

Lecture No. 1

Introduction to Software Engineering

☐ What is Software?

When we write a program for computer we named it as software. But software is not just a program; many things other than the program are also included in software.

Some of the constituted items of software are described below.

☐ Program: The program or code itself is definitely included in the software.

☐ Data: The data on which the program operates is also considered as part of the software.

☐ Documentation: Another very important thing that most of us forget is documentation. All the documents related to the software are also considered as part of the software.

So the software is not just the code written in Cobol, Java, Fortran or C++. It also includes the data and all the documentation related to the program.

☐ Why is it important?

Undoubtedly software is playing a vital role in all the field of life these days. We can see many software applications being operated around us in our daily routine.

Some of the major areas in which software has played an important role are identified as under.

☐ Business decision-making: Software systems have played a major role in businesses where you have to analyze your data and on the basis of that analysis you have to make business decisions. This process of data analysis and decision-making has become very accurate and easy by the use of software.

☐ Modern scientific investigation and engineering problem solving: Scientific investigations and engineering problem solving require an intensive amount of calculations and data analysis. The accuracy of these analyses is also very important in scientific applications. This process has become very easy and accurate by the use of software. For example software systems are becoming more

involved in bioinformatics and the process of DNA decoding is only possible by the use of software systems.

□ Games: We see many computer games these days that interests people of all ages. All these games are drive through software systems.

□ Embedded systems: We see many kinds of gadgets being employed in our daily used things, like small microcontrollers used in our cars, televisions, microwave ovens etc. All these systems are controlled through the software.

Similarly in many other fields like education, office automation, Internet applications etc, software is being used. Due to its central importance and massive use in many fields it is contributing a lot in terms of economic activity started by the software products. Billions and trillions of dollars are being invested in this field throughout the world every year.

□ Engineering

Before moving on to software engineering lets first discuss something about engineering itself. If you survey some of the dictionaries then you will find the following definition of engineering.

“The process of productive use of scientific knowledge is called engineering.”

1.1 Difference between Computer Science and Software Engineering

There are many engineering fields like electrical, mechanical and civil engineering. All these branches of engineering are based on physics. Physics itself is not engineering but the use of physics in making buildings, electronic devices and machines is engineering. When we use physics in constructing buildings then it is called civil engineering. When we use physics in making machines like engines or cars then it is called mechanical engineering. And when we apply the knowledge of physics in developing electronic devices then the process is called electrical engineering. The relation of computer science with software engineering is similar as the relation of physics with the electrical, mechanical or civil engineering or for that matter the relation of any basic science with any engineering field. So in this context we can define software engineering as: ”This is the process of utilizing our knowledge of computer science in effective production of software systems.”

What is software?

Computer programs and associated documentation. Software products may be developed for a particular customer or maybe developed for general market.

What are the attributes of good software?

Good software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable.

What is software engineering?

Software engineering is an engineering discipline that is concerned with all aspects of software production.

Goals of software engineering

The primary goals of software engineering are:

- To improve the quality of the software products.
- To increase the productivity
- To give job satisfaction to the software engineers

Software Quality Models:-

Flexibility and Extensibility

Flexibility is the ability of software to add/modify/remove functionality without damaging the current system. Extensibility is the ability of software to add functionality without damaging system, so it may be thought of as a subset of flexibility.

Maintainability and Readability

Maintainability is a little similar with flexibility but it focuses on modifications about error corrections and minor function modifications, not major functional

extensibilities. It can be supported with useful interface definitions, documentation, and self-documenting code and/or code documentation. The more correct and useful documentation exists, the more maintainability can be performed.

Performance and Efficiency

Performance is mostly about the response time of the software. Efficiency must be supported with resource utilization. Optimal source/performance ratio must be aimed.

Scalability

A scalable system responds user actions in an acceptable amount of time, even if load increases.

Availability, Robustness, Fault Tolerance and Reliability:

A robust software should not lose its availability even in most failure states. Even if some components are broken down, it may continue running. Besides, even if whole application crashes, it may recover itself using backup hardware and data with fault tolerance approaches. There should always be B and even C, D .. plans. Reliability also stands for the integrity and consistency of the software even under high load conditions.

Usability and Accessibility

User interfaces are the only visible parts of software according to the viewpoint of user. So, simplicity, taking less time to complete a job, fast learnability etc. are very important in this case.

Platform Compatibility and Portability

A quality software should run on as much various platforms as it can. So, more people can make use of it. In different contexts we may mention different platforms, this may be OS platforms, browser types etc. And portability is about adapting software that can run on different platforms, for being more platform compatible. In this sense, portability is also related with flexibility.

Testability and Manageability

Quality software requires quality testing. Source code should be tested with the most coverage and with the most efficient testing methods. This can be performed by using encapsulation, interfaces, patterns, low coupling etc. techniques correctly.

Creating a successful logging system is another very important issue about manageability.

Security

Security is a very important issue on software development, especially for web or mobile based ones which may have millions of users with the ability of remote accessing to system. You should construct a security policy and apply it correctly by leaving no entry points.

Functionality and Correctness

Functionality (or correctness) is the conformity of the software with actual requirements and specifications.

Quality factors are not meaningful when we are talking about unfunctional software. First, perform desired functionality and produce correct software, then apply quality factors on it. If you can perform both parallelly, it is the best.