Software Engineering Education - Present and Future

Kasi Periyasamy, David Riley, Kennt Hunt
Dept. of Comp.Sci.
Univ. of Wisconsin-La Crosse

Zaid Altahat
Dept. of Comp. Sci.
Univ. of Wisconsin-Parkside

Derek Riley
Dept. of Comp. Sci.
Milwaukee School of Engineering

Yan Shi
Dept. of Comp. Sci and Soft. Eng.
Univ. of Wisconsin-Platteville

Abstract

Rapid advancements in computing discipline requires continuous curricular changes in all areas in Computer Science. Software Engineering (SE), being one of the popular areas in Computer Science, deals with developing correct, consistent and maintainable software products using disciplined approaches to the development process. In addition to new discoveries in its own area, SE needs to be cope up with advancements in other fields. For example, advancements in software security requires considerable changes in software development process such as using risk-based life-cycle models. This panel discussion focuses on how software engineering education is expected to be modified in order to support other areas in Computer Science.

1 An overview of Software Engineering Education

Over the past 30 years, there has been a tremendous increase in Software Engineering (SE) education and equally strong job growth among SE professionals. Initially, SE was concerned with developing software products using very rigid processes such as waterfall models but it was quickly realized that other approaches such as prototyping and risk-based approaches are more appropriate than the earlier models. Currently, a lot of industries use agile methods for software development which de-emphasize pre-planning everything and heavy documentation during software development. In addition, the complexity of application domains such as distributed systems and mobile technology, along with expectations of high reliability and security have made huge impacts in using SE principles and practices. In the ACM Standard for undergraduate software engineering curriculum [3], it is stated Early approaches relied on expert interpretation of experience (structured approaches) as well as the use of models based on mathematical formalism (formal approaches). More recently, widespread practical experience and empirical studies have contributed to a better understanding of how and when these models and ideas work, how they need to be adapted to varying situations and conditions, and the importance of human interactions when defining and developing software systems. From this message, it is clear that SE curriculum needs to address the evolutions happening in software industries.

1.1 Curricular changes

Most of the curriculum changes in SE education were driven by the application of software engineering principles and techniques to other disciplines. As an example, emerging trends in software assurance, safety and reliability triggered a vast change in SE courses. Software products on mobile devices and embedded software systems, and developing software systems for healthcare applications bring in new dimensions to the curriculum requirements, particularly in practicing SE principles. The joint task force on computing curriculum by IEEE and ACM is working hard in revising the recommended SE curriculum for undergraduate degree programs [3] in order to keep it up to date so that it can be effectively applied to emerging trends such mobile applications, embedded systems and healthcare applications. Risk management, software security and reliability are now becoming core SE principles. A similar approach for a unified SE curriculum was underkane by Stevens Institute of Technology which is given in [2]. Because of the vast domain of topics, not all parts of the SE curriculum can be taught in any degree program. For instance, the SWEBOK [1] includes the topics that are expected to be covered in SE courses which range from mathematical and engineering foundations at one end to professional practice and SE economics at the other end of the spectrum. As a result, teaching institutions are burdened with choosing a reasonable subset of the curriculum that suits their student population.

2 Panel Discussion

The objectives of this panel discussion are two-fold:

1. To share our (the panelists) experiences in teaching software engineering in each of our campuses bringing out the uniqueness and achievements from each campus and then fostering our thoughts towards the future of SE education and curriculum development.

2. To address the following open-ended question:

   > What do you think about including emerging topics such as *Data Science, Big Data Analysis, Internet of Things, Information Security*, and *Mobile Computing* into the SE curriculum? The discussion will focus on the relevance of these topics to SE and the impact of expanding the horizon of SE curriculum.

The panel members have come from four different campuses in Wisconsin. Their names and biographic details are given in the Appendix.

References

[1] SWEBOK - Guide to the Software Engineering Body of Knowledge. IEEE Computer Society Press, 2004.

[2] Graduate Software Engineering (GSwE2009), Curriculum Guidelines for Graduate Degree Programs in Software Engineering. Stevens Institute of Technology, 2009.

[3] Software Engineering 2014 - Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. Joint Taksforce on Computing Curricula, IEEE/ACM, 2015.

