

Software Engineering Process Models Strengths and Limitations with SimSE

Abdul Hafeez^{1*}, Syed Muhammad Hassan¹, Samreen Javeed¹, Basit Hassan Quershi¹, Abdul Aziz², Muhammad Furqan³ and Imtiaz Hussain¹

¹Department of Computer Science, SMI University, ahkhan@smiu.edu.pk, m.hassan@smiu.edu.pk, SamreenJaveed@smiu.edu.pk, basithq@smiu.edu.pk, imtiaz@smiu.edu.pk,

²Department of Computer Science, National University of Computer and Emerging Science, abdulaziz@nu.edu.pk

³Department of Information Technology, Higher Technology Collage Muscat, muhammadfurqan@hct.edu.om

Abstract

Background/Objectives: This work mainly focuses on how game-based learning approach can improve the understanding of the software engineering courses content. **Methods/Statistical Analysis:** This study introduced SimSE in our software engineering course. We used it for explaining three different process models to students and we also compare these models and tried to explain the come up with a point that which model is better in a specific situation. **Findings:** Our results found a significant variance in the selection of the three different process model ($F=6.1, 0.05$), whereas, the Agile model was found to be more efficient as compare to the rest of two. **Application/Improvements:** A modern learning approach such as game-based learning can make more interest of students in the theoretical conceptual computer science courses such as software engineering.

Keywords: Game-based Learning, Process Model, SimSE, Software Engineering

1. Introduction

Information technology is extensively impacting on our daily lives today, it impacts how we did our work, communicate and collaborate¹. The requirement of skilled IT professional is continuously increasing as required more effective higher education. In the 21 centuries, the computer science discipline necessitates to attract quality students and make them be capable and competent IT professionals². However, computer science education has been multifaceted and complex nowadays due to rapidly change and advancement in technology. In³⁻⁴ the past years, most computer science courses are taught in

traditional ways which may not be adequate to keep up with modern concerns. A modern learning approach such as game-based learning can make more interest of students in the theoretical conceptual computer science courses such as software engineering⁵. Game-based learning uses games which define learning outcomes⁶⁻⁷. They are designed in a way to balance course content with game play. Games are considered a powerful instructional rule for an objective such as winning⁶ they can provide a wide range of benefits such as increasing learning effectiveness, interest, motivation, and persistence⁷⁻¹⁰. Therefore, game-based learning provides a promising alternative to teach computing in higher education.

*Author for correspondence

In software Engineering, process models are used to developed quality software in a systematic manner and improved the chance of completing the software on time. They provide a sequence of activities for software development. They are divided into different logical stages that allow organizing the development work efficiently. According to the latest research, there are about 12 to 15 process models and most of the software engineering graduates and undergraduate students find it difficult to follow any one of them for their project. Due to the vast variety in models, it is quite difficult to understand that which model is best for which domain or which model is feasible to follow. In the industrial practices, the selection of the process model is depending on the type of software but most of the cases the choice of a software engineering team. If the team access the software accurately then they choose the right process model. Each process model has strengths and limitations and provides better performance in some situation than the others. It is a prime challenge for every software development team to choose an appropriate model that covers all aspect of the software development. It is a crucial point which decide the success or failure of the software. Therefore, deciding the process model according to the software is a very important skill which must be built in the software engineering students. On the other side, software engineering courses are taught in the universities in a theoretical way without any support of real-life examples that why mostly students complain about the dryness of the subject and proper skills regarding process model has not been developed in the students. It is very important that the students properly understand how the process model implement and they must be familiar with the advantages and limitation of each model. That is why in the previous semester we changed our strategy to teaching process models. We used a game SimSE to teach students about the process models and tried to explain the way of artifacts development in various process models. One of the most significant factor behind the use of SimSE in the classroom is the explanation of how various tasks of software engineering are accomplished in different process models some time it become very difficult to

explain in front of students theoretically because in all models' tasks are mostly common. Due to the vast variety of process models, it was quite difficult for us to select models for comparisons. After a lot of research, we have selected: Waterfall Model, Incremental Model, Agile Mode (Extreme Programming).

