ScrumTutor: A Web-based Interactive Tutorial For Scrum Software Development

Sindhura Potineni, Srividya K Bansal, Ashish Amresh
Department of Engineering
Arizona State University – Poly Campus
Mesa, AZ 85212 USA
{sindhura.potineni, srividya.bansal, amresh}@asu.edu

Abstract— **In a traditional software engineering class, students are typically engaged in theoretical lectures followed by homework assignments or a project. Use of hands-on training and laboratory activities using real-world projects is more likely to produce students with a higher level of achievement and more confidence in the course material. If every topic or technique introduced in the course has a corresponding hands-on activity that demonstrates an application or use of the concept in the industry, students better understand the need for the technique and the learning environment is more interactive, engaging, and interesting to students. This paper presents a project called ScrumTutor that aims at providing an engaging learning experience of Scrum Software development process through a web-based Interactive tutorial. It is designed and developed for undergraduate students in introductory Software Engineering courses. This software tool introduces its users to modern software engineering methodology used these days in the software industry known as *Agile Software Development* that includes the Scrum framework for managing software projects or products.**

Keywords*— *Software Engineering Education, Agile Software Development, Scrum, Web-based interactive tutorial.

I. Introduction

Software Engineering courses are perceived as dry and boring. Traditional software engineering classes mostly consist of theoretical lectures and that do not engage students actively thereby resulting in students not learning key concepts [2, 5]. Therefore application of those key concepts in real-world scenarios is important for student learning. One can gain more knowledge and retain it only when (s)he is able to implement it or use it in a project that they might work on. An interactive tutorial designed to engage a user in active learning of software engineering concepts is of value. The requirements of this tutorial would be: i) introduce user to a course topic through the tutorial; ii) provide information on all the basic concepts and key terms of the topic; iii) allow users to practice what they have learnt in a user-friendly and engaging manner.

The goal of this project is the design and implementation of a web-based interactive game to teach Scrum framework for managing software development. Interaction with the user is divided in 3 significant phases where the user plays three different roles in each phase. For example, in the Scrum process there are multiple roles such as a product owner, scrum master, developer, knowledge manager, etc. The first phase is 'Observation'; here the user will observe the Agile process in action. The second phase is 'Data collection'; here the user will use the techniques observed in phase I and contribute towards data collection part of the implementation of the project. Finally the third phase is the 'Development' phase. Here the user performs a developer role on the team. The user picks one of the available tasks for development and implements it. The aim is to successfully implement all the assigned tasks. After going through all three phases the student would gain hands-on experience in executing a Scrum process. The interactive tutorial covers all aspects of an Agile Scrum process, describes the basic unit of Scrum called sprint, describes a sprint cycle and its duration, and demonstrates how a project is implemented in sprints.

This paper presents the prototype implementation of the ScrumTutor as an interactive tutorial with 2 phases. In phase I the user plays the role of an observer. The sprint is one week-long and the tutorial depicts activities from day one to day five. The tutorial involves four roles and uses a specific software project that involves developing a website to manage music, music albums, artists, and events using a Content Management System such as Drupal [13]. The Scrum roles simulated in the tutorial are a Product owner, a Scrum Master, and two team members. In later phases of the tutorial the user gets involved as a developer on the team. In the first phase, the team starts a weekly sprint by discussing the product that they are going to develop at a sprint Planning meeting led by the product owner. Each day of the sprint, the team has a daily standup meeting to discuss the tasks each member worked on yesterday, what they are going to work on today and if they have any impediments in completing their tasks. At midpoint during the weekly sprint, the team also has a Scrum review meeting where they discuss all the sprint tasks in detail, their status, and if any team member needs help on any task. In phase II, the user plays the role of a data collector and participates as one of the team member. The user is involved in the process and daily Scrum meetings where (s)he provides their status update and also collects data required to handle certain testing tasks for the team. The player gets involved in the project thereby gaining better understanding of the Scrum process.

978-1-4673-6217-7/13/$31.00 ©2013 IEEE

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 ScrumTutor followed by implementation in section 5. Section 6 presents a qualitative evaluation of the software tool and its results. The last section presents summary and future work.

