HOW TO IMPLEMENT SCRUM MANAGER IN THE SOFTWARE ENGINEERING ROUTE?

R. Grangel, M. Segarra, Ó. Belmonte

Dept. Llenguatges i Sistemes Informàtics, Universitat Jaume I (SPAIN)
Dept. Administració d'Empreses i Màrqueting, Universitat Jaume I (SPAIN)

Abstract

Teaching in Software Engineering has evolved in recent years at the same time that software development methodologies have. Therefore, there are a lot of good examples in the literature on how to use Agile Methods in the teaching of the subjects of this track. Most of them focused on explaining the teaching organization, the subsequent evaluation and the obtained academic results, as well as their benefits. However, there are still unanswered questions when implementing the principles of an agile methodology such as Scrum Manager in teaching practice.

This paper presents the result of a teaching experience of coordination of subjects in the context of Software Engineering carried out during four academic years. This has been formulated as best practices including the improvements that teachers have achieved in teaching after gathering feedback from students and the collaborative business environment. The main objective is to offer a help guide for teachers who want to use Scrum Manager in their subjects but who sometimes find problems that are not usually collected in the literature.

Keywords: Teaching Coordination, Software Engineering Track, Agile Methods, Scrum Manager, Best Practices.

1 INTRODUCTION

Nowadays, enterprises have been increasing the use of Agile Methodologies for software development [11]. These Methodologies are gaining wide acceptance as a methodology for the planning and management of computer projects, in the university context, and in particular in the subjects that follow Project Based Learning (PBL) methods [7, 8, 9, 12].

This work summarizes the experience of four teaching courses in the Degree in Computer Engineering, in a group of three subjects that use PBL and Agile Methodologies in the development of computer applications. Through this experience and the lessons learned in it, we propose a series of good practices to introduce Agile Methodologies jointly with PBL in computer science degrees. These good practices are categorized into three levels, depending on when they should be considered, either as an initial input into the development, during the development stage, or as a result of the development.

The rest of the paper is organized as follows. In Section 2, other similar teaching experiences are analyzed with the purpose of detecting synergies and possible improvements to the proposal of good practices. In Section 3 the context of the teaching experience is presented in order to make it repeatable. Finally, Section 4 shows the good practices analyzed, and the conclusions of the work are pointed out in Section 5.

2 PERFORMED WORK

The Project Based Learning (PBL) [3] has been, in many cases, the facilitator of entrance to Agile Methodologies in university teaching [7, 8, 9]. In particular, Scrum Manager is the most used Agile Methodology [5, 12]. The most common phases of the PBL process: (1) Define a problem or need to solve; (2) Group formation and research phase; (3) Specify the final product; (4) Evaluation of the entire process; they correspond to some of the phases of Scrum Manager: (1) Team formation; (2) Establishment of the Product Backlog; (3) Iterative creation of the Minimum Viable Product (MVP) through Sprints; (4) Review of Sprint; (5) Retrospective.

Some of the previous works found in the literature pay special attention to the improvement in time management when applying Agile Methodologies [5]. Others highlight the rapid capture of feedback in PBL through the use of Scrum [7]. Being the most proposed how to use or adapt Scrum in subjects that follow the PBL methodology. Techniques that are considered the cornerstone of the agile development of projects, such as Test-Driven Development (TDD), have been proposed as access to the agile development of projects, as in the case of [13].

In the cited works, a particular aspect of the agile development of projects is addressed; or a methodology for the implementation of these methods is proposed. However, there is no corollary of good practices that covers the entire agile development process drawn from the presented experiences.

The retrospective on our own work with Agile Methodologies and PBL allows us to point out a set of good practices in the implementation of these methodologies in the classroom, which allow, to a certain extent, to guarantee the success of the proposals.

