Chapter VI
Solutions to Challenges of Teaching "Systems Analysis and Design" for Undergraduate Software Engineers

Özlem Albayrak
Bilkent University, Turkey

ABSTRACT

This study is an enhancement of previous research presented at the 2nd AIS SIGSAND European Symposium on Systems Analysis and Design and its improved version presented at the 3rd National Software Engineering Symposium (UYMS) 2007. The AIS-SIGSAND 2007 study, the first phase, was part of ongoing research by which systems analysis and design-teaching experiences related to course evaluation items were enlightened. This study summarizes previous studies and introduces new findings suggested by those studies that relate to teaching challenges on systems analysis and design in software engineering. The first challenge studied is to decide a suitable evaluation item set in undergraduate level system analysis and design courses for software engineers. The second challenge relates to implicit assumptions made by software engineers during the analysis phase. Based on pre-interview, test, and post-interview data, the study presents a snapshot of an analysis in software engineering regarding implicit assumptions made by analysts. Related to these challenges, the study concludes with proposals on systems analysis and design education.

INTRODUCTION

"Software engineering education" is an important and a challenging arena that involves certain myths and human interaction (Ghezzi and Madrioli, 2005; Hawthorne and Perry, 2005; Hillburn and Watts, 2002; Morrogh, 2000; Vliet, 2005; Hazzan and Tomayko, 2005). Due to this importance,

there have been many studies conducted in this area. Several guidelines for software engineering education were prepared (Albayrak, 2003; Bagert, Hilburn, Hislop and Mengel, 1998; Thomas, Semeczko, Morarji and Mohay, 1994; Vliet, 2006). Some studies concentrated on pre-graduation challenges and studied software engineering curricula (Cifuentes and Hughes, 1994; Pullan and Oliver, 1994; Bagert 1998; Parnas, 1999; Schneider, Johnston and Joyce, 2005). Other studies were conducted to prepare software engineers for real life by suggesting industry and university collaboration (Clark, 2005; Ellis, Mead, Moreno and Seidman, 2003; Dawson and Newsham, 1997; Dawson, 2000; Yamaura and Onoma, 2002) or via software engineering projects (Aizamil, 2005; Liu, 2005; Morgan and Lear, 1994; Mohay, Morarji, Thomas, 1994; Oudshoom and Maciunas, 1994). A great deal has been written on the future of software engineering education (Boehm, 2006; Cianciarini, 2005; Bagert, et. al., 1998).

Software engineering is an integrated discipline. Systems analysis and design are two main elements of software development. For today's software engineers, understanding the problem correctly (analysis) and solving it in the best possible way (design) are very important. Thus, special emphasis must be given to teaching systems analysis and design to software engineers.

Studies on teaching systems analysis and design courses were conducted long before Hunter's research on attributes of excellent systems analysts (Hunter, 1994). System Analysis and Design (SAD) in a computer science curriculum was suggested by Spence and Grout in 1978 (Spence and Grout, 1978). Several aspects of SAD course development were studied (Golden, 1982; Goroff, 1982; McLeod, 1996; Larmour, 1997). Archer proposed a realistic approach to teaching SAD (Archer, 1985), while Olfman and Bostrom analyzed innovative teaching for SAD (Olfman and Bolstrom, 1992). Osborne proposed the use of a CASE tool for teaching systems analysis and design (Osborne, 1992), and Dick suggested the use of student interviews (Dick, 2005). During the 1990s, human factors related to SAD were investigated, and teamwork and the human factor in SAD teaching were studied (Fellers, 1993; Omland, 1999). Following the previous studies, Misic and Russo aimed to identify the importance of the educators' role in various systems development tasks, activities, and approaches and to compare educators' perceptions to those of practicing systems analysts (Misic and Russo, 1999).

Systems analysis and design are important phases in software engineering; hence, importance should be given to both of them. A software engineer should be armed with systems analysis and design related knowledge, not in a classical way but in a comprehensive way similar to that proposed in this chapter, so that software engineers are able to apply what they learn at universities to real-life, practical problems.

This study shares the experiences of preparing undergraduate software engineering students for SAD related subjects applicable to real-life, practical problems. The study is performed in three phases: The fist phase constructs the background for the AISSIGSAND paper and is mostly related to challenge of using different evaluation items to measure software engineers' success in systems analysis and design subjects. The first phase studies the challenges of applying different types of evaluation items in an SAD related undergraduate course. It can be utilized to help academicians who search for an appropriate combination of evaluation means for a course teaching SAD to undergraduate software engineering students. The second phase includes analysis related tests conducted to observe implicit assumptions embedded in analysis studies. Both the second and the third phase of the study deal with challenges related to implicit assumptions made by analysts during analysis. In the third phase of the research, experiments and pre and post interviews were conducted. The results of the second and the third phase of the experiments can be utilized by academicians who aim to avoid, or at least minimize,