Chapter I
Software Engineering Education: Past, Present, and Future

Gregory W. Hislop
Drexel University, USA

ABSTRACT

There is a strong and growing global demand for skilled software engineers. The institutions that educate software engineers are evolving and changing to meet this need. This chapter provides an overview of this effort to develop software engineering education. It discusses the historical development of software engineering education, provides some perspective on current status, and identifies some of the challenges faced by software engineering educators. The intended audience for this chapter is anyone interested in software engineering education who has not participated in the developments to the present time. The goal is to provide a summary background of how the discipline has evolved and pointers to key publications that are part of that history. Since this chapter surveys foundational topics in software engineering education, many of the topics touched on in this chapter are covered in more detail in other chapters of this volume.

INTRODUCTION

The demand for skilled software developers is growing at an extraordinary rate as software is being used in an ever widening set of domains. The increase in the use of the internet, the phenomenal rate of growth of available data, and new developments such as biosensors, grid computing, and cognitive machines require software engineers who can correctly engineer and modify these kinds of systems within budget and at a reasonable cost. As a result, educational institutions are under increasing pressure to produce educated and capable software engineers. However, educational institutions face many challenges in producing these software engineers that extend far beyond curriculum issues. Software engineering is still a discipline trying to define itself and find a place among the set of computing and engineering disciplines. As such, this chapter will address a mix of issues related to three themes:

- **Context:** The external issues that have influenced the development of software engineering education including the issue of organizational location of software engineering within a college or university, politics related to emergence of a new discipline, licensing, certification, and accreditation.
- **Community:** The collaboration, cooperation, and sharing of information among software engineering educators.
- **Curriculum:** The content and organization of degree programs and individual software engineering courses in other computing degrees.

The chapter is organized by looking at these issues historically, in the present, and for the future.

The intended audience for this chapter is anyone interested in software engineering education who has not participated in the developments to the present time. The goal is to provide a summary background of how the discipline has evolved and pointers to key publications that are part of that history.

DEVELOPMENT OF SOFTWARE ENGINEERING AS A DISCIPLINE

The problems of developing software were noticed as soon as significant software development activities began. The notion of software engineering as a solution to this problem is commonly dated to the NATO conference on this topic held in 1968 (Naur & Randell, 1969). Versions of the conference report and the report of a second conference held a year later are available at http://homepages.cs.ncl.ac.uk/brian.randell/NATO/.

This conference is noteworthy for the extent to which the range of topics currently recognized as central to software engineering were clearly identified even in this early effort. Organization of software development activities was clearly understood, at least from a waterfall model perspective. Key software engineering problems such as scale and complexity were clearly recognized, as were difficulties in estimation, and even the potential for things like construction of software from components.

A review of this material is helpful to make the point that software engineering has a core set of issues and problems that are stable over some extended period of time, and across very substantial technology changes. On the other hand, this same review is striking in indicating how modest progress has been in addressing software engineering issues decisively.

Although there is broad agreement on the need for solutions to the issues software engineering addresses, the question of whether software engineering should be a discipline has been a more divisive question. Almost 40 years after the NATO conference, computing professionals have not reached consensus on how to organize computing knowledge or the computing professions.

In academic discussions of the disciplines, the key issue for software engineering has been the relationship of software engineering to computer science. This debate has often been described using Venn diagrams to question whether the two disciplines intersect, are disjoint, or whether one is a subset of the other. A more recent set of diagrams in Computing Curricula 2005: The Overview Report (ACM & IEEE, 2005) clearly shows the disciplines as distinct but with substantial intersection.

Beyond academic circles, the separation of computing disciplines is generally ignored. The use of job titles and professional designations is almost completely ad hoc. With regard to software engineering, there is "no standard definition for this term when used in a job description. Its meaning varies widely among employers." (ACM & IEEE, 2005, p. 15) Like any computing profession label, the term is applied with no particular concern for formal education or certification of the person involved. While the