2. Background and Related Work

Computer games are promising method to teach software engineering courses. This method can accomplish high learning outcomes in difficult field of study. Two types of game-based methods are practiced for teaching software engineering: (1) game-based learning systems and (2) the game development process.

Programming languages and software process model are most significant areas of software engineering. Games of which teaching programming based areon visualization of program code e.g. turtle graphics¹¹⁻¹², Karel The Robot¹³⁻¹⁵, BlueJ¹⁶, DrJava¹⁷ and Alice¹⁸⁻¹⁹. They are used to creation of object world through frameworks such as Karel. Current process model teaching games are simulators such as Software Engineering Simulation by Animate Models (SESAM)²⁰, Role Playing Game for Software Engineers (RPGSE), Second Life²¹, Value-Based Software Engineering (SimVBSE)²², web simulator Simjava SP²³ and others. Many academicians have been used gaming simulator in their lectures such as Alex Baker and educational card game Problems and Programmers which simulates the software engineering process and teach process issues which are not sufficiently clarified in the lectures. They also performed an experiment in the lectures and note the improvement in the understanding of the software process²⁴.

In²⁵ authors presented an inwards-class game for teaching software engineering. The proposed game is a simulation of a real software engineering project. He also evaluated the game results through a survey. In the survey, they focused on issues related to the 4 P's model. In²⁶ authors experimented with 3D virtual world Second Life in their classes, (1) Ohio University, (2) University of Mary Washington. Additionally, they built an educational game

in Second Life, and in SimSE. In^{27,28} authors addressed the issues which faced by teacher in the software engineering teaching. They developed two game-based simulation tools for teaching software engineering: (1) Problems and Programmers and (2) SimSE.

The SIMSE software engineering educational game that simulates software engineering environment according to the chosen process model. It has 6 different process models, each process model has a different scenario and different method. SimSE is an interactive, fully graphical educational game, it is developed specifically to improve the skill of students in circumstances which are required for handling and understanding of software process issues. In the gameplay, at the starting of the game, a detailed description of software engineering task which will be expected to be performed is presented to the player. Usually, the description contains about goal of the game, the budget, available time to finish the project,

some informative hints to guide about the game and the detail of how the score will be calculated. It provides a fully graphical user interface, as shown in Figure 1. The central part of GUI is a virtual office where the software engineering activities are performed, containing typical office environment and team members. The team members communicate with the project manager (player) via over the head speech bubbles. These speech bubbles are used to inform the player about important information for example when the job was initiated or when it was finished, the occurrence of random events, and response to one of the player's actions.

2.1 Process Models

The waterfall model was the first process model. In it, each stage must be fully complete before the next stage and there is not allowed to parallel execution of development stages. In the incremental model product is designed,

