Professional Practices HU-511(BSCS), HU-601(BSIT)

Dr. Arbab Waseem Abbas

ICS/IT, FMCS, The University of Agriculture, Peshawar.

Lecture 1(week1,2 & 3)

Lecture # 1
Professional Practices and Computing
Introduction

Course Introduction:

A Computing graduate as professional has some responsibilities with respect to the society. This course develops student understanding about historical, social, economic, ethical, and professional issues related to the discipline of Computing. It identifies key sources for information and opinion about professionalism and ethics. Students analyze, evaluate, and assess ethical and professional computing case studies.

CLO No.	Course Learning Outcomes	Bloom Taxonomy
-	-	-

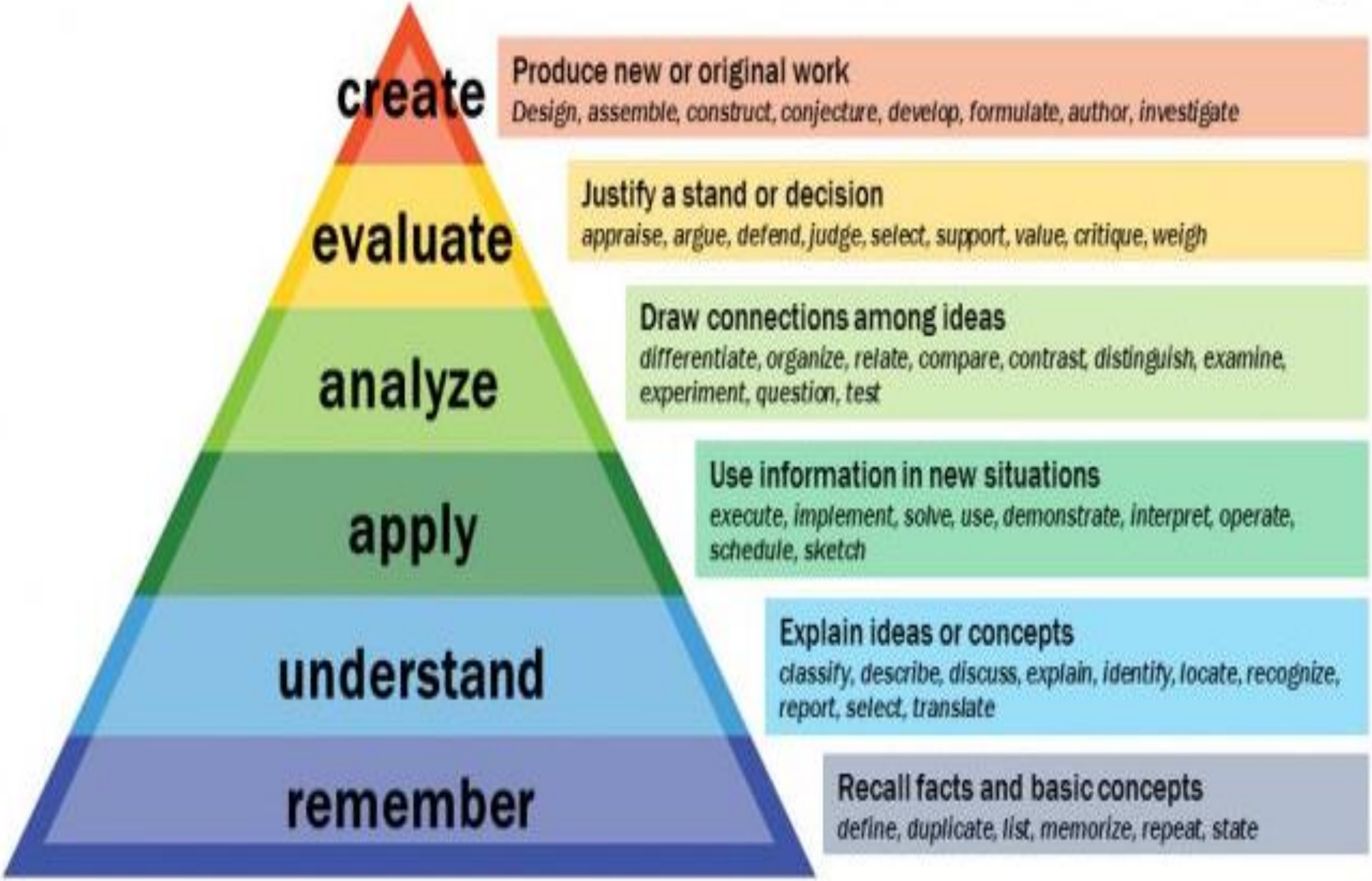
Course Outline:

Historical, social, and economic context of Computing (software engineering, Computer Science, Information Technology); Definitions of Computing (software engineering, Computer Science, Information Technology) subject areas and professional activities; professional societies; professional ethics; professional competency and life-long learning; uses, misuses, and risks of software; information security and privacy; business practices and the economics of software; intellectual property and software law (cyber law); social responsibilities, software related contracts, Software house organization. Intellectual Property Rights, The Framework of Employee Relations Law and Changing Management Practices, Human Resource Management and IT, Health and Safety at Work, Software Liability, Liability and Practice, Computer Misuse and the Criminal Law, Regulation and Control of Personal Information. Overview of the British Computer Society Code of Conduct, IEEE Code of Ethics, ACM Code of Ethics and Professional Conduct, ACM/IEEE Software Engineering Code of Ethics and Professional Practice. Accountability and Auditing, Social Application of Ethics.

Reference Materials: (or use any other standard and latest books)

1. Professional Issues in Software Engineering by Frank Bott, Allison Coleman, Jack Eaton and Diane Rowland, CRC Press; 3rd Edition (2000). ISBN-10: 0748409513
2. Computer Ethics by Deborah G. Johnson, Pearson; 4th Edition (January 3, 2009). ISBN-10: 0131112414
3. A Gift of Fire: Social, Legal, and Ethical Issues for Computing and the Internet (3rd Edition) by Sara Baase, Prentice Hall; 3rd Edition (2008). ISBN-10: 0136008488
4. Applied Professional Ethics by Gregory R. Beabout, University Press of America (1993). ISBN-10: 0819193747.

Bloom's Taxonomy



create

Produce new or original work

Design, assemble, construct, conjecture, develop, formulate, author, investigate

evaluate

Justify a stand or decision

appraise, argue, defend, judge, select, support, value, critique, weigh

analyze

Draw connections among ideas

differentiate, organize, relate, compare, contrast, distinguish, examine, experiment, question, test

apply

Use information in new situations

execute, implement, solve, use, demonstrate, interpret, operate, schedule, sketch

understand

Explain ideas or concepts

classify, describe, discuss, explain, identify, locate, recognize, report, select, translate

remember

Recall facts and basic concepts

define, duplicate, list, memorize, repeat, state

Requirement for Semester Project

Option 1: A complete research project

- Introduction (problem formulation/definition)
- Literature review
- The proposed method and analysis/Methodology
- Experiment/Results and Discussion
- Conclusion
- References

Option 2: A survey research

- A well-defined problem or topic
- A complete list of previous (typical) work on this problem (15+ papers under the topic)
- Clearly and briefly describe the topic
- Analyze each method/group and compare them
- Give the conclusion and list of references

Requirement for Semester Project

Requirements

•Select a topic and write a one-page proposal

•Progress report (discuss with the instructor)

•Research work and report writing

•Oral presentation

– in class presentation

– prerecorded video

•Final project report

Requirement for Final Project

Teamwork is acceptable for a research project (Option 1)

- <=2 people
- Get the permission from the instructor first
- Under a single topic, each member must have their own specific tasks
- One combined report with each member clearly stating their own contributions
- One combined presentation

Requirement for Semester Project

Written report

- Report format: the same as a conference paper
    - https://www.scribbr.com/ieee/ieee-paper-format/
- Executable code must be submitted with clear comments except for a survey study

Academic integrity (avoiding plagiarism)

- don’t copy other person’s work
- describe using your own words
- complete citation and acknowledgement whenever you use any other work (either published or online)

Requirement for Semester Project

Evaluation

- written report (be clear, complete, correct, etc.)
- code (be clear, complete, correct, etc.)
- oral presentation
- discussion with the instructor
- quality: publication-level project – extra credits

Requirement for Semester Project

Notes:

•You are encouraged to incorporate your own research expertise in, but the project topic must be related to the content of this course

•Discuss with the instructor on topic selection, progress, writing, and presentation

