

Teaching agile methodologies in a project management course

Berbegal-Mirabent, Jasmina^a; Gil-Doménech, Dolors^a and Berbegal-Mirabent, Nídia^a

^aDepartment of Economy and Business Organization, Universitat Internacional de Catalunya, Spain.

Abstract

The increasingly dynamic, competitive and volatile business environment that characterizes today's marketplace demands for rapid delivery of high-quality outcomes, aligning development with customer needs and company goals. Agile methodologies have gained widespread popularity due to their easy implementation and adaptability to different industrial contexts. Although these methodologies originally emerged in the software and computer science field, they have been rapidly imported to other disciplines such as management and business.

This paper describes an activity developed in a Project Management course during the academic year 2016/17. The activity is designed with the objective of not only teaching students agile methodologies (and scrum in particular), but also to allow them understand the basis of such methodologies by implementing a class project. The activity details as well as students' perceptions are analyzed and discussed. The feedback collected indicates that students positively valued the activity and that they believe that through this activity they have been able to learn the fundamentals of agile methodologies.

Keywords: *Agile methodologies; scrum; project management; higher education.*

1. Introduction

Building innovative business models, processes and products require a new approach to management in general, and project management in particular (Highsmith, 2009). Project management needs to move faster, become more flexible and responsive to customers. Agile methodologies aim at fulfilling this role. They reduce complexity by breaking down the long cycle of building requirements and testing. Instead, they use small, usable parcels that can be designed and tested in manageable short cycles (Cubric, 2013). The term “agile” is used as an umbrella in the fields of software development and project management, and includes methodologies such as Scrum, Feature Driven Development, Dynamic Systems Development Method, eXtreme Programming, Lean, Six Sigma, or Kanban, among others. Agile methodologies promote frequent inspection, adaptation, and a leadership philosophy that stimulates teamwork, self-organization and accountability. Change and flexibility are thus the trademarks of agile projects. Agile puts less emphasis on up-front plans and strict control and relies more on informal collaboration, coordination, and learning.

This study reports the experience of a class activity in which students learn the basics of agile methodologies. Although there is a myriad of agile methodologies that emerge as alternatives to traditional project management, we focus on scrum, perhaps one of the most widely used approaches to agile project management. For most companies, the adoption of scrum is the first step towards agility, what have caused a considerable increase in the use of such methodology during the last few years (Mahnic, 2012). Specifically the project is expected to simulate different roles, ceremonies (meetings) and sprints as a way to give students a broad overview of this methodology, its foundations, principles and advantages. The experience reported took place at the Universitat Internacional de Catalunya during the academic year 2016/17. Students participating in this project were enrolled in the Project Management course, a compulsory subject included in the Master’s Degree in Business Administration and Production Systems.

2. Theoretical underpinnings

In recent years, organizations worldwide have increasingly adopted agile development methodologies (Salo & Abrahamsson, 2008). The adoption of such methodologies has been proved to facilitate the management of the development process, to reduce overtime, and to positively impact on customers’ satisfaction (Mahnic, 2012). Agile methodologies are particularly important in the project management field, because, when applied correctly in the right setting, they result in better productivity and quality projects with lower risk. In this context, it is of utmost importance that project managers understand the applicability of agile development methodologies to their projects. Likewise, they should underline the key factors that can drive their projects to a proper consecution.

As opposed to “traditionalist” development methodologies, which are based on extensive planning and categorized processes, agile methods rely on “people and their creativity” (Dybå & Dingsøy, 2008). The foundations are in the Agile Manifesto (2001). There are four principles: (1) individuals and interactions over processes and tools, (2) working software over comprehensive documentation, (3) customer collaboration over contract negotiation, (4) responding to change over following a plan (<http://agilemanifesto.org/>). In this context, agile methods present the advantage of rapidly accommodating to today’s fast changing markets (Karlström & Runeson, 2005; Qumer & Henderson-Sellers, 2008).

Despite the growth in the application of agile methodologies in the industry, academic research on this topic is still scarce, as most of the existing publications are written by practitioners or consultants (Abrahamsson, Warsta, Siponen, & Ronkainen, 2003; Chow & Cao, 2008; Conboy, 2009). A need for more empirical studies is thus obvious. In the specific context of agile project management, learning experiences are underrepresented compared to the use of agile methodologies in software-related disciplines. Although in the recent years this trend seems to start reverting (Cubric, 2013), there is still a long way to go.

