

Subject: Software Engineering,

Chapter-1, Introduction

Software:

- Software is the collection of data, programs, procedures, routines and instructions that tell a computer or electronic device how to run, work and execute specific tasks.
- Software plays a role of interface between the user and the computer hardware which enables the communication between these two .
- A software product development company is the one which develops software for the users.

Types of Software:

Commonly software are classified into two main categories, which are namely as:

1. System Software:

- System software is a group of computer programs that direct the internal operations of computer system such as controlling I/O devices, managing the storage area within the computer, etc.
- In Simple word, it can be said that the system software is essentially an intermediary or even a middle layer between the user as well as hardware.
- System Software is of following types:
 - ✓ Operating System: An Operating system is an integrated set of programs that is used to manage the various resources and overall operation of the computer system. It provides the user within an interface that is easy to use than bare hardware. Some of the key example of operating system are as follows: MS Windows, macOS, Linux, Android, Ubuntu, Unix,etc.
 - ✓ Translator (language processor): As we have learned, there are multiple types of programming languages used to write programs. But computer only understands the program written in machine language. Hence all program written in other language must be translated into machine language program before executing them. Translator is a system software that converts the program written in other language into an equivalent machine language

program before executing them. Some of the translating programs are: compiler, interpreter and assembler.

- ✓ **Device Drivers::** Every hardware has to be controlled through software, these are the software which make any hardware work. Different hardware devices which require a driver to connect to a system easily consist of displays, printers, sound cards, hard disks, keyboard, and mice. Few of the examples of such drivers are: BIOS Driver, Motherboard Drivers, ROM Drivers, Printer Drivers, USB Drivers, Sound Card Driver, VGA Drivers etc.
- ✓ **Firmware::** The permanent software that is provided by the manufacture for controlling the system which is embedded in system's ROM. It is essentially a set of instructions which is permanently stored onto the hardware device. It offers vital information regarding how a particular device interacts with other hardware. For Example: Computer Peripherals Embedded Systems UEFI BIOS.
- ✓ **Utility Software:** Utility programs are also called system support software or service program or helper software, which provide useful services to the user. These are the programs that enhance or extend the operating system's capabilities or which simply offer new features not provided by OS itself.

2. **Application Software:** The software that is written to solve a specific user oriented problem using the computer is known as application software. The application software may be used for official purpose of the government engineering design, tech aid and so on. They are also popularly known as end user programs or even playing computer games. Two types of Application software:-

- ✓ Customized or Tailored Software
- ✓ Packaged Software

S.NO	Packaged Software	Customized Software
1.	Packaged software is a compilation of programs which are grouped together in order to provide publicly with different tools in the same group.	Custom software is a specific program that is advanced for a goal in a department or in a company.
2.	Cost of Packaged software is low.	While custom software's cost is higher than packaged software since custom software are made for a specific purpose.
3.	Packaged software is easy or simple to use because these software have an easy interface.	Whereas custom software is hard to use because these software are custom-built or customized.
4.	Its measures of success are market sharing and profitability.	Whereas it's measures of success is to provide satisfaction and acceptance to user.
5.	It's bench-marking is poor due to its size ability.	While it's bench-marking is good due to its benchmark data available.
6.	It is present in the market to everybody for buying and using.	While it is not commercially available, it is exclusive to third parties.
7.	Packaged software can't be modified.	But custom software can be modified or changed if there is need.

3. Web based and Mobile Application Software:

- A web-based application (also called webapp) is application software that is accessed over a network connection using Hyper Text Transfer protocol (HTTP), rather than existing within a device's memory.
- That is web app runs on a web server unlike computer-

based software programs that stored locally on the Operating system (OS) of the device .

- Web applications are accessed by the, user through a web browser with an active internet connection. However, web-based applications also may be client-based, where a small part of the program is downloaded to a user's desktop, but processing is done over the internet on an external server.
- For Example: Flash Games, Online Calculators ,and generating colors.

Advantages:

- Do not need to install software
- Storage flexibility
- Platform Independent
- Do not have to worry for backup

4. Mobile Application

- A mobile application, most commonly referred to as a mobile app or simply app, is a type of application software designed to run on a mobile device, such as a smartphone or tablet or watch
- Mobile applications frequently serve to provide users with similar services to those accessed on PCs.
- Apps are generally distributed from the application distribution platforms called app store, so users have to download apps from the app store.

Categories:

Game Apps: Pokémon Go, Pub

Business Apps: Facebook Pages Manager, Adobe Acrobat Reader

Educational App: Harmon CSIT, CSIT tutor

Entertainment App: Netflix, Sports Tak

Utility : Reminder ,Calculator

Advantages of Mobile Application

- Speed
- Ability to work offline
- Security
- User Interface
- Mobile Apps
- Functionality

Differences between system Software and application software:

System Software	Application Software
<p>The Software that is designed to control, integrate and manage the individual hardware components and application software is known as system software</p> <p>Low-level languages are used to write the system software.</p>	<p>The software that is written to solve a specific user oriented problem using the computer.</p> <p>While high-level languages are used to write the application software.</p>
<p>It is general-purpose software.</p>	<p>While it's a specific purpose software.</p>
<p>Without system software, the system stops and can't run.</p> <p>System software runs when the system is turned on and stops when the system is turned off.</p>	<p>While Without application software system always runs.</p> <p>While application software runs as per the user's request.</p>
<p>Example: System software is an operating system, etc.</p>	<p>Example: Application software is Photoshop, VLC player, etc.</p>
<p>System Software programming is more complex than application software.</p>	<p>Application software programming is simpler in comparison to system software.</p>

System Software	Application Software
System software runs independently.	Application software is dependent on system software because they need a set platform for its functioning.

