Project-Based Learning and Agile Methodologies in Electronic Courses: Effect of Student Population and Open Issues

Marina Zapater, Pedro Malagón, Juan-Mariano de Goyeneche, and José M. Moya

***Abstract*—Project-Based Learning (PBL) and Agile methodologies have proven to be very interesting instructional strategies in Electronics and Engineering education, because they provide practical learning skills that help students understand the basis of electronics. In this paper we analyze two courses, one belonging to a Master in Electronic Engineering and one to a Bachelor in Telecommunication Engineering that apply Agile-PBL methodologies, and compare the results obtained in both courses with a traditional laboratory course. Our results support previous work stating that Agile-PBL methodologies increase student satisfaction. However, we also highlight some open issues that negatively affect the implementation of these methodologies, such as planning overhead or accidental complexity. Moreover, we show how differences in the student population, mostly related to the time spent on-campus, their commitment to the course or part-time dedication, have an impact on the benefits of Agile-PBL methods. In these cases, Agile-PBL methodologies by themselves are not enough and need to be combined with other techniques to increase student motivation.**

***Index Terms*—Project-based learning, agile methodologies, scrum, engineering education, electronics.**

Original Research Paper
DOI: 10.7251/ELS1317082Z

I. Introduction

ENGINEERING education is one of the most modern sciences, specially when compared to physics, mathematics or medicine. However, the teaching methods have been historically dominated by the "chalk talk", emphasizing the teaching of the mathematical and physical background. However, according to a report by the Career Space Consortium [1], industry demands today's engineers to apply their knowledge in real-life situations. Moreover, personal and business skills, including team work skills, are considered to be an important part of the curriculum. Thus, university education needs to provide students with hands-on experiences and practical learning skills, that enable them to become competitive graduate students.

This need becomes particularly important in telecommunication and electronic engineering studies, because of the mixture of computer science, signal processing and communications courses. Hands-on experiences and practical scenarios help students understand the basis of electronics, as well as the design and integration phases of a product, and need to be a key part in the engineering education program.

Because of the particular needs of engineering education in general and electronics in particular, Project Based Learning (PBL) and Agile methodologies become very interesting instructional strategies. When adopting PBL in the classroom, the benefits sought are: the usage of a student-centric approach that allows the development of interdisciplinary skills (i.e. planning, team work or the learning of tools such as version control systems) and its beneficial impact on student motivation.

Agile methodologies, such as Scrum [2], outline methods for flexible and adaptable software development. Even though these methods come from the business pressure on software development companies, they have moved into the academic world quite fast, mainly because of their beneficial results on student learning and motivation. However, because they are initially thought to be applied to the industry world, the principles of the Agile Manifesto [3] need to be mapped into pedagogical principles [4].

In this paper we analyze and evaluate the results of applying Scrum and PBL in two electronic engineering courses of the Electronic Engineering Department at Universidad Politécnica de Madrid. The main teaching goals pursued when introducing these methodologies in the classroom are the following:

- Engaging students in real-world tasks, with realistic constraints in economic cost and development effort, that enable them to handle complex systems. From the electronic perspective, a whole team working on the design and implementation of a solution provides more work power than individuals. Students need to learn how to tackle cost and time design constraints, by negotiating the scope of a project with customers without a penalty in the quality of the solution.
- Cooperation and team work in a self-regulated environment. Students should be able to organize themselves, dividing the work in a way that enables them all to learn, and then integrate work together to deliver a solution. For an effective team work, they need to understand the benefits of good planning and the usage of development tools, e.g. software version control tools.
- A better understanding and learning of electronic design,

Manuscript received 23 November 2013. Received in revised form 19 December 2013. Accepted for publication 20 December 2013.

M. Zapater is with CEI Campus Moncloa, UCM-UPM, Madrid, Spain (e-mail: marina@die.upm.es).

P. Malagón, J.M. de Goyeneche and J.M. Moya are with the Electronic Engineering Dept. of Universidad Politécnica de Madrid, Madrid, Spain (e-mail: {malagon,goyeneche,josem}@die.upm.es).

that delivers high-quality solutions to a problem. Students are faced to a problem that needs to be solved. They have to learn new techniques in the areas involved in the design and implementation of their solution. To match the customer requirements, they have to take decisions during the design process, coping with incompleteness and imprecise information.
- Leveraging creativity and pro-activeness: students are encouraged to find their own solutions to a problem, either by becoming proficient in certain areas, or by seeking the advice of experts (i.e. instructors).

