

# Professional Software Development

**Steve McConnell**

stevemcc@construx.com  
www.construx.com/stevemcc/

© 2000-2003 Steven C. McConnell.  
All Rights Reserved.

Construx



## What I've Learned Since 1999

- ❖ Licensing is *way* more controversial than I thought
- ❖ The world didn't fall apart on Jan. 1,'00
- ❖ Lots of software projects do succeed
- ❖ Modern software development is truly impressive in many respects



Why Don't People Use  
Better Development  
Practices?



Gold Rush vs.  
Post-Gold Rush  
Software Development



## The Nub of the Problem

(As expressed 350 years ago)

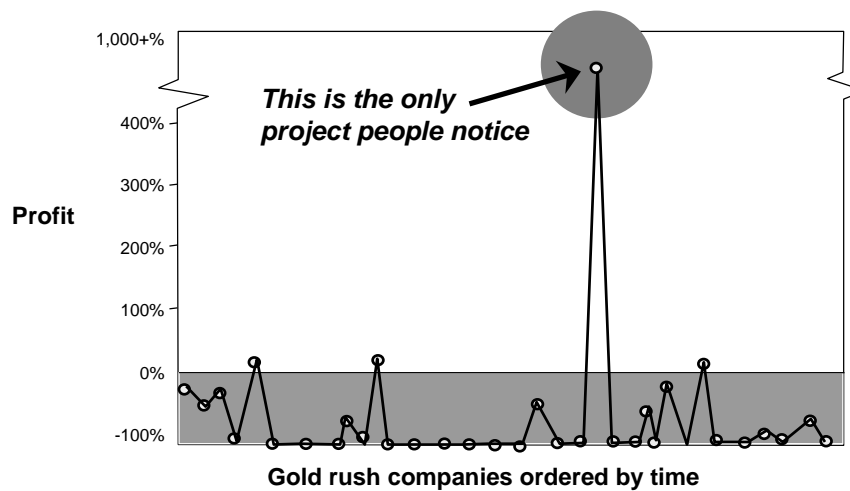
*“The root of all superstition is that men observe when a thing hits but not when it misses.”*

*-- Francis Bacon*



## The Nub of the Problem

(As expressed in today's terms)





## How Often Does Gold Rush Development Succeed?

- ❖ 81 percent of all small-company CEOs (<500 employees) surveyed thought an IPO was “very likely” or “somewhat likely”
- ❖ Of Inc. Magazine’s 500 fastest growing companies in 1998, 108 planned an IPO within 12 months
- ❖ One year later, only 8 of 108 had actually gone public or filed to go public
- ❖ IPOs (“going public”) is a red herring for the vast majority of companies



## Gold Rush Software Development

- ❖ **Customer Characteristics**
  - ♦ High risk tolerance
  - ♦ Time to market is critical, more important than reliability and other product characteristics
- ❖ **Developer Characteristics**
  - ♦ Small team sizes
  - ♦ Informal work practices
- ❖ **Sample Product: MS Mac Word 1.0**



## Post-Gold Rush Software Development

- ❖ **Customer Characteristics**
  - ◆ Low to moderate risk tolerance
  - ◆ Time to market is much less important than reliability, interoperability, usability, etc.
- ❖ **Developer Characteristics**
  - ◆ Medium to large team sizes
  - ◆ Increasingly formal work practices
- ❖ **Sample Product: Current version of MS Word for Windows**



## Observations

- ❖ **The problem with gold-rush development practices is that sometimes they work!**
- ❖ **Post-gold-practices work better even during a gold rush**
- ❖ **The rub: companies typically can't develop post-gold-rush abilities during a gold rush**

