

Chapter 50

Experiences in Software Engineering Education: Using Scrum, Agile Coaching, and Virtual Reality

Ezequiel Scott

ISISTAN (UNICEN-CONICET) Research Institute, Argentina

Guillermo Rodríguez

ISISTAN (UNICEN-CONICET) Research Institute, Argentina

Álvaro Soria

ISISTAN (UNICEN-CONICET) Research Institute, Argentina

Marcelo Campo

ISISTAN (UNICEN-CONICET) Research Institute, Argentina

ABSTRACT

Software Engineering courses aim to train students to succeed in meeting the challenges within competitive and ever-changing professional contexts. Thus, undergraduate courses require continual revision and updating so as to cater for the demands of the software industry and guarantee academic quality. In this context, Scrum results in both a suitable and a flexible framework to train students in the implementation of professional software engineering practices. However, current approaches fail to provide guidance and assistance in applying Scrum, or a platform to address limitations in time, scope, and facilities within university premises. In this chapter, the authors present a software engineering training model based on the integration of the Agile Coach role and a virtual-reality platform called Virtual Scrum. The findings highlight the benefits of integrating this innovative model in a capstone course. Not only does this approach strengthen the acquisition of current software engineering practices but also opens new possibilities in the design of training courses.

DOI: 10.4018/978-1-5225-3923-0.ch050

INTRODUCTION

The increasing complexity of the software industry, constant changes in system requirements, distributed environments and professional mobility are ordinary challenges to software developers, who are required to be competent and proactive by rapidly expanding software industries. In order to prepare undergraduate students for ongoing success in this context, Software Engineering (SE) courses must provide effective training in the application of software development practices typically implemented in large projects and organizations (Alfonso & Botia, 2005; Mahnic 2010).

In the light of the above, we structured a SE course based on Capability Maturity Model Integration (CMMI), since this model includes professional and reliable SE practices (Kulpa, 2008). In particular, we used the framework CMMI for development version 1.3 (CMMI-DEV 1.3), which aims to provide software organizations with superior project performance, client satisfaction, and quality of products and services. To support CMMI, we designed an initial version of the course based on the Rational Unified Process (RUP) (Kruchten, 2003) to develop a capstone project (Coupal & Boechler, 2005; Devedzic & Milenkovic, 2011).

Unfortunately, teaching SE to students running a software project following RUP presented several drawbacks. As it is a plan-driven development framework, RUP requires the association of project milestones with specific dates. This made students focus on meeting deadlines and delivering the agreed milestone, skipping activities in the RUP framework. Additionally, even though RUP encourages the overlapping of phases, students inevitably slid into a waterfall-like process (Osorio, 2011). Thus, it is difficult for inexperienced students to detect mistakes made in early stages until the development process reaches the final stages. Furthermore, the prescriptive nature of RUP and the definition of specific roles (i.e. project manager and business process analyst, IBM, n.d.) to perform planning and estimating activities fail to both promote actively participation of all team members, and encourage them to make commitments regarding those activities. Since SE involves people and social interaction, these aspects are cornerstone of its reality and crucial to professional training and development. As a consequence, we moved our RUP implementation of CMMI to Scrum.

The selection of Scrum is founded on the fact that it has increased in the last few years; in 2012, 57% of respondents used Scrum or its variants, and nowadays, more than 50% of the surveyed companies are utilizing Scrum as an effective path towards agility (Versionone, n.d.). First, this agile methodology concentrates on project management practices and includes monitoring and feedback activities that focus on transparency, team effort, effective time management and personal interactions. Second, Scrum facilitates the process of satisfying requirements, for while all the requirements appear in the backlog, only a limited set is fulfilled in each Sprint. Third, the retrospective meetings and continual customer contact allow early detection of impediments and rearrangements of working plans.

In line with the popularity and effectiveness of Scrum in professional contexts, some scholars have engaged in integrating Scrum in undergraduate software engineering courses. In (Alfonso & Botia, 2005) the authors compared the results of teaching Scrum practices in an undergraduate software course to those obtained in an experience using a waterfall-like rigid process. From the experience, the authors founded that the use of Scrum resulted in lower risks and process overhead (Diaz, 2009). Chuan-Hoo Tan et al. introduced Extreme Programming (XP), Scrum and Feature-Driven Development (FDD) to initiate future software professionals in the importance of agility, flexibility and adaptability in professional contexts. The research consisted in training students in developing and delivering large-scale systems

25 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/experiences-in-software-engineering-education/192922

Related Content

Technology Transfer Projects in the UK: An Analysis of University-Industry Collaboration

Martin George Wynn (2020). *Disruptive Technology: Concepts, Methodologies, Tools, and Applications* (pp. 383-405).

www.irma-international.org/chapter/technology-transfer-projects-in-the-uk/231196

Soil Cation Exchange Capacity Predicted by Learning From Multiple Modelling: Forming Multiple Models Run by SVM to Learn From ANN and Its Hybrid With Firefly Algorithm

Rahman Khatibi, Mohammad Ali Ghorbani, Rasoul Janiand Moslem Servati (2018). *Handbook of Research on Predictive Modeling and Optimization Methods in Science and Engineering* (pp. 465-480).

www.irma-international.org/chapter/soil-cation-exchange-capacity-predicted-by-learning-from-multiple-modelling/206762

Electronic Cooling

Shankara Murthy H. M., Niranjana Raiand Ramakrishna N. Hegde (2023). *Energy Systems Design for Low-Power Computing* (pp. 100-122).

www.irma-international.org/chapter/electronic-cooling/319991

Formalization Studies in Functional Size Measurement

Baris Özkanand Onur Demirörs (2012). *Computer Engineering: Concepts, Methodologies, Tools and Applications* (pp. 1758-1778).

www.irma-international.org/chapter/formalization-studies-functional-size-measurement/62543

Artificial Neural Network Modelling of Sequencing Batch Reactor Performance

Eldon R. Rene, Sung Joo Kim, Dae Hee Lee, Woo Bong Je, Mirian Estefanía Lópezand Hung Suck Park (2012). *Handbook of Research on Computational Science and Engineering: Theory and Practice* (pp. 456-479).

www.irma-international.org/chapter/artificial-neural-network-modelling-sequencing/60371