•Use the library and online resource

Paper Reading and Presentation

- A paper picked by yourself and approved by the instructor
  - Suggested paper source: To be provided
- Thorough understanding of the paper
- Prepare PPT slides
  - Clearly explain the main contributions in the selected paper
  - Critical comments – extra credit
- About 10 mins oral presentation for each student
  - – in class presentation
  - – prerecorded video

Proposal of Semester Project

Include

- Title and names of the team member
- Topic: a research project or a survey
- Brief introduction on the background
- Timeline and project management for a teamwork

At most one page

Each team only needs one abstract

On the Paper Reading (Both Sections)

includes:

- The paper you are going to present
  - Title, authors, where and when it was published, pages
  - Example: Sing Bing Kang, Ashish Kapoor, Dani Lischinski , “Personalization of Image Enhancement ”, in *Proceedings of IEEE Conference on computer vision and Pattern Recognition (CVPR)*, 2010

I will provide feedback (approve/suggest to change) to your selected paper

Where to Find the Paper

The paper you choose must be published in an official journal or conference!

A journal paper is preferred!

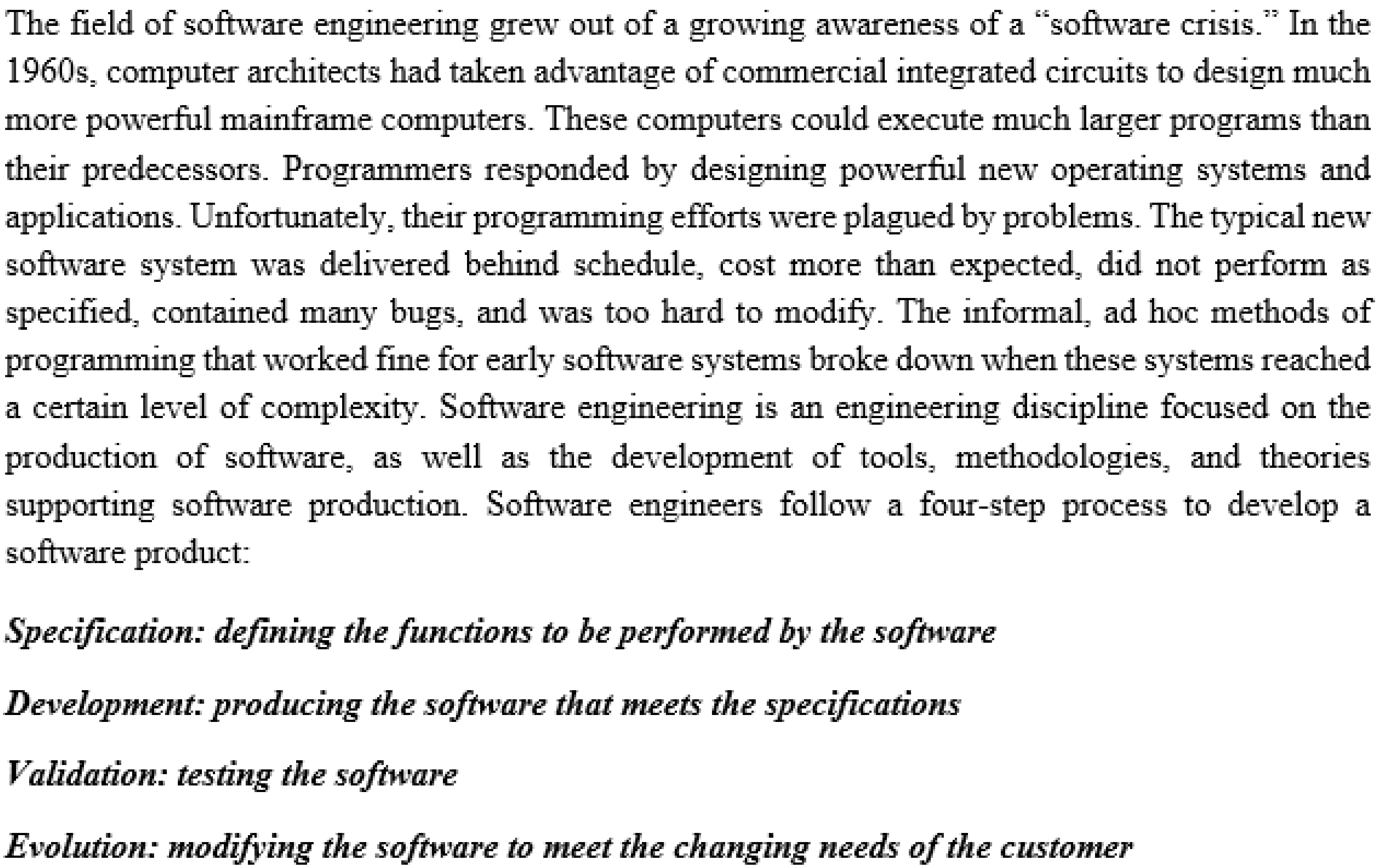
You can find papers from journals
IEEE Transactions on Pattern Analysis and Machine Intelligence
http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?reload=true&punumber=34
IEEE Transactions on Image Processing http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=83

