

The Internationalization of Software Engineering Education

Barbara Bernal Thomas, Sheryl L. Duggins

School of Computing and Software Engineering
Southern Polytechnic State University
Marietta, Georgia 30060
bthomas@spsu.edu, sduggins@spsu.edu

Abstract

Given that software engineering is a very young discipline and is still emerging, how does the international community unite to precisely define what knowledge constitutes software engineering? This paper examines an ongoing international collaboration seeking a definition of the software engineering body of knowledge. The significance of the results of the project on current software engineering curriculum is investigated and the state of software engineering degree programs around the world is examined.

Introduction

The software engineering discipline began when the term "software engineering" was first used in 1967 by a NATO study group to discuss "the problems of software" and popularized with the subsequent 1968 conference in Germany¹⁰. Over the last 35 years, the international software engineering community has made progress towards defining the discipline, with the last decade being marked by tremendous progress towards defining software engineering on an international scale.

The efforts of the Software Engineering Coordinating Committee (SWECC) are investigated along with the current impact of the Guide to the Software Engineering Body of Knowledge (SWEBOK) project. The purpose of SWEBOK is to identify the core body of knowledge necessary to define software engineering as a profession. The significance of the results of the SWEBOK project on current software engineering curriculum is investigated and the repercussions on future software engineering programs are examined from the viewpoints of international stakeholders from academia as well as industry. A report of the current software engineering international programs available and data regarding graduates and faculty of the programs is included. The debate on what knowledge comprises software engineering will be presented and the current state of international collaboration will be explored.

The challenges in identifying and characterizing the content of the software engineering body of knowledge were taken into consideration in the formulation and revision of Southern Polytechnic State University's undergraduate and graduate degrees in Software Engineering. Our efforts in

curriculum design will be explored and the influence of the international community on our programs will be discussed.

Southern Polytechnic State University's Software Engineering Programs

Marked by continual change, the last decade has seen steady progress in software engineering education. In a discipline that is this new, the question of what to teach is particularly difficult to answer; in an innovative field that is drastically changing as quickly as software engineering is, issues of currency and relevancy in curricular design take on entirely new dimensions.

In an effort to stay current, the School of Computing and Software Engineering at Southern Polytechnic State University conducted a curriculum evaluation study for our Master of Science in Software Engineering (MSSwE) degree and Bachelor of Science in Software Engineering (BSSwE). During the course of the evaluation, we searched the literature for curriculum guidance while gathering material on other software engineering programs currently offered around the world⁶. We identified the various cornerstones in the evolution of SE curriculum as it matured over the last two decades¹⁵.

One of the cornerstones that we found to be of particular relevance today is the work done by the SWECC on the SWEBOK project. Their international effort along with the current impact of the SWEBOK project on software engineering curriculum played a critical role in the revision of our MSSwE degree and the development of our BSSwE.

Software Engineering Coordinating Committee (SWECC)

In 1993, the Joint IEEE Computer Society and ACM Steering Committee for the Establishment of Software Engineering as a Profession was created. In 1998, this committee was superseded by the Software Engineering Coordinating Committee (SWECC) and was established to act as a "permanent entity to foster the evolution of software engineering as a professional computing discipline"⁸. It was determined that achieving consensus by the profession on a core body of knowledge was crucial for the evolution of software engineering to a profession. With the approval of both IEEE-CS and ACM, SWECC set up the Guide to the Software Engineering Body of Knowledge (SWEBOK) project⁸ to identify that subset of the body of knowledge that is "generally accepted." The objectives of the SWEBOK project are to:

1. Characterize the contents of the Software Engineering Body of Knowledge;
2. Provide a topical access to the Software Engineering Body of Knowledge;
3. Promote a consistent view of software engineering worldwide;
4. Clarify the place of, and set the boundary of, software engineering with respect to other disciplines (computer science, project management, electrical engineering and mathematics, etc.);
5. Provide a foundation for curriculum development and individual certification and licensing material.

