Chapter IV
Tasks in Software Engineering Education:
The Case of a Human Aspects of Software Engineering Course

Orit Hazzan
Technion - IIT, Israel

Jim Tomayko
Carnegie Mellon University, USA

ABSTRACT

The field of software engineering is multifaceted. Accordingly, students must be educated to cope with different kinds of tasks and questions. This chapter describes a collection of tasks that aim at improving students' skills in different ways. We illustrate our ideas by describing a course about human aspects of software engineering. The course objective is to increase learners' awareness with respect to problems, dilemmas, ethical questions, and other human-related situations that students may face in the software engineering world. We attempt to achieve this goal by posing different kinds of questions and tasks to the learners, which aim at enhancing their abstract thinking and expanding their analysis perspectives. The chapter is based on our experience teaching the course at Carnegie-Mellon University and at the Technion – Israel Institute of Technology.

INTRODUCTION

The complexity of software development environments is well known. This complexity includes technical aspects (such as IDEs and programming languages), cognitive aspects (for example, program comprehension) and social aspects of the profession (e.g., issues related to teamwork). As a result of this multifaceted nature, the discipline of software engineering requires that special attention be given to tasks executed by software engineering students.

This chapter presents a collection of tasks that can be integrated into software engineering education. The tasks presented here do not address software development activities (such as design or coding) but rather belong to peripheral topics related to the actual development of software. We suggest, however, that the discussion of these topics, when supported by students' engaging in a variety of tasks, has a direct influence on students' professional skills in general, and on their software development performance in particular.

We illustrate our ideas using a course on human aspects of software engineering. The course objective is to increase software engineering students' awareness of (a) the richness and complexity of various facets of the human aspect of software engineering and (b) problems, dilemmas, questions and conflicts that may arise with respect to human aspects of software engineering during the course of software development. The course is based on Tomayko and Hazzan (2004), and the tasks presented can be adapted to any software engineering course.

The Human Aspects of Software Engineering course is usually attended by senior undergraduate students or graduate students who already have some software development experience. Being an elective course, it is usually taught in a relatively small class setting. Indeed, as illustrated later on in the chapter, these course characteristics enable us to propose an interactive, hands-on and active teaching and learning style.

The importance attributed to active learning is based on the constructivist approach. Constructivism is a cognitive theory that examines the nature of learning processes. According to this approach, learners construct new knowledge by rearranging and refining their existing knowledge (cf. Davis, Maher and Nodding, 1990; Smith, diSessa and Roschelle, 1993). More specifically, the constructivism approach suggests that new knowledge is constructed *gradually,* based on the learner's existing mental structures and on feedback that the learner receives from the learning environments. In this process, mental structures are developed in steps, each elaborating on the preceding ones; although, there may, of course, also be regressions and "blind alleys". This construction process is closely related to the Piagetian mechanisms of assimilation and accommodation (Piaget, 1977). One way to support such gradual mental constructions is by providing learners with a suitable learning environment in which they are *active*. The working assumption is that the feedback, provided by a learning environment in which learners learn a complex concept in an active way, supports mental constructions of the learned concepts.

In this chapter, we start by presenting the course structure and then focus on the ten kinds of tasks used throughout the course. We explain the nature of each kind of tasks and how it may improve students' skills as software engineers. We conclude with some suggestions for implementing our approach in other courses.

BACKGROUND: HUMAN ASPECTS OF SOFTWARE ENGINEERING-COURSE DESCRIPTION

This section describes the different topics addressed in the course on Human Aspects of Software Engineering by highlighting their importance from the learners' perspective.

Lesson 1—The Nature of Software Engineering: This lesson aims at increasing learners' awareness that the success or failure of software development stem mainly from people-centered reasons rather than from technology-related reasons. By inviting learners to analyze different development environments, we illustrate the effects of human interaction in software development processes.

Lesson 2—Software Engineering Methods: This lesson focuses on models of several soft-