




# Agile Experiences in a Software Development Extension Course at a Software Engineering Bachelor's Degree

Carlos Felipe Arantes<sup>(✉)</sup> , Patrícia Lourenço Pereira, Soraia Lúcia da Silva, and Tadeu dos Reis Faria

Pontifícia Universidade Católica de Minas Gerais, Belo Horizonte, Minas Gerais, Brazil  
{carlos.arantes, patricia.pereira}@sga.pucminas.br,  
{soraialu, tadeurf}@pucminas.br

**Abstract.** This paper presents the results found in the application and adaptation of an agile approach to software development in a course called “Interdisciplinary Software Project IV” at the bachelor’s degree in Software Engineering from the Pontifical Catholic University of Minas Gerais over two years with twenty projects already done. In the fourth semester of the course, students have to develop a software with a social focus. They are introduced to real customers from the communities with real problems and need to incorporate useful knowledge of the current semester subjects into their system, with artifacts and documentation. Through the use of the Scrum framework, it was possible to observe several adaptations made to suit the academic context, although the practical reality of the course, and the nature of the clients for which the projects are developed. Throughout the paper are presented the results obtained and which can serve as an example for similar approaches.

**Keywords:** Software Engineering · Agile methodology · Interdisciplinary projects · Scrum · Education · Community-driven projects · Extension

## 1 Introduction

In the Bachelor of Software Engineering (SE) at the Pontifical Catholic University of Minas Gerais (PUC Minas) there is a compulsory course from the first to the sixth semester known as “Interdisciplinary Software Project” (ISP) in which the student must develop a fully functional software following all the development stages, from requirements elicitation to implementation and even maintenance, should a correction be required. For each semester, the current courses are incorporated, enhancing the bachelor with an interdisciplinary character that allows them to practice various concepts given inside the classroom combined with a serious practice with the introduction of clients and real needs. In this way, teachers can guide students through their journeys inside the course by introducing good market practices and raising the status of their work in addressing today’s social problems.

Specifically, the present paper brings the experiences and results of the fourth period ISP (ISP IV) in which 20 software projects have already been developed over two years of application of the course. They are free to choose the technology that they want to work, considering it is in agreement with the requisites. Such languages that have been chosen are: JavaScript, PHP, Java, C #, Python and frameworks such as Express, Zend, Spring and .NET among others; but with the differential of dealing with the chancellor course for the practice of the extension, in which the product realized must solve a problem with a social character.

According to the National Extension Policy [3], one of the advances that deserve attention concerns the institutionalization of extension. In this sense, PUC Minas's extension policy [7] addresses excitingly the importance of extension practices performed by the University, aiming at the development and formation of a society that somehow has barriers in various contexts such as social exclusion, difficulty in accessing knowledge and cultural marginalization. These support initiatives will, in some way, add to the university extension as a whole, improving the quality and management of activities and allowing students to deal with current societal problems generating professional and especially human growth.

In addition to the subjects included in ISP IV and the extension nature, students should develop their respective systems through agile methodology, applying Scrum and adapting it accordingly. Throughout the article, the applied method and the results of the agile approach in the practical academic environment are described.

## 2 Related Works

Souza, Oliveira, Grillo and Cico [10] bring that by allowing students the possibility to put into practice the techniques studied, given the development of their ability to divide problems, prioritize and establish chronograms; they have been given a huge gain in technical skills, especially in management.

Souza and Pinto [11] bring in their report that the results of the application of an agile methodology were very positive, and a huge development can be perceived in the students throughout the experience. They specifically report that by the simplicity of the agile method, students quickly understood how to work, divide and deliver functionality but that team success correlates with the agile culture present in those involved.

Billa and Cira [12] propose the use of the PBL approach, known as Problem Based Learning for Software Engineering Teaching. In this approach, students solve real problems by applying the concepts of SE, strongly connected to the same approach used with ISP IV.

## 3 Theoretical Background

It is emphasized in advance that, as it has an extension course too inside the course, a greater work is done on the importance of this type of projects, but at the same time, it is not neglected the application and teaching of processes and methodologies, requirements gathering and development, good software engineering practices and project management. In this section, the frameworks that guided and continue to guide the course as a whole are reported.