Although there are several studies focusing on the teaching of agile methodologies (e.g. Devedzic & Milenkovic, 2011; Lu, & Declue, 2011), “the literature on teaching agile project management is just starting to emerge” (Cubric, 2013, p.121). Mahnic (2010; 2012) describes a course designed to teach Scrum and project planning to undergraduate students, concluding that the best way to teach agile is through projects and practical work. In addition, Cubric (2013) proposes a new method for teaching agile project management and found positive results with respect to the degree of learning acquired by students.

3. Description of the activity

3.1. Purpose

The expected product outcome of the activity was a flowerpot with a plant. Figure 1 shows the material packs that were given to students.



Figure 1. Material packs (one per group), ready for starting the activity.

3.2. Setting up team members

The 40 students enrolled in the Project Management course were grouped in groups of 4. Because when conducting this activity students were already working in teams in another research project, the teams for this activity were the same as the ones already established.

Within each group, different roles were assigned:

- 1 **product owner** (PO), representing the end customer and/or other stakeholders and responsible for:
 - Defining the requirements and the ROI of the product
 - Prioritizing
 - Validating
 - Defining *what* needs to be done, but not *how*
- 3 **team members**, representing a collection of individuals working together. Particularly, team members were expected to:
 - Build and deliver the requested and committed product increments at each sprint
 - Adhere the same norms and rules, showing respect to each other
 - Collaborate
 - Self-organize themselves

The instructor of the course was assigned to be the **scrum master**. Her role was to act as a PO facilitator and ensure all teams had what they need to get the job done.

Ideally the POs should have taken all decisions individually, but given the instructional objective of the activity, decisions were jointly discussed and agreed with the whole class.

3.3. Setting up the project

The first task of POs was to create the product backlog. In scrum, the predominant way to express features on the agile product backlog is in the form of user stories. These are short, simple descriptions of the desired functionalities told from the perspective of the person who desires the new capability (usually, the user or an internal customer of the system). They typically follow a simple template:

As a <type of user>, I want/need <some goal> so that <some reason>.

User stories were written on sticky notes and latter arranged on the blackboard to facilitate planning and discussion. Once user stories were created, the teams drafted the acceptance criteria (define when each of the user stories would be considered to be done). These criteria, also called as conditions of acceptance or story requirements, should be testable and easy to evaluate (with a simple “yes” or “no” answer). Next step consisted in prioritizing user stories and in assigning story points. Story points represent how hard it is

for team members to complete the story. In this context, hard refers to complexity, effort and/or uncertainty.

By way of example, Table 1 shows the product backlog that was agreed, so that all teams had the same scheme to follow.

Table 1. Product backlog.

Order	User story	Story points
1	As a user I want to have all the necessary material to start the project	2
2	As a user I need to create a seedbed that allows planting flowers	5
3	As a user I want to have a nice and ornamented flowerpot	10
4	As a user I need the pot with soil to sustain the plant	3
5	As a user I need a flower to plant	5
6	As a user I need to plant the flower	5
7	As a user I need the flowerpot + the plant to look gorgeous	5

Lastly, agreement in the length of the sprints (development cycle or iterations) was reached. Each sprint was limited to 10 minutes. Likewise, with the whole class it was fixed that the maximum amount of story points that could be supported in a sprint was 10.

3.4. Project development

For illustrative purposes, in this section we describe how the first two sprints were conducted. Subsequent sprints followed a similar structure. The activity finished when the product was completed (after 5 sprints).

Sprint 1

Given the limitation of 10 story points per sprint, POs with their teams decided that the first sprint would only include the first two user stories, as in the example:

- As a user I want to have all the necessary material to start the project (2 points)
- As a user I need to create a seedbed that allows planting flowers (5 points)

Before starting with the sprint, the instructor provided details on how daily stand-ups usually work. These, are meetings in which attendees typically participate while standing. The discomfort of standing for a long period is intended to keep the meetings as short as possible. Due to the limited time assigned to each sprint, stand-ups were not conducted.

At the end of the sprint, the “review” ceremony was performed. In this meeting, the POs of each group decided which of the user stories were “done”, following the acceptance criteria defined before. Figure 2 shows how the classroom was arranged for the activity.



Figure 2. Project development.

Sprint 2

Following the example in Table 1, the second sprint focused on the user story ranked third (*As a user I want to have a nice and ornamented flowerpot*). Because this activity had 10 story points, this sprint was entirely devoted to accomplish this story.

Half way through the sprint and following the scrum methodology, a refinement meeting called product backlog grooming (in reference to keeping the backlog clean and orderly) was held. The outcome from this meeting was a clear consensus to break down this user story because it was too big (the initial estimation was too optimistic). Accordingly, the user story was divided into two: the first part should be finished by the end of this sprint, while the second part was moved to the next sprint.