The Electronic Engineering Department where this courses are taught, has a comprehensive background on PBL strategies and many years of experience in several courses of the electronics area [5]. More recently, the Scrum Agile development framework started to be used as teaching methodology in one undergrad PBL course. Until 2011, all PBL courses were part of the electronics curriculum of undergrad students. However, a couple of years ago, an Scrum PBL course also started to be taught in the master of science on electronic engineering. The contents of both courses, the undergrad and the master course are similar in terms of the contents taught: both focus on embedded systems design, covering from the system architecture and interface design to the programming of microcontrollers and drivers, and sensor conditioning.

In this work we evaluate Scrum-based Agile-PBL methodologies in two different courses, one undergraduate and one master course, with similar teaching goals from the electronic curricula perspective, but in which the students' profiles are very different, specially in terms of their dedication and engagement to the course. We analyze the influence of some population factors such as student motivation, group cohesion and part-time study in the results of Agile approaches in engineering courses. We show the impact of accidental complexity in our methodology, i.e. problems due to inefficient team works, planning, tool complexity, or any other impediment that diverts the attention from the electronics learning process. We analyze how this complexity affects the instructors and reduces the amount of time spent in really useful electronics-related problem solving, and propose contention methods to minimize the impact of accidental complexity.

We discuss solutions to the above mentioned problems and propose a Project-and-Problem Based Learning (P^2BL) that is being applied to former students this semester. The technique consists on generating real-life problems that affect the usual PBL work flow and put students face to face to common situations in a real-world scenario that they would not encounter in a typical classroom PBL project.

The remainder of this paper is organized as follows. Section II details the related work on the topic. Section III describes the methodology followed in the two courses, whereas Section IV shows the results and discussion. Finally, the most important conclusions are drawn in Section V.

II. Related work

For years, researchers from all areas, but especially from the pedagogy and psychology fields have claimed that from a constructivist perspective, humans are active learners that construct knowledge based on their effort and experience. Vigotsky [6], one of the better-known researchers in social constructivism, highlights the importance of the social context and the interaction with others in the process of thinking that leads to learning. From his perspective, students cannot be passive subjects that absorb knowledge from an instructor. Research in the area of engineering education also supports this claim and shows that learning and retention are more likely [7] when students are actively engaged in the learning experience. These works support that student-centric approaches, including PBL learning or Agile methodologies, which involve the student in the learning process, are likely to increase student motivation. In general, the feedback provided by students is very positive and the impact on learning is large [8]. Applying Agile methodologies to engineering courses implies the mapping of principles from the Agile Manifesto to the classroom. Previous work by Stewart et.al. [4] shows how this mapping can be done, and the considerations needed to be taken into account in the teaching scenario.

Most of the experiences in PBL applied to engineering education have focused on computer science courses for software development [9], [10]. Fewer approaches tackle the implementation of PBL and Agile methodologies in electronic design, hardware or hardware-software integration [11]. In order to design and implement complex systems, the project is developed by a group of students, between 7 and 14. However, complex systems and more manpower means more complexity in the project management. Because of the nature of the problems encountered by electrical and electronic engineers in the real world, applying PBL and Agile methodologies to electronics is particularly beneficial. On the one hand, it enables students to manage more complexity and face design decision problems with constrained costs and effort. On the other hand, professors have daily feedback checking the methodology reports and the group is focused on the achievement of the sprint goal.

The most common metric to evaluate the beneficial impact of PBL and Agile methodologies in education is student satisfaction, quantified by means of questionnaires filled in by students [12]–[14]. Moreover, research by Layman et.al. tries to correlate the appeal of Agile techniques with the personality types and learning styles of students [15]. Even though several works only provide the metric of student perception, others complement the data with course marks obtained by students [16], or even by monitoring the student activity (e.g. via version control tools) to identify metrics that would allow instructors to act proactively and identify patterns of low engagement and inefficient team work [17]. We propose the usage of both metrics, i.e. student satisfaction and students marks, in a quantitative way to measure the benefits of Agile methodologies in two different courses, and compare results to those obtained in traditional courses.

