

Challenges in Academia in Producing Prepared IT Workforce

Nary Subramanian, University of Texas at Tyler

Abstract. A frequent question practitioner asks an academician is “Why don’t educational institutions prepare students better for work in the IT area?” Often cited problems with current graduates from Computer Science, Computer Information Systems, and other IT disciplines, include lack of awareness of latest developments in the field, lack of knowledge of applying systematic process to solving problems, and a general unpreparedness to cooperatively work in teams. These issues are pertinent in an increasingly interconnected world where new technologies are emerging at a fast pace: big data analytics, computer security, cloud computing, and mobile computing, are but a few of the latest developments. While it may be appropriate for industry to demand properly prepared students who are work-ready from day one, it may be an eye-opening experience for most practitioners to know the problems faced on the other side of the fence, namely, at educational institutions. In Texas, for example, less than 25% students graduate in four years and less than 50% graduate in six. Typical course-load for a degree in Computer Science area is about 120 hours with about 60 hours dedicated to state mandated, college- and department-specific core courses; this leaves about 60 hours (roughly, 20 courses) to complete the discipline-specific requirements. If we consider the normal course sequence in the degree plan consisting of traditional courses such as programming, software development, and database design at the most only about four courses are left in the plan that can be considered for electives. Assuming students are ready to take an elective in a topic such as SCADA security, the next problem is the lack of appropriate labs and textbooks to teach such courses at the undergraduate level. Another issue that needs to be considered is the availability of faculty members to teach new courses in a manner that not only retains students’ interest but also increases enrollment. Therefore, producing well-educated students who are workforce-ready for cutting-edge technology companies within four years of college study is proving to be an increasingly tougher goal for the academia. This article considers this issue and suggests possible solutions to this perceived problem.

Introduction

Undergraduate education builds a nation’s workforce. This is especially true in IT related fields such as Computer Science (CS), Computer Information Systems (CIS), Information Technology (IT) or Management Information System (MIS). Students graduating with B.S. are employed by (and increasingly are also employing) leading technology companies such as Google, Apple, Microsoft, public, and government agencies. They form the foundation that helps develop cutting-edge technologies that require and provide jobs to millions of people around the globe. In fact, CEO’s of several technology firms are graduates from four-year degree programs such as, for example, Jeff Bezos of Amazon, Mark Zuckerberg of Facebook, and Tim Cook of Apple [1].

However, recently I have been asked by practitioners at several workshops and conferences, “Why do not educational institutions prepare students better for work in the IT area?” In a recent survey by CompTIA, 93% of employers indicated there exists a skills gap in the IT workforce that included both hard

and soft skills - lack of knowledge of latest technologies, poor problem solving ability, and inability to work cooperatively in teams [2]. Industry people expect, perhaps rightly so, that products of four-year degree programs from Universities be ready to contribute effectively right from day one. However, considering the dynamic growth of the IT industry, many employers seem to have been disappointed in this expectation as indicated by the survey [2]. These issues are relevant in an increasingly interconnected world where new technologies are emerging at a fast pace: big data analytics, cloud computing, game development, social networking, and mobile computing, are but a few of the latest developments.

In this article, I wish to point out the issues on the “other” side: the academic side, a view that is usually not very clear from the outside world. Major reasons that I consider contributing to this observation from industry practitioners can be categorized in three headings: poor four-year graduation rate, low discipline-specific course load, and lack of academic infrastructure for modern electives. I also suggest some remedial measures that can help improve situation many of which require industry participation. In this article I discuss mainly from the CS perspective though similar arguments may be made for other computing disciplines.

