The Role of Software Engineering in Undergraduate Education

J. Fernando Naveda  
jfn@cs.rit.edu  
Department of Computer Science  
Rochester Institute of Technology  
Rochester, NY 14623

Michael J. Lutz  
mjl@cs.rit.edu  
Department of Computer Science  
Rochester Institute of Technology  
Rochester, NY 14623

Thomas B. Hillburn  
hilburn@db.erau.edu  
Department of Computer Science  
Embry-Riddle Aeronautical University  
Daytona Beach, FL 32114

Linda M. Northrup  
lmn@sei.cmu.edu  
Software Engineering Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213

Michael C. Stinson  
stinson@cps.cmich.edu  
Department of Computer Science  
Central Michigan University  
Mount Peasant, MI 48859

Abstract - When first used at a NATO conference in 1969, the term “software Engineering” was more an aspiration than a fact. However, in the quarter century that has elapsed since this seminal conference, academic and industrial organizations have made great progress in software development, so much so that software engineering can now be considered a profession in its own right. Many masters-level programs have been created to address the need for more professional software engineers. A question remains, however, as to the place for software engineering in undergraduate studies.

To date, most work in undergraduate software engineering education has been performed by small clusters of computer science and engineering faculty, widely dispersed throughout academia. Building on recommendations from the Software Engineering Institute (SEI), the Association for Computing Machinery (ACM), and the IEEE Computer Society, these initial efforts have begun to bear fruit. Typically, the result is a set of software engineering courses arranged according to one of the following formats:

a) An elective or required course sequence.

b) Elective non-credit paracticums.

c) Software engineering topics introduced in existing courses.

Given industrial demands for more software development professionals, software engineering at the undergraduate level has become too important to ignore. In fact current efforts in the US and abroad are gearing towards the development of baccalaureate programs in software engineering. Although some schools will succeed in developing undergraduate software engineering programs, others will respond by continuing to provide software engineering education as part existing computer science and / or computer engineering curricula. What is more, it may be that existing curricula, which include a software engineering component, can coexist with a program specifically targeted to software engineering.

This panel will address the following issues in undergraduate software engineering:

a) What is the proper relationship between typical engineering programs and those in software engineering?

b) How should software engineering education differ from both computer science and computer engineering?

c) If undergraduate program degrees in software engineering are developed, how should such programs interact with other engineering programs?

d) How are minors and concentrations in software engineering affected by the presence of a baccalaureate in software engineering?