DOI: https://doi.org/10.46502/issn.1856-7576/2021.15.03.1

Scrum: A new framework applied to education

Scrum: Un nuevo marco aplicado a la educación

Antonieta Kuz
antonietakuz@gmail.com
https://orcid.org/0000-0002-8696-0859
Universidad Metropolitana para la Educación y el Trabajo
Buenos Aires, Argentina.

Recibido: 02/08/21
Aceptado: 20/10/21

Abstract

The digital transformation entails the inclusion and development of agile methodologies that allow adapting the way of working to the conditions of the project, achieving flexibility to the specific circumstances of the environment by working collaboratively. The extrapolation of Scrum to education pursues a purely practical and experiential learning, with the aim of achieving the full development of its autonomy, abilities and skills. Consequently, the objective pursued by this article is to highlight the different topics that are considered when working with Scrum and presenting one example of the application of agile methodologies in the classroom eduScrum.

Key Words: Scrum, Agile Methodologies, agile learning.

Resumen

La transformación digital conlleva la inclusión y desarrollo de metodologías ágiles que permitan adecuar la forma de trabajar a las condiciones del proyecto, logrando flexibilidad a las circunstancias específicas del entorno mediante el trabajo colaborativo. La extrapolación de Scrum a la educación persigue un aprendizaje puramente práctico y experiencial, con el objetivo de lograr el pleno desarrollo de su autonomía, habilidades y destrezas. En consecuencia, el objetivo que persigue este artículo es destacar los diferentes temas que se consideran al trabajar con Scrum y presentar un ejemplo de la aplicación de metodologías ágiles en el aula con eduScrum.

Palabras clave: Scrum, Metodologías Agiles, aprendizaje ágil.

1. Introduction

Today's world is in a stage of continuous change, which entails modifications in education. Since ancient times, classrooms have embodied a medium where knowledge can be disseminated, but today spaces outside the classroom are necessary as it is where students formulate their own notions and knowledge. In this way, the intelligent school is one that manages to interweave knowledge inside and outside the classroom. At present, there are several schools that re-pose the traditional way of teaching, so in this context it is essential to study various examples of schools in different countries of the world such as the United States, India, Spain, Finland (Sanmartín, 2016) among others with the explicit purpose of knowing in order to eventually and, gradually, implement a process of change and educational improvement.

According to the IEEE (1990), Software Engineering is the application of a systematic, disciplined and quantifiable approach to the development, operation and maintenance of software. A branch called Educational Software Engineering provides support when making software applications that implement learning processes. In addition, traditional methodologies are well known, but agile methodologies have been on the rise recently.

The agile movement seeks alternatives to traditional project management, as these approaches help teams respond to erraticity through incremental and iterative work cadences and empirical feedback. The implemented methodology provides opportunities to evaluate the direction of the project throughout the life cycle, through sprints or iterations; and this leads to you being able to build the right product.

Learning in this context focuses on skills for the 21st century, where the priority is learning to learn. Therefore, incorporating agile methodologies into the educational field requires adaptation to the teaching context, in general, and to that of the educational institution and subjects, in particular. In this way, educational institutions have begun to use Scrum to help teams learn to learn more effectively, in a more pleasant way, developing their capacities and collaborative work to a greater extent, strengthening teaching activity from a broader and renewed vision.

The objective of this article is to highlight the different topics that are considered when working with Scrum in classroom environments. It is structured as follows: in section 2, Software Engineering is contextualized, emphasizing the description of agile methodologies. Section 3 presents a brief survey of the XP, Crystal, and Scrum frameworks. In section 4 describes the methodologies. Section 5 describes Scrum in the classroom, briefly describing one application example eduScrum. Finally, the conclusions are presented.

2. Software Engineering: Context and Methodologies

Pressman describes Software Engineering as a discipline or area of Computer Science, which offers methods and techniques that allow the development and maintenance of quality software. It is made up of a series of templates that cover the methods, tools and procedures. Due to the particular characteristics of educational developments, it is necessary to consider the pedagogical aspects and communication with the user for each particular case together with the adaptation of the paradigms to educational theories (Salcedo Lagos, 2000).

