delivering prioritized client satisfaction to receive product increment, is achieved according to the dynamics that is filled with various problems and opportunities.

d) **Duration**

The instructor can choose the number of sprints according to the time allotted. But the game should ideally be played for at least two sprints and maximum of 3 sprints. These recommended duration of each sprint should be at least 1 week.

e) **Meeting sessions**

Before starting of the sprint, the product owner should spend a session with players to present the project and the product that is to be developed. Allow a small session for about half an hour to 45 minutes for planning and 15-20 minute session for review and retrospective in each sprint. The team can have 5 minute daily meeting sessions which can be coordinated by the scrum master in each team.

f) **Game Pre-requisites**

Before beginning of the game, there are following necessary components that should be taken care of:

- The role of product owner should be played by the person having project knowledge.
- The player in a role of scrum master should have basic knowledge of scrum.
- Decide an open source tool that supports scrum process and provide working knowledge of the open source tool to the players in advance of the game play.
- It is recommended that in one group only four to five teams should be allowed to compete against each other.

g) **Preparation and organization**

Ideally the game should be played in conjunction with the class room sessions. Regarding the organizational aspect following list is recommended for successful completion of the game.

- The teams should be briefed about the rules and expectations from the game.
- Limit the numbers of players in each team to 5 or 6.

- Each team should have one person in a role of scrum master.
- The role of a product owner can be played by an instructor/ moderator.
- It is recommended to conduct a small workshop before the game is played to develop the user stories with the players to create a product backlog and incorporate them in the open source tool. If the moderator has an existing product backlog, the same can also be used to save time.

Game play

The entire game can be organized into the following two parts:

1) **<u>Pre start of the Game</u>**

a) Plan a dummy software development project

- Select the Project to be simulated   in consultation with the product owner.
- Create at least 30-40 user stories for the project and create a project backlog in the open source tool.
- Prioritize all the user stories (Decision to be taken by the product Owner). (For priority rating: refer to the appendix table 1: **page 45**).
- Rate each user story on the value points on the scale of 1-5 at an average of 3 value point per user. (Decision to be taken by the product Owner)
- Decide the definition of done for each user story (Decision to be taken by the product Owner).

b) Team Selection and Budget

- Each Scrum team can have maximum of 6 team members in including the scrum Master. The moderator or the instructor will play the role of a Product Owner.
- All the teams get the same budget of $50000 (6 players in each team) $ 45000 (5 players in each team) for 3 sprints. The team can select the members from on the basis of the following attributes:
  1) ***Per hour rate:*** members with higher salary will take more of your budget.
  2) ***Skill Level:*** skill will be rated on a scale of 1-5, member with higher skill can do the work more efficiently.
  3) ***Personality:*** Personality is rated on a scale of 1-5. Higher rating of the member shows his tendency to be a good worker in terms of friendliness, professionalism etc.

(For choosing the development team members: refer to the appendix table 2: **page 46**).Teams are free to pay 5 team members i.e. developers, testers and fix their remunerations based on the provided table. All the teams will pay $ 50 per hour to the scrum master.

2) **Game Execution**

a) Sprint: Each team simulate project performance over 3 sprints. Each team simulate project performance over 3 sprints. In sequential order each sprint (7 development days) consists of one planning session, daily meetings (every day), a review session and retrospective session.

1) Sprint Planning

- Product owner (instructor) and team discusses **interdependencies** among the user stories. Development team members can give suggestion but the last word will be of the product owner.
- The product owner in discussion with the each team selects the **high priority and high value** stories to create **sprint backlog**.
- Teams **estimates the size** of the user stories and choose the number of tasks to be carried out and numbers of hours to perform each task to complete each user story. The teams can perform relative estimation on the basis of **Fibonacci sequence**. The scrum master in each team will conduct this session. The number of tasks can be chosen on the basis of complexity of the user story.

2) Sprint execution

Teams will be now start to execute the sprint. Each team will be given different problem scenarios, opportunity scenarios or a combination of both each day except day one of the 1st sprint when they will not get opportunity scenario. The choice of the scenarios given to the team will depend on the choice of the moderator/instructor (refer to the appendix List 1: Game scenario's: **Page 46**). The instructor should ensure that each team gets the similar kind of scenarios.

As the game is played, these scenarios will impact the teams on the basis of the skill and personality level of the team members in the project. Team will not able to complete the tasks according to the scheduled work hours and some of the task will remain incomplete for e.g. If a team has 13 tasks consisting of 48hrs of work in day, due to the impact of the scenarios, at the end of the day, team will only be able to complete 11 tasks and 2 tasks consisting of 6 hrs. Of work will remain incomplete.

If all the planned tasks for the sprint are carried out before the end of the sprint, team can add one or more user stories to current sprint backlog and try to complete maximum tasks related to these new user stories.

3) *Daily Standup:* Each day, the scrum master can conduct a 5 min minutes daily standup meeting in which the team can discuss their position in the game. The instructor can also make a list of questions regarding scrum framework and the same can be given to the teams for discussion. This would help the players to better understand the scrum.

4) *Sprint Review:* On day 7th i.e. on the last day of each sprint, sprint review session will be conducted where the product owner reviews the deliverable product and give his/her feedback. The Product owner can provide positive feedback by congratulating, encouraging or gives constructive advices. He can also give negative feedback that can impact the game at the time of review or at the start of a next sprint (refer to the appendix List 1: Game scenario's: **Page 46**).

Note: If it is found that the product has a bug, the user story will not be taken as complete and the team will not get score for that. Also In case where the tasks remains incomplete at the end of the sprint, the related user story would again go back to the product backlog and can only come back to the sprint backlog on the decisions of the product owner.

To eliminate the bias in the review, it is recommended that moderator should give each team should get one positive feedback and one negative feedback to each team during the sprints for e.g. If there are 4 team participating in the game, during sprint 1 review, 2 teams can get positive feedback and 2 teams can receive negative feedback whereas during sprint 2 review, the teams which received positive feedback in review 1 can get negative feedback in review 2 and the teams which received negative feedback in review 1 can get positive feedback.

5) *Score Calculation:* During the review, the scrum master from each team calculates the current standings of the teams score (Refer to the appendix Table 5: Overall sprint calculation, Page **47**). The team score is calculated on the basis of:

- Priority points delivered in a sprint (40%) (Refer to the appendix Table 3: priority points calculation, Page **47**).
- Value points delivered in a sprint (30%)
- Comparing the allotted budget used by each team (30%) (Refer to the appendix Table 4: budget point's calculation, Page **47**). (Refer to the appendix List 2: Resource budget calculation, **page 46).**
- **Note*:*** The first two teams which complete the sprint with the lowest amount of expenditure will score the highest score.

6**)** *Retrospective:* At the end of each sprint retrospective session is held. This specific session belongs to the players. During retrospective, Players will put forward their viewpoints about what they liked, disliked and hated during the sprint or what are the problems and issues they faced that impacted their team and what are they intend to do during the next sprint.

b) Winning the Game: Since the entire simulation game consists of 3 sprints, the overall score of each team in each sprint will be added to the points score by the team in the previous sprint to calculate the cumulative overall score of each team. After 3 sprints, the team that scores the highest cumulative overall score is declared a winner.

c) End of the Game: The game ends after the last retrospective and the thoughts needs to be given on the hurdles faced during the course of the game. This can be facilitated by the scrum master and chaired by the moderator. The teams can provide their suggestions and opinions on what they learned from the game.

d) Scrum Sim rules: Mentioned Below are the rules of the game:

1) All the teams should select at least 10 user stories.

2) All the teams should plan at least to work for 8hrs per person each day i.e. maximum of 48 hrs. Of work each day and 336 hrs. For each week.

