2023 Annual Conference & Exposition
Baltimore Convention Center, MD | June 25 - 28, 2023
The Harbor of Engineering Education for 130 Years
ASEE
Paper ID #38331

The State of the Practice Integrating Security in ABET Accredited Software Engineering Programs

Dr. Walter W. Schilling Jr., Milwaukee School of Engineering

Walter Schilling is a Professor in the Software Engineering program at the Milwaukee School of Engineering in Milwaukee, Wisconsin. He received his B.S.E.E. from Ohio Northern University and M.S. and Ph.D. from the University of Toledo. He worked for Ford Motor Company and Visteon as an Embedded Software Engineer for several years prior to returning for doctoral work. He has spent time at NASA Glenn Research Center in Cleveland, Ohio, and consulted for multiple embedded systems companies in the Midwest. In addition to one U.S. patent, Schilling has numerous publications in refereed international conferences and other journals. He received the Ohio Space Grant Consortium Doctoral Fellowship and has received awards from the IEEE Southeastern Michigan and IEEE Toledo Sections. He is a member of IEEE, IEEE Computer Society and ASEE. At MSOE, he coordinates courses in software verification, real time systems, operating systems, and cybersecurity topics.

The State of the Practice Integrating Security in ABET Accredited Software Engineering Programs

Abstract: Within the software engineering discipline, concerns related to security continue to grow. Since the early 2000's, the number of cyberattacks against deployed software systems has significantly grown. In 2014, recognizing this concern, a modification to the ABET EAC program accreditation criteria for software engineering was made, explicitly requiring topical coverage of security for accredited programs. Since taking effect in 2016, all programs in software engineering have been required to demonstrate appropriate coverage of the topic as part of the accreditation process. While the criteria requires that the topic of security be covered, the implementation has been left open to individual programs.

This article serves two purposes. First and foremost, it provides an updated status on the demographics of accredited software engineering programs. In doing so, it also provides a snapshot of the state of the practice of how security is integrated into program curricula by analyzing the 37 domestic ABET accredited bachelors' programs in software engineering. The article will identify at a high level the topics that are covered in the programs, as well as provide an overview of other aspects of the institutions which impact the depth and breadth of security coverage available to undergraduate students.

Introduction

From the academic standpoint, the first software programs began at the graduate level in the 1990's, with the target being students who had undergraduate computing degrees but desired additional study in the discipline. Over time, Software Engineering concepts began to trickle down into the undergraduate curriculums, typically in the Computer Science or Computer Engineering areas. In 1995, ISO/IEC 12207 [1] was published, providing a baseline for the discipline. The concept of a software engineering major was first put forth in 1997 [2]. Work then began on a set of guidelines for the development of an undergraduate curriculum [3].

In 1996, Rochester Institute of Technology admitted the first students into its program [4], which then led in 2001 to the first programs receiving ABET accreditation using the program criterion of Figure 1. As would be expected, evolution within the discipline continued, including the publication of the first book of knowledge for software engineering (SWEBOK) in 2004 [5] as well as the Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering [6], also in 2004.

PROGRAM CRITERIA FOR
SOFTWARE
AND SIMILARLY NAMED ENGINEERING PROGRAMS
Lead Society: CSAB
Cooperating Society: Institute of Electrical and Electronics Engineers

These program criteria apply to engineering programs which include software or similar modifiers in their titles.

1. Curriculum

The curriculum must provide both breadth and depth across the range of engineering and computer science topics implied by the title and objectives of the program.

The program must demonstrate that graduates have: the ability to analyze, design, verify, validate, implement, apply, and maintain software systems; the ability to appropriately apply discrete mathematics, probability and statistics, and relevant topics in computer science and supporting disciplines to complex software systems; and the ability to work in one or more significant application domains.

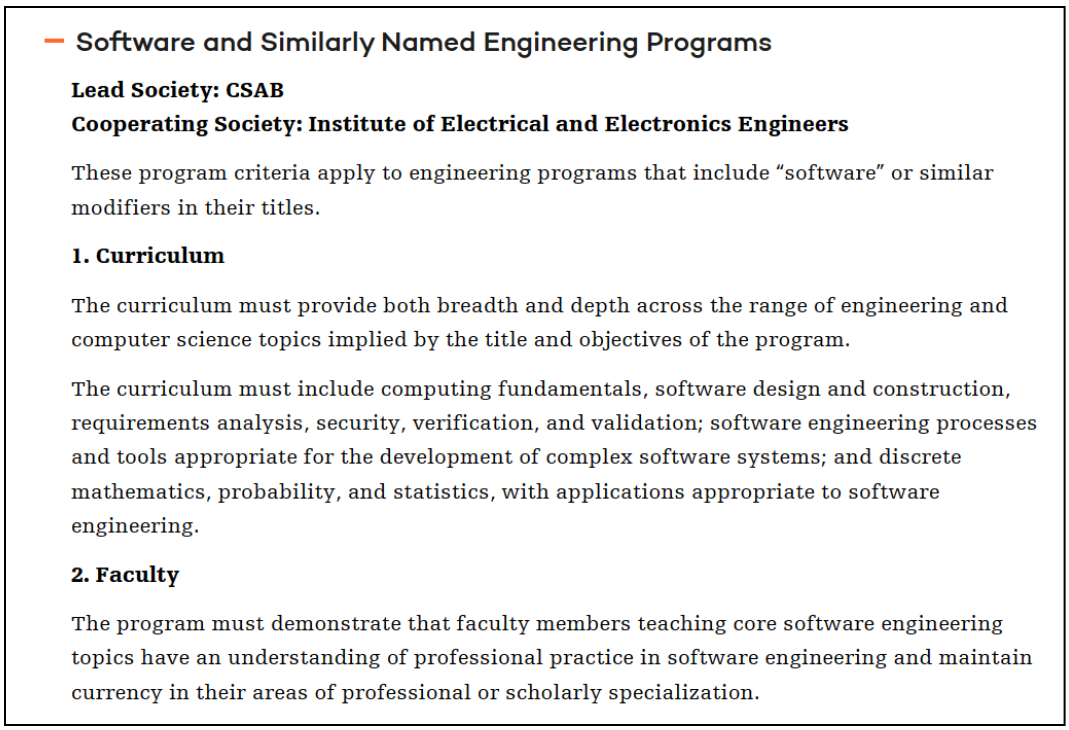
2. Faculty

The program shall demonstrate that those faculty teaching core software engineering material have practical software engineering experience.

