Software Engineering Degree Programs

Today, software engineering is a separate degree program in academia. The implications for the Department of Defense (DoD) are that the new graduate, once on the job, is going to be better prepared to contribute as a technical team member with appropriate experience and knowledge. This article outlines some of the changes that are taking place and briefly describes accreditation direction for such programs. It is anticipated that the new graduate will be better prepared to immediately contribute to software engineering organizational goals.

After more than 25 years in the defense software business and now in academia, this author is pleased to report that many academic institutions are beginning to offer, or are considering, undergraduate and graduate software engineering (SE) degree programs in addition to existing computer science programs. This is true of the author's institution where approval for its new SE bachelor's degree program is expected this year.

These emerging degree programs are more directed at the practical than the theoretical. Shari Pfleeger [1] synopsizes the difference between computer science and a software engineering focus very well in her recent text on software engineering where she writes, "*We can concentrate on the computers and programming languages themselves, or we can view them as tools to be used in designing and implementing a solution to a problem. Software engineering takes the latter view . . . .*"

The Institute of Electrical and Electronics Engineers Standard 610.12 defines software engineering as "*the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.*"

In all software engineering programs this author is familiar with, computer science, computer engineering, and industrial engineering are supporting fields. The combination of these disciplines and the focus on practical application will lead to a different kind of graduate than those coming out of more theoretical programs today. This change in emphasis is likely to meet with approval on the part of the DoD and industry who hire our graduates and expect them to be technically proficient and immediately productive.

This article outlines the essence of these new programs and provides some insight into a typical curriculum associated with them—a curriculum that should better meet the current needs of the software engineering community and address the software engineering crisis so often referred to in today's literature [2, 3].

Computer science programs have been around a long time and for the most part have included software engineering course offerings as required or elective coursework. Business information systems degrees also have been available for some time—many dating to the mid-1960s or early 1970s. Computer engineering programs also exist with well-established curricula. With this wide variety of related programs one might question the need to introduce a degree specifically in software engineering. It helps to look at it from two perspectives—first reviewing the focus of each of the degree programs mentioned above and secondly, by considering the state of our software infrastructure today and how existing programs might be expected to contribute to solving problems. These programs are characterized later in this article.

Computer science programs generally focus on the principles, applications, and technologies of computing with an emphasis on data, data structures, algorithms, computer architectures, and theory of computing. Computer science also includes the fields of artificial intelligence, graphics and/or scientific visualization, computer language design, structure, and translation. Problem solving and design methodology is often included. Such programs are largely, but not entirely, theoretical. Business information systems programs tend to focus the student more on the management aspects of providing computing services to an enterprise—costing, accounting, economic impact, systems analysis procedures, and business data processing. One university's course catalogue description for such a program [4] includes the statement " . . . must have a broad background and understanding of the business environment, including such topics as accounting, economics, law, management, production, marketing, finance, and communications."

Clearly, such information systems programs are directed at producing a graduate who serves an important role in the operations, planning, and maintenance functions associated with an enterprise's supporting computer infrastructure —but not one prepared to lead technical efforts in building large-scale, software-intensive systems. Computer engineering programs grew out of electrical engineering applied to the computing domain. Such programs tend to be more oriented to the hardware aspects of computing, with emphasis on circuit design and a strong complement of software courses to include algorithms and programming skills. The graduating computer engineer has a strong background in mathematics, circuit theory, digital devices, architecture, and chip design, as well as basic computer science that includes programming, data structures, networks, and systems theory.

Each of these three programs in any academic institution is comprised of somewhere around 120-140 semester hours (where one class, on average, is worth three credit hours). Once a student finishes all required core courses, about 40 percent of the course work is accomplished in the specific major field of study. It becomes apparent that universities today have little expansion ability in their programs, and adding a course usually means that something else must go. It would seem that encouraging any of the degree programs listed above to focus more on the tools and techniques of software engineering can only come at the expense of other important topics that form the essence of the degree.

Software Engineering

This is an area of interest that has been with us for around four decades. The early years involved very little engineering, comprised mostly of tools and procedures used in programming. About the mid-1960s we began to notice a significant problem in living with the software we had created—it was largely difficult and expensive to maintain. With about 450 different languages documented in the

DoD at that time, coupled with few programming standards that existed or were followed, today's software crises began to appear. We spent the next two decades researching ways to improve and measure our processes, reduce costs and ease the maintenance issue, correctly build software and to prove we did so, and incorporate rigorous process descriptions. We approached these problems largely through our computer science programs, which usually included a software engineering component, and a certain complement of skills associated with modern programming languages and procedures.

