ISSN: 0975-833X

Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 10, Issue, 11, pp.75490-75494, November, 2018
DOI: https://doi.org/10.24941/ijcr.33323.11.2018

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

RESEARCH ARTICLE

USAGE OF SCRUM IN A PROJECT MANAGEMENT CLASS IN A COMPUTER SCIENCE UNDERGRADUATE COURSE

***Rodrigo Alves Costa**

Universidade Estadual da Paraíba - Brazil

ARTICLE INFO

Article History:
Received 24th August, 2018
Received in revised form
27th September, 2018
Accepted 29th October, 2018
Published online 30th November, 2018

Key Words:

Education. Computer Science.
Agile. Scrum.
Project Management.

ABSTRACT

Software development agile approaches are mainly characterized for prioritizing the development of functionalities through executable code rather than the production of extensive written documentation and also for presenting quick responses to changes. In this context, this work aims to report the application of Scrum in an undergraduate classroom environment in the Computer Science course of the State University of Paraiba, during the development of a project for the Project Management subject, by presenting the entire project execution process. In the end, we achieved an experience that simulated a real world industry-like situation, common to that of software development companies, which facilitated the learning of the content of the subject.

***Citation: Rodrigo Alves Costa*. 2018.** "Usage of scrum in a project management class of a computer science undergraduate course", *International Journal of Current Research*, 10, (11), 75490-75494.

INTRODUCTION

Agile software development approaches diverse from traditional development processes mainly due to the fact that they prioritize the development of functionalities through executable code rather than the production of extensive written documentation, as well as to provide quick answers to changes and collaboration with the client instead of following rigid plans and contractual negotiations [Costa 2016]. According to a study carried out by Vargas (2017), after collecting information from a large number of professionals using different methods, both agile and traditional, it was noticed that agile approaches actually improve project results in terms of quality, customer satisfaction and productivity, without significantly increasing its cost. One of the most recognized approaches to agile management of software projects is Scrum [Schwaber 2004]. According to Sousa et al. (2015), management with Scrum involves the use of a framework of procedures and artifacts which lead to the objectives of self-organization of development teams. Such a framework can be used in the classroom environment in the teaching and learning processes, since Brennan (2011) suggest that alternative approaches can help students to learn more effectively, by applying student oriented exercises, like the substitution of regular lectures with the discussion of practical cases, the application of group dynamics, or the execution of a project from the beginning to its end by student workgroups.

Corresponding author:* **Rodrigo Alves Costa,
Universidade Estadual da Paraíba – Brazil.

In this scenario, this paper reports the experience of using the Scrum agile approach in the classroom environment in a software development project during the Project Management subject in the undergraduate course of the State University of Paraiba. The remainder of the paper is organized as follows: related work is discussed in the next section, followed by a materials and methods section which also presents the main mechanisms used in project development. Then, in the final section the article is completed and future works are presented.

Related Work

Wing (2016) reports the practical experience of undergraduate students in learning through the usage of problem-based teaching methodology (PBL), with the aim of contributing to the creation of a motivating environment, making the learning process more dynamic, collaborative and enjoyable. The work presents a strong relationship with the experience report presented in this paper, since it seeks to provide students with real life experiences. In contrast, a similar work by Meireles and Bonifácio (2015), which focused on the learning of undergraduate students by using the combination of problem-based teaching methodology (PBL) and the use of Scrum as a methodological tool for Project Management in order to construct real life projects applied to mobile devices, reported that students were divided into four (4) teams, and each one had the freedom to choose the theme of the mobile application. In the work presented here, although separated into similar

teams, the students joined efforts to develop a larger tool, performing a collaborative work and, in addition, a series of software tests were conducted in order to evaluate the quality of the project developed. Carvalho and Mello (2012) present the results of a research carried out in a small technology-based company, in which the objective of the study was to analyze the implementation of Scrum in software product development projects, as well as to understand and measure the impact of such implementation company-wise. The authors realized that with the adoption of the approach, communication was improved, as well as team motivation, project costs, time and risk were optimized, and team productivity was also increased. Branco, Coelho and Pires (2008) present the experience in the use of Scrum in offshore projects, highlighting the perceived advantages. Finally, it is possible to notice that there are several works that seek to promote for students the experience of a development process, but few specifically address the activities carried out in the academic environment as a means of learning, while the stages present in a production environment are closely followed and related to the theoretical content of the classroom. Thus, in this paper we report the experience in using the Scrum approach as an aid in the teaching process in the classroom environment.