Figure 1 2003 ABET Criteria for software engineering [25]

While many of the initial software engineering programs were similar, each program had a distinctive institutional history that shaped the program. These characteristics were documented extensively in 2005 [7] , 2010 [8], and 2011 [9]. Since these comparisons were made, many changes have occurred within the discipline, both on the academic and practicing fronts. In 2014, the SWEBOK [10] was revised, resulting in a new publication incorporating more current practices and knowledge. The undergraduate curriculum guide (SE2014) [11] was also revised in 2015. One common key addition to these documents was the inclusion of a separate topic for security. While security has always been an aspect of software engineering, the importance of security has increased dramatically.

The increased need for security coverage within software engineering led to changes in ABET criteria. In 2015, a proposal [12] was made to modify the criteria for software engineering, removing specifics about students being prepared to work in one or more application domains and specifically requiring coverage of security within the curriculum. This change was subsequently adopted for the 2016-2017 criteria, resulting in the criteria shown in Figure 2



— Software and Similarly Named Engineering Programs

Lead Society: CSAB
Cooperating Society: Institute of Electrical and Electronics Engineers

These program criteria apply to engineering programs that include "software" or similar modifiers in their titles.

1. Curriculum

The curriculum must provide both breadth and depth across the range of engineering and computer science topics implied by the title and objectives of the program.

The curriculum must include computing fundamentals, software design and construction, requirements analysis, security, verification, and validation; software engineering processes and tools appropriate for the development of complex software systems; and discrete mathematics, probability, and statistics, with applications appropriate to software engineering.

2. Faculty

The program must demonstrate that faculty members teaching core software engineering topics have an understanding of professional practice in software engineering and maintain currency in their areas of professional or scholarly specialization.

Figure 2 2016-2017 ABET Criteria for software engineering [13]

These changes, both in criteria and technology, mean that it is time again to perform a comparison study on accredited software engineering programs. The goal is to determine how the state of the academic discipline has changed, as well as to see how security has been integrated into software engineering programs.

Methodology and Study

The modification to the ABET Software Engineering Program criteria [12] ultimately required all programs to revise their programs. The structure of the change was not prescriptive: programs were free to add security into their program in the manner best deemed suitable to their environment. This could be done by modifying existing courses, course descriptions, and outcomes to address security. This could be done by requiring students to take an existing security course from another program (i.e. Computer Science, MIS, etc.). It could be accomplished by adding a new course to the curriculum specifically to address security. With any of these methods, however, it should be visible to the public how the program integrated security through catalog entries.

Many factors potentially would impact this decision. In some cases, there are several shared courses with an associated computer science program. The CAC ABET criteria for Computer Science was revised in 2019 [14] to specifically call out the that the curriculum must include "Principles and practices for secure computing." In these cases, a single shared course for both software engineers and computer

science majors might have been the logical approach. Or, these programs might define specific course outcomes related to security and embed them in multiple shared courses as course topics or course outcomes. This again would be acceptable. At other institutions, there is an associated National Center of Academic Excellence in Cybersecurity program in Cyber Operations (CAE-CO). These institutions would likely adopt one or more courses from the Cyber Operations program to address security. Still other institutions would either create a new course in security tailed to software engineering (i.e. DevSecOps or Developing Secure Software).

To analyze how programs have incorporated security, a data set describing all accredited programs was created. For each accredited software engineering program, the Carnegie Classification and other institutional data was recorded. From the ABET annual enrollment summary data published on the program's website, information about the number of graduates and total number of enrolled undergraduate students was obtained. The program itself was then reviewed, capturing the number of credits in the program, and how each of the major portions of the program criteria was met. To determine this, the catalog was reviewed to identify where the topics of software design and construction, requirements analysis, security, and verification and validation were covered. For each of these areas, the topics covered in the curriculum were also noted to identify common topics. Lastly, it was determined if the institution also held an academic center of excellence distinction for cybersecurity.

Program Composition Findings

One of the first questions to ask is how the number of programs has changed. In 2010, there were 18 accredited programs in software engineering. Most of these programs had been accredited in 2005 or earlier, as between 2006 and 2009 there were only single new programs accredited each year. This changed in 2014, and there has been an increase in the number of new accredited programs since then, as is shown in Figure 3. However, it is unclear what the impact of the Covid-19 pandemic will be upon this growth, as new accreditation visits were hampered by the pandemic.

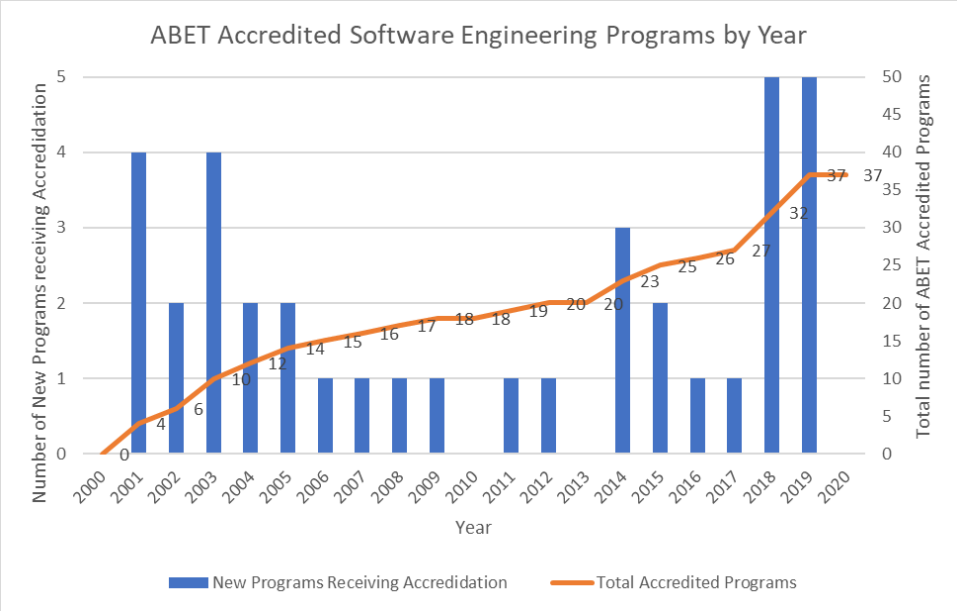


Figure 3 The number of new and existing ABET accredited software engineering programs by year. [15]

In terms of university classifications, there have been some changes in program demographics. Prior to 2010, there was a roughly even mix of public and private universities receiving their initial accreditation. However, as is shown in Figure 4. since 2010, most new programs receiving accreditation have been housed in public universities. When looking at the Carnegie Classification of institutions, the demographics have again remained largely unchanged, with the largest number of new programs developing at schools in the M1 classification, as is shown in Figure 5.