Poor 4-year Graduation Rate

Typical four-year graduation rates are low in US universities. As per the National Center for Education Statistics [3], for the year 2006 cohort, only 39% graduated within four years while only about 59% graduated within six years. However, data varies from state to state: in Texas, for example, the four-year graduation rate is less than 25% while six-year graduation rate is less than 50%. In engineering disciplines, the graduation rates are slightly higher at about 35% while the six-year rate is about 75% [4, 5]. This means out of every 100 incoming freshmen only about 35 are available for industries to hire at the end of four years. In fact, in 2011 about 11,000 undergraduate CS degrees were awarded in the US [6] and given a growth in enrollment of about 10% per year (based on [6]), I estimate the number of students in the incoming freshmen cohort for CS should have been about 26,000 in 2007 – about 42% four-year graduation rate. This means more than half the enrolled students failed to complete their degrees within the expected period of four years. There are several reasons for this: changed majors, funding problems, family issues, job issues, lack of pre-requisites, or simply failed classes.

What does this mean for an employer? When students do not complete the curriculum within the expected duration, they graduate with obsolete skills: they may have forgotten concepts learnt during the sophomore Data Structures class or even the freshman programming class (for example, in the survey [2], employers state that one of the reasons for the skills gap is the fast pace of change in the IT industry). In junior and senior classes, a student graduating late may not remember the basics studied in the pre-requisite class some years earlier – this requires the student to study harder to keep up with the class. Therefore, the employer is likely to find a student taking longer to graduate not as good as a student who graduates on time. In fact one study has concluded that students graduating late tend to take the job they get – not necessarily the job they want! [7]

Low Discipline-specific Course Load

Typical course-load for a four-year CS program in Texas (for example, [8]) includes up to 48 hours of state-mandated core, 42 - 51 hours of required courses, 9 - 12 hours of CS electives, and 21-28 hours of other electives and pre-requisites, for a total of 120 - 130 credit hours of courses. This means that only about 60 hours (or roughly, 20 three-hour courses) are dedicated to CS in which programming fundamentals, algorithms, computer organization, operating systems, database management systems, networking, and software development all need to be fit in a proper sequence over the period of four years. CS electives are usually department-specific and application oriented: they include options that cover latest developments such as computer security, e-commerce, mobile programming, bio-informatics, and the like. However, for some of these electives students usually do not have the necessary pre-requisites (“pre-reqs”) and it becomes their responsibility to learn the pre-req along with the course material. For example, for a decent Computer Security class advanced math helps, for e-commerce advanced database concepts help, for iOS programming knowledge of Objective C (or, now, Swift) helps, and for bio-informatics a working knowledge of genetics helps. If such pre-reqs could be somehow built into the main CS curriculum, then students are more likely to graduate with a deeper understanding of the subject and its application areas (of course, students may optionally pursue a Master’s degree – however, there are cost and time penalties, and this is discussed later).

In this regard it should be pointed out that there is an upper limit on the number of credit hours a student can take and still pay the lower in-state tuition fees. For example, in Texas, an undergraduate student enrolled after 2006 can take only 30 additional hours beyond what is required for degree completion and still pay tuition at the resident rate; any hours taken in excess of these 30 hours will attract fees at the non-resident rate, which is, in many cases, at least twice that of the resident rate. This rule prompts students to complete their degrees as quickly as possible in terms of credit hours – therefore taking electives becomes an expensive option. These excess hours include any courses repeated due to failing grades or for grade replacement.

Another factor to remember is that not all electives are offered all the time. Based on departmental requirements such as faculty availability, course schedules, and student enrollment, some electives may not be offered on a regular basis. Also, if an elective is offered but there is not sufficient student enrollment, which is usually between 10 and 15 students based on the level of the course, the course may be cancelled at the discretion of the department. So electives may not be the right way to expect students to acquire knowledge of the latest developments in CS.

Lack of Academic Infrastructure for Cutting-edge Electives

I have been asked by industry practitioners as to why we are not teaching distributed control systems and SCADA (supervisory control and data acquisition systems) to our students. The control system industry has become extensively computerized and networked with several overlapping fields of knowledge and finding a student trained in these concepts has become increasingly difficult. SCADA is but one such emerging area; there are others such as

cloud computing, big data analytics, and health informatics, where deep interdisciplinary knowledge is required of graduates. In fact, emerging industry paradigm seems to be the convergence of IT and OT (operations technology) – OT is the technology used for the core business processes including manufacturing, customer-service, and product development. Unfortunately, there are several problems that academia faces when trying to incorporate such modern industry requirements into the curriculum.