The emphasis of such programs was —and remains—primarily theoretical with a strong emphasis on improving our techniques and processes for tomorrow's development environments and systems. Today, software engineers are needed who have more depth in current accepted procedures, processes, tools, measures/metrics, and a quality focus. To meet this need, over the past few years the academic community began to agree that software engineering has matured to the point that it is now a credible separate degree program with a community whose needs are unmet by other programs. The accrediting body for engineering degree programs, the Accreditation Board for Engineering and Technology, adopted criteria for software engineering. Programs to meet industry's needs are beginning to emerge at the undergraduate level.

The new graduate, according to the desires expressed in the accreditation criteria (www.computer.org/tab/swecc/accred.html) will be able to demonstrate an ability to analyze, design, verify, validate, implement, and maintain software systems using appropriate processes, models, and metrics in software development. This graduate also is expected to possess necessary team and communication skills to function in a typical software development environment. Those of us that have worked in these "typical software development environments" will welcome this change of emphasis.

It has been this author's experience that graduates join industry with good technical skills but lack the office social skills so important to overall corporate success. We seem to have too many assignments in academia that all start with the instructions, " Do your own work, do not get help from any outside source, do not give any help to others working on this assignment, do not talk to others about this assignment . . ." and so forth.

When our students reach their first employers they are put onto a team, expected to cooperate with others, share their work, help each other, communicate, etc. We reverse the paradigm and many find it difficult to adapt. Our emerging software engineering degree programs should help modify this approach by introducing more team-oriented projects, focusing on deliveries that require documentation built to professional standards, and requiring students to think in terms of an entire project life cycle rather than on a specific program due before 3 p.m. Friday. The accreditation criteria also suggests a curriculum that should include approximately equal segments in software engineering, computer science, and supporting areas, with the total courses covering about three-fourths of the total academic program. This will undoubtedly allow our graduates to spend the time needed to learn corporate practices, quality, program management principles, and the process needed to deliver and maintain large-scale software-intensive systems.

Software Engineering Accreditation Expectations

It is the accrediting bodies in academia that insure certain educational goals are met and that they can be met in a particular institution. Periodically, degree programs are assessed by these accrediting bodies and are approved or denied accreditation. Software engineering criteria contain certain requirements that are most certain to find favor within the large DoD community of employers outlined below:

Faculty Requirements: A key expectation is that faculty who teach core software engineering courses should have substantial practical software engineering experience. Faculty must also be able to interact effectively with software practitioners. The implication appears to be that the faculty should have the ability and experience to teach practical skills necessary to enter the software engineering work force and to develop collaborations with industry. Many academic departments today promote the faculty's ability to obtain research grants from organizations like the National Science Foundation (NSF) or the Defense Advanced Research Projects Agency (DARPA), but do not value industrial or government agency collaborations as highly. Often NSF or DARPA grants, for instance, count heavily toward the tenure process that a faculty member faces, whereas industrial collaborations may not —removing the motivation to enter into such arrangements. The software engineering faculty member may, in future programs, be encouraged to reverse this trend.

Curriculum: The criteria specifically states that a central theme "is to engender an engineering discipline in students, enabling them to define and use processes, models, and metrics in software and system development." Curriculum guidance includes theory and practice, but the emphasis is on practice. Programs must include all aspects of software development and maintenance and provide for experience in a realistic team environment. Written and oral communication skill courses are promoted as appropriate supporting courses. Courses such as development and maintenance, requirements analysis, architecture and design, testing, and quality assurance are required. Who among us would not want such a graduate? These changes will certainly bring more valuable employees to the workplace and will allow them to become immediately productive.

Laboratory and Computing Resources: Substantial laboratory and computing resources are expected in these programs, both for student exposure and support as well as to support the research of the faculty members. Such laboratories are to be populated with a wide variety of tools, computing facilities, operating systems, and commercial products. Interestingly, the accreditation criterion also suggests meeting space to support team projects, an essential aspect of software engineering training. Laboratory support staff is recommended, although in most academic institutions, it is difficult to find funding for permanent laboratory support staff.

Institutional Support: Support from the university administration is essential for software engineering programs. The obvious support needed is opportunity for faculty to stay current, good library support, resources to build strong programs, and support for collaboration with industry and government software engineering organizations. Less obvious areas of support that would likely be considered helpful include tenure credit for working with industry, industrial affiliation agreements between the university or software engineering program, industry/government organizations, and support for distance learning technologies.