Software Engineering Body of Knowledge (SWEBOK) Project

SWEBOK is a four-year project with three scheduled reports: Straw Man, Stone Man and Iron Man. Each report builds upon the previous one. Straw Man and Stone Man have been completed and Iron Man is currently underway. Achieving consensus by the profession has been

facilitated by the open solicitation of reviews of the working reports ¹⁶. The project is developing broad consensus on the core of the software engineering discipline.

The initial Straw Man report selected the Knowledge Areas (KAs) based on “recognized, public and verifiable sources of information” including “tables of contents of general software engineering textbooks, the curricula of undergraduate and graduate programs in software engineering and the admission criteria for graduate programs” ⁸. The Straw Man KAs consisted of: Development Process, Maintenance Process, Configuration Management, Quality Assurance, Verification and Validation, Improvement Process, Software Development Methods, Software Development Environments, Software Engineering Overview and Definition, Measurement/Metrics, and Software Reliability ⁸. Potential knowledge areas were refined based on the ISO/IEC 12207 standard on Software Life-Cycle Processes. The Straw Man knowledge areas represent a life-cycle model of software development.

The Straw Man report also identified seven disciplines related to software engineering: cognitive science and human factors, computer engineering, computer science, management and management science, mathematics, project management, and systems engineering ⁷. Prior to that report, there had been an ongoing debate about the relationship between software engineering and computer science. For years, software engineering was considered an area within computer science, but with the release of the Straw Man report, software engineering is now widely recognized as an independent discipline that is related to computer science in the same way that it is related to mathematics, or computer engineering.

The Stone Man report is divided into sections that include the knowledge area identification and specification including breakdowns, rationale, standards, and relevant ties to the seven disciplines related to software engineering identified above. The ten knowledge areas identified in the Stone Man report (version 0.95) as of May 2001 are: ⁹

1. Software Requirements,
2. Software Design,
3. Software Construction,
4. Software Testing,
5. Software Maintenance,
6. Software Configuration Management,
7. Software Engineering Management,
8. Software Engineering Process,
9. Software Engineering Tools and Methods, and
10. Software Quality.

One major shift in direction that can be seen by examining the progression of the Stone Man documents is a gradual departure from the life-cycle model. There is a continuum of refinement of the knowledge areas away from partitioning based on phases of the life-cycle, to a new, continually evolving software engineering body of knowledge. It is evident that the Stone Man report extends the software engineering model beyond the influence of the ISO/IEC 12207 standard.

Another major shift in the direction of the project has to do with the role of licensing software engineers ⁵. As specified initially, the SWEBOK project had five objectives. However, many believe the fifth objective: to “provide a foundation for curriculum and individual certification

and licensing material” was given much more emphasis than originally intended. The next section will examine the issue of licensing in more detail.

The current phase of the SWEBOK, Iron Man, will be based on trial application of the current version of the guide in realistic situations. Using the SWEBOK guide, pilot trial applications will define the various competencies needed to implement a comprehensive set of processes for the software life cycle. The resulting definitions will pass through industry evaluation with a final evaluation of the utility of the guide in performing the task³.

Licensing Software Engineers

Concerned about the direction SWECC was moving, ACM established task forces to investigate the issue of licensing software engineers¹. The study found an “explicit and intimate link” between the SWEBOK project and “the intent and expectation for software engineering licensing”¹. Based on the study, the ACM Council decided in May 1999 that it could not support licensing of software engineers because current software engineering practice is too immature to warrant licensing.

In May 2000, the ACM Council decided that the framework of a licensed professional engineer, which was originally developed for civil engineers, “does not match the professional industrial practice of software engineering. Such licensing practices would give false assurances of competence even if the body of knowledge were mature; and would preclude many of the most qualified software engineers from becoming licensed”¹¹. Because SWECC became so closely linked with licensing of software engineers, the ACM Council decided to withdraw from SWECC.

