

Software

Program: A step by step instructions to perform a specific task on a computer in a programming language is called program, i.e., set of instructions.

Software: Is a program along with proper documentation (requirement analysis, design, coding, testing) and user manuals which mainly includes installation guide and other manuals.

Software = Program + documentation

Software, which is abbreviated as SW or S/W, is a set of programs that enables the hardware to perform a specific task. All the programs that run the computer are software. The software can be of two types: system software and application software.

Software Components

A program is a subset of software and it becomes software only if documentation and an operating procedure manual are prepared.

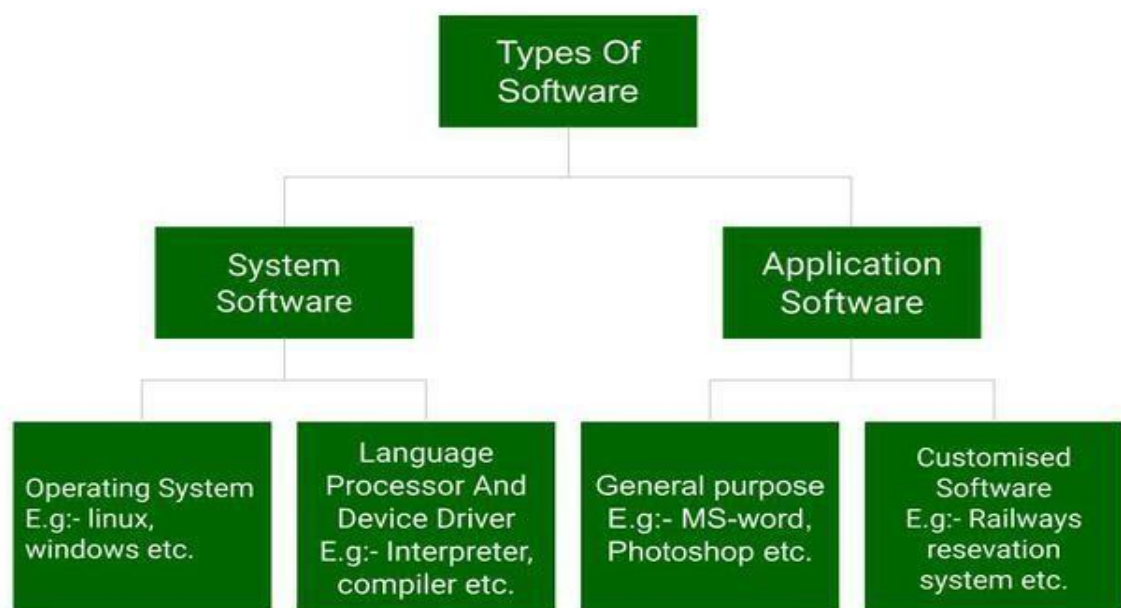
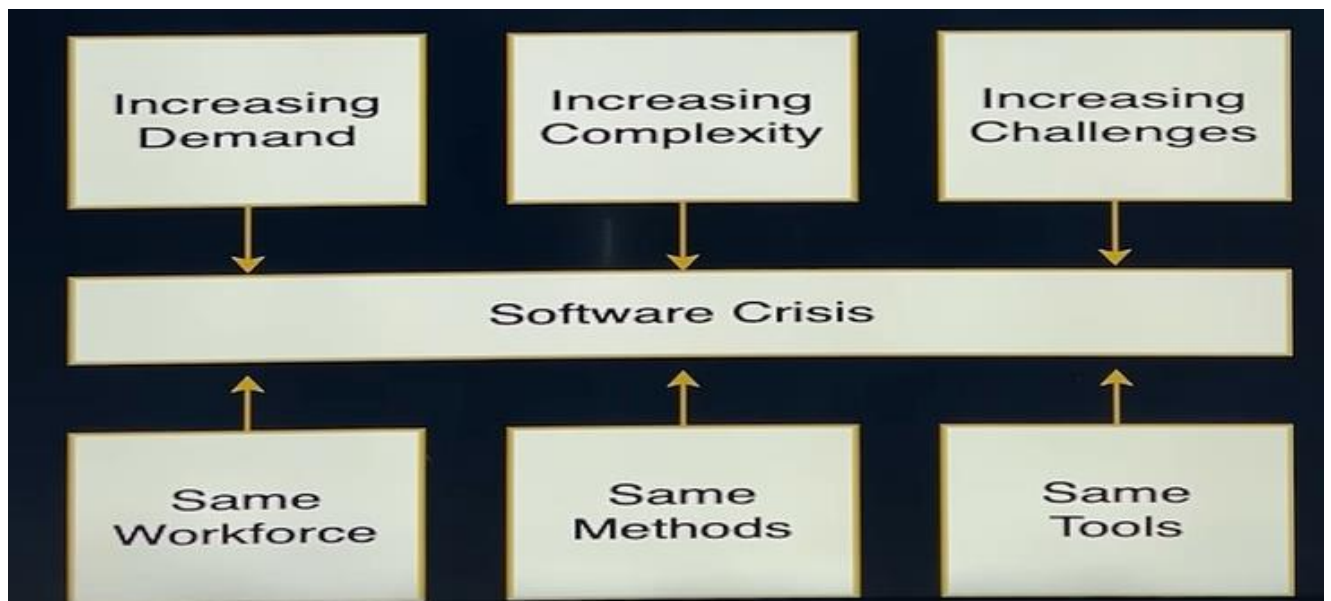
- **Program**
- **Documents**
 - Software documentation consist all the description, programs, graphics and instructions pertaining to design, coding, testing and preparation of software.
- **Operating Procedure (User Manual & Operational Manual)**

