Measuring the Impact of Agile Coaching on Students' Performance

Guillermo Rodríguez, Álvaro Soria and Marcelo Campo

Email addresses: {guillermo.rodriguez, alvaro.soria, marcelo.campo}@isistan.unicen.edu.ar

ISISTAN Research Institute (CONICET-UNICEN), Campus Universitario, Paraje Arroyo Seco, Tandil, Buenos Aires, Argentina

Journal: *IEEE Transactions on Education* (Impact Factor: 0.84), **01/2016**; *DOI:* 10.1109/TE.2015.2506624

Agile methods have been in the cutting-edge of software engineering as a means to improve management of software development processes. The widespread use of such methods in professional contexts has encouraged their integration into software engineering training and undergraduate courses. Although there are several research works that have focused on teaching Scrum through simulating a software development project, they have covered only the learning of practices within a Scrum team, and a few of them have tackled non-technical skills beyond the development practices. Thus, we claim that it is necessary to help students understand Scrum rules, clear project obstacles and tackle non-technical issues such as management and teamwork.

In this context, this work states the following research question: "How does agile coaching influence students' performance?". To answer this question, an original training model based on Scrum enhanced then with agile coaching to maximize student's performance is introduced. This model has been assessed by a case-study using a capstone project in the Software Engineering Workshop course within the Systems Engineering BSc program at the Faculty of Exact Sciences (Universidad Nacional del Centro de la Provincia de Buenos Aires). Prior to the course, students were introduced into the basis of software system design, object-oriented programming, operating systems and networks, and database management. The course aims to engage students to develop a capstone project, which is organized and prioritized according to the Product Backlog divided into three 4-weeks Sprints (students are expected to work for about 2 hours a day). The phases of the training model are Initial Phase (performed only once at the beginning), Sprint Planning Phase, Development Phase, Sprint Review and Retrospective Phase, and Product Delivery and Assessment Phase (performed only once at the end).

A comparison between a course version with and without coaching was conducted, in which the students' performance and their perceptions of using Scrum for the first time were evaluated. The students' performance was measured by the coverage of a set of recommended software engineering practices widely performed along a software development life-cycle, whereas the students' perception was evaluated throughout a survey to assess the impact of agile coaching on the training model.

The encouraging results showed that the training model allowed coached students to improve both their technical and non-technical skills. The Agile Coach allowed students to increase the coverage of the software engineering practices (approximately 22%) and enhance comprehension of Scrum; coached teams could both tackle more sophisticated project features and cover more software engineering practices than non-coached teams. Also, the survey confirmed that the course provided coached students with valuable non-technical skills, such as facilitated internalization of Scrum, faster solutions of impediments, and guidance by means of check-point meetings. Moreover, the results showed that coached students seem to have acquired highly necessary skills such as teamwork, communication and management.