

# Advancing Software Engineering Professional Education

Mark Ardis, Stevens Institute of Technology

Pierre Bourque, École de Technologie Supérieure

Thomas Hilburn, Embry-Riddle Aeronautical University

Kahina Lasfer, Stevens Institute of Technology

Scott Lucero, US Department of Defense

James McDonald, Monmouth University

Art Pyster, Stevens Institute of Technology

Mary Shaw, Carnegie Mellon University

// A reference curriculum for master's programs helps ensure software engineers possess the appropriate skills, knowledge, and experience to develop, maintain, and acquire complex systems. //

**THE LABELS SOFTWARE engineer** and *professional* abound in discussions concerning the development and use of computer systems and software products. Over the past 20 years, academics and practitioners have devoted much effort to enhancing and advancing the state of professional software engineering (SE) practice. In 1989, the Software Engineering Institute (SEI) at Carnegie Mellon University published a landmark report on graduate education in SE;<sup>1</sup> several universities have used its guidelines to establish graduate SE degree programs. But software development has changed dramatically in the intervening years. Software's scale, complexity, and criticality have mushroomed, yet no significant effort has been made to revisit and update the original SEI guidelines.

Figure 1 shows the evolution of SE education. Many cite the 1968 North Atlantic Treaty Organization Conference as the beginning of comprehensive discussion about the content and nature of SE.<sup>2</sup> The first graduate programs followed advice collected at a 1976 workshop and reports from the IEEE Computer Society (IEEE CS) Subcommittee on Model Curricula in Software Engineering.<sup>3,4</sup> In 1993, the Association for Computing Machinery (ACM) and the IEEE CS formed a joint steering committee to advance SE professionalization. In 1998, this was reconstituted as the Software Engineering Coordinating Committee (SWECC), which sponsored the development of a body of knowledge<sup>5</sup> and a code of ethics and professional conduct for software engineers.<sup>6</sup> In addition, SWECC sponsored educational activities that led to Accreditation Board of Engineering Technology (ABET) SE criteria and an undergraduate curriculum model.<sup>7,8</sup> We realize that Figure 1 presents a some-



what US-centric focus, but many of the activities it depicts were international in scope. We also recognize that significant contributions to the advancement of SE education occurred throughout the world, particularly in the development and accreditation of software educational programs in Canada, Mexico, Europe, Australia, Asia, and South America.

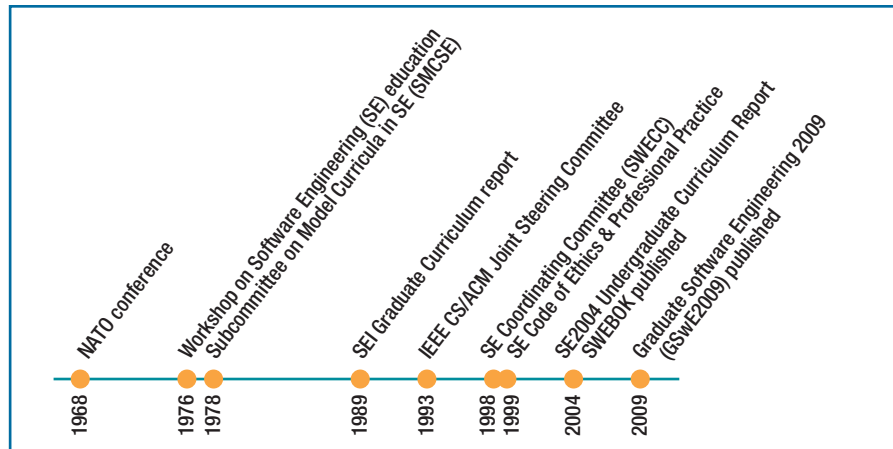
“Graduate Software Engineering 2009 (GSWE2009): Curriculum Guidelines for Graduate Degree Programs in Software Engineering” is a recently developed set of curriculum guidelines for master’s programs.<sup>9</sup> A group of SE educators and practitioners developed GSWE2009 as part of the Integrated Software & Systems Engineering (IS-SEC) curriculum project at Stevens Institute of Technology. An underlying focus of GSWE2009 is how to advance the state of SE practice and support a better understanding and agreement about the nature of professional software engineers.