Usage of the Scrum

Context

Scrum is an agile approach for the development of products and services [Cruz 2018]. According to Klein (2013), this approach does not define a specific technique for software development during the implementation stage. It focuses on how team members should work to produce a flexible system in a changing environment. There are three (3) important and well defined roles in Scrum, as follows [Wazlawick 2013]: (i) the Scrum Master is the facilitator and conflict solver; (ii) the Product Owner is the person responsible for the project itself; (iii) the Scrum team is the development team, and every member of such team interact to develop the product together. The Scrum cycle begins with a Stakeholder Meeting, during which the Project Vision is created [SBOK 2017]. The Product Owner then develops a Prioritized Product Backlog which contains a prioritized list of business and project requirements written in the form of User Stories. Each Sprint begins with a Sprint Planning Meeting during which high priority User Stories are considered for inclusion in the Sprint. The Sprint cycle process, seen in Figure 1, is triggered with the Prioritized Product Backlog, which contains all the functionality defined with the client / user, improvements and defect corrections to be developed in the project [Ramos and Junior 2017]. Based on this list, the Sprint is planned to run. Sprints are iterations during which product sets of Product Backlog (list of desired features) are implemented, and their duration can vary from 01 (one) to 06 (six) weeks [SBOK 2017]. In the beginning of each Sprint, the Product Owner, the Scrum Master, and the team participate in the planning meeting. In the first part of this meeting, the Product Owner, the Scrum Master and the team participate in order to select the tasks (Prioritized Product Backlog) that will be part of the Sprint Backlog as prioritized by the Product Owner. In the second part, the Scrum Master and the team participate to subdivide the user stories into the required tasks in order to better understand them to facilitate their implementation and development [Cruz 2017]. During the Sprint run, the Scrum Master and the team conduct daily meetings (which are named Daily Scrum Meetings) to keep track of the progress of the project. At the end of the Sprint, the team performs a presentation of the product increment developed for the Product Owner, whom in turn validates if what was requested corresponds to what was developed. Thus, the Sprint is finalized, starting a new cycle / iteration, until the final product is completed [Ramos and Junior 2017].

Project execution

For the development of the main project, the group of 20 (twenty) students was divided into 03 (three) teams. A questionnaire with questions related to Strengths and Weaknesses, Opportunities and Threats (SWOT) [SBOK 2017] was filled out by each student. The division of the teams happened through the affinity between the students verified through such SWOT analysis. Each team worked with different modules of the main software. In order to allow parallel development amongst the teams, development environment scenarios were created for each team using the GitHub tool, and these modules were integrated, synchronizing these environments in a single project. One member of each team was responsible for this task. There was a pre-set order for such synchronization to occur, and at the end of the process the main environment was all changed. The initial phase of the project was carried through research and studies sourced out from documents, book chapters and master's thesis related to each of the individual projects. Following the project comprehension phase, a meeting was held with the lecturer, who, during the project, assumed the roles of Product Owner and project customer. In this meeting, an interview was conducted to help clarify the objectives and requirements desired for the tool, as well as any concerns or questions and any other doubts from the development teams were solved.

Based on the requirements that were identified during the initial interview, each team was responsible for creating their Product Backlog according to their research, using a Google Docs spreadsheets for making it accessible for everyone at the same time, allowing access from any place independent of the access device. The fields composing the Product Backlog and subsequently that also make up the worksheet are: "desired" - requirement requested by the client; "In the form of ..." - column responsible for composing a justification of the requirement; "Type" - classification of the type of the requirement, i.e., a characteristic of the system or a presentation to be performed; "Priority" - receives a score according to the degree of urgency or importance of a request, allowing to identify an order of implementation; "Status" - responsible for storing the status of an implementation, being able to change between: TO DO (to be done), WIP (in progress) and DONE (completed); "Sprint Estimated" and "Sprint Done" - the first one stores the Sprint number on which the team expects to meet the requirement and the second one stores the actual Sprint number on which the request was actually served; "Responsible" - informs the person responsible for completing an item, finally; "Notes" - stores possible suggestions and additional information. After analyzing and documenting the requirements, the Sprint cycles started. To allow for this, before starting a Sprint, the teams selected the items that they understood that should be implemented, using the Sprint Backlog. In addition, for each new feature selected to be implemented, the progress status of the listed requirement was updated accordingly. In case such feature was not developed for any reasons by the end of the Sprint, it was returned to the Product Backlog to be allocated to the next Sprint, or to a future Sprint Backlog.