#Software Characteristics:

- **Software does not wear out**
Different things like clothes ,shoes, ornaments etc do wear out after some time. But, software once created never wears out. It can be used for a long as it is needed and in case of needed for any updating ,required changes can be made in the same software.
- **Software is not manufactured but it is developed. So it doesnot required any raw material for its development.**
- **Usability of Software**
- **Flexibility of Software**
- **Maintainability of Software**
- **Portability of Software**
- **Reliability of Software**

Attributes of Good Software:

- **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 to meet the changing needs of customer.
- **Dependability:** It is a property of software that reflects its 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

user. Dependability has a range of characteristics including availability, reliability, security and safety.

- I. Availability: The ability of the system to deliver services when requested by the user is called availability.
 - II. Reliability: The ability of the software to deliver services as specified is called reliability. It is a probability over a given period of time, that the software will correctly deliver services as expected by the user.
 - III. Safety: The ability of software to operate correctly without failure is called safety.
 - IV. Security: The ability of software to protect itself against accidental or unauthorized users is called security.
- **Efficiency:** Software should not make wasteful use of system resources such as memory, processor cycle. So, efficiency means, responsiveness, processing time, memory utilization.
 - **Usability:** Software must be usable without undue effort, by the type of user for whom it is designed. It means, it should have adequate documentation and appropriate interface.

Software Engineering and its importance:

- The term software Engineering is composed of two words, software and engineering.
- Software is program code or set of executable codes containing instructions that give desired functionality.
- Engineering is the process of designing and constructing something which assists a particular purpose and seeks a cost-effective solution to issues.
- So, Software engineering is an engineering discipline which is concerned with all aspects of software production from the early stage of requirement specification to maintaining the software after it has gone to use.

- Software Engineering was introduced to address timelines, budgets and reduced levels of quality . It ensures that the application is built consistently , correctly , on time and on budget and within requirements.

Need or Importance of Software Engineering:

1. **Large software:** It is easier to build a wall than to a house or building, likewise, as the size of software become large engineering has to step to give it a scientific process.
 2. **Scalability:** If the software process were not based on scientific and engineering concepts, it would be easier to re-create new software than to scale an existing one.
 3. **Cost:** As hardware industry has shown its skills and huge manufacturing has lower down the price of computer and electronic hardware. But the cost of software remains high if proper process is not adapted.
 4. **Dynamic Nature:** The always growing and adapting nature of software hugely depends upon the environment in which user works. If the nature of software is always changing, new enhancements need to be done in the existing one. This is where software engineering plays a good role.
 5. **Quality Management:** Better process of software development provides better and quality software product.
-

Fundamental Software Engineering Activities:

Q. 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 conforms 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
<p>i) Software Engineering is the study of how software systems are built.</p> <p>ii) It involves the study and application of software only.</p> <p>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.</p>	<p>i) Computer science is the study of how computers perform, theoretical and mathematical task.</p> <p>ii) It involves the study and application of software and hardware both.</p> <p>iii) It is not a structural process as everything is to be done in a process and requires proper study before executing.</p>
<p>iv) It involves some areas of study which are software development, software testing and Quality assurance.</p> <p>v) Software Engineering majorly defines architecture and structural properties.</p>	<p>iv) It involves areas of study which are networking, artificial intelligence, database systems etc.</p> <p>v) Computer science involves the study of both principles and the use of computers.</p>

Differences 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 users 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 and Cost of Software Engineering:

- i) Heterogeneity challenge: Heterogeneity means diversity or variety. There are different types of computer and with different kinds of support systems. The heterogeneity 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 nowadays want software systems quickly. This shortening delivery time of system for large and complex systems without compromising system quality is called delivery challenge.

C SIT Tutor

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:

⊗ 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 separate 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:

Software engineering is a systematic approach to development of software that takes into account practical cost, schedule and dependability issues. There are no universal software engineering methods and techniques that are suitable for all systems and all companies; rather diverse set of methods and tools has evolved. There are many different types of application diversity.

1. **Standalone Applications:** These are applications that run on a local computer such as PC; do not need to be connected to network. E.g. office package, CAD, Photo manipulation etc.
2. **Interactive transaction based application:** These are applications that execute on a remote computer and that are accessed by users from their own PCs or terminals. They include business applications such as e-commerce applications.
3. **Embedded control system:** These are software control systems that control and manage hardware devices. Example: software in mobile phones, software in microwave oven system to control cooking process.
4. **Batch processing system:** these are business systems that are designed to process data in large batches. Example: phone billing systems, salary payment systems.
Entertainment system: these are systems primarily for personal use and entertain the user. Example: Gaming software.

Internet Software Engineering:

⊗. 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:

- Software Engineering Ethics refers to the set of principles and values that guide the behavior and decision-making of professionals in the field of software engineering.
- These ethical guidelines are essential to ensure that software engineers and developers conduct themselves in a responsible and morally upright manner throughout the software development lifecycle.

⑩. Software engineering ethics: Following are some software engineering ethics:

- i) Confidentiality: We should normally respect the confidentiality of our employees or clients irrespective of whether a formal confidentiality 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).