**SOFTWARE EVALUATION**

**REVIEW OF SOFTWARE AND ENGINEERING**

**Introduction**

Software engineering is derived from two terms:

* Software
* Engineering

**SOFTWARE**

Is the collection of executable programming code, associated libraries and documentations

**ENGINEERING**

Is the development of products, using well-defined, scientific principles and methods.

Engineering can also be defined as application of sciences, tools and methods to find a cost effective solution to problems

**NB:**

* An application is a software that performs a specific function directly for the user or for another application program.
* Application is a package to perform a specific task.
* A program is a set of instructions written in a computer-readable language to perform a specific task on the computer.

**SOFTWARE ENGINEERING**

Definition: Is the systematic, disciplined, quantifiable (cost effective) approach to the development, operation, and maintenance of software; that is, the application of engineering to software.

Software engineering can also be defined as engineering technique to develop software.

**CHARACTERISTICS OF SOFTWARE.**

* Software is developed or engineered it’s not manufactured
* Software does not ware
* Software is custom.

**CHARACTERISTICS OF GOOD SOFTWARE**

The following are the characteristics of a good software

* Budget
* Usability
* Efficiency
* Reusability
* Adaptability
* Correctness
* Modularity
* Maintainability
* Functionality
* Dependability
* Security
* Safety
* Flexibility
* Scalability
* Portability
* Interoperability

**NEEDS OF SOFTWARE**

The following are the reasons why software is required

* To handle large software
* To increase scalability
* To control costs
* For improved quality management
* To control the dynamic nature of software

**IMPORTANCE OF SOFTWARE ENGINEERING**

The following are the importance of software engineering

**Save time**

* Delivering software projects on time is one of the most important criteria for determining project success, and it's also one of the most difficult.
* Schedule conflicts are the most common cause of project disputes, especially in the second half of projects when the real execution of the final working product takes place.
* Software engineering refers to the procedures that must be followed to ensure that a project is completed on schedule.

**Managing large projects**

* Large projects take a long time to complete and require a lot of work, planning, and administration. And it takes a lot of planning, guidance, testing, and maintenance to invest six to seven months of any organization.
* Companies devote a lot of resources to completing a plan, thus to handle a large project without problems, the organization must use software engineering approaches.

Red**uces complexity**

* Large software systems are inherently complex and difficult to develop; software engineering splits large problems into smaller challenges.
* All of these tiny challenges are solved independently of one another and then combined together to generate the software product

**Reliable software**

* Software reliability is defined as the "chance that the software will operate without failure for a specific amount of time."
* Software engineering creates frameworks for measuring and evaluating software quality. Models for estimating flaws inserted and removed over the software lifetime are one example.

**To reduce software costs**

* Developing large-scale software systems necessitates a lot of resources, such as people, software licenses, hardware, and so on.
* As Companies strive to develop cutting-edge software to fuel growth, establishing the overall budget becomes increasingly difficult.
* Software engineering gives a method for having regular engagement and collecting budget estimations in a methodical manner.

**Effectiveness**

* Effectiveness entails achieving the intended outcomes/doing the correct thing!
* Examples include completing missions and goals, generating satisfied consumers, and providing high-quality work.
* Effective use of communication, technology, organizational and individual knowledge, and resources are all part of this.

**QUALITIES OF A GOOD SOFTWARE ENGINEER**

* Familiarity with software engineering principles, as well as exposure to systematic approaches.
* Strong technical understanding of the project's scope (Domain knowledge).
* High levels of motivation.
* Ability to work in a team and have a solid understanding of computer science fundamentals
* Discipline
* Strong programming skills.
* Excellent communication abilities. These abilities include oral, writing, and interpersonal abilities.