Several works highlight in a qualitative way the most common issues found, such as the quality of the results being dependant on certain people, or the difficulties encountered to evaluate the time spent by students to learn and complete assignments [14]. Research by Kagan. et al. [18] highlights the importance of the teacher in the management of issues and considers him the customer that, within the scope of Agile methodologies, is part of the project team and establishes and negotiates the system requirements. Our work leverages this concept by analyzing the impressions of students and instructors, comparing the results obtained in the two PBL courses with traditional courses to understand the different problems that students encounter and how they can be solved. Moreover, we evaluate the impact on team work of part-time students enrolled in PBL courses.

III. Method

In this section we present the method followed in the two Agile-PBL courses under study, providing information on the structure of the study and the teaching goals, the project to be undertaken and the assessment.

A. Scope of the study and teaching objectives

The two Agile-PBL courses under study are respectively LSEL ("Laboratory of Electronic Systems") undergraduate course and LSE ("Laboratory of Embedded Systems") Master course. LSEL belongs to the Bachelor of Telecommunication Engineering, whereas LSE belongs to the Master of Electronic Engineering. Both courses are elective, take place during the winter semester, and the average number of students enrolled in each course is 10. In this paper we analyze the results corresponding to the last two academic years: 2011 and 2012.

The program of both courses is similar and covers the following topics:

- Electronic system architecture design, including the integration of various systems, and the design of their interfaces
- Design and implementation of sensors and actuators
- Microcontroller programming and communication protocols
- Linux device drivers programming
- Development of automatic benchmarks to test and debug the system

Moreover, because students work in groups to pursue certain goals, they are also required to:

- Learn to use software version control tools to share code with their team mates
- Analyze and measure the amount of time spent in tasks
- Division of complex tasks in simpler ones
- Tasks planning and effort estimation in terms of manpower
- Development of communication skills to interact with their team mates in an effective way.

From a pedagogical perspective, the particular goals that the professors of these courses pursued when implementing the methodology were the following:

- To propose the students a real-life project, similar to one they could face in industry, with more ambitious goals and a broader scope than other projects proposed in traditional pair-programming laboratories.
- To teach students a collection of new techniques regarding planning, code sharing, collaboration, team work and management.
- Motivate students towards electronics, leveraging proactiveness and creativity, letting them take their own design decisions according to the different abilities of the team.
- Understanding and learning the basics of electronic design and programming by devoting time to think and interact with the team members, producing high-quality solutions to problems.

B. Project undertaken and methodology

All the students in the course have to carry out together a complex project. The projects undertaken vary depending on the course and the academic year, but have the same methodology and planning, and the tasks they have to perform are similar. For the purpose of clarification, we briefly describe the projects developed by students in the courses and years under study: i) an automated model train that travels at variable speed depending on the traffic signaling in the railway and stops in each station, respecting barriers, ii) the electronic control of a car, including a motherboard that controls sensors (temperature, rain, doors opened) and actuators (lights, automatic cleaners, airbag, ABS, etc.) via a CAN bus; and iii) an intelligent house with automatic lights, flooding sensors, fire and security alarms that alert the owners via e-mail or SMS.

At the beginning of the course, the objectives of the project as well as the teaching goals are presented to students. There is just one team in the course, composed of all the students enrolled. The teacher plays the role of a customer, and discusses with the team the scope of the project. Once students and instructor agree on the requirements of the product, they generate a product backlog and students organize themselves, naming a team leader, and distributing themselves to different tasks.

Following the Scrum methodology, the project is divided into 5 sprints, each one with a duration of 2 weeks. In each sprint, tasks are prioritized according to the customers' needs and students estimate the effort needed to complete each task. The time per week devoted by students to the project is fixed below a maximum by the teacher, so the scope is flexible and can be modified by negotiating with the instructor. Depending on the effort estimation, students are committed to develop a certain number of tasks for that sprint. Tasks are further divided into sub-tasks and assigned to specific students or groups.

Students are required to hold weekly team meetings that allow them to be aware of the evolution of the overall project. Because the course officially consists on three hours of laboratory per week, these hours are used to hold the weekly meetings and discuss with the instructors. Students are also asked to track the time they devote to each sub-task in the sprint, so that in every meeting impediments and bottlenecks can be easily spotted. They also need to fill-in the sprint burndown chart every day, that indicates the total remaining team task hours within one sprint.

