

# Chapter 5.11

## How to Create a Credible Software Engineering Bachelors Program: Navigating the Waters of Program Development

**Stephen Frezza**

*Gannon University, USA*

**Mei-Huei Tang**

*Gannon University, USA*

**Barry J. Brinkman**

*Gannon University, USA*

### ABSTRACT

This chapter presents a case study in the development of a Software Engineering (SE) Bachelor's Degree program. It outlines issues in SE program development, various means to address those issues, and explains how the issues were addressed in the initial and ongoing development of an undergraduate SE program. By using SEEK and SWEBOK as requirements sources to define what an undergraduate software engineer needs to know, the authors walk through the creation of a sample curriculum at a small, comprehensive

university in the United States. Both the current and initial curricula are presented. The article discusses many items to consider in the process of planning and launching a new BSSE program, such as accreditation, curriculum guidelines, sources of information, and potential problems.

### INTRODUCTION

Software Engineering is one of the newer engineering disciplines to emerge. Starting with the coining of the 'Software Engineering' term in 1968

(Naur, 1969), there has been continual growth in interest in software engineering education. Initially, these efforts were primarily at the graduate level, serving software engineering practitioners with undergraduate degrees in Computer Science, Computer Engineering or other related fields. In 1998, in recognition of the needs of bachelors-level computing graduates, the Computer Society of the Institute for Electrical and Electronic Engineers (IEEE-CS) and the Association for Computing Machinery (ACM) established the Joint Task Force on Computing Curricula 2001 (CC2001) to undertake a major review of curriculum guidelines for undergraduate programs in computing (Diaz-Herrera, 2004). This and other efforts (EA, 2007; CEAB, 2006; ABET, 2005) added official recognition of the need for the establishment of effective undergraduate programs preparing students to become software engineers.

The underlying assumption is that creating a new degree program for a relatively new discipline (Software Engineering), in a professional area (Computing) that already has several well-established disciplines (Computer Science, Computer Engineering, Information Systems, etc.) necessarily comes with a number of significant development risks. This chapter takes the form of an extended experience report, in the hope of presenting an overview of these risks, and practical means to mitigate them. This work is primarily based on the authors' experience in developing a software engineering undergraduate program leading to a Bachelor of Science degree in Software Engineering (BSSE) at a small comprehensive university in the United States (Frezza, 2006). Effort has been made to generalize this experience, and include questions and issues encountered in other SE program development efforts, as well as raising issues that may be more critical in other organizational settings.

## ISSUES IN SE PROGRAM DEVELOPMENT

Developing a new undergraduate program, particularly one like Software Engineering that does not have long-established definitions can be (and for us was) a delicate business. Among the key stakeholders for a new SE program, the requirements for what belongs in such a major may not be well understood, or easily communicated. In all, our program development effort was similar to many of our software development experiences, in that the requirements management activities were significant, messy, and working to resolve them early proved worthwhile. Our undergraduate software engineering program, at the time of writing, has been developed, launched, gone through several on-going outcomes reviews, and we are currently preparing our first accreditation self-assessment.

Based on our reflection on the issues we encountered, and our post-design assessments, some of the key issues we've found in developing a new SE program include:

- **Organization:** Determining where the program is housed or sponsored within the institution
- **Vision:** Defining the style, or professional focus of the program
- **Accreditation:** Applying international and national standards to ensure program quality
- **Curriculum:** Designing the academic plan for students to meet or exceed the vision, and
- **Finding help:** Locating contacts to support program development

### Organization

Determining where an SE program is housed is important to its success. The issue centers on

26 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/create-credible-software-engineering-bachelors/29489](http://www.igi-global.com/chapter/create-credible-software-engineering-bachelors/29489)

## Related Content

---

### Quality Assurance Issues for Big Data Applications in Supply Chain Management

Kamalendu Pal (2022). *Research Anthology on Agile Software, Software Development, and Testing* (pp. 1458-1483).

[www.irma-international.org/chapter/quality-assurance-issues-for-big-data-applications-in-supply-chain-management/294527](http://www.irma-international.org/chapter/quality-assurance-issues-for-big-data-applications-in-supply-chain-management/294527)

### Improving Novice Programmers' Skills through Playability and Pattern Discovery: A Descriptive Study of a Game Building Workshop

Thiago Schumacher Barcelos, Roberto Muñoz Sotoand Ismar Frango Silveira (2015). *Human Factors in Software Development and Design* (pp. 141-172).

[www.irma-international.org/chapter/improving-novice-programmers-skills-through-playability-and-pattern-discovery/117300](http://www.irma-international.org/chapter/improving-novice-programmers-skills-through-playability-and-pattern-discovery/117300)

### Software Modernization of Legacy Systems for Web Services Interoperability

Chia-Chu Chiang (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications* (pp. 380-388).

[www.irma-international.org/chapter/software-modernization-legacy-systems-web/29398](http://www.irma-international.org/chapter/software-modernization-legacy-systems-web/29398)

### Multichannel Service Delivery Architecture: A Case Study

Randall E. Duranand Anh Duc Do (2015). *Handbook of Research on Innovations in Systems and Software Engineering* (pp. 589-601).

[www.irma-international.org/chapter/multichannel-service-delivery-architecture/117942](http://www.irma-international.org/chapter/multichannel-service-delivery-architecture/117942)

### A Framework for Homogeneous Cross-Project Defect Prediction

Lipika Goel, Mayank Sharma, Sunil Kumar Khatriand D. Damodaran (2021). *International Journal of Software Innovation* (pp. 52-68).

[www.irma-international.org/article/a-framework-for-homogeneous-cross-project-defect-prediction/266282](http://www.irma-international.org/article/a-framework-for-homogeneous-cross-project-defect-prediction/266282)