First of all, we need excellent laboratory facilities for teaching courses in these areas. Simulators may be used for teaching cloud computing concepts and Hadoop has been used to teach big-data analysis: for proper operation of both these systems we need trained IT personnel who can maintain both the hardware and software for optimum use of such systems. However, for SCADA we need expensive hardware/software kits and well trained lab administrators to run such systems (for example, we have such a lab at UT Tyler [9]). Likewise, for health informatics we need to be able to simulate or emulate confidential health records and their use in medical practice.

Second problem relates to useful academic textbooks in these areas (this was a problem while teaching mobile computing in its early days as well [10]). While several industry references exist, not all of these can be easily used in the classroom since most such text books assume minimum knowledge in their readers which is always not the case. In SCADA design, for example, ladder diagrams are used by professionals but most academic curriculum in CS and Electrical Engineering have never taught these concepts for last many years!

Thirdly, we need well-trained faculty members to teach such courses. While industry practitioners may be called for guest lectures, it may not be fair to expect them to also teach such classes with the rigor required. With the availability of free webinars conducted by industry experts as well as MOOC (Massive Open Online Course) classes that are conducted by both academic and industry experts, current faculty could use these resources to prepare themselves to teach modern courses. Another option we considered at UT Tyler was the Professor-in-Residence where a faculty member spends a month or more in the summer working at a local company. Professor-in-Residence could be voluntary or paid; however, this experience will help reduce the barrier between industry and academia and hopefully prepare the faculty to become a better teacher.

Some Suggestions for Improving the Situation

One thought that may arise in the minds of the readers is the need for the state mandated core in CS curriculum – that is, can we not use the time spent on the core on CS subjects? In fact this is the approach followed in many other parts of the world where bulk (90% or more) of the CS curriculum consists of only CS-related courses. However, it is widely acknowledged that exposure to the humanities, arts, and social sciences is essential to improving soft skills such as communication with peers, respect and understanding for other points of view, and being creative and innovative at work [11]. Therefore, the liberal arts core is essential to developing a well-rounded graduate in CS.

There are other possibilities to improve student’s learning experience within the four-year curriculum and they are discussed below.

Internship and Cooperative Experience

Many universities actively encourage their students to participate in internship and cooperative programs. Such programs involve students working for a company and learning specific skills that complement their academic experience. Examples include website development, single sign-on using active directory, SAP configuration, system administration, firewall configuration, programming using C#, Java, and C++, and the like. Internship and cooperative programs may be done concurrently with other courses during Fall and Spring semesters or exclusively during summer or even during regular semesters. Several programs are paid and may be located far from the parent University and, very often, also give academic credits to the students. In fact, some Universities have student services departments that provide active guidance to students for such programs. Thus, students participating in internship and cooperative experiences are usually more knowledgeable about the work environment in IT, perform better in class, and are usually more prepared to enter the workforce after graduation.

Meaningful Capstone Projects

Several CS programs require their seniors to participate in a capstone project. Typically this provides an opportunity to experience almost first-hand the typical IT job environment. If these capstone projects are conducted in their true spirit, many valuable job skills such as project management and collaborative skills can be learnt from them. Such projects become more meaningful if they are conducted with industry partners since the latter can not only mentor student groups but also possibly provide them employment or at least references upon graduation. At UT Tyler, we have been conducting these projects with interdisciplinary teams composed of both CS and CIS students where each team does a significant project for a local industry partner. Each team manages its project and works cooperatively to achieve objectives of the industry client. We found that this experience has helped develop well-rounded graduates with skills beyond that taught in the classroom that can be readily applied in practice. However, this requires close collaboration between educational institutions and industry that in turn requires open-minded faculty members and industry partners willing to appreciate each other's capabilities and constraints.

Industry Sponsorship of Education

Another possibility that exists is industry sponsoring their employees for undergraduate education on either full-time or part-time basis. Armed forces provides this option as does National Security Agency [12] though in the latter model the student studies at a University of his or her choice with a commitment to work at NSA subsequent to graduation. Many private employers have similar education sponsorship scheme for their current employees; perhaps they can follow the NSA model and hire incoming freshmen with the commitment that students work at the company during summers and join the company upon graduation. Such a scheme will not only produce well-educated employees but also employees who are already knowledgeable in the skills and technology that the company requires.