The impact that this decision will have on software engineering curriculum is yet to be seen. However, the SWECC represented a joint effort involving both IEEE-CS and ACM working together to develop a core body of software engineering knowledge and to foster the evolution of software engineering to a profession. The recent withdrawal of ACM from the SWECC due to differences in opinion on the emphasis being placed on licensing software engineers has changed the perceived value of the contributions of SWEBOK, although it still represents a strong international collaboration.

In Canada, the time for instituting licensing software engineers has already begun. Canada views software engineering as a specialty within engineering that has its own specific knowledge. Dave Parnas, from McMaster University in Ontario, Canada commented: “The most difficult task facing us in the licensure of software engineers is the identification of the special knowledge that should be required of those who practice in this field”¹⁴. A licensing authority created by legislation has the authority to enforce the awarding of licenses, the removal of licenses from licensed practitioners who do not follow established rules of good practice, and the ability to stop those who practice without a license in situations where a license is required. Legislation is currently in place in Canada for licensing software engineers, and some licensed software engineers are practicing today.¹⁴

The licensing of software engineers in the state of Texas, USA, was based on the common standard used for all fields of engineering. The Texas Board of Professional Engineers is charged by the state legislature with implementing and enforcing the licensing of professional engineers (P.E.s). Since the Texas Board interprets the legislative mandate as including software

engineers, those who offer services as software engineers are required to obtain the P.E. license in Texas. This raised the question of whether the rules enforced by the Texas Board seemed to force software engineers into a mold designed for other distinct disciplines so that a P.E. license could be issued, or whether software engineering was added to the P.E. license to allow the Board's governing regulations to match actual practice rather than to force the practice into a particular mold. Currently, the Texas licensing committee requested help from ACM and IEEE-CS to jointly define the body of knowledge on which software engineering licensing will be based ².

International Influence within the SWEBOK

The SWEBOK has two underlying principles, transparency and consensus-building, which incorporate a steady method of international involvement. The transparency principle is achieved by publishing the development process of the SWEBOK in a fully documented website www.swebok.org making it free and available to all. The consensus-building principle is that the development process of SWEBOK is designed to build, over time, consensus in industry, among professional societies and standards-setting bodies, and in academia. The consensus-building was facilitated by the development and review process itself. Specifically, there were four versions of the Stone Man Report (0.1, 0.5, 0.7, and 0.9) which were reviewed by the international community in stages appropriate to the level of the document (domain experts for 0.1; selected users for 0.5; and the international SE community at large to review 0.7). To get a sense of the level of international involvement in this project, of the 200 selected reviewers for version 0.5, 400 reviews were received originating from 25 different countries. The community that reviewed version 0.7 consists of more than 800 registered software engineers from both academia and industry from all around the world, with all comments made public as well as the identity of the reviewers ³.

The final stage of the SWEBOK is a diffusion and experimental usage of the current report in realistic situations. This diffusion will be delivered in conference workshops, presentations to companies and professional societies, and pilot project efforts. Feedback from all these forms of diffusion and experimental usage of the guide will be available on the SWEBOK website. Consensus by the community regarding the specification of the core body of knowledge is key for the evolution of software engineering (SE) to reach professional status ³.

International Collaboration on Software Engineering

While there is little debate about the international influence that the SWEBOK project has had on the SE community, it is by no means the only forum for international SE collaboration. There are a number of international software engineering conferences that seek to bring together academia and industry to share knowledge and push the boundaries of collaboration even further. These include the Conference of Software Engineering Education and Training (CSEET), which is in its 15th year, with the 2003 conference to be held in Madrid, Spain; the International Conference on Software Engineering (ICSE), which in its 23rd year was held in Toronto, Canada, in May 2001; the International Conference on Software Engineering and Knowledge Engineering (SEKE), which in its 13th year was held in Buenos Aires, Argentina, in June 2001; and there are several other international conferences the interested reader can locate. In addition to conferences, there is another type of international SE collaboration that deserves to be noted. There is an online forum, called the Forum for Advancing Software Engineering (FASE), which is located at www.cs.ttu.edu/fase, that is an online journal of SE with international contributors.

