

CEET 2017 Keynote Speakers

System Thinking for Software Engineering Educators

Richard E. (Dick) Fairley

Abstract:

Complex, computer-based systems are increasingly pervasive in all aspects of modern society, including communications, health care, transportation, commerce, civil infrastructure, and military applications. Software is a primary element of these systems; it provides the connections among the diverse elements of a modern system and the interfaces to the system's environment. In addition, software provides many features of modern systems (functionality, behavior, and quality attributes). Those of us charged with preparing the next generation of software engineers have the responsibility to prepare them for a world in which software is a critical element of complex systems. System thinking is a cognitive process that is widely applied in fields such as applied mathematics, psychology, biology, game theory, and social network analysis. It provides the foundation for systems engineering and is increasingly applicable to software engineering. This presentation will focus on systems thinking, why understanding and applying it is important for software engineers, and how systems thinking can be introduced in software engineering courses.

Dick Fairley Bio Sketch



Richard E. (Dick) Fairley is founder and principal associate of Software and Systems Engineering Associates (S2EA) and an adjunct faculty member in the doctoral program at Colorado Technical University. His professional interests are software engineering, information technology, systems engineering, and project management. Dr. Fairley was chair of the Software and Systems Engineering Committee of the IEEE Computer Society from 2010 to 2015. He was co-editor of the software engineering body of knowledge (SWEBOK V3), leader of the Computer Society team that produced the software engineering competency model (SWECOM), leader of the joint Computer Society-PMI team that developed the software extension to the PMBOK Guide (SWX), and was an author of the systems engineering body of knowledge (SEBoK). He is currently a SEBoK editor and an active member of INCOSE – the International Council on Systems Engineering.

Dr. Fairley's PhD is in computer science from the University of California, Los Angeles (UCLA). His bachelors and masters degrees are in electrical engineering. Before obtaining his PhD, Dr. Fairley worked in industry as an engineer and computer programmer. Dr. Fairley has held tenure at three universities but left each to pursue other opportunities. Dr. Fairley is a member of IEEE, the IEEE Computer Society, INCOSE, and PMI – the Project Management Institute. He and his wife Mary Jane live in the Colorado Mountains where he enjoys motorcycling, skiing, and jazz.

Teaching Agile Methods

Mark Paulk

Abstract:

Agile methods are increasingly being used in industry and studied in academia, but relatively little teaching of how to do agile right occurs at the undergraduate or graduate student level. Although many student teams may claim to be using an agile method on their project, an objective assessment is likely to note violations of fundamental agile practices. For example, I observed MSE student projects spanning multiple terms where, for the first year, no code was being written... yet one of the agile principles is that working software is the fundamental measure of progress. Applying what was learned in class on the project did not align with how agile works in this context. Many students find agile practices such as peer programming and collective code ownership (from Extreme Programming) personally uncomfortable. And changing the requirements for the project midway is a violation of academic norms! Exposing students to the agile paradigm can be a challenging experience for both the teacher and the student.

Mark Paulk Bio Sketch



Dr. Mark Paulk teaches software engineering at the University of Texas at Dallas and is a consultant and author in software engineering, software process improvement, high maturity practices, agile methods, and statistical thinking.

Dr. Paulk was a Senior Systems Scientist at the Institute for Software Research at Carnegie Mellon University from 2002 to 2012, co-authoring the eSourcing Capability Model for Service Providers. From 1987 to 2002, Dr. Paulk was with the Software Engineering Institute at Carnegie Mellon, where he led the work on the Capability Maturity Model for Software.

Dr. Paulk received his PhD in industrial engineering from the University of Pittsburgh, his MS in computer science from Vanderbilt University, and his BS in mathematics and computer science from the University of Alabama in Huntsville. He is a Fellow of the ASQ, a Certified ScrumMaster, and a Senior Member of the IEEE.