II. 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. The basic activities in any SDLC are the same. Each software lifecycle model specifies phases of the lifecycle and the order in which they will be executed. Initially gathered requirements are converted to basic design. Code is produced in the process of implementation that is evolved from the design. Deliverables from the implementation phase are tested against the requirements. Some of the traditional SDLC models are: 1) Waterfall – a linear-sequential lifecycle model that is very simple and easy to understand and use; 2) V Shaped - an extension of the waterfall model where testing starts early-on in the process; 3) Spiral - more favorable for large, expensive and complicated projects due to it extra emphasis on planning, risk assessment, and design of prototypes and simulations; 4) Agile – an iterative and incremental development methodology where requirements and solutions evolve through collaboration between cross-functional and self-organizing teams [1].

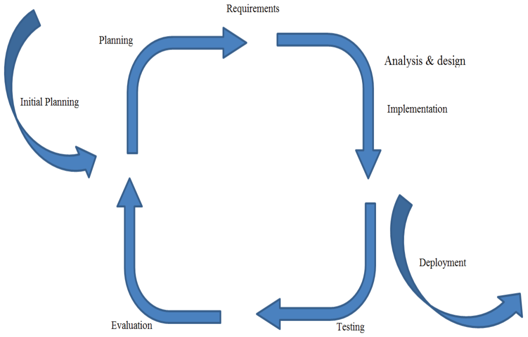


Figure 1: Agile process

A. Agile Process

The Agile Manifesto focuses on the following: individuals and interactions; working software; customer collaboration; and responding to change. Figure 1 depicts an Agile process. Some of the variations of an Agile process 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 the 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 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.
- *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.

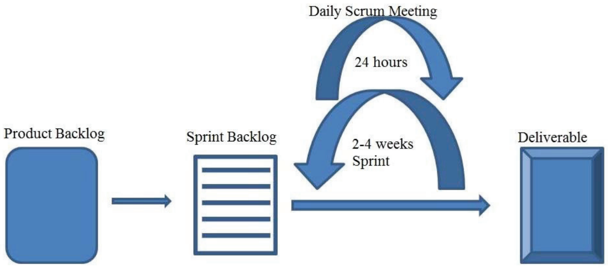


Figure 2 Sprint Cycle

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. Figure 3 shows a pictorial representation of a sprint cycle.

III. 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 ScrumTutor.

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 ScrumTutor that is a web-based interactive tutorial.

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 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. ScrumTutor has used this aspect in its design and addresses the area of Scrum that has not been addressed in SimSE.

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, ScrumTutor was designed to have multiple phases 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 ScrumTutor will be a useful tool to the Software Engineering Education community and can be customized to teach several other Agile Process concepts.

IV. Scrum Tutor – Design

The long-term goal of the project is to design Scrum Tutor as a simulation game with game features such as engaging user interaction, time factors, adjustability of player skill levels, re-playability, scoring, multi-player, and providing score statistics. The first version of ScrumTutor was designed with this end goal in mind.

A. What is being taught and how?

The initial version of ScrumTutor provides hands-on practical experience to students in a software engineering class. Students learn Scrum process by engaging with this tool as an observer and assimilating the flow of process and management of software development. The tutorial takes its user through a weeklong sprint with a Scrum team working on various development activities distributed from day 1 to day 5. On day 1 the user observes the product owner describing the product and the product backlog to the software team. The user is introduced to various concepts such as the Scrum board that displays the sprint backlog. User watches scrum task allocation happening during a sprint planning meeting, discussion of tasks and allocation among team members at the scrum board, and discussion on status of various activities through a daily standup meeting in front of the Scrum board. The tutorial has a pre-defined simulated time assigned to each day. When the pre-defined time for a day lapses, the user then progresses to day 2 and observes a daily scrum meeting held to discuss the activities of the team members performed on that day and on previous day, to get the updates from the team, and also discuss any impediment's that a team member might be facing. This process is repeated for the remainder of the days. The user observes these daily standup meetings happening at a fixed time everyday, example at 9:00am in the morning. During the remainder of the day the user observes the team working on their tasks in their cubicles and interacting with each other. On day 4, a sprint review meeting is held to discuss the progress made during the current sprint and the team assesses if there is a need to make any changes to their plan in order to successfully complete their tasks and deliver the artifacts. In this simulation, the sprint review meeting happens at around mid-point in the sprint at a fixed time, example 11:00am on day 4.

