The Changing IT Curricula

Mary Insabella
(614) 287-5207
minsabel@cscc.edu
Ty Fogle
(614) 287-5781
tfogle@cscc.edu
Columbus State Community College
550 East Spring Street
Columbus, OH 43216

Abstract

With the ever-changing nature of hardware and software, academic IT departments have a large dilemma—constantly changing with the current direction of the prevailing wind (Considering emerging technologies, certifications, private academic institutions and expense), or to maintain traditional courses of study based in teaching theoretical concepts, while adopting some of the newer technologies. Other considerations are that IT changes can't happen overnight and that there is administrative red-tape with which to contend; additionally, sometimes there is resistance to change from established faculty who don't wish to expend the time and energy to learn newer technologies. Furthermore, Two-year college IT departments must constantly evaluate High School curricula and articulations with traditional four-year colleges so a fit is made for the student, whether it is a two-year graduate who wants to enter or re-enter the workforce, a student who wants to complete an undergraduate degree, or one currently in IT who wants to learn new skills. Our paper and presentation will consider the above questions and will discuss what academic changes should occur in academic IT, what should stay the same, and how these departments can effect the changes necessary to survive, while also considering what possible steps can be taken to alleviate the significant strain constant change puts on faculty and resources.

History of IT at Columbus State

The Computer Information Technology (CIT) Department at Columbus State began in 1963 focused on training students in data processing outcomes used in the local economy. Subjects such as Assembly Language, CICS, JCL, COBOL and other batch oriented programming was taught in conjunction with accounting and other business-oriented classes in order to produce graduates useful to local companies (especially insurance) and state government agencies.

The PC and minicomputer revolution that began in the eighties and exploded in the nineties fundamentally changed the nature of work and the definition of what was possible. Due to this, productivity sky-rocketed in the nineties and seemingly everyone was online. Departmental changes were made to incorporate networking and word processing and general productivity based software, but the focus of the curricula was still on batch programming. Refinements were constantly made and C++ and Visual Basic were added in the mid-nineties as nice-to-knows, but the emphasis remained on COBOL and other related courses. Given the nature of the local economy, the emphasis on COBOL prior to 2000 was warranted and supplied students with ample opportunity to find well-paying positions.

A few years later, the boom went bust and experienced programmers, network administrators and other IT professionals with many years of experience could not find work in their chosen field. The challenge to us is to identify that which industry and higher education has not yet been able to put its finger on—what is the definition of an educated, prepared, IT graduate. In other words, "where do we go from here"?

Soft Skills

One of the areas that employers seem to desire are so-called "soft-skills." An article in *Career Opportunities News* defines soft skills as "the cluster of personality traits, social graces, facility with language, personal habits, friendliness, and optimism that mark each of us to varying degrees." An employer in the article was quoted as saying, "Don't worry so much about the technical skills. We need you to teach them how to show up on time, how to work in teams, and how to take supervision." We have a variety of corporate employees on our CIT advisory board who tell us that in addition to technical skills, we need to produce graduates with these abilities of communication (both written and oral), teamwork, punctuality, and a strong work ethic.

We've taken this advice to heart and have tried to incorporate these kinds of skills in our technical classes. Writing and presentation skills are fairly easy to include. Making students write in all of our courses is becoming the norm. Many of us in the department use group work or essay questions on tests that compel a student to work well with others to solve problems, or to express his or her opinion. For example, one team-building exercise Mary Insabella uses is to have students break into groups of four each, and solve a problem by building a three-dimensional structure, as high as possible, with only a balloon and masking tape. Each member of the group is given a role that corresponds to a real life group encounter. So one person will be a leader, one person will be a nay-sayer, etc. The goal of the exercise is not about who builds the best structure, but about how well the team interacts.

Another way to promote soft skills in our classes is to use presentation projects. These projects can be individual or team based. Students are urged to dress in business attire when doing their presentations. We emulate workplace presentations so they understand what might be required of them on the job. The skills required for these presentations are the summation of all the general soft skills we believe employers want to see in graduates.