### 3.1 Extension Practices

The university extension practice is one of the ways to develop an academic formation that integrates theory and practice in order to establish a dialogical relationship between systematized knowledge and society, enabling knowledge exchange between both. This is mainly because, in extension, it is expected that there will be an academic gain for the institution, that is, the actions developed will be internalized as methodological tests for community work, knowledge of real society problems, professional experience for teachers and students, development of procedures and technical standards; in short, a two-way street for everyone involved.

The actions of this extension practice that is presented broaden the classroom space, allowing the exchange of knowledge within and outside the academic environment and also contributes to the renewal of pedagogical processes through exchange and participation between internal and external communities to the university, which strengthens the University Extension Policy of PUC Minas [7]. The activities carried out in this work contribute to the PUC Minas Institutional Development Plan [8] in enabling students to articulate the theory of the current semester with the vivid experience, allowing also a direct contact with society contributing to a more humane and citizen formation; and are part of the Pedagogical Project of the Software Engineering course. Working with extension practices allows the teachers involved to rethink their activities (action-reflection - action), improving their knowledge and methodologies.

Regarding the students, it can be said that society has been demonstrating the need for a higher educated professional who has a more complete education, not only technical but also ethical, humanistic and cultural, who can work with other areas in multidisciplinary teams. Thus, the participation of students in these practices contributes to them acting in society with competence, responsibility and justice, contributing to the construction of a prosperous, solidary and fair country.

### 3.2 Agile Methodologies and Scrum

The ISP subject, in line with the latest practices adopted in the market, emerges students into contact with agile methodologies. The Agile Manifesto, the artifact that defines these methodologies, makes it clear that the valorization of individuals and interactions is prioritized over the use of processes and tools, which makes the main objective to build software over its documentation. To achieve this, customer collaboration throughout the development becomes more necessary than contract negotiation, and the team must be better prepared to respond to changes than following a plan [1]. Agile methods tend to refer to source code as the sole documentation artifact [4], with the aim of solving the problem of creating high quality software being built in a timely manner in the face of constantly changing requirements in the business environment, bringing the need for the development team to be adaptive and able to cope with changing requirements at any stage of development [5].

The most commonly used agile method among students is SCRUM, a structural framework based on empirical process control theories, which has been widely used to manage the development of complex products since the early 1990s [9]. It employs an iterative and incremental approach to improve predictability and risk control.

The widely used SCRUM artifacts and events are Product Backlog, Sprint Backlog and Sprint [2], the latter having a fixed time period, which can be from two to four weeks. For the subject in question, teachers guide students through two-week Sprints, which consist of Sprint Planning, Daily Scrum Meetings, development work, Sprint Review, and Sprint Retrospective. Within this range, the software increment is built and delivered by students, starting a new Sprint immediately after the previous one.

The SCRUM roles are Product Owner (PO), Scrum Master (SM) and Dev Team (DT), which constitutes the “Time Scrum” [9]. The PO is responsible for maximizing product value and managing Product Backlog. The DT is staffed by professionals who do the job of delivering a potentially reliably incremental “ready-made” product at the end of each Sprint, which is required by the Sprint Review. SM is responsible for constantly promoting framework knowledge by helping everyone understand the theory, practices, rules and values of the methodology, helping to maximize the value created by DT.

Through the concepts introduced by the method, teams should be multifunctional and self-organizing, bringing flexibility, creativity and productivity to the team, as well as making the team not dependent on others outside the DT or dependent solely on what is needed to develop the project and make decisions.

### 3.3 Interdisciplinary Software Project IV

Interdisciplinary Software Project IV (ISP IV) - course of the Department of Software Engineering and Information Systems at PUC Minas - was approved as the extension course of the Software Engineering bachelor’s degree at PUC Minas. Its function is to articulate the contents of the other courses of the fourth semester around the development of social-minded application software. The subject of software development should meet the perceptions of social need diagnosed by the Extension Coordination department, coordinator of the bachelor, or the demands of developing programs and applications from University extension projects. In this case, 34 h are computed in the valorization of the extension in the course [6].

This course is fundamental for the student to practice the extension activities and to understand the humanistic foundations aimed by PUC Minas and dealt with in the pedagogical project of the course. The objective is to promote the transformative interaction of the course with other sectors of society, constituting an indispensable component for the formation of students and teacher qualification. The extension actions of this course also consider the inseparability with Teaching and Research. In the interface with the research, the methodology incorporates investigative practice. In articulation with teaching the process happens through the mediating action between theory and practice.