This article focuses on the guiding principles for GSWE2009’s development, student prerequisites, the core body of knowledge (CBOK), and an architectural framework that supports flexible curriculum implementation. Download the full GSWE2009 document, with more detailed information, at [www.gswe2009.org](http://www.gswe2009.org).

## Guiding Principles

The GSWE2009 project team established a set of guiding principles to provide a foundation for project activities. The GSWE2009 document presents 17 guiding principles that act as the requirements. Three principles illustrate their nature:

- **Principle 1.** The principal purpose of GSWE2009 will be to provide a set of tailorable recommendations for developing and improving curricula that provide software engineering education at the master’s



**FIGURE 1.** Timeline showing the evolution of software engineering education. The figure identifies the key events, groups, and documents over the last 30 years that have influenced the acceptance and advancement of software engineering (SE) education.

degree level. It is not intended to be the basis for accreditation.

- **Principle 7.** The rapid evolution and professional nature of software engineering require ongoing review and revision of the corresponding curriculum.
- **Principle 8.** GSWE2009 will be sensitive to changes in technologies, practices, and applications, new developments in pedagogy, and the importance of lifelong learning.

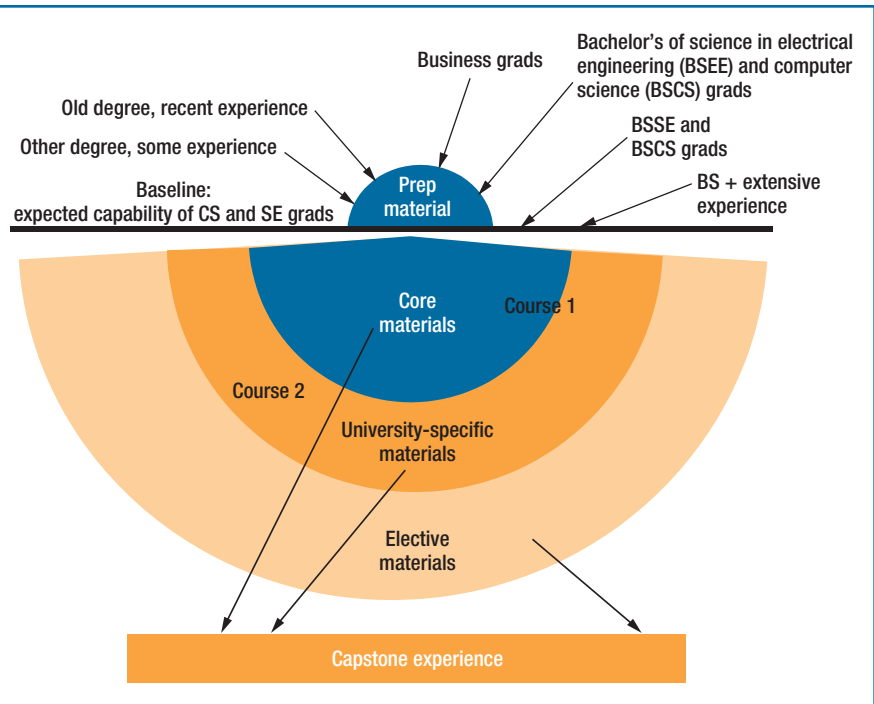
In support of Principle 7, representatives from the ACM and IEEE CS education boards acted as observers and communication links to their respective groups. In fall 2009, both organizations agreed to long-term sponsorship of GSWE2009, signifying that the curriculum guidelines met all their expectations regarding the quality of the development process and the product itself and that they take responsibility for maintaining the currency of the guidelines.

