

Four Roles of Instructor in Software Engineering Projects

Stan Kurkovsky
Department of Computer Science
Central Connecticut State University
New Britain, CT, USA

kurkovskysta@ccsu.edu

ABSTRACT

As a practice-oriented discipline, Software Engineering (SE) is most effectively taught by using a variety of hands-on experiences. Team-based projects where students can practice their technical and soft skills are a key feature of many software engineering courses aimed to prepare students for the realities of industrial software development.

Current research indicates that projects offered in SE courses should have a close connection to the real world, be large scale and complex. On the one hand, students should be able to relate to the requirements and understand them without spending much time acquiring domain knowledge. On the other hand, students should be exposed to changing and evolving requirements, which is one of the main challenges of real-world software projects. Software projects should be large enough and have enough features to engage all members of the student team while enabling them to practice some degree of specialization and experience the advantages and challenges of teamwork. Software projects should be rich enough to allow students practice all previously acquired SE concepts, bring together knowledge from other specialized CS disciplines, and apply a number of modern tools and technologies.

One of the most effective ways to learn SE is to expose students to an environment that resembles an industrial setting as much as possible. As a part of an academic program, SE courses can achieve this objective by offering students a balanced combination of three components: *teamwork* involving up-to-date *technology* grounded in a solid *theoretical foundation*. Instructors in SE courses are uniquely positioned to help maintain this balance by simultaneously playing four roles described below.

As a *customer* who contracts a student team to implement a particular project, the instructor may provide a verbal or written description of the project. It is the students' responsibility to conduct the requirements elicitation interviews and to prepare formal requirements specification for subsequent validation with the customer and possible adjustment. The instructor may act as a customer during the acceptance testing throughout the project to make sure that the team is on the right track at every iteration of the project. The instructor acting as a customer can also act as a proxy if the project has been offered by an industrial partner, or some other department or organization on campus.

As an *architect* initially specifying the overall structure of the solution, the instructor can help students choose the right software architecture. This is a very important choice that has to be made early because it impacts the overall flow of the project. Later architectural changes can be extremely costly and can significantly jeopardize meeting the deadlines. Most likely, students do not have experience in making such architectural decisions and, therefore, it is crucial for the instructor to provide some guidance in this important process.

As a *team manager* concerned with risk management, the instructor needs to pay attention that the team efforts are not jeopardized due to a potential lack of students' soft skills and that these skills are built up as the project progresses. Although it is the students' responsibility to create all artifacts emerging from the project, the instructor can help the team adhere to the schedule by providing feedback based on the inspection of all produced artifacts. In this respect, the roles of the team manager and the customer may somewhat overlap because the same person will be providing feedback to the team from the technical point of view (artifact inspection), as well as from the customer point of view (verification of product implementation at each iteration).

In addition to the traditional role of giving lectures, the instructor needs to be a *mentor* offering guidance to each team and individual students addressing their specific needs and answering questions unique to their projects and the responsibilities in the context of the project. Furthermore, the instructor is uniquely positioned to advise individual students on how to resolve potential conflicts within a team. Such an advice given at the right time could make a difference between completing the project on time and allowing interpersonal tensions completely ruin any chance of collaboration between the team members.

Categories and Subject Descriptors

D.2.0 [Software Engineering]: General.

K.3.2 [Computers and Education]: Computer and Information Science Education – *computer science education*.

General Terms

Management, Human Factors.

Keywords

Software engineering project, role of instructor.