

Chapter 14

A Project–Based Introduction to Agile Software Development

Marc Lainez
Agilar, Belgium

Yves Deville
Université Catholique de Louvain, Belgium

Adrien Dessy
Université Catholique de Louvain, Belgium

Cyrille Dejemeppe
Université Catholique de Louvain, Belgium

Jean-Baptiste Mairy
Université Catholique de Louvain, Belgium

Sascha Van Cauwelaert
Université Catholique de Louvain, Belgium

ABSTRACT

This chapter shows how a lightweight Agile process has been used to introduce Agile project development to young computer science students. This experience has been conducted on a project aimed at developing Android applications. The context, the process, and the results of this experiment are described in this chapter.

1. INTRODUCTION

Agile programming is an increasingly used paradigm for the development of software. Hence, in a university curriculum of computer science, it is essential that students have the opportunity to experience the Agile paradigm. At the Louvain School of Engineering (Université catholique de Louvain, Belgium), we decided to introduce Agile software development to students through a project, during the third year of their five year

curriculum. In this project, the students had to develop their own Android application.

In this chapter, we describe this experiment and report the results. The primary objective of the experiment was to assess if an Agile software development approach was suitable for young students and if the learning outcomes could be met by means of a totally project based approach. We describe a lightweight Agile process that can be used by students to experience Agile software development as well as the various tools that have been introduced to support the development of their project. At the end of the semester, all the teams came up with a functional mobile application; 13 out of the 15 apps were of high quality

DOI: 10.4018/978-1-4666-5800-4.ch014

and were distributed on the Google Play store as open-source apps.

This chapter is structured as follows. Section 2 presents the context of Louvain School of Engineering and provides a background on Agile software development. The process of this teaching experiment is described in Section 3: hypotheses (3.1), design of the process (3.2), how it has been pretested on a small set of teaching assistants (3.3), how it has been implemented for the students (3.4), and the results and evaluation of this experience (3.5). We propose directions for future research in Section 4 before the conclusion (Section 5).

2. BACKGROUND

2.1 The Context

The Louvain School of Engineering (Université catholique de Louvain, Belgium) introduced in 2012-2013 a one-semester project for its 300 third-year students of the Bachelor Degree. The students had to choose the project related to their major option. About 60 students picked the project in computer science for 2012-2013. The project was held in parallel with other courses and accounted for about 15% of the student workload of the semester.

A crucial pedagogical choice was made: the project should be a practical introduction to Agile software development. The objective was to let students develop their own Android application, in groups of four, using a lightweight Agile process. This lightweight process allowed us to focus on the key aspects of the Agile paradigm. Furthermore, we wanted to make the code available under an open-source license and to possibly release the applications to the Google Play Store by the end of the project. Besides technical skills, this project also aimed at developing cross-disciplinary skills such as modeling, teamwork, planning, management and communication.

Active learning, and more specifically problem-based learning (Boud & Feletti, 1998), is a long-standing practice at the Louvain School of Engineering. Active learning can be defined as anything course-related that all students in a class session are called upon to do other than simply watching, listening and taking notes (Felder & Brent, 2009). Essential elements in active learning are student activity and involvement in the learning process (Prince, 2004). It comes in multiple forms, among them: collaborative learning, where students work together in small groups towards a common goal, and problem-based learning, where relevant problems are introduced at the beginning of the instruction cycle and used to provide the context and motivation (Boud & Feletti, 1998; Johnson & Johnson, 1999; Prince, 2004). It may be noted that problem-based learning is often collaborative. The choice of an Agile software development approach fits very well with this commitment to active learning. Indeed, it promotes strong interactions and collaboration between students within a group and provides opportunity for rich interactions with the teaching team.

The challenge was also to introduce students to a new way of conducting projects. As a matter of fact, students tend to wait until the last days to work on result-focused projects (in contrast to process-focused projects). The development is thus done under extreme time constraints. As you may guess, this sometimes leads to disastrous results. The goal was to make the students realize that a lightweight Agile process can be used to plan and allocate the workload, giving significantly better results with very little overhead.

2.2 Agile Software Development

Commonly called Agile, this movement groups several approaches to software development and project management. It focuses on very high collaboration within the team and strong involvement of the customer in the development process by building cross-functional and self-organized

16 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/a-project-based-introduction-to-agile-software-development/102333

Related Content

Hubble's Expanding Universe: a Model for Quality in Technology Infused engineering and Technology Education

Judith Parker (2016). *International Journal of Quality Assurance in Engineering and Technology Education* (pp. 16-29).

www.irma-international.org/article/hubbles-expanding-universe-a-model-for-quality-in-technology-infused-engineering-and-technology-education/168589

Developing Engineering Students' Communication and Information Retrieval Skills Utilizing Capstone Projects

Aaron S. Blicblau and Jamal Naser (2015). *International Journal of Quality Assurance in Engineering and Technology Education* (pp. 1-20).

www.irma-international.org/article/developing-engineering-students-communication-and-information-retrieval-skills-utilizing-capstone-projects/147414

Designing Animated Simulations and Web-Based Assessments to Improve Electrical Engineering Education

Douglas L. Holton and Amit Verma (2010). *Web-Based Engineering Education: Critical Design and Effective Tools* (pp. 77-95).

www.irma-international.org/chapter/designing-animated-simulations-web-based/44729

Educational Technoethics Applied to Career Guidance

Manjit Singh Sidhu and Pilar Alejandra Cortés Pascual (2010). *Technology-Assisted Problem Solving for Engineering Education: Interactive Multimedia Applications* (pp. 175-188).

www.irma-international.org/chapter/educational-technoethics-applied-career-guidance/37892

Impacts of School Administration Autonomy Support on Students' Learning Motivation and Intentions to Drop out of Vocational School

Bui Thi Thuy Hang, Amrita Kaur and Arun Patil (2015). *International Journal of Quality Assurance in Engineering and Technology Education* (pp. 1-12).

www.irma-international.org/article/impacts-of-school-administration-autonomy-support-on-students-learning-motivation-and-intentions-to-drop-out-of-vocational-school/134873