3 CONTEXT OF THE EXPERIENCE

The teaching experience exposed in this article is based on the coordination of three subjects of the last course of the Degree in Computer Engineering of the *Universitat Jaume I de Castelló*: Business Initiative, Agile Methods and Software Engineering Workshop. The experience has been evolving over four academic years in a way that began with only the last two, which are optional subjects of the Software Engineering track. The main objective was for the students to carry out a single project to develop a web application that would serve as the coordinating axis and for the evaluation of both. In subsequent courses, the Business Initiative subject was added to the experience with the purpose of increasing the professional capacity of the students by promoting a series of transversal competences [1].

By incorporating this subject, the experience was adapted to the Triple Helix Model [4, 6] to establish synergies between the ecosystem of relationships that are exerted between the three helices (university, institutions and enterprises) with the aim of favoring innovation and knowledge transfer. The **helix university** is the teaching experience, while the **institutional helix** is represented by the *Parc Científic, Tecnològic i Empresarial de la Universitat Jaume I de Castelló (Espaitec)*, and finally the **business helix** is established through of the collaboration with diverse companies of the sector of the Information and Communications Technologies (ICT). Finally, it is worth noting the interdisciplinary nature of the teaching team, which endows the projects with a multidisciplinary and complementary approach.

3.1 Method of Work

The coordination of the aforementioned subjects was the result of trying to improve the professional competence of the students so that they would be able to undertake and innovate when proposing a computer project. For this, the three use, among other teaching-learning techniques, Project Based Learning (PBL) [3, 7, 9]. Furthermore, with the purpose of evaluating both generic and transversal competences [10], each team of students works on the same project that is started in the subject of Business Initiative and is finalized in the form of Minimum Viable Product (MVP) in the other two subjects.

In Figure 1 we can observe the organization of teaching in two semesters, while the subject of Business Initiative is taught in the first semester, the other two are taught next in the second and in parallel. In addition, the steps in the definition of the project carried out in each are detailed, as well as the main activities that are organized to promote the synergies of the Triple Helix Model.

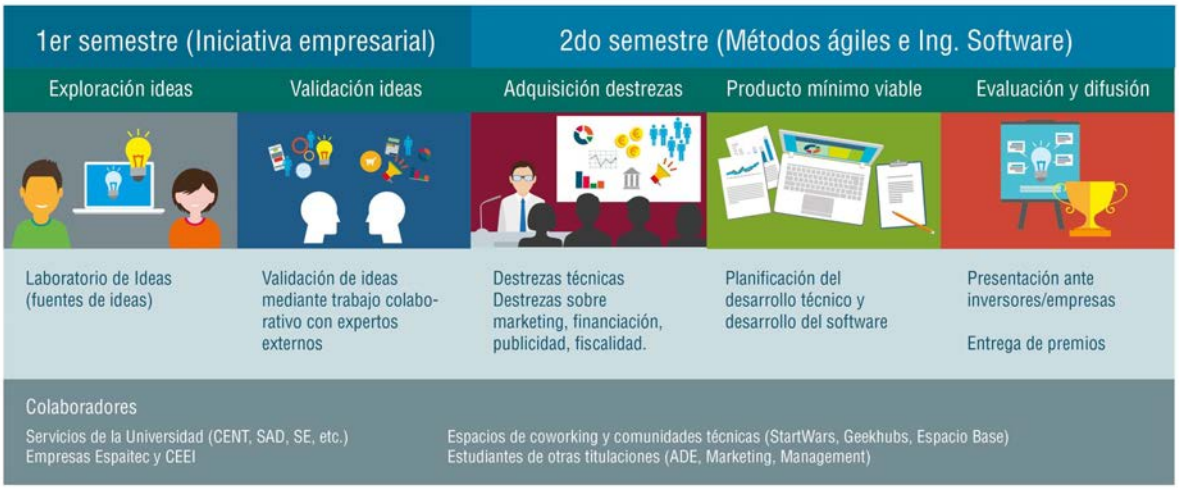
1er semestre (Iniciativa empresarial)
2do semestre (Métodos ágiles e Ing. Software)
Exploración ideas
Validación ideas
Adquisición destrezas
Producto mínimo viable
Evaluación y difusión
Laboratorio de Ideas (fuentes de ideas)
Validación de ideas mediante trabajo colaborativo con expertos externos
Destrezas técnicas
Destrezas sobre marketing, financiación, publicidad, fiscalidad.
Planificación del desarrollo técnico y desarrollo del software
Presentación ante inversores/empresas
Entrega de premios
Colaboradores
Servicios de la Universidad (CENT, SAD, SE, etc.)
Empresas Espaitec y CEEI
Espacios de coworking y comunidades técnicas (StartWars, Geekhubs, Espacio Base)
Estudiantes de otras titulaciones (ADE, Marketing, Management)

