The Software Engineering Profession

SWENET Module
April 2004

Developed with support from the National Science Foundation

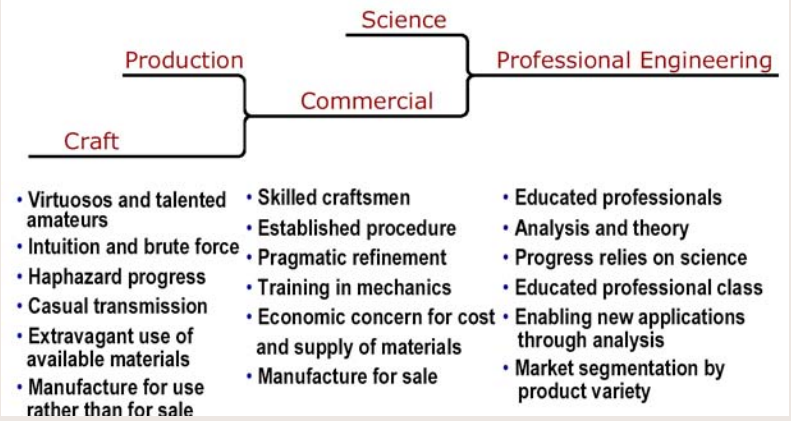
Overview

- What is an engineering profession?
- What does take to be a software engineer? (i.e. a good one!)
- What is the role of professional conduct and professional practice in software engineering?
- Ariane 5 Case Study Exercise

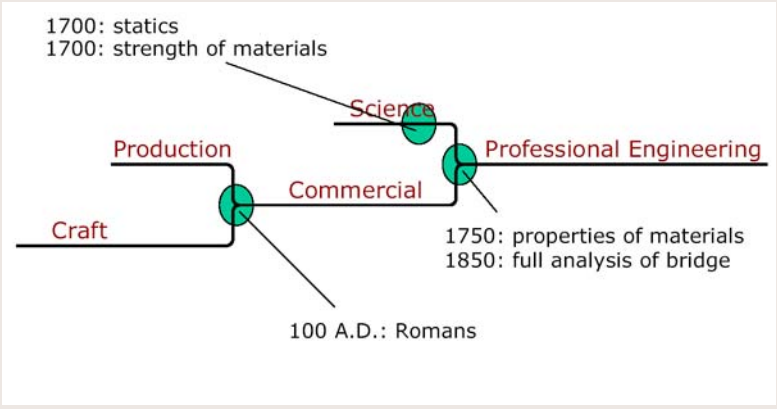
A Profession

- What is a profession?
  - What is a professional?
  - What is professional practice?
- What are the key features of a profession?

