Chapter 11
Practicing Soft Skills in Software Engineering: A Project-Based Didactical Approach

Yvonne Sedelmaier
Coburg University of Applied Sciences and Arts, Germany

Dieter Landes
Coburg University of Applied Sciences and Arts, Germany

ABSTRACT

Software Engineering requires a specific profile of technical expertise combined with context-sensitive soft skills. Therefore, university education in software engineering should foster both technical knowledge and soft skills. Students should be enabled to cope with complex situations in real life by applying and combining their theoretical knowledge with team and communication competencies. In this chapter, the authors report findings from a software engineering project course. They argue that project work is a suitable approach to foster soft skills. To that end, the authors provide justification from a pedagogical point of view, setting project-based learning into relation to action-orientated didactics. As teaching goals, they focus on experiencing a complete development project from end to end, following a software process model that needs to be adapted to the specific situation, self-determined planning and acting, including the organization of the project, teamwork and team communication, and self-reflection on individual roles and contributions, and on the performance of the project team as a whole. In order to achieve these goals, the authors form teams of bachelor students, which are headed by one master student each. It turned out that a clear separation of roles is inevitable within the team, but also with respect to instructors. Self-reflection processes concerning the team roles and the individual competencies are explicitly stimulated and cumulate in individual self-reports and post-mortem analysis sessions. The authors share findings of how well the approaches have worked and outline some ideas to improve things.

DOI: 10.4018/978-1-5225-3923-0.ch011

1. INTRODUCTION

Software is a core ingredient of nearly any part of our everyday life. This software, however, needs to be developed by highly skilled individuals. Consequently, in order to acquire and exercise these skills education in software engineering plays an important role in university education. Traditionally, universities laid their main emphasis in software engineering education on technical skills, such as e.g. programming or testing skills. In recent years, however, it has become increasingly evident that non-technical, also known as soft, skills are equally important as software is developed in teams of individuals who need to interact with each other and various stakeholders such as, e.g., customers or users of their software.

Software Engineering requires a specific profile of soft skills that is closely related to technical expertise. Recently, the authors conducted a survey with junior and senior managers with respect to which skills they expect from graduates in the software engineering field. Respondents of the survey always emphasized the importance of soft skills (Sedelmaier, Claren, & Landes, 2013).

Undoubtedly, software development requires profound technical knowledge (Sommerville, 2011). But evidently, this is not the only thing that matters. Rather, various soft skills are also needed, e.g. the ability to work in a team of hundreds of members spread around the world or the ability to communicate with various other players in the project. All interviewees want software engineers to analyze and understand complex situations and use a creative and solution-orientated approach. Several other researchers arrived at similar results and emphasize the importance of non-technical skills (Lu, Lo, & Lin, 2011; Richardson, Reid, Seidman, Pattinson, & Delaney, 2011; Rivera-Ibarra, Rodríguez-Jacobo, & Serrano-Vargas, 2010).

Although soft skills obviously are important, our survey showed that students tend to overestimate their capabilities with respect to both technical and non-technical competencies. Soft skills are core competencies of a software engineer and for this reason soft skills should be a core part of software engineering education at universities. These issues are quite difficult to teach because there are no clear cause-effect-relationships between the didactical approach and the learning outcomes. So it is difficult for teachers in software engineering to find out which didactical approach works best. Furthermore, instructors are generally not communication experts themselves and often have neither pedagogical nor didactical education background. Many things they do are not grounded on pedagogical expertise.

Thus, preparing students for the real life in software engineering is a serious challenge in university education. One approach to bring complexity and problem awareness into university education is to use project work. Project work fosters many soft skills such as communication skills and the ability to work together in a team. Interpersonal skills cannot be trained without other people around, and project work combines these competencies with the context in which they are needed. Furthermore, project work could offer students opportunities to understand inter-relationships between technical knowledge and soft skills.

Many software engineering projects fail due to at least one of the following reasons: scheduling, specifications and/or average manufacturing costs (Button & Sharrock, 1996). Button & Sharrock (1996) also state that software engineers tend to distinguish between two basic types of problems: “First, those that are due to deficiencies in the state of general engineering practice, and second, those that arise from the state of the project they were engaged in. Engineering work on any particular development thus does not involve only the resolution of the problems arising from the specific circumstances of the project itself, but also contends with problems that are recognized as generic problems of engineering work per se” (Button & Sharrock, 1996). Students hardly believe these facts. In their opinion they would do much better and lead the project to success if they were the actors. Project work in a university context