In order to encourage active learning and engagement, as the user is progressing from one day to the next, a number of quiz-type questions are presented to the user to test their understanding of the concepts. The user is allowed to process to the next day only after answering these questions correctly. If not, the user is redirected back to the previous step to observe the process and understand it.

ScrumTutor is designed to have multiple phases so that the user can more actively engage in the process and gain hands-on experience as they are learning and getting better at Scrum. This version of the tool is designed to have 2 phases. Phase 1 is the observation phase that was presented so far. Phase 2 is the data collection phase where the user is taken through a new sprint with a new set of tasks in the sprint backlog and the involvement of the user progressively increases. The user is involved as one of the team members participating in the meetings, providing inputs, and status updates. Tasks are allocated to the user that has to be completed for a successful completion of the sprint.

B. What product is being developed using the Scrum framework in ScrumTutor?

This software tool is designed such that any project/product can be used as an example and plugged in to demonstrate the Scrum framework, sprints, and various activities. In this version of the tool, the product being developed is a website to manage music, music albums, artists, events, and venues. This product is built using a Content Management System called Drupal. An existing Software requirements specification and product backlog of this product was used to demonstrate the process in Scrum Tutor.

Table 1: Sample formative & summative assessment questions

#	*Question*	*Answer Choices (correct answers in bold)*
1	Is the sprint backlog created during the planning meeting?	- **Yes** - No
2	How often do the teams integrate their work and rerun the regression tests?	- Only at the end of each sprint - Once per day - **Continuously as things change; potentially many times per day**
3	What happens during a scrum review meeting?	- **Discussion of sprint progress** - **Check if any member of the team needs help to complete their tasks** - Remove tasks from sprint backlog - Discuss a team outing
4	What is the duration of a daily scrum meeting?	- As long as necessary - 1 hour - **15 minutes**
5	Who describes the product backlog to the team?	- **Product owner** - Scrum Master - Knowledge Manager - Quality Manager
6	When is a sprint retrospective meeting be held?	- **At the end of each sprint** - It is not needed - **Few minutes before the sprint planning meeting starts** - Review/Retrospective meeting is the the same
7	During a sprint planning meeting, how many sprints are discussed and planned for?	- 4 sprints - **Current sprint only** - All sprints in the project
8	Who attends the sprint planning meeting?	- **The scrum development team** - Outside stakeholders - Manager of the team - **Scrum Master** - **Product owner**
9	How often is the sprint retrospective meeting conducted?	- Every day - **Every sprint** - Every project

C. What technology is used for implementation?

The long-terms goals of ScrumTutor is to provide education through a fun and engaging environment, provide the thrill of playing a game, and provide most features of a good game. With these end goals in mind, HTML and Javascript was chosen for UI development, C#.NET was chosen as the server-side technology and Visual Studio as the Integrated Development Environment (IDE).

D. Scoring/tests

ScrumTutor has a scoring mechanism through intermediate quizzes to motivate the user to do better, replay and revisit concepts that were not well understood, and gain further understanding of the process by playing subsequent phases of the game. A user is allowed to progress to the next phase only after they have received a certain minimum score. The quizzes are provided throughout the system in two ways: formative assessment presented to the user during a sprint while the scrum activities are happening and summative assessment presented to the user at the end of a sprint. Successful completion of the summative assessment allows the user to move on to the next phase (and next sprint of the project). Table 1 shows sample formative and summative assessment questions.

V. Scrum Tutor – Implementation

ScrumTutor was developed using C#.NET and the Model-View-Controller (MVC) architecture. Visual Studio 2010 was used as the Integrated Development Environment (IDE), C# .NET as the middleware and SQL Server 2008 R2 as the database management system. User interface was designed using HTML, JavaScript, and JQuery [12]. Technologies used, database design and user interface design are further described in the following sections.