Also, it integrates, through practical bias, knowledge of other courses of the period and several others taught in previous periods, thus promoting their interdisciplinary content. It contributes to the formation of the egress profile, as it enables the student to develop skills and abilities to assess real needs or problems with socially-oriented software in the form of observing the actions/work routine of the community through interviews (dialogued interaction) with the society, potential users and stakeholders.

The goal for the teacher is to identify potential problem situations in the community that can be solved through a software solution and to enable students to assess needs or

requirements for a socially based computational solution in the form of observing the routine actions/work of a community. The goal for the student is to develop the ability to dialogue with the client and users to gather requirements and user profile identification, besides the development of the solution itself. The goal for the community is to identify with the teacher possible problem situations and provide students with details of the problem that will serve as the basis for defining the requirements of social application software that can solve a specific community problem.

## 4 Methodology

Following are the steps of the work methodology used throughout the semesters to date in the course: (1) Identify potential communities, non-profit organizations and projects for society given the existence of problems that can be solved via software product. For example, these communities may be existing NGOs that have well-defined social objectives and need to record their actions through some software; (2) Prepare students to conduct requirements gathering in the form of observation and documentation of the community's work routine and dialogued interview with their stakeholders. At this time, the teacher prepares students for a sympathetic perception of the reality of the community by embracing their cause to provide their contributions. The character of humanistic formation of this course is fundamental here. This software will be made by people for people. The technological component of the software is secondary to the needs of organizations and society; (3) Visit the community to conduct an interview with potential users of the software, also identifying the user profile (comfort level with technology, ease/difficulty in handling applications on the computer). The purpose of the interview is to raise the software requirements and the needs of this community in front of informatization. The person responsible for the entity will also be invited to attend the classroom for a brief presentation of the organization and its demands; (4) Describe/specify the requirements raised by creating the so-called software specification documents; (5) Validate the document with the community or a prototype. Where also demonstrated and explained the relationship with Scrum. This stage is already a proposal for intervention in the community routine and needs community validation. (6) Design and implement the software described in the Software Requirements Specification Document that has been validated by the community; (7) Present the software to the community for evaluation, and correction of possible defects and minor adjustments. At this moment, the intervention proposal is implemented. If the community accepts the intervention, it will use the software; (8) Perform usability testing with real users to identify improvement points for the software. It is important to note that these activities are developed in an interdisciplinary manner in the fourth period.

With the introduction of artifacts in Scrum, the Sprints were defined with an average of fifteen days each, and students are given greater attention - and as a consequence directly reflected in the grade obtained in the course - regarding the application of: (1) the methodology and its artifacts; (2) the participation of the customer or their Product Owner throughout the process; (3) the well-executed division of tasks. It is because of the nature of the course that students are given a greater attention to good software engineering practices and therefore a considerable part of the project development is closely

connected to the techniques and to a correct and validated documentation approach. - following agile principles - of the software.

Weekly students meet with one or more teachers of the course to explain the difficulties encountered and are guided by these teachers on how to correct the deviations. In addition, the teachers choose the backlog of each Sprint, given the scope of each system individually. For each item prioritized for Sprint, the acceptance criteria set by the teachers and the clients are validated. All monitoring of the project is carried out through project management and control tools by both the teachers and the students. It is important emphasize that the follow-up is constant. Incomplete, failed or nonexistent use of agile methodology artifacts is considered as a review point to be worked with students throughout all the reserved classes for the follow-up.

## 5 Results

The course uses software development cycle planning mechanisms that contribute to: (1) the enrichment of teamwork skills, fostering the distribution of responsibilities and collaboration, whether with clients or teammates, based on software models, both through written and oral communication; (2) work in a team; (3) lead teams and deal with people from different realities; (4) allow students to self-evaluate and evaluate their pairs and develop characteristics favorable to interpersonal relationships; (5) enables the decision-making exercise on the appropriate selection of technologies for the software solution and the development of creative and innovative solutions, since it generates a relationship of trust with the customer; (6) It also gives students the opportunity to play different roles, as they are, at different times, developers, consultants, moderators, reviewers, researchers, instructors, technical leaders and project managers. Behold, the student himself stimulates the development of his autonomy, driving the creation of business-focused planning, through an application chosen by the group, always focusing on the research of market demands and with a more humane look in the proposal of solutions considering social realities.