Another way in which industry can get latest tools in the hands of students is to sponsor equipment for labs – hardware and software that companies make can be supplied to labs in colleges and universities. Faculty members can integrate such equipment into appropriate coursework. For example, SPEA America donated semiconductor-testing equipment to UT Tyler [13] and this donation helped establish a center that benefitted both faculty and students.

Increase Course Load with MOOC/online Classes

Another option is to increase the number of courses required to obtain a CS major degree – in fact, in many other countries the liberal arts core is not required and the curriculum consists mostly of CS courses. But here we hit another hurdle – sometimes the maximum course loads are mandated by the state. For example, in Texas, 120-hour limit has been imposed by the state and at UT Tyler we had to redesign our curriculum from 127 hours a few years back to 120 hours now. However, there are many programs that are ABET accredited [14] which mandates them to satisfy minimum set of requirements – such CS programs are given exemptions and usually have 6-10 hours more than the state mandate. However, industry has not always insisted on ABET accreditation for CS programs (in fact, there are only 18 ABET accredited CS programs in Texas [15] out of more than 100 programs offering this degree) and the extra expense on the additional courses could be a burden for many students.

Expenses for the additional courses can be offset by taking online courses offered by several junior colleges and universities. There are also MOOC courses from sites such as Coursera [16] and Udacity [17] that students can enroll for free and improve their skillset and knowledge. Such courses may be taken in the summer months when students are not enrolled in normal classes. Universities are exploring ways to provide credit for such online coursework, although much work still remains to be done to make this process smooth and reliable.

Summer Research Experience for Undergraduates

A valuable opportunity exists in the summer months for exposing undergraduate students to the latest developments in the field of CS – the summer research experience program conducted by several universities often funded by major federal agencies. During this summer program, undergraduate students work alongside graduate students on cutting-edge research under the supervision of a faculty member. Such programs can serve as platforms for introducing, analyzing, and evaluating latest developments in the field and getting students interested in them. Sometimes industry partners are also involved in such research programs in which case students are exposed to the latest developments in both academia and practice. Such research experience could lead to co-authored papers, conference attendance, and patent applications, all of which serve to widen the knowledge horizon of students.

Study Abroad

Another option that students at US Universities have to expand their horizon is the study abroad programs. Study abroad programs offer students the opportunity to study a subject of

their choice at an educational institution abroad and get credit for those courses transferred to their primary institution in the US. Students opting to study in non-English speaking countries either already know or learn the language before going abroad. However, students going abroad typically return with a broader understanding of the world and tend to be more understanding employees. Also, some universities abroad offer specialized courses that can be taken by study abroad program students and many of the courses may be offered in English itself. However, study abroad experience may be intensive in summers, can be expensive, and not all credits may be transferred to the parent University.

Encouraging Graduate Education

Currently very few of undergraduates go on to join grad school immediately after graduation – one estimate puts this number between 10% and 30% [18]. By joining grad school for earning an M.S. or Ph.D. degrees, students get to specialize in a field of their choice and become more useful to work in the industry. In fact graduate education is considered key to improve US competitiveness and innovation [19]. However, there are many issues to consider: typically this means students have to pay more to obtain a graduate degree; in many cases, graduate degree does not automatically mean better pay; students lose out on time to obtain valuable experience; sometimes industry does not require PhD or even MS students. For some grad programs field or work experience is preferred – such as for example, cyber forensics. Therefore, while grad programs may not be the first choice for many graduating seniors this may provide an avenue for some of them to acquire advanced knowledge required by the IT industry.

