Institute of Computer Science
Vikram University, Ujjain (M.P.)

SOFTWARE ENGINEERING

M.Sc. Computer Science

BY

Mr. Shekhar Disawal

SOFTWARE ENGINEERING M.SC 4 SEM

SOFTWARE

Software can be defined as a collection of programs, documentation and operating procedures. Institute of Electrical and Electronic Engineers (IEEE) defines software as "a collection of computer programs, procedures, rules, and associated documentation and data". Software possesses no mass, no volume, and no colour, which makes it a nondegradable entity over a long period. Software does not wear out or get tired. According to the definition of IEEE, software is not just programs, but includes all the associated documentation and data.

Software is responsible for managing, controlling, and integrating the hardware components of a computer system and to accomplish any given specific task. Software instructs the computer about what to do and how to do it. For example, software instructs the hardware how to print a document, take input from the user, and display the output.

Computers need instructions to carry out the intended task. These instructions are given in the form of computer programs. Computer programs are written in computer programming languages, such as C, C++, and so on. A set of programs, which is specifically written to provide users a precise functionality like solving specific problem is termed as software package.

For example, an accounting software package helps users in performing accounting related activities.

Software Characteristics:

Different individuals judge software on different basis. This is because they are involved with the software in different ways. For example, users want the software to perform according to their requirements. Similarly, developers involved in designing, coding and maintenance of the software evaluate the software by looking at the internal characteristics of the products, before delivering it to the user. Software characteristics are classified into six major components. These components are listed below:

• **Functionality:** Refers to the degree of performance of the software against its intended purpose.

• **Reliability:** Refers to the ability of software to perform a required function under given conditions for a specified period.

• **Usability:** Refers the degree to which software is easy to use.

• **Efficiency:** Refers to the ability of software to use system resource in the most effective and efficient manner.

• **Maintainability:** Refers to the ease with which a software system can be modified to add capabilities, improve system performance, or correct errors.

• **Portability:** Refers to the ease with which software developers can transfer software from one platform to another, without (or with minimum) changes. In simple terms, it refers to the ability of software to function properly on different hardware and software platforms without making any changes in it.

In addition to the above-mentioned characteristics, robustness and integrity are also considered to be important. Robustness refers to the extent to which software can continue to operate correctly despite the introduction of invalid input, while integrity refers to the extent to which unauthorised access or modification of software or data can be controlled in the computer system.

Software Components:

A software component is defined as an independent executable software element with a well-defined set of inputs, outputs and interface. All the services provided by a component are made available through the interface and all the interactions with the component are done through that interface. Council and Heinmann define the term software component as follow.

"*A software component is a software element that conforms to a component model and can be independently deployed and composed without modification according to a composition."*

Szyperski describes the term as follows.

"A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third parties."

The software engineering that emphasizes the design and development of computer-based systems using software components is called component-based software engineering (CBSE). The main objective of CBSE is to standardize the interfaces between software components so that they can be assembled easily to develop new software. The basic idea behind CBSE is to reuse the existing components. The components developed for a specific application have to be generalized to make them reusable. In other words, the more generalized interface, the

greater the reusability. Apart from the advantage of reusability, the components have the following advantages.

- ✓ The components are independent and hence, they do not interfere with each other and are easy to modify.
- ✓ The inner workings of the components are hidden from the user.
- ✓ The components do not have to be compiled prior to their use with other components.
- ✓ The communication and interaction with the components are done through well-defined interfaces.
- ✓ The component platforms are shared and hence, the development costs are reduced.

There are some characteristics that a software program must possess before it qualifies as a component. These are given below.

- ➢ It should be independent and deployable, that is, it has to be a self-contained and standalone entity and it should not depend on other software components for its use.
- ➢ It should provide some pre-defined interfaces and all interactions must take through these interfaces.
- ➢ It should have a complete documentation so that the users of the component can decide whether or not the component is meeting their needs.
- ➢ It has to conform to some specified standards.
- ➢ It should be language-independent.

Software Components Applications:

Software can be applied in countless situations, such as in business, education, social sector, and in other fields. The only thing that is required is a defined set of procedural steps. That is, software can be engaged in any field, which can be described in logical and related steps. Every software is designed to suit some specific goals. These goals are data processing, information sharing, communication, and so on. Software is classified according to the range of potential applications. These classifications are listed below:

• **System software:** This class of software is responsible for managing and controlling operations of a computer system. System software is a group of programs rather than one program and is responsible for using computer resources efficiently and effectively.