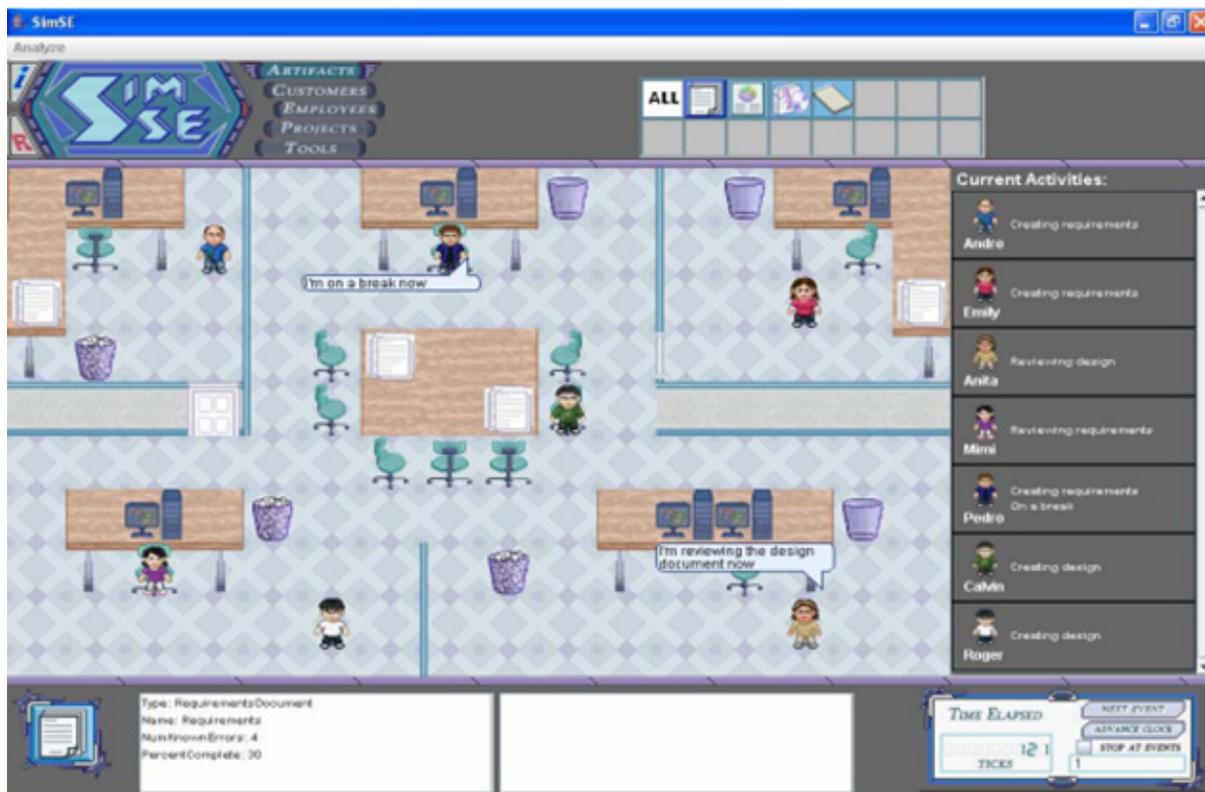


Figure 1. Graphical user interface of SIMSE.

implemented incrementally until the product is finalized. It combines the elements of the waterfall model with the iterative development. Agile methodology improves the quality of software and its provides ability to adapt the changes in requirement during the development. It provides iterative and frequent small releases throughout the development.

3. Methodology

The course in which we used SimSE was CSC202, a one-semester introductory software engineering course at SMI University. We used SimSE in two classes. In the 4th weeks of the semester after completely discussed different process models, we gave a short fifteen minutes tutorial about how to play SimSE, and gave assignment to the 68 students of two classes. They played three SimSE models waterfall, incremental, and agile at least five times or until they got above 90% points and note the final score of each play. In the end, we also conducted the student's interview and asked questions about each process model and finally, we compare the process model from students' point of view.

4. Result and discussion

97% of the students attempted the SimSE extra credit assignment, so the interest was significant. The students' scores in the first attempted of waterfall model was very low due to their complex nature. Table 1 shows the score of the first attempt of water fall model the result shows that no student got above 50% and 70% did not complete the project on time in the first play and in the last attempts they could not cross 70% and 60% finish the project on time.

In the interview we found that student faced too many problems in the waterfall due to the sequential execution of the phases such as: single activity, requirement gathering, and design required too much of time, difficulty inteam management, and very poor resource management. In the end, students realized that waterfall model is not suitable for long and ongoing project and a very high level of risk and uncertainty and where requirements are at risk of changing. It is used where requirements are clear and there will be no changes in the development of time such as defense projects, migration projects: where

Table 1. The first attempt of water fall model

Score	No of student 1 st play	No of students 2 nd play	No of students 3 rd play	No of student 4 th play	No of student 5 th play
10-20	28	12	5	0	0
21-30	21	22	15	8	1
31-40	16	23	18	12	5
41-50	3	7	17	23	25
51-60	0	4	11	18	24
61-70	0	0	2	7	11
71-80	0	0	0	0	2
81-90	0	0	0	0	0
91-100	0	0	0	0	0

requirements remain the same and only platform or languages may vary/ change (Table 2).

