Software Engineering

[CACS253]

BCA 4th Sem

Er. Sital Prasad Mandal

Mechi Multiple Campus

Bhadrapur, Jhapa, Nepal

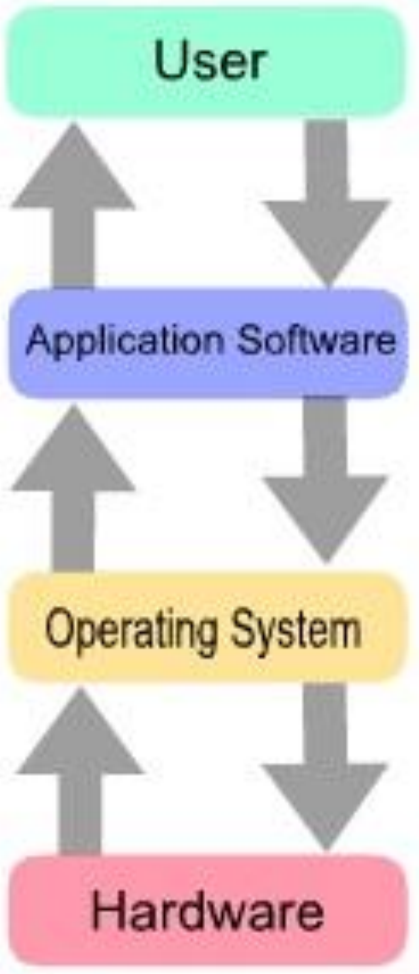
1. Definition of Software
2. Type of Software
3. Characteristic of Software
4. Attributes of Good Software
5. Definition of Software Engineering
6. Software Engineering Costs
7. Key Challenges that Software Engineering Facing
8. System Engineering and Software Engineering
9. Professional Practice

What is a Software?

- **Software** is a computer program with its documentation such as requirements, design models and user manuals.
- Software products may be
  - **Generic** - developed to be sold to a range of different customers e.g. general PC software such as Excel or Word.
  - **Custom** (bespoke, tailored) - developed for a single customer according to their specification.

Type of Software

- The software has mainly divided into two categories:
  1. Application software
  2. System software



Type of Software

Application Software

Applications software also called end-user programs or merely an application. It resides above system software. The end-user uses applications software for a specific purpose. It programmed for simple as well as complex tasks. It either be installed or access online. It can be a single program or a group of small programs that referred to as an application suite.

Application software can be used by the user to complete specific tasks, such as creating word processors documents, spreadsheets, presentations, graphics, CAD/CAM, sending the email, etc.

Type of Software

System Software

System Software (the type of computer program) provides a platform to run a computer's hardware and computer application to utilize system resources and solve their computation problem.

It is written in a low-level language, like assembly language, so it can easily interact with hardware with the primary level. It controls the working of peripheral devices.

System software act as a scheduler for the execution of the processes and arrange the sequence according to their priority and I/O devices requirement and creation of the process.

The best-known example of system software is the operating system (OS). It responsible for manages all the other programs on a computer.

The operating system creates an interface between user and hardware and also in application software and hardware. Examples of software are Windows 7, Windows XP, and Windows 10.

Generally, the user does not interact with the System Software directly. The user interacts with the GUI created by System Software. Through this, GUI user interacts with applications installed in the system.

Type of Software

Other examples of system software and what each does

Utility: Utility Program-Usually utility software installed during the installation of OS. Utility program helps in analyze, configure, secure, optimize and maintain the system. List of utility program-Disk checker, disk cleaner, disk derangement, disk space analyzer, disk backup etc.

Open Source: Open source software is available with source code. Software is an intellectual or collaborative effort of programmers to improve the quality of software. Usually, the software is available with a license for those programmers who want to change the functionality according to their choice. Examples are- Apache, Tomcat, Ubuntu, MySQL (database), and Firefox etc.

Public domain software: Public domain software is free to use the software in the public domain. The software can be modified, distributed or sold without any attribution. It is copyright, unpatented software, where Freeware and Open-Source software have copyright.

Attributes of Good Software

A good software should :

- deliver the required functionality and performance to the user
- be maintainable: can be evolved to meet changing needs
- be dependable: reliable and trustworthy
- be efficient : should not waste the system resources
- be acceptable by end-user, i.e., usable, understandable and compatible with other systems.

Characteristic of Software

Software characteristics are classified into six major components.

