

Guest Editorial

Software-intensive systems are increasing in importance due to their essential role for supporting everyday activities, as well as sustaining the global economy. Therefore, preparing software engineering students for technical design, decision making, and responsibility is critical. The evolution of software application domains toward web services, grid computing, intelligent software agents, and autonomic computing are leading to development of novel, agile, and flexible methodologies for software engineering. However, significant roadblocks exist in keeping teaching practices current in the face of rapid change and bringing the state of the art in research to software engineering education due to the steep learning curve and quagmire of formal details. Yet, a fundamental component of any engineering discipline is the use of sound techniques and processes that facilitate the production of engineering artifacts. To this end, this special issue aims to bring together the most recent advancements in strategies and tools to improve the teaching and learning of software engineering skills within the emerging landscape of next generation software development issues and challenges.

To this end, the purpose of this special issue is to facilitate dissemination of the most recent advancements and emerging trends in the theory, methodology, and toolkits of Software Engineering Education. The articles of this special issue provide discussions and evaluations of new practices and tools for educating next generation software engineers in light of the observed trends in the software enterprise. For instance, the article by Pierre N. Robillard and Mihaela Dulipovici presents an empirical study to compare the effectiveness of agile and disciplined processes. The authors examine the quality of the requirements implementation, the project total effort, the process activity effort and the product size to assess the suitability of such alternative processes to facilitate informed decision in capstone project design. Similarly, Lucas Layman, Laurie Williams, Kelli Slaten, Sarah Berenseon, and Mladen Vouk focus on bringing agile and plan-driven software development methodologies together within a single course structure through emphasis on collaboration and active learning. They discuss the changing needs and expectations of the industry and illustrate how the proposed course appeals to the specific learning needs of a diverse group of students. The findings of Rubby Casallas and Nicolas Lopez in their article that is entitled “An Environment to Help Develop Professional Software Engineering Skills for Undergraduate Students” is consistent with the earlier articles. They emphasize the role of controlled educational environments that promote active learning through self-assessment. The *QualDev* environment, which is introduced in their study, provides real world experience where students face diverse challenges regarding teamwork issues, processes, rapid technology changes, and tools that real software companies deal with on a day-to-day basis.

Recognizing the significance of Global Software Development and Open Source Software Communities in today's software engineering practice, David Carrington discusses how open source software can be used as a resource to address educational objectives, including but not limited to understanding the dynamics of software evolution. The challenges for teaching staff are also delineated to provide useful guidance in incorporating the use of open source software in software engineering courses. The premise of the article is based on the observation that community forms of software engineering organizations are becoming common and accepted style of innovative engineering practice, and hence familiarity with the dynamics underlying such user innovation communities will be essential for future software engineers. Patricia Lago, Henry Muccini, Ljerka Bens-Dukic, Ivica Crnovix, and Sasikumar Punekkat present a European graduate level Software Engineering program that aims preparation of students in global development of complex, large-scale systems. They explain the challenges pertaining to development of the curriculum along with discussion on how the program achieves the objective of educating global software engineers. Kevin A. Gary, in his article “The Software Enterprise: Practicing Best Practices in Software Engineering Education”, emphasizes the notion of best practices in comparison to rigid processes in the context of large scale projects so that contextual learning can be attained. The author points out that Software Enterprise reflects the belief that engineering know-how is acquired and applied through its contextual and repeated practice, and not through a traditional teaching and learning model. Nicholaos Petalidis points in the same direction by asserting the need for adapting to cultural shift in software engineering practice through more emphasis on iterative, and even trial-and-error strategy for discovering solutions.

Model-based software engineering is an emergent trend, where emphasis in software construction is shifting towards analysis and design. Realizing this trend, Guillermo Jiménez-Díaz, Mercedes Gómez-Albarrán, Pedro A. González-Calero present an approach to teach design patterns with the aim of promoting active learning through participatory role-play sessions or games during which students refactor designs. Realizing the potential of gaming and role-playing in engineering education, Wen-Hsiung Wu,

Wei-Fan Chen, Tsung-Li Wang, and Chuong-Ho Su develop and evaluate a Game-based Software Engineering Educational System (GBSEES) that adopts role-playing strategy to facilitate software development in a team-based environment.

In closing, we would like to express our thanks and appreciation to all the authors who submitted their manuscripts for publication in this special issue. Without their intellectual work and creative contributions, this special issue could not exist. Although the limitations of space regrettably leave some important contributions and topics, we hope this issue provides accessibility to significant technical issues involving the diverse use emerging techniques and methods in Software Engineering education. The contribution of the reviewers was critical in assuring the highest quality in the published papers. We would like to thank all of our reviewers for their inputs and recommendations to improve the technical quality of the special issue. We also would like to express our sincere thanks to the Editor-in-Chief, Professor Michael Wald, for his support and suggestions that improved the overall process and allowed editing of this issue on time, while attaining high level of quality.

Stephen H. Edwards
Computer Science, College of Engineering,
Virginia Tech, Torgersen Hall, Suite 3160A,
Blacksburg, VA, 24061, USA

Levent Yilmaz
Computer Science & Software Engineering, College of Engineering,
Auburn University, Shelby Center for Engineering Technology,
Suite 3116, Auburn, AL, 36849-5347, USA