In the present method, the teacher plays multiple roles: customer, advisor and expert in the field. Scrum methodology includes two special roles, that are performed by two students: the Product Owner and the Scrum Master. The Scrum Master is responsible for the correct implementation of the methodology by the team, the preparation of the Scrum tool, checking that it is being used properly, and the fostering of the team interaction. The Product owner is responsible for the communication with the client in order to solve impediments that might appear. The course does not have a fixed amount of theory classes, but students are advised to request teachers a theory lesson when they need more background on a certain topic. These lessons are requested at any time by the Product Owner, who negotiates the topic with the professor.

At the end of each sprint, the students present the results obtained (i.e. tasks performed) in that sprint. Students must always deliver, at least, one complete functional task per sprint. They comment the impediments and difficulties encountered and negotiate with the instructor the next tasks to be performed, the priorities and, again, estimate the effort.

At the end of the semester, the students have to make an oral presentation to the customer and the open public, presenting the work developed and the goals achieved. They also have to handle high-quality technical documentation on the solution.

C. Assessment

The assessment of the courses is divided into three different aspects. Some aspects have an impact on the final mark of the course, whereas others are just gathered for the purpose of qualitative and quantitative analysis of the PBL courses. The goal is to be able to assess the proficiency of students in the following areas:

- Delivering a product that meets the requirements of the customer.
- Learning the basic electronic concepts of the course, completing tasks with the highest possible quality given the time constraints.
- Facing the challenges proactively, proposing alternatives that lead to a solution.
- Using the planning and software version control tools provided by the instructors.

1) Teachers: Teachers evaluate the students in two different ways: i) grading the overall team and ii) grading students individually.

In each sprint, teachers evaluate the team work, in terms of the number of tasks accomplished and delivered to the customer, grading them from 0 to 10. Only functional tasks are evaluated, and marks are given depending on the degree of accomplishment of the requirements and the quality of the work. Non-functional tasks count as not delivered and are not evaluated at all.

At the end of the semester, the grades obtained by the team in each sprint are averaged. Team work evaluation mark has a weight of 40% on the overall course grade. As can be seen, final assessment is highly dependent on team work evaluation. This way, students always give more importance to task delivery and team work than to individual aspects and competitiveness within the team members.

Teachers also provide an individual grade at the end of the semester to each student. This mark has a weight of 40% on the overall course grade. It is based on the individual degree of proficiency students showed on the tasks that were assigned to them during the course, as well as on the management and team work skills exhibited.

2) Team mates: In each sprint, all the students provide an overall assessment of their team mates in terms of their ability to accomplish the assigned tasks, their team work skills, and their contribution to the project.

At the end of the semester the grades obtained by a student in each sprint are averaged. The assessment of team mates has a weight of 20% on the overall course grade.

3) Individual assessment: For teachers to gain a deeper knowledge on the learning process of students and to obtain their opinion on how profitable the course was, students have to fill-in some self-assessment questionnaires, which do not have an impact in the final mark of the course.

At the beginning of the course the students are asked to evaluate their skills in a broad range of topics, grading themselves from 0 to 10 in the following aspects: i) management skills, ii) electronic systems design, iii) electronic programming, iv) team work experience and iv) use of English

At the end of the course, they fill-in again the same questionnaire, so that teachers can analyze how much the students have learned during the course, and gain a general impression on the benefits.

IV. Results and discussion

In this section we evaluate the advantages and drawbacks of the Agile-PBL experiences when compared to other traditional laboratory courses by means of analyzing student satisfaction. We also provide a qualitative analysis of the most common issues perceived by the teachers when applying these methodologies. Furthermore, we analyze in a quantitative and qualitative way the differences between the experience in the LSE Master course and the LSEL undergraduate course, in terms of students' marks and teacher's perception.

