Scrum Based Quality Enhancement Model for Supervising Final Year Projects of Computer Science

Hafiza Sadia Hassan, Nayyar Iqbal, Ahsan Raza Sattar, Usman Rafi

Abstract— The challenges of today's computer science courses are not the specific technical competencies, but rather organizational issues which are a hindrance in effective teaching. By introducing new teaching methodologies we can cope with them and make teaching efficient and effective and can achieve worldwide standards. The purpose of this research is to introduce the integration of agile methods in a capstone course for Bachelor of Science degree. With the integration of Scrum model into computer science teaching techniques, an educational beneficent alternate of traditional teaching methodologies is identified. Rapid prototyping and incremental development provide many opportunities for students to improve and reflect themselves. The focus of Scrum on self-organizing teams offers a podium to practice project organization, by empowering students to take the obligation for the product development process. This can enhance students' preparation for a future career as professionals. By using it, we can reduce the gap between educational and industrial hubs and can compete with global industry. We observed some encouraging points, i.e. such as the practical nature of learning by example. The main conclusion of this research is that, a modified version of Scrum methodology is necessary to work in our academic scenario. This paper discusses a case study regarding the application of scrum model in supervising the capstone projects of students of BS(IT) classes in GPCSF.

Index Terms— Agile methodologies for education, Scrum in education, Education quality enhancement, FYP Scrum, Educational Scrum, Educational sprint retrospective, Role of scrum master in education.

1 Introduction

Computer science is a vital part of our life; computer science is currently a key empowering influence for disclosure and development in the most different fields of endeavour, making it an unbelievably applicable course of study. Computer researchers design the future by developing architectures and systems for more propelled figuring, and by building up the applications that work inside of those frameworks.

Quality education in computer science is essential in our country's Education system. Teachers and Instructors, particularly, college and university lecturers, are the primary power of quality education [1]. For enhancing quality education, college and university educators need to adjust their ideas and their strategies, set up new philosophy, assume their own part and satisfy their own particular task. Through effective and efficient teaching reforms, we ought to concentrate on instructing and enhance teaching quality and impact. Teaching is a technique while educating is an object.

At present, a critical vital measure of our national teaching is to change the exam-oriented education into quality-driven education. Classroom teaching is a fundamental channel of valuable quality education. Educators must change the thought of educating, change instructing systems to enhance students' learning and intuition. Moreover, Students need to study hard, sincere, considering, genuine practice so as to improve their capacity to learn and build up their capability to think, to upgrade their functional capacity and their capacity of development.

Quality education is a fundamental framework in our nation's preparation. Teachers are the standard force of quality education. Henceforth, teachers need to conform their demonstrating thought and should develop new demonstrating conviction framework, accept their own part and fulfil their own errand. Through competent demonstrating change, we should focus on training and upgrade the quality and effect of computer Science education. So there is an urgent need to find new teaching methodologies to achieve this quality education. This research presents a new teaching methodology to achieve above mentioned objectives.

Better approaches for overseeing improvement inside the software development industry arose, first time in the 21st century and bring change in the method for the software development [2]. Makers of different new strategies in programming advancement met up and framed the Agile Manifesto in 2001[3], although the core of agile methodologies can be found in 1957[4]. Agile Software Development is an umbrella term which which is used for many iterative as well as incremental methodologies all having cros-functional andself-arranging groups, quickly reacting to change [5]. These strategies have gotten great appraisals from the software industry and numerous huge organizations use these methodologies in their improvement nowadays. Scrum is the most utilized

- *Hafiza Sadia Hassan, Research Scholar, Department of Computer Science, University of Agriculture, Faisalabad, Pakistan. E-mail: sadiahassan60@gmail.com*
- *Nayyar Iqbal, Lecturer, Department of Computer Science, University of Agriculture, Faisalabad, Pakistan. E-mail: nayyar_iqbal@hotmail.com*
- *Ahsan Raza Sattar, Assistant Professor, Department of Computer Science, Univeristy of Agriculture, Faisalabad, Pakistan. E-mail: ahsan_raza@uaf.edu.pk*
- *Usman Rafi, Research Scholar, Department of Computer Science, University of Agriculture, Faisalabad, Pakistan. E-mail: usmanrafimscs@gmail.com*

methodology amongst the agile methods, and is right now utilized by organizations, for example, Microsoft, Google Yahoo! and that's just the beginning.