Provides information about what software is how to work with it how to install it on your system and how to control all the activities of the software.

Software Crisis

Problems with software?

- The major problem with current scenario in Software industry is, software usually overrun their development cost, they exceed their development duration limits, are usually of poor quality.
- As long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a mild problem, and now we have gigantic computers, programming has become an equally gigantic problem. — Edsger Dijkstra, The Humble Programmer (EWD340), Communications of the ACM



1) System Software

The system software is the main software that runs the computer. When you turn on the computer, it activates the hardware and controls and coordinates their

functioning. The application programs are also controlled by system software. An operating system is an example of system software.

Features of system software:

Let us discuss some of the features of System Software:

1. System Software is closer to the computer system.
2. System Software is written in a low-level language in general.
3. System software is difficult to design and understand.
4. System software is fast in speed (working speed).
5. System software is less interactive for the users in comparison to application software.

Types of system software:

It has three subtypes which are:

1. **Operating System:** It is the main program of a computer system. When the computer system ON it is the first software that loads into the computer's memory. Basically, it manages all the resources such as memory, CPU, printer, hard disk, etc., and provides an interface to the user, which helps the user to interact with the computer system. It also provides various services to other computer software. Examples of operating systems are Linux, Apple macOS, Microsoft Windows, etc.
2. **Language Processor:** As we know that system software converts the human-readable language into a machine language and vice versa. So, the conversion is done by the language processor. It converts programs written in high-level programming languages like Java, C, C++, Python, etc. (known as source code), into sets of instructions that are easily readable by machines (known as object code or machine code).
3. **Device Driver:** A device driver is a program or software that controls a device and helps that device to perform its functions. Every device like a printer, mouse, modem, etc. needs a driver to connect with the computer system eternally. So, when you connect a new device with your computer system, first you need to install the driver of that device so that your operating system knows how to control or manage that device.

2) Application Software:

Software that performs special functions or provides functions that are much more than the basic operation of the computer is known as application software. Or in other words, application software is designed to perform a specific task for end-users. It is a product or a program that is designed only to fulfill end-users'

requirements. It includes word processors, spreadsheets, database management, inventory, payroll programs, etc.

Features of application software:

Let us discuss some of the features of Application Software:

1. An important feature of application software is it performs more specialized tasks like word processing, spreadsheets, email, etc.
2. Mostly, the size of the software is big, so it requires more storage space.
3. Application software is more interactive for the users, so it is easy to use and design.
4. The application software is easy to design and understand.
5. Application software is written in a high-level language in general.