Closer Industry-Academia Relationship

At national and professional levels industry already has a say in academia through ABET accreditation and Professional Engineer (PE) exams [20]. ABET evaluates computing disciplines (CS, CIS, and IT) through its Computing Accreditation Commission (CAC) and if the programs satisfy minimum requirements they receive ABET accreditation. PE exam has been started recently for software engineering that assures the hiring company that the person possessing this certification satisfies minimum industry requirements. Therefore, industry input already has an impact on curriculum. However, industry practitioners can participate even more in the functioning of local colleges and universities by volunteering to become a member Industry Advisory Boards that most computing departments institute, attending senior design/capstone project presentations and providing feedback, delivering guest lectures, inviting faculty and students to visit their offices/factories, participating in career fairs, visiting colleges/universities on open days, and the like. Such involvement will not only provide an insight into the problems, if any, faced by local academia but also provides opportunities to interact with administration officials, faculty members, and students. These sessions can be used to observe, advise, and recommend to colleges/universities ideas for possible improvements.

Another form of industry-academia partnership is service-learning (SL) [21]. SL is an experiential learning wherein students learn by performing some service to a community client as part of course requirements and their learning is assessed by the instructor by means of reflection assignments. Within IT field, several of the initiatives suggested earlier including internships, cooperative education, capstone projects, and summer research experience could all be considered examples of SL. In order for SL to be more widely adopted in computing programs, industry may need to be more closely involved – for example, database classes may do a project for a company and networking classes may actually participate in creating new or troubleshooting existing networks in a company. At UT Tyler we have established the Center for Teaching Excellence and Innovation where SL is an essential component [22].

Conclusion

Industry seems to find several graduating seniors in the computing sciences field lacking in skills that will make them immediately productive in the industry. Typical problem areas seem to be the lack of awareness of latest developments in the field, lack of knowledge of applying a systematic process to solve problems, and a general inability to work cooperatively in teams. This article has explored this issue from an academic standpoint and discusses the constraints faced by the academic community in preparing its undergraduate students – these constraints can be categorized under poor four-year graduation rate, low discipline-specific course load, and lack of academic infrastructure for modern electives. These categories reflect the fact that only about 42% of freshmen graduate from computing programs in four years, that there are only about 20 courses or less in which to teach the core topics in CS, and that there are not enough labs, books, and faculty members to teach modern electives.

There are several avenues for improving the situation and these include internship and cooperative education experiences, meaningful capstone projects, industry sponsorship of undergraduate education, increase course load with MOOC and online classes, summer research experience for undergraduates, study abroad programs, and encouraging graduate education. The article discusses the pros and cons of these options. However, many of these initiatives require a closer cooperation between academia and industry and hopefully this helps reduce the skills-gap that industry seems to face with graduates from four-year programs in the United States.

Another issue that the IT industry faces is sufficient diversity in the workforce, which as per recent media reports, is lacking in terms of women and minority employment [23]. However, this problem exists in academia too, especially in the engineering and computer science disciplines where women and minority participation is low [24]. At UT Tyler, we conduct science camps for high-school girl students in the summer to encourage them to pursue STEM (science, technology, engineering, and math) careers. Another issue to consider is that a significant number (12% in 2012 as per [24]) of undergraduate students in Computer Science are part-time – such students usually take

longer to graduate [25]. Also many part-time students are trying improving their skills in their quest for lifelong learning and it is hoped that industry supports their endeavor [26]. However, it has been this author's experience that part-time students often contribute to a more mature discussion in the classroom.

In this article I have tried to answer the question that has been asked frequently by people in industry: "Why do not educational institutions prepare students better for work in the IT area?" There are several challenges that academia faces in trying to ensure all their "products" satisfy all industry expectations at the time of graduation and I have discussed many of these problems from the point-of-view of an educator and provided some directions in which industry could actively help in remedying this situation. Hopefully, the expectation gap steadily reduces in the future and students in the IT area find the transition to industry more seamless and enjoyable.