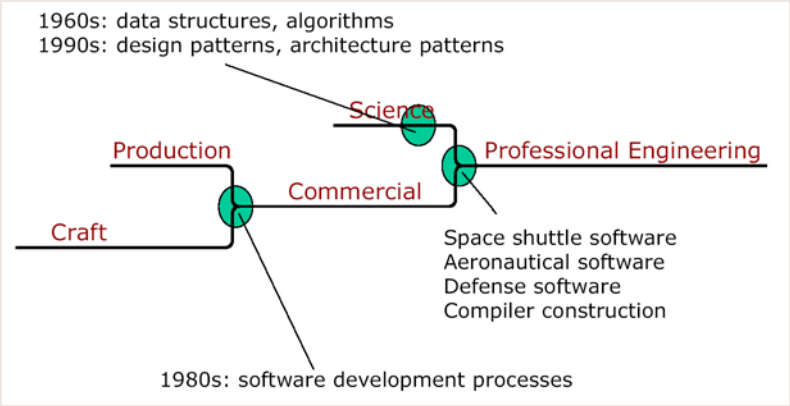
Evolution of an Engineering Profession



Example: Civil Engineering



Evolution of Software Engineering

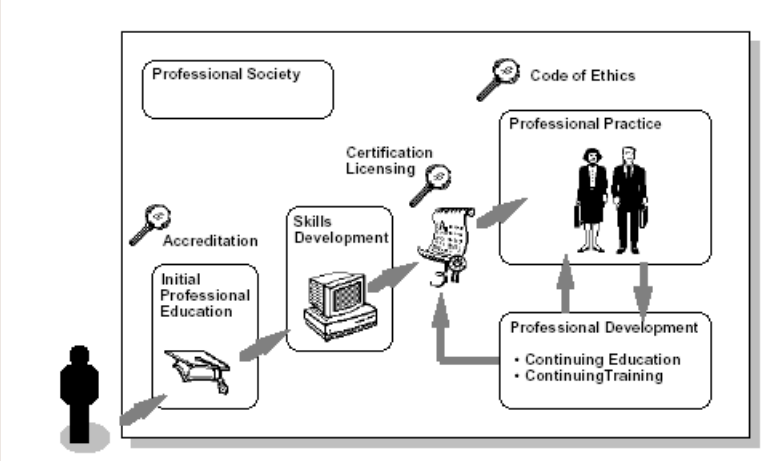


Ford-Gibbs Model 1

- In 1996, Gary Ford and Norm Gibbs, of the SEI, published a report on the software engineering profession [Ford 1996]. In the report they devised a framework for defining and assessing a profession. The below table and the next slide outline this framework.

Practitioner Level	Infrastructure Level
Professionals	Initial Professional Education
Knowledge	Accreditation
Professional Practice	Skills Development
	Certification
	Professional Development
	Code of Ethics
	Professional Society

Ford-Gibbs Model 2



[Ford 1996]

Initial Professional Education

- Undergraduate programs for SE education:
  - B.S. Computer Engineering
  - B.S. Computer Science
  - B.S. Electrical Engineering
  - B.S. Software Engineering
    - 20+ U.S. programs
    - 4 ABET accredited programs

Accreditation

- Accreditation is a designation that an organization or business has met a combination of standards and abilities that are put in place for public safety, welfare and confidence.
  - Colleges and Hospitals are accredited. Hospitals are accredited.
  - People are not accredited.
    - People may be certified or licensed.
- Curriculum guidance and accreditation standards and criteria are provided by a number of accreditation organizations across a variety of nations and regions:
  - Accreditation Board for Engineering and Technology (ABET (http://www.abet.org/)
  - British Computer Society (BCS) (http://www1.bcs.org.uk/)
  - Canadian Engineering Accreditation Board (CEAB (http://www.ccpe.ca/)
  - Japan Accreditation Board for Engineering (JABEE) (http://www.jabee.org/)

ABET SE Criteria

PROGRAM CRITERIA FOR SOFTWARE AND SIMILARLY NAMED ENGINEERING PROGRAMS

Submitted by The Institute of Electrical and Electronics Engineers, Inc.

These program criteria apply to engineering programs which include software or similar modifiers in their titles.

1. Curriculum

The curriculum must provide both breadth and depth across the range of engineering and computer science topics implied by the title and objectives of the program.

The program must demonstrate that graduates have: the ability to analyze, design, verify, validate, implement, apply, and maintain software systems; the ability to appropriately apply discrete mathematics, probability and statistics, and relevant topics in computer science and supporting disciplines to complex software systems; and the ability to work in one or more significant application domains.

ABET General Criteria

- Criterion 4 for all engineering programs states (in part):
  - "Students must be prepared for engineering practice through the curriculum culminating in a major design experience based on the knowledge and skills acquired in earlier course work and incorporating engineering standards and realistic constraints that include most of the following considerations: economic; environmental; sustainability; manufacturability; ethical; health and safety; social; and political."

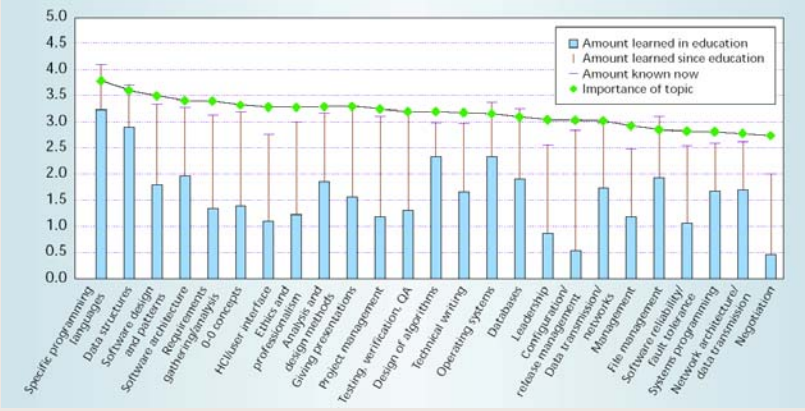
Skills/Professional Development

- "Skills development" concerns engineers developing practitioner skills by applying knowledge learned in their "initial professional education".
- "Professional development" concerns an SE becoming a "professional", and maintaining that status.
- ABET general graduate outcome:
  - a recognition of the need for, and an ability to engage in life-long learning

What SE Knowledge Is Important?