Students are assessed and scored according to the documents and functionalities produced: (1) Software Specification Document; (2) Socially oriented solution; Software usability testing report by its end-users; (3) Reports proposed by PROEX, responsible for extension zeal at PUC Minas, through the Extension Course Management tool. And, at the end of the course, students complete an extension practice assessment form consisting of eleven questions and can choose from the following answers: strongly disagree, partially disagree, undecided, partially agree, totally agree, or not applicable. Approximately 86% totally agreed that the practice allowed the students to value the exchange of knowledge between the University and other sectors of society. About 71% fully agreed that it was possible to build new knowledge from the challenges presented in practice; 70% totally agreed that acting in practice made possible the integration of models, concepts and methodologies from various areas of knowledge. Approximately 79% fully agreed that it was possible to exercise ethical posture and respect for diversity. About 71% fully agreed that the work carried out made it possible to verify the social relevance of the profession, to be attentive to social, regional or local development. Over 70% fully agreed that they learned to articulate theory with lived practice, developing professional skills and competences.

It is also noteworthy that throughout the semesters, there was a need for teachers to take the roles of Product Owner and Scrum Master, given the distance from real customers and the possible impediments caused by it. For this, teachers need to become better acquainted with each client's domains, the condition they are in and the needs of each one; they become a bridge to resolve conflicts and bottlenecks with clients. But even so, the students also incorporate some of this role of PO, given that due to the volume of groups it does not allow very specific attention to a particular group, being an unnecessary control, but which makes room for the student to develop autonomy and initiative. It was also noted that the students had a high acceptance with the introduction of technologies for managing their tasks such as Trello, Jira, Asana and Kanban among others allowing them to deal with project management at an early beginning.

At the end of the work, the application is tested and evaluated by the community, which is also invited to participate in the final presentation of the project together with the bachelor's faculty. Almost 100% of the partners have approved the developed systems, including the importance of the tool in achieving the daily activities of each organization and in the transformative power that information technology plays in the current century making the students trained by the course aware of the responsibility they carry in building the future.

## References

1. Beck, K., et al.: Manifesto for agile software development. *Agil. Alliance 2009* (2001). 2006
2. CIENT, Interinstitucional Editor et al.: Difficulties in the Adoption and Use of Scrum Method in Brazilian Companies. *As Dificuldades na Adoção e Uso de Método Scrum em Empresas Brasileiras Utilizando Processos Plan-Driven: Estudo de Caso Múltiplo*, vol. 8, pp. 66–79 (2017)
3. Extension Forum of rectors of the Brazilian public universities. *National University Extension Policy*, Manaus, Brasil (2012)
4. Fraga, B.S., Barbosa, M.W.: Requirements engineering in agile methods: a systematic literature review. In: *XIII Brazilian Symposium on Information Systems*, June 2017, pp. 309–315 (2017)
5. De Lucia, A., Qusef, A.: Requirements engineering in agile software development. *J. Emerg. Technol. Web Intell.* **2**(3), 212–221 (2010)
6. Pontifícia Universidade Católica de Minas Gerais: *Pedagogical project of the Computer Engineering degree* (2017)
7. Pontifícia Universidade Católica de Minas Gerais: *University Extension Policy of PUC Minas. PUC Minas/Pró-Reitoria de Extensão*, Belo Horizonte (2006)
8. Pontifícia Universidade Católica de Minas Gerais: *Institutional Development Plan: 2012 to 2016*. PUC Minas, Belo Horizonte (2011)
9. Schwaber, K., Sutherland, J.: *Scrum Guide: The Definitive Guide to Scrum* (2017)
10. Souza, S., Oliveira, B., Grillo, F., Cico, C.: *Building digital platforms while teaching Software Engineering: an experience report*. IX Software Engineering Education Forum. Fees (2016)
11. Souza, S., Pinto, V.: *Building social applications while teaching software engineering: an experience report*. Third Congress of Graduation from São Paulo University. USP (2017)
12. Billa, C.Z., Cera, M.C.: *Using problem solving to approximate theory and practice in software engineering*. V Software Engineering Education Forum. Fees (2012)