Can Components Save Us From the Software Talent Shortage?

Web.Builder San Francisco
April 14, 1998

Avron Barr and Shirley Tessler
Stanford Computer Industry Project
http://www-scip.stanford.edu/scip/

The Stanford University Computer Industry Project

- An Alfred P. Sloan Foundation Industry Study
- SCIP's Corporate Sponsors
  - Andersen Consulting
  - Booz-Allen & Hamilton
  - British Petroleum (BP)
  - Citibank N.A.
  - CMP Media, Inc.
  - Daiwa Inst. of Research Ltd.
  - Ernst & Young LLP
  - Fujitsu
  - FX Research Laboratory
  - Goodyear Tire and Rubber
  - Merrill Lynch
  - Microsoft Corporation
  - Montgomery Securities
  - Moore Corporation Ltd.
  - NTT
  - Perot Systems Corp.
  - PeopleSoft
  - Philips
  - Sumitomo Corporation
  - Symantec
  - Toshiba America, Inc.

SCIP Software Research Team

- Professor William F. Miller
  - Professor of Computer Science & Public & Private Management
  - President Emeritus of SRI International
- Professor Edward A. Feigenbaum
  - Professor of Computer Science
  - Recently returned from leave as Chief Scientist, US Air Force
- Avron Barr and Shirley G. Tessler
  - Co-Directors of the SCIP Software Research Team
  - Consultants on corporate use of advanced software technologies

Issues That Will Shape the Software Industry

- Intellectual property: patents, piracy
- Global competition, trade and immigration
- Consolidation & antitrust
- Distribution mechanisms and channels
- Quality, systems failures & litigation
- Education of software professionals
- Software development tools and methods
- Software project management practices
- Global capacity for software development

This Afternoon’s Question

- If there really is a shortage of software developers …
- If off-the-shelf componentware increases the productivity of developers …
- If web developers are better able to take advantage of component frameworks than other developers …
- Will we be able to get all the code written, someday?

Is There Really a Shortage of Software Developers?

The Scope of the Problem

Software Development Capacity Supply vs. Demand

- The world's supply of software development talent has natural limits
- The world's demand for software seems unbounded — every new idea for what computers can do …
- The productivity of developers (e.g., tools) hasn't increased as fast as demand
- We have reached global capacity

The Software Industry
Software teams & Resultant Value Chain

	Worldwide Expenditures	Impact, Value	Examples
Publishing	$122B	Tools Education Entertainment	Microsoft, Oracle, Nintendo
Services	$252B	Mostly software	EDS, IBM / GS, Andersen
In-house/ MIS	$700B+	Productivity Informed ops. Strategic apps.	Payroll, mfg. automation Yield mgt., supply logistics FedEx, home banking
Embedded	?	Functionality Communication	Consumer electronics (auto) Complex systems (airplane)

1997 data from IDC.

The Importance of Software

- SW products & services are a $374B industry
- The modern enterprise runs on software
  - A $1 trillion$^+$ annual expense worldwide
- Software is a competitive weapon
  - Not just a productivity investment anymore
  - Frequent flier, FedEx Ship, home banking, …
- Product functionality is achieved in SW
  - Cellular phones, airbags, Tamagotchi
- The use of SW in business is accelerating

The Importance of Software 2.0

- William Gates is the richest man in the US
- 3 of the next 10 richest men in the country also made their $ billions in software
- Software has become a valuable industrial commodity, like oil …

A Prolonged Shortage of Talent Will Reshape the SW Industry

- A year ago, the ITAA reported 190,000 unfilled positions in IT in the US alone, excluding government and non-profit organizations. This year, they estimate 340,000 open positions.
- The fundamental cause of the shortage is the rise in demand for software of all types.
- In the early '90s, the rising demand was masked:
  - Cutbacks in aerospace / defense spending
  - Massive layoffs at IBM, AT&T, Digital, etc.
  - Widespread downsizing of corporate IS depts.

