

Software Engineering: The Profession

Tom Hilburn (hilburn@erau.edu)
Embry-Riddle Aeronautical University

Software Technologies Distinguished Lecture Series
Florida Gulf Coast University
College of Engineering
2/24/2011

Abstract

In this seminar we will discuss the software engineering profession, addressing the following questions: **What is software engineering and what is its current state?** What do software engineers do? (OR what should they be doing?) **What do we mean by the term “professional” and how does one become a professional software engineer?** (including topics in curriculum, accreditation, professional development, and “professionalism” issues) **What’s the future of software engineering?** (including topics in job market trends and what capabilities will be needed in the future)

What is Software Engineering? 1

- The establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines [Bauer 1972].
- The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software [IEEE 1990].
- Goal: produce quality software in a timely, cost-effective manner

3

What is Software Engineering? 2

Software Engineering \neq “just” Programming

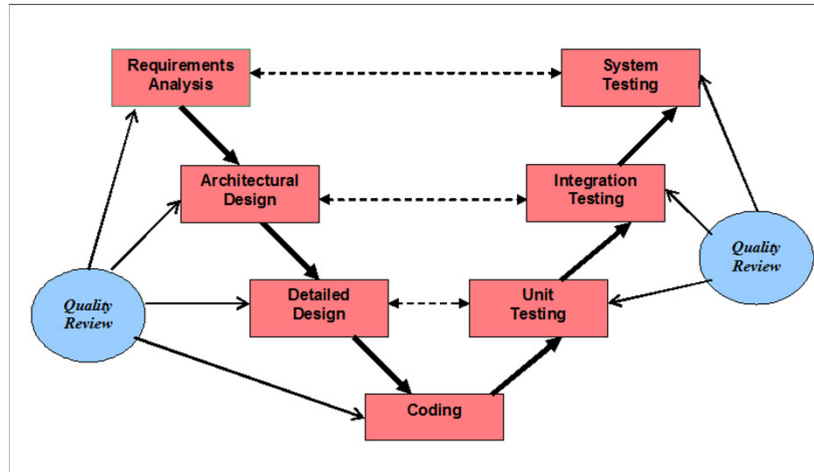
- Real Programmers (cowboy programmers)
 - know better than the users what they need.
 - thinks analysis/design is for wimp -- Nike mode is best ("Just do it").
 - don't draw pictures. Pictures are, after all, the illiterate's form of documentation. Cavemen drew pictures; look how much it did for them.
 - don't comment their code. If it was hard to write, it should be hard to read.
 - write programs that never work right the first time. But if you throw them on the machine they can be patched into working in only a few 30-hour debugging sessions.
 - never work 9 to 5. If any real programmers are around at 9 am, it's because they were up all night.
 - implement changes at will, without prior discussion with those whose work the decision will affect.

4

[Johnson 2003, Van Vleck 1982].

What is Software Engineering? 3

- The V Model of Software Engineering



5

State of the Practice

- Software is an important part of almost every aspect of human endeavor today.
- Software engineering methods and technology have advanced greatly.
- Professionals with software engineering knowledge and skills are in high demand.
- Unfortunately, serious problems still need to be addressed.
 - problems with schedule
 - problems with cost
 - problems with quality

6

Software Development Problems 1

- Software is becoming larger and more complex.
e.g., the code in consumer products is doubling every couple of years, e.g., NT 1993 (5 million LOC) → XP 2001 (45 million LOC)
- Many software products have poor quality.
e.g., Software testing typically finds as much as 25 defects per 1000 lines of code.
- Serious schedule overruns are common.
e.g., One in four large-scale development projects is never completed.
- Software development is costly.
e.g., one estimate is that embedded software costs \$15-\$40 per LOC; a study found that FAA's Advance Automation System cost over \$700 per line of code.

