Software engineering

Cliff notes version
Software engineering

- Software is used by many people (and systems) to accomplish many tasks.

- Software engineering applies engineering principles (building stuff that works) to computer programs.

- Two main players:
  - Engineers, who have a responsibility to build reliable software that does no harm.
  - Users, who are only concerned with their own expectations and tasks.
One definition

  - The application of a systematic, disciplined, quantifiable approach to development, operation, and maintenance of software; that is, the application of engineering to software.
Some textbook definitions

- # Ian Sommerville, Software Engineering, 5th edition. The specification, development, management, and evolution of software systems. Theories, methods, and tools needed to develop software.

- # Stephen R. Schach, Software Engineering, 2nd Edition. A discipline whose aim is the production of quality software, delivered on time, within budget, and satisfying users' needs.

- # Shari Lawrence Pfleeger, Software Engineering: the Production of Quality Software, 2nd Edition. Designing and developing high-quality software. Application of computer science techniques to a variety of problems. We are problem-solvers rather than theoreticians.
CSE vs. software engineering

- Computer science is concerned with theory and fundamentals; software engineering is concerned with the practicalities of developing useful software.

- Computer science theories are currently insufficient to act as a complete underpinning for software engineering (or so people say).
Hobby projects

- Garden
- Tree house
- Go kart
Engineering projects

- Industrial agriculture
- Skyscraper
- Race car
Ariane 5

http://www.youtube.com/watch?v=kYUrqdUyEpl&feature=related

“Ariane 5's first test flight (Ariane 5 Flight 501) on 4 June 1996 failed, with the rocket self-destructing 37 seconds after launch because of a malfunction in the control software, which was arguably one of the most expensive computer bugs in history. A data conversion from 64-bit floating point to 16-bit signed integer value had caused a processor trap (operand error). The floating point number had a value too large to be represented by a 16-bit signed integer. Efficiency considerations had led to the disabling of the software handler (in Ada code) for this trap, although other conversions of comparable variables in the code remained protected.” - wikipedia
2003 blackout
What is software?

- Computer software is composed of:
  - Programs
  - Documentation
  - Data required to build the system
Software in CSE20212

- Data
  - Lab handout
  - Requirements (aka the rubric)

- Documentation
  - “Code is expected to be well-commented”
  - Required report.txt each week

- Program
  - Executable generated from your source code using make
Chicken or the egg?

- Software is both a product and a vehicle to developing a program

- Software is built (often custom-built), not manufactured.

- Software does “deteriorate” with changing hardware/computing platforms.
CASE

- CASE stands for Computer Aided Software Engineering; it can be used to mean any computer-based tool for software planning, development, and evolution.

- Includes Editors, Compilers, Debuggers, Edit-Compile-Debug environments, Code Generators, Documentation Generators, Configuration Management, Release Management, Project Management, Scheduling, Tracking, Requirements Tracing, …
Examples of software

- System software
- Real-time software
- Web-based software
- Artificial intelligence software
- Scientific software
- Business software
- ...
In general, the successes outnumber failures over the past few decades.

Even so, problems encountered are related to developing and supporting software, rather than building “correct” functions.
“Field of dreams”
Some myths

- A general statement is all that is needed
  - Constant communication between customers and developers is needed to ensure satisfaction

- Once a program is written, it is done
  - Maintaining a program is never done until the product is discontinued.
More myths

- Software standards provide sufficient guidance
  - May be outdated and seldom used

- Adding people is a good way to catch up on a project
  - Only works with proper planning and coordination.
Why documentation

Some people think (maybe you?) software engineering is all about creating large and unnecessary documentation and comments.

Advantages:

- Provides the basis of support after completion.
- Results in things being done right the first time, limits additional work needed to complete or maintain software.
Process overview

- Building high quality software requires software process.

- A software process provides a framework for managing activities that may get out of control.

- Different projects require different software processes.
“Proof is in the pudding”

- Remember: products are programs, documentation and data

- However, the best indication of success is how well a tool works
Phases

- There are a few generic phases of software engineering:
  - Definition - focuses on what
    - Planning, requirements analysis
  - Development - focuses on how
    - Design, code generation, testing
  - Validation - does it work?
    - Unit testing, etc.
  - Support - focuses on change
    - Maintenance and all its forms
Typical framework

- Software engineering work tasks
- Project milestones
- Products
- Quality assurance
Example models: waterfall

- Requirements
- Design
- Implementation
- Testing
- Maintenance
Example models: Prototyping
Rapid development

- Uses many reusable components
- Short development cycle
- May use scripting languages for added speed to test a prototype
Spiral model
Other types

- Concurrent development
  - Used in client/server, similar to spiral
- Formal methods
  - Uses rigorous mathematical notation
- Incremental model
  - Small but usable pieces
What are the key challenges facing software engineering?

- Coping with legacy systems, coping with increasing diversity 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
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