The Capacity Crisis in Computer Science
Priorities for a Time of Unprecedented Growth

Eric Roberts
Professor of Computer Science
Stanford University

NSF Priorities Workshop
Orlando, Florida
January 8, 2014
The Beauty

- Software-intensive systems are perhaps the most intellectually complex artifacts created by humans
  - Software is invisible to most of the world
  - The best software is simple, elegant, and full of drama as manifest in the cunning patterns that form its structure and command its behavior
Passion, Beauty, Joy, and Awe

*That there is currently a crisis in computing education is not in doubt.*

— McGettrick et al., *SIGCSE 2007*

- As everyone has now been aware for some time, computing enrollments in the United States and most of Europe have plummeted since 2001.

- This drop is of significant economic concern because those same countries are training far fewer people than they need to fill the available positions. In the United States, there are now many more jobs in the IT sector than there were at the height of the dot-com boom, with all projections pointing toward continued growth.

—From my SIGCSE PBJA panel presentation in 2008
What a Difference a Year Makes

*That there is currently a crisis in computing education is not in doubt.*
— McGettrick et al., *SIGCSE 2007*

- As everyone has now been aware for some time, computing enrollments in the United States and most of Europe have plummeted since 2001. *Are enrollments still falling?*

- This drop is of significant economic concern because those same countries are training far fewer people than they need to fill the available positions. In the United States, there are now many more jobs in the IT sector than there were at the height of the dot-com boom, with all projections pointing toward continued growth. *Is this still true after the meltdown?*

- In Silicon Valley and at Stanford, the answers are clear:
  - Demand for talented software developers is as high as ever.
  - CS enrollments are skyrocketing, nearly erasing any previous loss.

  —From my SIGCSE PBJA panel presentation in 2009
The educational data comes from the National Center for Education Statistics IPEDS (Integrated Postsecondary Education Data System) Data Center. The data used is for degrees granted in the 2008-2009 academic year.

The employment data comes from the Department of Labor’s Occupational Outlook Handbook for 2010-11. This handbook includes employment for 2008 as well as a 10-year projection to 2018. I manually selected which occupations mapped to which degrees. I calculated job openings per year as 10% of the expected job growth over 2008-2018 plus 2.5% of the number of jobs in 2008. This second term describes the number of jobs opening as people retire. It assumes that people work for 40 years and leave a job at a uniform rate; the latter is of course not true in difficult economic times.

Top Ten Majors at Stanford (as of July 2013)

1. Computer Science
2. Human Biology
3. Engineering IDMs (mostly Product Design)
4. Science, Technology, and Society
5. Biology
6. Economics
7. Management Science and Engineering
8. Mechanical Engineering
9. International Relations
10. Psychology
Top Majors at Stanford
(as of July 2012)

Computer Science: 400
Human Biology: 339
Engineering: 291
Biology: 194
Economics: 187
Psychology: 187
International Relations: 146
Management Science and Engineering: 136
History: 133
English: 127
Earth Systems: 119
Political Science: 109
Electrical Engineering: 108
Symbolic Systems: 105
Mathematics: 105
Public Policy: 84
Physics: 68
Chemical Engineering: 65
Art: 61
Mathematical & Computational Science: 58
Communications: 55
Top Majors at Stanford
(as of July 2013)

<table>
<thead>
<tr>
<th>Major</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Science</td>
<td>512</td>
</tr>
<tr>
<td>Human Biology</td>
<td>330</td>
</tr>
<tr>
<td>Engineering</td>
<td>289</td>
</tr>
<tr>
<td>Biology</td>
<td>209</td>
</tr>
<tr>
<td>Economics</td>
<td>203</td>
</tr>
<tr>
<td>Electrical Engineering</td>
<td>175</td>
</tr>
<tr>
<td>Mechanical Engineering</td>
<td>144</td>
</tr>
<tr>
<td>International Relations</td>
<td>140</td>
</tr>
<tr>
<td>Psychology</td>
<td>121</td>
</tr>
<tr>
<td>Political Science</td>
<td>119</td>
</tr>
<tr>
<td>English</td>
<td>106</td>
</tr>
<tr>
<td>Earth Systems</td>
<td>105</td>
</tr>
<tr>
<td>Symbolic Systems</td>
<td>103</td>
</tr>
<tr>
<td>History</td>
<td>102</td>
</tr>
<tr>
<td>Mathematics</td>
<td>94</td>
</tr>
<tr>
<td>Computational Science</td>
<td>92</td>
</tr>
<tr>
<td>Chemical Engineering</td>
<td>88</td>
</tr>
<tr>
<td>Public Policy</td>
<td>70</td>
</tr>
<tr>
<td>Physics</td>
<td>68</td>
</tr>
<tr>
<td>Communication</td>
<td>61</td>
</tr>
<tr>
<td>Communication</td>
<td>59</td>
</tr>
<tr>
<td>Communication</td>
<td>52</td>
</tr>
</tbody>
</table>
The Student Growth Is Now Common

- The problem of the 2000s was insufficient student demand.
- The problem of the 2010s will be insufficient faculty capacity.
- What we face is a “success disaster” in the making.
Success Disasters
Success Disasters

Word Spy - success disaster
wordspy.com  S ▼
Apr 11, 2012 - Massive problems created when a person or company is unable to handle an overwhelming success.

