Unit 1 – Chapter 2
Introduction to Software Engineering

Outcomes

- Differentiate between a Program and Software
- Gain more understanding about the Software crisis
- Characteristics of Software
- Types of Software

2.0 Introduction

To develop a sound footing in software engineering, knowledge about the items listed below shall be helpful.

- Program vs. Software
- Complexity of software development requiring engineering approach
- ISO measures of software characteristics
- Types of Software

2.1 Program vs. Software

During your academic time, you are mostly dealing with programs. Once you enter the software industry, you shall be working on software projects. It is helpful to understand the differences between a program and software.

Attributes	Program	Software
Description	A program is a set of instructions written in a programming language used to execute a specific task or particular function.	Software is a collection of several programs and other procedures and documentation
Flexibility	A program may not be software.	Software can be a program.

Consists of	A program consists of a set of instructions that are coded in a programming language like c, C++, PHP, Java etc.	Software consists of bundles of programs and data files. Programs in specific software use these data files to perform dedicated types of tasks.
Functionality & Features	Program has limited functionality and fewer features.	Software has lots of functionality and features such as GUI, input/output data, process etc.
Dependability	Program functionality is dependent on the compiler.	Software functionality is dependent on the operating system.
Creation Time	A program takes less time to build/make.	Software takes relatively more time to build/make when compared to a program.
Development Approach	Program development approach may be un-procedural, unorganized and unplanned.	Software development approach is systematic, organized and very well planned.
Size	The size of a program may range from kilobytes (KB) to megabytes (MB).	The size of software may range from megabytes (MB) to Gigabytes (GB).
Documentation	May have patchy documentation mostly technical	Comprehensive documentation including user manuals

After perusing the above table the major differences are of size, complexity, and breadth of functionality.

2.2 Characteristics/Features of Software (based on ISO 9126 Software Quality Model)

Software is defined as a collection of data, programs, procedures, and associated documentation and rules. Software is an abstract logical system and that makes it different from other things human beings build. Software is developed or engineered; it is not manufactured like conventional products we are familiar with like cars, toys, machines. The software doesn't "wear out.” but it deteriorates (due to changes). Hardware has bathtub curve of failure rate (high failure rate in the beginning, then drop to steady-state, then loss of function due to cumulative effects of dust, vibration, abuse occurs). The ISO 9126–1 software quality model identifies 6 main quality characteristics, namely:

Functionality

Functionality means the ability to perform a task or a function which any product or service is designed. For example, a light bulb should be able to provide light based on the specified parameters. In the case of software, functionality refers to the suitability, accuracy, interoperability, compliance, security features of the software to its intended purpose.

Reliability

Once a software system is functioning, as specified, and delivered the reliability characteristic defines the capability of the system to maintain its service provision under defined conditions for defined periods. For example, it could mean the failure rate of the software is once per million hours of operation.

Usability

It is the extent to which the software can be utilized with ease and the amount of effort or time required to learn how to use the software.

Efficiency

It is the ability of software to use the resources of the system most effectively and efficiently. Software must make effective use of system storage, memory, and network resources and execute a command as per required timing.

Maintainability

It is the ease with which the modifications can be made in software to extend or enhance its functionality, improve its performance, or resolve bugs.

Portability

This characteristic refers to how well the software can adapt to changes in its environment. It is the ease with which software developers can re-launch software from one platform to another, without (or with a minimum) changes. In simple terms, software must be made in a way that should be platform-independent.

2.3 More on Software Crisis

Software crisis is the result of the incremental failure rate of software development which led to an incomplete and degrading performance of software products. Software crisis can be traced to unplanned, poorly written, hard to read, error-prone software that often lacks good documentation.

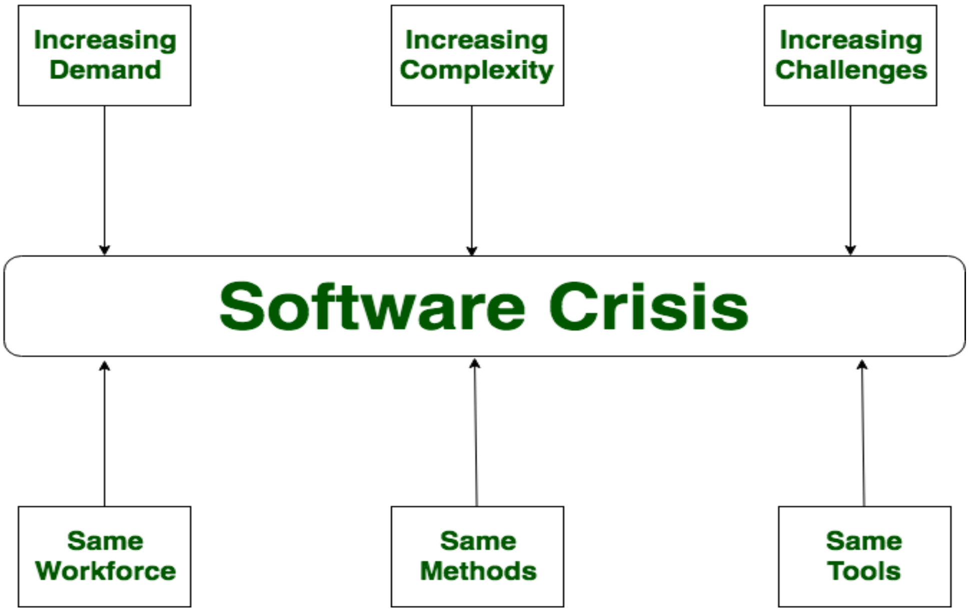

Figure 1.2.1: Software Crisis

2.3.1 Software Crisis in terms of statistics

Below are statistical figures that illustrate the outcomes of typical software projects undertaken. From the data, it is clear that there is a lot of scope for improvement in Software project management.

- Success 16%
- Failure 31%
- Over budget 53%

2.3.2 Problems due to the software crisis

Below are some of the consequences of software crisis

- Projects running over-budget
- Projects running over-time
- Software very inefficient
- Projects were unmanageable and code difficult to maintain
- Software of low quality
- The software often did not meet requirements

2.3.3 Factors contributing to the software crisis

Listed are the factors contributing to the software crisis.

- Large and complex problems,
- Lack of inadequate training in software engineering,
- Increasing skill shortage,
- Low productivity improvements,
- Requirement change being inevitable,
- Requirements not getting frozen,
- Customer is not clear with their requirements,
- Market demand/conditions change
- Manpower turnover,
- Technology/Process change

2.3.4 Illustration of a famous Software Crisis

In 19th-century date format used was **ddmmyy** (say 1st Jan 1999 was written as 010199 and it year is assumed to be 1999 by default). But, when the 20th century came, (say 1st Jan 2000 was written as **010100** and according to old format the year was assumed to be 1900). Software that used two digits to represent years produced incorrect results or failed.

To store the complete year we require a width of 4 places. In some of the legacy code, the year was represented using a width of 2 places. As the year was represented using a width of 2 digits, after 99 the year would display as 00. This aspect would cause problems when using the year for data processing. The old format of the date was **ddmmyy**. The bug was fixed by changing the format of the date to **ddmmyyyy**.

2.4 Types of Software

The most significant factor in determining which software engineering methods and techniques are used is dependent on the type of software that is being developed. The features of different types of software are summarized below.

System Software: A collection of programs written to manage the resources of the computer system and simplify applications programming Examples are Compiler, device driver, editors, operating system. They require closer knowledge of hardware to develop.

Application Software: It is used to address a specific need for an end-user. Examples of application software are Word, Excel, and PowerPoint among others.

Scientific / Engineering Software: category of software used to facilitate the engineering functions and tasks. For example, software used in applications like CAD, astronomy, automotive stress analysis, molecular Biology, weather forecasting, space Shuttle orbital dynamic, automated manufacturing.

Embedded Software: Embedded software is a piece of software that is embedded in hardware or non-PC devices. It is written specifically for the particular hardware that it runs on and usually has processing and memory constraints because of the device’s limited computing capabilities. Examples are software in mobile phones, software in Anti Lock Braking in a car, software in a microwave oven to control the cooking process.

Web application: A web application (or web app) is application software that runs on a web server, unlike computer-based software programs that are run locally on the operating system (OS) of the device. Web applications are accessed by the user through a web browser with an active network connection.

Artificial intelligence software: Software that is capable of intelligent behaviour. Creating intelligent software involves simulating several capabilities, including reasoning, learning, problem-solving, perception, and knowledge representation. These include- robotic, expert systems, pattern recognition, image and voice, artificial neural network, game playing, theorem proving. They solve extremely Complex problems.

Networking Software: Networking Software provides the required support for computers and devices to interact. The networking software is also used when software is running on a network of computers (such as the World Wide Web). It includes all network management software, server software, security and encryption software and software to develop web-based applications like HTML, PHP, XML, etc.

Business Software: This category of software is used to support business applications and is the most widely used category of software. Examples are software for inventory management, accounts, banking, hospitals, schools, stock markets, etc.

Entertainment Software: Education and entertainment software provides a powerful tool for educational agencies, especially those that deal with educating young children. There is a wide range of entertainment software such as computer games, educational games, translation software, mapping software, etc.

Utility Software: The programs coming under this category perform specific tasks and are different from other software in terms of size, cost and complexity. Examples are anti-virus software, voice recognition software, compression programs, etc.

Document Management Software: Document Management Software is used to track, manage and store documents to reduce the paperwork. Such systems are capable of keeping a record of the various versions created and modified by different users (history tracking). They commonly provide storage, versioning, metadata, security, as well as indexing and retrieval capabilities.

Reservation Software: A Reservation system is primarily used to store and retrieve information and perform transactions related to travel, car rental, hotels, or other activities.

2.5 Conclusion

By understanding the difference between a program and software, we are able to appreciate the challenges in developing software applications and the need for software engineering. Construction of software is one the most challenging human endeavor and developing software applications without a systematic approach aided by engineering principles is doomed to fail.

***