A. Agile-PBL courses vs pair-programming laboratories

1) Quantitative analysis: At the end of the semester, all students from all courses are asked to fill-in a questionnaire to gather their opinion on metrics related to their satisfaction on the courses they have taken. The questions are the same for all courses, and students need to grade them from "0: disagree" to "5: completely agree". Here we summarize the questions that reveal to be most important to our study:

- Q1 : The workload of the course is in accordance with the amount of credits
- Q2 : The dedication needed to pass this course is in accordance with the program
- Q3 : The assessment is in accordance to the assignments
- Q4 : I've increased my knowledge in the skills described in the course program

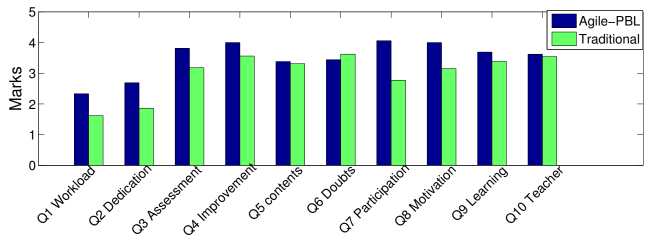

Fig. 1. Average marks in the questionnaire for the Agile-PBL undergraduate course vs an undergraduate traditional laboratory

- Q5 : The course contents are clearly explained and important topics are highlighted
- Q6 : The assistance of the professor during office hours to solve my questions has been very helpful.
- Q7 : The teacher promotes student participation and proactiveness
- Q8 : The course has motivated me
- Q9 : The professor has helped me to learn and I've increased my skills
- Q10 : The professor has fulfilled my teaching expectations

To assess student general satisfaction towards the Agile-PBL courses in a quantitative way, we compare the results obtained in this questionnaire in the LSEL Undergraduate course, with the results obtained for a traditional pair-programming electronics laboratory course of the Bachelor of Telecommunication Engineering. This traditional course is taught by the same professors than LSEL, the average number of students per course is also 10, and the contents cover similar but more basic aspects of electronic design and microcontroller programming.

Figure 1 shows the average marks obtained in the questionnaire for the students in the LSEL course when compared to the traditional course. So that the two populations are equivalent, we show the comparison only between undergraduate students (from LSEL and the traditional course), without including the Master students.

As can be seen, students belonging to the Agile-PBL undergraduate course (LSEL) give higher marks in almost all metrics, including the ones related to motivation, learning and participation. The lack of theory classes does not have an impact on the perception of students about the important topics of the course. However, the students consider that in a traditional course, their doubts are better solved during office hours.

We see that Agile-PBL methodologies outperform the metrics of student satisfaction in almost all areas. However, even though students are satisfied with the methodology, we next highlight some drawbacks spotted by the teachers.

2) Qualitative analysis: To qualitatively analyze the drawbacks of the Agile-PBL methodologies, we interview the teachers of both the Master and the Undergraduate course to know their perception. Instructors from the two courses agree on the following issues that arise when implementing agile methodologies in the classroom:

- Planning and team work organization represent an overhead to the work developed by students. For some groups it is easier to meet outside the laboratory to coordinate their tasks. For other groups, specially for Master courses, students have difficulties coordinating their work. They lose many time in planning and their organization is sometimes inefficient.
- The usage of code-sharing tools is, in general, not efficient. Students have trouble learning how to use software version control tools, do not understand the benefits and often abandon its usage.
- Accidental complexity is large and complicated to manage by instructors. By accidental complexity we understand all those impediments that are not directly correlated with the tasks assigned to students, i.e. team work issues or students not managing properly coding tools (i.e. cross-compilers, IDEs, etc.).
- Only the team leader has an overall vision of the problem, whereas the other team mates are only experts in their task. This works properly in the real-world environment, however, in the classroom, all students should learn at least the minimum important concepts of each task and understand and participate in the high-level design.

B. Master vs Undergraduate courses

The profile of students attending the master and undergraduate courses is significantly different, and this has an impact on the results of the PBL experience.

On the one hand, undergraduate students are full-time students that in most cases share classroom with their team mates in several other courses. They spend more time together in class, at the library or doing their assignments together, and thus consider themselves a group. This improves team communication and allows them to keep track on the project. They are more motivated towards innovative techniques and face Agile-PBL methodologies as a challenge.