Groupon's Success Disaster - Redfin Real Estate Blog
blog.redfin.com/blog/2010/09/groups_success_disaster.html ▼
by Glenn Kelman - in 694 Google+ circles
Sep 16, 2010 - Pingback: Groupon's Success Disaster (Glenn Kelman/Redfin Corporate Blog). Pingback: Groupon's Success Disaster (Glenn Kelman/Redfin ...)

Success Disaster - CRIENGLISH.com
english.cri.cn/8706/2013/04/22/1943s760903.htm ▼
Apr 22, 2013 - The phrase I'm going to introduce to you today is -- success disaster. Its definition goes like this: (clip, success disaster). Success disaster noun.

Amazon Web Services Blog: Avoiding a Success Disaster
aws.typepad.com/aws/2006/11/avoiding_a_succ.html ▼
Nov 2, 2006 - For a while I have been using the term "success disaster" to characterize what can happen on the web all too easily. What's a success disaster?

Free Holiday Cloud Consultation to Avoid a Success Disaster
Is your campaign ready for the Holiday Season? Avoid a success disaster and review your project in detail with one of our cloud experts.
Success Disasters

The MBone has been regarded as one of the Internet's "success disasters": an experiment that has rapidly outgrown the confines of the lab or the testbed, using prototype software that was never meant to operate at the scale that is being demanded by its users.

—Ajit S. Thyagarajan et al., "Making the MBone Real," /SOC, May 10, 1995
We’ve Been Here Before

• The situation that Computer Science is likely to face over the next five years is not unprecedented. Much the same situation occurred in the 1980s.

• In the early 1980s, the availability of personal computers led to an explosion of student interest in computer science. At some universities, student demand doubled in a single year.

• Unfortunately, universities did not have the capacity to satisfy the growing demand.
  – Workloads for faculty increased substantially.
  – Some faculty members began to leave for greener pastures.
  – Replacement faculty were nearly impossible to find.
  – Graduate students turned away from academic careers.
COMPUTER MANPOWER - IS THERE A CRISIS?

Kent K. Curtis
National Science Foundation
Washington, D.C. 20550

ABSTRACT

Factors influencing the supply and demand of computer manpower are analyzed in the context of available data on scientific manpower including statistics on degrees awarded in various discipline at the Bachelor's, Master's, and Ph.D. levels, faculty mobility, job mobility among professionals, starting salary trends, comparative unemployment statistics and economic projections. It is found that there is a shortage of computer manpower which is expected to persist for the foreseeable future but that society is responding, perhaps as rapidly as possible, to provide the trained people required by business, industry and government. Only the educational institutions of the U.S. have what might be described as a crisis, a staffing problem which seems to have no solution within the context of normal supply/demand forces.
• *Students are not entering graduate school* but are being lured by attractive salaries and professional opportunities at the bachelor's level. Note the ratios in [Fig. 1](#). Computer Science has the lowest ratio of graduate degrees to bachelor's degrees of any discipline by a wide margin. There is no doubt that the bachelor's degree is proving to be a sufficient entree into the professional computer field for most people.

• *Graduate students are leaving graduate school* without completing their Ph.D.'s. In this case, note the ratios in [Fig. 2](#). Either that statement is true or an inordinate number of people are entering graduate school only for the master's degree. As [Fig. 3](#) shows, there is a higher ratio of part-time students to full-time student in Computer Science than in any other discipline. Since most part-time students are probably not Ph.D. candidates or planning to go on for a Ph.D., we can surmise that there are, indeed, more terminal master's students in Computer Science than in other disciplines. Even after subtracting out that difference, however, the number of Ph.D.s in Computer Science remains singularly low, indicating that many prospective Ph.D. students are dropping out. Anecdotal evidence strongly supports this conclusion.

• *Faculty are leaving academia for industry.* [3] The Computer Science Section of NSF conducted a faculty mobility study to understand better the reasons faculty change jobs. The number of faculty in Computer Science who made a job change in 1980, the year of the survey, was also estimated. Some results are shown in [Fig. 4](#). The survey confirmed that a substantial fraction of the faculty changing jobs is moving to industry. Other data shows that the percentage going into industry is higher in the computer field than in any other disciplines, even other branches of engineering. [Fig. 5](#) shows the percentage of faculty moving into industry for several areas of engineering. It is clear that the loss of faculty to industry is twice as high in computer science as in any other field of engineering. Finally, [Fig. 6](#) shows that the computer science faculties are not being replenished by new Ph.D.'s. Few Ph.D.s are being produced and only slightly more than half of those go into academic careers. This is unusually low in comparison with other disciplines.
4.1 Higher Education