- **Functionality**: Refers to the degree of performance of the software against its intended purpose.
- **Reliability**: Refers to the ability of the software to provide desired functionality under the given conditions.
- **Usability**: Refers to the extent to which the software can be used with ease.
- **Efficiency**: Refers to the ability of the software to use system resources in the most effective and efficient manner.
- **Maintainability**: Refers to the ease with which the modifications can be made in a software system to extend its functionality, improve its performance, or correct errors.
- **Portability**: Refers to the ease with which software developers can transfer software from one platform to another, without (or with minimum) changes. In simple terms, it refers to the ability of software to function properly on different hardware and software platforms without making any changes in it.

*In addition to the above mentioned characteristics, **robustness** and **integrity** are also important. **Robustness** refers to the degree to which the software can keep on functioning in spite of being provided with invalid data while **integrity** refers to the degree to which unauthorized access to the software or data can be prevented.*

Software Crisis

The notion of software engineering was first proposed in 1968 at a conference to discuss what was then called 'software crisis':

- Informal (ad-hoc) software development
- Major projects were sometimes years late and over budget,
- Softwares were unreliable, difficult to maintain and performed poorly.

Some Facts

- Failed software projects in USA costs $81 bn annually.
  - http://www.it-cortex.com/Stat_Failure_Rate.htm
- Failure can be severe and expensive.
- Problems may happen during any stage of the Software development life-cycle.
- Purpose of SWE is to avoid problems and hence failure

What is Software Engineering?

SWE is an engineering discipline that 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

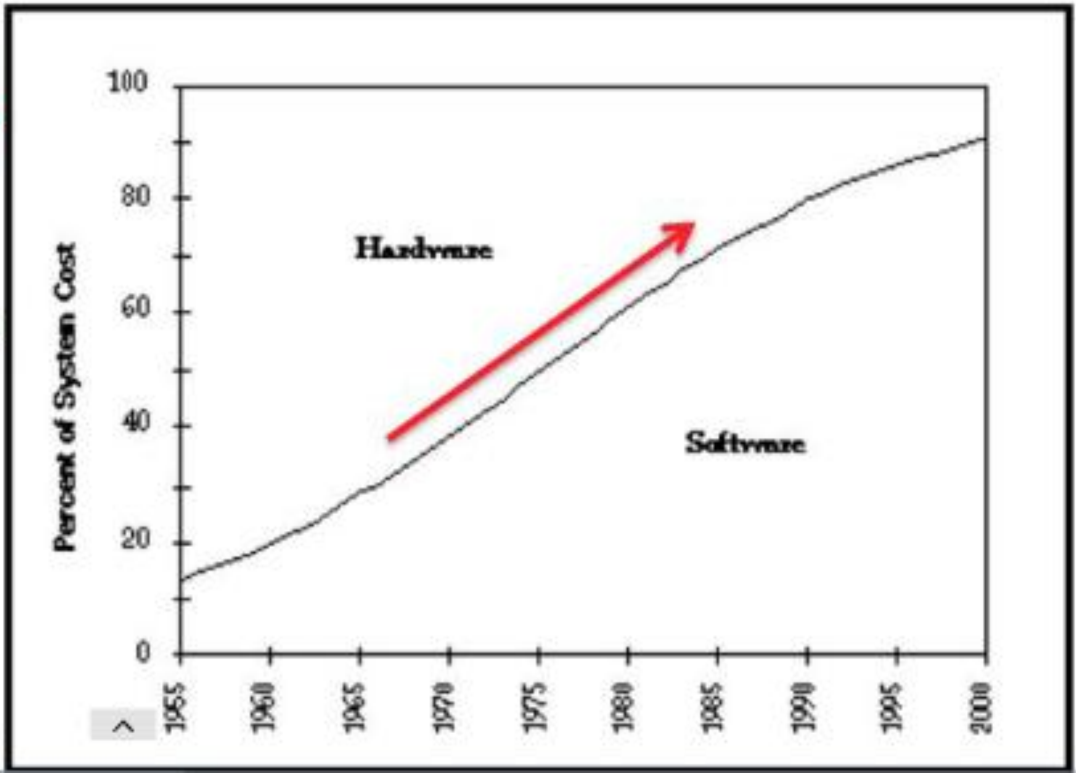
Depending on the problem to be solved (the development constraints + the resources available.)