# Maturation of a Profession



## Art Without Engineering

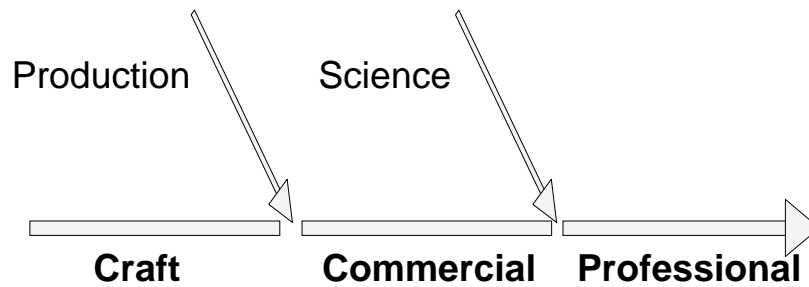




## Art With Engineering



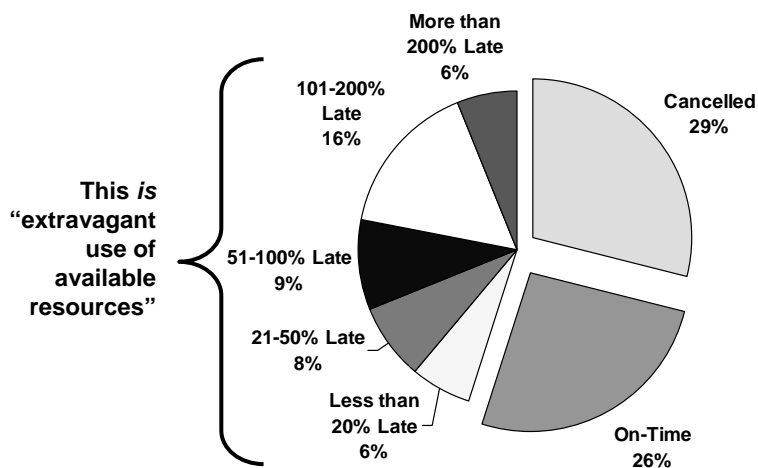
## Development of a Profession



# State of the Practice



## State of the Practice: Project Resolution



Source: (Standish Group, 1999)





## Best Practices

(year first available)

- ❖ **Project planning and management practices**
  - ◆ Automated estimation tools (1973)
  - ◆ Evolutionary delivery (1988)
  - ◆ Measurement (1977)
  - ◆ Productivity environments (1984)
  - ◆ Risk-management planning (1981)
- ❖ **Requirements engineering practices**
  - ◆ Change board (1978)
  - ◆ Throwaway user interface prototyping (1975)
  - ◆ JAD sessions (1985)



## Best Practices

(year first available, cont.)

- ❖ **Design practices**
  - ◆ Information hiding (1972)
  - ◆ Design for change (1979)
- ❖ **Construction practices**
  - ◆ Source code control (1980)
  - ◆ Incremental integration (1979)
- ❖ **Quality assurance practices**
  - ◆ Branch-coverage testing (1979)
  - ◆ Inspections (1976)
- ❖ **Process improvement**
  - ◆ SW-CMM (1987)
  - ◆ Software Engineering Process Groups (1988)



## Cargo Cult Software Engineering

In the South Seas there is a cargo cult of people. During the war they saw airplanes with lots of good materials, and they want the same thing to happen now. So they've arranged to make things like runways, to put fires along the sides of the runways, to make a wooden hut for a man to sit in, with two wooden pieces on his head for headphones and bars of bamboo sticking out like antennas—he's the controller—and they wait for the airplanes to land. They're doing everything right. The form is perfect. It looks exactly the way it looked before. But it doesn't work. No airplanes land. So I call these things cargo cult science, because they follow all the apparent precepts and forms of scientific investigation, but they're missing something essential, because the planes don't land.

— Richard Feynman



## Examples of Cargo Cult Software Engineering

- ❖ Producing documentation for the sake of documentation
- ❖ Working overtime for the sake of working overtime
- ❖ Slavish adherence to the SW-CMM
- ❖ Uncritical adoption of RUP
- ❖ Uncritical adoption of Extreme Programming
- ❖ Any elevation of form over substance