## Student Outcomes

Although the guiding principles influenced all aspects of the project, they particularly helped shape the GSWE2009 student outcomes:

- **CBOK.** Master the core body of knowledge.

- **Domain.** Master SE in one application domain, such as finance, medical, transportation, or telecommunications, and in one application type, such as real-time, embedded, safety-critical, or highly distributed systems.
- **Depth.** Master at least one knowledge area or subarea from the CBOK to the Bloom Synthesis level.<sup>10</sup>
- **Ethics.** Be able to make ethical professional decisions and practice ethical professional behavior.
- **Sys eng.** Understand the relationship between SE and systems engineering (sys eng) and apply sys eng principles and practices to SE.
- **Team.** Be an effective team member on both multinational and geographically distributed teams; effectively communicate both orally and in writing; and lead one area of project development, such as project management, requirements analysis, architecture, construction, or quality assurance.
- **Reconcile.** Be able to reconcile conflicting project objectives and find acceptable compromises within limitations of cost, time, knowledge, risk, existing systems, and organizations.
- **Perspective.** Understand and appreciate feasibility analysis, negotiation, and good stakeholder



**FIGURE 2.** Architectural structure of a Graduate Software Engineering 2009 (GSWE2009) master's program showing key components and how they're related. The architectural components include preparation expectations, core materials, university-specific materials, electives, and a capstone experience component.

- The equivalent of an introductory course in SE.
- At least two years of practical experience in some aspect of SE or software development. This experience should include team participation, development of a successfully delivered program or component, and an update or repair to an existing program or component.

Note that these baseline expectations aren't admission requirements, which are set by individual universities and programs. However, deviations from these expectations could require lengthening the program to achieve the outcomes. A student can compensate for the lack of a formal education by more extensive experience; a university can offer a student lacking certain knowledge or skills an opportunity to take additional leveling courses; or a student lacking experience can take an internship or follow some other means to gain that experience while in the degree program.

### Architecture and CBOK

The architectural model in Figure 2 depicts the structure of the GSWE2009 curriculum. It identifies, via the CBOK, the minimal material that all programs should include and makes provisions for each institution to develop its own programs. The curriculum architecture is similar to the one Mark Ardis and Gary Ford proposed<sup>1</sup> and is compatible with the existing master's programs for which Arthur Pyster and his colleagues describe the course and curriculum data.<sup>11</sup> It's intended to provide a structural basis for programs that deliver the GSWE2009 outcomes.

The curriculum architecture includes preparatory material, core materials, university-specific materials, elective materials, and a capstone experience. The heavy black line in Figure 2 represents the baseline preparatory knowledge for students in

communication in a typical software development environment, and perform those tasks well. Have effective work habits and be a leader.

- *Learn.* Be able to learn new models, techniques, and technologies as they emerge and appreciate the necessity of such continuing professional development.
- *Tech.* Be able to analyze a current significant software technology, articulate its strengths and weaknesses, compare it to alternative technologies, and specify and promote improvements or extensions to that technology.

Notice that the outcomes strike a balance between technical and nontechnical skills. While "CBOK," "domain," "depth," "sys eng," and "tech" are primarily technical in nature, "ethics," "team," "reconcile," "perspective," and "learn" emphasize nontechnical capa-

bilities and would be important to any modern engineering discipline.

### Entry Prerequisites

In concert with developing student outcomes, the project team determined a baseline of what educators should expect about student capability upon entry to a GSWE program to achieve these outcomes. Consequently, the GSWE2009 Graduate Software Engineering Reference Curriculum (GSWERC) presumes that an entering student meets the following prerequisites:

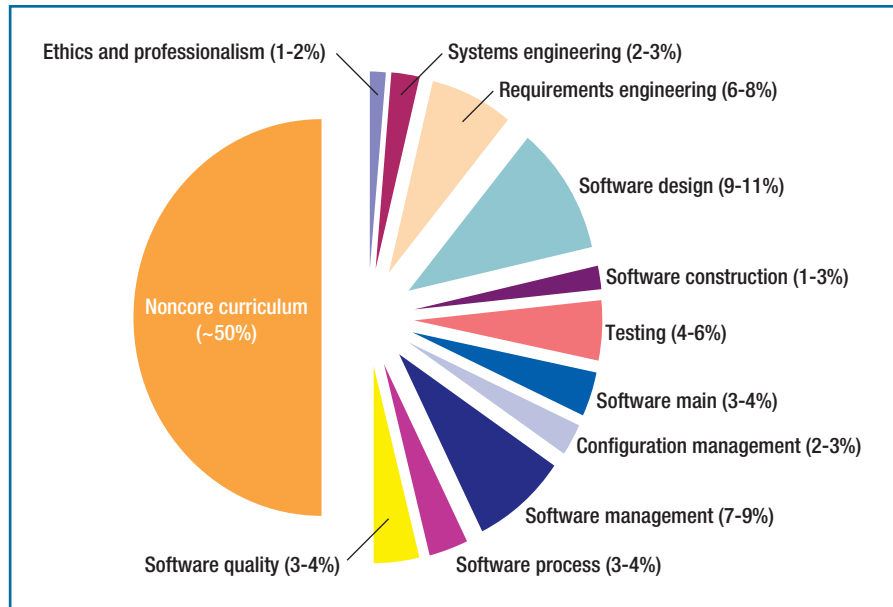
- The equivalent of an undergraduate degree in computing or an undergraduate degree in an engineering or scientific field with a minor in computing. The expected knowledge from such a degree, specified in the GSWE2009 document, consists of mathematics and computing fundamentals.

a GSwE2009 master's program. For example, a student could achieve the preparation through an undergraduate computing or engineering degree (bachelor's of science in computer engineering, computer science, electrical engineering, or SE), plus two years of software development experience. Students master the material below the heavy black line after achieving the baseline preparation. Individual programs will determine how to prepare students whose background falls short; the "prep material" element above the expectations line represents this additional preparation.

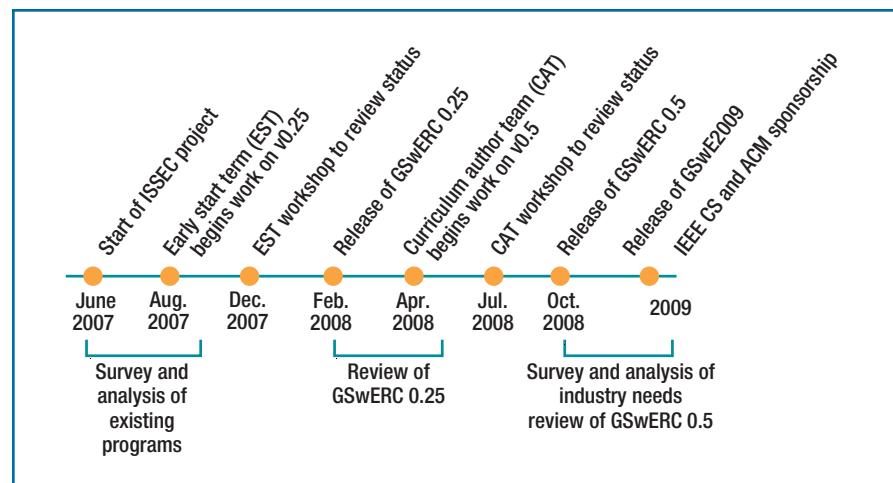
GSwE2009 strongly recommends that students demonstrate their accumulated skills and knowledge in a capstone experience, which might be a project, a practicum, or a thesis. Students completing the curriculum must understand and appreciate the importance of negotiation, effective work habits, leadership, and good communication with stakeholders in a typical software development environment.

