Software Engineering

3rd lecture

Lecture By: Zakirullah Ezam

Key challenges facing Software Engineering

Software Engineering in the 21st century faces three key challenges . Heterogeneity, delivery and trust .

- **Heterogeneity:** Developing techniques for building software that can cope with heterogeneous platforms and execution environments.
- **Delivery:** Developing techniques that lead to faster delivery of software.

Continue……

- **Trust:** Developing techniques that demonstrate that software can be trusted by its users.

According to that we said.

Coping with increasing diversity .

Demands for reduced delivery times.

And developing trustworthy software.

The best Software Engineering Techniques and method

While all software projects have to be professionally managed and developed, different techniques are appropriate for different types of system. For example games should always be developed using a series of prototypes whereas safety critical control systems require a complete and analyzable specification to be developed. Therefore we cannot say that one method is batter than another.

Continue…

- Structured approaches to software development which include system models, notations, rules, design advice and process guidance.
- **Model descriptions:** Descriptions of graphical models which should be produced.
- **Rules:** Constraint applied to system models.
- **Recommendations:** Advice on good design practice.
- **Process guidance:** What activities to follow.

The attributes of good software

The software should deliver the required functionality and performance to the user and should be maintainable, dependable and acceptable.

- **Maintainability:** Software must evolve to meet changing needs.
- **Dependability :** Software must be trustworthy .
- **Efficiency:** Software should not make wasteful use of system resources.
- **Acceptability :** Software must accepted by the users for which it was designed.

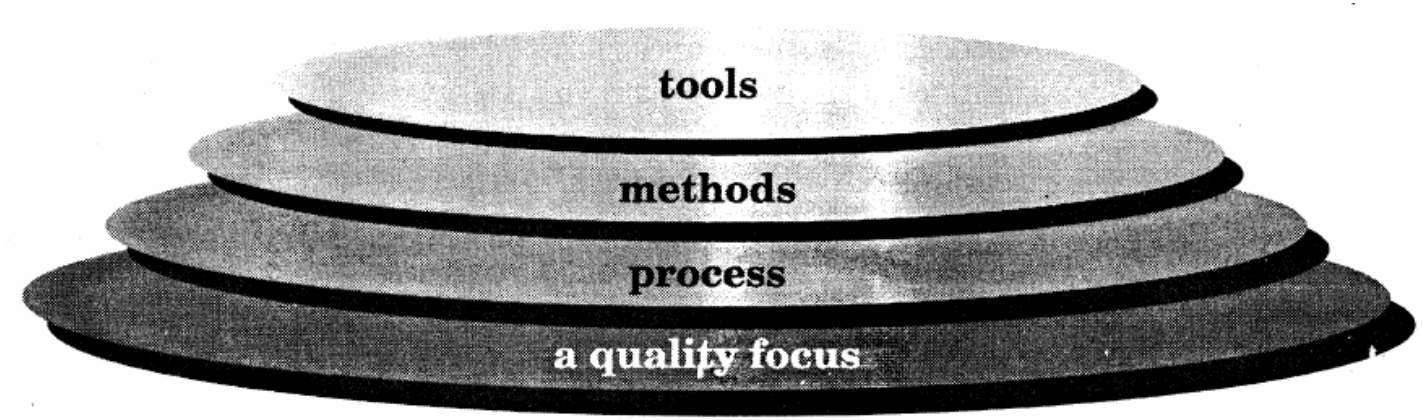
This means it must be understandable, usable and compatible with other systems.

Process, Methods and Tools.

- Software Engineering is a layered technology.
- Any engineering activity must rest on the organizational commitment to quality therefore the foundation layer of software engineering is the focus on quality.

Process

Defines key process areas that enables the rational and timely development of computer software.



Process, Methods and Tools.

- Key process areas forms the basis for management control of software projects:
  - Which technical methods are applied.
  - How key products (models, documents, reports etc) are produced.
  - How milestones are established.
  - How quality is ensured.
  - How changes are managed.

Methods

Methods includes a broad array of tasks that includes requirements analysis, design, program construction, testing and maintenance.

Each method includes modeling activates and other descriptive techniques.

Continue

- **Tools**
  - Instruments or automated systems used to support the process and methods.
  - Editors, Compilers, Drawing software (AutoCAD, Visio, MS-Project Management), Computer Aided Software Engineering (CASE).

What is CASE (Computer-Aided Software Engineering)

- Software systems that 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.

Generic View of Software Engineering

The work that is associated with software engineering can be categorized into three generic phases, regardless of the application area, project size and complexity.

➢ Definition Phase

Definition phase focuses on “What”, that is, during definition phase a software developer attempts to identify:

- What information is to be processed.
- What function and performance is desired .
- What system behavior can be expected .
- What interfaces are to be established.
- What design constraints exists.
- What validation criteria are required.
- That is the key requirements of the system and the software are identified.

Generic View of Software Engineering.

Development Phase

The development phase focuses on “How”, that is, during development phase a software engineer attempts to define:

- How data are to be structured.
- How function are to be implemented as a software architecture.
- How procedural details are to be specified.
- How interfaces are to be characterized.
- How design will be translated into a programming language.
- How testing will be performed.

- The methods applied during development phase will vary, but three specific task should always occur:
  - Software Design.
  - Code Generation.
  - Software Testing.

Generic View of Software Engineering.

Maintenance Phase

- The maintenance phase focuses on changes.
- The maintenance phase reapplies the steps of the definition and development phases, but does so in the context of the existing software.

Four types of changes are encountered during the maintenance phase:

Correction

- Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software.
- Corrective maintenance changes the software to correct errors detected.

Adaptation

- Over time, the original environment .e.g. CPU, OS, business rules etc for which the software was developed is likely to change.
- Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

Generic View of Software Engineering.

- **Enhancement**
  - As software is used, the customer/organization will recognized additional functions that will provide benefit.
  - Enhancement maintenance extends the software beyond its original functional requirements.
- **Prevention**
  - Software deteriorates due to change, Preventive maintenance enable the software to serve the needs of its end users.
  - Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Generic View of Software Engineering.

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 responsible way if they are to be respected as professionals.
- Ethical behavior is more than simply upholding the law.

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 out with their competence.
- **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) to extremely serious (dissemination of viruses).

IEEE Code of Ethics

- The professional societies in the US have cooperated to produce a code of ethical practice.
- Members of these organizations sign up to the code of practice when they join.
- The Code contains eight Principles related to the behavior of members and decisions made by professional software engineers, including practitioners, educators, managers, supervisors and policy makers, as well as trainees and students of the profession.

Code of ethics - preamble

- **Preamble:** The short version of the code summarizes aspirations at a high level of the abstraction, the clauses that are included in the full version give examples and details of how these aspirations change the way we act as software engineering professionals. Without the aspirations, the details can become legalistic and tedious; without the details, the aspirations can become high sounding but empty; together, the aspirations and the details form a cohesive code.

- 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:

Code of ethics - 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.

Principle Continues..

- **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

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.

Key points

- Software engineering is an engineering discipline that 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 that are involved in developing software products. Basic activities are software specification, development, validation and evolution.
- Methods are organized 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.

Continues..

- CASE tools are software systems which are designed to support routine activities in the software process such as editing design diagrams, checking diagram consistency and keeping track of program tests which have been run.
- 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 behavior expected of their members.