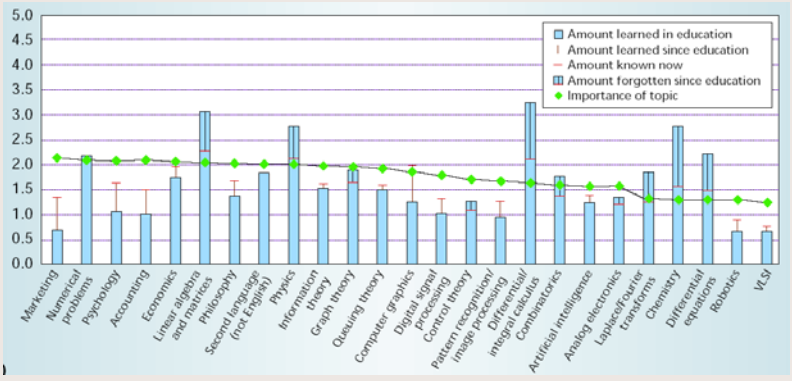
- From May to October 1998, Timothy Lethbridge [Lethbridge 2000] surveyed software professionals representing a wide variety of industries, job functions, and countries to learn which educational topics have proved most important to them in their careers and to identify the topics for which their education or current knowledge could be improved.
- Survey responses were received from186 software developers:
  - wide variety of backgrounds (most with degrees in CS, SE or IS)
  - from 42 countries (most from North America).

Most Important Topics



[Lethbridge 2000]

Least Important Topics



[Lethbridge 2000]

Certification and Licensing

- Certification is a voluntary process administered by a profession.
  - What are some examples of professional certification?
- Licensing is a mandatory process administered by a governmental authority.
  - What are some examples of professional licensing?

CSDP

- The IEEE computer Society has developed a certification program for software engineers, titled Certified Software development Professional (CSDP)
  - http://www.computer.org/certification/
- The CSDP credential is intended for individuals with 5 or more years of experience, and tests expertise in 11 essential software engineering knowledge areas:
  - Business Practices and Engineering Economics
  - Software Requirements
  - Software Design
  - Software Construction
  - Software Testing
  - Software Maintenance
  - Software Configuration Management
  - Software Engineering Management
  - Software Engineering Process
  - Software Engineering Tools and Methods
  - Software Quality
- an alternate view of certification: http://www.systemsguild.com/GuildSite/TDM/certification.html

SE Practices - Worldwide

- Cusumano, et. al. gathered data from 104 software projects,in four global regions, and analyzed their use of various SE practices [Cusumano 2003].

Practice / No. of Projects	India	Japan	US	Europe	Total
	24	27	31	22	104
Architectural Specification	83.3%	70.4%	54.8%	72.7%	69.2%
Functional Specification	95.8%	92.6%	74.2%	81.8%	85.6%
Detailed Design	100.0%	85.2%	32.3%	68.2%	69.2%
Code Generation	62.5%	40.7%	51.6%	54.5%	51.9%
Design Review	100.0%	100.0%	77.4%	77.3%	88.5%
Code Review	95.8%	74.1%	71.0%	81.8%	79.8%
Subcycles	79.2%	44.4%	54.8%	86.4%	64.4%
Beta Testing	66.7%	66.7%	77.4%	81.8%	73.1%
Pair Testing	54.2%	44.4%	35.5%	31.8%	41.3%
Pair Programming	58.3%	22.2%	35.5%	27.2%	35.3%
Daily Builds					
At the Start	16.7%	22.2%	35.5%	9.1%	22.1%
In the Build	12.5%	25.9%	29.0%	27.3%	24.0%
At the End	29.2%	37.0%	35.5%	40.9%	35.6%
Regression Testing	91.7%	96.3%	71.0%	77.3%	83.7%

SE Performance - Worldwide

- Using the same 104 projects, Cusumano, et. al. collected data about project productivity and quality [Cusumano 2003].