Types of application software:

There are different types of application software and those are:

1. **General Purpose Software:** This type of application software is used for a variety of tasks and it is not limited to performing a specific task only. For example, MS-Word, MS-Excel, PowerPoint, etc.
2. **Customized Software:** This type of application software is used or designed to perform specific tasks or functions or designed for specific organizations. For example, railway reservation system, airline reservation system, invoice management system, etc.
3. **Utility Software:** This type of application software is used to support the computer infrastructure. It is designed to analyze, configure, optimize and maintains the system, and take care of its requirements as well. For example, antivirus, disk fragmenter, memory tester, disk repair, disk cleaners, registry cleaners, disk space analyzer, etc.

Difference between system software and application software

System Software	Application Software
It is designed to manage the resources of the computer system, like memory and process management, etc.	It is designed to fulfill the requirements of the user for performing specific tasks.
Written in a low-level language	Written in a high-level language
Less interactive for the users	More interactive for the users

System software plays vital role for the effective functioning of a system.	Application software is not so important for the functioning of the system, as it is task specific.
It is independent of the application software to run.	It needs system software to run.

Attributes of Good Software



1. Acceptance

The essential attribute of good software is the acceptance of software worldwide, especially to the users for whom they are designed and developed.

You may have come across expensive software that is not easy to use as its features and characteristics are difficult to understand and later implement.

Any software's primary goal or objective must be to provide the results that the user requires; therefore, software acceptance is needed. Firstly, the user interface of any program must be user-friendly; secondly, the operations and tasks must perform at a reasonable speed.

Most importantly, they must provide the exact output required by the users.

2. Security

The key challenge of any online and offline software is its vulnerability against hackers and spammers. The database of any software plays a vital role in a client's business.

If, in any case, the database is compromised and data is shared with unauthorized sources may badly harm business and may negatively affect personal and professional life.

Therefore, **characteristics of good software and Qualities of good software** entirely depend upon the software developers while developing the software that takes preventive measures to avoid hacking.

The security threat is and will remain the primary threat to any software. There is no permanent solution to this problem; therefore, this issue must be taken seriously and handled accordingly.

3. Efficiency

The job or work of good and quality software is to provide exact and correct results required by the user.

In this process of operation, the program needs to be more efficient, which enhances its creditability.

They must have some characteristics like memory management and memory resources.

The effectiveness of any software is seen when it produces results with speed and accuracy.

4. Easy to Maintain

This is the primary objective of any software that must be easy to maintain and update user requests or demands.

Any software is imperfect; therefore, the need for improvements doors is always open as the business is not working traditionally.

They are ever-changing and incorporate modern techniques and methods; therefore, updating and enhancing the software is always on the cards.

Good and quality software engineers have to build software in such a fashion that can be easily modified, edited, and updated with the everchanging demands of the users.

This software needs to be maintained easily without any issues that make software good software.

This feature of the software is also considered one of the significant and essential attributes of good software.

5. Usability

The software's usability depends upon many factors, which are mentioned below.

- User-friendliness
- Security features
- Simplicity
- Ease of Using
- Platform Independent
- Supportability

What is Software Engineering?

The term **software engineering** is the product of two words, **software**, and **engineering**.

Engineering is the application of **scientific** and **practical** knowledge to **invent, design, build, maintain, and improve frameworks, processes, etc.**

Software Engineering is an engineering branch related to the evolution of software product using well-defined scientific principles, techniques, and procedures.

- In software industry we understand that there is an urgent need to use a proper strategies, process and development cycles so that we can produce, or we can design quality products that are within budget, with in time and must satisfy the requirement of their users.
- Software engineering is the systematic application of engineering principles and methods to the design, development, testing, and maintenance of software products. It involves the use of various tools, techniques, and methodologies to manage the software development process and ensure the quality, reliability, and maintainability of software products.

Software Engineering is the **systematic, Disciplined, Cost effective techniques** for software development.

Engineering approach to develop a software

The result of software engineering is an effective and reliable software product.

Why Software Engineering? Software Crisis & its Solution: What was the Software Crisis?