Scrum is one of the lightweight methods which was recommended for the management of product development projects but now a days it is mainly used for product development. Some people consider Scrum, a method just relevant to software development. But, when taking a gander at Scrum methodology itself, its foundations don’t have immediate connection to software development, there are many example of that concept like Scrum method was applied in distributed software development [3] as well as it is used for Teaching globally distributed project courses successfully [6].

Consequently, it is conceivable to conclude that Scrum can be utilized as a part of any sort of item improvement and can be adopted as a teaching methodology. Yet, there is very little research literature presented on the use of agile methodologes in non-software situations as well as case studies on the use of Scrum in these areas especially in teaching methodologies are scarce. Here, we want to explore if teachers of computer science adopt scrum as a teaching methodology, they can attain the same benefits of using Scrum as a teaching methodology as in software development.

Scrum teaching methodology is an incremental and iterative framework, in contrast to the sequential traditional framework. The Scrum teaching methodology makes teaching easier, effective and efficient. The method consists of several events, roles and artefacts which are illustrated in Figure.

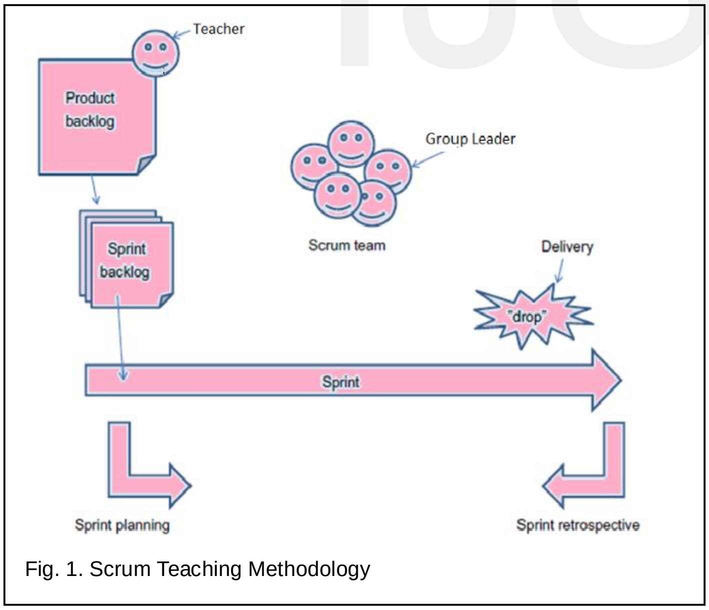


Fig. 1. Scrum Teaching Methodology

Scrum Roles: There are three roles in Scrum, the teacher, the group leader and the Team. Together these three roles frame the Scrum Team. The teacher, like the product owner is accountable of the education system. That person prioritizes project contents according to the requirements. Group leader is a student within the class with some experiences in the subject and he or she supports the teacher and the Team in fulfilling with the Scrum roles. The group leader protects the team from external disruption, in order for the team to be focused on their work. The Team is a group of approximately seven or less students that executes the prioritized tasks and complete the task assigned by the teacher within a specific time.

Product Backlog: Product backlog is a prioritized list of the project contents with estimated time that are supposed to be completed under this project. The estimated time are in days. The items presented in given list can be functional as well as non-functional requirements. The sprint backlog can be defined as the list of tasks that the team of the students is going to perform during the educational sprint.

Scrum Events: Educational Sprint is a fixed period of time in which the student team performs a specific task related to their course. Sprints continues to occur after one another without having a pause and are time bound. Sprint planning meeting is conducted at the start of every sprint. In this meeting there are two main question which are to be answered. Firstly, what is/are the specific tasks for this sprint? Secondly, how will these tasks get done? In sprint educational retrospective meeting, the team discussed the passed sprint and concludes that what could be improved to make upcoming sprints more productive. At the end of the sprints, students’ team presents their work in the deliverable form, which can a module of a software, assignment. The teaher evaluates the results of the sprints at the end of every sprint and at the end of the semester teams submits theire projects as a whole.

The purpose of this research is to see if Scrum methodology can be used as a teaching methodology for Computer science education, with or without alteration. The investigation was conducted upon students of Department of Computer Science, Govt. Postgraduate College, Samanabad, Faisalabad (GPCSF), Pakistan. For teaching, Scrum is chosen as a methodology to teach BS(Information Technology) 8th semester [7]. This research paper is a study of the outcomes of this experiment that continued from March to July 2015.

