

# Scrumify: A software game to introduce agile software development methods

Bailey Ammons, Srividya K. Bansal

School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Mesa, AZ 85212 USA  
srividya.bansal@asu.edu

**Abstract:** Software Engineering courses are perceived as being dry and boring. The education community has been looking at active learning and hands-on training techniques to engage students better and thereby provide a better understanding of concepts. This paper presents a web-based game called Scrumify, developed by undergraduate students, that is designed to introduce students to the Scrum software development process through story elements, and quiz mechanics. The game is designed to reinforce software engineering concepts more specifically agile software processes. Quizzes within the game are designed to grant points that inject a sense of competition for the users and motivation to learn the agile methodology concepts. Scrumify serves two objectives: software engineering education for audience of the game and web-based game development and software testing education for the developers of the game. Use of this game in classroom provides an active learning technique for agile software processes. Students are more engaged with games and thereby absorb the concepts better. The game was evaluated in a sophomore software engineering class and the results are presented.

**Keywords:** Software Engineering Education, Scrum, Agile methods

## 1. Introduction

Traditional Software Engineering courses mostly consist of theoretical lectures that are perceived as boring and dry. Students are not engaged actively thereby resulting in students not learning the key concepts and thereby are not well prepared for the industry after graduation. Engaging the users in an active learning environment through mechanism that students naturally gravitate to is of great value. This paper presents the design, implementation, and evaluation of a web-based game called Scrumify that is designed to teach Scrum – an agile software development process. The game is designed with 4 levels and various concepts and terminology from Scrum are introduced in each level as the user progresses through the game. After going through all the levels, students gain a good understanding of the Scrum process. Learning is achieved at the Remember and Understand learning levels of the Bloom's taxonomy within the Cognitive domain. The game covers all aspects of an Agile Scrum process, describes the basic unit of Scrum called sprint, describes sprint cycle and

its duration, and demonstrates how a project is implemented in sprints.

This paper presents the prototype implementation of Scrumify, a web-based game that is designed to have 5 levels. As the game user progresses through the different levels various Scrum process concepts are revealed. There are quizzes at the end of each level that will determine if a user can proceed to the next level. Usability testing and evaluation of the game are presented.

The remainder of this paper is organized as follows: Section 2 provides background material on Agile processes. Section 3 presents related work in Software Engineering Education. Section 4 presents the System design of Scrumify followed by implementation in section 5. Section 6 presents the testing and qualitative evaluation of the web-based game and its results. The last section presents summary and future work.

## 2. Background

Software Development Lifecycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project, from an initial feasibility study through maintenance of the completed application [1]. There are many different types of SDLC models, and industry uses one that best suits their needs. Agile is an iterative and incremental development methodology where requirements and solutions evolve through collaboration between cross-functional and self-organizing teams [1].

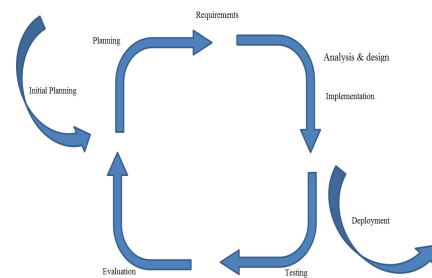


Figure 1: Agile process

### A. Agile Process

The Agile Manifesto focuses on following: individuals and interactions; working software; customer collaboration; and responding to change. Figure 1 depicts an Agile process. Different kinds of Agile processes are as follows:

- *Agile Unified Process (AUP)* is a simple and easy to understand process to develop business application

software. It includes agile techniques such as test-driven development, agile modeling, agile change management, and database refactoring for higher productivity.

- *Kanban* is a just-in-time delivery process that does not overload developers. It uses visual process management that provides information on what is to be produced, when it is produced, and how much is produced.
- *Scrum* is a popular framework derived from agile development that provides a flexible and holistic strategy where the entire team works as a unit towards one common goal.

