Chapter 10
Controlled Experiments as Means to Teach Soft Skills in Software Engineering

Marco Kuhrmann
Technische Universität München, Germany

Henning Femmer
Technische Universität München, Germany

Jonas Eckhardt
Technische Universität München, Germany

ABSTRACT

The job profile of a Software Engineer not only includes so-called "hard-skills" (e.g. specifying, programming, or building architectures) but also "soft skills" like awareness of team effects and similar human factors. These skills are typically hard to teach in classrooms, and current education, hence, mostly focuses on hard rather than soft skills. Yet, since software development is becoming more and more spread across different sites in a globally distributed manner, the importance of soft skills increases rapidly. However, there are only a few practical guides to teach such tacit knowledge to Software Engineering students. In this chapter, the authors describe an approach that combines theoretical lectures, practical experiments, and discussion sessions to fill this gap. They describe the processes of creating, planning, executing, and evaluating these sessions, so that soft skill topics can be taught in a university course. The authors present two example implementations of the approach. The first implementation lets students experience and reflect on group dynamics and team-internal effects in a project situation. The second implementation enables students to understand the challenges of a distributed software development setting. With this knowledge, the authors critically discuss the contribution of experimentation to university teaching.

DOI: 10.4018/978-1-4666-5800-4.ch010

INTRODUCTION

Software Engineering (SE) aims at developing software-intensive systems in a systematic, methodically sound, and economic manner to master the key challenges of time constraints, budget adherence, quality and functionality. Beyond the technical questions that cover topics such as programming or testing, SE also addresses topics that deal with the organization and the management of software projects. Although both perspectives are important, today's SE education is often focused on the technical topics, as those are easier to teach (Kuhrmann et al., 2013), e.g. it is easier to teach coding and evaluating a program than to teach performing successfully in a project.

As a consequence, the current education contains a conflict between the curricula, the students' self-perception and the reality that students face when leaving the university. Especially partners from industry note that the students are usually not ready for work, even if they master a couple of programming languages. One of our partners told us: "I need to qualify a graduate for 2 or 3 extra months to make him fit." Besides company-specific knowledge (which a university is not able to teach) most partners from industry complain about missing soft skills, a missing understanding of how organizations and projects work, and missing skills regarding teamwork (Kuhrmann, 2012). In consequence, while we have a curriculum strongly addressing the technical topics, we still have gaps in appropriately educating the students on the organization and management level. We must enable students to experience that soft skills play an important role besides deep technical understanding.

Academic Curricula and Soft Skills

When teaching the required soft skills, one of the main challenges is to fit the required setting into the academic context and its constraints. For instance, since grades are given to rate the individual success, students are often forced to be "lone fighters" and teachers set up their courses aiming at clearly separated individual performance instead of team effort. And in situations where students work in teams, we, as supervisors, often observe critical situations that are not caused by technical aspects, but by communication or behavioral issues, and we often see that students struggle in fixing such a situation. However, this is not surprising, as students often have no chance to learn about the "right" communication and interaction patterns in teams, or psychology basics that enable them to realize, analyze and solve a conflict in a team setting.

To underpin and explain the clash of interests, we give an example of typical teaching formats in an academic curriculum, in this case at the Technische Universität München (Table 1), and discuss their contribution to the above-mentioned skills afterwards.

As the collection of teaching formats in Table 1 shows, only two of the listed teachings formats address practical work. Of these, the guided research aims at scientific work and only the lab provides a context in which practical problems are addressed. However, lab courses are usually focused on teaching a specific practical topic, e.g. "Windows App Development," and are not focused on teaching soft skills. During the lab courses, students usually work in a team and might even experience the resulting phenomena; however, as lab courses are not focused on teaching soft skills, this experience is only a "by-product" and not made explicit. Hence, there are few opportunities for students to experience the importance of soft skills, and in these situations the focus is on software craftsmanship instead of understanding the team effects.

The current situation highlights this conflict: Practical work is dominated by theory, and soft skills cannot be learned theoretically. Most university curricula lack in providing space to appropriately teach soft skills, which would require courses in which students have to work together,