

## 1. Definition of Software Engineering

**Software Engineering** Computer Science की वह branch है जिसमें software को develop करने, design करने, test करने और maintain करने के लिए **systematic, disciplined** और **quantifiable approach** का उपयोग किया जाता है।

### ◆ Main Points:

1. यह software बनाने की engineering approach है।
- 2.
3. इसमें planning, analysis, design, coding, testing और maintenance शामिल होता है।
4. इसका उद्देश्य high quality, reliable और cost-effective software बनाना है।
5. User की आवश्यकता को पूरा करना software engineering का मुख्य कार्य है।

## 2. Need of Software Engineering

Software Engineering की आवश्यकता क्यों पड़ी?

1. **Increasing Complexity:** Software applications बढ़े और complex होते जा रहे हैं।
2. **Quality Requirement:** Software error-free और reliable होना चाहिए।
3. **Cost Control:** Development में time और budget दोनों को manage करना ज़रूरी है।
4. **Team Coordination:** Large projects में कई developers मिलकर काम करते हैं, systematic approach ज़रूरी है।
5. **User Satisfaction:** User की बदलती needs को fulfill करना।

## 3. Software vs Program

### Program

Small code written to perform a task.  
केवल single function कर सकता है।  
इसे अकेला programmer लिख सकता है।  
Maintenance की ज़रूरत नहीं।

### Software

Collection of programs + documentation + user manual.  
Multiple functions और complex tasks कर सकता है।  
इसे large team develop करती है।  
Regular maintenance ज़रूरी।

## 4. Characteristics of Software

1. **Intangible (अमूर्त):** Software को छू नहीं सकते, केवल use कर सकते हैं।
2. **Developed not Manufactured:** Hardware की तरह factory में नहीं बनता, develop किया जाता है।
3. **Customizable:** User की need के अनुसार बदला जा सकता है।
4. **High Development Cost:** इसे बनाने में ज़्यादा human effort और cost लगता है।

5. **Reusability:** एक बार बनने के बाद इसे कई जगह इस्तेमाल किया जा सकता है।
6. **Evolves over Time:** Requirement बदलने पर इसे update करना पड़ता है।

## 5. Goals of Software Engineering

1. **High Quality Software** बनाना।
2. **Cost-effective development** करना।
3. **Time bound delivery** करना।
4. **User needs** को fulfill करना।
5. **Easy Maintenance** और future enhancement possible करना।

## 6. Importance of Software Engineering

1. Software की demand हर sector में बढ़ रही है।
2. Systematic process से errors और cost दोनों कम होते हैं।
3. Reliable और secure software develop होता है।
4. Large teams और big projects को manage करना आसान होता है।
5. User satisfaction और long-term support provide करता है।

# System Concepts (Software Engineering – Unit 1)

## 1. Definition of System

**System** का अर्थ है – अलग-अलग components (parts) का एक ऐसा organized group, जो एक common goal (समान उद्देश्य) को achieve करने के लिए आपस में interact करता है।

👉 Example:

- **Computer System** = Hardware + Software + Users
- **College System** = Students + Teachers + Administration + Rules

## 2. Characteristics of a System

1. **Organization (संगठन):** System के सभी components एक specific order में arranged होते हैं।
2. **Interaction (परस्पर क्रिया):** Components आपस में interact करते हैं।
3. **Interdependence (परस्पर निर्भरता):** हर component दूसरे पर depend करता है।
4. **Integration (एकीकरण):** सभी parts मिलकर एक complete system बनाते हैं।
5. **Goal Oriented (उद्देश्यपूर्ण):** System का एक निश्चित उद्देश्य (goal) होता है।

## 3. Elements of a System

1. **Input:** Data या resources जो system को provide किए जाते हैं।
  - Example: Student data college system में input है।
2. **Process:** Input को process करके output generate करना।

- Example: Teaching process → Knowledge delivery |
  - 3. **Output:** Process का result |
    - Example: Trained students |
  - 4. **Feedback:** System को improve करने के लिए information वापस देना।
    - Example: Student feedback for teachers |
  - 5. **Control:** System को सही दिशा में चलाने के लिए rules और monitoring |
- 

## 4. Types of Systems

1. **Physical System:** