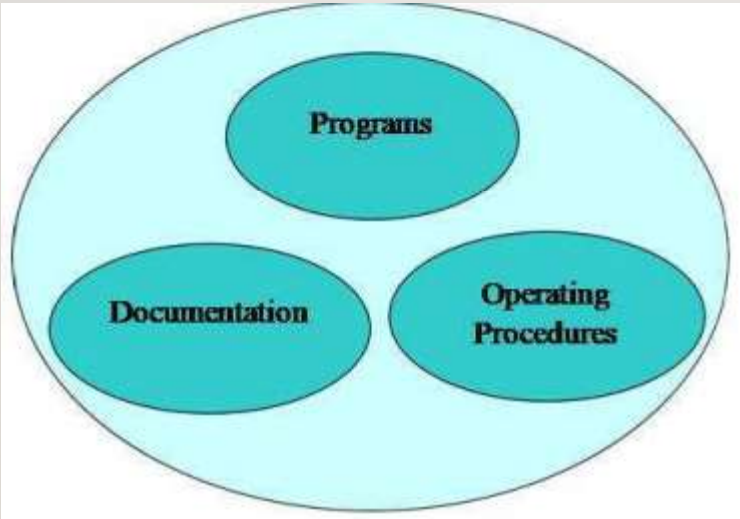
# Software

- **Software** is a set of computer programs and associated documentation and data.
- Software is defined as a collection of computer programs, procedures, rules, and data.
- Software is a set of programs, which is designed to perform a well-defined function.
- A program is a sequence of instructions written to solve a particular problem.