Computing Environment

One of the major areas of disagreement in our department is over technical curricula. Many senior faculty take the position that computer learning is in essence theoretical and cognitive because the technology changes are so frequent that no one can teach to a particular vendor or product, although those products may be used in the course of teaching general theory. For example, some take the position that Assembler should be taught as a first language to teach basic computer logic. This is also seen as a weed-out tool. How serious are students and how hard are they willing to work at a difficult language?

Other faculty argue that specific tools currently used in the industry are more relevant to teach. Visual Basic is seen by this group as a better entry level language because graduates are more likely encounter VB in a work situation, and it also provides an environment in .NET that paves the way to other languages such as C Sharp, C++ and J Sharp. The argument can be made further, that being a two-year school, Columbus State's mission is to provide associate degree seeking students the ability to secure well-paying jobs in their chosen field.

Some of these issues with which our department is struggling are addressed by a document entitled "Computing Curricula 2001" (hereinafter referred to as CC2001), published by an association of the Institute for Electrical and Electronics Engineers (IEEE) known as the IEEE Computer Society. This publication discusses what core courses had traditionally been taught in Computer Science, what core courses should be taught going forward, and the general topics necessary for graduates to take throughout the remainder of their education. Even though the document is almost four years old, its relevance speaks to much of the technology in the world today. The association's most recent previous curricula document, CC1991, mentioned only six hours of networking in most IT curricula, and none of the issues of web computing were touched. Add to this the focus on object-oriented versus batch programming, and their resultant effects on systems, and one realizes this is a different world of computing as compared with a relatively short time ago.

Further complicating curricular decisions is our locality. Columbus, Ohio is a banking and insurance hub, and the state capital. These industries and government agencies tend to use a lot of COBOL and the question is what will happen as those COBOL programmers start to retire? Will the records be converted or will there be a new need to train COBOL programmers? And, as a college, how much should the local environment influence our curricula?

Also, proprietary tools are more complicated to choose. They seem to come and go with regularity. We often get calls from prospective students asking if we teach a tool that their company is now using. As often as not, it is the first time we're hearing the name of this package. It is often some obscure piece of software that someone in their company has decided to try out. Of course, it's impossible for us to teach every piece of software on the market.

Essentially, as CC 2001 makes clear, the body of knowledge in Computer Science is increasing so rapidly "that the restrictions of most degree programs make it difficult to add new topics without taking others away. It is often impossible to cover new areas without reducing the amount of time devoted to more traditional topics whose importance has arguably faded with time. The CC2001 task force has therefore sought to reduce the required level of coverage in most areas so as to make room for new areas."

Burdens on Faculty, Burdens on Budgets

The on-going discussion also has some political and business implications. For every new thing on the market, someone in our department needs to expend effort to try to understand it. As many of us know, each computer language can be a universe unto itself, and learning a new one requires extensive effort and study. The political realities are that IT faculty in our institution are remunerated the same as other faculty in the college, and some don't want to have to constantly

retool in order to stay cutting edge. This is understandable, especially for faculty who are nearing retirement. After all, academics in English or Mathematics or History deal with some changes in subject matter as well as changes in educational theory and practice, but nothing on the order with academic IT departments.

However, faculty must realize "this is the business we've chosen," and that curricular changes must be considered as an on-going part of a department's mission. The days are gone where people worked on mainframe computers with predictable software. Web, networking, object-oriented programming, and changes in business models and systems require us to make necessary changes if we are to survive.

Additional issues revolve around budget constraints, which can limit practical teaching exercises in computer academics. The essential need for specific labs in which to do networking and computer repair, as well as the specific equipment and software required for students to experientially learn the material, mean that extra capital is needed in IT departments. This is not always readily understood by administrators who sometimes look at academic departments from a traditional Arts and Sciences viewpoint.

Other Considerations for Curricula in 2005

Ethics

In his paper on *Integrating Computer Ethics into the Computer Science Curriculum,* Keith Miller writes about the plausibility of incorporating ethics into our classes. He brings up the following issues:

- *"The curriculum already is overcrowded. Including ethical issues requires that important technical issues be ignored or given less attention than they should be given.*
- *Computer science faculty have little experience in ethics, and they are uncomfortable teaching in this area.*
- *Professors unschooled in formal ethical techniques may fall into the trap of preaching a moral code (an abuse of their position) instead of raising questions, elaborating possible answers, and exploring justifications (activities which properly belong to ethics).*
- *Any ethics taught will be diluted at best and possibly erroneous"*

Dr. Miller proposes some of the following ethical questions: "Who is responsible for the accuracy of stored information? Who owns information? When is sharing ideas stealing products? How do we distinguish between technical judgments and moral choices?" Some of these are questions can easily be incorporated into computer classes, but others might take us into murky waters. Do we want to engage in social and political commentary?

