Note Junction

Best Note Provider

Note By: Roshan BiSt

UNIT-1

Introduction

Software is set of computer programs and associated documentation. Software engineering is strategy for producing quality software. It is the establishment and use of sound engineering principles in order to obtain economically reliable software and works efficiently on real machines. It is an engineering discipline which is concerned with all aspects of software production.

⊛. Software Types:

i) **Generic:** These are the systems used for developing a general purpose software. From designing and marketing perspective, this kind of development is very difficult. Large number of users may be using this kind of software. Development team controls the process of generic software development. Word-editing software is it's example.

also called bespoke

ii) **Custom:** These are the systems used for developing a software product as per the needs of particular customer. This development does not require marketing, because it is developed for appropriate group of users. Customer determines the process of software development in this type of product. Control system for electronic device is it's example.

OR characteristics of software

⊛. Attributes of Good Software:

The essential attributes of good software are as follows:-

i) **Maintainability:** Software must evolve to meet changing need of customer. It is called maintainability. So, software should be written in such a way that, it may evolve changing needs of customer.

ii) **Dependability:** Dependability of software is a property of software that reflects it's trustworthiness. It is the degree of confidence a user has that the system will operate as they expect and the system will not fail in normal use.

iii) **Portability:** It refers to the ease with which software developers can transfer software from one platform to another.

iv) Efficiency: Software should not make wasteful use of system resources such as memory, processor cycle. So, efficiency means, responsiveness, processing time, memory utilization.

v) Usability: Software must be easly usable, it should not be too complicated to use. It should have adequate documentation and appropriate interface.

⊛. Advantages of software engineering:

→ Improved quality.
→ Improved reliability
→ Improved productivity
→ Improved requirement specification
→ Improved cost and schedule estimates.
→ Well defined process.

⊛. Importance of software engineering (Why software engineering?):

→ The economics 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.
→ Software engineering expendicture represents a significant fraction of GNP in all developed countries.

⊛. Fundamental Software Engineering Activities:

There are many different kinds of software processes, but each and every one involve following four fundamental activities:

i) Software Specification:- Software specification is the process of understanding and defining what services are required from the system and identifying the constraints on the system operation and development. It is a critical stage, because any error in this stage will lead to later problems in software design and implementation.

ii) Development: It includes software design and implementation. It is the process of converting the system specifications into an executable system.

iii) Validation: It is concerned with building the right system. It is intended to show that a system confirms to its specifications and meets the user expectations.

iv) Evolution: Evolution is the time to time maintenance of the system to meet changing needs of customer with time. So, software should be written in such a way that it may evolve changing needs of customer.

⊛. Differences between software engineering and computer science:

Software Engineering	Computer Science
i) Software Engineering is the study of how software systems are built.	i) Computer science is the study of how computers perform theoretical and mathematical tasks.
ii) It involves the study and application of software only.	ii) It involves the study and application of software and hardware both.
iii) It is the structural process of checking, verifying, finding the errors and bugs according to the need of software and then provide a solution for removing that bug.	iii) It is not a structural process as everything is to be done in a process and requires proper study before executing
iv) It involves some areas of study which are software development, software testing and Quality assurance.	iv) It involves areas of study which are networking, artificial intelligence, database systems etc.
v) Software Engineering majorly defines architecture and structural properties.	v) Computer science involves the study of both principles and the use of computers.

⊛. Differentiate between software engineering and system engineering:

Software engineering	System engineering
i) Software engineering is an engineering discipline that is concerned with all aspects of software production.	i) System engineering is a field of engineering and engineering management that focus on how to design and manage complex system over their life cycle.
ii) Software engineering highly focuses on implementing quality software.	ii) System engineering highly focuses on the user's and domains.
iii) Software engineering includes in computer science or computer based engineering background.	iii) System engineering may ~~cover~~ cover a broader area, entire system development
iv) Software engineering focus solely on software components.	iv) System engineering focus on hardware engineering
v) Software engineering is newly developed discipline.	v) System engineering is an older discipline

⊛. Challenges of software engineering:

i) Heterogenity challenge: Heterogenity means diversity or variety. There are different types of computer and with different kinds of support systems. The heterogenity challenge is the challenge of developing techniques to build software which is flexible to support by most of the systems.

ii) The legacy challenge: The legacy challenge is the challenge of maintaining and updating this software in such a way that excessive costs are avoided and essential business services continue to be delivered.

iii) The delivery challenge: Software engineering techniques are time-consuming, to achieve better software quality. Most of businesses nowdays want software systems quickly. This shortening delivery time of system for large and complex systems without compromising system quality is called delivery challenge.

iv) Trust challenge: A software is twisted with all aspects of our lives, it is essential that we can trust that software, so the trust challenge is to develop techniques that demonstrate that software can be trusted by its users.

v) Risk challenge: In safety-critical areas such as space, aviation, nuclear power plants, etc. the cost of software failure can be massive because lives are at risk. Dealing with the increased complexity of software need for new applications.

⊛. Cost of software engineering:

The distribution of costs across the different activities in the software process depends on the process used and the type of software that is being developed. For example, real-time software usually requires more extensive validation and testing than web-based systems. So, roughly 60 percent of costs are development costs and 40 percent are testing costs. For customer software evolution costs often exceeds development costs. Distribution of costs depends on the development model that is used.

⊛. Professional software development:

Lots of people write programs. People in business write spreadsheet programs to simplify their jobs, scientists and engineers write programs to process their experimental data, some people write programs for their own interest and enjoyment. However, most software development is a professional activity where software is developed for business purposes. This developed software is maintained and changed throughout its life.

Software engineering is intended to support professional software development, rather than individual programming. A professionally developed software system is often more than a single program, a system may consist of several seperate programs and configuration files that are used to set up these programs. It may include system documentation, which describes structure of the system, user documentation, which explains how to use the system and web sites for users to download recent product information.

⊛. Software engineering diversity:

There are no universal software engineering methods that are suitable for all systems and all companies. Rather, a diverse set of software engineering methods and tools has evolved over the past 50 years. The most significant factor in determining which software engineering methods and techniques are most important is the type of application that is being developed.

We should make as effective use as possible for existing resources. This means that, where appropriate, we should reuse software that has already been developed rather than write new software.

⊛. Internet software engineering:

Rather than local system, the internet is now a platform for running applications. Internet service allows application functionality to be accessed over the internet. With the help of internet, instead of writing software and deploying it on users PC, the software can be developed on web server that can be accessed through browsers. This made it much cheaper to change and upgrade the software as there was no need to install the software on every PC.

⊛. Software engineering ethics:

Following are some software engineering ethics:

i) Confidentiability: We should normally respect the confidentiality of our employees or clients irrespective of whether a formal confidentiability agreement has been signed.

ii) Competence: We should never misrepresent our skills and the level of competency. We should never accept any work which is out of our competency.

iii) Intellectual property rights: We should be aware of local laws governing the use of intellectual property such as patents and copyright.

iv) Computer misuse: We should not use our technical skills to misuse other people's computers. Computer misuse ranges from simple (like game playing on an employer's machine) to extremely serious (dissemination of viruses).