All methodology must be adapted to the context, which can be successful applications in any software project, taking into account all the project variables such as development time, human resources, among others. On the one hand, there are traditional methodologies that, from their origins addressed a large number of project contexts, which without hesitation required a relatively high effort so that they can be adapted also considering the existence of changing requirements providing a disciplined approach to assigning tasks and responsibilities within a development organization. We find agile methodologies that provide a tailored solution for projects and provides controlled process, with few principles; flexible process with adaptation.

At the beginning of 2001, seventeen experts from the software industry met in Snowbird, Utah; with the objective of combining and describing the principles and values that they believed should allow teams to develop software quickly, responding to the changes that could arise during the course of the project. They sought to offer an alternative to traditional software development processes, which is why at that meeting the term agile was coined and they outlined the agile manifesto, a document that summarizes this philosophy. Following this meeting, a non-profit organization dedicated to promoting agile software development concepts was created, called The Agile Alliance.

2.1. Agile manifesto and principles

The agile manifesto values (Uribe y Valencia, 2007) the individual and the interactions of the development team on the process and tools (people are the main success factor of a software project, so it is essential to build a good work team that is the environment), developing software that works more than getting good documentation (not producing documents if they are not necessary), collaboration with the client more than negotiating a contract (looking for constant interaction between the client and the development team) and responding to changes rather than strictly following a plan (planning must be flexible and open). The values that shape the manifesto support its principles, these being the characteristics of agile processes:

1) The priority is to satisfy the customer through early and continuous deliveries of software that adds value.
2) Welcome and capture changes so that the customer has a competitive advantage.

3) Frequently deliver software that works from a couple of weeks to a couple of months, with the shortest possible time interval between deliveries.
4) Business people and developers must work together throughout the project.
5) Build the project around motivated individuals, give them the environment and support they need, and trust them to finish the job.
6) Face-to-face dialogue is the most efficient and effective method of communicating information within a development team.
7) Software that works is the primary measure of progress.
8) Agile processes promote sustainable development.
9) Continual attention to technical quality and good design improves agility.
10) Simplicity is essential.
11) Self-organized teams.
12) At regular intervals, the team reflects on how to become more effective, and adjusts its behavior accordingly.

3. Agile Frameworks

Taking into account and based on the Agile Manifesto, we can say that an Agile Framework can be defined as a specific software-development approach, and represents an overarching philosophy for software development, emphasizing the value of iterating quickly. There are many agile frameworks used and it is possible to modify parts of the frameworks as they see fit and as they iterate on their own agile processes. We detail below the most important agile frameworks:

Extreme Programming (XP): the project began in early March 1996 and enabled its developers to confidently respond to changing customer needs, even in the life cycle as it does emphasis on teamwork. In XP, managers, customers, and developers are equal partners in a collaborative team, which promotes self-organization around the problem. It provides a simple environment with simple rules based on values and principles, making it easy for teams to be highly productive.

Crystal: Alistair Cockburn developed a set of methodologies that are characterized by being focused on the members of the development team (this being a key factor) as well as reducing the number of artefacts produced. Along these lines, software development itself is considered a cooperative game that enables invention and communication, although limited by the resources to be used. The use of the word "crystal" refers to the various facets of a gemstone - each different face in an underlying core represents values and principles, while each facet represents a specific set of elements such as techniques, roles, tools and standards.

Scrum: is a process or technique for building products, and a framework that has been used to manage complex product development since the early 1990s. Jeff Sutherland et al, (2011) describes that Scrum was born as a new and different way of organizing human effort, rather than a way of conceiving work. This framework was given a name

that originated in rugby, where Scrum contained the perfect metaphor for what Sutherland understood as teamwork: coupling, unity of purpose, and clarity of goals. The Scrum team includes three roles: the product owner (decides what work should be done), the scrum master (acts as a helpful leader, helping the team and the organization to make the best use of scrum), and members of the development team (builds the product incrementally, in a series of sprints). A sprint is a fixed period of time with preference in the shorter intervals. In each sprint, the Scrum team will build and deliver a product increment, where each increment is a recognizable, operational, and visibly improved subset of the product that meets clear acceptance criteria and is built to a quality level called the Definition of Done (in English, Definition of Done). In Figure 1 you can see the summary of the relationship between artefacts and Scrum activities or tasks (Sutherland, 2016).