Let us consider the conundrum facing the computer field in higher education first. It is experiencing an exponentially increasing demand for its product with an inelastic labor supply. How has it reacted? NSF has made a survey of the responses of engineering departments, including computer science departments in schools of engineering, to the increasing demand for undergraduate education in engineering. There is a consistent pattern in their responses and the results can be applied without exception to the computer field whether the departments are located in engineering schools or elsewhere. 80% of the universities are responding by increasing teaching loads, 50% by decreasing course offerings and concentrating their available faculty on larger but fewer courses, and 66% are using more graduate-student teaching assistants or part-time faculty. 35% report reduced research opportunities for faculty as a result. In brief, they are using a combination of rational management measures to adjust as well as they can to the severe manpower constraints under which they must operate. However, these measures make the universities' environments less attractive for employment and are exactly counterproductive to their need to maintain and expand their labor supply. They are also counterproductive to producing more new faculty since the image graduate students get of academic careers is one of harassment, frustration, and too few rewards. The universities are truly being choked by demand for their own product and have a formidable people-flow problem, analogous to but much more difficult to address than the cash-flow problem which often afflicts rapidly growing businesses. There are no manpower banks which can provide credit.
5.0 CONCLUSION

The preceding analysis shows that the United States has an imbalance in the supply and demand for computer manpower which is expected to persist for the foreseeable future but that our society is adjusting rapidly to redress that imbalance in every sector except education. In education, the channels for adjusting supply and demand which are available to business, industry and government, principally mobility of personnel among job and shifts in the interests of students training for jobs, are ineffective and we cannot hope for short term solutions. This places our educational institutions under great stress and could lead industry to develop alternate methods or new institutions to provide the professional computer personnel it needs if the traditional educational institutions are unable to adjust. One can imagine fundamental changes in the structure and management of institutions, both of higher and pre-college education, which could help resolve the present dilemma but not without altering time-honored traditions. No other technological advance in history has been so rapid or so compelling in confronting education with the hard choice of either embracing fundamental change or accepting a reduction in its traditional role of training students for the future.
1. What challenges make it difficult for institutions to build the necessary capacity?
Proposed Research Questions for NSF

1. What challenges make it difficult for institutions to build the necessary capacity?

2. How can we maximize the production of students with the exceptional levels of talent that the industry demands?
Variations in Programmer Productivity

• In 1968, a study by Sackman, Erikson, and Grant revealed that programmers with the same level of experience exhibit variations of more than 20 to 1 in the time required to solve particular programming problems.


• Most industry insiders believe that the productivity variance is even higher today. In 2005, Google’s VP for Engineering, Alan Eustace, told The Wall Street Journal that one top-notch engineer is worth 300 times or more than the average.
The Microsoft Programming Personae

Microsoft’s cultural lore defines three types of programmers:

Mort is your most common developer, who doesn’t have a CS background, may even be a recent newcomer, and doesn’t quite understand what the computer is doing under the covers, but who writes the dinky IT programs that make businesses run. Elvis, more knowledgeable, cares about code quality, but has a life too. Einstein writes some serious-ass piece of code like device drivers, wants to get things done, needs to be able to go low level and high level, needs a language without restrictions to get his job done.

Proposed Research Questions for NSF

1. What challenges make it difficult for institutions to build the necessary capacity?

2. How can we maximize the production of students with the exceptional levels of talent that the industry demands?

3. What strategies — including those outside traditional academic programs and curricula — will help to expand the pipeline of students in this critical area?
WHEN French entrepreneurs decided in March to launch a swanky new school for software developers, they thought they were on to something. But even they were startled by its popularity. For 1,000 student places starting this autumn on a three-year course, they have fully 50,000 applications.

France has a skills mismatch. Joblessness has reached 10.6%, a 14-year high. For the under-25s, it is 26%. Yet, according to a poll by the French Association of Software Publishers and Internet Solutions, 72% of software firms are having trouble recruiting—and 91% of those are seeking software engineers and developers.
Dick Gabriel’s Proposal for a Software MFA

Another apt comparison can be found in the creative writing arts. It is entirely possible to become an extraordinary writer by one’s self, by simply writing and reading, and many excellent writers progress this way. A faster way to gain competence is through a Master of Fine Arts program, which is designed to rapidly increase one’s skills and to get one prepared to bring to bear critical thinking to the process of continuing improvement. Some believe that all aspects of software design and development are really engineering or scientific disciplines where the models of engineering and science apply, and I will not quarrel with them nor try to convince them otherwise.

This proposal is predicated on the belief that being a good software designer and developer requires talent, and that talent can be developed. We explicitly liken the practice of software to the practice of fine art.

http://www.dreamsongs.com/MFASoftware.html
Alternative Models of Software Education

Although Dick Gabriel’s model of an MFA in software is worth investigating, it may be more appropriate to create “conservatories” for the teaching of software arts, similar to music conservatories. One possibility might be some sort of New England Conservatory of Coding. (Or perhaps a Hogwarts School for Software Wizardry.)

Another model might be to create intensive programs that encourage students to focus on the art of software development, in much the same way that programs like the University of Virginia’s Semester at Sea program offers a concentrated immersion in oceanography, geography, and cultural anthropology.
References


The End