The global economy rests on a software base. So the one logical imperative is that that software ought to be secure. Sadly, that is not the case. Instead according to the President’s Information Technology Advisory Committee (PITAC) “commonly used software engineering practices permit dangerous defects that let attackers compromise millions of computers every year”. This disconcertingly straightforward summary of the situation raises one obvious question, “given the critical importance of software and the fact that we have been producing it for so many years, why haven’t we done a better job of producing it right?”

Admittedly software work is complex and the actual product only exists in the virtual universe. Consider how hard it would be for physical engineers to build a product that can’t be seen and which is subject to instantaneous change based on the whim of customers and developers. At the same time, the technology itself changes so fast that most developers have to run just to stay in place. Not to mention the customers themselves, who when given a choice between “bug ridden and cheap” or “well made and expensive” will always pick what’s behind door number one. Add to that the fact that it is hard to systematically assure the reliability of something you can’t see and you get an idea what kind of problems the industry faces.

Nevertheless, there has also been a lot of talk over the past twenty years about well-defined and disciplined processes. So we must have some idea about how to do things right. Apparently we have just have not been able to get that knowledge embedded far enough into the consciousness of the industry and the public to make a dent in the problem. That is where formal education comes in.

Education shapes society. That effect should be obvious given the fact that we spend our formative years sitting together in common classrooms. In fact, no matter what will happen to us for the rest of our lives, it is guaranteed that we will spend our early years sitting in a classroom. And so school is the one place we all share. It is also the one experience that we have where the express purpose is learning. For that reason, education can be an extremely powerful force for societal change. And it is that potential influence that also makes the education process the likeliest candidate to do something about the problems of insecure software.

However the solution isn’t as simple as it seems. There are two major hurdles that prevent us from just rolling out the software security content in every classroom and waiting for the good practice to start. First, software security engineering is an emerging field. Consequently, it is not exactly clear what we should be teach-
The chief problem is finding the right scope. All evidence points to the fact that the body of knowledge for software security is cross-cutting. What that means is that element of the discipline could legitimately be taught in everything from engineering, through business and public policy, to law schools.

These are very different places indeed, so there are cultural overtones to the practical question of who ought to be teaching software security engineering and where it should be taught. That cultural difference also raises the question of “to aggregate, or not to aggregate”. If we leave the teaching of software security engineering as it currently sits we are not going to be able to coordinate that teaching, let alone evolve the field into a coherent discipline. However, if we pull all of the software security engineering education into a single discipline that begs the question of “what traditional place should we locate it in?”, since the engineers are not going to like working in a law school and vice versa.

Moreover, because the body of knowledge is so new it is unlikely that anybody on any university’s current faculty will really have much authoritative background in the discipline. That’s assuming we could say with certainty exactly what knowledge the discipline was composed of. Nonetheless, it is very hard to ask people who have specialized in some aspect of the field to just drop what they have been doing all these years and pick up a new line of teaching and research. In fact given the freedom that tenure brings, it is probably safe to say that we are going to have to wait for a new breed of faculty member to come along before we are going to make much progress on that front.

The lack of agreement about what constitutes the body of knowledge and the lack of specifically focused faculty is compounded by the fact there are no central accreditation bodies to unify the discipline. In essence, nobody in authority certifies the field of software assurance, or standardizes it, or even makes the sort of recommendations that educators need in order to develop curricula. So until somebody authoritative takes ownership of the field we educators are going to be left with “best guess” - and that is no way to build a curriculum.

What you will read in the rest of this issue are four papers that discuss how to bring software security engineering education into the mainstream. They represent many avenues of thought, which have been shared in the international community of educators about this critical topic. “A Rigorous Approach to the Definition of an International Vocational Master’s Degree in Information Security Management” by Girard et al. outlines a systematic and rigorous approach to the identification of competencies to underlie a skills card that would support the job profile for the Chief Security Officer. In conjunction with that, an associated education program is presented which would satisfy those required competencies. “Development of a Master of Software Assurance Reference Curriculum” by Mead et al. presents an overview of the Master of Software Assurance curriculum project, including its history, student prerequisites and outcomes, a core body of knowledge, and curriculum architecture. These suggestions can be used to implement a viable Master of Software Assurance program. “Secure Software Education – A Contextual Model-Based Approach” by Simpson et al. provides a high-level overview of a suggested ontology and pedagogy for secure information systems. This framework represents a part of an information assurance curriculum. It establishes an approach designed for human short-term cognition, and application of enhanced computational techniques. “Assimilating and Optimizing Software Assurance in the SDLC: A Framework and Step-Wise Approach” by Adeniji and Lee presents a methodical approach towards achieving and enhancing software assurance. Security knowledge, techniques and methodologies are studied and characterized in a proposed framework. The framework has been implemented in a graduate-level course, and the paper shares the lessons and experiences learned to improve.
future secure software engineering education.
It is our considered opinion that this sort of
wide-ranging dialogue are the first steps in
overcoming the existing hurdles and it begins
to ensure that software assurance will become
the mature discipline that we expect it to be.

We would not have been able to do this
alone, and so we would like to acknowledge
Valori Nicolai for her outstanding work in
managing the review process, Jennifer Kent for
her editorial support, and our colleagues who
served as reviewers for this issue:

Saeed Abu-Nimeh
Julia Allen
Rita Barrios
Art Conklin
Glenn Dietrich
Tom Hilburn
Paul el Khoury
Yair Levy
Seiya Miyazaki
Remzi Seker
Carol Woody

Nancy R. Mead
Dan Shoemaker
Guest Editors
IJSSE

Nancy R. Mead is a senior member of the technical staff in the CERT Program at the Software Engineering Institute (SEI). Mead is also a faculty member in the Master of Software Engineering and Master of Information Systems Management programs at Carnegie Mellon University. Her research interests are in the areas of information security, software requirements engineering, and software architectures. Mead has more than 150 publications and invited presentations. She is a Fellow of the Institute of Electrical and Electronic Engineers, Inc. (IEEE) and a Distinguished Member of the Association for Computing Machinery (ACM). Dr. Mead received her PhD in mathematics from the Polytechnic Institute of New York, and received a BA and an MS in mathematics from New York University.

Dan Shoemaker is the Director of the Institute for Cyber Security Studies, a National Security Agency (NSA) Center of Academic Excellence, at the University of Detroit Mercy. Dr. Shoemaker is a well-known speaker and writer in the area of cybersecurity. He is a Professor at UDM, where he has been the Chair of Computer and Information Systems since 1985. He is also a visiting Professor in cybersecurity at London Southbank University in the U.K. His Ph.D. is from the University of Michigan, in Ann Arbor. Dr. Shoemaker is Co-Chair of the Workforce Training and Education working group for the secure software assurance initiative within the Department of Homeland Security's National Cybersecurity Division (NCSD). Dr. Shoemaker is one of the earliest academic participants in the development of Software Engineering as a discipline, starting at SEI in the fall of 1987. He is the co-author of McGraw-Hill's best selling book on Cybersecurity “Information Assurance for the Enterprise”. He is also a prolific writer and speaker on cybersecurity topics across the Nation.