Appendix

Presenters and their biographies:

1. Kasi Periyasamy (organizer and moderator)
   Deaprtment of Computer Science, University of Wisconsin-La Crosse

   > Dr. Kasi Periyasamy is a Professor and Program Director for the Master of Software Engineering (MSE) program in the Department of Computer Science at University of Wisconsin-La Crosse (UWL), Wisconsin since 1999. Prior to joining UWL, he was in the Department of Computer Science at University of Manitoba, Winnipeg, Canada for eight years. Dr. Periyasamy has published numerous papers in software engineering and had been the keynote speaker in some international conferences. His primary research interests include formal specifications, verification and validation, and software project management. He is the co-author of the book

Specification of Software Systems; the first edition of the book was published in 1998 and the second edition in 2011, both by Springer-Verlag. He also contributed to a chapter in a book on teaching software project management skills; the book title is *Overcoming Challenges in Software Engineering Education*, Liguo Yu (Ed.), published by IGI Global in Spring 2014.

2. David Riley
   Department of Computer Science, University of Wisconsin-La Crosse

   Dr. David Riley has been a Professor of Computer Science at the University of Wisconsin - La Crosse since 1978. Dr. Riley wrote the original proposal for the UWL Master of Software Engineering degree and has taught various software engineering courses, both required and elective, graduate and undergraduate. Recently, Dr. Riley's interests include information security and computational thinking. His latest textbook is *Computational Thinking for the Modern Problem Solver*, co-authored with Dr. Kenny Hunt.

3. Zaid ALtahat
   Deaprtment of Computer Science, University of Wisconsin-Parkside

   Dr. Zait Altahat is a Professor and the co-director of the App Factory in the Department of Computer Science at the University of Wisconsin-Parkside (UWP), Wisconsin since 2016. Prior to joining UWP, he was working in the industry for close to 20 years. He worked in the Telecom with Motorola, in the Healthcare with GE Healthcare & Baxter, and in Software Engineering research group with UniqueSoft. At these employers, Dr. Altahat worked on the prototype, design and implementation of several challenging software solutions. Dr. Altahat has publications and research in Software Engineering topics such as Aspect Oriented Modeling.

4. Kenny Hunt
   Department of Computer Science, University of Wisconsin-La Crosse

   Dr. Kenny Hunt has over twenty-five years of experience in the fields of computer science and engineering as both an educator and practitioner. His technical expertise spans a broad array of the computational spectrum; from the design of research satellite electronics to the development of large-scale cloud-based web applications. His primary research interests include image processing, software architecture, and web development. He is the author of *The Art of Image Processing with Java*, an upper-level college textbook on image processing fundamentals. He is the co-author of *Computational Thinking for the Modern Problem Solver*, an introductory text to computer science that is organized around the grand ideas expounded by the computational thinking movement.

5. Derek Riley
   Department of Computer Science, Milwaukee School of Engineering

   Dr. Derek Riley is an Associate Professor in the Electrical Engineering and Computer Science department at Milwaukee School of Engineering (MSOE). Prior to his role at MSOE he was a faculty member at the University of Wisconsin-Parkside and Middle Tennessee State University. Derek earned a PhD in computer science from Vanderbilt University in 2009 with a dissertation on high performance modeling and simulation methods. He has continued this research by extending formal model analysis methods using crowdsourcing and has numerous papers in these topics. His current research interests have extended to include teaching agile software processes and software product line maintenance.

6. Yan Shi
   Department of Computer Science and Software Engineering, University of Wisconsin-Platteville

   Dr. Yan Shi received her PhD in Computer Science from the University of Texas at Dallas in 2011. She has been an Assistant Professor in the Department of Computer Science and Software Engineering at the University of Wisconsin-Platteville since 2011 and is now the Program Coordinator Software Engineering program. She is a member of ACM, IEEE and ISCA (Internationa Society of Computers and Their Applications). She has been serving on the Board of Directors of ISCA since 2014.

   Dr. Shi's research interests focus on data engineering in wireless communications and mobile computing, software verification and smart fault localization, and machine learning. She has published her research in many reputable journals and conferences such as IEEE Transactions on Mobile Computing and INFOCOM. She serves as a peer referee in many journals and conferences such as IEEE Transactions on Parallel and Distributed Systems and IEEE Transactions on Wireless Communications. She was also the Program Chair for the International Conference on Software Engineering and Data Engineering in 2013-16.