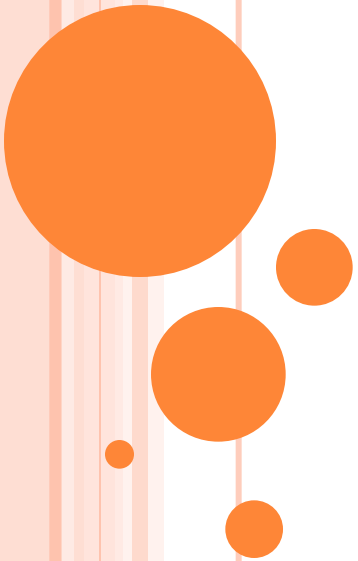
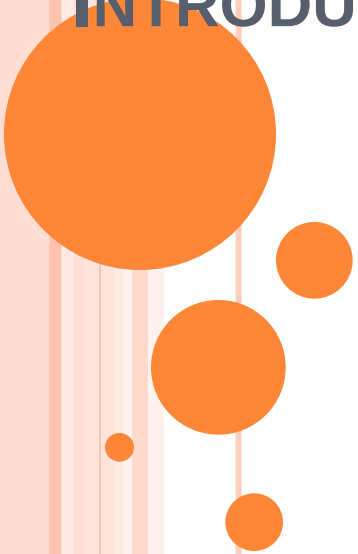
Software Engineering



Lecture-1

Introduction to Software Engineering



What is Software?

“ Software is a set of instructions to acquire inputs and to manipulate them to produce the desired output in terms of functions and performance as determined by the user of the software. It also include a set of documents, such as the software manual , meant for users to understand the software system.”

Description of the Software

A software is described by its capabilities. The capabilities relate to the functions it executes, the features it provides and the facilities it offers. Software written for Sales-order processing would have different functions to process different types of sales order from different market segments . The features for example , would be to handle multi-currency computing, updating product , sales and Tax status. The facilities could be printing of sales orders, email to customers and reports to the store department to dispatch the goods.

Classes of Software

Software is classified into two classes:

- **Generic Software**:

  is designed for broad customer market whose requirements are very common, fairly stable and well understood by the software engineer.

- **Customized Software:**

  is developed for a customer where domain , environment and requirements are being unique to that customer and cannot be satisfied by generic products.

What is Good Software?

Software has number of attributes which decide whether it is a good or bad . The definition of a good software changes with the person who evaluates it. The software is required by the customer , used by the end users of an organization and developed by software engineer . Each one will evaluate the different attributes differently in order to decide whether the software is good.

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

Software - Characteristics

- Software has a dual role. It is a product, but also a vehicle for delivering a product.
- Software is a logical rather than a physical system element.
- Software has characteristics that differ considerably from those of hardware.
- - Software is developed or engineered, it is not manufactured in the classical sense.
- - Software doesn’t “wear out”.
- - Most software is custom-built, rather than being assembled from existing components.

Types of Software

- **System Software-** A collection of programs written to service other programs at system level.

  For example, compiler, operating systems.

- **Real-time Software-** Programs that monitor/analyze/control real world events as they occur.

- **Business Software-** Programs that access, analyze and process business information.

- **Engineering and Scientific Software -** Software using “number crunching” algorithms for different science and applications. System simulation, computer-aided design.

Types of Software

- **Embedded Software-:**

  Embedded software resides in read-only memory and is used to control products and systems for the consumer and industrial markets. It has very limited and esoteric functions and control capability.

- **Artificial Intelligence (AI) Software**:

  Programs make use of AI techniques and methods to solve complex problems. Active areas are expert systems, pattern recognition, games

Types of Software

- **Internet Software :**

  Programs that support internet accesses and applications. For example, search engine, browser, e-commerce software, authoring tools.

- **Software Tools and CASE environment :**

  Tools and programs that help the construction of application software and systems. For example, test tools, version control tools.

Software Engineering

- "A systematic approach to the analysis, design, implementation and maintenance of software."

  (*The Free On-Line Dictionary of Computing)*
- " The systematic application of tools and techniques in the development of computer-based applications."

  (Sue Conger in *The New Software Engineering)*
- " Software Engineering is about designing and developing high-quality software."

  (Shari Lawrence Pfleeger in *Software Engineering -- The Production of Quality Software)*

What is Software Engineering?

Although hundreds of authors have developed personal definitions of software engineering, a definition proposed by Fritz Bauer[NAU69] provides a basis:

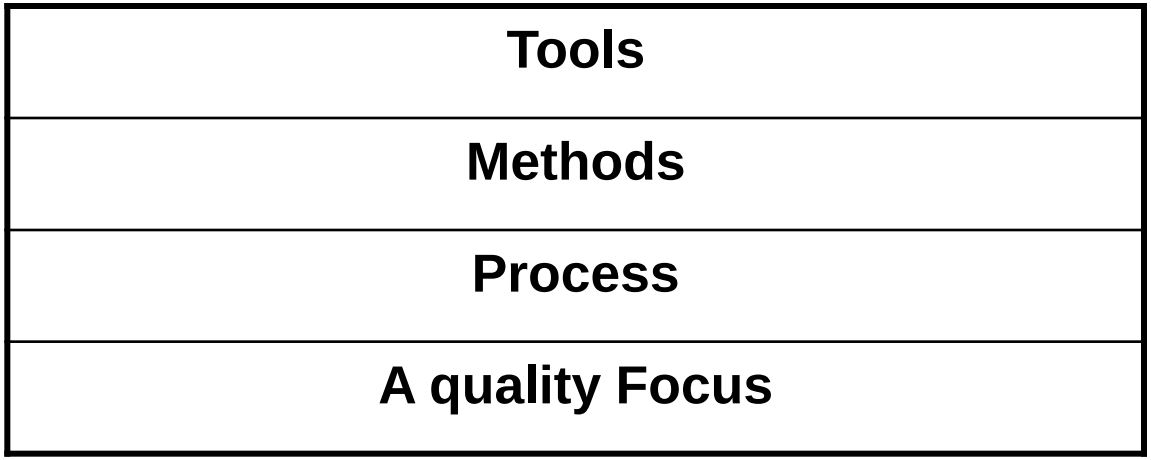
- **"[Software engineering is] the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines."**