Figure 1. Teaching experience model [2].

3.2 Generic Competences

Regarding contents in each of the subjects, different generic competences are worked on the same project in a coordinated way throughout an academic year. Considering that each of the three subjects is assigned six ECTS credits, that means 450 hours of individual dedication to a project that is usually done in teams of five or six students. These generic competences are:

- **Business Initiative**: entrepreneurial and innovation skills are developed that make students start from a business idea about an ICT project until it generates a business plan.

  In order to exercise these competences, activities are carried out to generate and filter ideas and business opportunities. In a second phase, both internal mechanisms (teachers of the other two subjects) and external mechanisms (collaborators of the Triple Helix Model) are used to validate these ideas. And finally, the work team elaborates a business plan on the validated idea.

- **Agile Methods**: the competence of applying the appropriate techniques to the development of software projects, and in particular, those based on agile principles, as well as the critical capacity to determine the convenience or not of using this type of techniques considering the context and the requirements of the project.

  The project that the students have defined at the level of the business plan in the previous subject is analyzed and planned according to an Agile Methodology such as Scrum Manager [14]. Then, the different sprints are carried out for the development of the project and its follow-up through agile management tools.

- **Software Engineering Workshop**: technical skills related to the development of web applications are learned using leading technologies and tools. In particular, the server side focuses on the development of REST APIs from Java (JSR-311[1]). The client side focuses on javascript.

  The project that the students plan, execute and control follows the methodology Scrum Manager, it must be able to apply all the technical skills learned in this subject, since it is carried out in parallel with the previous one.

3.3 Continuous Improvement

The coordination process of the subject is carried out through regular meetings of all the teachers and the most punctual contact that is carried out informally but with enough assiduity. In addition, a very close relationship is established with the student, so that it is possible to collect an almost immediate feedback of the different activities that take place. In this way, it is possible to have a faster response than another kind of feedback as the institutional does not allow.

[1] JAX-RS: The Java^{TM} API for RESTful Web Services. https://jcp.org/en/jsr/detail?id=311.

At the institutional level, the university collects data through anonymous surveys of students on the progress of the subjects and the performance of the teaching staff. However, these results are usually given at the end of the academic year when the next one is already planned, so any improvement in the subject is only reflected after two academic years. Although these results have been important to verify the good progress of the subjects in these four academic years, the teaching staff has implemented other mechanisms for collecting information from students.

Therefore, the process to collect the students' feedback followed by teacher staff is based on the same agile principles that are the objective of these subjects. This process allows to gather the students' feedback not at the end of the process but more frequently in order to be more flexible and able make improvements if necessary. In addition to this feedback at the end of the project, information on the performance of entrepreneurial skills of students has been collected through surveys [2], and also on the progress of the subjects by holding a joint session with all teachers to identify opportunities for improvement.

4 GOOD PRACTICES

The process of continuous improvement that has been carried out during the last four academic years, together with the previous analysis of other teaching experiences in the field of Software Engineering for the implementation of Scrum Manager has led us to collect in the form of good practices those issues that are not usually addressed in depth in the literature on how to implement Scrum Manager in the subjects of Software Engineering.