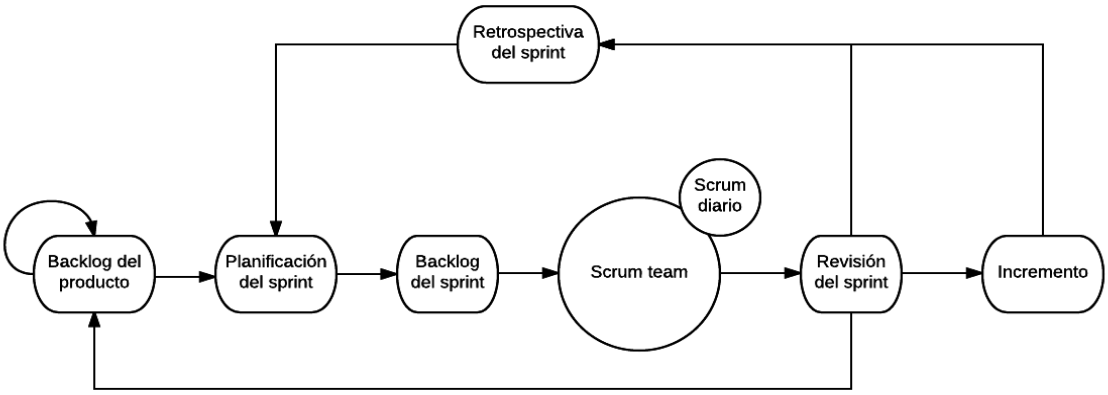

Figure 1. Scrum activities. Source: Own authorship.

4. Methodology

The research methodology is qualitative, through a documentary theoretical-methodological analysis, applying the search for information, organization and analysis of information.

5. The application of Scrum to the classroom

In the educational field, students need to develop general competencies, that is: capacities, abilities and aptitudes that will be useful in the academic environment and later in their professional career. Consequently, general competencies require active learning methods that enable the development of organizational capacity, planning, leadership, evaluation, self-evaluation, teamwork, among others. Online, agile methodologies offer principles, values and practices that can embody the solution to the context presented because the acquisition of skills is flexible and simple.

Based on the above, Scrum favors the creation of a conducive environment for students to be creative, enabling, on the one hand, that the classroom experience is enriching and

reliable, and on the other, the development of character with greater depth in the process learning, glimpsing the progress of the study through the successfully completed sprints. That is, Scrum through inspection, adaptation and transparency, becomes a learning framework (Pérez Benedi et al., 2011)

An analysis process is necessary for the implementation of Scrum in the academic environment, since it is essential to specifically define the context where it will be applied. This implementation presents two stages, the first concerns the way in which the agile process is transferred in the teaching process and the second, the identification of results will be the artifacts for it.

Agile classroom is designed through the integration of five (5) elements, where each of them can be combined in different ways in order to achieve specific objectives and the context. In fact, it will be most powerful when used synergistically as a complete system.

The elements are:

- Visible Class (in English, visible classroom): a visual learning management system.
- Learning Rhythm: a complete and iterative learning cycle.
- Collaborate: a situation model to increase collaboration capacity and defines the learning relationship between students.
- Empower: a situation model to increase the capacity for empowerment, and defines the learning relationship and the limits of choice between the teacher and the students.
- The journey: a scaffolding path to evolve any classroom towards self-organization. Integrate all other elements.

5.1. edScrum

The team model at eduScrum[1] (Delhij, van Solingen y Wijnands, 2015) is designed for optimal autonomy, collaboration, flexibility, creativity, motivation, and productivity. To understand how to apply it, it is essential to understand in advance the parts that make up a team. On one hand the teacher is the Product Owner and the responsible for determining what needs to be learned, monitoring and improving the quality of educational outcomes, and evaluating those outcomes always based on the definition of "finished" and the acceptance criteria. The teacher is expressly focused on the subject and should encourage cooperative work between teams. It defines the acceptance criteria that allow you to monitor the quality of what has been learned, for example: the minimum results of the assessments. Teachers sometimes act as super scrum masters, for the purpose of learning or teaching. On the other, the student team collaborates to achieve the learning objectives that are required at the end of the sprint, taking into account the acceptance criteria. The teams are self-organized, meaning that no one person can tell the team how to achieve the learning objectives. They are multidisciplinary, that is, they

[1] EduScrum, web site https://www.eduscrum.nl/es/resources/videos/, last visit, July 2021

have the skills to achieve learning objectives and personal development together. It is worth noting that students may have specific skills, but the responsibility is group.