Other premier conferences or journals, CVPR, ICCV, ECCV, IEEE Trans. Medical Imaging …

- Professional Practice is defined as, where a student is required to extend his/her knowledge and skills in a practical environment.

Software Engineering

Summary



The field of software engineering grew out of a growing awareness of a "software crisis." In the 1960s, computer architects had taken advantage of commercial integrated circuits to design much more powerful mainframe computers. These computers could execute much larger programs than their predecessors. Programmers responded by designing powerful new operating systems and applications. Unfortunately, their programming efforts were plagued by problems. The typical new software system was delivered behind schedule, cost more than expected, did not perform as specified, contained many bugs, and was too hard to modify. The informal, ad hoc methods of programming that worked fine for early software systems broke down when these systems reached a certain level of complexity. Software engineering is an engineering discipline focused on the production of software, as well as the development of tools, methodologies, and theories supporting software production. Software engineers follow a four-step process to develop a software product:

Specification: defining the functions to be performed by the software

Development: producing the software that meets the specifications

Validation: testing the software

Evolution: modifying the software to meet the changing needs of the customer

Objectives

- To introduce software products, software engineering and to explain their importance
- To set out the answers to several key questions about software engineering
- To introduce ethical (principled) and professional issues (matters) and to explain why they are of concern to software engineers

Software engineering

- The economies of ALL developed nations are dependent on software
- More and more systems are software controlled
- Software engineering is concerned with theories, methods and tools for professional software development

Software costs

- Software costs often dominate system costs. The costs of software on a PC are often greater than the hardware cost
- Software costs more to maintain than it does to develop. For systems with a long life, maintenance costs may be several times development costs
- Software engineering is concerned with cost-effective software development

What is software?

- Computer programs and associated documentation (plus configuration data and user training)
- SW is not only *programs* but also all *associated documentation*, and *configuration data* that make these programs operate correctly.
- More specifically, a SW system consists of
  - *separate programs*
  - *configuration files* setting up these programs
  - *system documentation* describing the structure of the system in good detail
  - *user documentation* explaining how to use and operate the system.

Two Classes of SW Products

Software products may be developed for a particular customer or developed for a general market

- Generic (shrink-wrapped) - developed to be sold to a range of different customers
- Bespoke (custom) - developed for a single customer according to their specification

What are the attributes of good software?

- The software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable
- Maintainability
  - Software must evolve to meet changing needs
- Dependability
  - Software must be trustworthy
- Efficiency
  - Software should not make wasteful use of system resources
- Usability
  - Software must be usable by the users for which it was designed

…attributes of good software

(these two are not always required)

- Robustness
  - Software should fail only under extreme conditions
- Portability
  - Should be possible to move from one environment to another

The software crisis

- Advances in hardware technologies made it possible to build powerful computers
  - This allowed building of more complex and powerful software
- Existing software development methodologies were not capable of handling such large projects.
- Hence projects had many problems:
  - Over budget
  - Late delivery
  - Requirements not met
  - Poor usability

What is software engineering?

- Software engineering is an engineering discipline which is concerned with all aspects of software production (Sommerville, 2001)
- Software engineers should adopt a systematic and organised approach to their work and use appropriate tools and techniques depending on the problem to be solved, the development constraints and the resources available

What is the difference between software engineering and system engineering?

- System engineering is concerned with all aspects of computer-based systems development including hardware, software and process engineering. Software engineering is part of this process
- System engineers are involved in system specification, architectural design, integration and deployment

Why does a software engineer need to understand system engineering aspects?