# Components of Software

- **Software = Program + Documentation + Operation Procedures**



# Contd…

**There are two types of software –**

- System Software
- Application Software

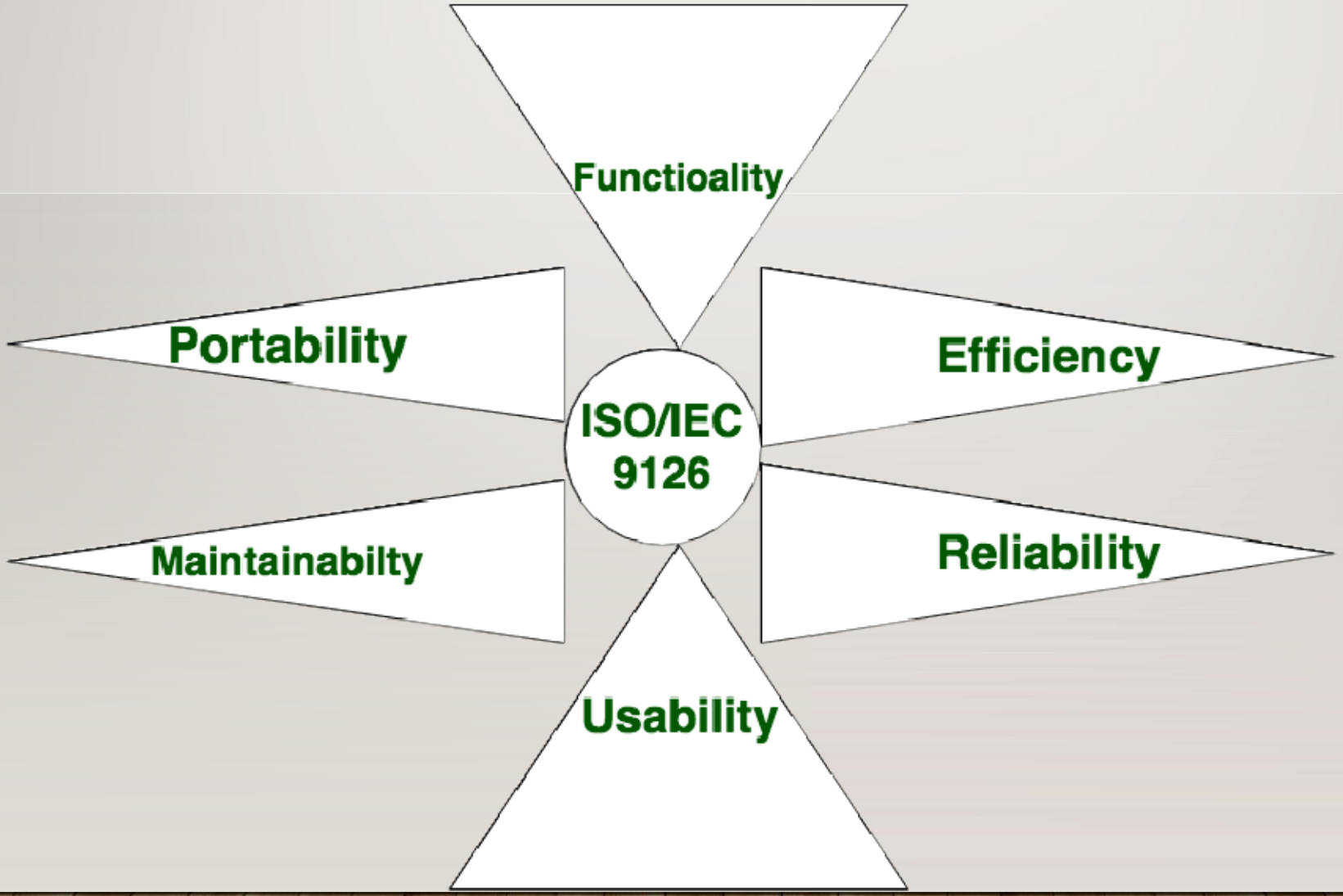
# System Software

- The system software is a collection of programs designed to operate, control, and extend the processing capabilities of the computer itself.
- System software is generally prepared by the computer manufacturers.
- These software products comprise of programs written in low-level languages, which interact with the hardware at a very basic level.
- System software serves as the interface between the hardware and the end users.
- Some examples of system software are Operating System, Compilers, Interpreter, Assemblers, etc

# Application Software

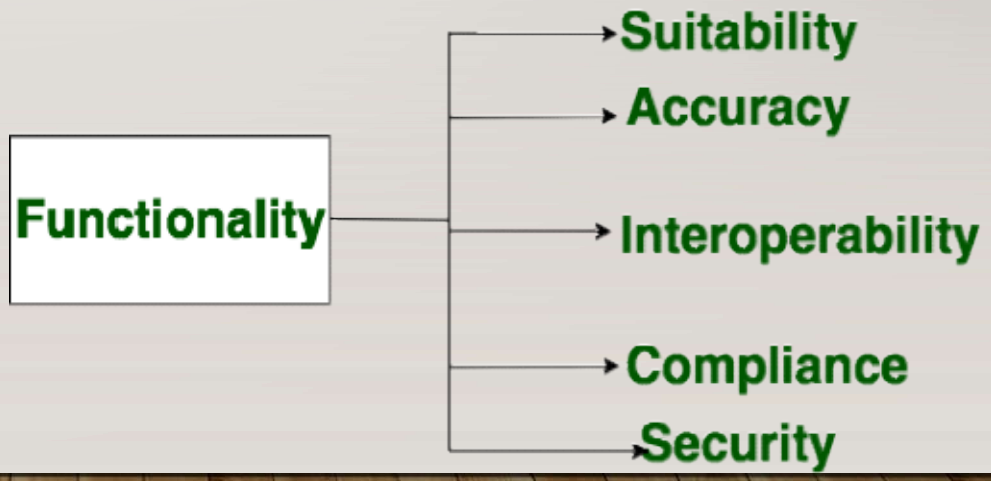
- Application software products are designed to satisfy a particular need of a particular environment.
- All software applications prepared in the computer lab can come under the category of Application software.
- Application software may consist of a single program, such as Microsoft's notepad for writing and editing a simple text.
- It may also consist of a collection of programs, often called a software package, which work together to accomplish a task, such as a spreadsheet package