	India	Japan	US	Europe	Total
No. of Projects	24	27	31	22	104
LOC/programmer month	209	469	270	436	374
Defects/KLOC (12 mon. after delivery)	0.263	0.020	0.400	0.225	0.150

- Note: One has be careful about drawing general conclusions from such a study: the number of projects is small compared to the total number of worldwide projects; data reporting was voluntary; and there are other factors by which a project is judged successful or unsuccessful.

Code of Ethics

- In 1999, the *Software Engineering Code of Ethics and Professional Practice* (SE Code) was developed by by the ACM/IEEE-CS Joint Task Force on Software Engineering Ethics and Professional Practices and jointly approved by the ACM and the IEEE-CS as the standard for teaching and practicing software engineering. (http://www.acm.org/serving/se/code.htm)
- The preamble to the SE Code (short version) states:
  - Software engineers shall commit themselves to making the analysis, specification, design, development, testing and maintenance of software a beneficial and respected profession. In accordance with their commitment to the health, safety and welfare of the public, software engineers shall adhere to the following Eight Principles:

SE Code Principles

- PUBLIC - Software engineers shall act consistently with the public interest.
- 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.
- PRODUCT - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.
- JUDGMENT - Software engineers shall maintain integrity and independence in their professional judgment.
- MANAGEMENT - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
- PROFESSION - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.
- COLLEAGUES - Software engineers shall be fair to and supportive of their colleagues.
- 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.

Wisdom from an SE Philosopher

- Are we making progress?
  - "We're lost, but we're making good time."
  - "It's not too far, it just seems like it."
  - "You've got to be careful if you don't know where you're going 'cause you might not get there."

Ariane 5 Launch – June 4, 1996

Ariane 5 Accident 2

- The European Space Agency (ESA) spent 10 years and $7 billion to produce the Ariane 5, a giant rocket capable of launching a pair of three-ton satellites into orbit with each launch. It was intended to give Europe overwhelming supremacy in the commercial space business. [Gleick 1996]
- On its maiden launch, June 4, 1996, the unmanned Ariane 5 rocket, launched by the European Space Agency, exploded just forty seconds after its lift-off from Kourou, French Guiana.
  - The destroyed rocket and its cargo were valued at $500 million.
  - A board of inquiry investigated the causes of the explosion and in two weeks issued a report [ESA 1996]. It turned out that the cause of the failure was a software error in the inertial reference system.

Ariane 5 Case Study Exercise

- Team Exercise
  - Each team reads through the exercise booklet for the Ariane 5 case.
  - Each team completes a “professionalism” Case Analysis of the Ariane 5 accident.
  - Each team presents their analysis to the class. Their may be disagreement, so a minority report is possible.

References 1

- [ABET 2003] Accreditation Board for Engineering and Technology, *Criteria For Accrediting Engineering Programs* (Effective for Evaluations During the 2004-2005 Accreditation Cycle), November 2003.
- [ACM 1999] ACM/IEEE-CS Joint Task Force on Software Engineering Ethics and Professional Practices, Software Engineering Code of Ethics and Professional Practice (Version 5.2), 1999.
- [Bourque 2001] Bourque P. and Dupuis R., eds. *Guide to the Software Engineering Body of Knowledge*, IEEE CS Press, Los Alamitos, Calif., 2001. (http://www.swebok.org/)
- [Cusumano 2003] Cusumano, M., et. al., Software Development Worldwide: The State of the Practice, *IEEE Software*, November/December 2003.
- [ESA 1996] Ariane 5 Inquiry Board, *Ariane 5 Flight 501 Failure Report*, European Space Agency, July 19, 1996.

References 2

- [Gleick 1996] Gleick, James, A Bug and a Crash, *New York Times Magazine*, December1, 1996.
- [Ford 1996] Ford, G. and Gibbs, N. E., *A Mature Profession of Software Engineering*, CMU/SEI-96-TR-004, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 1996.
- [Jezequel 1997] Jezequel , J. and Meyer, B., Design by Contract: The Lessons of Ariane, Computer, January 1997.
- [Lethbridge 2000] Lethbridge, T., What Knowledge is Important to a Software Engineer?, *Computer*, Vol 33, No. 6, pp. 44-50, May 2000.
- [Shaw 1990] Shaw, Mary, Prospects for an Engineering Discipline of Software, *IEEE Software*, vol 7, no 6, November 1990, pp.15-24.
- [Wasserman 1996] Wasserman, Tony, Toward a Discipline of Software Engineering, *IEEE Software*, 13:6:23-31, Nov 1996.