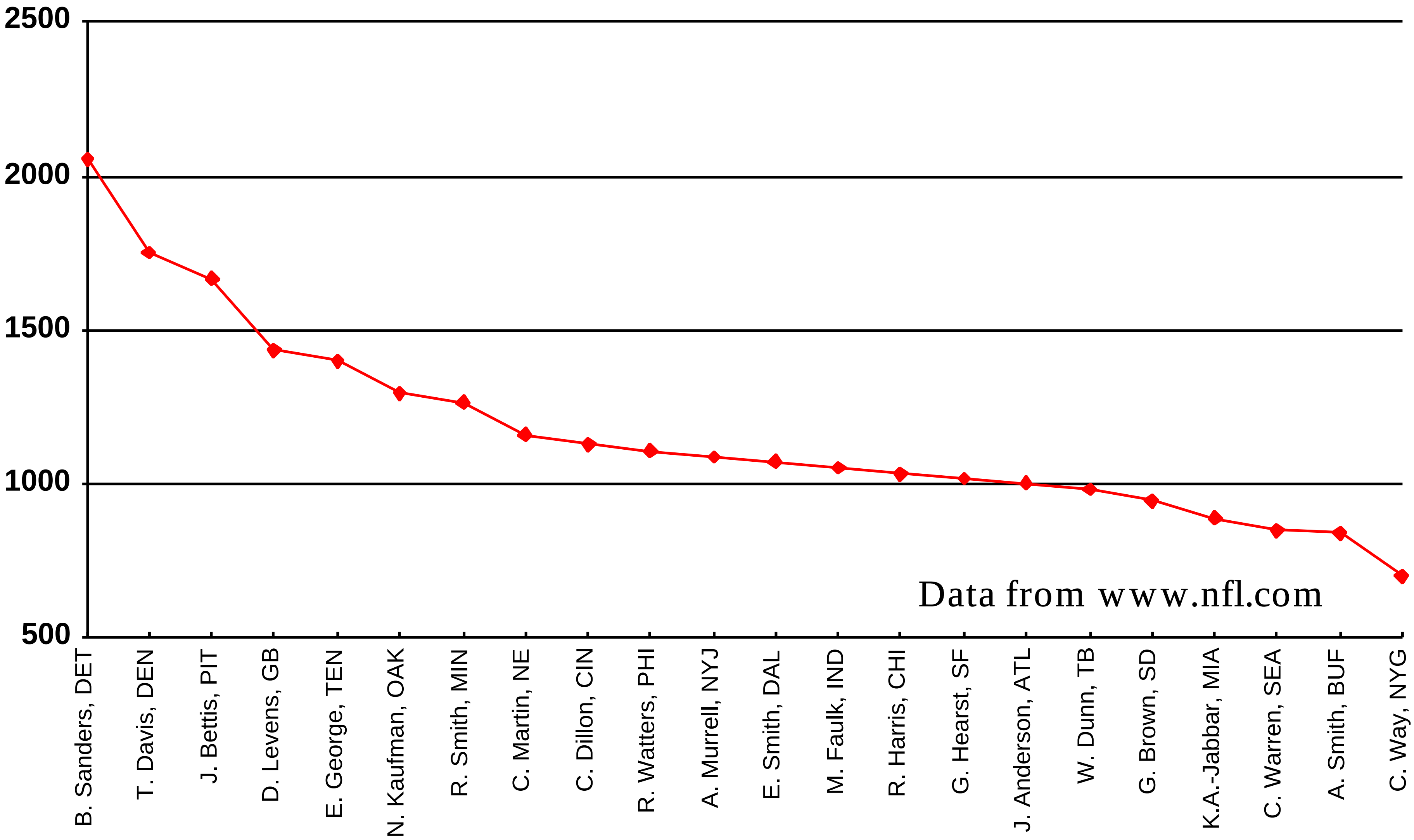
Some Common Misconceptions About the SW Talent Shortage

- Software involves talent as well as skill.
- This shortage is not local to a region or to the US — it is global. There are no major untapped pools of talent abroad.
- It is not limited to Year 2000, Java, SAP or any other technical specialty.
- The shortage is not focused on high-tech industries. In fact, they are the last to feel the pinch.

What is Software Talent?