To understand the process generally, sprints coincide with well-established periods and each of them consists of the points listed below. It should be noted that the composition of the team during the sprint and the scope are not changed, while the quality can be renegotiated with the teacher. In the same way, and unlike Scrum, a sprint cannot be canceled in eduScrum (Wijnands y Stolze, 2019) but it is possible that additional assignments are provided to achieve the expected results.

The sheet is a chronological representation of the work of the sprint that allows to achieve the learning objectives, and the tasks and assignments move according to how they change status: pending, busy and finished - students move the tasks of the panel to- do (pending) to done (finished) passing through “in progress”.

Two types of cards can be found on the boards (each of them is a chapter of a topic): stories (owned by the teacher) and tasks (owned by students). Now, on the first day of the sprint, the teams take each chapter and break it down into several tasks so that when all the tasks in the story are complete, all the students have learned that chapter. The impediments must be dealt with, such as various doubts that should be consulted with the teacher, among others; enabling the self-organization of students when distributing the work, explaining to each other and others. It should be noted that most of the class time is really effective study time and collaborative learning, also allowing retrospective on how to improve the process that they have followed and which parts they are going to eliminate, maintain or improve in the next sprint.

When all the tasks of a chapter are in the “donate” panel then the remaining teams and the teacher understand that the students in that group have already understood the topic, so the teacher can ask them questions about those learned chapters and in case that they do not dominate any of them, reposition the card to "in progress".

6. Conclusions

As presented at the beginning of the article, Software Engineering has undergone several changes, where agile methodologies can be adjusted to a wide range of software development projects where development teams are small, with short deadlines, fickle requirements, based on in ICTs, who need a customized solution, simple and with high levels of quality. These methodologies focus on people and the software product, valuing collaborative work.

In the educational field, the application of agile methodologies can be seen in examples such as agile learning where knowledge comes from experience and decisions are made based on what is known. With regard to the pedagogical layer, it is feasible to mention that it allows to achieve learning, Scrum seeks for students to learn effectively, because teams self-organize and work in sprints, which allows them to learn topics evolving the

learning process. Through this agile work method, it is feasible to improve the quality of classes and learning, motivating students to a greater extent. Stakeholders use reviews and retrospectives to evaluate learning processes in the same sprint, allowing improvements to be made in the next sprint, if necessary.

7. References

Delhij, A., van Solingen, R., & Wijnands, W. (2015). The eduScrum guide. The rules of the Game. Recuperado de: https://eduscrum.com.ru/wp-content/uploads/2019/10/The_eduScrum_Guide_EN_1.2.pdf

IEEE (1990). IEEE Standard Glossary of Software Engineering Terminology. En IEEE Std 610.12-1990, pp.1-84

Pérez Benedi, J., Yagüe Panadero, A., Díaz Fernández, Y., & Alonso Villaverde, S. (2011). Un Primer Paso a la agilidad: Retrospectivas para el aprendizaje de la ingeniería del SW. 2 Conferencia Agile-Spain CAS 2011, Castellón, España.

Sanmartín, O. (2016) Así son las escuelas más innovadoras del mundo. El Mundo. Recuperado de: https://bit.ly/31U8AqB

Salcedo Lagos, P. (2000). Ingeniería de software educativo, teorías y metodologías que la sustentan. Ingeniería informática, Nº 6, 1-9

Sutherland, J., van Solingen, D.M., & Rustenburg, E. (2011). The Power of SCRUM. CreateSpace.

Sutherland, J. (2016). Scrum: El arte de hacer el doble de trabajo en la mitad de tiempo. Océano.

Uribe, E. y Valencia, L. (2007). Del manifiesto ágil sus valores y principios. Scientia Et Technica, 13(34), 381-386

Wijnands, W. y Stolze, A. (2019). Transforming education with eduScrum. En Agile and lean concepts for teaching and learning (pp. 95-114). Singapore: Springer.