We have divided the good practices into three sections, referring to whether they are prior to the Scrum Manager process, that is, they could be considered an input of the process; they have to do with the process itself; or with their evaluation, that is, they would be the output of the process.

4.1 Input: Case Study

In relation to the practical case, the teaching experience began offering the teachers the same case for all the students. From that practical case, each group of students had to implement their own project proposing some improvement or parameterizing the case in a way that was different from the rest. Already in the first year we saw that being a fourth-year subject, giving students such an academic case, and the same to everyone, did not favor their initiative and made all the work hours dedicated to the project have a very attractive result in terms of the project but not very useful at the same time as repeated.

It was for this reason that the input of the project was changed and the students were given the possibility of deciding on what practical case they wanted to do, inviting them to be as real as possible, on a topic they liked, a problem they thought they could solve by developing a web application, etc. At first, the teachers' fear was that the students, accustomed to a practical case drawn up by the teaching staff, would not want to do this work or even have no idea about what to do. To avoid problems if this type of situation occurred, the teacher prepared a standard case to provide the students in case they did not have any idea on which to do the project.

In addition, they were explained the reason why they were not provided with a practical case as they got used to have a case in other subjects of the Software Engineering track: (1) it is a greater effort that the students must define their own case study, but it also supposes to reach a higher level within the competencies according to Bloom's Taxonomy; (2) the result of this effort can have a real utility and more satisfaction for the students, since they choose what they like to work on. The response of the students was very positive from the start and all the groups proposed their own cases.

In the process of finding proposals for the students to innovate in their case studies, Business Initiative subject was incorporated into the teaching experience, which, although it does not belong to the Software Engineering track, is an ideal travel companion for this type of subjects. Since it allows the students to carry out a complete innovation process for the proposal of their project. In this subject, not only are techniques learned that favor the generation and filtering of ideas but they are stimulated with talks and events organized by the members of the Triple Helix Model so that their proposal has inputs from multiple sources and is as interdisciplinary as possible. On the other hand, the subject has a second stage in which the ideas are validated either by the faculty of the other two subjects that give feedback to the students, or by external experts that explain for example what funding possibilities it has and where they can pivot to have greater guarantee of success.

With all this, the students manage to define not only a project proposal or practical case but a business plan that has allowed them to investigate the potential clients, the business model, the possible competitors, how to differentiate themselves in the market, etc. At this point it should be noted that with the collaboration with ICT companies, which represent the business helix, students are offered the possibility of working on ideas that these enterprises have in stand-by. With this approach, we have seen in the last two academic years, how students are more motivated working with their own idea to which it gives shape during the process of creating the business plan.

4.2 Process: Roles

The roles that are established in the Scrum Manager process are the Product Owner, the Scrum Master, the development team and the Stakeholders [14]. These roles have been assigned to one or the other of the different actors participating in the Scrum process.

The role of Product Owner has been changing depending on who proposed or defined the case study. Initially when the case study was defined by the teaching staff and each group of students was responsible for adapting it to their vision, this role was exercised by the faculty. Since there are two subjects that are taught in parallel and depending on the organization of each year, the Agile Methods faculty or the Software Engineering Workshop teachers have exchanged the roles of Product Owner when doing the Sprint Review Meeting. This was so because the one who had a clear vision of how the product had to be was the one who proposed the case study and therefore in this case, the teaching staff.

However, this assignment generated numerous problems: (1) although the faculty exercised the role of Product Owner in the sprint reviews, this role was exercised by the whole team when it came to defining the Product Backlog; (2) since the role was exercised by the faculty of one or another subject, so that the teacher that did not participate in the review meeting was more detached from the product; (3) and perhaps the most important, as soon as students were given the freedom to make their own project proposal, it did not make sense for the Product Owner to be the teacher since those who have a clearer vision of what is going to be the project is the group of students themselves, so it is they who have to adopt that role. Therefore, it was clear that the role of Product Owner should be exercised from the group of students. Although this solved the third problem, the first and second remained unresolved.