3) The moderator should ensure that all the teams have tasks for 48hrs of work each day and 336 hrs. For each week.

4) Each sprint will be of one week each.

5) The team will try to complete 48 hrs. Of work every day. The completed work should be updated every evening in the tool.

Exceptions:

a) *The teams can complete more than the planned work i.e. 48hrs. If the team saves time because of review or opportunity scenarios for e.g. imagine a scenario where as bug is resolved at an early stage and a team is able to save 3 hrs. Of QA work. In this case, the QA's will be able to complete more tasks on that day. Similarly in case of a good sprint review, a team may save some extra work in next sprint which would mean that on day 1 of the sprint 2, that team will not be able to complete more than the planned work.*

b) *The teams can complete less than the planned work i.e. 48hrs. If the team loses time because of problem scenarios for e.g. imagine a scenario where as bug is resolved at later stage and a team* *loses 3 hrs. Of QA work. In this case, the QA's will be unable to complete the planned tasks for that day.*

The teams can have more hours of work allotted if they complete their entire tasks before the end of the sprint.

6) The work done on the last day of each sprint (7th day) of each sprint can only be marked as completed after conducting the review and retrospective sessions.

7) Each member of the team will be paid for 8 hrs. Of work each day.

8) Each team can replace 2 of its members from the development team (i.e. change in the pay structure) after each sprint but this will result in losing 8hrs. Of work in the sprint .which means in normal circumstances, if one team member is replaced, that team will only be able to complete maximum of 40 hrs. Of work instead of 48hrs. On day 1 of the new sprint.

9) Each team will pay scrum master $ 50 per hr.

10) The total money paid to the project team including scrum master cannot exceed more than the budget fixed. If any team exceeds the given budget in the simulation, that team goes out of the game.

11) After each sprint, cumulative score of all the teams should be published so that all the teams know their respective standings.

12) The task completed each day should be marked complete in the tool every day.

13) A user story cannot be marked as complete till the time all the tasks associated with it are marked complete.

14) Any incomplete user story at the end of the sprint shall go back to the product backlog and will only comeback in the next sprint after product owner's approval.

15) In case of a bug is found in the product, the user story will not be taken as complete and the team will not get score for that.

16) In a single user story, the task should follow a sequence for example, tasks related to testing cannot be marked as complete before tasks related to coding or unit testing are complete but tasks such as writing test cases can be simultaneously done with coding tasks. This clarity on this should be given to the players before the game execution.

17) Consistency should be maintained in estimating the number of hours for performing the tasks for e.g. in sprint 1, for a user story of size 5, the total numbers of hours estimated are 30 hrs. Now in the second sprint, the total numbers of hours for the same size of user story should be in the range of plus/minus 20-25%. **Note**: The user story with smaller size would require lesser number of hours to compete as compared to a larger story.

18) There could be a situation in the game where on a given day, a particular member will not have enough hours to complete the task but his other team member with similar skill have extra time. Therefore the teams can use the extra time of the other team member (having extra hours) to close that particular task. For example on a given day, developer "A" can only perform 5 hours of work and developer "B" can perform 8 hours of work. Now developer "A" is assigned an important task which can impact the other team members. Since developer "A" has lesser number of hours with him, the task remains incomplete. To complete the important task, some of the hours of the developer "B" can be used to complete that important task.

Results and Findings

Experimental Design:

In order to perform the qualitative and quantitative initial evaluation of the simulation game, a simple experiment was designed in which 6 graduate students from Harrisburg University enrolled in the project management course played the game and then were asked to submit written feedback in the forms of answers to a questionnaire. The students played the game in a team where they were assigned different scrum roles. To play the game, it was decided to use Scrummage as an open source scrum tool. Scrummage is an application that supports scrum process and is provided by the Harrisburg University to its students enrolled in agile program.

To introduce the simulation game Scrum Sim to the students, a game manual was prepared and provided to the students. The game manual had a brief description about Scrum Sim, its learning objectives, project description to be simulated, game play and rules Apart from the manual, two 10 minutes videos were created and provided to the students explaining them on how to use scrummage application and how to play Scrum Sim simulation game using the tool (Refer to the appendix List 4: Scrum Sim Manual and List 5: Video Links Page **46**).

Due to the time constraint in testing the game, certain adjustments were done in order to test the game with in the provided timeframe. Therefore following things were decided:

1) Instead of simulating all the three sprints, only two sprints will be simulated.
2) Instead of simulating the 2 sprints in 2 weeks, 2 sprints should be simulated within a week.
3) Instead of taking 10 user stories, students can choose only 6 user's stories will be selected in each sprint.
4) Students will be provided with set of user stories in the scrum tool.
5) Set of basic tasks should be already provided to the students with in each user story.
6) The ceremony of daily sprint meetings will not be tested as it does not have an impact on the outcome of the simulation game.

The entire validation process was divided into 7 sessions and the team along with the moderator collaborated via Adobe connect sessions. In the initial session, the students were first introduced to the game, its rules and the game play. The students were also given a training using the scrum tool on how to play the game in this initial session. In the later 6 sessions, students played the simulation game by completing 2 sprints. In each session, students usually simulated 2-3 days of work. Following this they completed a questionnaire stating their thoughts and feelings about the game in general, their opinions about the effectiveness of the simulation game in teaching Scrum Process and ceremonies. The questionnaire consisted of both open ended and closed ended questions. Some of these questions required a numerical answer on a scale of 1(low) to 5 (high), whereas other questions allowed students to provide their responses in free form.

Experimental Results:

The students feeling about the game were favorable. The below mentioned Table 6 presents the results for the numerical score questions, with indicating 1 as the lowest and 5 as a highest score. Overall the results are very good.. All the questions indicate positive feedback with an average values ranging above 4.17 out of 5 except question 2 with an average value of 3.3. The average value of 3.3 in question 2 is a positive indication that shows the simulation game was neither complex nor too easy for the students to play. Therefore the response from the students is indication of clear vote of confidence in the game and its merits by the students who participated in the game.

Table 6: Questionnaire numerical scores

Questions	Student 1	Student 2	Student 3	Student 4	Student 5	Student 6	Average
1. Do you consider Scrum Sim enjoyable to play?	4	5	5	4	5	4	**4.50**
2. What was the level of difficulty of Scrum Sim?	2	3	4	5	4	2	**3.33**

3. Do you think Scrum Sim was able to enhance your knowledge of scrum process?	4	5	5	2	4	5	**4.17**
4. Do you think Scrum Sim helps students to keep the trace-ability of the user stories?	5	5	5	4	4	5	**4.67**
5. Do you think Scrum Sim facilitates team making estimation and clarifies the daily work by having the work available for team members?	4	5	5	4	4	5	**4.50**
6. Do you think Scrum Sim provides students with metrics to present the team performance through the sprint retrospective meeting?	4	5	5	3	4	5	**4.33**
7. Does Scrum Sim facilitates the availability of Scrum artifacts?	5	5	5	3	4	5	**4.50**

Regarding question no 1, to reinforce the idea students also wrote:

- "*The game is applicable to real life scenarios".*
- *"Sessions were very useful". Scrum Sim is interesting if you know the rules.*
- *"Enjoyed playing with scenarios, Scenarios make game harder".*
- *"Learned how to use scrummage tool".*
- *"Learning and playing the game was fun".*

Regarding question no 3, to reinforce the idea students also wrote:

- *"It helps to understand the process from beginning to end of the sprint even though standup meeting of 15 minutes was not applicable".*
- *"The game was very much informative".*
- *"I am able to understand the scrum process in a better way".*
- *"Gained valuable knowledge".*

Regarding question no 6, to reinforce the idea students also wrote:

- "*The game definitely provides students with metrics to present the team performance".*

The other feedback received on this question were:

- "*Tracking and burnout charts needs to be done manually".*
- *"It was little difficult to discuss everything virtually with team members. A little distraction or improper communication could make things difficult. If everyone in team participates in adobe sessions then it will be lot more easier to work as a team. SM should be more active in these activities who can communicate well and try to understand individual's perceptions.*

Student's answers to the open ended questions also reflected their positive feelings about the Scrum Sim. Regarding the question "***Would you suggest including Scrum Sim in a software engineering/Project management course"?*** , the student's wrote:

- "*Yes, It helps to understand the cross functionality of the team".*
- *"Yes, it could be a nice learning tool".*
- *"Sure. It will help students to understand scrum process practically. They will know how these tools are helpful in organizations".*
- *"Strongly recommend".*
- *"It can be a good idea*".