Students' scores of the first attempt of the incremental model also low in the incremental model. As shown in Table 2, the score of the first attempt of incremental model no student got above 60% and 50% did not complete the project on time in the first play and in the last attempts, they could not cross 90% and 75% finish the project on time. Because, in the incremental model, a module or small group of modules are constructed in a single increment and in the one increments the rules of waterfall are followed. In this experiment students already play the SimSE in waterfall model therefore, their results are better than the waterfall. In the interview we found that student also faced some problems with the incremental model such as a lot of advanced planning, clear and complete specification of the system required to properly divide the project into the increment, each phase of an increment is fixed and cannot be overlap. The incremental model can be used in the situation where requirements are clear and can implement phase wise. It is suitable for web site and product line software.

Finally, students played the Agile module in SimSE and got above 90% score in the first attempt as shown in Table 3. In interview we found that students faced very few problems in agile model such as difficult to measure progress because progress depends across various cycles, it requires more energy and time due to software engineers and customer constantly communicate with each other and its short cycle not leave enough time for the design review therefore, designers face so many redevelop, the requirements and procedure which are developed in each iteration should be independent and not have dependencies with other requirements outside of the iteration. It is used in small and medium-sized projects, where multiple variantsis required and when major deliverable can be broken down and produced in the incremental discrete package.

We also applied F-test between and within the three different software models to check the significance difference among the three (waterfall model, Incremental Model, Extreme Programming Model). A significant variance found among the three software models ($F=6.1$,

Table 2. Game play and their score

Score	No of student 1 st play	No of students 2 nd play	No of students 3 rd play	No of student 4 th play	No of student 5 th play
10-20	22	8	1	0	0
21-30	18	12	9	9	0
31-40	20	17	13	9	0
41-50	5	18	22	18	12
51-60	3	10	15	19	19
61-70	0	3	7	10	25
71-80	0	0	1	3	10
81-90	0	0	0	0	2
91-100	0	0	0	0	0

Table 3. Results and score of the Agile model

Score	No of student 1 st play	No of students 2 nd play	No of students 3 rd play	No of student 4 th play	No of student 5 th play
10-20	7	2	0	0	0
21-30	5	6	2	1	0
31-40	18	13	6	3	0
41-50	26	22	12	4	2
51-60	11	13	25	7	5
61-70	1	9	17	26	20
71-80	0	3	6	22	35
81-90	0	0	0	5	2
91-100	0	0	0	0	1

Table 4. F-distribution between and within three different software models

Source	Bracket Term	S.S	d.f	M.S	F	p-value
Between Groups	$[A] = \frac{\sum A_i^2}{n} =$ 68953.8	67936.8	2	33968.4	6.1	<0.05
Within Groups	$[Y] = \sum y_{i,j}^2 =$ 59909	9044.8	12	5661.4		
Total	[T]= 68952.6	9043.6	14	4852,6		

0.05) as shown in Table 4, which is highly suggestive and in favor of the Agile.

5. Conclusion

In this paper we have compared three major software development life cycle models with the help of SimSE Simulator. Selection of appropriate development life cycle model is very important, software must be delivered to

the client within timeframe and must have desired quality. In this regard, we applied F-test between and within the three different software models to check the significance difference among the three groups. A significant variance found among the three software models. This study makes software development model selection process easy which improves the quality of software and decreases the software failure rate.