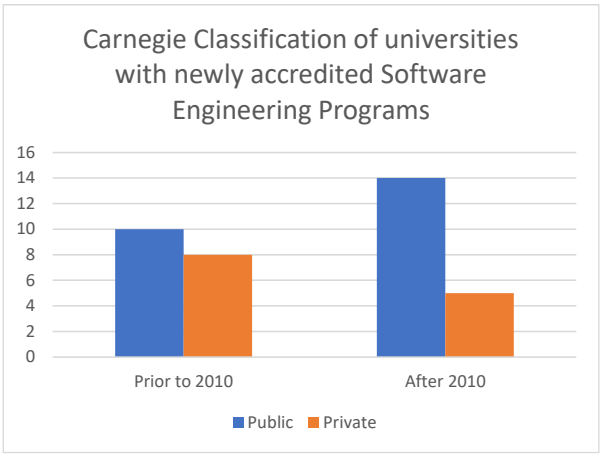


Figure 4 Type of university receiving accreditation.

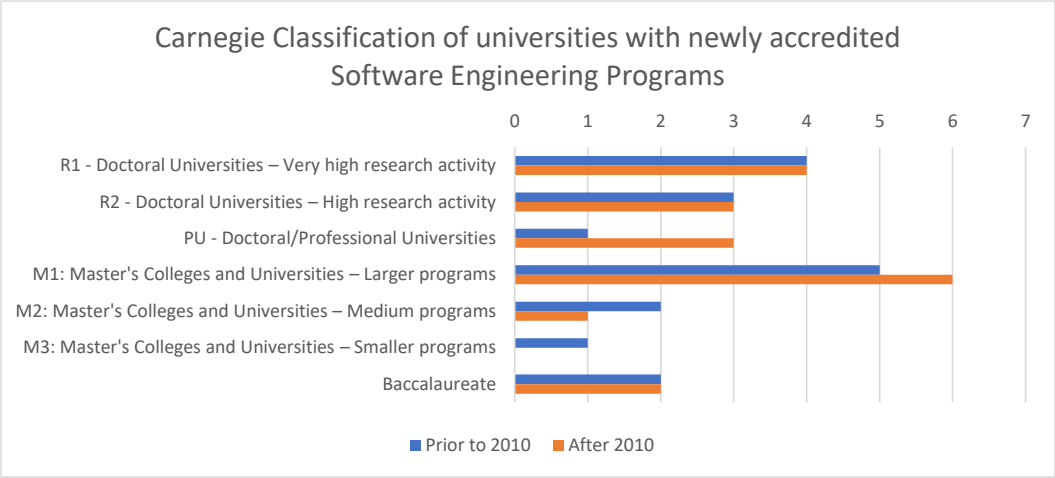


Figure 5 Carnegie Classification for new Software Engineering programs.

Geographically, there is a slight expansion. While the initial programs were predominantly located in the East or Midwest, newer programs are being established in the western part of the county, as is shown in Figure 6.

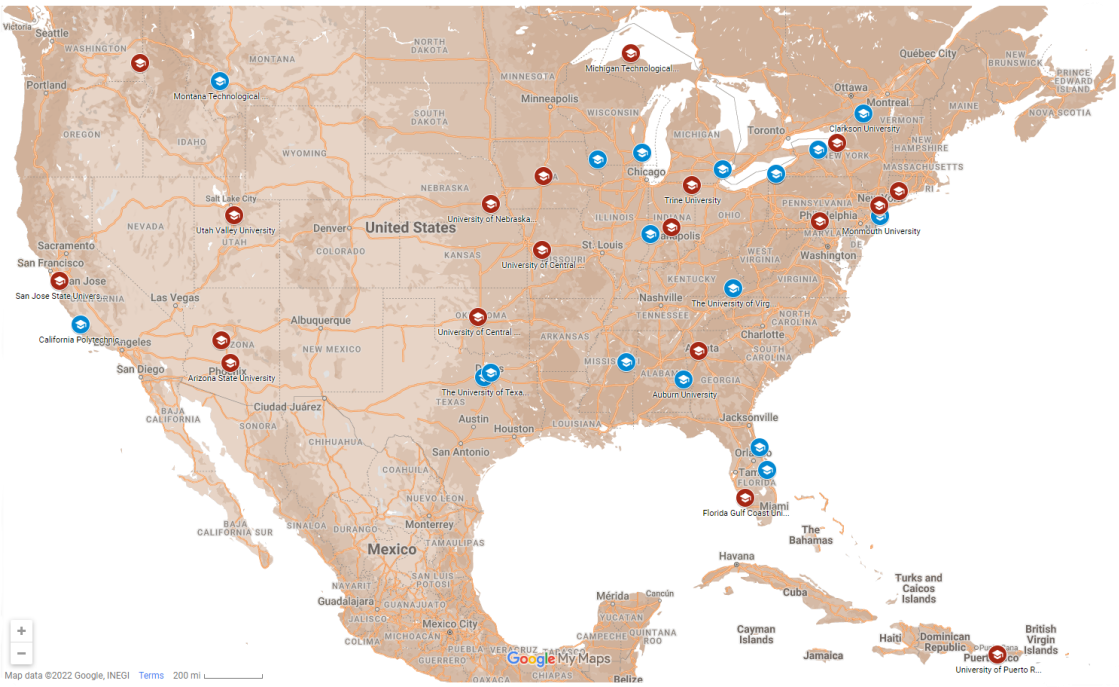


Figure 6 Location of ABET Accredited Software Engineering Programs in the United States. Red Programs have been added since 2010.

