The Software Engineering Education Project's objective is to recommend curricula to prepare professionals in software engineering. This article reports on SWEEP's initial progress and its future goals to address undergraduate- and graduate-level accreditation for software engineering and technology programs.

Program Criteria for Software Engineering Accreditation Programs

Gerald L. Engel, University of Connecticut, Stamford

Traditionally, to register as an engineer, you must first graduate from an accredited program. Consequently, to establish academic programs and to gain recognition as a profession, software engineering must have accreditation guidelines.

The Software Engineering Education Project, a Software Engineering Coordinating Committee (SWECC) project, became active in 1997 with a mandate to develop accreditation criteria for undergraduate software engineering programs. The criteria—completed by November 1998 and approved by both the IEEE Computer Society and the Association for Computing Machinery—appear to have already significantly affected the accreditation development process.

0740-7459/99/$10.00 © 1999 IEEE

In this article, I will present SWEEP's work to date, discuss the undergraduate program model accreditation criteria, and review the current status of graduate and preprofessional programs.

Undergraduate Programs

SWEEP modeled its criteria for undergraduate programs (http://www.computer.org/tab/swecc/Accred.htm) after the Computer Science Accreditation Commission/Computing Sciences Accreditation Board's 1996 Criteria for Accrediting Programs in Computer Science in the United States (http://www.csab.org/criteria96.html). SWEEP selected this form because it felt the field's newness required more specification than that provided by either the Accreditation Board for Engineering and Technology's Engineering Criteria 2000 or CSAC/CSAB's 2000 Criteria. However, SWEEP believes that a program that meets the approved criteria will also meet these other standards. Nothing in SWEEP's criteria was intended to limit its applicability to the US.

Assuming the reader's familiarity with CSAC/CSAB criteria, I use its general structure and will now highlight points where its software engineering criteria differs from SWEEP's.

Much confusion exists regarding how software engineers at all levels are prepared to meet their future goals.

Evaluative criteria and faculty requirements

SWEEP's new requirements differ significantly in three ways from CSAC/CSAB faculty requirements.[1] First, SWEEP requires that faculty provide sufficient direction and guidance to students and student teams. A software engineering program requires considerable time to both identify and implement project work, naturally placing extra demands on instructors beyond their more traditional teaching duties.

The other two points address the need for a firsthand understanding of software engineering: by maintaining effective interaction with software professionals and by requiring faculty teaching core computer engineering classes to have substantial practical software engineering experience.

Curriculum

As you might expect, SWEEP has made the greatest changes in terms of curriculum, replacing the entire section. Software engineering encompasses theory, technology, practice, and the application of software in computer-based systems. To meet these needs, SWEEP requires that a program be divided in four, approximately equal segments: software engineering, computer science and engineering, appropriate supporting areas, and advanced materials. Combined, these four categories represent about 75% of an undergraduate program, leaving 25% for individually selected electives and institutional requirements.

For software engineering, SWEEP addresses processes and techniques for developing and maintaining large systems. Courses should cover

- requirements analysis,
- software architecture and design,
- testing and quality assurance,
- software management,
- selection and use of software tools and components,
- computer and human interaction, and
- maintenance and documentation.

Also, this section should involve substantial design work and a variety of languages and systems. This focus should also introduce ethical, social, legal, economic, and safety issues, as well as appropriate use of engineering standards. Subsequent work should also reinforce these issues.

The computer science section covers most of the required computer science and engineering core; however, there is room for variance from the traditional computer science program. Particularly, this section addresses: algorithms and data structures, computer architecture, databases, programming languages, operating systems, and networking.

The supporting areas section covers other materials necessary to function as a software professional, such as communication skills (oral, written, listening, and teamwork) and mathematics (discrete, probability, and statistical).

Advanced work should meet overall program objectives by providing depth in one or more core areas. It might incorporate further study in previously discussed topics, or it might involve work in additional areas of theory or technology. This part of the program must include work in one or more significant software engineering application domains.

Additionally, a software engineering program must address all aspects of software development and maintenance and provide experience in a realistic team environment. This aspect should be integrated throughout the program, which must include a major project that integrates most of the curriculum's aspects.

Laboratory and computing resources

SWEEP added three elements to the laboratory and computing resource requirements, requiring

- access to commercial tools that support pertinent software development tasks;
- meeting space to support team projects; and
- a laboratory plan to address hardware and software development, acquisition, management, and maintenance.

Students

In addition to the CSAC/CSAB student requirements (listed as program administration), SWEEP's criteria specifies that a program must have sufficient students to ensure realistic projects and activities.

Graduate Programs

SWEEP has concentrated on undergraduate program requirements. However, to fully address the field's needs, SWEEP believes a continuum of educational programs—ranging from preprofessional to continuing education—is necessary. Recently, SWEEP has begun addressing these needs in graduate-level and preprofessional programs.

For advanced-level programs, SWEEP plans to make recommendations for accreditation criteria. Historically, program accreditation is applied to the program that provided a field's first professional degree. As a result, ABET, for example, accredits master's degree programs in an engineering field only if no accredited basic-level program exists at the same institution.

However, SWEEP believes that software engineering must occasionally deviate from this practice. Many computer science programs, for example, began as graduate (usually master's) programs and later migrated to undergraduate programs. Some of these master's programs remain, but they offer an undergraduate degree, still called a master's. These programs can provide a valuable service, especially for individuals changing careers. However, there needs to be a clear differentiation between these remedial programs and truly advanced programs, which, by definition, should build on basic-level programs.