A. Technologies

C#.NET is an object oriented programming language developed by Microsoft; it is mostly implemented in the IDE called Visual Studio, a product of Microsoft. Visual Studio 2010 was used as an IDE to write the functional code in C# to access the database and the user interface.

SQL Server 2008 is a 'Database Management System' developed to allow definition, creation, querying, updating and administrating an organized collection of data. SQL Server Express 2008 R2 is the version used in this application to maintain the assessment data, user scores, and business logic data required for the simulation.

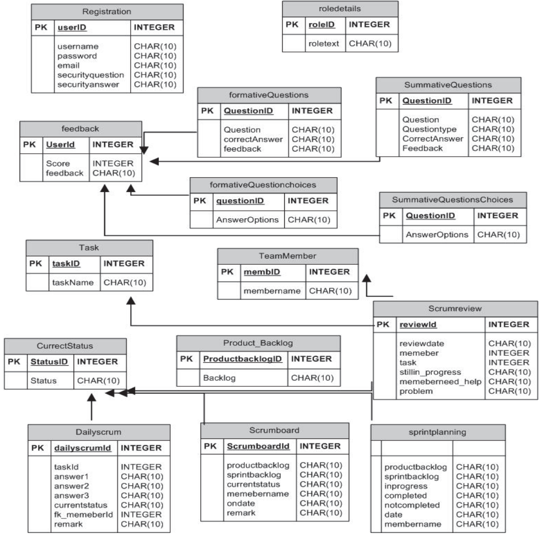
JavaScript is a client-side scripting language used to provide interactions with the user. Client-side scripts can be written to control, communicate and alter the content that needs to be displayed.

JQuery, an open source software, is 'a fast and concise JavaScript library that simplifies HTML document traversals, event handling, animations, and Ajax interactions for rapid Web application development [12].

B. Database Design

The database of ScrumTutor consists of several tables to save application data, user profile information, user scores, and statistics. The detailed database design is shown in Figure 3. Some of the important tables are listed and briefly described below:

- *Registration* – This table is used to store user login information. When a student or faculty registers for the first time their data is saved in this table.
- *Roledetails* – This table tracks roles of the scrum team.
- *FormativeQuestions* – This table is used to store all the formative assessment questions that are presented to the user at the end of every day in a sprint with the answer choices and the correct answer.
- *SummativeQuestions* – This table is used to store all the summative assessment questions that are presented to the user at the end of a sprint with the answer choices and correct answer.
- *Task* – This table has detailed description of all the tasks involved in a sprint.
- *TeamMember* – This table maintains the scrum team member information and description.
- *SprintStatus* – This table stores data about a specific sprint, status information and task assignments from all the meetings.



Registration

PK	userID	INTEGER
	username password email securityquestion securityanswer	CHAR(10) CHAR(10) CHAR(10) CHAR(10) CHAR(10)

roledetails

PK	roleID	INTEGER
	roletext	CHAR(10)

formativeQuestions

PK	QuestionID	INTEGER
	Question correctAnswer feedback	CHAR(10) CHAR(10) CHAR(10)

SummativeQuestions

PK	QuestionID	INTEGER
	Question Questiontype CorrectAnswer Feedback	CHAR(10) CHAR(10) CHAR(10) CHAR(10)

feedback

PK	UserId	INTEGER
	Score feedback	INTEGER CHAR(10)

formativeQuestionchoices

PK	questionID	INTEGER
	AnswerOptions	CHAR(10)

SummativeQuestionsChoices

PK	QuestionID	INTEGER
	AnswerOptions	CHAR(10)

Task

PK	taskID	INTEGER
	taskName	CHAR(10)

TeamMember

PK	membID	INTEGER
	membername	CHAR(10)

Scrumreview

PK	reviewId	INTEGER
	reviewdate memeber task stillin_progress memeberneed_help problem	CHAR(10) INTEGER INTEGER CHAR(10) CHAR(10) CHAR(10)