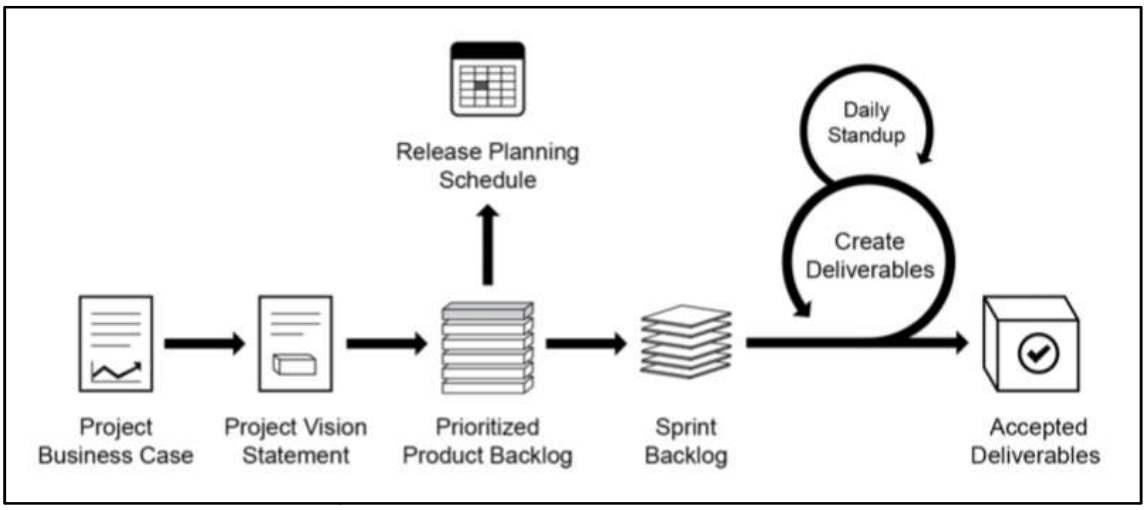
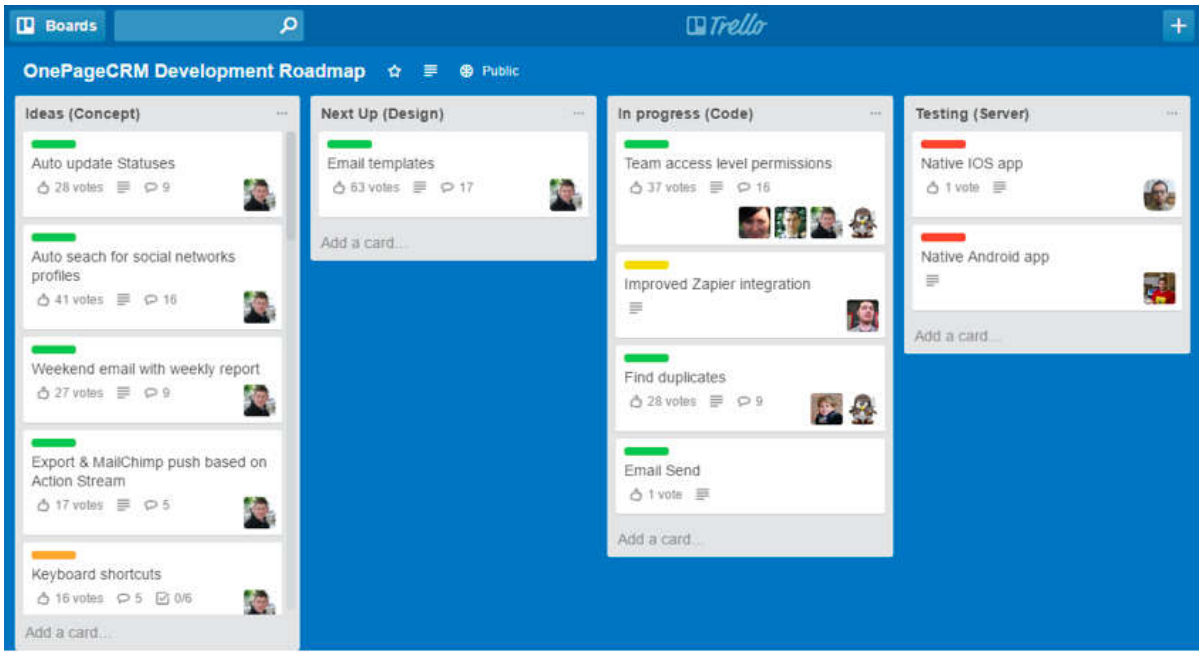


