1. Software Engineering

According to IEE Software engineering can be defined as:
Software Engineering: (1) The application of a **systematic**, **disciplined**, **quantifiable** approach to the **development**, **operation**, and **maintenance** of software; that is, the application of engineering to software.
(2) The study of approaches as in (1).

- Software product means software for a large/medium size and complex problem.
- We get the real advantage of software engineering when it is applied to a project.
- Though it can also be used for the development of programs/small assignments.

A definition proposed by Fritz Bauer [69] at the seminal conference on the subject still serves as a basis for discussion:
Software engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.

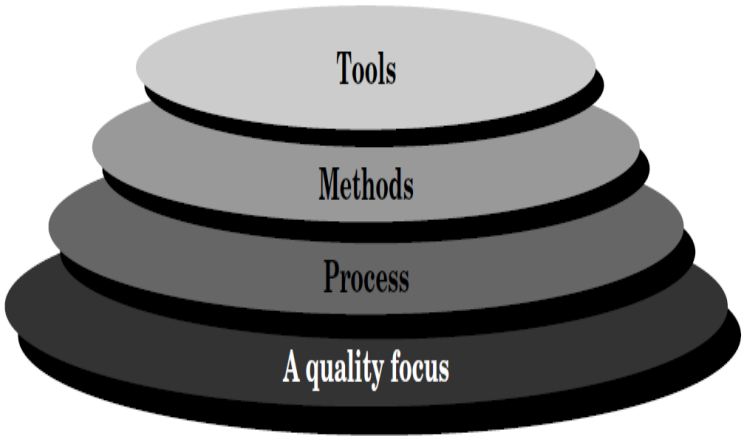
Another definition of software engineering may be:
Software engineering is the application of science and mathematics, by which the computer hardware is made useful to the user via software (computer programs, procedures, and associated documentation).

- The main objective of software engineering is to produce good quality software with minimum cost and within the limited allowed time period.
- The discipline of software engineering provides methodologies, which make software development process closer to the scientific method and away from being an art.
- That is, these methodologies are repeatable.
- It means that if different people with the same requirements apply the methodology, similar software will be produced.
- The phrase useful to user emphasizes the needs of the user and the software's interface with the user.
- It means that user needs should be given due importance in the development of software, and the final software product should give importance to the user interface.

2. Software Engineering as a Layered Technology (Process, method, and tools)

Software engineering is a layered technology.

- Any engineering approach (including software engineering) must rest on an organizational commitment to quality. Total quality management and similar philosophies foster a continuous process improvement culture, and this culture ultimately leads to the development of increasingly more mature approaches to software engineering.
- The bedrock that supports software engineering is a **quality focus**.



Layered approach

Process Layer:

- The foundation for software engineering is the ***process*** layer.
- Software engineering process is the glue that holds the technology layers together and enables rational and timely development of computer software.
- Process defines a framework for a set of ***key process areas*** (KPAs) that must be established for effective delivery of software engineering technology.
- The key process areas form the basis for management control of software projects and establish the context in which technical methods are applied, work products (models, documents, data, reports, forms, etc.) are produced, milestones are established, quality is ensured, and change is properly managed.

Methods Layer:

- Software engineering ***methods*** provide the technical how-to's for building software.
- Methods encompass a broad array of tasks that include requirements analysis, design, program construction, testing, and support.
- Software engineering methods rely on a set of basic principles that govern each area of the technology and include modeling activities and other descriptive techniques.

Tools Layer:

- Software engineering ***tools*** provide automated or semi-automated support for the process and the methods.
- When tools are integrated so that information created by one tool can be used by another, a system for the support of software development, called *computer-aided software engineering*, is established.
- CASE combines software, hardware, and a software engineering database (a repository containing important information about analysis, design, program construction, and testing) to create a software engineering environment analogous to CAD/CAE (computer-aided design/engineering) for hardware.