ABSTRACT

Community and culture significantly influence value orientation, perceived needs, and motivation as well as provide the ground for creating shared understanding. All disciplines have their own cultures, and all cultures evolve through cross-cultural exchanges. The computing community has created and documented a sound body of knowledge of software engineering (IEEE SWEBOK). It is one of finest examples of multi-cultural synthesis of many disciplines especially engineering, computer science, and even social sciences. With the very large scale worldwide endeavor on computing or software engineering education, it is now time to leverage education and ‘learning’ related research to create and document a theoretically sound body of knowledge of software developers’ education. Such a body of knowledge should naturally require us to synthesis the evolving disciplines of software engineering and higher education.

In this thesis, we discuss our study and investigations about the following types of questions:
1. How has software development education evolved, specifically with reference to educational research?
2. What is meant by competent and professionally oriented computing engineers, especially with respect to software engineering? What are the essential attributes? What is the relative importance of these attributes?
3. What is the degree with which the various components of traditional processes of engineering education succeed in creating opportunities for enhancing these competencies? What students think about their educational experiences? What students think works well for them? What processes do professional engineers recommend?
4. What pedagogical practices succeed in developing competencies, and under what circumstances? What comes in the way of implementing these strategies? What kinds of lectures are effective for learning in the views of students and faculty? What factors block students from effective learning? How to overcome these difficulties?
5. What kind of instructional interventions are required? How can the existing education theories/strategies/methodologies be used to educate competent computing engineers? Do we need new theories of learning for software development education? If so, what would be main aspects of such a theory of learning?
In this study, the research processes included a wide-ranging survey of published literature in diverse areas of software development, computer science and IT education, engineering education, professional and higher education, learning theories, thinking, instruction design, and human development. The research also included a study of a large number of comments written by professional software developers about contemporary issues related to software development processes, required competencies, endorsements, etc., in various professional forums. More than three hundred professionals of more than sixty organizations from various countries have been consulted and/or surveyed on various issues. More than one thousand undergraduate computing students, and more than one hundred faculty members, have also been surveyed on selected issues.

We have proposed a three-tier taxonomy of twelve competencies and a comprehensive unified framework of pedagogic engagements in software development education. We have also discussed some instructional interventions developed by us, manifesting some aspects of this framework. All these interventions were administered in a chosen set of existing computing courses. Some new courses have also been developed in the process. The development of the framework of pedagogic engagement, and these interventions for instructional reform of software development education, has been an intertwined and highly spiral process.

We hope that our proposed framework of pedagogic engagement in software development education will help the community of software development educators and researchers to create a variety of interventions that will help in extending the ‘Software Engineering Body of Knowledge’ (SWEBOK) to ‘Software Development Education Body of Knowledge’ (SDEBOK). Designers of educational programs for other professions can also adapt this framework and methodology.