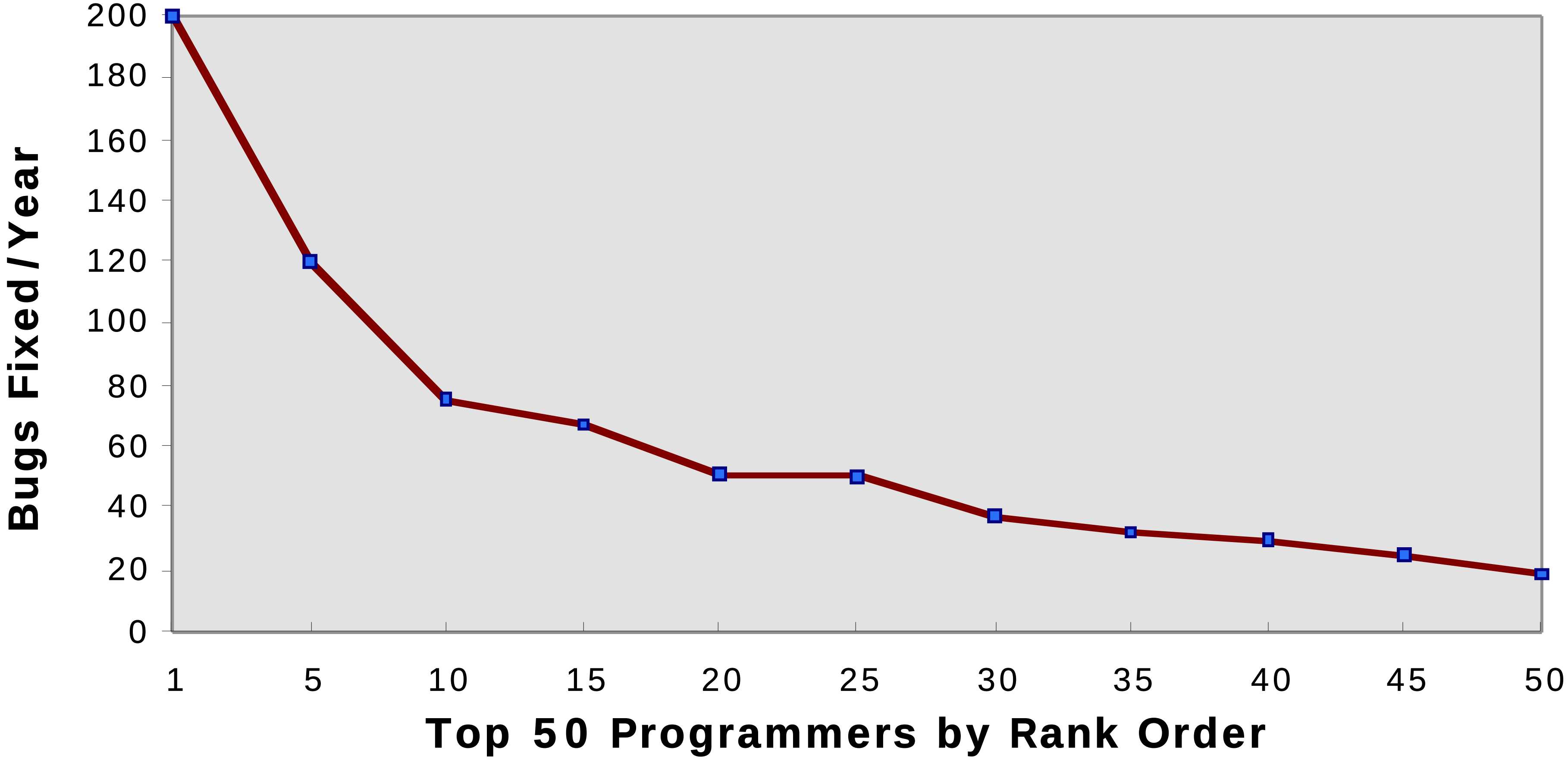
- Everyone has the same computers — it's the software people who create business value:
  - Analysis — needs, vision, and requirements
  - Architecture — form and function
  - Design — usability and construction
  - Development — still craftsmanship
  - Debugging & maintenance — skill & temperament
  - Testing — still undervalued
  - Documentation, training and support
  - Project management — key to success