To solve the first problem, a seminar was designed prior to the execution of the project that simulated a Sprint Planning meeting for the first sprint. In this seminar, not only this first meeting is carried out, but also the definition of the Product Backlog is previously made. In this task the whole group of students participates with the role of Product Owner and through the technique of the pyramid of cooperative work the Product Backlog is built and refined. In this way the whole group of students participates cooperatively, understands the work that is going to be done and assumes the role of Product Owner. Then, in the successive sprints the role of Product Owner is rotated among the team members and in each iteration, there is a person in charge of keeping the Product Backlog up to date and being responsible for validating the increase in the Sprint Review Meeting.

Regarding the second problem, the solution was given by assigning the Scrum Manager role to the faculty and having each of the professors participate in different meetings, as explained in the next section. The role of the Scrum Master was initially assigned on a rotating basis in each sprint to a member of the group of students. The main problem is that this role was usually confused with the leader of the project and therefore its role ended up being diluted and it was not a guarantee that the Scrum process was developed according to the rules. For all the above, it was considered that teachers have greater knowledge of the Scrum process and in their role of evaluator can also assume the functions of a Scrum Master making the process is carried out according to the rules, even eliminating impediments or facilitating the working conditions of the group. Now, the only drawback of this election is that it should be the faculty who led the Sprint Review Meeting in their Scrum Master role. Since it was believed that it was better for the meeting to be directed by the students as well, since it will be a situation that they will have to face professionally, an additional Scrum Master was designated for each group of students that also rotated along the sprints and whose only function was to lead the Sprint Review Meeting.

Finally, the role of Stakeholder was not adopted until the last course in which the need was seen to enhance the roles of the teaching staff as an important part of the Scrum application to the planning and follow-up of software development projects. What was done was to appoint a member of each team as a Stakeholder on a rotating basis in each sprint. In this way, the student to whom the role of

Stakeholder was assigned was required to attend the Sprint Review Meetings of the rest of the class groups. With this action, a double purpose is achieved and it is to reinforce the roles of the roles during the process and to make the students' learning greater when conducting an evaluation among equals. In addition to these meetings, the teacher of the Agile Methods subject also participates as a Stakeholder.

Thus, the roles are assigned in the following way:

- **Product Owner**: assigned to the entire group of students during the Product Backlog construction seminar and assigned to a group member of students on a rotating basis during the sprints.
- **Scrum Master*:*** assigned to the teaching staff of the subjects taught in parallel during the second semester for the entire process except for the Sprint Review Meeting that is assigned to a member of the group of students on a rotating basis during the sprints.
- **Stakeholder*:*** assigned to a member of the group of students on a rotating basis during the sprints and in addition to the teacher of the Agile Methods subject.
- **Development Team*:*** the rest of the group of students that does not have any of the previous rounds during the sprints.

4.3 Process: Meetings

It is important to carry out the meetings that the process indicates and to give them the appropriate documentary support. Since the first courses we have tried to transmit to the students the importance of the meetings and the necessary skills to carry out efficient meetings. In the Scrum process implanted in this teaching experience, three types of meetings have been carried out: Sprint Planning, Daily Scrum and Sprint Review, leaving aside other types of meetings such as the Scrum of the Scrum as they are not necessary.

As far as Sprint Planning is concerned, it is perhaps the most difficult to carry out since, with the exception of the first seminar in which the Product Backlog is also built, the rest of the sprint tends to dilute and not be carried out correctly. This produces that sometimes the Product Backlog is not updated with new user's stories that nevertheless yes that are being executed. In this sense it is necessary to be disciplined and the reassignment of roles has favored a better use of it. Finally, the laboratory sessions have been designed so that the review meeting of one sprint can be linked to the Sprint Planning meeting of the next to force a more formal development of it.