The delimitations of the study are that only one case is examined in detail. The outcomes may subsequently not be generally applicable to other groups. The time allotment of the experimental study was 5 months, bringing about the awesome measure of information and thoughts emerging, however, not all viewpoints and points of view could be obliged inside the scope of this study. It may likewise be conceivable that the author's nearby collaboration with the groups may have an effect on the outcomes; it could potentially influence the conclusions drawn in the study.

The remainder of this paper is ordered as follows: Section II presents the background and scenario of the research. Section III throws light on the sprint planning for the course. In Section IV, critical analysis is presented. Finally, in section V, some concluding remarks and future work of the study is presented.

2 BACKGROUND

The motivation of this research is the graduate course, in computer science and information technology, at Govt. Postgraduate College, Samanabad, Faisalabad (GPCSF). In this discipline, students learn different types of software development concepts and techniques, like programming, database design, Artificial intelligence, Human Computer interaction, project management and web development. These courses have practical component also, in the form of capstone project in which they can put most of the theoretical part of their study into practice. They have to develop their projects in the form of groups. But it is observed that most of the projects remained incomplete even at the end of course, or they had fundamental flaws and their correction in time is impossible and these projects could not achieve the standard of implementation. Moreover, due to lack of retrospective sessions and meetings, the students also couldn`t pay attention to the practical implementation of learned skills.

The main cause of the given above problem is that there is a gap between theoretical work and practical one. There is another thing to be observed, it's a fact that agile methodologies were a part of the graduate program, but these methodologies were taught totally in an abstract way, which results in boredom, or some students perceive it in a wrong way. To resolve these two problems, principles and concepts of agile methodologies were used to reorganize the course [8]. The scrum was implemented as an iterative approach of teaching organization, despite the fact it is not specially made as a teaching methodology.

According to the introduced methodology, the capstone project of the students were to be managed and developed iteratively. At sprints or short period of time students have to deliver their project in parts or modules. At the same time, the theoretical concepts which were necessary for the success of sprints were reorganized.

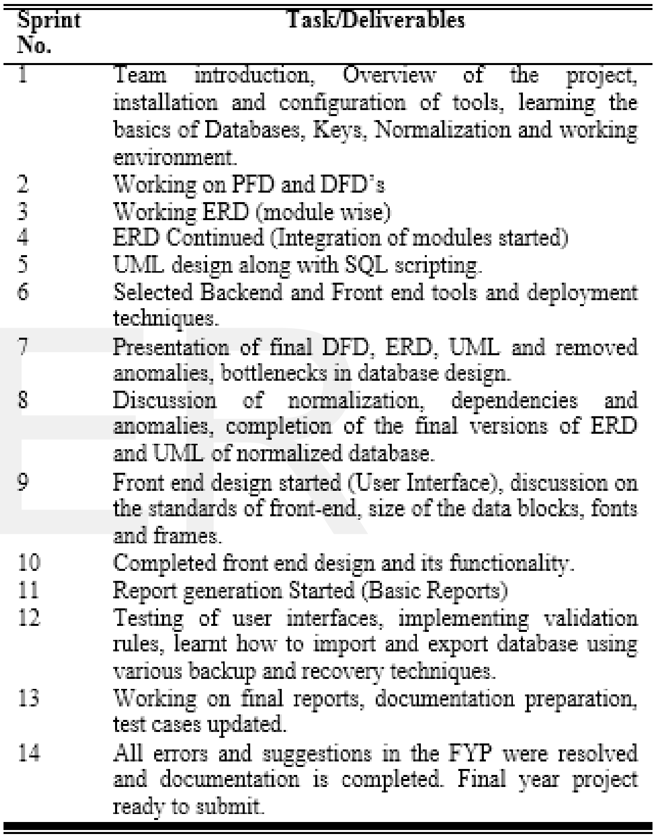
The main purpose behind this research was to make students able to develop software earlier, and reduce the possibility of delivering incomplete projects at the end of the degree. Implementation of the chosen method would also make them practice the implementation of the scrum in the development of software. It would reduce the gap which exists between theoretical computer science teaching and practical scrum concepts.

The coming section illustrates the course setup and presents the course project, according to agile methodology.