When looking at programs, all except for three universities were using the semester academic term. Required credit hours for the program ranged between 114 and 153, with a median value of 124 credit hours. Schools with the Carnegie classification of Baccalaureate tended to have slightly larger programs, with the median program requiring 127 credit hours to complete versus 124 for R1 and R2 universities.

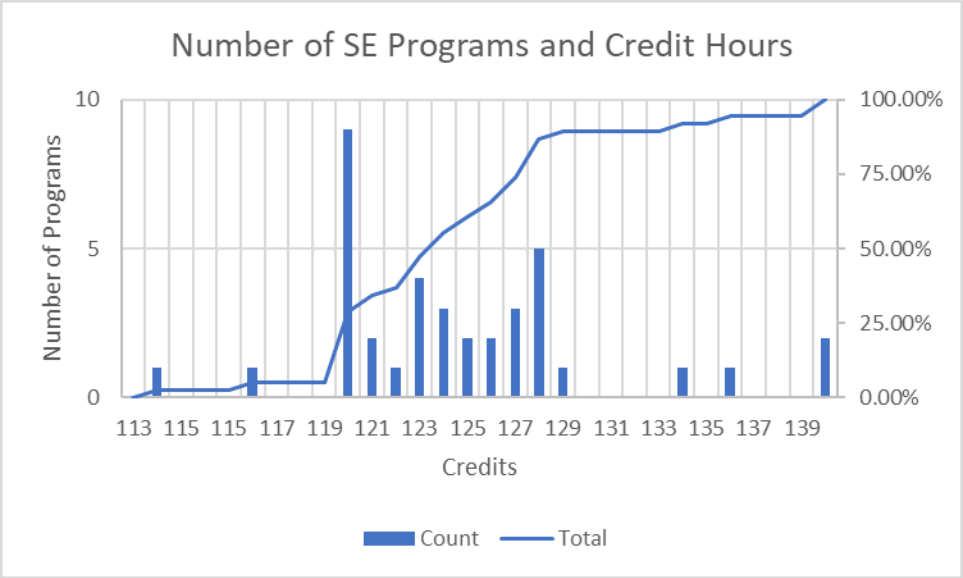


Figure 7 Credit Hours in Programs

Due to the methodology of collecting the data, it is not possible to know exactly how many graduates there were in any given year or how many students were enrolled in any given year, as the data recorded was the most recent information published on the program's website. But, combining data from all programs from the most recent year reported, there were a total of 8497 students enrolled in a software engineering program and 1394 graduates. This approximation indicates that software engineering may be slightly larger than Metallurgical and Materials Engineering based upon 2019 ASEE estimates. [16]

Within the accredited programs, there were vast differences in the number of students enrolled in software engineering and the number of annual graduates. The smallest program enrolled just 12 students while the largest program had 1706 students, as is shown in Figure 8. The number of graduates varied similarly, from the smallest value of 1 to the largest value of 175. Overall, the median program enrollment was 117 and the median number of graduates is 21.5.

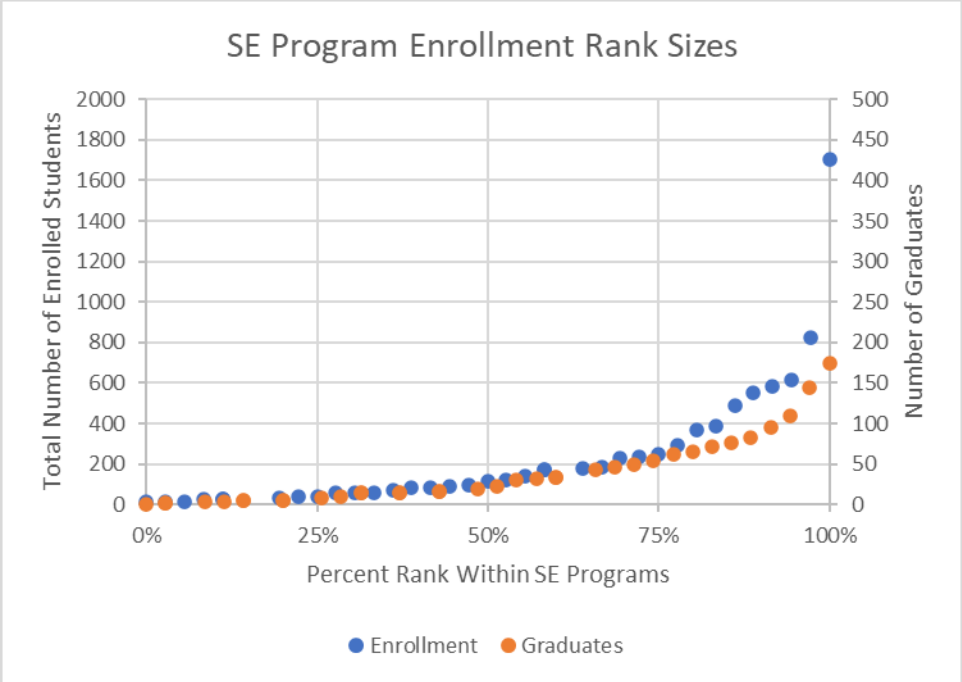


Figure 8 Enrollment and graduate sizes, collected from individual program's websites and ABET enrollment information.