The content primarily consists of the CBOK and its extensions, which are strongly related to the outcomes "CBOK," "domain," "depth," "sys eng," "ethics," and "reconcile." Figure 3 depicts the organization of the CBOK, designating the percentage of curriculum content for each core area. Notice that the CBOK occupies approximately 50 percent of the curriculum, allowing flexibility and specialization in curriculum design and permitting its extension to support outcomes domain and depth.

The primary source for developing the CBOK was the SE body of knowledge (SWEBOK).<sup>5</sup> The project team also derived knowledge elements from Software Engineering 2004<sup>8</sup> and other sources.<sup>12-14</sup> In the study and analysis of these sources, the project team decided that although the SWEBOK organization and content would dominate, they needed to make changes in



**FIGURE 3.** Core body of knowledge (CBOK) organization showing the percentage of curriculum content for a Graduate Software Engineering 2009 (GSWE2009) master's program. Note that CBOK occupies approximately 50 percent of the curriculum, allowing flexibility and specialization in curriculum design.



**FIGURE 4.** Graduate Software Engineering 2009 (GSWE2009) timeline listing the project periods and dates devoted to surveying and analysis, guideline development, and version releases and reviews.

various areas and topics to support the GSwE2009's expected student outcomes and accommodate the needs and views of academia, industry, and the professional computing societies. Thus, GSwE2009 includes two knowledge areas that aren't in the current version of the SWEBOK: systems engineering fundamentals, and ethics and professional conduct.

## GSWE2009 Development

The GSwE2009 project team formed in summer 2007 and worked for two years on the curriculum guidelines. Figure 4 shows a timeline for GSwE2009 development.

Through teleconferences and face-to-face meetings, the team pursued project activities and deliverables. The team began by studying various source docu-

ments and existing master's of software engineering (MSE) programs.<sup>11</sup> Next, it developed project guiding principles, and then, using these principles, established the outcomes that GSwE graduates must achieve. Concurrent to and interacting with the development of the student outcomes, the team determined the background a student must have to succeed in a GSwE program. Subsequently, the guidelines, outcomes, and expected student background proved to be the principal source for guidance and control of GSwE work.


The preparatory knowledge for GSwE2009 was a foundation element in the development of the CBOK—the key component in the GSwE2009 curriculum's architecture. The process of establishing the CBOK and the architecture was interactive: the project team initiated and refined these two crucial GSwE components through a set of versions that the project subteams shared, reviewed, discussed, modified, and then finalized into a full, approved internal version.

Next, more than 100 reviewers from academia, government, and industry participated in an external review of the various GSwE releases listed in Figure 4. The project team assembled, organized, and assigned reviewer comments to the appropriate subteam for disposition, which it then resolved in a public and transparent manner. Subteam recommendations for addressing review comments were presented and discussed at a full team project meeting. The team then revised the initial version and prepared the final GSwE2009 document.

**T**he GSwE2009 document is the initial step in advancing professional SE by enhancing graduate SE education programs. For GSwE2009 to succeed, the model must be available, understood by the targeted academic and industrial com-

munities, viewed as a key reference for SE curriculum development, and actually used in the development and modification of software-assurance-focused curricula. The sponsorship status of GSwE2009 with the ACM and the IEEE CS is a critical step in advancing these goals.

The project has initiated a study of existing MSE programs to determine how closely they align with GSwE2009.<sup>15</sup> Most of these programs have curricula that are consistent with almost all the GSwE2009 recommendations. We have anecdotal evidence that some programs are modifying their curricula to more fully agree with GSwE2009. We're also conducting a survey to determine whether the software industry's needs are best met by students who accomplish the GSwE2009 outcomes.

We encourage stakeholders in professional SE education (managers, practitioners, and educators) to become involved in this effort: download the full document; read and study it; share your views with your colleagues; and promote or lead an effort to create a new program or enhance an existing one using GSwE2009 guidance. 