The Daily Scrum in the first courses was left to each group to do in its own way at the beginning of a new laboratory session. At present, this meeting has been changed and is carried out jointly by all the class at the beginning of the labs of the Software Engineering Workshop subject, with that it follows that actually all the groups carry out this type of meeting. Although there is the disadvantage of not being as agile as would be desired in a professional environment, the number of students in the laboratory group is not very high, and this disadvantage is compensated by the benefit at the academic level of the students' feedback. Since students listen to their own classmates on what subjects they are working on and the problems that have arisen, they can sometimes even be the same. It is important to note that the technologies used are novel and therefore knowing that a colleague from a different group has the same technical problem means that they can then look for a common solution.

Finally, the sprint review meeting has also undergone modifications since the first curves due to the different assignment of roles discussed in the previous section. While in the beginning the meeting was led by the teacher even when the role of Scrum Master fell on the student, by what is the teacher and is the one who evaluates, until the current situation in which the direction of the meeting falls on the student who in that sprint has the role of Scrum Master. Even though the teacher has assigned this role for the rest of the process.

4.4 Process: Tools

The tools used in the teaching experience have also varied throughout the four academic years in which it has been taught, although students have always been left free to choose the tool to be used to carry out the planning and management of the project according to Scrum. The main tools used have been:

- **Trello**: for the management of the Product Backlog and its follow-up by means of a Kanban board, it was used in the beginning for its simplicity and gratuitousness. Although it is very intuitive to use and quite powerful, it is suitable for all types of projects, it is not focused on software development projects and therefore there is some information that cannot be specified. In addition, it has the disadvantage of limiting the reports that can be obtained.
- **Jira**: used for the same purpose with an educational license. It is much more powerful, especially in terms of the reports it generates and being focused on software development can include more specific information and even link to incidents in the development of the code.
- **Confluence**: used to generate project documentation as a complement to Jira.
- **Slack**: is used to promote channels of communication between members of the student group and also the faculty. It allows better communication between teams without continuously using mail, a simple chat or WhatsApp, at the same time as it allows for a follow-up of conversations and the creation of channels for different topics.

4.5 Output: Evaluation

Finally, as far as evaluation is concerned, it has been unequal throughout the four academic years and we have realised that it is necessary to systematise it in order to be able to obtain more reliable results from the improvements that are being introduced. For this reason, in addition to the institutional survey that is carried out in most universities but which tends to have a late result, our experience is that it is necessary to establish our own mechanism for collecting information that allows us to collect data both on learning and on the process itself. To this end, we have defined three types of mechanisms, which are offered here as a possibility, but which undoubtedly have to be adapted to each particular case, even though their implementation is essential:

- **Survey on the performance of the assessed competences** in order to implement mechanisms that improve learning. In our case, we began by passing a survey on the performance of entrepreneurial competencies on the part of students, since this is one of the main objectives of the teaching experience. We have modified this survey throughout the courses in order to obtain more precise information on learning. The survey and its application during the academic year 2015/2016 can be consulted in [2].
- **Survey on the progress of the experience** in order to evaluate the process itself, to check that the path we have designed for the students is adequate, the activities are significant, they are well distributed over time, etc. This survey differs from the previous one in that it should not be centred on the acquisition of competence but on the process and the organisational part of the subject. In order to carry out this type of survey, it is very useful to use a star-grass day, as can be seen in Figure 2, each of the aspects that we want to include in the target being parameterizable.
- **Formal meeting with the students to evaluate the process**, although with the surveys a great amount of information is obtained in a very fast and anonymous way, and the informal feedback of the day to day with the students is very valuable, it is good to establish a formal meeting to carry out a process of reflection on the progress of the assignment. As far as possible, it should be set when the assessment process is finalised or about to be finalised in such a way that students feel free to respond and contribute their vision of how to improve the subject.

