

White Paper on IT Education
Andrew Bernat
Computing Research Association
(but a CS prof for 20 years before that)

The academic computing enterprise faces two shortages: students and research funding. Both of these relate to the current image of computing among students, their parents, the general population and the political class. To solve the shortages, we must work on the image.

Our educational image problem. For years computing has been seen as the safest path to a highly regarded, stable and well paying job. Thus students flowed into the field and our greatest challenge was keeping our numbers to reasonable levels so that we could pretend to provide an education. Thus, poorly thought through curricula with no theoretical or research basis, material presented poorly in the classroom, courses and requirements established as filters, etc. It is not surprising that, when the perception of employment changed, students ran from our programs (see, e.g., <http://www.cra.org/CRN/articles/may05/vegso.html> and <http://www.cra.org/info/taulbee/bachelors>).

Our research image problem. We succeeded. When people use a computer, they basically get what they want. Particularly when compared to five years ago. They search and information pops up. They have multiple windows they can navigate through. They have interconnectivity. So what's left except refinement? And refinement is development, not research. Computing professionals have done a very poor job of exciting the world about where we could go and where we could be; instead we've quietly worked on the next version of Windows or Office or an efficiency enhancement of 5%. No wonder the next big thing is no longer computing. Yet computing is an incredible multiplier - improvements in computing create improvements everywhere else. See <http://www.cra.org/govaffairs/blog/index.php> for more information.

Back to IT curriculum.

Step one is for the computing community to understand what it is about computing that is intellectually exciting and motivating. (From my perspective we have it all - we not only get to explore a universe, like physics or astronomy, but we also get to create that universe; we are a field every bit as full of creativity as, e.g., art.)

Step two is for the computing community to figure out how such material must be taught in order for students to learn it. We know precious little about how students comprehend abstraction, programming, etc. We need a research base in these areas. Our current courses are designed by anecdote or personal preference. We wouldn't conduct lab work this way. Other fields have worked to gain understanding of how students learn in their context; computing needs to.

Concomitantly we need to recognize that computing is absolutely fundamental to intellectual achievement in any field. Just as an educated person needs to know something about mathematics, physics, art, literature, etc., an educated person needs to understand computing, its uses and its limitations. We need to build academic structures that encourage this broad-based understanding of computing while simultaneously allowing students to dive into computing as far as they wish to go.

1. Preparing undergraduates for computing careers: There is a natural tension between employers who need employees who can produce now and faculty who wish students to leave with theoretical grounding so that they are prepared for the future.

2. Transforming the educational experience: We need to know how students learn our field, and then we need to build courses using this information.
3. Models for transforming computing education: I am enamored of using problem-based learning with appropriate scaffolding as a powerful delivery mechanism. PBL has been successful, even evaluated.
4. Inhibitors and strategies: Academic pigeonholes fight against broad-based computing education. This is not a trivial issue. In any interdisciplinary effort, questions such as who gets course credits, who evaluates the faculty, etc., are absolutely vital. It is vital that the large research institutions be involved in the entire process, but the faculty at such institutions are generally not valued for their participation in educational activities - how to get them involved without penalizing them?
5. Who might participate: To date curricula have been driven by the professional societies and that is likely to continue. What is required is for people with a broader vision to get involved and then there has to be follow-through. As an example, recall the breadth-first curriculum concept that was developed by ACM-IEEE-CS as an attempt to convey the entirety of computing to students, not just the nuts and bolts. But there was no coordinated follow-through to provide teaching materials, etc.

(If you have read this far, you probably have figured that I am being deliberately provocative. I do not think that we should be spending our time discussing whether objects come first, last or somewhere in between. The Titanic is sinking.)