### B. Scrum

Scrum is an agile software development framework in which its key goal is to allow flexibility while rapidly developing software. It also introduces people specifically focused on adhering to the Scrum process, and perpetuate good practices. It is an iterative and incremental process where the *product* that is being developed known as the *product backlog* is described and discussed among the team members. The product backlog is divided into smaller tasks that are assigned to the team members. The set of tasks that the team works on during a sprint is called the *sprint backlog*. A physical board called the *scrum board* is used to keep track of all these tasks and assignments. The roles in the Scrum framework are as follows:

- *Product Owner* is responsible for the whole product idea, manages the return on investment (ROI) for the effort by the team, keeps track of prioritization of the product backlog & release plans, and could also act as a team member.
- *Scrum master* promotes Scrum process, supports to resolve any impediments, makes a team self-organized, keeps the sprint backlog visible, protects the team from external interference and disturbances to get along with the flow of work, and has no authority on the team. The scrum master is not the team leader, but acts as a buffer between the team and any distracting influences. The Scrum Master ensures that the Scrum process is used as intended. He holds meetings to ensure the process is running properly.
- *Team member* is cross-functional who possess skills of a tester, business analyst and not just a developer and strongly collaborates with other team members.

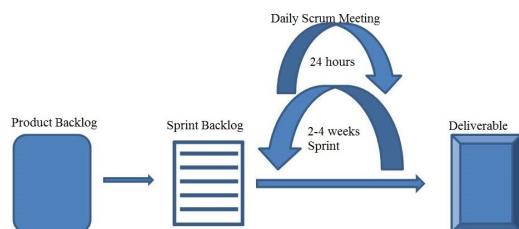


Figure 2 Sprint Cycle

### C. Sprint

The scrum process consists of all members contributing to the development and implementation of the product in a

specific period of time called a *sprint*. Sprint is the basic unit of development in Scrum. It is restricted to a specific duration that can last anywhere from a week to a month. A sprint starts with a planning meeting where product under development is discussed and the sprint tasks are derived. They are divided and assigned to each member of the team. A *daily scrum* meeting also known as a *standup* meeting is held every day where all members of the team provide a status update of their tasks that were performed on the previous day, the tasks that they are currently working on, and also inform the team of any impediments that they are facing in completing their tasks. This meeting has specific guidelines which during each meeting, the team spends time on product backlog refinement which will aid in the next sprint. This backlog is an ordered list of requirements for a product. This can contain features, bug fixes, estimations, non-functional requirements, business value, and development effort. Figure 2 shows a pictorial representation of a sprint cycle.

### 3. Related Work

The use of simulation games for education has become quite widespread in the last few years [2-4]. Most of the related work have established and published results that students found simulation games for education useful in learning and helped students in being more engaged in class. Although games have been used to teach various concepts in a number of fields, there are very few games that teach concepts in Software engineering. In this section we present related work on simulation games for teaching software engineering and other related literature that provided inspiration for the design and implementation of Scrumify. Researchers have worked on simulation games to teach software engineering processes through an experimental card game that highlights process issues that were not sufficiently addressed in lectures and projects [5]. This game uses physical cards to reinforce software engineering concepts unlike Scrumify that is a web-based educational game.

SimSE [6] is another game-based software engineering simulation environment that provides a number of games to teach various software development lifecycle models such as waterfall model, incremental model, rapid prototyping model, rational unified process, and extreme programming model. SimSE does not teach Agile software development that is the most recent and popular model. ScrumTutor [14] addresses Agile processes and draws inspiration from SimSE that creates a game scenario with an office background and team members working in an office thereby simulating a real-world environment.

Students were introduced to Software Engineering Interview process through a decision-based computer game in a study at Rowan University [7]. Students learnt about interview process that is needed in requirements analysis phase of software engineering life cycle. By playing the game multiple times students realized that gathering facts before design and implementation is important for the success of the project. Based on this work, Scrumify was designed to have multiple levels so that concepts can be reinforced to students to make sure they attain a high level

of understanding.

Researchers have also used the concept of games for teaching various management concepts such as Risk management and knowledge management [8, 9]. They assessed the effectiveness in meeting learning objectives and their findings clearly demonstrated the advantage of a simulation game. Other related work includes evaluation of object-oriented design patterns in game development [10] and seniors rehabilitation using video game technology [11]. These projects are close to education through interactive tools and games but are in other domains. We believe Scrumify will be a useful tool to the Software Engineering Education community and can be customized to teach several other Agile Process concepts.