Figure 1. Scrum Development Cycle for a Sprint Source: [SBOK 2017]



Source: Author

Figure 2. Trello 2018 Extract from one of the Projects

The features selected for each Sprint Backlog were associated with a deadline for development and delivery to the client, and such deadline was defined at the beginning of the subject. The classroom agreed that each Sprint would last no more 15 (fifteen) days. This duration was chosen due to the time available for the semester, meanwhile facilitating the follow-up and tracking of the course. There was only one moment when the stipulated deadline was not met in the second Sprint. The SBOK (2017) accepts that during the initial interactions, due to the need of calibration and due to the expected learning curve, some Sprints allow for flexibility for adjustment – the class as a whole agreed to take the measure to extend the length of this Sprint length to three (3) weeks. For each team, a Scrum Master was elected, however this position was rotating - that is, in each Sprint a new Scrum Master was elected, allowing for all the students to having the opportunity to participate in this experience, thus performing the leadership role at least once. This tactic of rotation was particularly interesting and had a positive impact during the project development, because initially some of the team members did not possess the same level of commitment as the others.

However, it was noticed that the level of commitment of each member was increased when playing the role of a Scrum Master, as it was each member's responsibility to be the team's interface with the Product Owner, and all the criticisms, orientations and questions were directed to the Scrum Master, who had to take the responsibility to pass along such information to the rest of the team. This student was also responsible to remove impediments and roadblocks and tracking the activities of the group. However, it is worth mentioning that some members had difficulty taking the role of Scrum Master, most of the time because they either did not have a fully developed professional leadership profile yet or did not possess good communication skills. Most of the times though, hurdles like these were overcame with the help of the rest of team, which enhances the characteristic of a Scrum team as committed and self-organized. During the progress of the Sprints, each team was responsible to conduct daily meetings (Daily Scrum Meeting) to disseminate knowledge and update on the progress of the project. These meetings took place in two forms, face-to-face and virtual. In face-to-face meetings, the teams took advantage of the bi-weekly classes

appointments to establish the cadence of communications. In addition, it was necessary to resort to the intervals between the classes of the subject and the space reserved by the lecturer for discussion and updating of the progress to perform such meetings. In the virtual form, videoconference and chat resources were used, using tools available for social networking such as Skype, Hangouts and Facebook.

In addition to the meetings needed to keep the team aligned and on track, the Trello tool was largely used. Trello consists of an activity manager than can be used to track progress of tasks and it was used to replace the Scrum dashboard, allowing each team to effectively control the tasks, establishing an organization, and at the same time it was possible to classify the activities to be performed in "Next Up", "In Progress" and "Testing", as can be seen in Figure 2. After the completion of each Sprint, the completed tasks were reorganized and grouped according to Sprint numbering, thus maintaining a history. At the end of each Sprint, the teams performed validation activities, such as performing unit testing, integration, performance, and system tests, and compiling obtained results that were recorded through spreadsheets. At the end of a Sprint, it was necessary to perform Sprint Reviews to the Product Owner, in order to demonstrate the functionalities implemented and, consequently, defining the upcoming activities in the subsequent Sprints.

Project completion

When the scope of the development of all the three teams was complete and all the requirements were validated as finalized, the three modules were merged into a single system, unifying and creating a stable version of the developed tool, which was presented as the final work of the subject. Then, there was a rearrangement among the teams, through which the students started to work profile-oriented. Therefore, the 20 (twenty) students were split into 03 (three) new teams, and this time the students were free to choose the area of their greatest interest. This was done as a means to renew the commitment by the students and thus allowing for the activities to be completed more quickly. However, it was difficult to define the time required for the development of each task by each team member. The reason for that was because of the required by each student due to the new way of working required in the new structure, and also due to the fact that some of the members in the newly formed teams had never worked together as a Scrum team. In addition, each member had a schedule of availability and particular working pace, which also required some time for calibration. The assembled teams were basically organized as follows: development, testing and documentation. Throughout the project, the deliverables were managed using GitHub, which was chosen for being an open source tool for continuous integration and for including features that allow for documentation, code management, open source repository and enable the creation of groups of collaborators. Thanks to GitHub, source code, test results and all tool documentation are available in the repository that can be accessed publicly at a public address available on the Web.