Another interesting type of international collaboration between academia and industry can be found in the consortium of international universities known as the International Software Engineering University Consortium (ISEUC – pronounced “I see, You see”) ¹². This consortium currently has 35 members from Australia, Canada, the U.K., and twenty states in the U.S.A., who provide Software Engineering courses to students all over the world via distance learning. “ISEUC is based on international best SE education practices, accreditation standards, credit programs and professional education” and provides undergraduate and graduate credit courses, as well as post-graduate professional education courses ¹¹. The goal for the ISEUC is to provide access to software engineering education for software-intensive industries on a global scale from international universities with established programs or degrees in software engineering. The courses are primarily delivered asynchronously using Web-based technology, but also include some synchronous delivery, including either face-to-face, video conferencing, or interactive sessions. For more information or to consider becoming a member, visit www.iseuc.org.

Current International SE Programs

The first results of a large-scale survey of international academic software engineering programs found almost 6800 students enrolled in SE degree programs in 11 different countries ¹³. There were 60 institutions that participated in the survey, and the breakdown of the institutions is as follows: thirty-five institutions were from the United States; four institutions were from Canada; two institutions were from the United Kingdom and (Stuttgart, Slovakia); and one institution each from Ireland, Sweden, Spain, South America, Australia and New Zealand. This survey incorporated existing databases compiled by Software Engineering Institution (SEI), Forum for Advancing Software engineering Education (FASE), and other research endeavors to create a reliable source of information. The database confirms 5500 people have obtained degrees in SE, degrees with concentrations in SE, or certificates in SE since 1979, as of January, 2001. The survey includes private, public, military, and consortia type of institutions. There were 94 different academic programs offered by 60 institutions with 350 full-time faculty and nearly 200 part-time faculty teaching the software engineering curriculum.

The survey contained a section for the respondent to select the definition of software engineering that their institution most agrees with. There were five definitions of software engineering given including the definitions from: the Texas Board of Professional Engineers; the Computing Research Association; IEEE Standards Collection: SE Section; NATO; and Parnas; or if none of those definitions matched their institution’s, respondents were asked to state their own definition. The results from the 85 respondents were:

- 29 selected the IEEE Standards Collection: SE definition,
- 23 selected the Texas Board of Professional Engineers’ definition,
- 21 selected NATO’s definition,
- 17 offered personal definitions,
- 14 chose Parnas’ definition, and
- 9 selected the definition for the Computing Research Association ¹³.

The data from this survey shows one of the fundamental problems in the SE community: the lack of a widely accepted definition of software engineering. The October 2001 issue of *Ubiquity*, an online IT publication and forum for ACM, contains the article "What is Software Engineering?" ⁴. One of the discussion forums generated in response is found at

<http://www.cs.ttu.edu/fase/fase-talk.htm#ubiquity>. This forum contains numerous responses that illustrate the lack of a common definition of software engineering.

Conclusions

The still emerging field of software engineering does have an international community with a common goal of reaching some consensus regarding knowledge within the SE field. The main contribution of the SWEBOK guide is an ability to diffuse the current information and opinions on an international scale about the relevance of the body of knowledge for SE education. Another benefit that SWEBOK provides is regarding curricular design. Institutions can use the SWEBOK guide as a tool to provide assistance in developing the curriculum for any SE degree program.

Perhaps the most significant development that resulted from the efforts of the SWECC is that the field of software engineering now has international influence and is both known and respected throughout the world. Industry hires software engineers, companies know of software engineering, numerous SE books can be found, and licensing efforts are active. While SE is still struggling to define itself, it is evolving into an international discipline with a rich mixture of topics and specialties.