7

Software Development Problems 2

European Space Agency - Ariane 5 – June 4, 1996



8

Software Development Problems 3



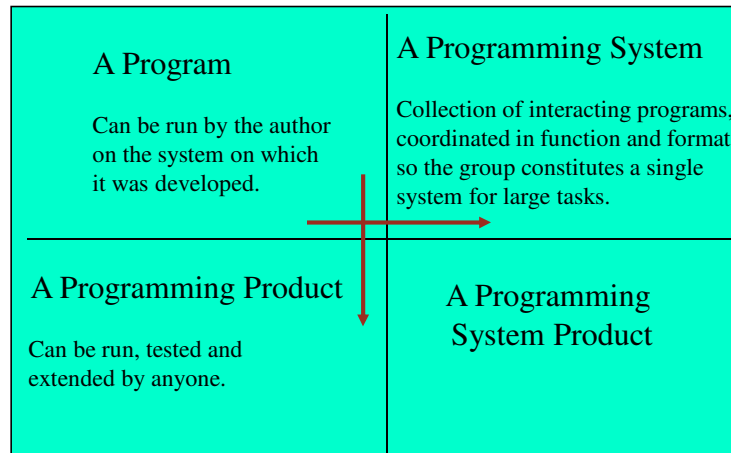
9

Why is good software so difficult to produce?

- The nature of software is different from other man-made products [Brooks 1995].
 - Software is complex.
 - Software is not visible.
 - Software does not have to conform to the laws of nature.
 - Software is easy to change.
- Software methods and technology is very dynamic.
- We need “system products” not individual programs.

10

Systems not Programs



[Brooks 1995]

11

Software Engineering Maturity

- Many do not understand the content or nature of software engineering.
- Although improving, the software engineering profession is still viewed as immature; and many organizations still have ad hoc software development processes.
- Some do not view software development as an “engineering” activity.
- There no universal agreement or acceptance about what constitutes “good software engineering practice”.

12

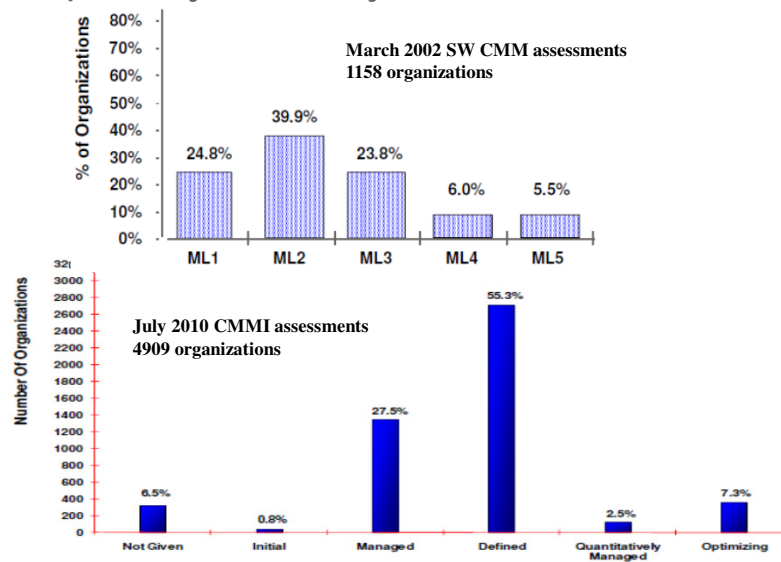
A Study of SwE Practices

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

13

[Cusumano 2003]

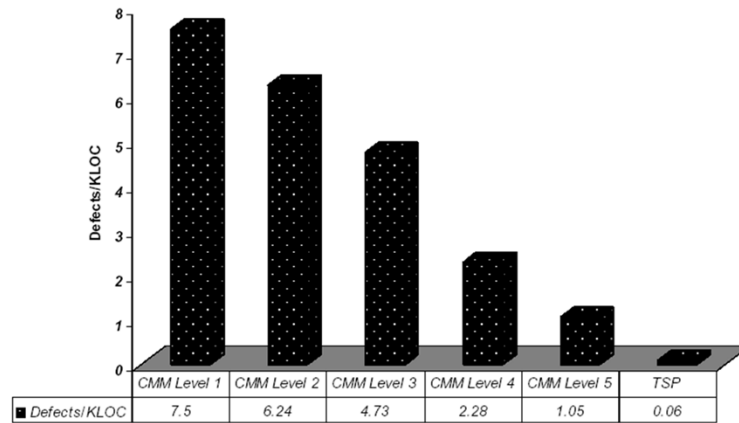
Capability Maturity Model Assessments



14

[SEI 2010]

