**Software Engineering**

**1- Introduction**

when a computer software succeeds- when it meets the needs of the people who use it, when it performs flawlessly over a long period of time, when it is easy to modify and even easier to use it can does change things for the better. But when software fails- when its users are dissatisfied, when it is error prone, when it is difficult to change and even harder to use- bad things can and do happen. It is necessary to build software that makes things better, avoiding the bad things that lurk in the shadow of failed efforts. The successful need discipline when software is designed and built and need an engineering approach.

**1.1 What is software?**

Software is computer programs and associated documentation such as requirements, design models and user manuals. Software products may be developed for particular customer or may be developed for a general market. New software can be created by developing new programs, configuring generic software systems or reusing existing software. Software products may be.

**Generic-** developed to be sold to a range of different customers e.g PC software such as Excel or Word.

**Bespoke (custom) -** developed for a single customer according to their specification.

**1.2 Software Characteristics**

To gain an understanding of software, it is important to examine the characteristics of software that make it different from other things that human beings build. When hardware is built, the human creative process (analysis, design, construction, testing) is ultimately translated into a physical form. If we build a new computer, our initial sketches, formal design drawings, and bread boarded prototype evolve into a physical product (chips, circuit boards, power supplies, etc). Software is a logically rather than a physical system element. Therefore, software has characteristics that are considerably different than those of hardware:

1. Software is developed or engineered; it is not manufactured in the classical sense.
2. Software doesn’t “wear out” when a hardware component wears out; it is replaced by a spare part. There are no software spare parts. Every software failure indicates an error in design or in the process through which design was translated into machine executable code. Therefore, software maintenance involves considerably more complexity than hardware maintenance.
3. Although the industry is moving toward component- based assembly, most software continues to be custom built.

**1.3 Software Applications**

Software may be applied in any situation for which a prespecified set of procedural steps (i.e., an algorithm) has been defined. The following software areas indicate the breadth of potential applications:

1. **System software:** system software is a collection of programs written to service other programs. Some system software (e.g., compilers, editors, and file management utilities) process complex, but determinate, information structures.
2. **Real-time software:** software that monitors/analyzes/controls real-world events as they occur is called real time. Elements of real-time software include a data gathering component that collects and formats information from an external environment.
3. **Business software:** business information processing is the largest single software application area. Applications in this area restructure exiting data in a way that facilitates business operations or management decision making.
4. **Engineering and scientific software:** modern applications within the engineering/ scientific area are moving away from conventional numerical algorithms. Computer- aided design, system simulation, and other interactive applications have begun to take on real-time and even system software characteristics.
5. **Embedded software:** Intelligent products have become commonplace in nearly every consumer and industrial market. Embedded software resides in read-only memory and is used to control products and systems for the consumer and industrial markets. Embedded software can perform very limited functions (e.g., keypad control for a microwave oven) or provide significant function and control capability (e.g., digital functions in an automobile such as fuel control, dashboard displays, and braking systems).
6. **Personal computer software:** word processing, spreadsheets, computer graphics, multimedia, environment, database management, personal and business financial applications, external network, and database access are only a few of hundreds of applications.
7. **Web-base software:** the web pages retrieved by a browser are software that incorporates executable instructions (e.g., HTML, or Java), and data (e.g., hypertext and a variety of visual and audio formats).
8. **Artificial intelligence software:** Artificial intelligence (AI) software makes use of non-numerical algorithms to solve complex problems that are not amenable to computation or straight forward analysis. Expert systems, also called knowledge based systems, pattern recognition (image and voice), artificial neural networks, theorem proving, and game playing are representative of applications within this category.

**1.4 Software Crisis:**

Software Crisis has characterized the problems associated with software development. It includes slow, evolutionary change punctuated by explosive technological changes in disciplines associated with software.

**1.5 Software engineering, Computer Science and System Engineering.**

**Software engineering** is an engineering discipline that is concerned with all aspects of software production. Software engineers should adopt a systematic and organized approach to their work and use appropriate tools and techniques depending on the problem to be solved, the development constraints and the resources available. Software engineering is concerned with the practicalities of developing and delivering useful software.

**Computer science** is concerned with theory and fundamentals. Computer science theories are still insufficient to act as a complete underpinning for software engineering (unlike e.g physics and electrical engineering).

**System engineering** is concerned with all aspects of computer-based systems development including hardware, software and process engineering. Software engineering is part of this process concerned with developing the software infrastructure, control, applications and databases in the system. System engineering is involved in system specification, architectural design, integration and deployment.

**1.6 Attributes of Good Software**

The software should deliver the required functionality and performance to the user and should be maintainable, dependable and acceptable.

* Maintainability: software must evolve to meet changing needs;
* Dependability: software must be trustworthy;
* Efficiency: software should not make wasteful use of system resources;
* Acceptability: software must accepted by the users for which it was designed. This means it must be understandable, usable and compatible with other systems.