For example, operating system is system software, which controls the hardware, manages memory and multi-tasking functions, and acts as an interface between applications programs and the computer.

• **Real-time software:** This class of software observes, analyses, and controls real world events as they occur. Generally, a real-time system guarantees a response to an external event within a specified period of time. For example, real-time software is used for navigation in which the computer must react to a steady flow of new information without interruption. Most of the defence organisations all over the world use real-time software to control their military hardware.

• **Business software:** This class of software is widely used in areas where the management and control of financial activities is of utmost importance. The fundamental component of a business software comprises of payroll, inventory, accounting, and software that permits user to access relevant data from the database. These activities are usually performed with the help of specialised business software that facilitates efficient framework in the business operation and in management decisions.

• **Engineering and scientific software:** This class of software has emerged as a powerful tool to provide help in the research and development of next generation technology. Applications, such as study of celestial bodies, study of under-surface activities, and programming of orbital path for space shuttle are heavily dependent on engineering and scientific software. This software is designed to perform precise calculations on complex numerical data that are obtained during real-time environment.

• **Artificial intelligence (AI) software:** This class of software is used where the problem-solving technique is non-algorithmic in nature. The solutions of such problems are generally non-agreeable to computation or straightforward analysis. Instead, these problems require specific problem-solving strategies that include expert system, pattern recognition, and game playing techniques. In addition, it involves different kinds of searching techniques including the use of heuristics. The role of artificial intelligence software is to add certain degree of intelligence into the mechanical hardware to have the desired work done in an agile manner.

• **Web-based software:** This class of software acts as an interface between the user and the Internet. Data on the Internet can be in the form of text, audio, or video format, linked with hyperlinks. Web browser is a web-based software that retrieves web pages from the Internet. The software incorporates executable instructions written in special scripting languages, such as CGI or ASP. Apart from providing navigation on the web, this software also supports additional features that are useful while surfing the Internet.

• **Personal computer (PC) software:** This class of software is used for official and personal use on daily basis. The personal computer software market has grown over the last two decades

from normal text editor to word processor and from simple paintbrush to advance image-editing software. This software is used predominantly in almost every field, whether it is database management system, financial accounting package, or a multimedia-based software. It has emerged as a versatile tool for daily life applications.

SOFTWARE ENGINEERING: DEFINITION

IEEE defines software engineering as "*the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software*." In a nutshell, software engineering can be defined as the technological and managerial discipline concerned with systematic production and maintenance of software that is developed and modified on time and within cost estimates.

Software engineering is a discipline, which can be described as the combination of techniques of engineering and all aspects of software development. This includes design, implementation, and maintenance of software. It includes standardised approach to program development, both in its managerial and technical aspects.

The foundation for software engineering lies in the good working knowledge of computer science theory and practice. The theoretical background involves knowing how and when to use data structures, algorithms, and understanding what problems can be solved and what cannot. The practical knowledge includes through understanding of the workings of the hardware as well as thorough knowledge of the available programming languages and tools.

One of the main objectives of software engineering is to help developers obtain high-quality software. This quality is achieved through use of *Total Quality Management*, which enables continuous process improvement custom that leads to the development of more established approaches to software engineering.

software engineer should possess the following qualities:

• **Problem solving skills:** Software engineer should develop algorithms and solve programming problems.

• **Programming skills:** Software engineer should be well versed in data structures and algorithms, and must be expert in one or more programming languages and possess strong programming capabilities.

• **Design approaches:** Software engineer should be familiar with numerous design approaches required during the development of software, at the same time, they should be able to translate

ambiguous requirements and needs into precise specifications, and be able to converse with the use of a system in terms of applications.

• **Software technologies:** Software engineer should have good understanding of software technologies. Ability to move among several levels of abstractions at different stages of the software project, from specific application procedures and requirements to the detailed coding level is also required.

• **Project management:** Software engineer should know how to make a project work, on time and on budget, in order to produce quality applications and systems.

• **Model of the application:** Software engineer should be able to create and use a model of the application to guide choices of the many trade-offs that will be faced by him.

In addition to the above-mentioned qualities, software engineer should have good communication and interpersonal skills. Moreover, knowledge of object-orientation, quality concept, International Organization of Standardization (ISO standards), and Capability Maturity Model (CMM) are also required. The tasks performed by software engineers have evolved rapidly, which has resulted in new areas of specialization and changing technology. Software engineers often work as part of a team that designs new hardware and software. This team comprises of engineering, marketing, manufacturing, and designing people who work together until the software is released.