## Food for Thought

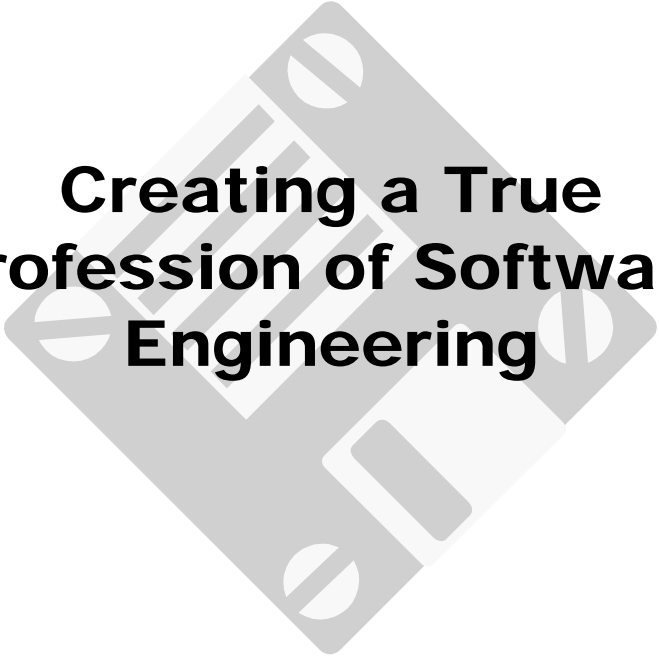
(Courtesy of the U.S. Department of Agriculture)

- ❖ **U.S. Agricultural Extension Service has been the most successful technology diffusion program in the world**
- ❖ **Agricultural extension employs 17,000 people to serve 3.8 million farm workers**
- ❖ **SEI employs 300 people to serve 1.8 million software workers**
- ❖ **Diffusion has not and will not occur by itself!**



## General Approach to Establishing Better Practices

- ❖ **Organizational level**
  - ♦ Hospitals, universities, accounting firms
  - ♦ Mechanism is organizational assessment  
*This is already happening through SW-CMM*
- ❖ **Individual level: *Professionalism***
  - ♦ Doctors, nurses, professors, accountants
  - ♦ Mechanisms are certification and licensing  
*This is underway in software engineering, but is not yet widespread*



# Creating a True Profession of Software Engineering



## A Profession for the Twenty-First Century

### ❖ Elements of a Profession

- ◆ Body of Knowledge
- ◆ Accredited university education
- ◆ Apprenticeship period
- ◆ Specialty exam after apprenticeship for licensing/certification
- ◆ Code of ethics
- ◆ Oversight by professional societies



## Current Status of the Software Engineering Profession

- ❖ Many of the elements already exist
- ❖ The rest are under development and/or being put into place now
- ❖ Further details on the following slides...



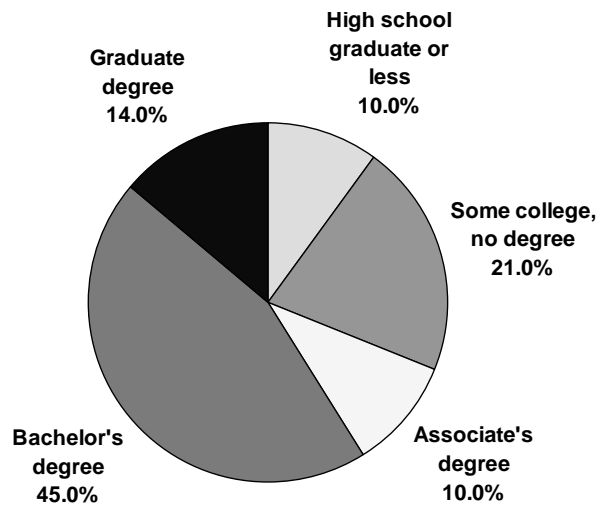
## The Software Engineering Body of Knowledge (SWEBOK)

- ❖ Software Requirements
- ❖ Software Design
- ❖ Software Construction
- ❖ Software Testing
- ❖ Software Maintenance
- ❖ Software Configuration Management
- ❖ Software Quality
- ❖ Software Engineering Management
- ❖ Software Tools and Methods
- ❖ Software Engineering Process

[www.swebok.org](http://www.swebok.org)



## Programmer Education



## More on Education

- ❖ About 40% of all workers have a software-related degree
- ❖ About half of the 40% obtained a degree in something else first
- ❖ About 20% have a degree in something else, but no software-related degree
- ❖ About 40% do not have a 4-year degree



## Graduate (Masters) Programs Are Well Established

### U.S.

- ❖ Azusa Pacific University
- ❖ Carnegie Mellon University
- ❖ Embry-Riddle Aeronautical University
- ❖ Kansas State University
- ❖ Seattle University
- ❖ Texas Christian University
- ❖ University of Texas at Austin
- ❖ University of Maryland
- ❖ University of Minnesota Institute of Technology
- ❖ Andrews University
- ❖ Colorado Technical University at Colorado Springs
- ❖ DePaul University