#### 4. Scrumify – Design

The main objective of the game is to get as many points as possible while killing enemies, picking up skulls and pages, and answering each quiz to the best of game user's ability. The main characters in the game are: (i) Scrum Princess: a scene character who pops up in text scenes; (ii) Scrum Knight: the main character/hero who has a sword that he uses to attack, he can interact with other characters, and he can interact with objects in the game; (iii) Scrum Lord: antagonist and the final boss in the game; (iv) Scrum Lord Minions: basic characters that bar way along the game; (v) Friendly Scrum Knight Phantom: a friendly phantom who pops up occasionally to help guide the Scrum Knight through the game. Starting with level one the player is introduced to the controls and the main objective of the game. The main purpose is to collect all scrum pages (which provide information to the user about the scrum process). Some of the pages are laying around each level while others reside in various chests. To gain access to the pages in the chests, the user must take a quick quiz and answer each question to the best of their ability. Irrespective of the quiz score the user will get the page from the chest, but if they do not get 100% on the quiz, the user will loose some health. Each page obtained will help the game user diminish the possibility of loosing health because they provide key information that will help them answer each chest quiz. If the player looses all three hearts of their health, they must restart the level. Another entity that can prevent the user from progressing to the next level is a series of enemies. Unlike the chests, the enemies are not to aid the user with level progression and answering quizzes, but to hinder the progression of the level by killing the user and restarting the level. The four types of enemies are: (i) a fire ball throwing enemy that chucks fireballs at the user; (ii) a punching enemy that tries to punch the user; (iii) a sprinting enemy, who attacks the user by running into them; (iv) the final boss, which is a combination of the three other enemies. Each level has each type of enemy and chests. In the first two levels the player can only move, jump and attack while in the last three levels the user learns how to wall jump. Wall jumping gives the user the ability to reach new heights by bouncing off walls. Each of the first four levels contain chests and enemies and have the purpose of teaching the game user the scrum process. As

for the last level, this puts an end to the story line of the game. The game's main objective is to teach scrum, but the story line is that the game user is a knight out on a quest to save the princess who was kidnapped by the scrum lord. After navigating through each level, the user will enter the final level, which has the boss fight. Once the boss dies and explains what happened to the princess, the game ends.

##### A. Level 1 (Marshy Overgrown castle)

The objectives of this level are as follows: (i) introduce player to controls; (ii) obtain level pages about scrum; (iii) teach the player the game controls and tell them the importance of reading and understanding the pages they collect; (iv) roles and sprints. The Scrum Knight is on a quest to save his kingdom from destruction, where he meets the Friendly Scrum Knight Phantom. The Scrum Knight can save his kingdom by retrieving all the pages from the Scrum Book and after proving his usefulness by giving the Knight instructions the Phantom joins the game user on their quest. The castle seems unstable, so the Scrum Knight states that they must hurry. After finding first page the Phantom recommends that the player read these pages thoroughly and explore the entire area because these may help in their quest.

##### B. Level 2 (Desert Landscape)

The objectives of this level are as follows: (i) player collects more pages about scrum; (ii) introduce **Sprint** ability to player; (iii) introduce concepts of Sprint meetings. The Phantom warns the Scrum Knight to be careful in this place. After reaching the end of this dungeon, the Phantom states that there are no more pages to be found and that he is bored of this place.

##### C. Level 3 (Ice cave)

The objectives of this level are as follows: (i) player collects the final pages about scrum; (ii) introduce **Double Jump** to player; (iii) introduce Scrum Artifacts; (iv) player revisits the princess; (v) princess uses magic to combine pages; (vi) opening a portal to the Scrum Lord. After arguing about the Scrum Knight's clothing choice, the Knight reveals that his quest was given to him by the princess herself. There is also a 4th wall-breaking scene where the Phantom refers to the player that the Knight is oblivious to. The Phantom requests a question from the Knight and he denies it, thus ending the level.

##### D. Level 4 (Dark portal to Scrum lord)

The objectives of this level are as follows: (i) defeat the Scrum Lord; (ii) demonstrate knowledge of the Scrum process; (iii) complete the story. The Phantom notes how 'cozy' this level is and then they banter on how the Phantom won't shut up. The Phantom also reminds the Scrum Knight that he will need double-jumping for this level. There are no other interactions in this level.

##### E. Boss Level

The Scrum Lord reveals that it was him all along and he killed the king and princess for the Scrum book that the

princess destroyed shortly before her death. He would have died if he collected it himself, so he made the knight do it for him.

The story can be modified for depth. This initial prototype has just enough content to spur a user forward from the beginning of the game, and giving closure at the

end of the game. The main goal is to teach Scrum to the game user. This educational game would be more effective if the education was baked into the storyline, possibly teaching by metaphor or simile.