Support for distance learning technologies would assist in electronically including working adults in software engineering organizations as members of software engineering classes on campus.

Industrial affiliation agreements would lead to exchange opportunities, student co-op programs, internships, and work collaborations.

Summary

Change is on the horizon for those concerned with addressing the need for software engineers in the future. This is a welcome change, heralding the maturing of our discipline in the eyes of others. Several undergraduate software engineering programs exist today and more are planned.

The need is clear. This nation has an aging software infrastructure that must be maintained and replaced. We have begun to develop the necessary rigor in our profession to teach a common body of knowledge (see www.swebok.org for more on this subject).

At least one state—Texas—has begun a process of licensing software engineers as professionals, and we have learned hard lessons over the years regarding the need for process and how to measure it. Never has there been a more exciting time to teach software engineering and research improvements in the state of the practice.

Those in the software engineering profession can look forward to the new, improved graduate beginning to appear on the scene in the immediate future. For more information on the planned program at Mississippi State University, please contact the author directly. For more information on software engineering programs visit the following web sites:
www.swecc.org
faculty.db.erau.edu/hilburn/se-educ/
www.lrgl.uqam.ca/publications/pdf/365.pdf
www.cs.utexas.edu/users/ethics/professional.html
www.main.org/peboard/softw.htm◆

References

1. Pfleeger, Shari, *Software Engineering Theory and Practice*, Prentice Hall, 1998
2. Gibbs, W.W., “Software’s Chronic Crisis,” *Scientific American*, Sept. 1994, pp. 86-95.
3. Royce, W.W., Current Problems, *Aerospace Software Engineering: A Collection of Concepts*, C. Anderson and M. Dorfman, eds., American Institute of Aeronautics, Inc., Washington, D.C., 1991.
4. Mississippi State University, *The Bulletin of the Mississippi State University 1998-1999*, P.O. Box 5268, Mississippi State, Miss. 39762

About the Author

Rayford B. Vaughn Jr. spent 26 years in the Army as a software engineer. His key assignments included commander of the Army's Information Systems Software Center headquartered at Fort Belvoir, Va. and the first Director of the Pentagon Single Agency Manager for Information Technology Services. Upon retirement as a colonel in 1995, he joined Electronic Data Systems (EDS) Military Systems as vice president of Defense Information Systems Agency (DISA) Integration Services, where he was responsible for all EDS contracts issued by DISA. In October 1997, he accepted a position as an associate professor of computer science at Mississippi State University, where he teaches software engineering and computer security courses at the graduate and undergraduate level. He is a past contributor to CROSSTALK and a frequent speaker at the DoD Software Technology Conference.

Department of Computer Science
PO Box 9637
Mississippi State, Miss. 39762
Voice: 662-325-7450
Fax: 662-325-8997
E-mail: vaughn@cs.msstate.edu
Internet: http://www.cs.msstate.edu

Education and Training Web Sites

http://rbse.jsc.nasa.gov/ricis/research-area/et/et.html
Links cover, but are not limited to, software engineering and Ada training; research in intelligent tutoring systems for knowledge-poor domains; software engineering and training implementation research; hypermedia tools for building technical training systems, information systems research center; continued research and development with the microcomputer intelligence for technical training concept; and establishing an integrated education program in software engineering and computer engineering training, phase one.

http://www.isse.gmu.edu/~ofut/rsrch/see.html
Visit this site for papers on "Maintaining Knowledge Currency in the 21st Century, "A Five-Year Perspective on Software Engineering Graduate Programs at George Mason University," "Integrating Research, Reuse, and Integration into Software Engineering Courses," and "Anatomy of a Software Engineering Project."

http://www.asafm.army.mil/profdev/acteds.htm
This site is by the Army Civilian Training Education and Development System. Those viewing it need Adobe Acrobat Reader or another application capable of reading .pdf files. This site provides systematic training and development for careerists from intern to senior managerial and executive levels. The plan outlines sequential and progressive training in functional specialties, leadership, supervision, and managerial development.

http://www.seminarfinder.com/search/
This is used to find seminars, Web-based training programs and continuing education. Visitors to this site can browse by city or topic. There also are links to training, and affiliate programs.

http://www.iee.org.uk/PG/S5
Professional Group S5 of the Institution of Electrical Engineers provides this site on the processes of education and training in the context of electrical, electronic, manufacturing, and software engineering. It covers, but is not limited to, such things as methods of attracting young people into the profession and continuing professional development, special educational requirements for identifiable sectors of industry and related industrial training programs, and education and training aspects of specific subject areas.