CurrectStatus

PK	StatusID	INTEGER
	Status	CHAR(10)

Product_Backlog

PK	ProductbacklogID	INTEGER
	Backlog	CHAR(10)

Dailyscrum

PK	dailyscrumId	INTEGER
	taskId answer1 answer2 answer3 currentstatus fk_memeberId remark	INTEGER CHAR(10) CHAR(10) CHAR(10) CHAR(10) INTEGER CHAR(10)

Scrumboard

PK	ScrumboardId	INTEGER
	productbacklog sprintbacklog currentstatus memebername ondate remark	CHAR(10) CHAR(10) CHAR(10) CHAR(10) CHAR(10) CHAR(10)

sprintplanning

	productbacklog sprintbacklog inprogress completed notcompleted date membername	CHAR(10) CHAR(10) CHAR(10) CHAR(10) CHAR(10) CHAR(10) CHAR(10)

Figure 3: Database Design

C. *User Interface Design*

Figure 4 presents screenshots of the login, registration, and home page of ScrumTutor tool. Figure 4 also shows screenshots that introduce Scrum framework and values of Scrum. It also shows screens that introduce the Scrum team, Scrum board and terminology at the beginning of the tutorial. Figure 5 shows screenshots of a sprint cycle going from day 1-5 in a week. Day 1 includes a discussion of the product, a sprint planning, followed by assignment of tasks on the Scrum board. Day 2 - 5 involve daily Scrum meetings that happen at same time and place in the morning followed by the team interacting and working on their tasks. In addition to the daily Scrum meeting, on day 4 a scrum review meeting is held. Day 5 ends with a sprint retrospective meeting.

VI. SCRUM TUTOR – EVALUATION

ScrumTutor has been tested on multiple browsers like Mozilla Firefox 19.0.2, Internet Explorer 10 and Google Chrome 26.0. The functionality of the application was tested thoroughly to ensure that an undergraduate student could go through the complete tutorial without any issues. An evaluation study was conducted at Arizona State University (ASU) to assess the usefulness of the tool. Students from the Masters in Computer Science Engineering program at ASU were surveyed prior to their usage of ScrumTutor to assess their understanding of Scrum. They were then allowed to use ScrumTutor and go through phases 1 and 2 of the tutorial. A post-survey of these same students was conducted after they used the tool and the students were asked the same questions to assess improvement in learning. Table 2 shows some of the questions on the pre- and post-survey. Unanimously, all of them felt that the tool was useful, easy to follow and gave them a better understanding of the Scrum process and the basic workflow unit in Scrum called sprint. Many of them felt that going through this tutorial provided them an industry perspective and context of how this Software Engineering concept in used in the industry on a project while working in teams. Some of them said that they better understood the team dynamics and various roles than earlier. Students also commented that they felt this was an innovative and engaging pedagogical practice for teaching software engineering processes.

Table 2: Pre- and Post-Survey Questions about ScrumTutor tool

#	Survey Question
1	What do you know about agile software development?
2	Describe Scrum framework?
3	Where do you think these concept can be used and applied?
4	Did ScrumTutor help in improving your knowledge of Scrum (Post)

VII. CONCLUSIONS AND FUTURE WORK

ScrumTutor provides hands-on experience on Scrum Framework to undergraduate students The interactive tutorial has multiple learning phases, i.e., Observation and Data Collection in this version and Development in the future versions. An evaluation survey of students about this tool provides preliminary results that using ScrumTutor enhanced their knowledge of Scrum, provided hands-on training of concepts learnt and provided a contextualized reinforcement of Agile process via the music software product that was managed and developed using ScrumTutor. Based on these results we are confident that a full version of ScrumTutor that is a game-based tool with multiple levels of learning through phases will be a useful addition to the software engineering classes. Future work on this project includes the implementation of phase III where the user plays the role of a 'team member'. The user gets to be part of the software team and contributes as a developer. User picks a task and implements it within the given timeframe of the sprint. Furthermore, sprints can be customized based on the project that is being implemented and managed. If the project is large, a sprint can be anywhere from two to four weeks. A Mobile app version of this game/tutorial for various platforms such as Android, iOS, and Windows will be a great addition and is