**Figure 3: User-interface of Scrumify**

## 5. Scrumify - Implementation

The game was implemented as a web-based game through JavaScript and web hosting software to either local or external servers. This game can be hosted on a server so that players can play the game online. The overall implementation of the game is dependent on two main tools, the game engine and the level engine. Impact, a javascript gaming engine, allows for HTML5 games for desktop and mobile browsers and was used in this project [12]. The level engine, Weltmeister [13], was used to help build the level and add entities, which comprise of enemies, blocks, doors, chests, scene triggers (which trigger cut scenes), and collision objects, to the desired location.

From there, the game engine helped with game physics (such as momentum and gravity), collision (enemy attack collision, player attack collision, scene trigger collision, block collision (which helped the player navigate the map without falling through the world), and chest collision). With both these tools, we were able to enhance the game by using objects that can collide, move, or be controlled. Figure 3 shows a screenshot of the game.

## 6. Testing and Evaluation

### A. Testing

Testing of this game was conducted through a sophomore level course on Software Quality and Testing. The students in the class had to understand this code base, come up with a software test plan, execute the test plan, and identify errors. Usability testing of the game was also conducted in a sophomore Software Engineering class with 30 students. The following list of bugs were identified and fixed.

- Endless loop in game dialog such as talking to skeleton.

- Chests would not drop Scrum letter when completed and/or cause an endless loop resulting in either the player dying or getting a large score.
- Character movement when in dialog, even though the screen did not update, the character was still able to move. This bug would also lead to a potential bug where the character becomes uncontrollable after the scene, resulting in the user to restart the game.
- Game will not load levels past level 2 properly.
- Game allowed player to wall jump on any surface, including the ceiling.
- Game allowed player to wall jump on levels not intended for the player to have this ability.
- Certain portions with provided level maps did not have a collision layer on edges of the world, this resulted in player being able to jump outside of the map and fall endlessly.
- Enemy Boss on final level would not always spawn if cut scene were skipped too quickly.
- When the player died, the game remained idle without the player spawning to get a second chance to retry the level.
- Enemies could instantly kill the player if player collided with them due to not having a timer between hits.
- Player was able to avoid important cut scenes by jumping around them.

The Software testing team did thorough testing and made a number of adjustments to the game as follows:

- Re-implemented the chest system to allow users to simply fail a test and forgo points missed, because testing repetitively does not ensure understanding, rather reinforcing guessing.
- Fixed the wall-jump system so that it cannot be easily exploited to climb walls from the left hand side.

- Removed wall-jump capabilities in the first two levels per game design choice that was not initially implemented.
- Added enemies/visuals to make the game more intuitive/appealing to the users.
- Added a respawn function so that the user doesn't have to restart the whole game.
- Adjusted levels to prevent sliding through diagonal surfaces.
- Included/redid text throughout messages in the game to make the instructions more obvious and look less sloppy.
- Added elements to levels based on first-level design to encourage point capture for the user.
- Made enemies only capable of dealing damage after a short period of time has passed from the last attack to prevent instant death.

#### B. Evaluation

The game was evaluated in a class of 30 students at the sophomore level. For the overall evaluation of the game the following evaluation instruments were used:

- Quizzes at each level of the game
- Pre-evaluation survey (included questions on demographic and Scrum process)
- Post-evaluation survey (included questions on Game mechanics and Scrum process)

Table 1, 2, and 3 show the questions used in the survey.

**Table 1: Quiz questions in Scrumify**

Level (Scrum concepts introduced)	Questions
1. Product Owner, Development Team, Scrum Master, Sprint	What is the responsibility of the Scrum Master? Is the Scrum Master the team leader? Is the Scrum Master the same as the Project Manager?
2. Daily Scrum, Backlog Refinement, Sprint Planning meeting, End of Cycle	Check the three questions that every team member answers during a meeting. <ul style="list-style-type: none"> <li>• How long should a meeting be?</li> <li>• Where should the meeting happen everyday?</li> <li>• Which of these is normally after a Daily Scrum?</li> <li>• What questions are on the agenda?</li> </ul> Select the objectives that are in a sprint-planning meeting. When should a sprint planning meeting be held?
3 Product Backlog, Sprint Backlog, Artifacts, Burndown	Which of these are in a sprint review meeting? What should be done at the sprint retrospective? Product backlog is an ordered list of _____ for a product. Check what a product backlog can contain. Items are ordered by the product owner based what considerations? What is a sprint dialog? What are features broken down into? Tasks are assigned for the next sprint. What is a sprint burn down?