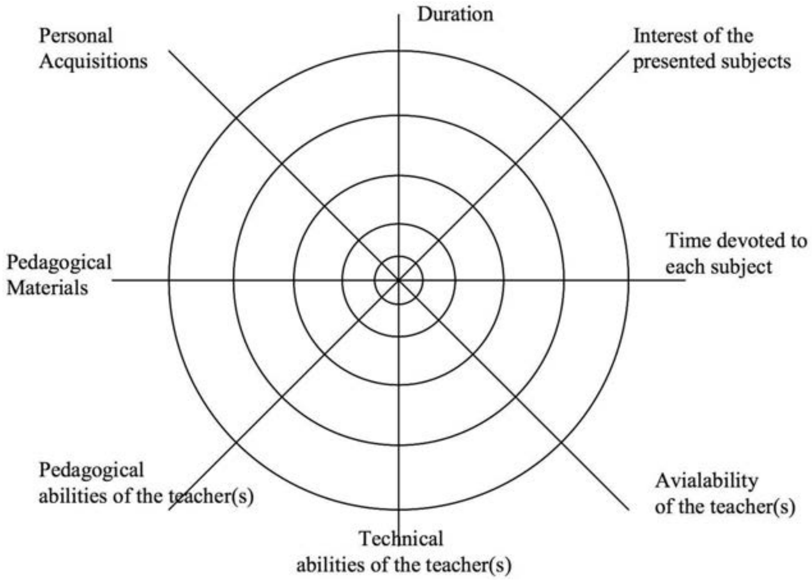
Duration
Interest of the presented subjects
Time devoted to each subject
Avialability of the teacher(s)
Technical abilities of the teacher(s)
Pedagogical abilities of the teacher(s)
Pedagogical Materials
Personal Acquisitions

Figure 2: Star diagram for evaluating the process.

5 CONCLUSIONS

The exercise of reflection on the process of implementation of Scrum Manager in our subjects has led us to write these good practices that are intended to be a guide for teachers who are making similar experiences. In them we explain the maturation process that the experience has had and how we have reached a situation that according to the feedback of the students and collaborators of the Triple Helix Model are better than the initial proposal that we had. Table 1 presents the good practices derived from this teaching experience.

Even with this reflection we are left with open questions that would be the object of another study and that arise in the classroom when the students have a more professional profile. Among other things, how do you present a budget to a client when you follow a Scrum Manager process, how do you choose the sprints in Scrum Manager when the project is not new but rather maintenance, how do you implement the Scrum of the Scrum in enterprises with more than ten employees, etc.?

Table 1: Decalogue of good practices for the implementation of Scrum Manager in the Software Engineering track.

Number	***Type***	***Description***
1	Input	It is better that the students are motivated by the project they are going to carry out, and therefore that they actively participate in its definition.
2	Input	It is necessary to prepare and accompany students in the process of generating new business ideas through a multidisciplinary team that makes contributions to the project from different points of view.
3	Input	The process of generating, filtering and maturing a business idea requires a certain amount of time.
4	Process	The role of Product Owner must be assumed by the students on a rotating basis in each sprint.
5	Process	The role of the Scrum Master must be assumed by the faculty except for the sprint review meeting which will be assumed by the students on a rotating basis in each sprint.
6	Process	Importance should be given to meetings.
7	Process	Although making a Kanban board with post-its can be very graphic and can be used in some class as an example is necessary the help of computer tools to facilitate the monitoring of the project.

Number	*Type*	*Description*
8	Output	It is necessary to collect feedback on the process as soon as possible to give an agile response.
9	Output	Each Sprint Review presents a project with reduced but viable functionality.
10	Output	The project documentation is generated during the entire development phase, it is not a final task.

ACKNOWLEDGEMENTS

The authors are grateful for the support of the educational improvement and innovation projects *GIE Enginyeria del Programari 2.0* and *SPIE Seminario Permanente de Innovación Educativa Aprendiendo a Emprender Colaborativa e Interdisciplinarmente*, both funded by the *Universitat Jaume I de Castelló*.