# aracteristics of Software are classified into six major component



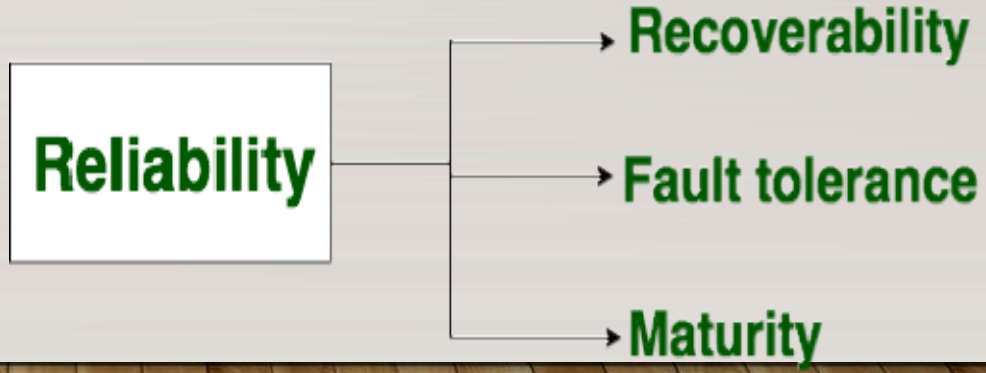
# Functionality

- It refers to the suitability, accuracy, interoperability, compliance, security of software which is measured as degree of performance of the software against its intended purpose.
- Functionality refers to the set of features and capabilities that a software program or system provides to its users.
- **Required functions are:**



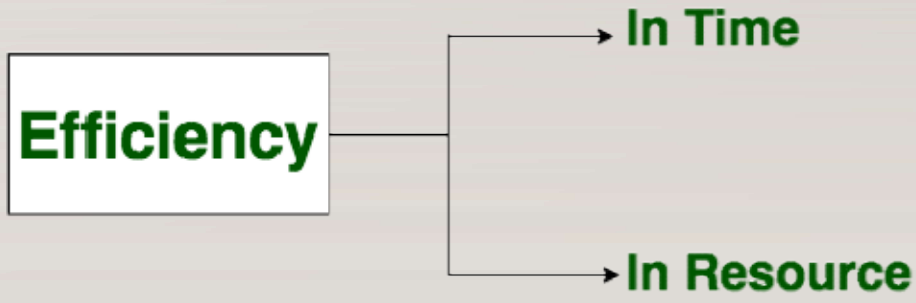
# Reliability

- Reliability is a characteristic of software that refers to its ability to perform its intended functions correctly and consistently over time.
- Refers to the recoverability, fault tolerance, maturity of software, which is basically a capability of the software that provide required functionality under the given situations.
- it helps ensure that the software will work correctly and not fail unexpectedly.



# Efficiency

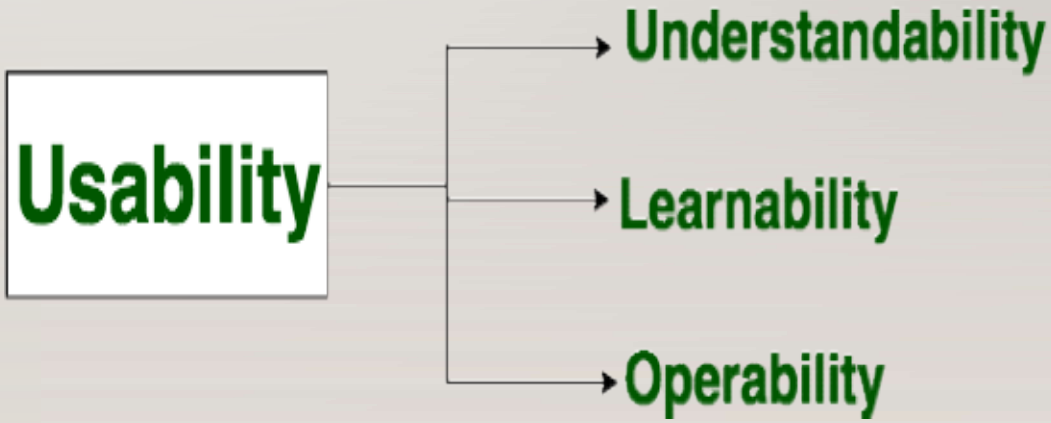
- It refers to the ability of the software to use system resources in the most effective and efficient manner.
- The software should make effective use of storage space and executive command as per desired timing requirements.
- High efficiency means that a software program can perform its intended functions quickly and with minimal use of resources.