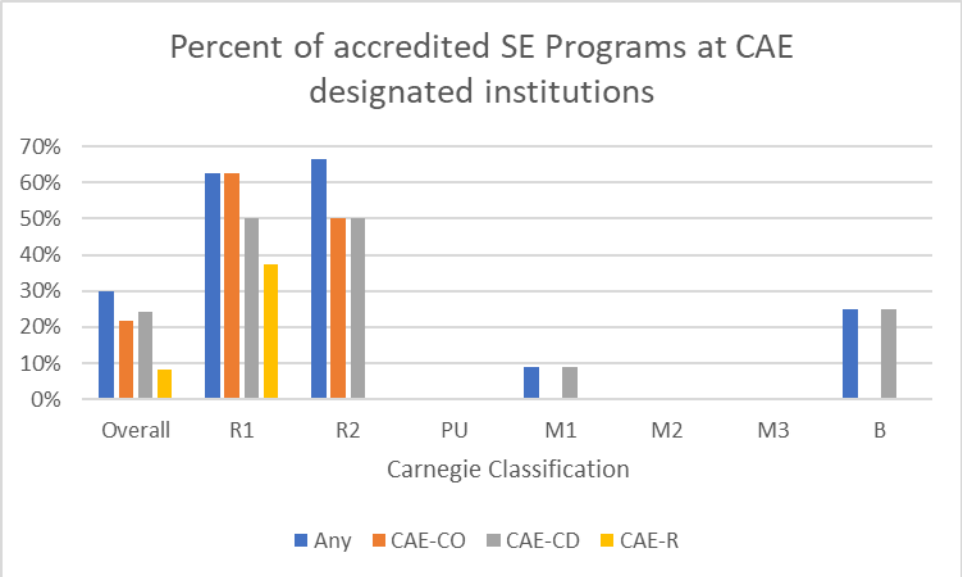


Figure 9 Accredited Software Engineering Programs at CAE-CO and CAE-CD designated institutions

Another interesting analysis was to look at the relationship between accredited software engineering institutions and National Centers of Academic Excellence in Cybersecurity (NCAE-C) designees, specifically the CAE Cyber Operations (CAE-CO) and CAE-Cyber Defense (CAE-CD) designations. While

cybersecurity has a separate set of accreditation criterion and is a separate discipline of engineering, it is closely related to software engineering. A center of excellence designation could certainly be an advantage for a software engineering program, as there would be greater faculty expertise within the area as well as a richer set of courses that could potentially be used to teach software engineering majors cybersecurity material. Overall, it was very common for the schools with R1 and R2 Carnegie Classifications to also be a designated cybersecurity center of excellence but was quite rare for other classifications.

Security Analysis

Since 2010, there have been numerous significant changes within the discipline. The debut of the iPhone in 2007 and the Android operating system in 2008 changed the domain for many software developments projects. 2009 saw the birth of DevOps practices with the famous presentation *10+ Deploys Per Day: Dev and Ops Cooperation at Flickr* [17]. Improvements in technology led to increasing virtualization being employed in software engineering, and cloud technology has become more ubiquitous within software architectural design.

The period since 2010 has been marked by a significant increase in security related concerns, as is shown in Figure 10. While security has always been a part of software engineering, the growth in vulnerabilities has increased its importance to the discipline. In preparing the SE2014 curriculum guide [11], feedback from constituents received by the taskforce indicated that security had now reached a level of importance that it be specifically acknowledged within the guidelines [18].

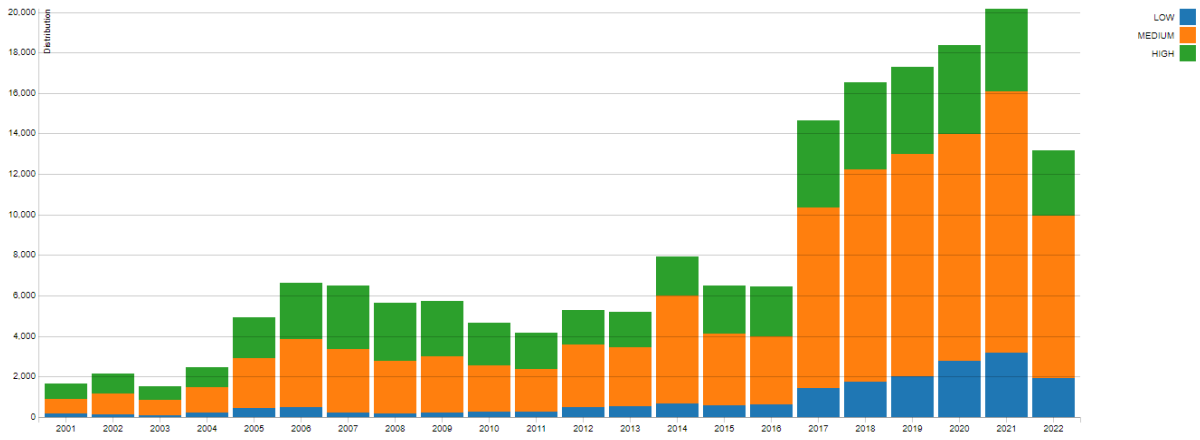


Figure 10 The change in the number and severity of vulnerabilities over time [19]

One of the key questions for this study was to determine the coverage of security within the software engineering programs. To do this, the catalog entries for each accredited program were analyzed. If, in reviewing the catalog, a course title clearly indicated it was a security related course or if the catalog description clearly indicated that the course was targeted at security, the program was marked as having a required security course and the course description was captured for further analysis. However, if no course clearly was focused on security, the catalog entries for all courses were reviewed to try and identify security topics within the program. If one or more course descriptions clearly indicated that they were security topics, those topics were noted, and the program was marked as having topical coverage within existing courses. By logging this information, it was hoped that an understanding of security coverage could be obtained. This exercise was then repeated for the topics of software design and construction, requirements analysis, and verification and validation. The intent was to determine if there was a fundamental difference in how the programs managed security, which is relatively new to the criterion, versus the other topics, which were generally present in previous criterion.

Figure 11 shows the high-level results of this study. Overall, for the verification and the design and construction topics, most programs have one or more courses that specifically target this aspect of software engineering. For the verification area, sample course titles included “Software Testing and Quality Assurance”, “Software Testing and Quality Engineering”, and “Software Verification, Validation and Testing”. In the design and construction area, sample course titles included “Software Engineering: Design and Process”, “Software Construction”, “Software Design and Architecture”, and “Crafting Quality Code”.