1997 NFL Regular Season Rushing Statistics



Data from www.nfl.com

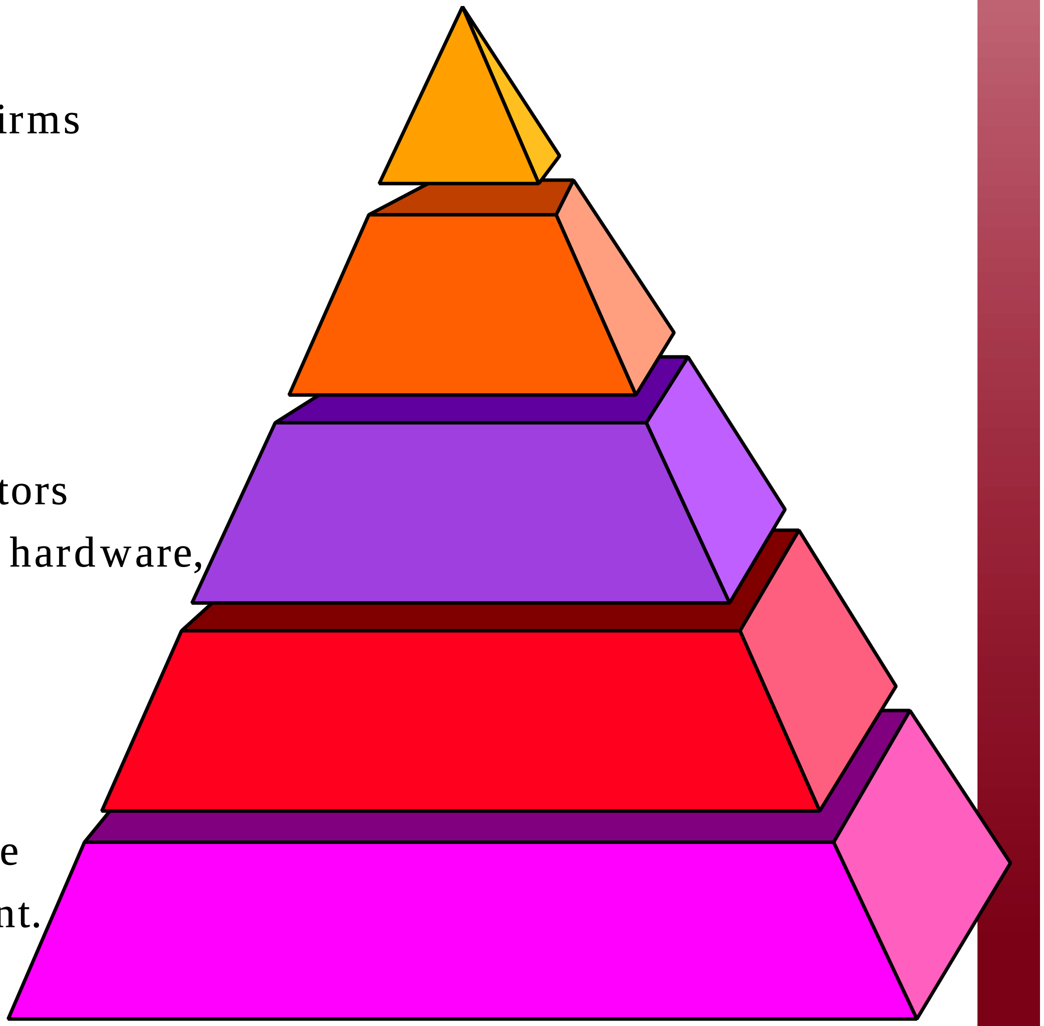
Software Talent — The Best are Significantly Better



“Not All Programmers Are Created Equal,” G. Edward Bryan, IEEE, 1994

The Software Workforce: Who’s Getting the Top Talent?