On the other hand, Master students are generally part-time students. They spent most of their time off-campus and only meet their team mates during the laboratory sessions of other courses, or sometimes just during the weekly meetings. Communication between these teams is less fluid, and planning is worse. By investigating the time spent by the Master course teams filling-in the backlog and the burnout charts, we can observe that they tend to do these tasks in the last minute, just before the meeting. They also exhibit less proactivity and tend to demand less theory classes. The usage of software version control tools is also lower than in the Undergraduate course. Even though Master students should be more motivated towards these methodologies, because they expect to apply for a job in a very short time (some of them already work part-time), they seem less committed than the Undergraduate students. However, it is important to note that they are not less efficient than Undergraduates in completing the assigned tasks, even though their solutions have poorer quality. Regarding the individual marks assigned by teachers, the average for Undergraduate students is 8.6 on 10, whereas for Master students it decreases to 7.4 on 10.

These facts can also be observed by analyzing the marks that the students give to their mates in sprints of the Master

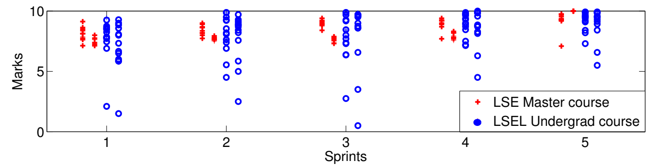

Fig. 2. Marks given by students to their team mates for each sprint, split between Master and Undergraduate courses

and Undergraduate courses (see Figure 2). The average of the marks for each sprint is similar, however, there is much more variance in the marks given by undergraduate students. As can be seen, Master students tend to always give the same mark to all students, whereas Undergraduate students are more strict and assess their fellow students with higher or lower marks depending on their commitment or the degree of task accomplishment.

C. Discussion

Generally speaking, Agile-PBL methodologies increase student satisfaction when comparing them to a traditional approach. However, instructors highlight several important issues that have to be tackled to improve the efficiency of these methodologies. Perhaps the most critical one is managing accidental complexity. To this end, we propose the development of tutorials and examples that lead the students when they learn the usage of new tools. We also believe that developing semi-automated tools to review the code written by students could help spot the main bottlenecks in the development flow and, thus, indicate instructors where to center the focus of their explanations. By reducing the accidental complexity of these methodologies, instructors will be able to focus on the real problem solving of the electronic design, thus making the learning process more efficient.

However, in the comparison between the Master and Undergraduate courses we have seen that the benefits of the methodology vary depending on the student population. Because Undergraduate students are the most proficient, we see how this issue is not correlated with the baseline knowledge. The differences between both courses are mostly related to the attitude of students: the time spent on campus and interacting with their fellow students, their commitment to the course and their motivation affect the results of Agile-PBL methodologies.

Moreover, Master students are more inclined to consider planning and Scrum an overhead, rather than a tool, and also abandon the usage of software version control tools, because they do not find a usage for that. In other words, they believe the tools provided by instructors try to solve problems that do not exist. For this particular population under study, PBL by itself does not seem to be enough. In these cases, we propose an approach based on PBL, but generating on purpose the specific problems that are found in a real-life situation in the design work flow.

That is, we propose a Project-and-Problems Based Learning (P^2BL) approach, that faces students with real-life problems that can only be solved by using the tools provided by professors. These problems might not arise during the semester, with just a dozen students, but are the reason for the development of those methodologies and tools. If students are not sufficiently motivated to embrace them with just a theoretical justification of their usage, we believe they will voluntarily adopt them when they suffer the problems they were designed to solve. This certainly constitutes an added overhead for teachers, that must design the scenarios so the problems (i.e., software conflicts, the need to recover older versions, the need to provide an accurate measure of the effort done and remaining...) arise naturally, but we expect the improvements in student progress to be worth the effort.

This new technique is being applied in the former year to the LSE Master course with the goal of improving the students results to the level of the LSEL Undergraduate course.

V. Conclusions

In this paper we have analyzed and evaluated the usage of Scrum-based agile methodologies in two Project-Based Learning courses, one belonging to a Master of Electronic Engineering course and the other belonging to a Bachelor on Telecommunication Engineering. We have seen that motivation and student satisfaction are higher when compared to a traditional course. However, there are still some open issues that negatively affect the implementation of Agile-PBL based methodologies, such as planning overhead or the accidental complexity that impedes instructors to dedicate time to real electronics design doubts and problems. We have also evaluated the differences between two different Agile-PBL courses, and showed that part-time students benefit less from these methodologies, mainly because they spend less time on campus and their commitment is lower. These students see some of the tools provided by instructors as an overhead, and are reluctant to use them. To this end, in courses where students exhibit these characteristics we propose to face them with problems that can only be solved via the usage of these tools.