Regarding the question: ***"In general what is your opinion about Scrum Sim"***? , the students responded:

- *"Scrum Sim is beneficial, it helps to relate with the scrum artifacts".*
- *" Scrum Sim is a very good game that subjects the user to the lifecycle of project handled in agile"*
- *"It's great, should have more automated features.*
- *"I would say it is convenient and user friendly".*
- *"Game was well structured with all scenarios as if it is actual project".*
- *"It's an interesting game. It provides understanding of the scrum/agile framework"*.

Regarding the question: ***"What was the most difficult part of the game"?*** The response of the students varied such as:

- *"Assigning tasks to users".*
- *"Make a judgement which tasks to finish to complete the user story".*
- *"When scenario comes and when we lose hours it was difficult to manage work and hours".*
- *"Working with scenarios"*.

For feedback questionnaires (Refer to the appendix List 3: Feedback questionnaire Page **46**)

Observational study

While the students were playing the game during the experiment, an observational study was also performed in which students were observed and some general questions were also asked during the Scrum Sim play sessions. The purpose of this study was to investigate the effectiveness of the simulation game by observing the student's behavior during the game play. Therefore highlights of some of the most important insight and lessons learned about the Scrum Sim are:

- The knowledge of operating the open source tool is of utmost importance. Without the knowledge of the tool, it is difficult for students to play the game.

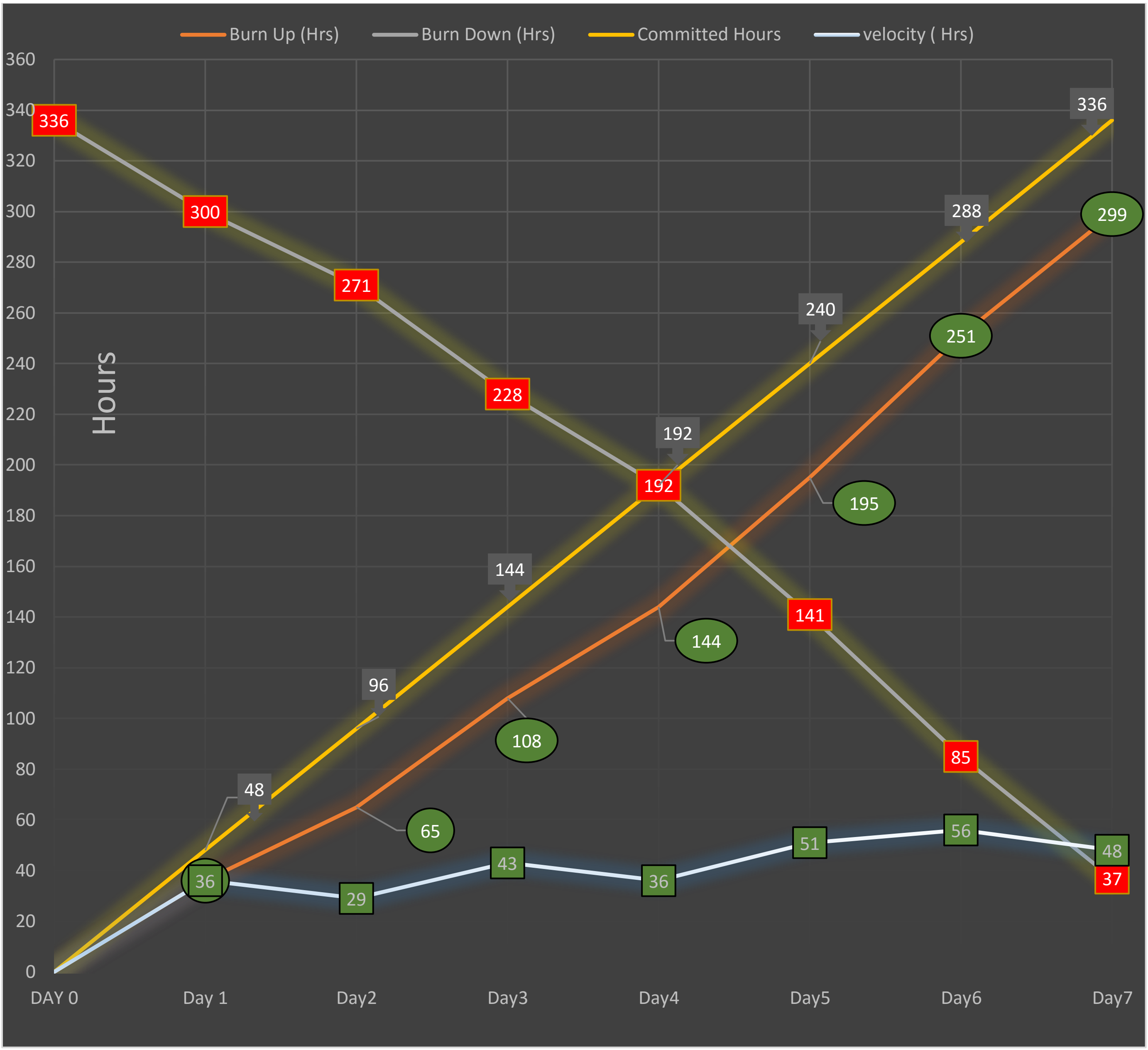
- The most difficult parts for the students in the game was to assign the task to the team members according to their roles, assigning 56 hours of work to each team member for one week.
- There were instances when a user story with lesser story point had more hours work assigned to it rather than a user story with higher story points.
- The knowledge of the game rules are of great importance.
- The role of the moderator is also important as he needs to guide a team during the game play especially about the rules.
- The game reconfirmed the author's concept that Scrum Sim is based on 4 strategic pillars :
    I. Selection of the team members i.e. their skill and personality levels.
    II. Selection of user stories to be delivered based on priority, value and independencies on other user stories.
    III. Organizing the work between the team members i.e. assigning the tasks and work hours to each team member.
    IV. How to approach the completion of work i.e. finishing the work to complete the assigned tasks according to the constraints.

Findings

The results of the experimental and observational study indicate that while most of the answers were positive, it was noticed that student lacking the knowledge of the scrum tool had a little bit of difficulty in playing the game. Further students with prior knowledge of scrum would do much better in understanding the game and it dynamics. This point establishes the point of view that game should not be used in isolation and instead should be used to complement the theoretical concepts taught in the classroom settings. The evaluation process also demonstrated that the training session or a dry run of the simulation game with the students can help them to understand the simulation from a strategic point of view. In nutshell, the evaluation of the game was very positive and demonstrated that the game could project the procedures, practices and ceremonies of the scrum in a comprehensive manner by depicting a closer look of real word scenarios. Further the simulation game has the potential to be a part of a software engineering or a project management course.