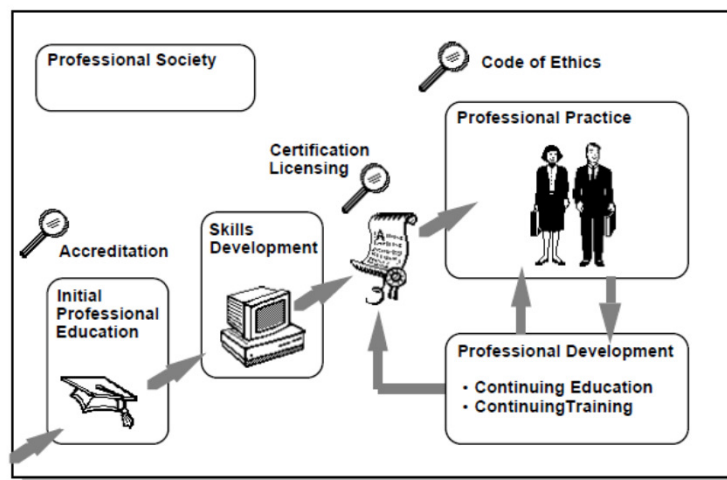
CMM Quality



[Davis 2003]

15

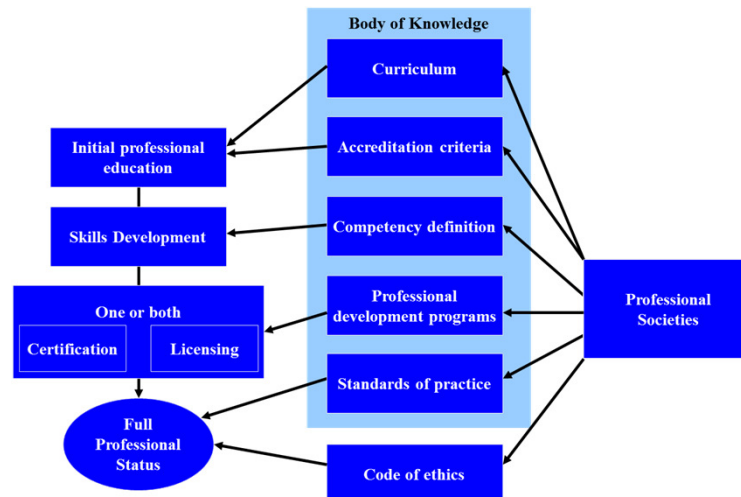
Advancing a Profession



[Ford 1996]

16

Advancing Software Engineering 1



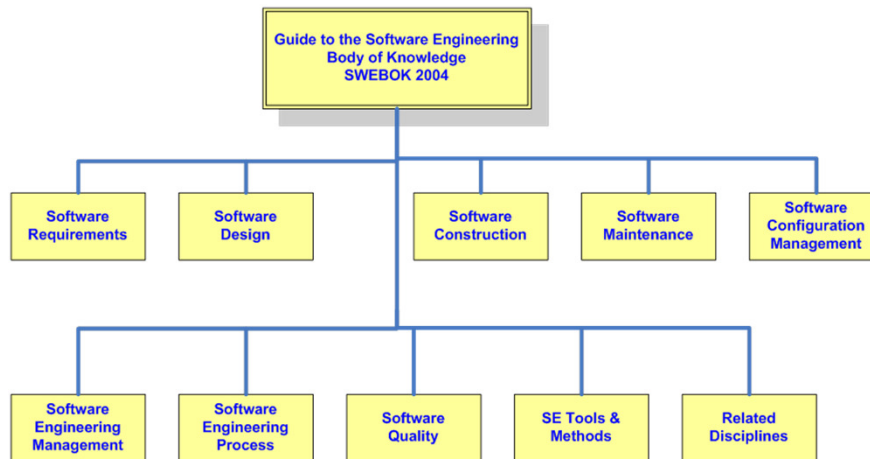
17

Advancing Software Engineering 2

- In recent years, task forces and working groups from the ACM, IEEE-CS, and the Software Engineering Institute have been at work to advance the state of software engineering.
 - Software Engineering Body of Knowledge (SWEBOK)
(<http://www.swebok.org/>)
 - Accreditation Criteria for SWE Programs
(<http://www.abet.org/>)
 - Software Engineering Code of Ethics and Professional Practice
(<http://www.acm.org/serving/se/code.htm>)
 - Undergraduate and Graduate Reference Curricula
(<http://www.computer.org/portal/web/education/Curricula>)
 - Certified Software Development Professional
(<http://www.computer.org/certification/>)