- Many software systems are part of a larger system
- System engineering decisions have direct impacts on software
- Many systems now have lots of software parts

Computer Science vs Information Technology

- Though there are many overlaps between computer science and IT, computer science work generally means designing and building computers and computer programs. IT work deals more with running the computers of an organization, and making sure the related networks, systems, and security are operational.
- Computer scientists design and develop the software programs that IT professionals use and maintain. IT specialists take what they've learned about computers and apply that knowledge in businesses across all industries.

What is a software process?

- A set of activities and associated results whose goal is the development or evolution of a software product
- Generic (general) activities in all software processes are:
  - Specification - what the system should do and its development constraints
  - Development - production of the software system
  - Validation - checking that the software is what the customer wants
  - Evolution - changing the software in response to changing demands

Specification

- Determine system requirements
- Understand constraints
- Determine feasibility
- End products
    - High-level statement of requirements
    - Mock-up of user interface
    - Low-level requirements statement

Development

- Create high-level design
- Discover and resolve mistakes, omissions in specification
- CASE tools to support design process
- Object-oriented systems have advantages
- After detailed design, actual programs written
- Result: working software system

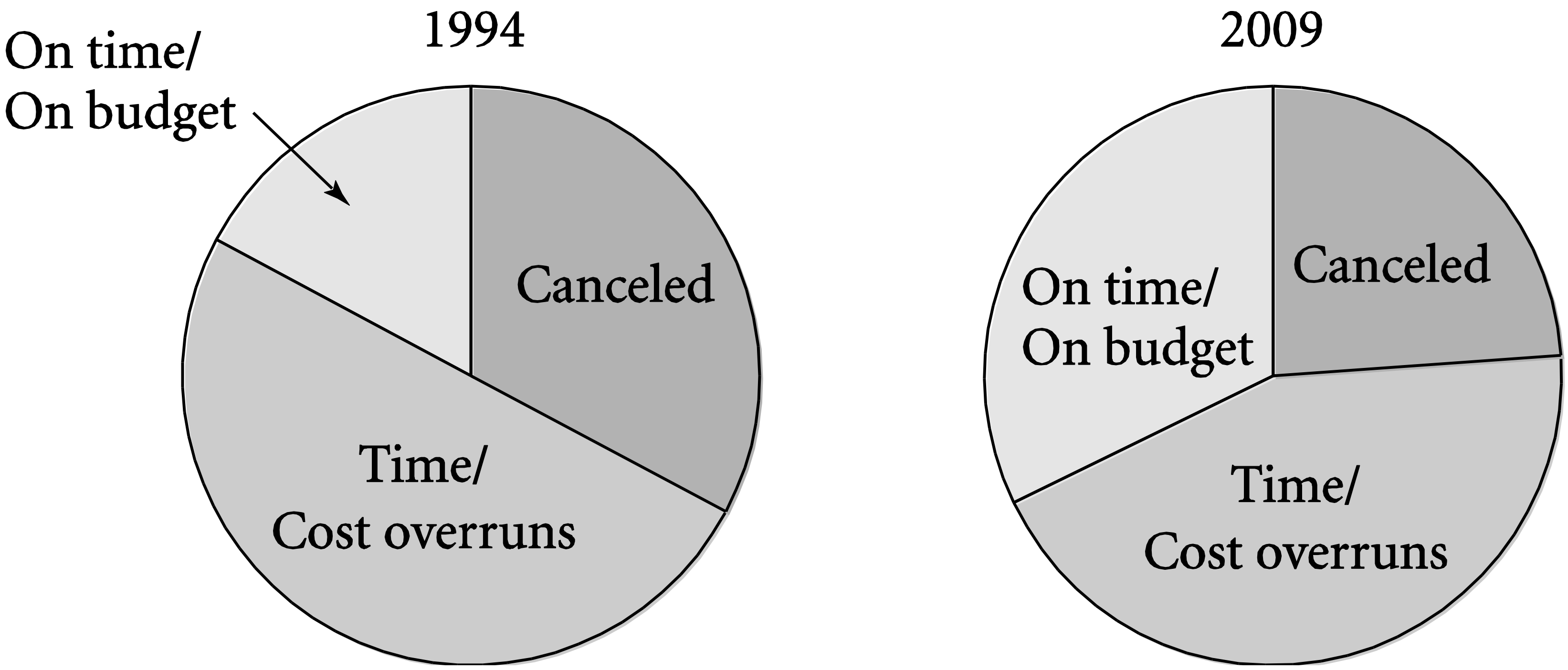
Validation (Testing)