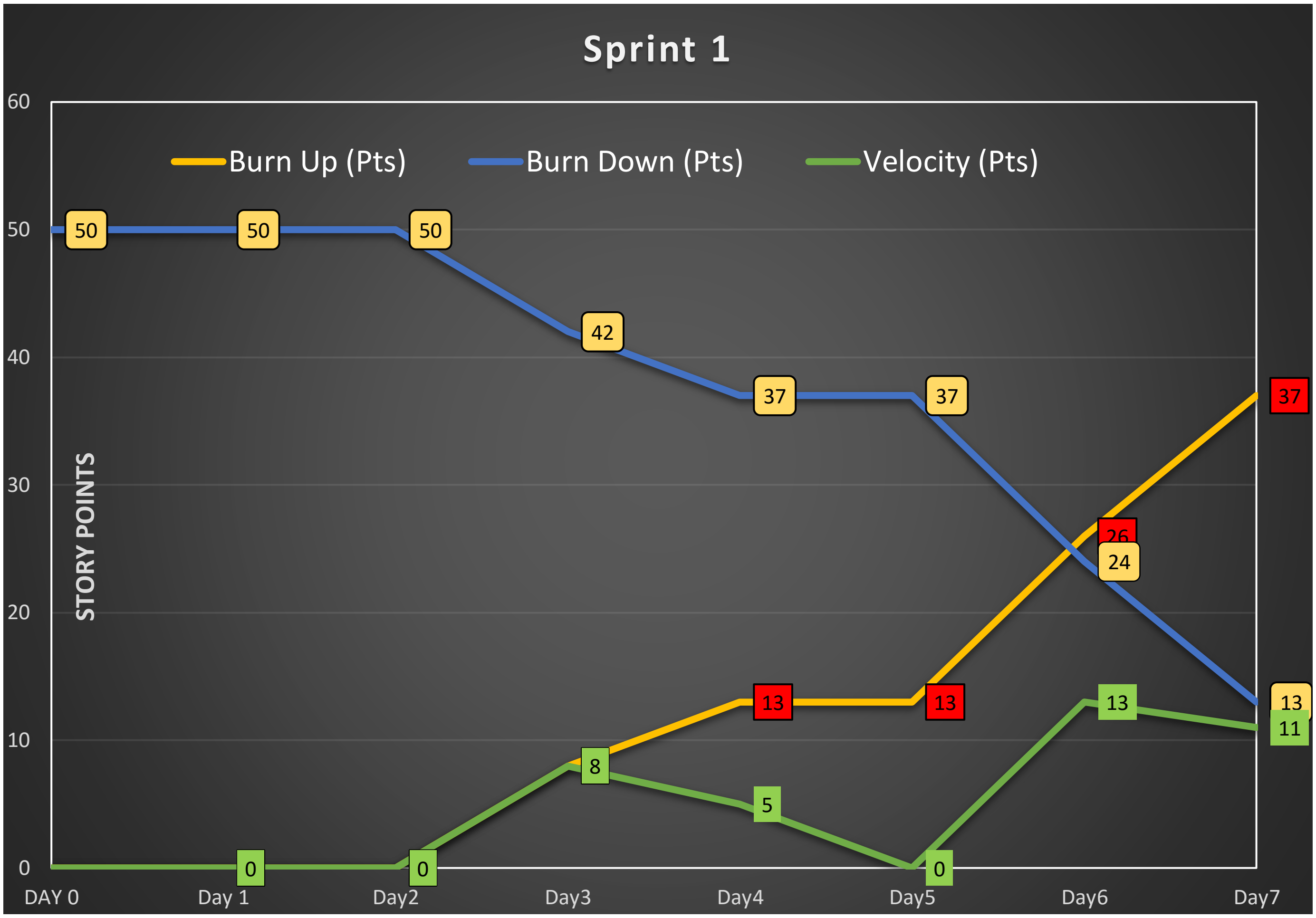
The below mentioned figures represents the scrum artifacts of the evaluation process:

Figure 1: Scrum Sim artifacts Sprint 1(Based on Work Hours)



The above figure describes the work done (Hours) by the scrum team during sprint 1. The team committed to complete 336 hrs. Of work but could only complete 299 hrs. Of work and 37 hrs. Of work remained incomplete at the end of sprint 1. The yellow line on the chart depicts the committed hours of work to be completed each day called i.e. 48 hrs. Of work. The orange line describes the actual work done by the team each day for e.g. on day 1, the team target was to complete 48 hrs. Of work but team could only complete 36 hrs. Of work. Similarly on day 2, the team should have finished 96 hrs. (48hrs+ 48hrs.) Of work but the scrum team finished only 65 hrs. Of work. This means that on day 1 team could complete 36 hrs. Of work and on day 2 another 29 hrs. Of work (depicted by daily velocity line).

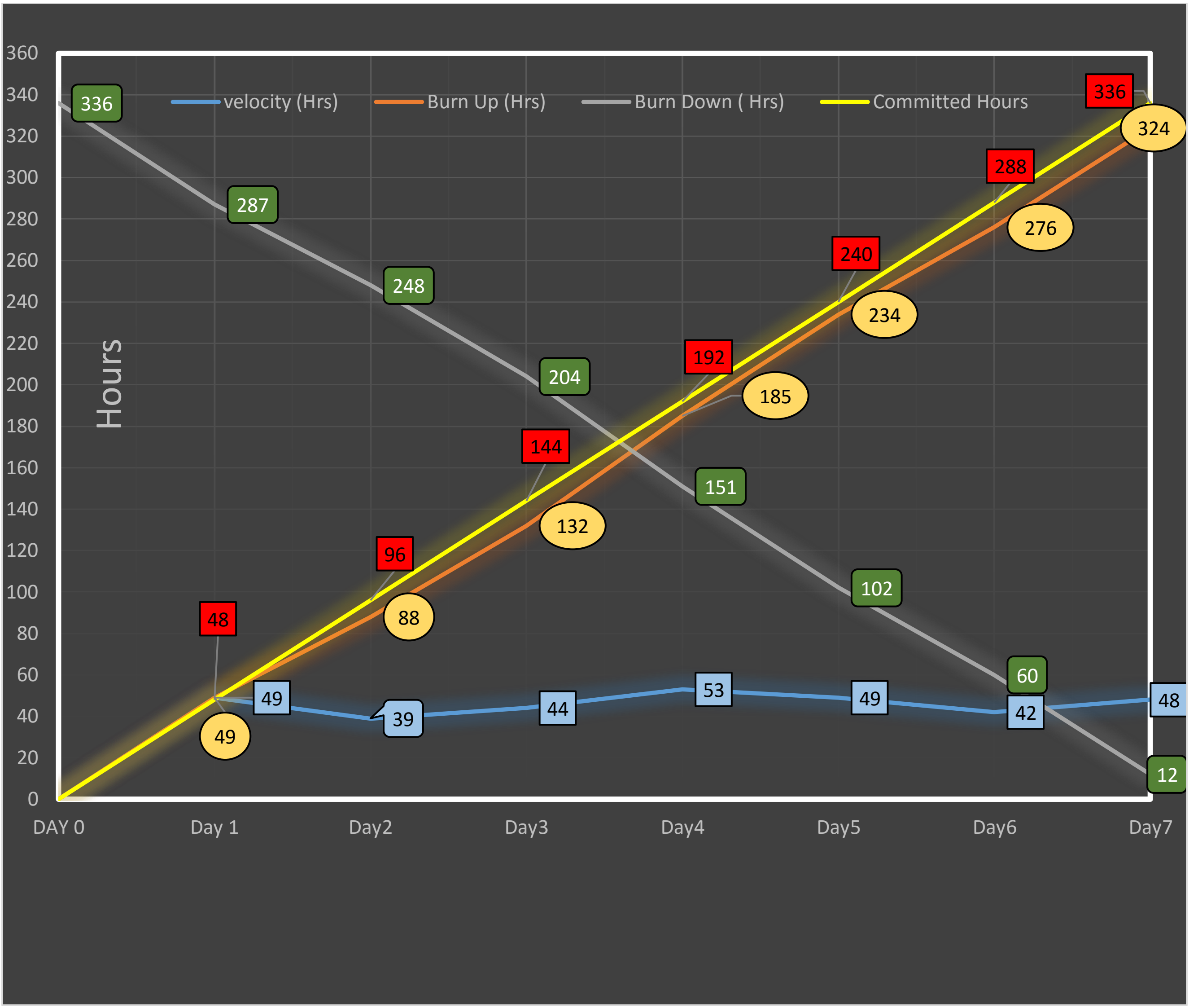
Figure 2: Scrum Sim artifacts Sprint 1(Based on Story Points)