**Table 2: Pre-evaluation Survey**

#	Question	Answer Choices
1	What is the highest degree you have completed? (if currently enrolled, mark the highest degree received)	High school diploma or equivalent Some college, Bachelors, Masters Doctorate, None of the above
2	What is your field of study? (e.g. Software Engineer, Pre-med, etc.)	Open-ended

3	Your gender	Male, Female, Other
4	Your age group?	12-17, 18-24, 25-29, 30-34, 35-39, 40-49, 50-59, 60+
5	Your race?	American Indian or Alaska Native, Asian, African American, Native Hawaiian or Pacific Islander, White, Hispanic or Latino, Other
6	What best describes your knowledge as a Software Engineer?	<ul style="list-style-type: none"> <li>• Savvy software developer</li> <li>• Beginning software developer</li> <li>• Know the concepts of software development but don't code</li> <li>• Don't know anything about software</li> </ul>
7	What type of gamer would you describe yourself as?	<ul style="list-style-type: none"> <li>• Hardcore Gamer</li> <li>• Casual Gamer</li> <li>• I don't play video games</li> </ul>
8	How Familiar are you with the Scrum Process? (Choose which best suits your level of understanding)	<ul style="list-style-type: none"> <li>• Know about and understand the process</li> <li>• Know about the process</li> <li>• Heard of it but unfamiliar with the process</li> <li>• Unaware of it</li> </ul>
9	Please give your best description of the scrum process.	Open-ended

**Table 3: Post-evaluation survey**

#	Question	Answer Choices
1	Please provide a description of the purpose of the game	Open ended
2	What is your current level of understanding of the Scrum process	<ul style="list-style-type: none"> <li>• Understand the process and purpose</li> <li>• Understand the purpose</li> <li>• Aware of its existence</li> <li>• Did not know, it was described in game</li> <li>• What is Scrum? Ignored the story of the game, only played it</li> </ul>
3	Was game difficult to play?	Very Simple, Easy, Moderate, Somewhat difficult, Hard, Impossible
4	Please list issues identified with player movement, if any	Open-ended
5	Please list issues identified with level 1, if any	Open-ended
6	Please list issues identified with level 2, if any	Open-ended
7	Please list suggestions about improving game play	Open-ended
8	Who does the product owner represent?	Stakeholders, Customers, The Project team, The Scrum master
9	What is the range of team size on a Scrum Team?	14, 12, 10, 7
10	Is the Scrum Master the team leader?	True, False
11	What is a sprint?	<ul style="list-style-type: none"> <li>• A basic unit of development in Scrum</li> <li>• Running really fast</li> <li>• A meeting in Scrum</li> <li>• Where the developers code extremely fast</li> </ul>
12	Which one of these is not normally after a Daily Scrum?	<ul style="list-style-type: none"> <li>• Discussing your work</li> <li>• Sending an appointed member of your team</li> <li>• Restarting the meeting</li> </ul>

13	What is the responsibility of the Scrum Master?	<ul style="list-style-type: none"> <li>• Rush the team through development</li> <li>• Remove all impediments for the team</li> <li>• To make sure everyone knows he/she is the master</li> </ul>
----	---	--

### C. Analysis of the results

The data from the pre-survey showed that most participants were Software Engineering students and beginning programmers as they were in their 2<sup>nd</sup> year of degree program. Most of the participants were in the 18-24 age group with some college experience. Majority of them said that they were casual gamers. Majority of the participants answered that they are unaware of the Scrum process. In the post-evaluation survey users provided valuable feedback about the game that have been incorporated into the list of enhancements for future work (listed in the following section). Regarding the difficulty of the game, most users answered that the game was somewhat difficult or hard to play. So the game story and game play need to be improved in order to provide a better user experience. 75-80% of the participants were able to answer the questions about the Scrum process correctly thereby demonstrating that Scrumify helped provide better learning for students and active learning technique for instructors. Others were the ones who struggled to get through the game because of some glitches in the game or the user experience not being as effective as it can be.

### 7. Conclusions & Future Work

The main goal of this project is to teach Scrum (a agile software development process) to the game user at undergraduate sophomore level (2<sup>nd</sup> year students). This educational game development involved an undergraduate student for development of the game there by teaching web-based game development concepts. A small team of undergraduate students enrolled in the Honor's program were involved in Software testing of this game thereby learning concepts of Software test plan development and execution. This educational game would be more effective if the education was baked into the storyline, possibly teaching by metaphor or simile. Story can be further modified for depth. This initial prototype has just enough content to spur a user forward from the beginning of the game, and giving closure at the end of the game. A number of enhancements were identified during the evaluation phase of the project. Bringing on visual artists and storywriters on the team might best handle development of the game visual elements so the game can be more enjoyable and usable. As future work on this project, the following enhancements are planned:

- Pause-menu that allows a user to traverse previously collected pages, adjust audio options, re-assign game controls, and save/restart progress on levels.
- Visual enhancements could be made to make the game to make it more interesting (just more overall diversity).
- Enhancing the difficulty to scale with level-progress.
- Including more story-elements and levels to play.

