Panel Discussion:
Graduate Programs in Software Engineering

Moderator:

Gary Ford
Software Engineering Institute

Panel Members:

Alfs Berztiss
Computer Science Department
University of Pittsburgh

Daniel R. Bidwell
Computer Information Science Department
Andrews University

Bernice M. Folz
Dean, Department of Quantitative Methods and Computer Science
College of St. Thomas

Norman E. Gibbs
Professor and Director of Software Engineering Education
Carnegie Mellon University

Daniel Olivier
Intermetrics, Inc.

Panelists discussed the motivation for and structure of the master’s curriculum in software engineering at their respective universities (at National University in the case of Mr. Olivier). Short descriptions of these programs follow.
A software engineering program based on engineering principles

A. T. Berztiss and W. D. Hurley
Department of Computer Science
University of Pittsburgh, Pittsburgh, PA 15260

In his "silver bullet" paper Frederick Brooks suggested three fruitful approaches to dealing with the essential difficulties of software development. First, spread costs by developing easily adaptable products. Second, let software systems evolve from rudimentary prototypes. Third, produce great designers. Since great designers are hard to come by, prototypes, and the software that evolves from them, will have to be developed by many competent craftsmen rather than a few outstanding artists. The goal of our MS program in software engineering education is to develop such competence.

Our program is based on the adaptation of standard engineering practices to the development of software with special emphasis on prototyping. Although there is the major difference between conventional and software engineering in that engineers in general produce tangible objects, but software engineers produce abstractions, software engineers engage in the same activities as other engineers.

Our purpose was to develop a software engineering curriculum that emphasizes general engineering principles. To this end we first identified ten tasks that engineers perform: (1) define specifications for the objects to be constructed; (2) resolve conflicting requirements and adapt ideal specifications to real-life constraints by setting priorities; (3) construct objects from standard components, or modify standard components to suit special circumstances; (4) anticipate change and make allowances for modifications; (5) transfer theory into practice; (6) scale up, either in production or in size; (7) perform tests in order to maintain the quality of products at predetermined measurable levels, with respect to reliability, robustness, efficiency, testability, modifiability and portability, reusability, understandability, and other quality attributes; (8) determine what tools are needed for a given job; (9) apply management techniques to the previous activities; (10) engage in technological adventure when needed.

Since software engineering is concerned with large-scale software development projects, participation in group projects must be a component of education in software engineering. We emphasize information systems because we think them representative of the systems our students will have to develop in their professional lives. Systems that our groups have built include management systems for a supermarket chain, a metropolitan transit system, and a cultural institution.

The Department of Computer Science has already been offering two graduate courses in software engineering for the past three years:

CS231A Specification and Design introduces a number of specification languages, such as Larch, VDM, Z, and SF, and uses them in specification case studies. Emphasis is on prototyping, and the topics covered include software quality attributes, domain modeling, modularization criteria and the handling of scale-up and maintenance problems, the interplay of modularization and reuse throughout software development, validation of specifications and designs, and knowledge-based system techniques for maintaining specification and design information. The course also covers tools for specification and design. In the future there will no longer be a group project associated with this course. Instead, specification languages will be studied in greater depth.