Software Engineering

Lecture-1

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.

What is Software Engineering?

- Software engineering is an engineering discipline which is concerned with all aspects of software production.
- Software engineers should
  - adopt a systematic and organized approach to their work
  - use appropriate tools and techniques

Classic Definition

- *"The establishment and use of sound engineering principles in order to obtain software that is reliable and works efficiently on real machines."*
- *The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; i.e., the application of engineering to software.*

Product vs. Process

- Software has a dual role. It is a product and a process.
- Software is a product
  - Delivers computing potential.
  - Produces, manages, acquires, modifies, displays, or transmits information
- Software is a vehicle for delivering a product
  - Supports or directly provides system functionality
  - Controls other programs (e.g., an operating system)
  - Affects communications (e.g., networking software)
  - Helps build other software (e.g., software tools)

Characteristics of Software

- Software is engineered, not manufactured.
- Software doesn't wear out, it declines, because of changes.
- Software is complex

Where to use?

- Software is used in many applications:
  - system software
  - application software
  - engineering/scientific software
  - Real-time, embedded software
  - product-line software
  - Web applications
  - AI software

Software Engineering vs. Computer Science

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

Software Engineering vs. 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

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 a Software Process?

- A set of activities whose goal is the development or evolution of software.
- Generic 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

How Software changes?

- Software must be adapted to meet the needs of new computing environments or technology.
- Software must be enhanced to implement new business requirements.
- Software must be extended to make it interoperable with other more modern systems or databases.
- Software must be re-architected to make it feasible within a network environment.

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

Software Myths

- There are certain myths associated with software, which affect managers, customers (and other non-technical stakeholders) and practitioners.
- These myths are believable because they often have elements of truth, **but** invariably lead to bad decisions.
- **Therefore,** insist on reality as you navigate your way through software engineering.

Managment Myths

- Managers with software responsibility are often under pressure to maintain budgets, keep schedules from slipping, and improve quality.
- This pressure leads the manager to believe in a software myth assuming it would lessen the pressure.
  - A book full of standards and procedures provide everything we need.
  - If we get behind schedule, we can add programmers.

Customer Myths

- Often customers do not have sufficient information about the software, and neither do the managers and practitioners correct their misinformation.
- This causes them to believe in certain myths which unfortunately lead to false expectations and dissatisfaction with the developer.
  - To begin programming, a general statement of objectives is sufficient, we can fill the details later.
  - Project requirements continually change, but change can be easily accommodated because software is flexible.

Practitioner's Myths

- Practitioners also believe in certain myths about software and its engineering process.
  - Writing the program and get it to work, then our job is done.
  - Until we get the program running, we have no way of assessing the quality.
  - The only deliverable work product is the working program.

Issues of Professional Responsibility

- *Confidentiality*
  - Engineers should normally respect the confidentiality 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. They should not knowingly accept work which is beyond their competence.

Issues of Professional Responsibility

- *Intellectual property rights*
  - Engineers should be aware of local laws governing the use of intellectual property such as patents, 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 of viruses).

Key Points

- Software engineering is an engineering discipline which is concerned with all aspects of software production.
- Software products consist of developed programs and associated documentation. Essential product attributes are maintainability, dependability, efficiency and usability.
- The software process consists of activities which are involved in developing software products. Basic activities are software specification, development, validation and evolution.

Key Points

- Software Methods are organised ways of producing software. They include suggestions for the process to be followed, the notations to be used, rules governing the system descriptions which are produced and design guidelines.
- Software engineers have responsibilities to the engineering profession and society. They should not simply be concerned with technical issues.
- Professional societies publish codes of conduct which set out the standards of behaviour expected of their members.

Reading

- Chapter-1.