The above figure describes the work done (Story Points) by the scrum team during sprint 1. The team committed to deliver user stories worth 50 story points but could only deliver user stories worth 37 story points and 13 story points worth user stories could not be delivered at the end of sprint 1. The green line on the chart depicts the delivered story points each day for e.g. on day 1, the scrum team could not deliver any user story but on day 3, the team delivered user stories worth 8 story points (depicted by velocity).

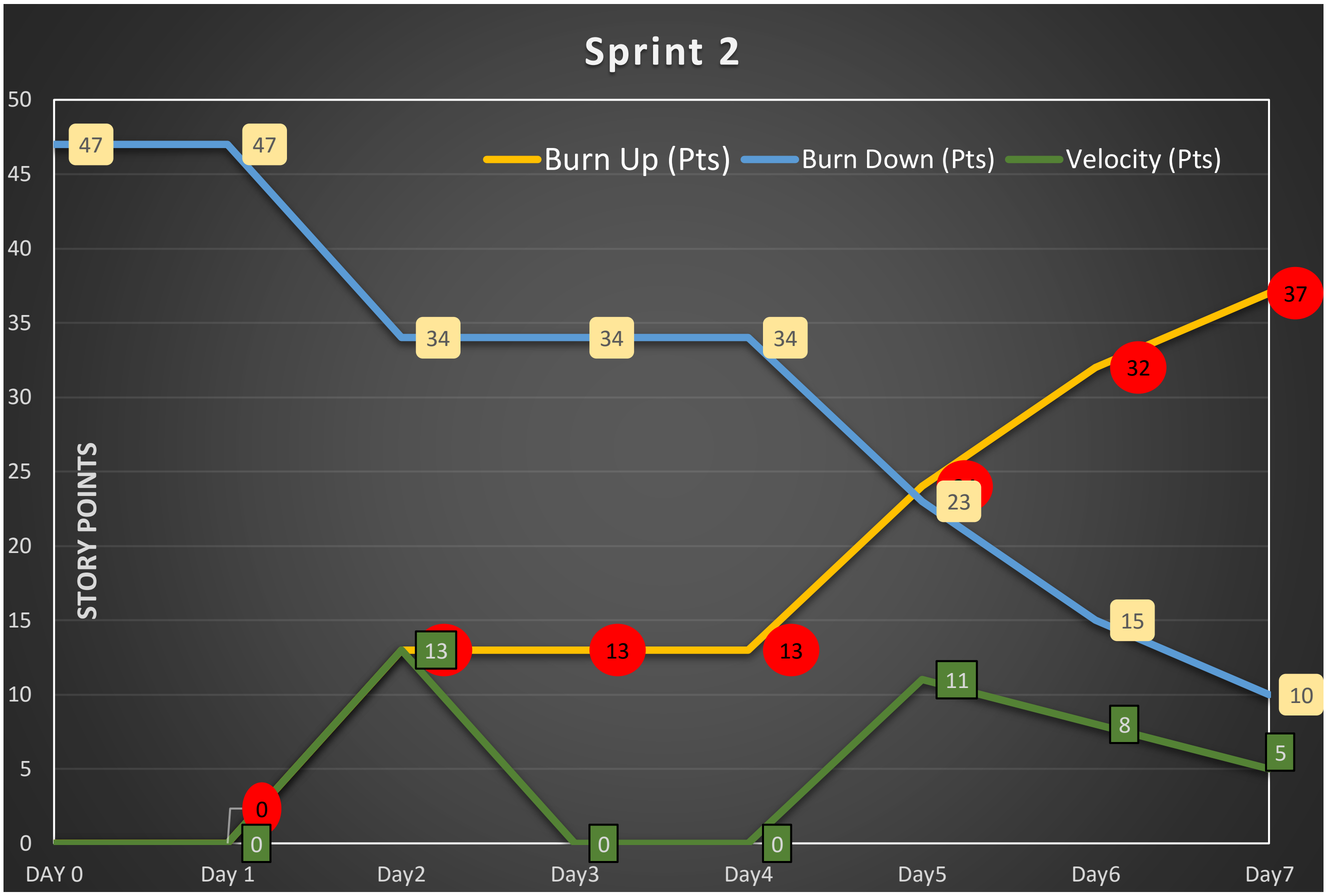
The Yellow line (depicted by burn up) describes the cumulative story points delivered at end of each day for e.g. by the end of day 4, user stories worth 13 story points were delivered by the project team which is cumulative total of story points delivered from day 1 to day 4. The blue line (burn down) describes the difference between the committed story points and the cumulative delivered story points at the end of each day of the sprint for e.g. in the above figure, the scrum team did not deliver any user stories Story till day 2 but by the end of day 3, the team delivered 8 story points which meant that the team is yet to deliver 42 story points at the start of day 4.

Figure 3: Scrum Sim artifacts Sprint 2 (Based on Work Hours)



The above figure describes the work done (Hours) by the scrum team during sprint 2. The team committed to complete 336 hrs. Of work but could only complete 324 hrs. Of work and 12 hrs. Of work remained incomplete at the end of sprint 2. The yellow line on the chart depicts the committed hours of work to be completed each day called i.e. 48 hrs. Of work. The orange line describes the actual work done by the team each day for e.g. on day 1, the team target was to complete 48 hrs. Of work but team completed 49 hrs. Of work. Similarly on day 2, the team should have finished 96 hrs. (48hrs+ 48hrs.) Of work but the scrum team finished only 88 hrs. Of work. This means that on day 1 team could complete 49 hrs. Of work and on day 2 another 39 hrs. Of work (depicted by daily velocity line).

Figure 4: Scrum Sim artifacts Sprint 2 (Based on Story Points)



The above figure describes the work done (Story Points) by the scrum team during sprint 2. The team committed to deliver user stories worth 47 story points but could deliver user stories worth 37 story points and 10 story points worth user stories could not be delivered at the end of sprint 2. The green line on the chart depicts the delivered story points each day for e.g. on day 1, the scrum team could not deliver any user story but on day 2, the team delivered user stories worth 13 story points (depicted by velocity).

The yellow line (depicted by burn up) describes the cumulative story points delivered at end of each day for e.g. by the end of day 5, user stories worth 23 story points were delivered by the project team which is cumulative total of story points delivered from day 1 to day 5. The blue line (burn down) describes the difference between the committed story points and the cumulative delivered story points at the end of each day of the sprint for e.g. in the above figure, the scrum team did not deliver any user stories Story till day 1 but by the end of day 2, the team delivered 13 story points which meant that the team is yet to deliver 34 story points at the start of day 3.