# Usability

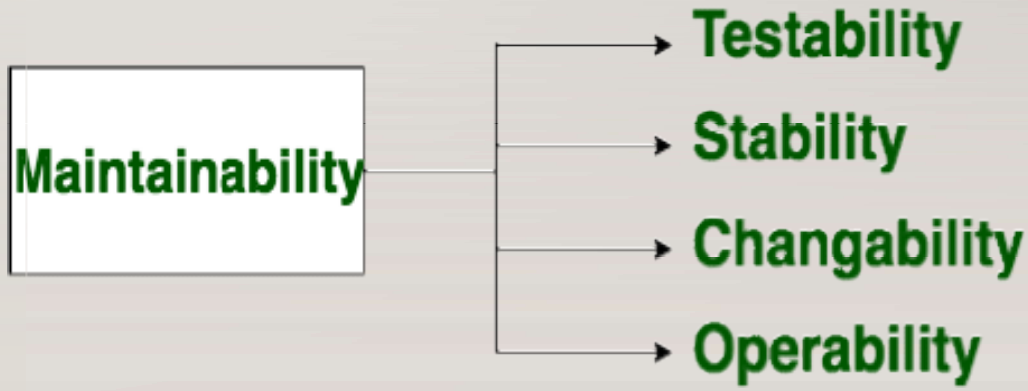
- It refers to the extent to which the software can be used with ease.
- The amount of effort or time required to learn how to use the software.

**Required functions are**



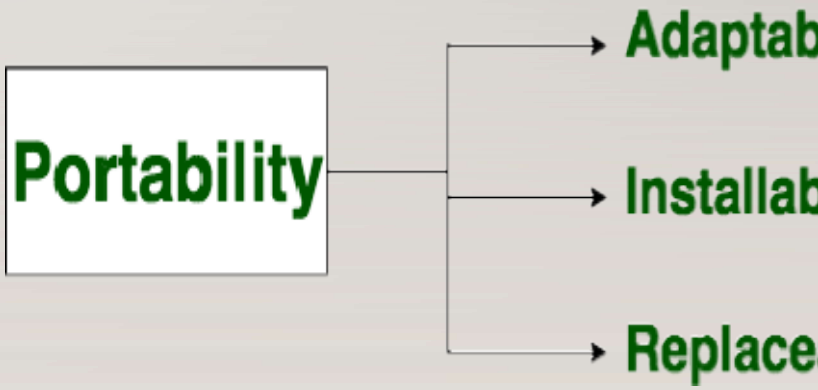
# Maintainability

- It is the ease with which the modifications can be made in a software to extend or enhance its functionality, improve its performance, or resolve bugs.



# Portability

- A set of attributes that bears on the ability of software to be transferred from one environment to another, without or minimum changes.
- In simple terms, software must be made in way that it should be platform independent.



# Characteristics Of Good Software

- Operational
- Transitional
- Maintenance

# Operational

- In operational categories, the factors that decide the software performance in operations. It can be measured on:
  - Budget
  - Usability
  - Efficiency
  - Correctness
  - Functionality
  - Dependability
  - Security
  - Safety

# Transitional

- When the software is moved from one platform to another, the factors deciding the software quality
  - Portability
  - Interoperability
  - Reusability
  - Adaptability

# Maintenance

- This aspect talks about how well software has the capabilities to adapt itself in the quickly changing environment:
  - Flexibility
  - Maintainability
  - Modularity
  - Scalability