### Acknowledgments

GSwE2009 is the product of 43 authors from more than 24 organizations who selflessly came together to improve global software engineering graduate education. The comments and suggestions of over 100 external reviewers significantly improved the quality of the final product. We thank Kristen Baldwin, Bruce Amato, Scott Lucero, and others in the US Office of the Secretary of Defense for their involvement, leadership, and financial support. We also thank the ACM, the Brazilian Computer Society, IEEE CS, INCOSE (International Council on Systems Engineering), and the NDIA (National Defense Industrial Association) Systems Engineering Division.

### References

1. M. Ardis and G. Ford, *SEI Report on Graduate Software Engineering Education*, CMU/SEI 89-TR-21, Software Eng. Inst., Carnegie Mellon Univ., 1989.

2. *Software Engineering: Report of a Conference Sponsored by the NATO Science Committee*, P. Naur and B. Randell, eds., NATO, 1968.
3. P. Freeman, A.I. Wasserman, and R.E. Fairley, "Essential Elements of Software Engineering Education," *Proc. 2nd Int'l Conf. Software Eng.* (ICSE 76), IEEE Press, 1976, pp. 116–122.
4. P. Freeman and A.I. Wasserman, "A Proposed Curriculum for Software Engineering Education," *Proc. 3rd Int'l Conf. Software Eng.* (ICSE 78), IEEE Press, 1978, pp. 56–62.
5. *Guide to the Software Engineering Body of Knowledge (SWEBOOK)*, A. Abran et al., eds, IEEE CS Press; [www.computer.org/portal/web/swebok/htmlformat](http://www.computer.org/portal/web/swebok/htmlformat).
6. *Software Engineering Code of Ethics and Professional Practice*, v. 5.2, ACM/IEEE-CS Joint Task Force on Software Engineering Ethics and Professional Practices, ACM Press, 1999; [www.acm.org/serving/se/code.htm](http://www.acm.org/serving/se/code.htm).
7. Criteria for Accrediting Engineering Programs (Effective for Evaluations during the 2010–2011 Accreditation Cycle), *Accreditation Board for Engineering and Technology*, 2009; [www.abet.org/forms.shtml](http://www.abet.org/forms.shtml).
8. *Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*, ACM/IEEE CS Joint Task Force on Computing Curricula, ACM Press, 2004; [www.acm.org/education/curricula.html](http://www.acm.org/education/curricula.html).
9. *Graduate Software Eng. 2009 (GSwE2009): Curriculum Guidelines for Graduate Degree Programs in Software Eng.*, Integrated Software and Systems Eng. Curriculum Project, Stevens Inst. Technology, A. Pyster, ed., 2009; [www.gswE2009.org](http://www.gswE2009.org).
10. *Taxonomy of Educational Objectives: The Classification of Educational Goals, Handbook 1: Cognitive Domain*, B.S. Bloom, ed., Longmans, 1956.
11. A. Pyster, et al., "Master's Degrees in Software Engineering: An Analysis of 28 University Programs," *IEEE Software*, vol. 26, no. 5, 2009, pp. 94–101.
12. *INCOSE Systems Engineering Handbook*, v. 3.1, C. Haskins, ed., INCOSE-TP-2003-002-03.1, INCOSE, 2007.
13. *Software Engineering for the 21st Century: A Basis for Rethinking the Curriculum*, tech. report CMU-ISRI-05-108, M. Shaw, ed., Inst. for Software Research, Carnegie Mellon Univ., 2005.
14. S. Tockey, *Return on Software: Maximizing the Return on Your Software Investment*, Addison-Wesley, 2004.
15. *Comparisons of GSwE2009 to Current Master's Programs in Software Engineering*, D. Frailey, M. Ardis, and N. Hutchison, eds., Integrated Software & Systems Engineering Curriculum Project, Stevens Inst., 2009.



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.