- Ensure software satisfies specification
- Ensure software meets user’s needs
- Challenges to testing software
  - Noncontinuous responses to changes in input
  - Exhaustive testing impossible
  - Testing reveals bugs, but cannot prove none exist
- Test modules, then subsystems, then system

Software Quality Is Improving

- Standish Group tracks IT projects
- Situation in 1994
  - 1/3 projects cancelled before completion
  - 1/2 projects had time and/or cost overruns
  - 1/6 projects completed on time and on budget
- Situation in 2009
  - 1/4 projects cancelled
  - 5/12 projects had time and/or cost overruns
  - 1/3 projects completed on time and on budget

Success of IT Projects Over Time



1-35

8.8 Software Warranties and Vendor Liability

Shrinkwrap Warranties

- Some say you accept software "as is"
- Some offer 90-day replacement or money-back guarantee
- None accept liability for harm caused by use of software

Are Software Warranties Enforceable?

- Mass-marketed software and software included in sale of hardware likely to be considered a good by a court of law
- Uniform Commercial Code applies to goods, despite what warranties may say

Key Court Cases

- Step-Saver Data Systems v. Wyse Technology and the Software Link
  - Court ruled that provisions of UCC held
- ProCD v. Zeidenberg
  - Court ruled shrinkwrap licenses are enforceable
- Mortenson v. Timberline Software
  - Court ruled in favor of Timberline and licensing agreement that limited consequential damages

Moral Responsibility of Software Manufacturers

- If vendors were responsible for harmful consequences of defects
  - Companies would test software more
  - They would have to purchase liability insurance
  - Software would cost more
  - Start-ups would be affected more than big companies
  - Less innovation in software industry?
  - Software would be more reliable?
- Making vendors responsible for harmful consequences of defects may be a bad idea, but…
- Consumers should not have to pay for bug fixes

What is CASE ?
(Computer-Aided Software Engineering)

- Software systems which are intended to provide automated support for software process activities. CASE systems are often used for method support
- Upper-CASE
  - Tools to support the early process activities of requirements and design
- Lower-CASE
  - Tools to support later activities such as programming, debugging and testing

What are the costs of software engineering?

- Roughly 60% of costs are development costs, 40% are testing costs. For custom software, evolution costs often exceed development costs
- Costs vary depending on
  - the type of system being developed and
  - the requirements of system attributes such as performance and system reliability
- Distribution of costs depends on the development model that is used

What are the key challenges facing software engineering?

- Coping with legacy (old) systems, coping with increasing diversity (variety) and coping with demands for reduced delivery times
- Legacy systems
  - Old, valuable systems must be maintained and updated
- Heterogeneity
  - Systems are distributed and include a mix of hardware and software
- Delivery
  - There is increasing pressure for faster delivery of software

Professional and ethical responsibility

- Software engineering involves wider responsibilities than simply the application of technical skills.
- Software engineers must behave in an honest and ethically (morel, principled) responsible way if they are to be respected as professionals.

Issues of professional responsibility

- *Confidentiality*
  - Engineers should normally respect the confidentiality (privacy) of their employers or clients irrespective of whether or not a formal confidentiality agreement has been signed.
- *Competence*
  - Engineers should not misrepresent their level of competence (capability). They should not knowingly accept work which is outside their competence.

Issues of professional responsibility

- *Intellectual property rights*
  - Engineers should be aware of local laws governing the use of intellectual (scholar) property such as patents (exclusive rights), copyright, etc. They should be careful to ensure that the intellectual property of employers and clients is protected.
- *Computer misuse*
  - Software engineers should not use their technical skills to misuse other people's computers. Computer misuse ranges from relatively trivial (game playing on an employer's machine, say) to extremely serious (dissemination (distribution) of viruses).

Ethical dilemmas

- Disagreement in principle with the policies of senior management
- Your employer acts in an unethical way and releases a safety-critical system without finishing the testing of the system
- Participation in the development of military weapons systems or nuclear systems