This iteration concluded with the sprint review and the sprint retrospective. In the former, the review looked at what the team was building. In the later, the ceremony was led by the scrum master, and teams discussed the gains and pains of the just-concluded sprint in order to determine if anything needed to be changed to make the next sprint more productive. Resulting from this meeting, some story points were adjusted. Also, potential conflicts between team members were addressed in order to align the project's goal with team members' actions and commitment.

4. Assessment and results

Once the final product was obtained (see Figure 3), a debriefing session was held, consisting in a review of the fundamentals and principles of agile methodologies. Students were encouraged to share their experiences during the activity and link their perceptions with the pros and cons of this methodology. Because students played different roles, they were also able to experience the responsibilities each role entailed and deeply understand their participation in the various ceremonies of the scrum methodology.



Figure 3. Final products.

This session was also an excellent opportunity to discuss students' opinions and gather their suggestions for improvement. Aiming at formally capturing students' feedback, an anonymous survey was designed. Table 2 displays the results.

Table 2. Students' feedback (scale from 1 – lowest to 5 – highest)

Questions	Average	SD
The activity was well organized, prepared and structured	4.27	0.90
The lecturer encourages student to participate in the activity	4.27	1.01
The instructions of the activity are clearly communicated by the lecturer	4.55	0.69
Teaching resources were appropriate in that they facilitate learning	4.64	0.67
This activity has helped me improve my knowledge, skills, or attitudes	4.36	0.81

As it can be observed, all items in the survey were highly evaluated (>4, in a 1 to 5 scale). The most outstanding items were the appropriateness of the resources for the learning process, and the way in which the instructor communicated the instructions of the activity. It is also remarkable that students converged in the usefulness of the activity as a means to improve their knowledge in this topic.

5. Discussion and conclusion

This paper describes the details of an activity carried out during the fall semester of the academic year 2016/17 in a course of Project Management. In order to prepare students for the increasing use of agile methodologies in industry, this activity was designed exposing students to agile methodologies, and scrum in particular.

Survey results confirm that students showed a positive attitude towards learning agile methodologies through practical project work. It becomes evident that students were enthusiastic about the practical approach used in the activity. We hope these results encourage other lecturers to design similar activities.

One limitation of this experience is that, although being a hands-on activity, the main output was building a flowerpot with a plant. Future editions might consider designing project activities that are more close to real world industrial environment, this way students would experience challenges similar to those project managers face in their daily routines. Forthcoming instructional experiences might also consider expanding this activity to other agile methodologies such as XP, which has been proven to be very effective when combined with scrum. Likewise, students' performance in the form of grades might also be useful in order to test whether the knowledge acquired has been assimilated.

References

- Abrahamsson, P., Warsta, J., Siponen, M. T., & Ronkainen, J. (2003). New directions on agile methods: A comparative analysis. In *Proceedings of the 25th International Conference on Software Engineering (ICSE'03)* (pp. 244–254). IEEE Press.
- Chow, T., & Cao, D.-B. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software, 81*(6), 961–971.
- Conboy, K. (2009). Agility from first principles: Reconstructing the concept of agility in information systems development. *Information Systems Research, 20*(3), 329–354.
- Cubic, M. (2013). An agile method for teaching agile in business schools. *International Journal of Management Education, 11*(3), 119–131.
- Devedzic, V., & Milenkovic, S. A. R. (2011). Teaching agile software development: a case study. *IEEE Transactions on Education, 54*(2), 273–278.
- Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology, 50*(9–10), 833–859.
- Highsmith, J. (2009). *Agile project management: creating innovative products*. Pearson Education.)
- Karlström, D., & Runeson, P. (2005). Combining agile methods with stage-gate project management. *IEEE Software, 22*(3), 43–49.

- Lu, B., & DeClue, T. (2011). Teaching agile methodology in a software engineering capstone course. *Journal of Computing Sciences in Colleges*, 26(5), 293–299.
- Mahnic, V. (2012). A Capstone Course on Agile Software Development Using Scrum. *IEEE Transactions on Education*, 55(1), 99–106.
- Qumer, A., & Henderson-Sellers, B. (2008). A framework to support the evaluation, adoption and improvement of agile methods in practice. *Journal of Systems and Software*, 81(11), 1899–1919.
- Salo, O., & Abrahamsson, P. (2008). Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of Extreme Programming and Scrum. *Software, IET*, 2(1), 58–64.