SWEEP believes that software engineering programs will soon need to make the same distinction and, hence, require advanced-level criteria. As a basis for its advanced-level criteria, SWEEP uses the General Advanced-Level Criteria in the 1997–1998 ABET Criteria for Accrediting Engineering Programs.[2] SWEEP selected this version rather than the later Criteria 2000[3] version because Criteria 2000 did not meet its need for additional specification levels in a newly developing field. SWEEP believes that the resulting criterion will be an interpretation of Criteria 2000's requirements, so that if a program meets SWEEP's criteria, it will also meet Criteria 2000's requirements.

As envisioned, the advanced-level criteria will require sufficient faculty committed to the program to effectively offer the program. Additionally, these faculty must be qualified, as indicated in the basic-level criteria. As with any program, the institution must specify the program's objectives in terms of what its graduates will need to be able to develop and show how it will meet those objectives.

Graduate programs must insure that each graduate meets all of the general predefined, basic-level software engineering curricular content criteria. The program must then require the equivalent of at least one year of study beyond the basic-level requirements. The SWECC's Software Engineering Body of Knowledge project will additionally specify the requisite advanced-level materials.

Advanced programs must include a software engineering project or software engineering research activity, resulting in a thesis or report that demonstrates both mastery of the subject matter and a high level of communication skills.

Additional requirements for the accreditation of graduate software engineering programs remain effectively the same as those for basic-level programs, including laboratory and computing resources, program administration, and institutional support.

Preprofessional Programs

Although SWEEP has not completed the accreditation criteria specifications for software technician programs, this area remains of significant concern. Much confusion exists regarding how software engineers at all levels—undergraduate, graduate, and technician—are prepared to meet their future goals. Regrettably, quite often, employers select an inappropriate backgrounding, producing an unsatisfactory result for all concerned.

The key to preparing software technicians is to prepare them for immediate employment in the software community. Simultaneously, however, they should receive a foundation that would let them continue to upper-division studies without penalty.

The technical design of preprofessional programs must strongly emphasize the systems and techniques needed for supporting a software engineering project, including the use of appropriate resources (both software and hardware). Work in the program must include exposure to current systems, which is crucial to immediate job placement, and also include significant portions of the core software engineering material specified in the undergraduate program requirements.

With the accreditation guidelines for undergraduate software engineering programs complete, SWEEP will turn to undergraduate curriculum issues. At the same time, SWEEP will finalize the accreditation criteria for the preprofessional and graduate programs.

Work on the undergraduate program will involve a number of strategies. Much of the work will look at those aspects of software engineering that differentiate it from computer science and engineering. Because this often involves issues of practice, professionalism, and ethics, SWEEP will interact significantly with the appropriate SWECC group working in these areas. To best serve the field, SWEEP will widely distribute its work results, and collaborate with other similarly focused groups and individuals. Plans include hosting an invitational working conference of interested professionals, which will include the IEEE Computer Society and ACM Curriculum 2001 project and the Working Group on Software Engineering Education and Training.

In developing its undergraduate curriculum, SWEEP recognizes that it must include much of the material identified by the SWECC Body of Knowledge project and will coordinate with that group. Through this synergy, SWEEP hopes to form a consensus on the philosophy of undergraduate software engineering programs. This will lead to the next phase which will be to turn the philosophy into specific curriculum recommendations. Similar work will then follow to accomplish the same things with the preprofessional and graduate programs. ❖

Acknowledgments

I cochair SWEEP with Richard LeBlanc (Georgia Institute of Technology). Current SWEEP members are Bruce H. Barnes (George Mason University), Laurie H. Werth (University of Texas, Austin), Martin Griss (Hewlett Packard), and Tony Wasserman (Software Method & Tools).

In developing its recommendations, SWEEP works closely with groups interested in software engineering education. These include the joint IEEE Computer Society/ACM group working on Curriculum 2001, the ad hoc Working Group on Software Engineering Education and Training, and the joint working party of the British Computer Society and the Institution of Electrical Engineers on undergraduate curricula for software engineering.

Input regarding all aspects of the project is sincerely solicited.

References

1. *1998 Annual Report*, Computing Sciences Accreditation Board, Stamford, Conn., 1998.
2. *Criteria for Accrediting Programs in Engineering in the United States*, Accreditation Board for Engineering and Technology, Baltimore, Md., 1997–1998.
3. *Criteria for Accrediting Programs in Engineering in the United States*, Accreditation Board for Engineering and Technology, Baltimore, Md., 1998–1999.

About the Author

Gerald L. Engel is the Leonhardt Professor of Computer Science and Engineering at the University of Connecticut, Stamford. His research interests have focused on education issues in computer science and engineering, and ethical and public policy concerns in computing. He has a BS in mathematics from Hampden-Sydney College, VA; an MA in mathematics from Louisiana State University; and a DEd in computer science from Pennsylvania State University. He has served as Vice President for Education of the IEEE Computer Society and as Vice President of the Computing Sciences Accreditation Board. He is currently President of the IEEE Society for Social Implications of Technology and cochair of the joint ACM/IEEE Computer Society Software Engineering Education Project.

Contact Engel at Computer Science and Eng., Univ. of Connecticut, Stamford, 1 University Place, Stamford, CT, 06901-2315; gengel@Stamford.stam.uconn.edu.