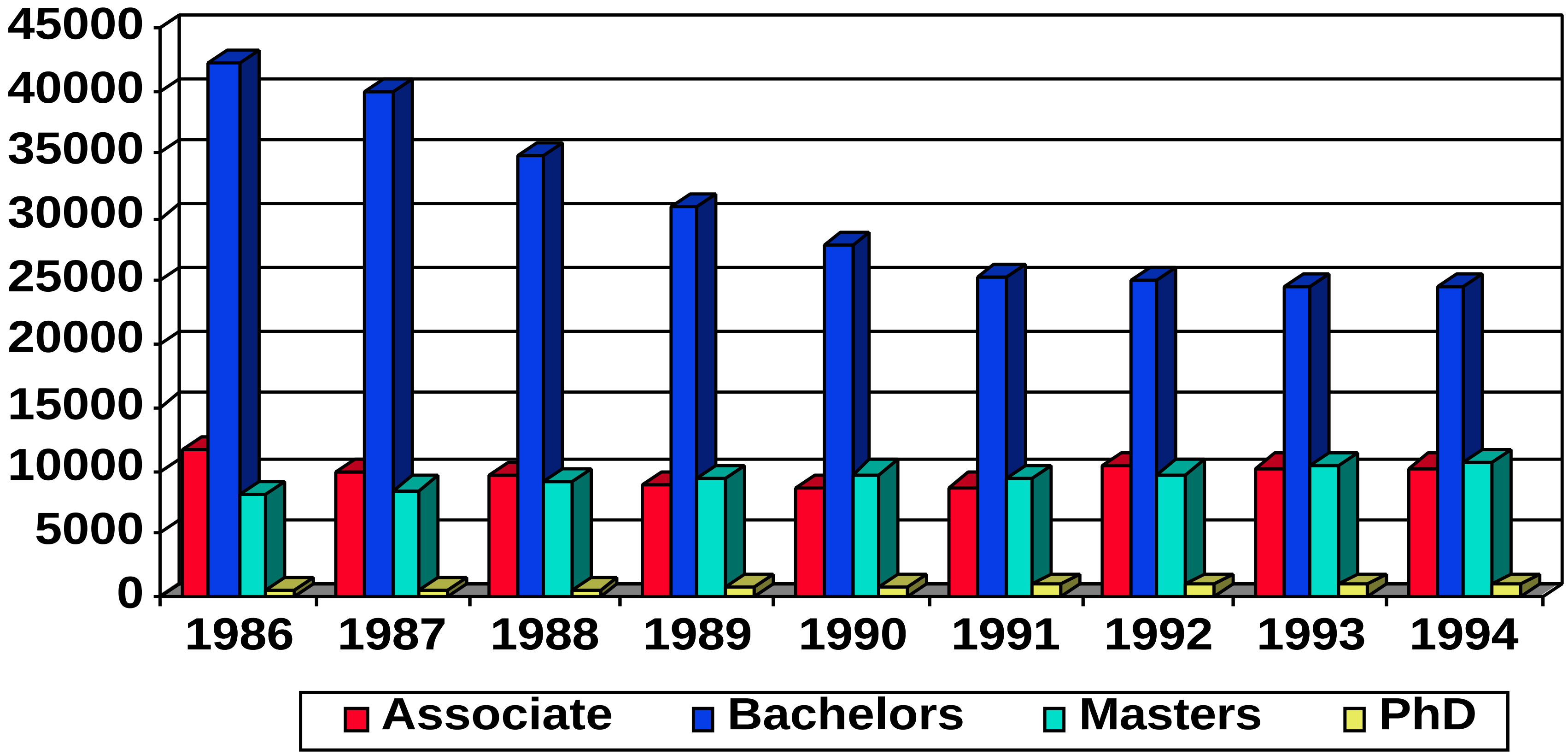
- **Tier 1 — Hot software companies**
  - Software start-ups & boutique services firms
  - Software publishers
  - Wall Street
  - R & D (corporate & university)
- **Tier 2 — Software-aware companies**
  - VARs, consulting firms, systems integrators
  - Software intensive industries (computer hardware, communications, financial services)
  - Aerospace systems firms
- **Tier 3 — Everyone else**
  - Other industries with incidental software
  - Most IS application development & maint.
  - DoD, federal, state & local government