Bibliography

1. ACM Council "A Summary of the ACM Position on Software Engineering as a Licensed Engineering Profession" (July 17, 2000) http://www.acm.org/serving/se_policy/selep_main.html
2. ACM Panel on Professional Licensing in Software Engineering "Report to Council" (May 15,1999) http://www.acm.org/serving/se_policy/report.html
3. Bourque, R., Dupuis, R., Moore, J. W., Deweese, P. (May 2000) One Day Workshop: Guide to the Software Engineering Body of Knowledge: Overview and Applications, presented at STC 2000, The Twelfth Annual Software Technology Conference, Salt Lake City, Utah.
4. Curran, B. (October 22, 2001), "What is Software Engineering?" Ubiquity. Views, Volume 2, Issue 33. http://www.acm/ubiquity/views/b_curran_1.html
5. Duggins, S. L. (March 2001) "Curriculum Impact of the Maturing Software Engineering Profession" in Proceedings in the 2001 ASEE Southeastern Section Conference.
6. Duggins, S. L. & Thomas, B. B. (February 2002) "An Historical Investigation of Graduate Software Engineering Curriculum" in Proceedings of the Fifteenth Conference on Software Engineering Education & Training, pp. 78-87, IEEE Computer Society, Los Alamitos, California.
7. Dupuis, R., Bourque, P., Abran, A., Moore, J. W., & Tripp, L. (March 26, 1999) A Baseline for a list of Related Disciplines for the Stone Man Version of the Guide to the Software Engineering Body of Knowledge, <http://www.swebok.org/>
8. Guide to the Software Engineering Body of Knowledge - A Straw Man Version - September 1998.
9. Guide to the Software Engineering Body of Knowledge – A Stone Man Version (Version 0.95) SWEBOK May 2001. <http://www.swebok.org/stoneman/version095/>
10. Naur, P. & Randell, B. (eds.) (1969) Software Engineering: A Report on a Conference Sponsored by the NATO Science Committee, NATO.
11. Notkin, D., Gorlick, M., & Shaw, M. (May 2000) An Assessment of Software Engineering Body of Knowledge Efforts. <http://www.acm.org/>
12. Modesitt, Kenneth L. (February 2002) "International Software Engineering University Consortium (ISEUC): A Glimpse into the Future of University and Industry Collaboration" In Proceedings of the Fifteenth Conference on Software Engineering Education & Training, pp. 32-41, IEEE Computer Society, Los Alamitos, California.

13. Modesitt, Kenneth L., Bagert, Don, & Werth, Laurie (April 2001) "Academic Software Engineering: What Is and What Could Be? Results of the First Annual Survey for International SE Programs" In Proceedings of the 23rd International Conference on Software Engineering.
14. Parnas, D. L. (October 2-8, 2001) " Do You Have a License to Drive That Mouse?" Ubiquity. Interviews, Volume 2, Issue 30. http://www.acm.org/ubiquity/interviews/d_parnas_1.html.
15. Thomas, B. B., & Duggins, S. L. (April 2000) "The Evolutionary Changes of Undergraduate Software Engineering Curriculum" In Proceedings in the 2000 ASEE Southeastern Section Conference.
16. Tripp, L. & Frailey, D. J. (Feb. 2, 1999) IEEE Computer Society and ACM Software Engineering Coordinating Committee (SWECC) Overview.

BARBARA BERNAL THOMAS

Thomas is a full professor in the School of Computing and Software Engineering at Southern Polytechnic State University for the last seventeen years. The areas of Software Engineering, User-Centered Design and Computer Graphics & Multimedia are the focus endeavors. She is a co-founder of the SPSU Usability Research Lab and is directly involved in corporate-sponsor ULAB projects.

SHERYL L. DUGGINS

Sheryl Duggins is an associate professor in the Computer Science department of Southern Polytechnic State University in Marietta, Georgia. She has served as the Graduate Coordinator for Software Engineering and was integral in developing the Master of Science in Software Engineering degree. She obtained her Ph.D. degree in Computer Science from the University of Florida in 1991.