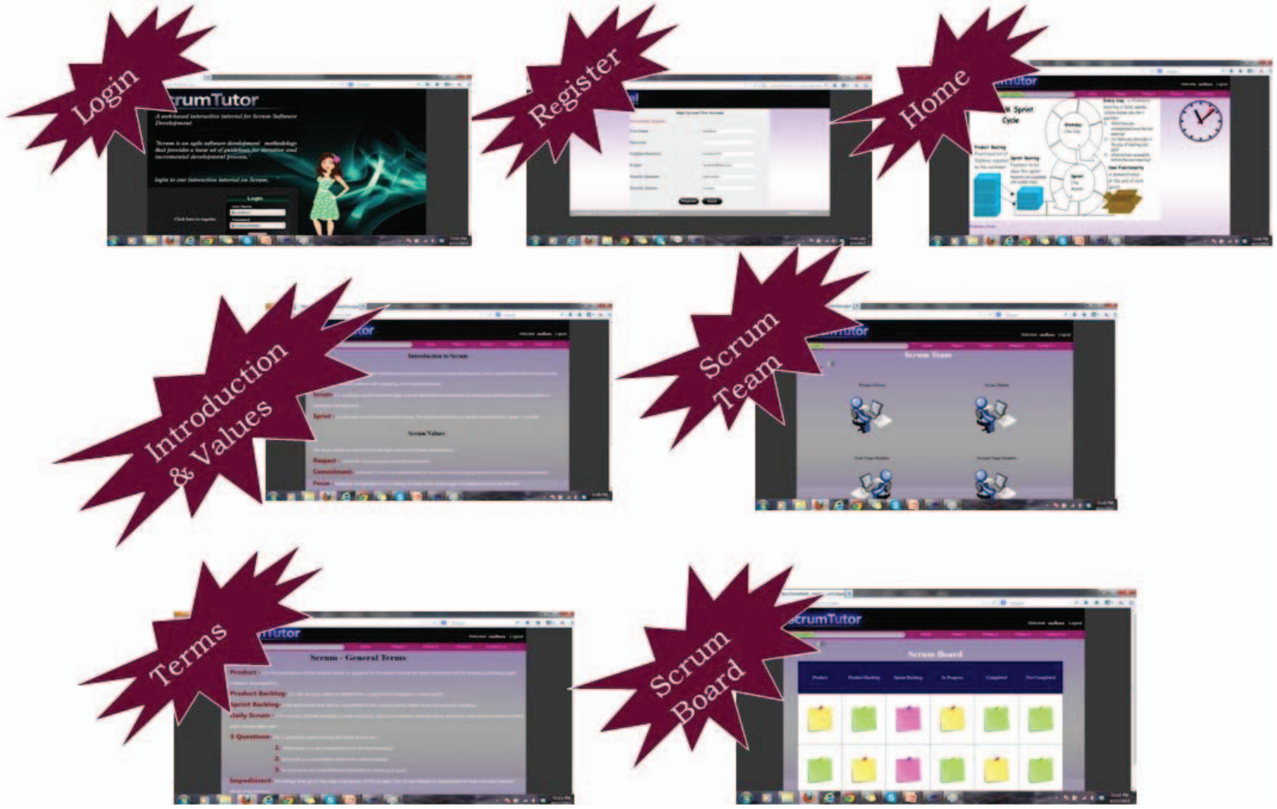


Figure 4: User Interface Screenshots showing terminology and Scrum Process

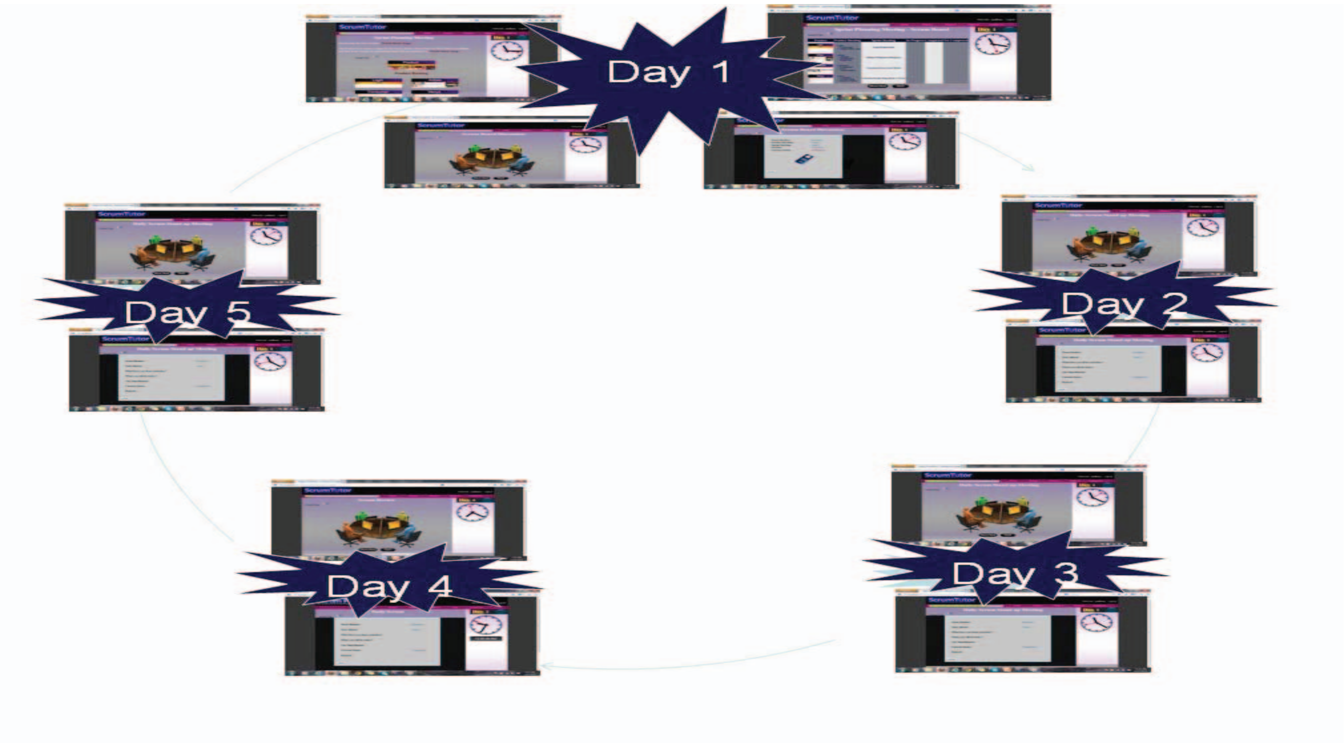


Figure 5: User Interface screenshots showing sprint cycle in Phase 1

likely to be adopted by a larger audience. A pilot study will be conducted in introductory Software Engineering classes at sophomore level at ASU. Data will be collected on the usability of the tool, student learning outcomes, student reflection of the tool and faculty reflection of the tool. Phase I and II of the tool can be further improvised with game-like features by adding more user interaction. HTML5 may be used to provide a rich user interface.

REFERENCES

[1] I. Sommerville, "Software Engineering" (9th 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 International 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", IEEE Conf. on Software Engineering Education and 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, Univ. of California, Irvine, 2006.

[7] A. Rusu, R. Russell, R. Cocco, "Simulating the Software Engineering Interview Process using a Decision-based Serious Computer Game", IEEE International Conference on Computer Games (CGAMES), 2011.

[8] A. Chua, "The Design and Implementation of a Simulation Game for Teaching Knowledge Management", Journal of the American Society for Information Science and 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", IEEE Workshop on Games & Software Engg. (GAS) at Intl. Conf. on Soft. Engg, June 2012.

[12] *"*jQuery*"*[Online]. Available: http://jquery.com/ [Accessed: June-2013]

[13] "Drupal"[Online]. Available:http://drupal.org/ [Accessed: June-2013]