Misconceptions, continued

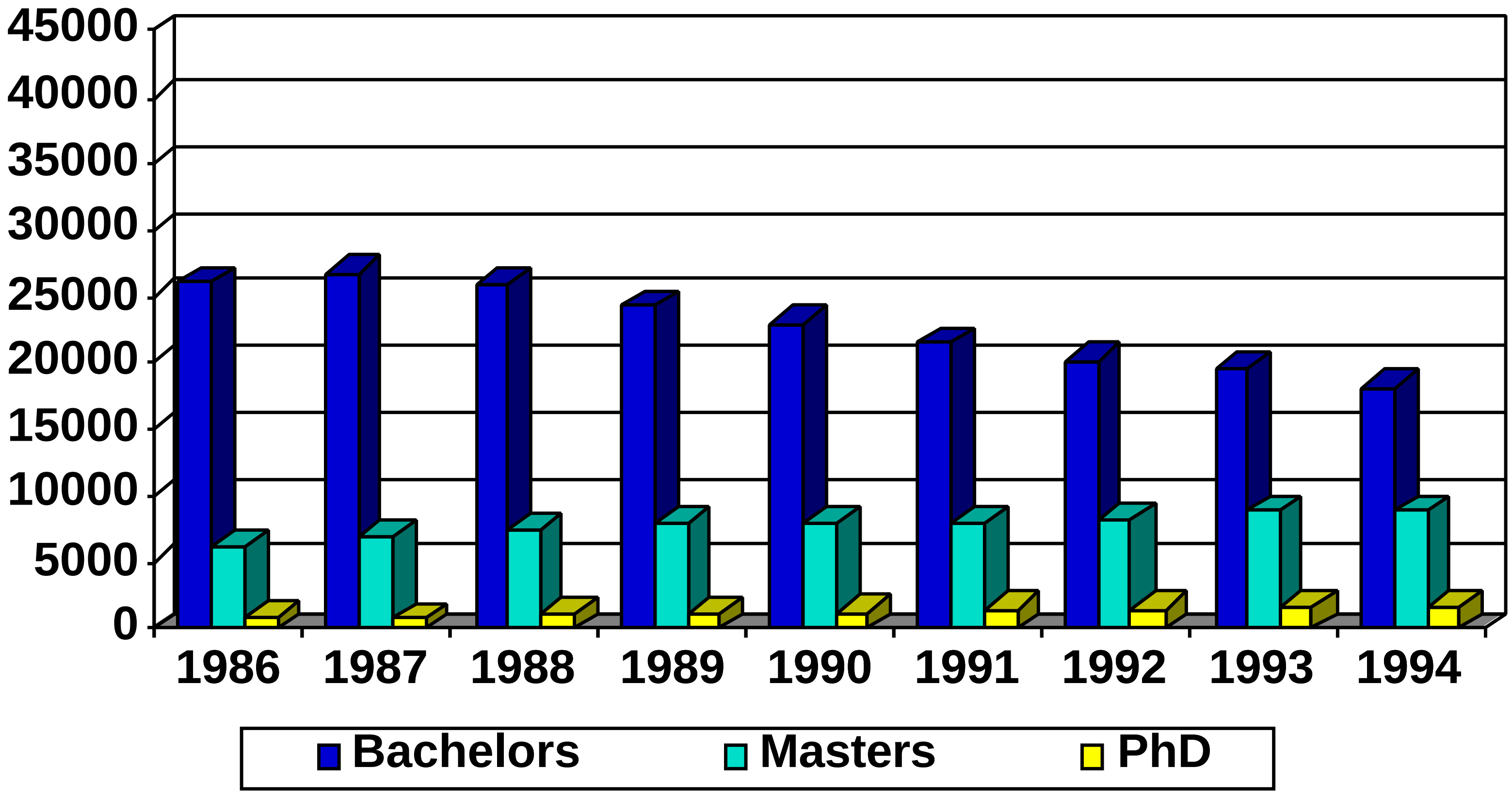
- The shortage is not temporary and will not be very sensitive to economic cycles.
- There are no magic technologies that will soon make programmers unnecessary.
- Rising salaries, global sourcing and training initiatives will only slowly increase the world’s capacity.
- The first generation of programmers is about to retire.

CS Graduates in the US 1986-1994



Source: Science and Engineering Indicators, 1996, 1997
Note: Associate Degrees includes Math and CS

EE Graduates in the US 1986-1994



Source: Science and Engineering Indicators, 1996, 1997

Will Components Save the Day?

(At least for web developers?)

A History of Technologies to Increase Developer Productivity

- Programming languages and compilers
- Database technology
- Structured programming
- Computer-assisted software engineering
- Knowledge-based programming
- Object-oriented programming & analysis
- Component-based assembly of software

Components — Two Common Definitions

- Software designed to work as a component of a larger application
  - Designed to work with other components in the same or other computers
  - It is possible to mix components from different manufacturers in a single system
- Microsoft: “A discrete unit of code built on ActiveX™ technologies...”