Discussion

Validating a simulation game in the field of education is a difficult task. Since there are multiple interacting factors that can impact the educational environment, therefore it may not be possible to segregate the outcome of an educational approach. Also adjustments fundamental to the educational validation include ethical problems involved in treatment, the challenges of tracking student's and getting significant numbers of students as subjects. Further the evaluation of software engineering skills is not straightforward and can only be best known when a student enters the real world (Navarro & Hoek, 2007). Nevertheless there are some general lessons that can be extracted from our evaluation experience and can be applied in different situations. One of the most important lessons that was learned was that the evaluation do not turn out exactly according to the plan. An example of this was that initially it was difficult to find the volunteers for validating the Scrum Sim and a few volunteers did not participate despite agreeing for the participation. Moreover during the different game play sessions, there were some students who were not regular in attending the sessions. Thirdly continuous follow up was required to collect the feedback from the volunteers.

There were some other key insights that were gained from the feedback given by the student's volunteers. The feedback received from the students reiterated the importance of knowing the rules of the game. One of the feedback received stated that the tracking and artifacts such as burn down were done manually. During the game play, the tracking of the game and burn down charts were created manually in a spreadsheet because of the tweaking's done to evaluate the Scrum Sim with in a given time constraint of one week. In the normal course of the game play, scrummage (scrum tool used for validation) would have automatically tracked the work hours and had demonstrated the artefact's according to the updation done in the tool on daily basis. Another insight which was highlighted during the game play was that the involvement of each team member is very important especially during organizing the work between team members. This is how the simulation game demonstrates the importance of self-organizing teams. A small distraction can make things difficult in the game. Also the student playing the role of the Scrum master needs to communicate and co-ordinate activities between the team members during the game. The same opinions were also reinforced by the feedback received from one of the student's volunteer. Therefore, in order to avoid these kind of issues, our suggestion would be to:

- Allow the participating teams to have a trial run of the simulation game for a day or two before the start of real simulation where teams can compete with each other.

- Take measures to ensure that the participation of all team members.

Conclusion and Future work

Scrum Sim represents the first attempt to provide better understanding to the software engineering/project management students about the scrum processes by creating a simulation based team game that can played using an open source tool. It addresses many of the limitations of the existing board or virtual games and bring additional benefits in the form of collaborative learning and enjoyable play. It is the opinion of the author that this simulation game can be used in conjunction with the classroom lectures and Scrum Sim allows students to gain a thorough understanding of the scrum framework by depicting the real world scenarios and reinforces practically many lessons that are presented in the class room lectures and build's a student understanding of their actions and role in the overall scrum process.

Scrum Sim is visual in nature and is enjoyable and fun to play, has decent complexity to illustrate real life scenarios, promotes collaborative learnings and provides immediate feedback to the players about the lessons to be learned. In our opinion Scrum Sim in its current state represents good balance between our stated objectives.

The result of our experiment and observational study confirm that students believe that the use of Scrum Sim simulation game can enhance their learning regarding the scrum process as most of the students volunteers agreed that the Scrum Sim was enjoyable and fun to play and enhanced their knowledge of scrum. Further the students also recommended to include Scrum Sim as a part of software engineering/project management course.

Future work should consider the following:

1) Further evaluation of the simulation game by introducing the game to a larger set of subjects further bifurcating into subjects with scrum experience and non- scrum experience to build better understanding of how the simulation game can help in teaching scrum process. This would also provide a broad sample size results.
2) The game can also be evaluated by using some of the other open source tools which would help to know the best suited open source tools for the effectiveness of the simulation game.

3) More variabilities could be added to Scrum sim simulation game such as release features or including non-IT projects in the scrum simulation game.
4) Future work should also consider the development of a full online simulation program that other educational and training companies could possibly license and use. This would include taking all the scenarios, charts, Scrum software, rules, etc. and put completely online.

References

1. Baker, A., Navarro, E. O., & Van Der Hoek, A. (2003, May). Problems and programmers: an educational software engineering card game. *In Software Engineering, 2003. Proceedings. 25th International Conference on* (pp. 614-619). IEEE

2. Baker, A., Navarro, E. O. & Van Der Hoek, A. (2005). An experimental card game for teaching Software engineering processes. *Journal of Systems and Software*, *75*(1), 3–16.

3. 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.

4. Connolly, T. M., Stansfield, M. & Hainey, T. (2007). An application of games-based learning within software engineering. *British Journal of Educational Technology*, *38*(3), 416–428.

5. Caulfield, C., Xia, J. C., Veal, D., & Maj, S. P. (2011 a). A systematic survey of games used for software engineering education. *Modern Applied Science*, *5*(6), 28.

6. Caulfield, C. W., Veal, D., & Maj, S. P. (2011 b). Teaching software engineering management–issues and perspectives. *International Journal of Computer Science and Network Security*, *11*(7), 50-54.

7. De Smale, S., Overmans, T., Jeuring, J. & van de Grint, L. (2015). The Effect of Simulations and Games on Learning Objectives in Tertiary Education: A Systematic review.

8. Drappa, A. & Ludewig, J. (2000). Simulation in software engineering training. In *Proceedings of the 22nd international conference on Software engineering* (pp. 199–208).

9. Fernandes, J. M. & Sousa, S. M. (2010). Playscrum-a card game to learn the scrum agile method. In *Games and Virtual Worlds for Serious Applications (VS-GAMES), 2010 Second International Conference on* (pp. 52–59).

10. Gkritsi, A. (2011). *Scrum Game: An Agile Software Management Game*. (Doctoral dissertation, University of Southampton).

11. Klabbers, J. H. (2009). Terminological ambiguity game and simulation. *Simulation \& gaming*, *40*(4), 446–463.

12. Krivitsky, a Scrum Simulation with LEGO Bricks (2011), 29 Feb 2016, https://www.scrumalliance.org/system/resource_files/0000/3689/Scrum-Simulation-with-LEGO-Bricks-v2.0.pdf.

13. Kropp, M., Meier, A., Mateescu, M. & Zahn, C. (2014). Teaching and learning agile collaboration. In *Software Engineering Education and Training (CSEE\&T), 2014 IEEE 27th Conference on* (pp. 139–148).

14. Kurkovsky, S. (2015, June). Teaching Software Engineering with LEGO Serious Play. In Proceedings of the 2015 A*CM Conference on Innovation and Technology in Computer Science Education* (pp. 213-218). ACM.

15. Mahnic, V. (2015). Scrum in software engineering courses: an outline of the literature. *Global Journal of Engineering Education*, 17(2).

16. Narayanasamy, V., Wong, K. W., Fung, C. C. & Rai, S. (2006). *Distinguishing games and Simulation games from simulators.* Computers in Entertainment (CIE), 4(2), 9

17. Navarro, E. O. & van der Hoek, A. (2004). SimSE: an educational simulation game for teaching the Software engineering process. In *ACM SIGCSE Bulletin* (Vol. 36, pp. 233–233).

18. Navarro, E. O., Baker, A., & Van Der Hoek, A. (2004, January). Teaching software engineering using simulation games. *In ICSIE'04: Proceedings of the 2004 International Conference on Simulation in Education.*

19. Navarro, E. & Van Der Hoek, A. (2005). Design and evaluation of an educational software process simulation environment and associated model. In *Software Engineering Education \&\# x00026; Training, 18th Conference on* (pp. 25–32).

20. Navarro, E. O., & Van Der Hoek, A. (2007, July). Comprehensive evaluation of an educational software engineering simulation environment. I*n Software Engineering Education & Training, 2007. CSEET'07. 20th Conference on* (pp. 195-202). IEEE.