# Software Engineering

- Software engineering is a systematic engineering approach to software development.
- Software engineering is the process of analyzing user requirements and then designing, developing, testing, and maintaining software which will satisfy those requirements..
- **Software engineering** is an engineering branch associated with development of software product using well-defined scientific principles, methods and procedures.
- The outcome of software engineering is an efficient and reliable software product.
- Important reasons for using software engineering are: 1) Large software, 2) Scalability 3) Adaptability 4) Cost and 5) Dynamic Nature

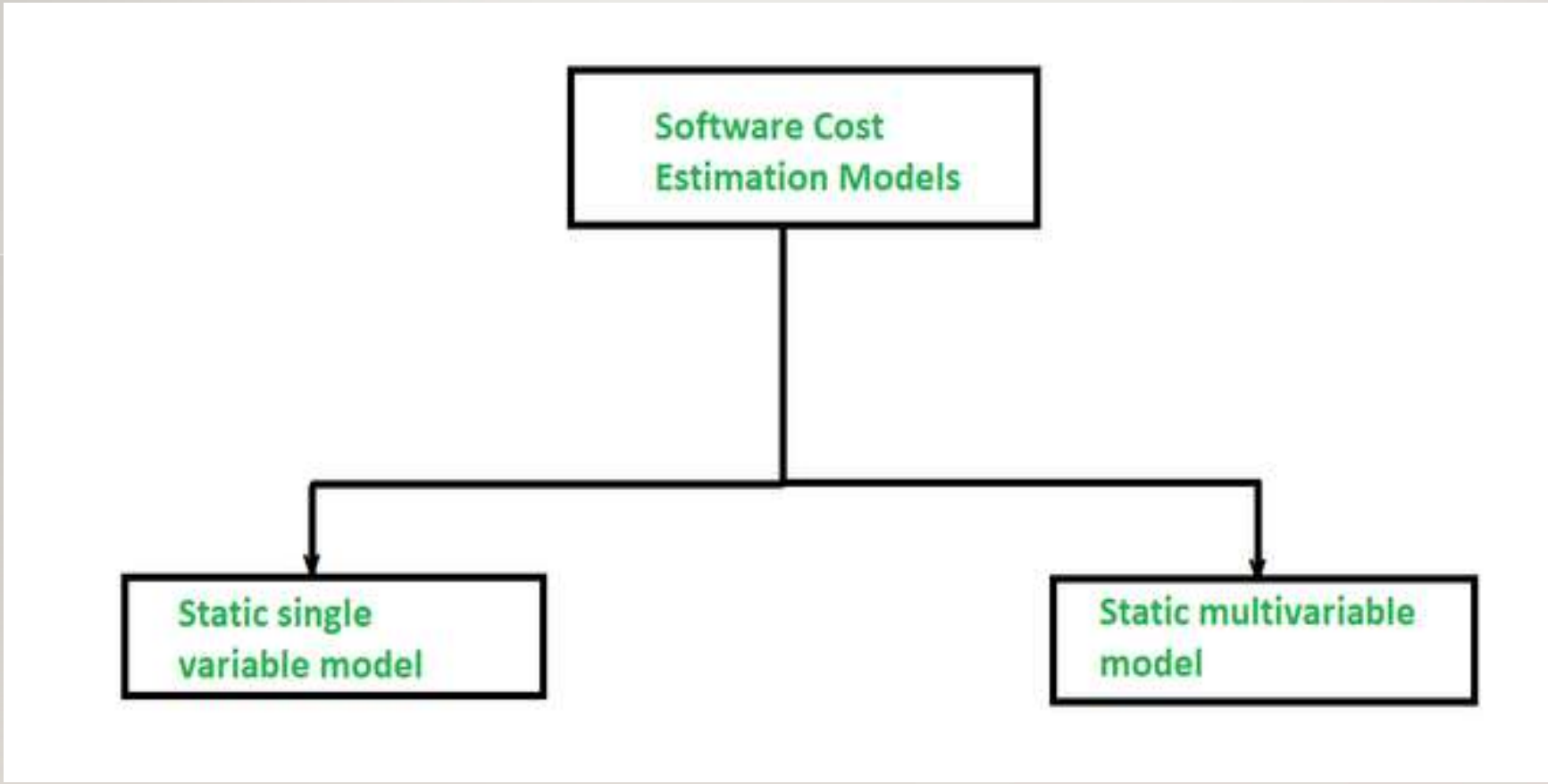
# Software Engineering Cost

- In starting a new **software project**, it is important to know how much it will cost to develop and how much development time will it take. These estimates are needed before development is initiated, but how is this done?
- Several estimation procedures have been developed and are having the following attributes in common.
- Project scope must be established in advanced.
- Software metrics are used as a support from which evaluation is made.
- The project is broken into small PCs which are estimated individually. To achieve true cost & schedule estimate, several option arise.

# Contd…

- Delay estimation
- Used symbol decomposition techniques to generate project cost and schedule estimates.
- Acquire one or more automated estimation tools.

## Below are the two models in estimating the cost of a software project:



# 1. Static Single Variable Model

- The Methods using this model utilizes an equation to get the desired values such as cost, time, and effort, etc. And these all depend on the same variable used as a predictor like, size. Below is the example of the most common equation:

$$C = aL^b$$

- Where C is cost, L is size and a, b are constants.

# 2. Static Multivariable Model

- These models are also known as **multivariable models.**
- This model is often based on the first equation and actually depends on several variables representing different aspects of the software development environment.

Equations are:

$$E = 5.2L^{0.91}$$

$$D = 4.1L^{0.36}$$

Where E is in Person-months, D is duration which is months.

# Key Challenges Facing Software Engineering

## 1. The legacy challenge

- The majority of software systems that are in use today were developed many years yet they perform critical business functions.
- The legacy challenge is the challenge of maintaining and updating this software in such a way that excessive costs are avoided and essential business services continue to be delivered.

## 2. Heterogeneity challenge

- Increasingly, systems are required to operate as distributed systems across networks that include different types of computers and with different kinds of support systems.