6. References

1. 10 hot IT skills for 2013 [internet]. <https://www.computerworld.com/article/2492784/10-hot-it-skills-for-2013.html> . Date accessed: 2013.
2. The Joint Task Force for Computing Curricula [internet]. <https://dl.acm.org/citation.cfm?id=2534860>. Date accessed: 2013.
3. Preparing computer science graduates for the 21st Century [internet]. <https://ojs.lib.uwo.ca/index.php/tips/article/view/3552>. Date accessed: 2011.
4. Game2 Learn: improving the motivation of CS1 students [internet]. https://www.researchgate.net/publication/228799919_Game2Learn_improving_the_motivation_of_CS1_students. Date accessed: 02/2008.
5. Engagement : Gaming throughout the Curriculum [internet]. https://www.researchgate.net/publication/221536695_Engagement_Gaming_throughout_the_curriculum. Date accessed: 04/2009.
6. Serious games [internet]. https://en.wikipedia.org/wiki/Serious_game. Date accessed: 08/08/2019.
7. Digital game-based learning [internet]. <http://www.marcprensky.com/writing/Prensky%20-%20Ch2-Digital%20Game-Based%20Learning.pdf>. Date accessed: 2001.
8. Kirriemuir CJ, McFarlane A. Literature Review in Games and Learning Literature Review in Games and Learning. Context. 2004; 3(2): 208–213.
9. The use of computer and video games for learning [internet]. https://dera.ioe.ac.uk/5270/7/041529_Redacted.pdf. Date accessed: 2004.
10. de Freitas S. Using games and simulations for supporting learning, Learning, Media and Technology. Digital Games and Learning. 2006; 31(4): 343–358. <https://doi.org/10.1080/17439880601021967>.
11. Web Turtle [internet]. <http://www.sonic.net/~nbs/webturtle/examples/>. Date accessed: 2013.
12. Here, there and everywhere - on recurring use of turtle graphics in cs1 [internet]. <https://dl.acm.org/citation.cfm?doid=359369.359375>. Date accessed: 2000.
13. Karel the robot: a gentle introduction to the art of programming [internet]. <https://www.amazon.in/Karel-Robot-Gentle-Introduction-Programming/dp/0471089281>. Date accessed: 08/04/1981.
14. Robot a gentle introduction to the art of object-oriented programming in java [internet]. <https://www.amazon.in/Karel-Robot-Introduction-Object-Oriented-Programming/dp/0970579519>. Date accessed: 05/04/2013.
15. Sanders D, Jerroo BD. A tool for introducing object-oriented programming. SIGCSE Technicak Symposium on Computer Science education. 2003; 201–204. <https://doi.org/10.1145/792548.611968>.
16. Kölling M. Teaching Object Orientation with the Blue Environment. Journal of Object-Oriented Programming. 2004; 12(2): 14–23.
17. Game Programming in Introductory Courses With Direct State Manipulation [internet]. <https://dl.acm.org/citation.cfm?id=1067465>. Date accessed: 2005.
18. Teaching object-first in introductory computer science [internet]. <https://dl.acm.org/citation.cfm?id=611966>. Date accessed: 2003.
19. Alice 3 Software [internet]. <https://www.cmu.edu/homepage/computing/2009/winter/alice-3-software.shtml>. Date accessed: 2014.
20. Drappa, Ludewig J. Simulation in Software Engineering Education, Proceedings of the 22nd International Conference on Software Engineering. 2000; 199–208. <https://doi.org/10.1145/337180.337203>.
21. Zhu Q, Wang T, Shenglong T. Adapting Game Technology to Support Software Engineering Process Teaching: From SimSE to Mo-SE Process. Proceedings of Third International Conference on Natural Computation (ICNC). 2007; 777–780. <https://doi.org/10.1109/ICNC.2007.159>.
22. Overmars M. Teaching Computer Science through Game Design. IEEE Computer. 2004; 37(4):. 81–83. <https://doi.org/10.1109/MC.2004.1297314>.
23. Teaching Software Engineering Through Game Design [internet]. <https://dl.acm.org/citation.cfm?id=1067482>. Date accessed: 2005.
24. Baker A, Navarro EO, van der Hoek A. An experimental card game for teaching software engineering. Proceedings 16th Conference on Software Engineering Education and Training. 2003; 216–223.
25. Teaching software development by means of a classroom game: The software development game [internet]. file:///C:/Users/a/Downloads/354-450-1-PB.pdf. Date accessed: 2014.
26. Ye E, Liu C, Polack-Wahl JA. Enhancing software engineering education using teaching aids in 3-D online virtual worlds. 2007 37th Annual Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports, Milwaukee, WI. 2007; 8–13.
27. Teaching software engineering using simulation games [internet]. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.124.2661&rep=rep1&type=pdf>. Date accessed: 2004.
28. Problems and Programmers: an educational software engineering card game [internet]. <https://www.ics.uci.edu/~emilyo/papers/ICSE2003.pdf>. Date accessed: 2003.