Conclusions and Further Work

The traditional classroom environment needs new approaches in order to allow for the modernization of learning and to ensure that the teacher and students develop new experiences. In order to provide students with a real-world experience, based on methods used in the industry and in the labor market, and in order to generate and simulate such methods and experiences in the teaching environment, usual practices in the academy have to be adapted. The practical application of the Scrum method along with the development process presented positive and negative results. The benefits resulted in characteristics which can also be observed in software development corporations, such as obtaining software implemented on an increasingly fast pace and with satisfactory quality, promoting moments of learning during the project execution and simulating other real industry situations. The points to improve are not directly related to the methodology, but rather consequences related to the lack of practice and experience of the key team players engaged in the approach, who are still undergraduate students. Among the points of criticism raised by such students were the short deadlines for deliverables, since students reported being accustomed to longer periods and slower rhythms to deliver work; in addition, there were problems related to the handling of the Github tool, since the teams did not present previous knowledge of it, and it was necessary to study the tool and then perform joint meetings among the teams to share obtained expertise. Finally, it is very clear that the practical application of the agile Scrum methodology in a classroom environment was successful, presenting a functional system, whilst emphasizing in the involved students a sense of commitment throughout the process. As future work, we intend to apply Scrum in a new subject with the same group of students, allowing for a comparison between the results obtained in the subject of Software, which will probably allow for a more in-depth analysis. In parallel, the use of Scrum during the Project Management subject will go on, also allowing for the learning of new concepts related to Project Management.

Acknowledgments

Our thanks reach out to every student who participated in the entire development process using Scrum. The author would like to thank the Computer Science Course at UEPB and the Direction of the Centro de Ciências Exatas e Sociais Aplicadas (CCEA) for the support of the development of this research.

REFERENCES

Branco, Jônathas Diógenes Castelo; COELHO, Ciro Carneiro; PIRES, Carlo Giovano, S. 2008. Utilizando Scrum em projetos off-shore. In: I Congresso Tecnológico Infobrasil, Fortaleza.

Brennan, Karen. 2011. Creative computing: a design-based introduction to computational thinking. ScratchEd.

BRQ. 2016. Metodologias Ágeis de desenvolvimento de software, http://www.brq.com/metodologias-ageis/.

Carvalho, Bernardo Vasconcelos de; MELLO, Carlos Henrique Pereira. 2012. Aplicação do método ágil scrum no desenvolvimento de produtos de software em uma pequena empresa de base tecnológica. Gestão & Produção: Scielo, São Carlos, v. 19, n. 3, p.557-573.

Costa, Rodrigo Alves. 2016. Gerenciamento de projetos de TI. 2 ed. Rio de Janeiro. ESR.

Cruz, Fábio. 2018. Scrum e Agile em Projetos. 2 ed. Rio de Janeiro. Brasport.

Klein, Laura. 2013. UX for Lean Startups: Faster, Smarter User Experience Research and Design. 1 ed. Sebastapool, CA. O'Reilly.

Meireles, Maria Costa; BONIFÁCIO, Bruno. 2015. Uso de Métodos Ágeis e Aprendizagem Baseada em Problema no Ensino de Engenharia de Software: Um Relato de Experiência. In: Anais do XXVI Simpósio Brasileiro de Informática na Educação (XXVI SBIE). p.180-189.

Ramos, Aline Bentes; JUNIOR, Dalton. C. V. 2017. A Influência do Papel do Scrum Master no Desenvolvimento de Projetos Scrum. Revista Gestão e Projetos, São Paulo, v. 8, n. 3, p.106-119.

SBOK. 2017. A Guide to the Scrum Body of Knowledge (SBOK Guide). 3ed. SCRUMStudy, VMEdu, Inc.

Schwaber, Ken. 2004. Agile Project management with Scrum. Microsoft Press.

Sousa. Hanna Tátila *et al*. 2015. Apoio Automatizado ao Planejamento de Sprints em Projetos Scrum. In Anais XI Simpósio Brasileiro de Sistemas de Informação. SBC.

Vargas, Ricardo Viana. 2017. "Gerenciamento de Projetos: estabelecendo diferenciais competitivos."8ed. Rio de Janeiro. Brasport.

Wazlawick, Raul Sidney. 2013. Engenharia de Software: Conceitos e Práticas. Elsevier, Rio de Janeiro.

Wing, Jeannette. 2016. Pensamento computacional: um conjunto de atitudes e habilidades que todos, não só cientistas da computação, ficaram ansiosos para aprender e usar. In: Revista Brasileira de Ensino e Tecnologia, https://periodicos.utfpr.edu.br/rbect/article/view/4711/pdf

****