4 Course Project

The whole capstone project and theoretical course related to the project were not applied in a sequential or tradition way. Since iterative process based on Scrum model, the actual division of the tasks or activities was different. The course outline was divided into fourteen sprints of one week each. Students were directed to deliver a part of their project, at the end of each sprint. At the end of the semester, students were supposed to deliver the project as a whole. Daily meetings with the teacher or supervisor were arranged. The student of the group and their supervisor remain connected mostly on Facebook. The students discuss their obstacles with each other and their teacher on FB when they were in remote places, which resulted in increased collaboration between student teams and product owner. They discussed their impediments with each other and their teacher and completed their task within time. In the bachelor degree, students must develop a medium size project as a part of their course. The project should be developed iteratively, for reducing the risk of incomplete project, at the end of their degree. In this case study students were allowed to deliver partial products at the end of some sprints. In some sprints, students could make a plan to deliver some adjustments or enhancements, as a substitute of a product. The given table is the list of fourteen sprints and their tasks assigned and deliverables during a sprint.

TABLE 1
Sprint Based Course Schedule



Sprint No.	Task/Deliverables
1	Team introduction, Overview of the project, installation and configuration of tools, learning the basics of Databases, Keys, Normalization and working environment.
2	Working on PFD and DFD's
3	Working ERD (module wise)
4	ERD Continued (Integration of modules started)
5	UML design along with SQL scripting.
6	Selected Backend and Front end tools and deployment techniques.
7	Presentation of final DFD, ERD, UML and removed anomalies, bottlenecks in database design.
8	Discussion of normalization, dependencies and anomalies, completion of the final versions of ERD and UML of normalized database.
9	Front end design started (User Interface), discussion on the standards of front-end, size of the data blocks, fonts and frames.
10	Completed front end design and its functionality.
11	Report generation Started (Basic Reports)
12	Testing of user interfaces, implementing validation rules, learnt how to import and export database using various backup and recovery techniques.
13	Working on final reports, documentation preparation, test cases updated.
14	All errors and suggestions in the FYP were resolved and documentation is completed. Final year project ready to submit.

Each team was assigned a product owner. The teacher as playing the role of product owner, was responsible for the results [9]. He was supposed to help the team during sprint meetings. As a teaching methodology, Scrum methodology was followed, students had to practice the Scrum Master's role. That's why, in every sprint, a different team member had to play the role of Scrum Master.

Control during the sprint would be performed with the help of a Facebook group. In this group, students and teachers should upload and maintain all the essential information to allow an appropriate management of the progress in each sprint.

Mostly, when a sprint is planned, a timeline of the sprint would be made. All the activities according to its specific time would be performed. All the activities, which would be performed on each day would be uploaded on the group. The scrum master would be responsible for the maintenance of the FB group.

5 Critical Exploration

Overall, the teaching methodology was successful, as both of its goals were achieved:
1. The group was able to deliver the complete project at the end of the semester
2. Students succeeded to implement some of the agile principles in practice, implementing them as they were learning these methodologies. This resulted in a better teaching of agile concepts [10].

There were some other points to be noted. In the presented case study, the teachers acting as PO had earlier experience with Scrum in practical developments, which provided a good alignment between the course contents and the rules of agile development [11]. The presence of the experienced teacher as PO, and his permanent availability, was also a cause of producing positive results, as it provides a facility for the students to clear their doubts and removing their impediments without taking too much time. The role of Scrum master was also a good experience for the student it would be beneficial for them in practical life.

However, we observed many problems with the approach, and we believe these could be addressed to provide even better results in the future.

A first problem relates to the gradual evolution of the learning process in a normal course, and its contrast to the incremental and iterative approach proposed by most agile methods. There is a natural incompatibility between these two approaches, because each sprint represents a small, but complete, development iteration. This means that, in each sprint, the development team must perform activities that relate to almost all software engineering disciplines, even if only a little part of them at a time. Although we attempted to introduce these disciplines at the same time, much content was only really assimilated by students after two or three sprints, after the discipline was finished.

But, there were some problems which were experienced in this case study while using the purposed iterative methodology. If they are resolved they provide better results.

Firstly, there is a natural incompatibility between gradual progresses of the learning process in a regular course and the incremental approach offered by most agile methodologies. Because every sprint in practical environment provides a small but a complete development iteration and they have to deliver a part of the actual product. But it was not possible in each and every sprint of the teaching environment. The students could not provide a complete part of deliverable product as at they had not enough knowledge for the implementation of a whole part of the software. Resultantly, in some sprints they have to deliver an incomplete piece of product and in some other sprints they are bound to enhance these incomplete pieces of product.
Secondly, the problem was related to self-management of the scrum teams. The lack of responsibility and discipline in the Scrum team made it sometimes impossible for the Scrum master to manage task execution in a proper fashion.