- Provide feedback to user on their quiz responses.
- Improving the player movement so that his momentum is less of a learning curve to new players.
- Making the double-jump and wall-jump not focus on the player-box's left corner so that user can't exploit these features. Ideally these features would focus on feet, and left or right shoulders respectively of the character.
- A scoreboard that records the players points, to encourage competitive play.
- Making health persist through level transitions, and some way to earn life back.
- Perfect the scene-drawing system so that there is no screen pausing made when the player enters a screen, this leads to player-confusion and frustration.
- Ability to put an end to the game. Once the boss is dead, the game does not.
- Final scores can be provided to the user at the end of the game. Currently it is a value displayed at the top of the screen. Score can be used and stored against other players to make the game more competitive and give a larger appeal for the player to take the chest quizzes and progress to next level if user attempted all chest quizzes.
- Levels can be made more visually appealing. Each level has a generic color background with some decorative objects, but more of those objects should be added as well as a non-single color background. Some potential backgrounds could include, for ice level, an ice block appeal to make the user feel as if they are in an ice castle.

### Acknowledgement

The authors wish to thank Nick Carney, Jacob Dobkins, Justin Riegel, and Andrew West for their work on usability testing of the game. We also thank all participants involved in evaluation of the game.

### References

- [1] I. Sommerville, "Software Engineering" (9<sup>th</sup> ed.), *Chapter 2: Software processes: Software process models and Chapter 3: Agile Software Development*, Boston, MA, 2011.
- [2] J. Pieper, "Learning Software Engineering Processes through Playing Games," IEEE Intl. Workshop on Games and Software Engineering (GAS) at Intl. Conf. on Software Engineering, June 2012.
- [3] N. Tillman, et al, "Pex4Fun: Teaching and Learning Computer Science via Social Gaming", Conf. on Software Engg. Education & Training (CSEET) at Intl. Conf. on Software Engg., June 2012.
- [4] D. Ismailovic, et al, "Adaptive Serious Game Development", IEEE International Workshop on Games and Software Engineering (GAS) at International Conference on Software Engineering, June 2012.
- [5] A. Baker, E. O. Navarro, A. Hoek, "An experimental card game for teaching software engineering processes", *Journal of Systems and Software*, IEEE, pp. 3-16, 2005.
- [6] E. O. Navarro, "SimSE: A software engineering simulation environment for software process education", *Doctoral Dissertation*, School of Information & Computer Sciences, UC Irvine, 2006.
- [7] A. Rusu, R. Russell, R. Cocco, "Simulating the Software Engineering Interview Process using a Decision-based Serious Computer Game", *IEEE Intl. Conference on Computer Games (CGAMES)*, 2011.
- [8] A. Chua, "The Design and Implementation of a Simulation Game for Teaching Knowledge Management", *Journal of American Society for Information Science & Technology*, pp. 1207-1216, 2005.
- [9] G. Taran, "Using Games in Software Engineering Education to Teach Risk Management", *IEEE Conf. on Software Engineering Education & Training (CSEET)*, pp. 211-220, 2007.

- [10] A. Ampatzoglou, A. Chatzigeorgiou, "Evaluation of object-oriented design patterns in game development", Elsevier Informaiton and Software Technology, Vol. 49, pp. 445-454, 2007.
- [11] D. Maggiorini, L. A. Ripamonti, E. Zanon, "Supporting Seniors Rehabilitation through Videogame Technology", Workshop on Games & Software Engg. (GAS) at Intl. Conf. on Soft. Engg, 2012.
- [12] "Impact: Javascript engine" [Online]. Available: <http://impactjs.com> [Accessed: Oct-16]
- [13] "Weltmeister: Level Editor"[Online]. Available: <http://impactjs.com/documentation/weltmeister> [Accessed: Oct-16]
- [14] S. Potineni, S. Bansal, A. Amresh. "ScrumTutor: Web-based Interactive Tutorial for Scrum Software Development". In IEEE ICACCI: Intl. Symposium on Women in Computing and Informatics (WCI), 2013.