REFERENCES

[1] Ó. Belmonte, M. Segarra, R. Grangel, S. Aguado. Desde la Iniciativa Empresarial hacia el éxito pasando por Metodologías Ágiles e Ingeniería del Software. Actas de las XXII Jornadas de Enseñanza Universitaria de la Informática (JENUI 2016). 2016, Editorial AENUI, ISBN 978-84-16642-30-4. Premio SISTEDES 2016 al mejor trabajo presentado en el JENUI 2016 dentro del ámbito de SISTEDES (Sociedad de Ingeniería de Software y Tecnologías de Desarrollo de Software).

[2] M. Segarra, R. Grangel, Ó. Belmonte, S. Aguado. ¿Como potenciar el emprendimiento y la innovación mediante el desarrollo de proyectos de base tecnológica en el contexto docente universitario? Revista Tecnología, Ciencia y Educación, pp. 11-33, 2017, ISSN 2444-250X. Premio Estudios Financieros 2016, Accésit en la categoría Educación y Nuevas Tecnologías.

[3] Phyllis C Blumenfeld, Elliot Soloway, Ronald W Marx, Joseph S Krajcik, Mark Guzdial, and Annemarie Palincsar. Motivating project-based learning: Sustaining the doing, supporting the learning. Educational psychologist, 26(3-4):369–398, 1991.

[4] H.G. Chang Castillo. El modelo de la triple hélice como un medio para la vinculación entre la universidad y empresa. Revista Nacional de administración, 1(1):85–94, 2010.

[5] Luis Castillo. Resultados preliminares más significativos tras cuatro años de aplicación de la metodología scrum en las prácticas de laboratorio. JENUI 2017: XXIII Jornadas de Enseñanza Universitaria de la Informática (2017), p 281-289, pages 255–262, 2017.

[6] H. Etzkowitz, A. Webster, C. Gebhardt, and B.R.C. Terra. The future of the university and the university of the future: evolution of ivory tower to entrepreneurial paradigm. Research policy, 29(2):313–330, 2000.

[7] Pablo Fernández Montes, José María García Rodríguez, José Antonio Parejo Maestre, and Antonio Ruiz Cortés. Extendiendo ABP con feedback ágil en la enseñanza de Ingeniería del Software. JENUI 2017: XXIII Jornadas de Enseñanza Universitaria de la Informática (2017), p 281-289, pages 281–289, 2017.

[8] Alfredo Goñi, Jesús Ibáñez, Jon Iturrioz, and José Ángel Vadillo. Aprendizaje basado en proyectos usando metodologías ágiles para una asignatura básica de ingeniería del software. Jornadas de Enseñanza Universitaria de la Informática (20es: 2014: Oviedo), 2014.

[9] Alfredo Goñi, Jesús Ibáñez, Jon Iturrioz, and José Ángel Vadillo. Ingeniería del Software I: Aprendizaje Basado en Proyectos mediante metodologías Ágiles. Jornadas de Enseñanza Universitaria de la Informática (19na: 2013: Castellón), 2014.

[10] J. González, R. Wagenaar, et al. Tuning educational structures in Europe. University of Deusto Final report. Phase one. Bilbao, 2003.

[11] Kirsi Korhonen. Evaluating the impact of an agile transformation: a longitudinal case study in a distributed context. Software Quality Journal, 21(4):599–624, 2013.

[12] Patricio Letelier Torres and Mª Penadés Gramaje. Una estrategia para la enseñanza de metodologías ágiles. In Jornadas de Enseñanza Universitaria de la Informática (19es: 2013: Castelló de la Plana). Universitat Jaume I. Escola Superior de Tecnologia i Ciències Experimentals, 2013.

[13] Raúl Marticorena, Carlos López, and Yania Crespo. Estudio de la distribución docente de pruebas del software y refactoring para la incorporación de metodologías ágiles. Actas JENUI, pages 247–254, 2004.

[14] K. Schwaber. SCRUM development process. In OOPSLA'95 Workshop on Business Object Design and Implementation, 1995.