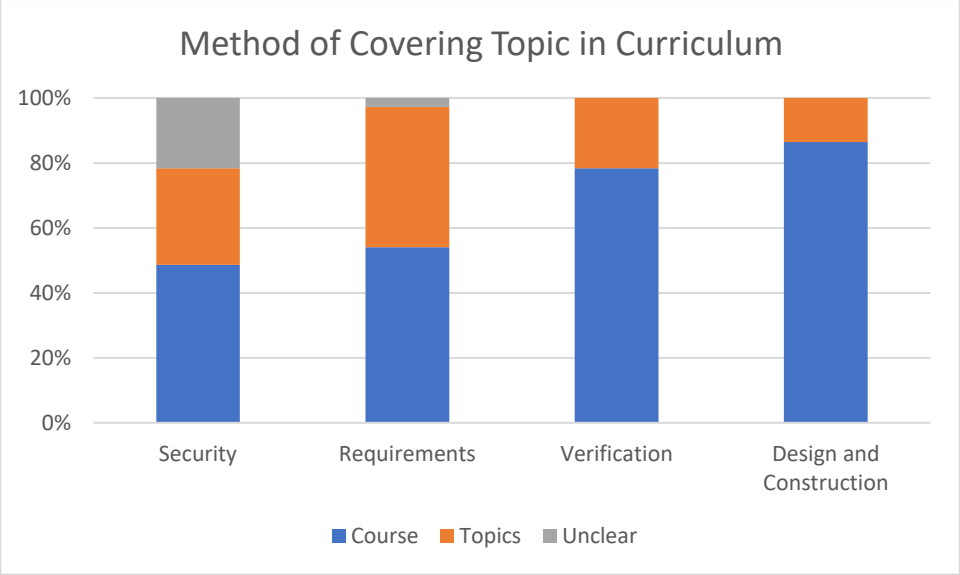


Figure 11 The techniques for covering topics in SE curricula.

For the requirements area, it was slightly less common for programs to have a single required course focusing on this area of software engineering. While 53% did follow this model, with course titles including “Requirements Engineering and Specification”, “Full Stack Development 1: Software Requirements Analysis”, “Software and Safety Requirements Engineering”, and “Requirements Elicitation, Modeling, and Analysis”, many of the other programs embedded requirements concepts into other courses. This may have been a project-based course or a general introduction to software engineering with a more substantial focus on requirements. In all but one case, requirements were clearly mentioned in one or more courses as a topic outside of the capstone design experience.

However, the security area is somewhat concerning. In reviewing the course descriptions, 49% of programs clearly required a course which focused on security. Course titles, seemed to be significantly more varied, ranging from courses which clearly targeted security in the software engineering domain, such as “Engineering Secure Software Systems”, “Software Analysis and Verification for Safety and Security”, “DevOps Principles and Practices”, and “Secure Programming”, to courses which were clearly not as targeted to the discipline, such as “Introduction to Cybersecurity”, “Data Privacy and Security”, and “Fundamentals of Cyber Security and Cryptography”. Thirty percent of programs had security embedded in one or more courses with one or more clear topics related to security inside of a class serving a different purpose. For the remaining programs, however, no evidence of security coverage could be obtained from catalog entries. This last finding is concerning.

The analysis was then performed breaking programs down by Carnegie Classification, as is show in Figure 12. Overall, while there are slight differences across classifications, the pattern is remarkably similar, with most programs requiring specific courses in design, construction, and verification and less in the security and requirements areas.

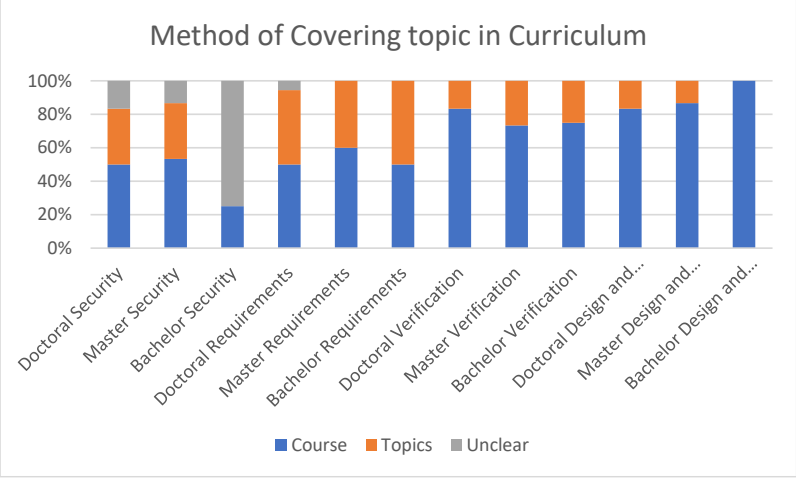


Figure 12 The techniques for covering topics in SE curricula broken out by Carnegie Classification.

With the course descriptions now being collected for each area, we wanted to look at what topics were included in these courses. To do this, the course descriptions were converted into word clouds, with the most common topics appearing the largest, and these word clouds were then compared with the topics in the SE2014 document [11].

Figure 13 is the word cloud for Verification. Fifty-five unique words were identified for this cloud, with occurrences between 2 and 37 times in the 20 course descriptions. Overall, there is a high degree of overlap between the topics listed in the SE2014 document and the course topics included in courses within the verification area. There are some areas that are not necessarily called out in the course descriptions, such as "user interface testing" and "testing based upon operational profiles", but the majority of topics seem to be in program verification courses. Additionally, many of the key terms are consistent and appear in multiple courses as well as across institutions, which is a positive sign.

Figure 13 Word cloud for Verification Related Course Descriptions

The same analysis was performed on the requirements courses (Figure 14) and design and construction courses (Figure 15). The requirements cloud contained thirty-one unique words which occurred between 2 and 33 times in the twenty course descriptions for requirements courses. The design and construction cloud contained forty-nine words which occurred between 3 and 32 times in the course descriptions for the 32 programs which specifically identified design and construction courses. As with the verification topic, many of the keywords were extremely common in courses at different institutions, and overall, there was a strong overlap with the SE2014 document. There were some outliers or terms which only occurred in one program, but these were rare.