21. Paasivaara, M., Heikkilä, V., Lassenius, C., & Toivola, T. (2014, May). Teaching students scrum using LEGO blocks. In *Companion Proceedings of the 36th International Conference on Software Engineering* (pp. 382-391). ACM

22. Jeffry K. Pinto. & Diane H. Parente (2009). SIM4PROJECTS™ A Project Management simulation for Classroom Instruction, from: http://www.simprojectonline.com/documents/ playerquickstart .pdf

23. Jeffry K. Pinto. & Diane H. Parente (2004). SIMPROJECT™ A Project Management Simulation for classroom Instruction, from: http:// http://www.jsu.edu/depart/ccba/featherstone/380/ SimProject_1.2_ PlayersManual.pdf

24. Ramingwong, S., & Ramingwong, L. (2015). Plasticine scrum: An alternative solution for simulating scrum software development. *In Information Science and Applications (pp. 851-858).* Springer Berlin Heidelberg.

25. Rodriguez, G., Soria, Á. & Campo, M. (2015). Virtual Scrum: A teaching aid to introduce undergraduate software engineering students to *scrum*. *Computer Applications in Engineering Education, 23*(1), 147-156.

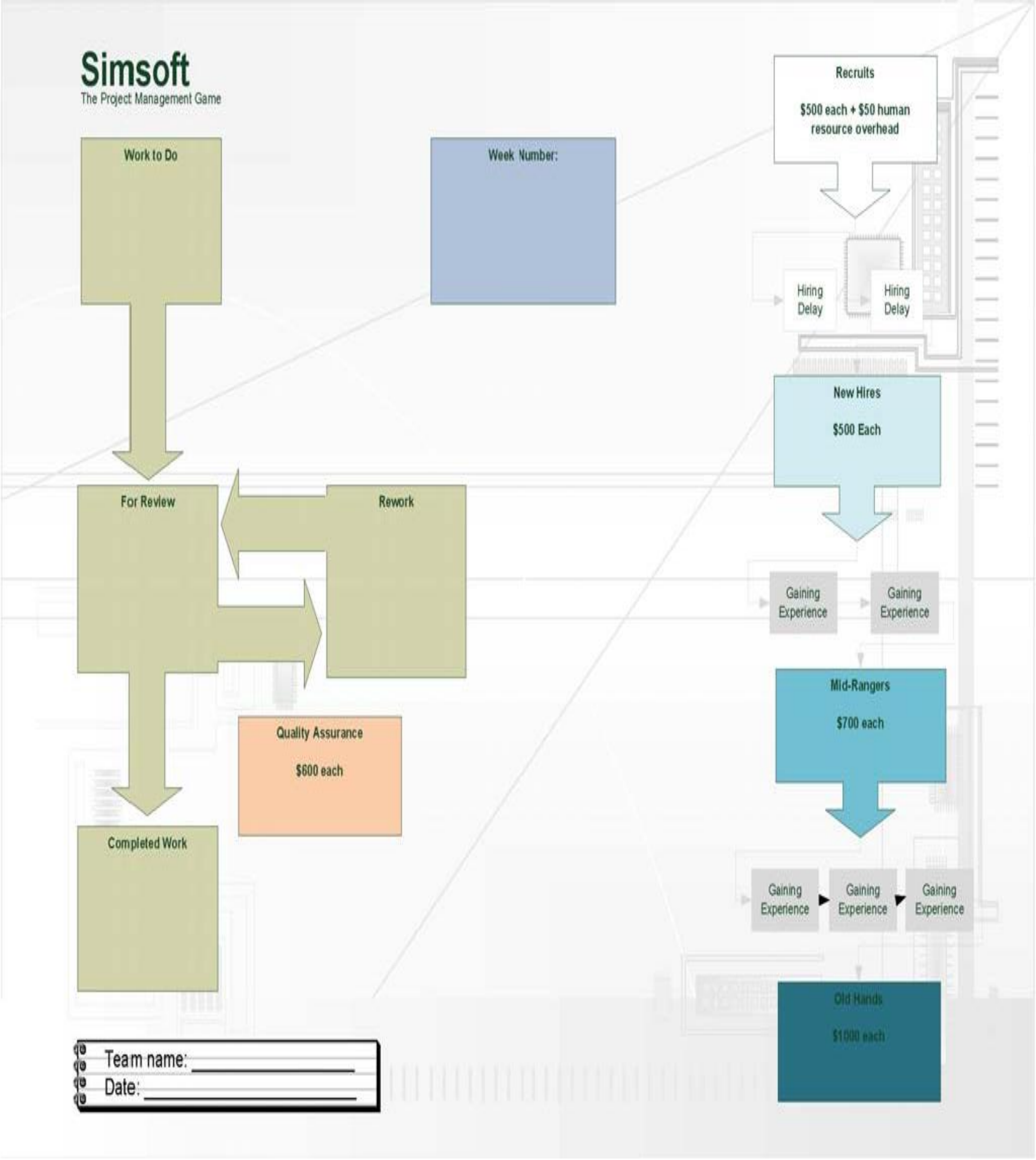
26. Schwaber, K., *Agile Project Management with Scrum*. Redmond, WA: Microsoft Press (2004).

27. Velić, M., Padavić, I., & Dobrović, Ž. (2012, January). Metamodel of agile project management and the process of building with LEGO® bricks. *In 23rd Central European Conference on Information and Intelligent Systems (CECIIS)* (pp. 481-193).

28. Vogel, J. J., Vogel, D. S., Cannon-Bowers, J., Bowers, C. A., Muse, K. & Wright, M. (2006). Computer gaming and interactive simulations for learning: A meta-analysis. *Journal of Educational Computing Research*, *34*(3), 229–24.

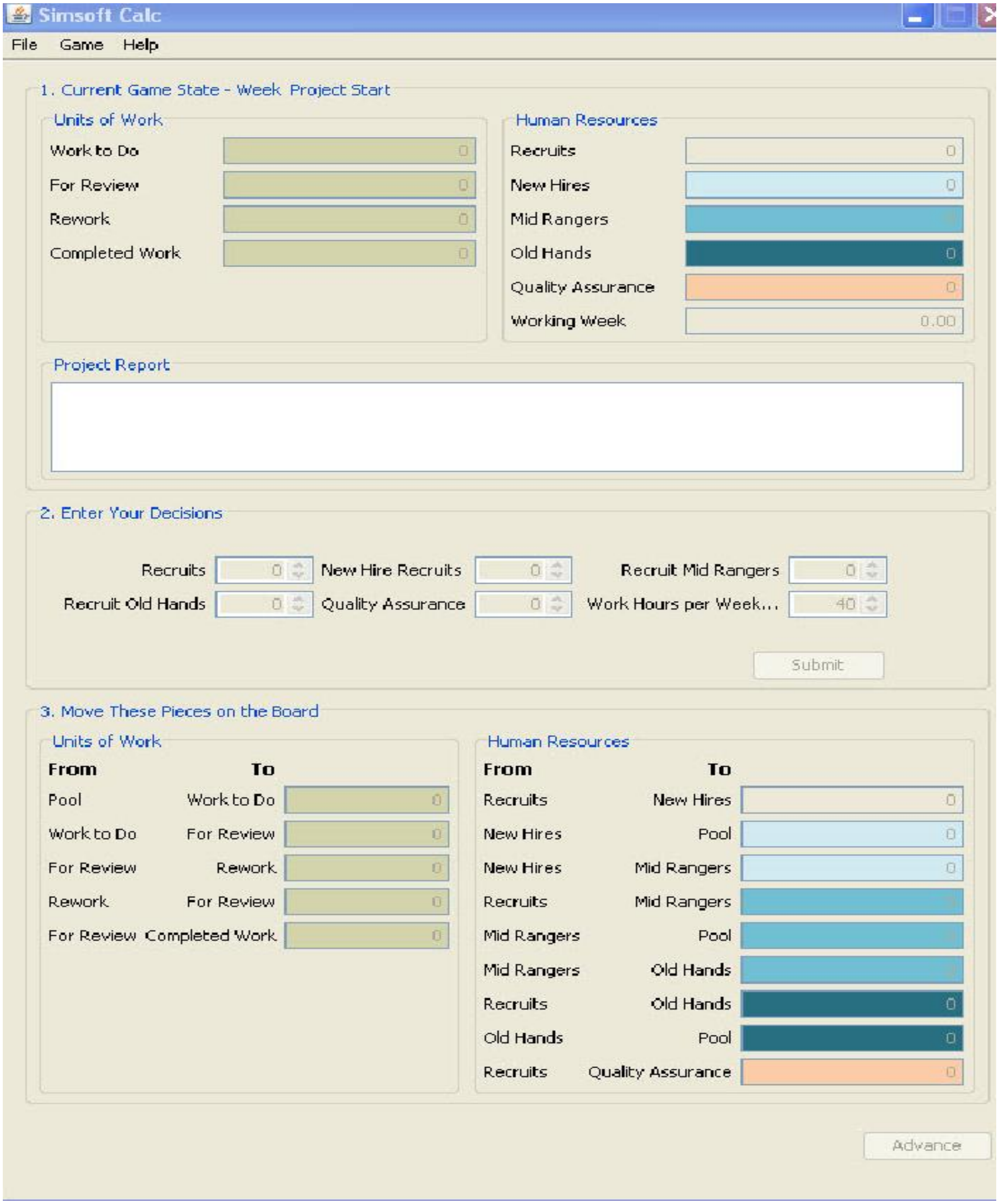
29. Von Wangenheim, C. G., Savi, R., & Borgatto, A. F. (2013). SCRUMIA—an educational game for teaching SCRUM in computing courses. *Journal of Systems and Software, 86(10), 2675-2687.*