### U.S. (cont.)

- ❖ Drexel University
- ❖ Mercer University
- ❖ Monmouth University
- ❖ Southern Polytechnic State University
- ❖ Southern Methodist University
- ❖ University of Houston, Clear Lake
- ❖ University of Michigan
- ❖ University of Scranton
- ❖ University of Southern California
- ❖ University of St. Thomas

### Canada

- ❖ Universite du Quebec a Montreal
- ❖ University of Calgary



## New Undergraduate Programs Are Being Created

### U.S.

- ❖ Auburn University
- ❖ Clarkson University
- ❖ Embry-Riddle Aeronautical University
- ❖ Florida State
- ❖ U. of Michigan-Dearborn
- ❖ Milwaukee School of Engineering
- ❖ Mississippi State University
- ❖ Monmouth University
- ❖ Montana Tech
- ❖ Rochester Institute of Technology
- ❖ U. Texas at Arlington
- ❖ U. Texas at Dallas
- ❖ University of Wisconsin-Platteville

### Canada

- ❖ McMaster University
- ❖ Memorial University of Newfoundland
- ❖ University of Ottawa

\* These programs have been accredited by ABET



## **Certification of Software Engineers (Voluntary)**

- ❖ **IEEE has developed a Certified Software Development Professional program, which was rolled out in 2002**



## **Licensing is Underway**

- ❖ **Texas initiated licensing of professional software engineers in 1998**
- ❖ **British Columbia initiated licensing in June 1999**
- ❖ **Ontario initiated licensing in September 1999**
- ❖ **Other states and provinces are studying the issue including Illinois, Alberta, Quebec, and others**





## Who Will Licensing Affect?

- ❖ Most people will never need to be licensed
- ❖ Not for in-house programming work
- ❖ Not everyone on a team--only the person who signs his or her name
- ❖ Percentage of engineers licensed in other engineering fields ranges from 44% for civil engineers to 8% for chemical engineers
- ❖ Software will be on the low end of the range (or lower)



## ACM/IEEE-CS Code of Ethics

- 1 **PUBLIC** - Software engineers shall act consistently with the public interest.
- 2 **CLIENT AND EMPLOYER** - Software engineers shall act in a manner that is in the best interests of their client and employer, consistent with the public interest.
- 3 **PRODUCT** - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.
- 4 **JUDGMENT** - Software engineers shall maintain integrity and independence in their professional judgment.



## ACM/IEEE-CS Code of Ethics (continued)

- 5 **MANAGEMENT** - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
- 6 **PROFESSION** - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.
- 7 **COLLEAGUES** - Software engineers shall be fair to and supportive of their colleagues.
- 8 **SELF** - Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.



## Professional Societies

- ❖ **IEEE Computer Society**  
[www.computer.organization](http://www.computer.organization)



## **How Construx Can Help Support Your Company's Professionalism**

- ❖ **Organizational Assessments and Recommendations**
- ❖ **Training Needs Assessment**
- ❖ **Seminars**
- ❖ **Focused Workshops (requirements, estimation, etc.)**
- ❖ **Coaching & Consulting**



## Professionalism Resources

- ❖ Construx Software's professionalism website, professional ladder and recommended reading  
[www.construx.com/profession/](http://www.construx.com/profession/)  
[www.construx.com/ladder/](http://www.construx.com/ladder/)
- ❖ IEEE-CS Software Engineering Certification Homepage <http://www.computer.org/certification/>
- ❖ University of Texas's software professionalism website  
[www.cs.utexas.edu/users/ethics/professional.html](http://www.cs.utexas.edu/users/ethics/professional.html)
- ❖ *Professional Software Development*, by Steve McConnell (Addison Wesley, 2004)



## Closing Thought

(from 350 years ago)

*“Hope is a good breakfast, but it is a bad supper.”*

*-- Francis Bacon*



# Construx

Delivering Software Project Success

- ❖ **Software Projects**
- ❖ **Coaching & Consulting**
- ❖ **Training**

❖ ***info@construx.com***

❖ ***www.construx.com***