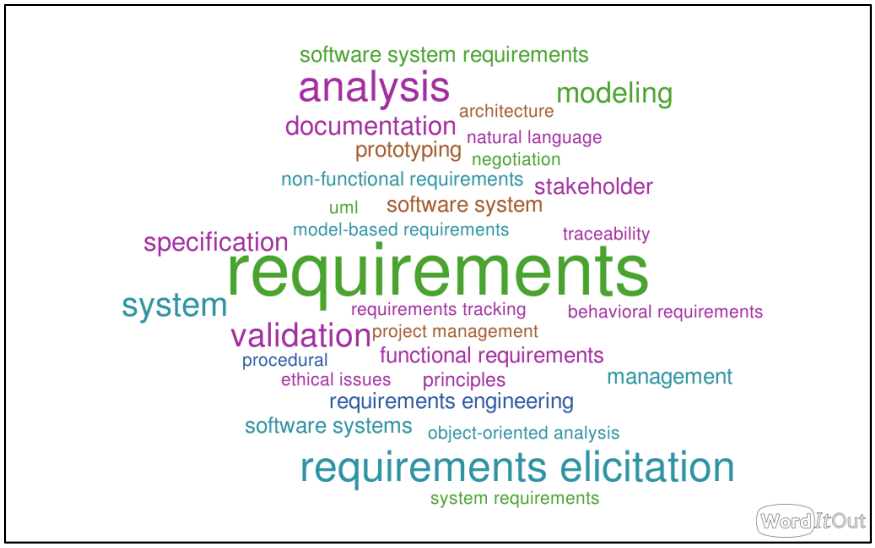


Figure 14 Word Cloud for Requirements

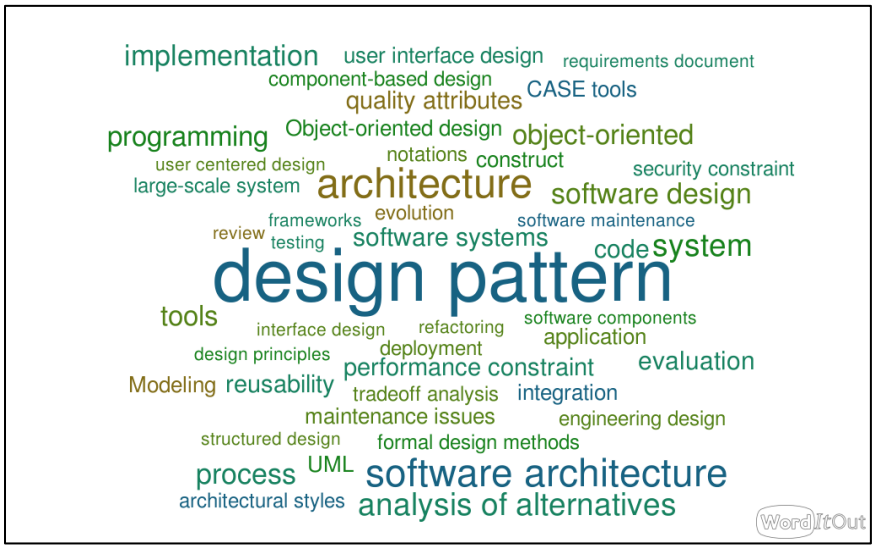


Figure 15 Word Cloud for Design and Construction

The security word cloud, shown in Figure 16, is quite different from the other clouds. Overall, the cloud contained 38 words which occurred between 2 and 9 times in the 18 course descriptions. Versus the other clouds, there were just as many words, but the words occurred less often. Furthermore, when compared with the SE2014 document and the recommended security topics, shown in Figure 17, there were many more topics which are not readily apparent in the course descriptions. This does not necessarily mean that these topics are not covered, but the course catalog entries do not reflect coverage of these topics. Based on this analysis, security coverage is not as consistent as it is with other topics in the curriculum.

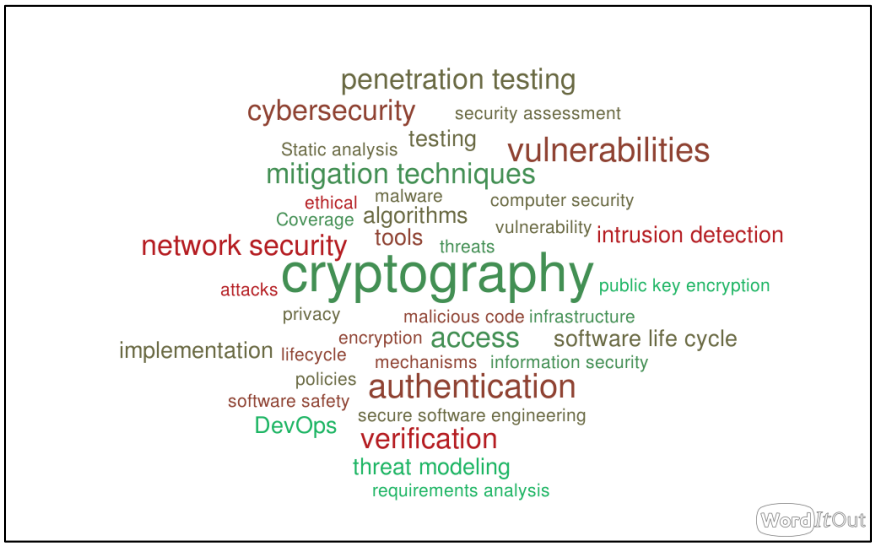


Figure 16 Security Word Cloud

Reference		k,c,a	E,D	Hours
SEC	**Security**			20
SEC.sfd	Security fundamentals			4
SEC.sfd.1	Information assurance concepts (confidentiality, integrity, and availability)	k	E	
SEC.sfd.2	Nature of threats (e.g., natural, intentional, and accidental)	k	E	
SEC.sfd.3	Encryption, digital signatures, message authentication, and hash functions	c	E	
SEC.sfd.4	Common cryptographic protocols (applications, strengths, and weaknesses)	c	E	
SEC.sfd.5	Nontechnical security issues (e.g., social engineering)	c	E	
SEC.net	Computer and network security			8
SEC.net.1	Network security threats and attacks	k	E	
SEC.net.2	Use of cryptography for network security	k	E	
SEC.net.3	Protection and defense mechanisms and tools	c	E	
SEC.dev	Developing secure software			8
SEC.dev.1	Building security into the software development life cycle	c	E	
SEC.dev.2	Security in requirements analysis and specification	a	E	
SEC.dev.3	Secure design principles and patterns	a	E	
SEC.dev.4	Secure software construction techniques	a	E	
SEC.dev.5	Security-related verification and validation	a	E	

Figure 17 SE2014 units and topics for security. [11]

Overall Analysis

Overall, in reviewing existing accredited programs, the security component is slightly less developed and consistent than other areas of the programs. In a considerable number of programs, it is not possible to identify the topics related to security covered within the program from the course catalog entries. This is not surprising, for this is the newest core area to be required within the criteria and its importance has grown significantly in the last decade. The security area also has undergone more change than other topics within the discipline.

