

## Chapter XVI

# How to Create a Credible Software Engineering Bachelor's 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 SWEBOOK 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 faculty ownership of the program, and administrative support for the students. Many SE programs are organized in the same academic housing as Computer Science programs, but this is not universally the case. At issue is the blend of CS, IS, and Engineering courses currently

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-bachelor/29605](http://www.igi-global.com/chapter/create-credible-software-engineering-bachelor/29605)

## Related Content

---

### Using Epistemic Game Development to Teach Software Development Skills

Christos Gatzoulis, Andreas S. Andreou, Panagiotis Zaharias and Yiorgos Chrysanthou (2022). *Research Anthology on Agile Software, Software Development, and Testing* (pp. 861-883).

[www.irma-international.org/chapter/using-epistemic-game-development-to-teach-software-development-skills/294499](http://www.irma-international.org/chapter/using-epistemic-game-development-to-teach-software-development-skills/294499)

### A New Novel Hybrid Dynamic Color Segmentation Model for Road Signs in Noisy Conditions

Akram Abdel Qader (2021). *International Journal of Software Innovation* (pp. 1-22).

[www.irma-international.org/article/a-new-novel-hybrid-dynamic-color-segmentation-model-for-road-signs-in-noisy-conditions/290431](http://www.irma-international.org/article/a-new-novel-hybrid-dynamic-color-segmentation-model-for-road-signs-in-noisy-conditions/290431)

### A State-Based Intention Driven Declarative Process Model

Pinna Soffer (2013). *International Journal of Information System Modeling and Design* (pp. 44-64).

[www.irma-international.org/article/state-based-intention-driven-declarative/80244](http://www.irma-international.org/article/state-based-intention-driven-declarative/80244)

### A Simple Solution to Prevent Parameter Tampering in Web Applications

Ouzhan Menemenciolu and Ihami Muharrem Orak (2018). *Application Development and Design: Concepts, Methodologies, Tools, and Applications* (pp. 1491-1507).

[www.irma-international.org/chapter/a-simple-solution-to-prevent-parameter-tampering-in-web-applications/188266](http://www.irma-international.org/chapter/a-simple-solution-to-prevent-parameter-tampering-in-web-applications/188266)

### Lowering Coupling in Distributed Applications With Compliance and Conformance

José Carlos Martins Delgado (2020). *Applications and Approaches to Object-Oriented Software Design: Emerging Research and Opportunities* (pp. 35-77).

[www.irma-international.org/chapter/lowering-coupling-in-distributed-applications-with-compliance-and-conformance/249320](http://www.irma-international.org/chapter/lowering-coupling-in-distributed-applications-with-compliance-and-conformance/249320)