18

Advancing Software Engineering 3



19

[Bourque 2004]

Advancing Software Engineering 4

ABET Criteria For Software And Similarly Named Engineering Programs

- Lead Society: CSAB
- Cooperating Society: Institute of Electrical and Electronics Engineers
- These program criteria apply to engineering programs that include “software” or similar modifiers in their titles.
 - 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;
 - the ability to work in one or more significant application domains;
 - and the ability to manage the development of software systems.

(<http://www.abet.org/>)

20

Advancing Software Engineering 5

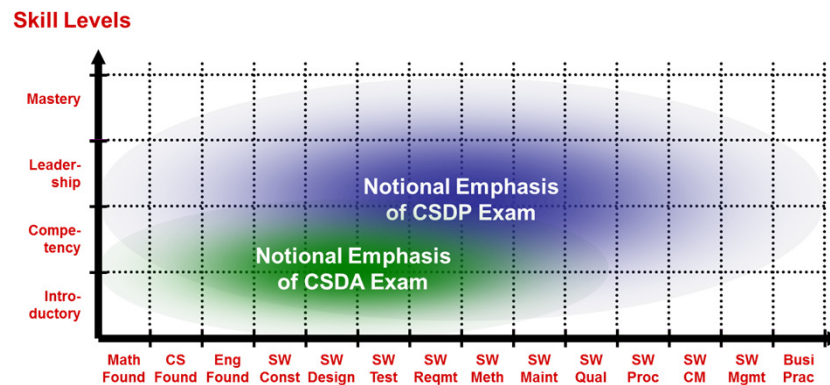


- Certified Software Development Professional (CSDP)
 - The program is intended for experienced software development and software engineering professionals.
 - Started in 2001 and launched in 2002
- Certified Software Development Associate (CSDA)
 - This program is intended for beginning software development and software engineering associates. CSDA is the first step towards becoming a CSDP.
 - Started in 2007 and launched in 2008.

21

Advancing Software Engineering 6

Relationships of CSDP and CSDA Certifications



22

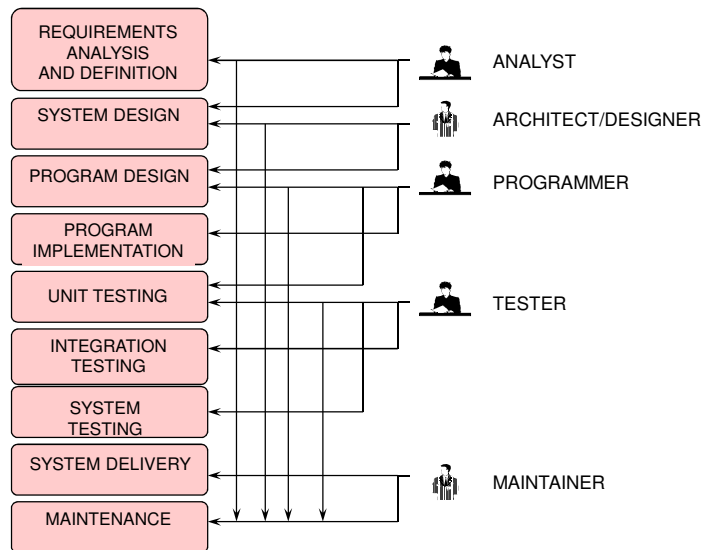
Advancing Software Engineering 7



- “You can observe a lot by just watching.”
- “you’ve got to be careful if you don’t know where you’re going ‘cause you might not get there.”
- “We’re lost, but we’re making good time.”
- “It’s not too far, it just seems like it.”

