



Text Book is Important

✳ What is software?

⇒ Computer programs and associated documentation. Software products maybe developed for a particular customer or may be developed for a general market.

✳ What are the attributes of good software?

⇒ Good software should be 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
- development
- validation
- 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.

⊕ What are the key challenges facing software engineering?

- ⇒ Coping with increasing diversity, demands for ~~reducing~~ reduced delivery time, and developing trustworthy software.

⊕ What are the cost of software engineering?

- ⇒
- development cost \Rightarrow 60%.
 - testing cost \Rightarrow 40%.
 - for custom software, evolution cost often exceed development cost.

⊕ What are the best software engineering techniques and methods?

- ⇒ While all software project have to be professionally managed and developed, different techniques are appropriate for different types of system.
- 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, 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 system. Web-based system development has led to important advances in programming languages and software reuse.

(*) Types of Software Product:

i. Generic Products:

- stand alone system, produced by organization, sold on the open market to any customer
- Excel, Word etc.

ii. Customized Products:

- these are systems that are commissioned by a particular customer.
- system for electronic device, air traffic control system, etc.

iii. Enterprise Resource Planning (ERP) System:

- generic product but adapted in future to suit the requirement of a customer.

★ Key Phases of Software Engineering:

i. Engineering Discipline:

- apply theories, methods, and tools where these are appropriate.
- use them selectively
- always try to discover solution to problem.

ii. All aspects of software production:

- also includes activities such as software project management and the development of tools, methods, and theories to support software production.

★ Reasons of Software Engineering:

- everyone rely on advanced software system.
- we need to be able to produce reliable and trustworthy systems economically and quickly.
- it is usually cheaper, in the long run, to use software engineering methods and techniques for software system.
- the majority of costs are the costs of changing the software after it has gone into use.

★ Maintainability:

- Software should be written in such a way so that it can evolve to meet the changing needs of customers. This is a critical attribute because software change is an inevitable requirement of a changing business environment.



Dependability and security:

- Software dependability includes a range of characteristics including reliability, security, and safety. Dependable software should not cause physical or economic damage in the event of system failure. Malicious users should not be able to access or damage the system.



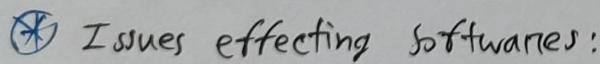
Efficiency:

- Software should not make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes responsiveness, processing time, memory utilization etc.



Acceptability:

- Software must be acceptable to the types of users for which it is designed. This means that it must be understandable, useable, and compatible with other systems that they use.



Issues effecting softwares:

i. Heterogeneity:

- systems are required to operate as distributed systems across network that include different types of computer and mobile devices.
- need to compatible on all devices and also in old system.

ii. Business and Social change:

- Business and society are changing incredibly quickly as emerging economies develop and new technologies become available.
- They need to be able to change their existing software and rapidly develop new software.
- need to reduce the time to deliver new system to customers.

iii. Security & trust:

- It is essential to trust a software before we.
- especially true for remote software system accessed through a web page or web service interface.
- we need to make sure that, malicious users cannot attack our system and that information security is maintained.

④ Applications of software engineering:

i. Stand-alone applications

- run on a local computer, such as PC
- includes all necessary function, do not need to be connected to a network
- Office, CAD, Photo Editor etc.

ii. Interactive transaction-based applications:

- these are applications that execute on a remote computer and that are accessed by users from their own PC.
- e-commerce applications.

iii. Embedded control systems:

- These are software control systems that control and manage hardware devices
- Software in a mobile phone, software in a microwave oven.

iv. Batch processing systems:

- These are business systems that are designed to process data in large batches.
- They process large numbers of individual inputs to create corresponding outputs.
- periodic billing system, salary payment system.

v. Entertainment System:

- These are systems that are primarily for personal use and which are intended to entertain the user.
- various kind of games.

vi. System for modeling & simulation:

- These are systems that are developed by scientists and engineers to model physical processes or situations, which include many, separate, interacting objects.
- require high-performance parallel system.

VII. Data Collection Systems:

- These are systems that collect data from their environment using a set of sensor and send that data to other systems for processing.

VIII. Systems of Systems:

- These are systems that are composed of a number of other software systems.
- spreadsheet program (generic software product)

Fundamentals of Software Engineering:

- i.
 - should be developed using a managed and understood development process.
 - should plan the development process and have clear ideas of what will be produced and when it will be completed.
- ii.
 - Dependability and performance are important for all types of systems.
 - Software should behave as expected, without failures
 - should be available for use, when required
 - should be secure against external attack,
 - system should perform efficiently and should not waste resources.

- iii. - Understanding and managing the software specification, and requirements are important.
 - You have to manage customers expectations so that a useful system can be delivered within budget and to schedule.
- iv. - You should make as effective use as possible of existing resources.
 - should reuse software that has already been developed rather than write new software.

Software Engineering Ethics:

i. Confidentiality:

- You should normally respect the confidentiality of your employers or clients irrespective of whether or not a formal confidentiality agreement has been signed.

ii. Competence:

- You should not misrepresent your level of competence. You should not knowingly accept work that is outside your competence.

iii. Intellectual Property Rights:

- You should be aware of local laws governing the use of intellectual property such as patents and copyright.
- You should be careful to ensure that the intellectual property of employers and clients is protected.

IV. Computer misuse:

- You should not use your technical skills to misuse other peoples computers.
- Game playing on an employer's machine
 - dissemination of viruses or other malware.

L-2 / 13. 02.2024 /

From Lecture Slide-2

⊕ Activities of software engineering:

(i) Software specification

- The functionality of the software and constraints on its operation must be defined.

(ii) Software design and implementation:

- The software to meet the specification must be produced.

(iii) Software validation:

- The software must be validated to ensure that it does what the customer wants.

(iv) Software evolution:

- The software must evolve to meet changing customer needs.

* Product Description Includes:

⇒ Products : outcomes of a process activity

⇒ Roles : which reflect the responsibilities of the people involved in the process.

⇒ Pre- and Post-condition : statements that are true before and after a process activity has been enacted or a product produced.

Pre ⇒ agreement of requirements

Post ⇒ UML which describe the entire products architecture.

* Software Process Model:

i. The waterfall model

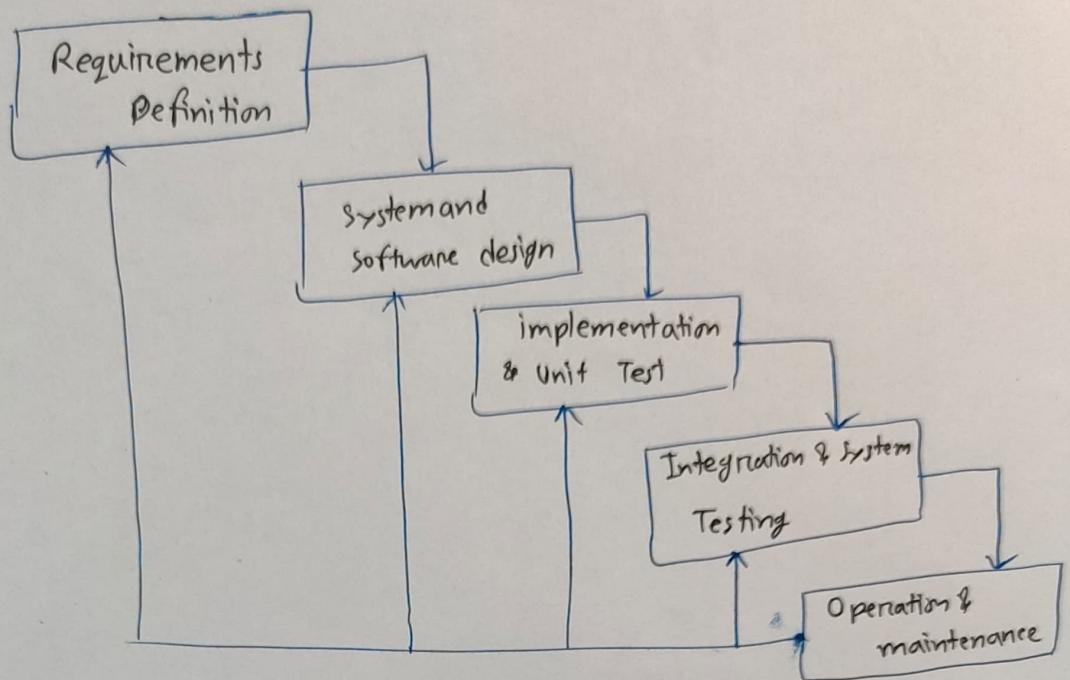
ii. Incremental development

iii. Reuse-oriented software engineering

* The waterfall model

- This takes the fundamental process activities of specification, development, validation, and evolution and represent them as separate process phases.

⇒ There are five stages:



i Requirements analysis & Definition:

- The system's services, constraints and goals are established by consultation with system users. They are then defined in details and serve as a system specification.

ii System and software design:

- The system design process allocates the requirements to either hardware or software by establishing an overall system architecture. Software design involves identifying and describing the fundamental software system abstractions and their relationships.

(iii) Implementation and unit testing:

- During this stage, the software design is realized as a set of programs or program units. Unit testing involves verifying that each unit meets its specification.

(iv) Integration and system testing

- The individual program units or programs are integrated and tested as a complete system to ensure that the software requirements have been met. After testing, the software system is delivered to the customer.

(v) Operation and maintenance:

Discontinued