Acknowledgements

I thank the reviewer of the original version of this article for providing valuable feedback that helped to significantly improve the quality of this article. ♦

ABOUT THE AUTHOR



Nary (Narayanan) Subramanian is currently an Associate Professor of Computer Science at The University of Texas at Tyler, Tyler, Texas. Dr. Subramanian received his Ph.D. in Computer Science from The University of Texas at Dallas. His specialization is software engineering with particular focus on software architectures and requirements engineering. He co-founded the International Workshop on System/Software Architectures (IWSSA) and served as a co-chair for seven years between 2002 and 2011. He established and directed the Center for Petroleum Security Research at UT Tyler. He is a Fellow of Service Learning at UT Tyler's Center for Teaching Excellence and Innovation. He has over fifteen years' experience in industry in engineering, sales, and management. He is a member of the IEEE. His research interests include software engineering, system engineering, and security engineering.

Phone: 903-566-7309

E-mail: nsubramanian@uttyler.edu

REFERENCES

1. Gene Marks, "Why Do Nerds Become Successful CEOs?", February 10, 2014, available at <http://www.forbes.com/sites/quickerbetteertech/2014/02/10/why-do-nerds-become-successful-ceos/>
2. CompTIA, "State of IT Skills Gap: Full Report", February 2012, available at http://www.wired.com/wp-content/uploads/blogs/wiredenterprise/wp-content/uploads/2012/03/Report_-_CompTIA_IT_Skills_Gap_study_-_Full_Report.sflb_.pdf
3. http://nces.ed.gov/programs/digest/d13/tables/dt13_326.10.asp viewed on August 5th, 2014.
4. <http://www.utexas.edu/graduation-rates/documents/GRAD-REPORT.pdf>
5. <http://dars.tamu.edu/dars/files/ee/ee2a3d0b-38cf-4a0b-addc-847d802aee7b.pdf>
6. http://cra.org/uploads/documents/resources/taulbee/CS_Degree_and_Enrollment_Trends_2010-11.pdf
7. Carmen Aina and Giorgia Casalone, "Does time-to-degree matter? The effect of delayed graduation on employment and wages", AlmaLaurea Working Papers, No. 38, September 2011, available at <http://www2.alma laurea.it/universita/publicazioni/wp/pdf/wp38.pdf>
8. <https://www.uttyler.edu/catalog/degreeplans/files/COSCBS.pdf>
9. <https://www.uttyler.edu/cpsr>
10. S. K. S. Gupta and P. K. Srimani, "Experience in Teaching a Graduate Course in Mobile Computing", 30th Annual Conference on Frontiers in Education, October, 2000, pp. S1C6-S1C11 (Vol. 2).
11. Charles M. Vest, "Educating Engineers for 2020 and Beyond", National Academy of Engineering, <http://engineeringchallenges.org/cms/7126/7639.aspx> viewed on August 14th, 2014.
12. http://www.nsa.gov/careers/opportunities_4_u/students/stokes.shtml
13. <http://technews.tmcnet.com/business-video/news/2010/10/05/5049852.htm>
14. <http://www.abet.org>
15. <http://main.abet.org/aps/Accreditedprogramsearch.aspx>
16. <http://www.coursera.org>
17. <http://www.udacity.com>
18. <http://www.idealists.org/info/GradEducation/Resources/Preparing/ReasonsToWait>
19. http://www.cpec.ca.gov/CompleteReports/ExternalDocuments/GR_GradEdAmComp_0407.pdf
20. <http://ncees.org/exams/pe-exam/>
21. Barbara Jacoby, *Service Learning in Higher Education: Concepts and Practices*, Jossey-Bass, 1996
22. <http://www.uttyler.edu/ctei>
23. Elizabeth Weise, "Tech: Where the women and minorities aren't", USA Today, August 15, 2014, available at <http://www.usatoday.com/story/tech/2014/05/29/silicon-valley-tech-diversity-hiring-women-minorities/9735713/>
24. Brian L. Yoder, "Engineering by the Numbers", available at <http://www.asee.org/papers-and-publications/publications/11-47.pdf>
25. Jonathan Dame, "Part-time students least likely to graduate on time", USA Today, December 21, 2013, available at <http://www.usatoday.com/story/news/nation/2013/12/21/part-time-students-graduation-rate/4156239/>
26. David Kember, et al., "How students cope with part-time study", *Active Learning in Higher Education*, Vol. 6, Issue 3, 2005, pp. 230 - 242.