23

Software Engineering Jobs



24

The Future of the Profession 1

- Money Magazine – 2010 Top Jobs in the USA

Rank	Job title	Job growth (10-yr forecast)
1	Software Architect	34%
2	Physician Assistant	39%
3	Management Consultant	24%
4	Physical Therapist	30%
5	Environmental Engineer	31%
6	Civil Engineer	24%
7	Database Administrator	20%
8	Sales Director	15%
9	Certified Public Accountant	22%
10	Biomedical Engineer	72%

25

(<http://money.cnn.com/magazines/moneymag/bestjobs/2010/>, accessed 2/20/2011)

The Future of the Profession 2

- Google search of “<profession> jobs”

Job Type	Google Results
Software Engineer	1,940,000
Mechanical Engineer	957,000
Registered Nurse	928,000
Civil Engineer	718,000
Computer Scientist	496,000
Physician Assistant	341,000
Management Consultant	194,000
Certified Public Accountant	78,300

26

The Future of the Profession 3

- From the *Occupational Outlook Handbook, 2010-11 Edition* [BLS 2011]
 - Computer software engineers and computer programmers held about 1.3 million jobs in 2008. Approximately 514,800 were computer applications software engineers, about 394,800 were computer systems software engineers, and about 426,700 were computer programmers.
 - In May 2008, median annual wages of wage-and-salary computer applications software engineers were \$85,430.
 - Computer software engineers are among the occupations projected to grow the fastest and add the most new jobs over the 2008-18 decade, resulting in excellent job prospects.
 - Employment of computer programmers is expected to decline by 3 percent through 2018.
 - New growth areas will also continue to arise from rapidly evolving technologies: internet, mobile and wireless technology, cyber security. “Computer systems design and related services” will represent about 40% of jobs.

27

The Future of the Profession 4

- “The tar pit of software engineering will continue to be sticky for a long time to come . . . software systems are perhaps the most intricate of man’s handiworks. This complex craft will demand our continual development of the discipline, our learning to compose in larger units, our best use of new tools, our best adaptation of proven engineering management methods, liberal application of common sense, and a God-given humility to recognize our fallibility and limitations.”

[Brooks 1995]

- “Improvement requires change, and changing the behavior of software engineers is a nontrivial problem.”

Watts Humphrey, *CrossTalk*, February 1998

28

References

- [Bauer 1972] Bauer, F.L., Software Engineering, *Information Processing*, 71, 1972.
- [BLS 2011] *Occupational Outlook Handbook*, 2010-11 Edition, U.S. Bureau of Labor Statistics. (<http://www.bls.gov/oco/>, accessed 2/20/2011)
- [Bourque 2004] Bourque P. and Dupuis R., eds. *Guide to the Software Engineering Body of Knowledge*, IEEE CS Press, Los Alamitos, Calif., 2004. (<http://www.swebok.org/>, accessed 2/20/2011)
- [Brooks 1995] Brooks, Fred P., *The Mythical Man-Month*, 20th Anniversary Edition, Addison Wesley, 1995.
- [Cusumano 2003] Cusumano, M., et. al., Software Development Worldwide: The State of the Practice, *IEEE Software*, November/December 2003.
- [Davis 2003] Davis, N. and Mullaney, J., *The Team Software Process (TSP) in Practice: A Summary of Recent Results*, CMU/SEI-2003-TR-014, Software Engineering Institute, Carnegie Mellon University, September 2003.
- [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.
- [IEEE 1990] IEEE STD 610.12-1990, *IEEE Standard Glossary of Software Engineering Terminology*, IEEE Computer Society, 1990.
- [Johnson 2003] Johnson, L.J., *Cowboy Programmers*, Fawcette Technical Publications, June 2003. (http://www.mvps.org/st-software/LJ_Option.htm, accessed 2/17/2011)
- [SEI 2010] CMMI for Development, Version 1.3, CMU/SEI-2010-TR-033, Software Engineering Institute, Carnegie Mellon University, November 2010. (<http://www.sei.cmu.edu/reports/10tr033.pdf>, accessed 2/20/2011)
- [Van Vleck 1982] Van Vleck, Tom, Real Programmers Don't Write Specs, October 25, 1982. (<http://www.multicians.org/thvv/realprogs.html>, accessed 2/17/2011)

Questions?