Operating as a two year institution further constricts how much we can add to the mix. We often have to restrain ourselves as we are discussing what can be added to our curricula. Many times we get carried away with possibilities and someone has to say the number "110." We all know what that means. That is the maximum number of credit hours we can squeeze into a degree pro-

gram. It would be nearly impossible for us to create a course specifically in computer ethics--There simply is not room. So in order to address these issues, we have to try to incorporate them as outcomes in individual courses.

Articulation Agreements

A major focus in any college technical program's curricula should be content being taught on other planes of a student's experience. The Ohio Board of Regents is working on changing minimum computer outcomes requirements for all post secondary graduates. In effect, they are saying that high school programs need to teach to a different and higher level, which means that our program's introductory courses will be at a higher level.

Teaching to a higher level is not new to many in our discipline, as many high schools in our area are teaching courses for A+ and networking certifications, and are teaching programming languages. With higher skill and knowledge levels, some students are able to use some high school courses to satisfy a few College requirements. Ohio's tech Prep Consortium has articulation agreements with area high schools and some community colleges that reflect this partnership. While college educators need to be wary of accepting certain courses because they may not be up to our standards, we can no longer say that a high school student must start from scratch.

Likewise, four-year colleges must be willing to make a serious effort to fully articulate a community college student who has earned an associate degree. The quality of education that student received at an accredited community college should be assumed to be equal to that of the first two years of a four year institution. There may be a few differences in practical IT courses from the four-year institution, but the number of courses needed to transfer in at junior level should be minimal. Indeed, statistics show a seventy percent graduation rate at a four-year college for those who received an associate degree from a community college, versus a sixty percent graduation rate for students who started at the four year college

Conclusion

How can we best serve our students? That's the question for us at Columbus State. In a community college environment, especially, one must ask that question regarding his or her students vis-à-vis the locality. Workforce development needs to be preeminent in our experience in Columbus Ohio and creating a program with regard to transferability to a four year institution is secondary. But each college needs to answer this for themselves. Our goal in Columbus, Ohio may be the opposite from Columbus, Georgia..

Defining ultimate student goals, however, is only part of the solution. The other part is tackling those issues raised herein. How do we motivate faculty to constantly stay current in their field and integrate this learning into revised or new courses. What should be the role of articulation agreements with regard to the future of computer education between High Schools, two-year colleges and four-year colleges and universities? How do we make administration realize the importance of our programs and the need to fund them? These are questions our paper is raising, and one that our presentation will hope to answer.

References

"Integrating Computer Ethics into the Computer Science Curriculum," Keith Miller, the Research Center on Computing & Society, Southern Connecticut State University, 2004. http://www.southernct.edu/organizations/rccs/index.html

"Measure your Soft Skills Smarts," Peter Vogt, MonsterTRAK, 2005. http://content.monstertrak.monster.com/resources/archive/jobhunt/softskills

"'Soft Skills' a key to employment today," Career Opportunities News, October 2002, Volume 20, Number 2, Ferguson Publishing Company, www.fergpubco.com

"The Curriculum Foundations Workshop in Computer Science," Allen Tucker, Curriculum Foundations Workshops, Bowdoin College, October, 1999. http://ww.maa.org/cupm/crafty/focus/cfwcs.html

"Computing Curricula 2001," Computer Society of the Institute For Electrical and Electronic Engineers (IEEE-CS), December 15, 2001.

"Memorandum of Understanding—2004 Program Review of Computer Science Department, College of Arts and Sciences," University of Hawaii, October 15, 2004.

"Resources to Support the Information Science and Technology Initiative." Pennsylvania State University, 1998. http://www.psu.edu/ufs/IST/resources.html