**MARK ARDIS** is a distinguished service professor in the School of Systems and Enterprises at Stevens Institute of Technology. His interests include the professionalization of software engineering, especially

through teaching and technology transfer. Ardis has a PhD in computer science from the University of Maryland. He's a member of the ACM and the IEEE CS. Contact him at [mark.ardis@stevens.edu](mailto:mark.ardis@stevens.edu).



**SCOTT LUCERO** is the deputy director of software engineering in the Office of the Director, Defense Research and Engineering, in the US Department of Defense. His research interests include expanding the discipline of systems engineering to rapidly developed, complex, adaptable systems. Lucero has an MS in computer science from the George Washington University. Contact him at [scott.lucero@osd.mil](mailto:scott.lucero@osd.mil).

through teaching and technology transfer. Ardis has a PhD in computer science from the University of Maryland. He's a member of the ACM and the IEEE CS. Contact him at [mark.ardis@stevens.edu](mailto:mark.ardis@stevens.edu).



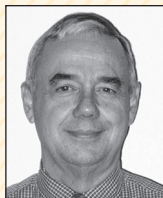
**PIERRE BOURQUE** is an associate professor and the director of the MSE Program at the Université du Québec's École de Technologie Supérieure. His research interests include the maturation of the software engineering

discipline and profession, software measurement, software project estimation, and the fundamental principles of software engineering. Bourque has a PhD in software engineering from the University of Ulster. He's a coeditor of the *Guide to the Software Engineering Body of Knowledge* (SWEBOK). Contact him at [pierre.bourque@etsmtl.ca](mailto:pierre.bourque@etsmtl.ca).



**JAMES MCDONALD** is an associate professor of software engineering at Monmouth University. His research interests include software project and organizational management, software verification, validation and

maintenance, and engineering education. McDonald has a PhD in electrical engineering from New York University. He's a senior life member of IEEE, and a member of the IEEE CS, the ACM, and the ASEE. Contact him at [jamesmc@monmouth.edu](mailto:jamesmc@monmouth.edu).



**THOMAS HILBURN** is a professor emeritus of software engineering at Embry-Riddle Aeronautical University. His research interests include software processes, object-oriented development, and software

engineering education. Hilburn has a PhD in mathematics from Louisiana Tech University. He's a member of the ACM and the IEEE CS and currently chairs the Curriculum Committee for the IEEE CS Educational Activities Board. Contact him at [hilburn@erau.edu](mailto:hilburn@erau.edu).



**ART PYSTER** is a distinguished research professor and deputy executive director of the systems engineering research center at the Stevens Institute of Technology. His primary technical interests include

the integration of systems and software engineering and research on how to more rapidly and effectively develop the systems engineering and software engineering workforce. Pyster has a PhD in computer and information sciences from Ohio State University. He's a senior member of the IEEE CS and a fellow and a member of the board of directors of the International Council on Systems Engineering. Contact him at [art.pyster@stevens.edu](mailto:art.pyster@stevens.edu).



**KAHINA LASFER** is a PhD candidate in the School of Systems Engineering at Stevens Institute of Technology. Her research interests include analyzing and creating a systems-based approach for graduate

systems engineering education in the 21st century. Lasfer has an MS in computer engineering from the Stevens Institute of Technology. Contact her at [kahina07@gmail.com](mailto:kahina07@gmail.com).



**MARY SHAW** is the Alan J. Perlis Professor of Computer Science at Carnegie Mellon University's School of Computer Science. Her research interests include value-based software engineering, everyday software, software

engineering research paradigms, and software architecture. Shaw has a PhD in computer science from Carnegie Mellon University. She's a fellow of the ACM, IEEE, and the American Association for the Advancement of Science. Contact her at [shaw@cs.cmu.edu](mailto:shaw@cs.cmu.edu).

# IEEE Software

FIND US ON  
**FACEBOOK  
& TWITTER!**

[facebook.com/  
ieeesoftware](http://facebook.com/ieeesoftware)

[twitter.com/  
ieeesoftware](http://twitter.com/ieeesoftware)