- The heterogeneity challenge is the challenge of developing techniques to build dependable software which is flexible enough to cope with this heterogeneity.

# CONTD…

**3. The delivery challenge**

- Many traditional software engineering techniques are time-consuming.
- The time they take is required to achieve software quality.
- However, businesses today must be responsive and change very rapidly.
- Their supporting software must change equally rapidly.
- The delivery challenge is the challenge of shortening delivery times for large and complex systems without compromising system quality.

# Contd…

**4. Trust challenge**

- As software is intertwined will all aspects of our lives, it is essential that we can trust that software, so, the trust challenge is to develop techniques that demonstrate that software can be trusted by its users.

**5. Risk challenge**

- In safety-critical areas such as space, aviation, nuclear power plants, etc. the cost of software failure can be massive because lives are at risk.
- Dealing with the increased complexity of software needed for new applications.

# 1. System Engineer

- System engineers are involed in system specification, architectural design, integration and deployment.
- A System Engineer is a person who deals with the overall management of engineering projects during their life cycle (focusing more on physical aspects).
- They follows an interdisciplinary approach governing the total technical and managerial effort required to transform requirements into solutions.
- They are generally focused with all aspects of computer based system development not only this but also hardware, software and process engineering etc. are included.

# Professional Practice

- Professional practice is a term which usually refers to the conduct as well as the work of someone from a specific profession.
- It refers to professional responsibility. Professional practice is the way an individual behaves in the workplace.
- A software engineer displays professionalism notably through adherence to codes of ethics and professional conduct and to standards and practices that are established by the engineer's professional community.
- The professional community is often represented by one or more professional societies; those societies publish codes of ethics and professional conduct as well as criteria for admittance to the community.

# Contd…

- Those criteria form the basis for accreditation and licensing activities and may be used as a measure to determine engineering competence or negligence.

# Assignment Questions

- Define software and explain its characteristics and aslo mention the various types of software product
- Explain some of the software applications you have noticed
- Explain attributes of good software in detail
- What are the key challenges facing software engineering