- ❖ It was in the late 1960s when many software projects failed.
- ❖ Many software became over budget. Output was an unreliable software which is expensive to maintain.
- ❖ Larger software was difficult and quite expensive to maintain.
- ❖ Lots of software not able to satisfy the growing requirements of the customer.
- ❖ Complexities of software projects increased whenever its hardware capability increased.
- ❖ Demand for new software increased faster compared with the ability to generate new software.
- ❖ All the above issues lead to 'Software Crisis.'



Y2K Problem Detail:

The Y2K bug was a computer flaw, or bug, that may have caused problems when dealing with dates beyond December 31, 1999. The flaw, faced by computer programmers and users all over the world on January 1, 2000, is also known as the "millennium bug."

(The letter K, which stands for kilo (a unit of 1000), is commonly used to represent the number 1,000. So, Y2K stands for Year 2000.) As the year 2000 approached, computer programmers realized that computers might not interpret 00 as 2000, but as 1900. Activities that were programmed on a daily or yearly basis would be damaged or flawed.

Arian-5 space rocket

In 1996, Arian-5 space rocket, developed at the cost of \$7000 million over a period of 10 years was destroyed within less than a minute after its launch. The crash occurred because there was a software bug in the rocket guidance system.

Banks of US Problem

In 1996, one of the largest banks of US credited accounts of nearly 800 customers with approximately \$924 lacs. Later, it was detected that the problem occurred due to a programming bug in the banking software.

The Solution

Solution was to the problem was transforming unorganized coding effort into a software engineering discipline. These engineering models helped companies to streamline operations and deliver software meeting customer requirements.

- The late 1970s saw the widespread uses of software engineering principles.
- In the 1980s saw the automation of software engineering process and growth of (CASE) Computer-Aided Software Engineering.
- The 1990s have seen an increased emphasis on the 'management' aspects of projects standard of quality and processes just like ISO 9001

Why is Software Engineering required?

Software Engineering is required due to the following reasons:

- To manage large software
- For more Scalability
- Cost Management
- To manage the dynamic nature of software

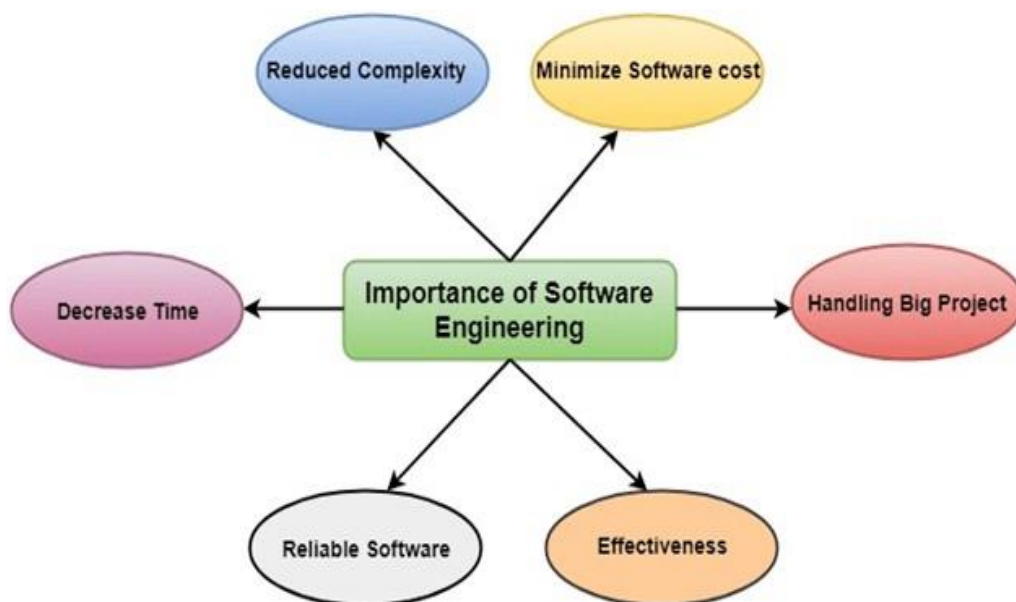
- For better quality Management

Need of Software Engineering

The necessity of software engineering appears because of a higher rate of progress in user requirements and the environment on which the program is working.