The IEEE [IEE93] has developed a more comprehensive definition when it states:

- **"Software Engineering: (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (2) The study of approaches as in (1)."**

What is Software Engineering?

- *Pressman's view:*
  "Software engineering is a layered technology (Figure 2.1)"



What is Software Engineering?

- **Software methods:**
- Software engineering methods provide the technical “how to” for building
- software. Methods --> how to encompass a broad array of tasks:

  - requirements analysis, design, coding, testing, and maintenance
- Software engineering methods rely on a set of basic principles.

What is Software Engineering?

- **Software process:**

  Software engineering process is the glue that holds:

  - technology together
  - enables rational and timely development of computer software.

  Software engineering process is a framework of a set of key process areas.

  It forms a basis for:

  - project management, budget and schedule control
  - applications of technical methods
  - product quality control

What is Software Engineering?

- **Software tools:**

  - programs provide automated or semi-automated support for the process and methods.

  - programs support engineers to perform their tasks in a systematic and/or automatic manner.

Why Software Engineering?

- **Objectives:**

- Identify new problems and solutions in software production.

- Study new systematic methods, principles, approaches for system analysis,

  design, implementation, testing and maintenance.

- Provide new ways to control, manage, and monitor software process.

- Build new software tools and environment to support software engineering.

Why Software Engineering?

- **Major Goals:**

- To increase software **productivity** and **quality**.
- To effectively control software **schedule** and planning.
- To reduce the **cost** of software development.
- To meet the **customers**' needs and requirements.
- To enhance the conduction of software engineering **process**.
- To improve the current **software engineering practice**.
- To support the engineers' activities in a systematic and efficient manner.

Programming versus Software Engineering

- **Programming**

  1. The process of translating a problem from its physical environment into a language that a computer can understand and obey. (*Webster's New World Dictionary of Computer Terms)*

  2. The art of debugging a blank sheet of paper.

  3. A pastime similar to banging one's head against a wall, but with fewer opportunities for rewards. (2 and 3 from *The New Hacker's Dictionary)*

- **Software Engineering** (according to Fritz Bauer)

  "The establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines."

What is the difference between software engineering and computer science?

Computer Science is concerned with **Software Engineering**

- theory
- fundamentals

- the practicalities of developing
- delivering useful software

Computer science theories are currently insufficient to act as a complete underpinning for software engineering, BUT it is a foundation for practical aspects of software engineering

What is the difference between software engineering and system engineering?

- **Software engineering** is part of **System engineering**
- **System engineering** is concerned with all aspects of computer-based systems development including
  - hardware,
  - software and
  - process engineering
- **System engineers** are involved in

  system specification

  architectural design

  integration and **deployment**

Components of Software Engineering

SE approach has two components , namely systems engineering approach and development engineering approach. The software and its quality depends upon the system in which it is installed.

The system here has a broad meanings. The understanding of the system can be achieved by the System study and Analysis.

the System study and Analysis is carried out through SEM(Systems Engineering and Methodology). The SEM steps are as under:

- Define the Objective of the system
- Define the boundaries of the system

Components of Software Engineering

- Factories the system into different components
- Understand the relationship between various components
- Define relationship in terms of inputs, outputs and processes
- Understand the role of hardware and software
- Identify the key operational and functional requirements
- Model the system for analysis and development
- Discuss the system with the customer

Components of Software Engineering

Development Engineering methodology has of translating the system requirements as software system goal , and proceeds to achieve it through a series of steps. The development engineering steps are

- Requirement definition and specification
- Design solution to deliver the requirements
- Determine the architecture for the delivery of solution
- Customer development and planning
- Software testing components
- Integration of system components
- Implementation

Components of Software Engineering

Software development engineering is carried out in two ways

- Structured System Analysis and Design (SSAD)
- Object Oriented System Analysis and Design (OOSAD)

Structured System Analysis and Design (SSAD)

The SSAD approach in which the system and its requirements are decomposed in structured manner. Software development is carried out using sub-system structure, tested and integrated and implemented.

Components of Software Engineering

- **Object Oriented System Analysis and Design (OOSAD)**

  In contrast , the OOSAD development approach recommended the analysis of domain and builds objects of model independent of the system under consideration.

  The object could represents a function , process or document evolved for the organization. Each object has attributes that describes the methods to perform and relationship to other objects.

Comparison between SSAD And OOSAD

- In SSAD the focus is on the functions and the data structure designed for those functions. Functions , data and processing methods are closely coupled. In OOSAD , however , objects and processing methods are decoupled from the data.
- In SSAD , skill lies in decomposing the system whereas in OOSAD skill lies in modeling the organization and its business in the objects.
- SSAD and OOSAD are dissimilar in focus but similar in that both propose a problem solving methodology and a set of techniques and tools to assist the S/W engineer analyze , model ,design and develop the system.