Software Engineering Cost

- Software costs often dominate computer system costs. The costs of software on a PC are often greater than the hardware cost.

- Software costs more to with a long life, mainter development costs.

- Software engineering is development.



The Big Picture

- Computer Science
- Computer Engineering
- System Engineering
- Software Engineering

- Science is Theory
- Engineering is more practical
- Software $\subseteq$ Computer $\subseteq$ System

Importance of SWE

- The economies of ALL developed nations depends (somehow) on software.
- More and more systems are software controlled
- Software cost often dominate computer system costs. The costs of software on a PC are often greater than the hardware cost.
- SWE develop theories, methods and tools that help to build cost-effective and high-quality software.

SWE vs. Computer Science

- Computer science is concerned with theory and fundamentals
- SWE is concerned with the practicalities of developing and delivering useful software.
- Computer science theories are still insufficient to act as a complete foundation for software engineering.

SWE vs. System Engineering

- System engineering is concerned with all aspects of computer-based systems development including hardware, software and process engineering.
- SWE is part of the process concerned with developing the software infrastructure, control, applications and databases in the system.
- System engineers are involved in system specification, architectural design, integration and deployment.

What is a Software Process (SP)?

- SP is a set of activities whose goal is the development or evolution of software.
- General activities in all SPs are:
  - Specification: what should the system do and what are its development constraints?
  - Development: production of the software
  - Validation: checking that the software is what the customer wants
  - Evolution: changing the software in response to changing demands.

What is a Software Process Model (SPM)?

- SPM is a simplified representation of a software process, presented from a specific perspective such as
  - Workflow perspective: sequence of activities;
  - Data-flow perspective: information flow;
  - Role/action perspective: who does what.
- Examples of generic SPM
  - Waterfall
  - Iterative development
  - Component-based software engineering

What is CASE?

- CASE (Computer-Aided Software Engineering) are software systems that are intended to provide automated support for software process activities.
- CASE are often used for method support.
  - Upper-CASE: support the early process activities of requirements and design;
  - Lower-CASE: support later activities such as programming, debugging and testing.

Key Challenges Facing Software Engineering

- Heterogeneity: developing techniques for building software that can cope with heterogeneous (different) platforms and execution environments
- Delivery: developing techniques that lead to faster delivery of software
- Trust: developing techniques that demonstrate that software can be trusted by its users.

Professional Practice

The Software Engineering Professional Practice knowledge area (KA) is concerned **with the knowledge, skills, and attitudes** that software engineers must possess to practice software engineering in a professional, responsible, and ethical manner.

Summary

- SWE 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 acceptability.
- SP consists of activities that are involved in developing softwares. Basic activities are software specification, development, validation and evolution.
- CASE tools are software systems which are designed to support routine activities in the SP

FAQs about software engineering

- What is software?
- What is software engineering?
- What is the difference between software engineering and computer science?
- What is the difference between software engineering and system engineering?
- What is a software process?
- What is a software process model?
- What are the costs of software engineering?
- What are software engineering methods?
- What is CASE (Computer-Aided Software Engineering)
- What are the attributes of good software?
- What are the key challenges facing software engineering?

Frequently asked questions about software engineering

Question	Answer
What is software?	Computer programs and associated documentation. Software products may be developed for a particular customer or may be developed for a general market.
What are the attributes of good software?	Good software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable.
What is software engineering?	Software engineering is an engineering discipline that is concerned with all aspects of software production.
What are the fundamental software engineering activities?	Software specification, software development, software validation and software evolution.
What is the difference between software engineering and computer science?	Computer science focuses on theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software.
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 more general process.

Frequently asked questions about software engineering

Question	Answer
What are the key challenges facing software engineering?	Coping with increasing diversity, demands for reduced delivery times and developing trustworthy software.
What are the costs of software engineering?	Roughly 60% of software costs are development costs, 40% are testing costs. For custom software, evolution costs often exceed development costs.
What are the best software engineering techniques and methods?	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. You can't, therefore, say that one method is better than another.
What differences has the web made to software engineering?	The web has led to the availability of software services and the possibility of developing highly distributed service-based systems. Web-based systems development has led to important advances in programming languages and software reuse.

Reference

1. https://padakuu.com/article/790-characteristics-of-software-in-software-engineering

2. http://www.tutorialsspace.com/Software-Engineering/11-Characteristics-Of-Software.aspx

3. https://www.geeksforgeeks.org/software-engineering-software-characteristics/