At the present time, the IEEE SWEBOK is undergoing revision [20]. This new version includes sections specifically dedicated to software architecture, software engineering operations, and software security. Once released, this document will provide a new baseline for topical knowledge that software engineering programs should consider including within their programs for security. Other publications, such as The Cyber Security Body of Knowledge [21] include updated coverage of software engineering topics related to security, and the newly released IEEE standard for DevOps [22] has material which ultimately needs to be considered for inclusion in software engineering programs. And it is likely that the release of the revised SWEBOK will eventually result in a new guide for undergraduate software engineering curriculums.

Ultimately, as the area of security standardizes, more programs may institute a course specifically related to secure software development. This may take the form of DevSecOps, as has already occurred at some institutions, or it may be in the form of a secure design and implementation course.

Given the importance of security to Software Engineering, it is imperative that the programs continue to improve teachings in this area. This review, while thorough in some regards, was also very limited in that it only could look at catalog entries and curricula. In evaluating programs, evaluators review a more thorough self-study which may better reflect the teaching of security within the program. This may be a pessimistic view of the situation, as the catalog entries may not be entirely current due to institutional policies and individual instructors may include topics that are not explicitly called out in the catalog. What is concerning, however, is that these limitations do not appear to be present for other core topics, such as verification, construction, or requirements, which does truly indicate that there may be a more significant gap within programs.

Bibliography

[1] ISO/IEC 12207: Information technology - Software life cycle processes, International Organization for Standardization , 1995.

[2] T. Hilburn, "Software engineering education: a modest proposal," *IEEE Software,* pp. 44-48, November 1997.

[3] D. Bagert, T. Hilburn, G. Hislop and S. Mengel, "Guidelines for software education: meeting the needs of the 21st Century," in *28th Annual Frontiers in Education Conference. Moving from 'Teacher-Centered' to 'Learner-Centered' Education.*, Tempe, 1998.

[4] J. Naveda, D. Bagert, S. Seldman, J. Armarego, T. Hilburn and S. Eisenbach, "Developing an undergraduate software engineering degree," in *Proceedings 16th Conference on Software Engineering Education and Training*, Madrid, 2003.

[5] A. Abran, J. Moore , P. Bourque and R. Dupuis , "Guide to the Software Engineering Body of Knowledge (SWEBOK)," The Institute of Electrical and Electronic Engineers, Inc., Piscataway, 2004.

[6] Joint Task Force on Computing Curricula, "Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering," IEEE Computer Society , 2004.

[7] B. Bernal, "Reviews Of Curriculum Guides For Professional Software Engineers," in *ASEE Annual Conference*, Portland, 2005.

[8] S. Conry, "Software Engineering: Where Do Curricula Stand Today?," in *2010 ASEE Annual Conference & Exposition*, Louisville, 2010.

[9] S. Conry, "Software Engineering, Computer Engineering, Computer Science: Sibling Disciplines with Diverse Cultures," in *ASEE Annual Conference & Exposition*, Vancouver, BC, 2011.

[10] IEEE Computer Society, "Guide to the Software Engineering Body of Knowledge (SWEBOK Version 3)," IEEE Computer Society, 2014.

[11] Joint Task Force on Computing Curricula, "Software Engineering 2014 Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering," IEEE Computer Society, 2015.

[12] ABET, *2015-2016 Criteria for Accrediting Engineering Programs,* Baltimore: ABET, 2014.

[13] ABET, *Criteria for Accrediting Engineering Programs, 2016 – 2017,* Baltimore: ABET, 2015.

[14] ABET, *Criteria for Accrediting Computing Programs, 2019 – 2020,* Baltimore: ABET, 2019.

[15] ABET, "ABET Accredited Program Search," ABET, [Online]. Available: https://amspub.abet.org/aps/name-search?searchType=institution. [Accessed 2022 December].

[16] "Engineering and Enghineering Technology by the Numbers," ASEE, Washington, DC, 2020.

[17] J. Allspaw and P. Hammond, "10+ Deploys Per Day: Dev and Ops Cooperation at Flickr," in *Velocity*, San Jose, 2009.

[18] M. Ardis, D. Budgen, G. W. Hislop, J. Offutt, M. Sebern and W. Visser, "SE 2014: Curriculum Gudelines for Undergraduate Degree Programs in Software Engineering," *IEEE Computer,* vol. 48, no. 11, pp. 106-109, 2015.

[19] "NVD - CVSS Severity Distribution Over Time," NIST, [Online]. Available: https://nvd.nist.gov/general/visualizations/vulnerability-visualizations/cvss-severity-distribution-over-time#CVSSSeverityOverTime. [Accessed 2 December 2022].

[20] "SWEBOK Evolution," IEEE Computer Society, 9 January 2023. [Online]. Available: https://www.computer.org/volunteering/boards-and-committees/professional-educational-activities/software-engineering-committee/swebok-evolution. [Accessed 15 January 2023].

[21] A. Rishid, H. Chivers, E. Lupu, A. Martin and S. Schneider, The Cyber Security Body of Knowledge, London: National Cyber Security Center, 2021.

[22] IEEE, *IEEE Standard for DevOps: Building Reliable and Secure Systems Including Applications Build, Package, and Deployment,* New York: IEEE Standards, 2021.

[23] A. G. Oettinger, "President's Letter to the ACM Membership," *Communications of the ACM,* vol. 9, no. 8, pp. 545-546, 1966.

[24] NATO Science Committee, "Software Engineering," NATO, Brussels, 1968.

[25] ABET, *Criteria For Accrediting Engineering Programs,* 2003.