Appendix



Simsoft

*Figure **A**: SIMSOFT Board (Caulfield et al, 2011 b)*



1. Current Game State - Week Project Start

Units of Work

Work to Do	0
For Review	0
Rework	0
Completed Work	0

Human Resources

Recruits	0
New Hires	0
Mid Rangers	0
Old Hands	0
Quality Assurance	0
Working Week	0.00

Project Report

2. Enter Your Decisions

Recruits	0	New Hire Recruits	0	Recruit Mid Rangers	0
Recruit Old Hands	0	Quality Assurance	0	Work Hours per Week...	40

3. Move These Pieces on the Board

Units of Work

From	To	
Pool	Work to Do	0
Work to Do	For Review	0
For Review	Rework	0
Rework	For Review	0
For Review	Completed Work	0

Human Resources

From	To	
Recruits	New Hires	0
New Hires	Pool	0
New Hires	Mid Rangers	0
Recruits	Mid Rangers	0
Mid Rangers	Pool	0
Mid Rangers	Old Hands	0
Recruits	Old Hands	0
Old Hands	Pool	0
Recruits	Quality Assurance	0

*Figure **B**: SIMSOFT Dashboard (Caulfield et al, 2011 b)*

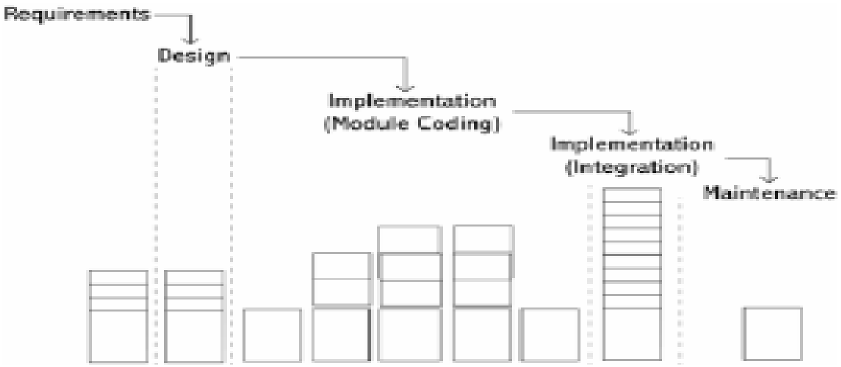
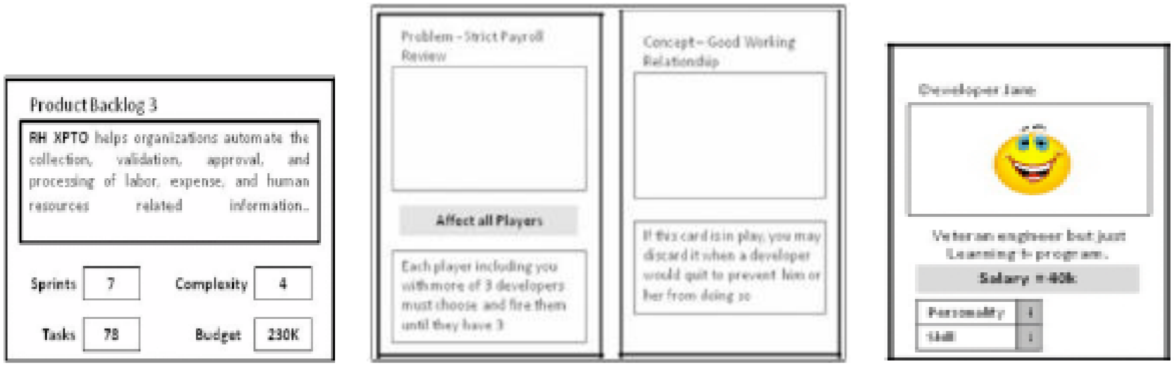


Figure ***C:*** *Phases of Problems and Programmers Game (Baker et al, 2005)*



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

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

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

Veteran engineer but just Learning to program.

Figure ***d:*** *Examples of play scrum cards (Fernandes & Sousa, 2010)*

User Story Priority
P1- critical
P2- high
P3- Medium
P4- Low

Table 1: Scrum Sim (Priority Rating)

Attributes	**M1**	**M2**	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**	**M9**	**M10**
Rate P/H	$70	$60	$55	$55	$50	$40	$35	$40	$40	$65
Skill	5	3	4	3	2	2	2	5	1	4
Personality	2	3	2	2	5	4	3	1	5	3

Table 2: Scrum Sim -Team Selection (member selection attributes)

Game Scenarios.xlsx

List 1: Scrum Sim (problem Scenarios, Opportunity and Review scenarios)

Resource sheet.xlsx

List 2: Scrum Sim (Resource sheet expenditure)

feedback.pdf

List 3: Feedback Questionnaire

List 4: Scrum Sim Manual

SCRUM SIM
DOC.docx

List 5: Video links

1) Scrummage Overview: https://www.youtube.com/watch?v=LY5SLze6_Js

2) Scrum Sim Overview: https://www.youtube.com/watch?v=Qg01JyDeygo

User Story Priority	**Points allotted on Completion**
P1- critical	4
P2- high	3
P3- Medium	2
P4- Low	1

Table 3: Scrum Sim- Priority points calculation

Example: In a group if four teams are competing and assume team Alpha, spends the lowest amount from the allotted budget followed gamma, beta and UV respectively. The scoring is explained in the below table

Team name	**Points Allotted**
Alpha	5
Gamma	4
Beta	3
Gamma	3

Table 4: Scrum Sim- Budget points calculation

Team Name	**Total priority points**	**%age (40)**	**Total value points**	**%age (30)**	**Budget points**	**%age (30)**	**Overall team score**
Alpha	24	9.6	20	6	5	1.5	**17.1**
Gamma	28	11.2	24	7.2	4	1.2	**19.6**
Beta	30	12	22	6.6	3	0.9	**19.5**
UV	26	10.4	18	5.4	3	0.9	**16.7**

Table 5: Scrum Sim- Overall Team score calculation