- **Huge Programming:** It is simpler to manufacture a wall than to a house or building, similarly, as the measure of programming become extensive engineering has to step to give it a scientific process.
- **Cost:** As the hardware industry has demonstrated its skills and huge manufacturing has let down the cost of computer and electronic hardware. But the cost of programming remains high if the proper process is not adapted.
- **Dynamic Nature:** The continually growing and adapting nature of programming hugely depends upon the environment in which the client works. If the quality of the software is continually changing, new upgrades need to be done in the existing one.
- **Quality Management:** Better procedure of software development provides a better and quality software product.

Importance of Software Engineering



1. **Reduces complexity:** Big software is always complicated and challenging to progress. Software engineering has a great solution to reduce the complication of any project. Software engineering divides big problems into various small issues. And then start solving each small issue one by one. All these small problems are solved independently to each other.

2. **To minimize software cost:** Software needs a lot of hard work and software engineers are highly paid experts. A lot of manpower is required to develop software with a large number of codes. But in software engineering, programmers project everything and decrease all those things that are not needed. In turn, the cost for software productions becomes less as compared to any software that does not use software engineering method.
3. **To decrease time:** Anything that is not made according to the project always wastes time. And if you are making great software, then you may need to run many codes to get the definitive running code. This is a very time-consuming procedure, and if it is not well handled, then this can take a lot of time. So if you are making your software according to the software engineering method, then it will decrease a lot of time.
4. **Handling big projects:** Big projects are not done in a couple of days, and they need lots of patience, planning, and management. And to invest six and seven months of any company, it requires heaps of planning, direction, testing, and maintenance. No one can say that he has given four months of a company to the task, and the project is still in its first stage. Because the company has provided many resources to the plan and it should be completed. So to handle a big project without any problem, the company has to go for a software engineering method.
5. **Reliable software:** Software should be secure, means if you have delivered the software, then it should work for at least its given time or subscription. And if any bugs come in the software, the company is responsible for solving all these bugs. Because in software engineering, testing and maintenance are given, so there is no worry of its reliability.
6. **Effectiveness:** Effectiveness comes if anything has made according to the standards. Software standards are the big target of companies to make it more effective. So, Software becomes more effective in the act with the help of software engineering.

Difference between Software Engineering and Computer Science

Here are the important differences between Software Engineering VS Computer Science:

Parameter	Software Engineering	Computer Science
Definition	Software engineering is defined as a process of analyzing user requirements and then designing, building, and testing software applications.	Computer science is a discipline that involves the design and understanding of computers and computational processes.

Meanings	Software Engineering is a study of how software systems are built.	Computer Science is the study of how computers perform theoretical and mathematical jobs.
Selection	You should choose Software Engineering if you want to learn the overall life cycle of how specific software is built and maintained.	You should choose Computer Science if you want to get into a specialized field in CS like artificial intelligence, machine learning, security, or graphics.
Project management	Students of software engineering will likely take courses on project management, both in undergraduate and graduate programs.	Project management is often included in the computer science curriculum. Mostly as part of a software engineering course.
Course include	In Software Engineering, you will also learn programming languages and general computing principles.	Computer science students will study how data is stored, processed, and applied on various other computing devices.
Scope	Emerging occupations related to software engineering depend on the state of software and technology in the future.	It is a field of computer science which also includes careers in cloud computing and AI technology .
Developer	The average salary for a software engineer is \$107,932 per year.	The average salary for a computer scientist is \$103,643 per year in the United States.

Difference between Software Engineering and System Engineering

Software Engineering	System Engineering
Software Engineering is a course in the Computer Science branch in engineering colleges. Software Engineering deals with the computer software.	System Engineering is a course which deals with the computer system hardware and the computer networks.
After completing the Software Engineering course one can develop the software	After completing the System Engineering course one can resolve all the issues related to the computer hardware as

application and can test the software's.	well as the computer networks.
A software engineer can maintain all the software related problems and can also handle the large software's and updates of the software's.	A system engineer can maintain the system hardware and in the development of the hardware based on the software.
The software engineers get more salaries when compared to the System engineers.	The System engineers get less salary when compared with the salary of Software engineers.
The Software engineers are more focused on the process of developing compete and efficient software.	The System engineers are more focused on the users and the domains of the computer systems.