Acknowledgment

Research by Marina Zapater has been partly supported by a PICATA predoctoral fellowship of the Moncloa Campus of International Excellence (UCM-UPM). This work has been partly funded by the Spanish Ministry of Economy and Competitivity under research grant TEC2012-33892.

References

[1] C. S. Consortium, "Curriculum development guidelines: New ict curricula for the 21st century: designing tomorrow's education," 2001. [Online]. Available: http://www.cedefop.europa.eu/etv/Upload/Information_resources/Bookshop/50/2204_en.pdf

[2] L. Rising and N. Janoff, "The scrum software development process for small teams," *Software, IEEE*, vol. 17, no. 4, pp. 26–32, 2000.

[3] "Agile manifesto." [Online]. Available: http://agilemanifesto.org/

[4] J. C. Stewart, C. S. DeCusatis, K. Kidder, J. R. Massi, and K. M. Anne, "Evaluating agile principles in active and cooperative learning," 2009.

[5] J. Macías-Guarasa, J. M. Montero, R. San-Segundo, Á. Araujo, and O. Nieto-Taladriz, "A project-based learning approach to design electronic systems curricula," *IEEE Transactions on Education*, vol. 49, no. 3, pp. 389–397, 2006.

[6] L. Vygotsky, E. Hanfmann, G. Vakar, and A. Kozulin, *Thought and Language*. MIT Press, 2012.

[7] V. Tinto, *Leaving College: Rethinking the Causes and Cures of Student Attrition*. University of Chicago Press, 1993.

[8] P. Reed, "An agile classroom experience." in *AGILE*, G. Melnik, P. Kruchten, and M. Poppendieck, Eds. IEEE Computer Society, 2008, pp. 478–483.

[9] Éric Germain and P. N. Robillard, "Engineering-based processes and agile methodologies for software development: a comparative case study," *Journal of Systems and Software*, vol. 75, no. 1–2, pp. 17–27, 2005.

[10] A. Martinez-Mones, E. Gomez-Sanchez, Y. A. Dimitriadis, I. M. Jorrin-Abellan, B. Rubia-Avi, and G. Vega-Gorgojo, "Multiple case studies to enhance project-based learning in a computer architecture course," *IEEE Trans. on Educ.*, vol. 48, no. 3, pp. 482–489, Aug. 2005.

[11] A. Kumar, S. Fernando, and R. Panicker, "Project-based learning in embedded systems education using an fpga platform," *Education, IEEE Transactions on*, vol. 56, no. 4, pp. 407–415, 2013.

[12] M. Frank, I. Lavy, and D. Elata, "Implementing the project-based learning approach in an academic engineering course," *International Journal of Technology and Design Education*, vol. 13, no. 3, pp. 273–288, 2003.

[13] L. Layman, T. Cornwell, and L. Williams, "Personality types, learning styles, and an agile approach to software engineering education," *SIGCSE*, vol. 38, no. 1, pp. 428–432, Mar. 2006.

[14] G. Melnik and F. Maurer, "Introducing agile methods in learning environments: Lessons learned," in *Extreme Programming and Agile Methods - XP/Agile Universe 2003*, ser. Lecture Notes in Computer Science, F. Maurer and D. Wells, Eds. Springer Berlin Heidelberg, 2003, vol. 2753, pp. 172–184.

[15] L. Layman, T. Cornwell, and L. A. Williams, "Personality types, learning styles, and an agile approach to software engineering education," in *Technical Symposium on Computer Science Education*, 2006, pp. 428–432.

[16] P. Neal, M. Ho, G. Fimbres-Weihs, F. Hussain, and Y. Cinar, "Project-based learning for first-year engineering students: Design of CO_2 sequestration," *Australasian Journal of Engineering Education*, vol. 17, no. 2, pp. 101–117, 2011.

[17] D. Dietsch, A. Podelski, J. Nam, P. M. Papadopoulos, and M. Schäf, "Monitoring student activity in collaborative software development," *CoRR*, vol. abs/1305.0787, 2013.

[18] S. Kagan and M. Kagan, "The structural approach: six keys to cooperative learning," 1994.