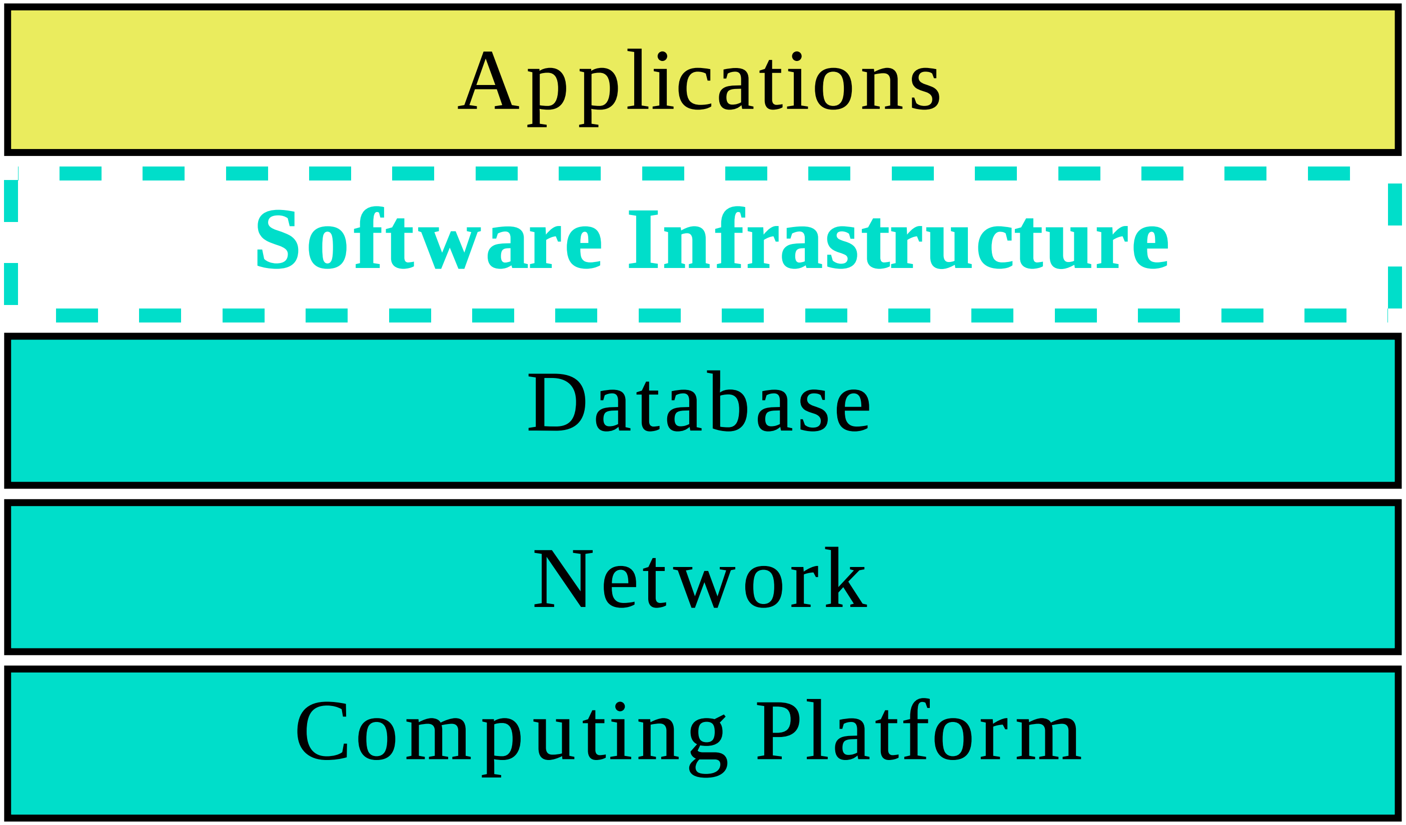
The Promise of Componentware

- Published software — economies of scale
- Modular, self-contained, standard interfaces (interoperable), black-box
- Allow development teams to “assemble” substantial parts of applications
  - Dramatic productivity increases
  - May reduce the wizard / coder ratio

Barriers to Reuse, Generally

- Components of what? — appl. architecture
- “Business model” — economies of scale
- Infrastructure investment: Selling the idea, managing code library, training & support
- Requires changing programmers’ habits about “other people’s code”
- Poor functionality, testing, interoperability
- Slow diffusion across “software worlds”

Extending the IT Infrastructure



But the Web Has Some Advantages

- Coherent community
- Communication
- Object-oriented at all layers
- Rapid growth
- Rich “frameworks”
- Standards, de facto and otherwise
- Too much to do too fast

The Demand for Software — A Competitive Necessity

- Quantity and variation
  - Published, enterprise & embedded software
  - Wintel, SAP-ification, corporate mergers, …
- Quality
  - “Minimum” quality is really difficult
  - Systems and product failures — litigation
- Complexity
  - This is rocket science
  - Innovation, speed and legacy

The Truth About Software

The better the tools,
the higher the bar.

Question Number 8.