6 Conclusion And Future Work

The purpose of this research was to test hypothesis that whether it is possible to use scrum as a teaching methodology in the supervision of final year projects or not. A case study was conducted at GPCSF in Faisalabad, where teachers used Scrum as a teaching methodology for students' capstone project. It was set out to be an experiment to conclude that whether or not to carry on using the framework. The experiment lasted for 5 months. The deployment of the Scrum method for that particular case study can be seen as a success, in spite of not satisfying some criteria and breaking some of its standards. The principal conclusion of the study is that Scrum can work as a teaching methodology. It is impractical to say regardless of whether it is superior to different teaching, on the other hand it has proven accommodating and indicated advance for the situation studies exhibited.

Although, the integration of scrum and teaching process has proven effective and beneficial for the students, many issues and problems has been observed during the process. It influenced the overall practice, from normal theoretical classes to a better practical experience revolving around an actual development environment, which simulates the real challenges of this practical environment. Students of the case study were not only forced to understand the principles and the concepts of agile methodologies but it also put them in practice and observe the difficulties and challenges faced by the practitioners. Particularly, the students had to deal with process enhancement challenges, technical challenges, inter team relationship and personal interaction.

Keeping in mind the problems and difficulties which were observed, we conclude that some problems cannot be easily resolved, such as related to dedication and lack of time. Future work should deal with better distribution of the disciplines together with the sprints, to deliver a closer alignment with the deliverables.

Better selection of projects, to only comprise those that are agreeable to a more logical division from simpler functionality to a more multifaceted one, and Self-management formalization, to assistance Scrum Masters to deal with their teams.

As a future work, this model can be applied to general educational subjects in computer science. Workshop sessions can also be employed using the same method. Performance evaluation of traditional teaching methods in comparison to this innovative method can also be done.

References

[1] J. Shengji, Y. Hongmei, and S. Yimin, "On the importance of teaching methods and teacher's devotion in education," *In Proc. of Int. Forum Inf. Technol. Appl. IFITA 2009*, vol. 1, pp. 764–765, 2009.

[2] T. Dybå, and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Inf. Softw. Technol.*, vol. 50, no. 9–10, pp. 833–859, 2008.

[3] H. Sverrisdottir, H. Ingason, and H. Jonasson, "The Role of the Prod-

uct Owner in Scrum-comparison between Theory and Practices," *Procedia-Social Behav. Sci*., vol. 119, pp. 257–267, 2014.

[4] M. Edin Grimheden, "Can agile methods enhance mechatronics design education?," *Mechatronics*, vol. 23, no. 8, pp. 967–973, 2013.

[5] N. B. Moe, T. Dingsøyr, and T. Dybå, "A teamwork model for understanding an agile team: A case study of a Scrum project," *Inf. Softw. Technol*., vol. 52, no. 5, pp. 480–491, 2010.

[6] D. Damian, C. Lassenius, M. Paasivaara, A. Borici, and A. Schröter, "Teaching a globally distributed project course using Scrum practices," *2012 2nd Int. Work. Collab. Teach. Glob. Distrib. Softw. Dev. CTGDSD 2012 - Proc.*, pp. 30–34, 2012.

[7] D. Strode and J. Clark, "Methodology in software development capstone projects," *20th Annu. Conf. NACCQ, Nelson, NZ*, pp. 243–251, 2007.

[8] B. Isong, "A Methodology for Teaching Computer Programming: first year students' perspective," *Int. J. Mod. Educ. Comput. Sci*., vol. 6, no. 9, pp. 15–21, 2014.

[9] H. Sverrisdottir, H. Ingason, and H. Jonasson, "The Role of the Product Owner in Scrum-comparison between Theory and Practices," *Procedia-Social Behav. Sci*., vol. 119, pp. 257–267, 2014.

[10] M. Cubric, "An agile method for teaching agile in business schools," *Int. J. Manag. Educ.*, vol. 11, no. 3, pp. 119–131, 2013.

[11] H. Sverrisdottir, H. Ingason, and H. Jonasson, "The Role of the Product Owner in Scrum-comparison between Theory and Practices," *Procedia-Social Behav. Sci*., vol. 119, pp. 257–267, 2014.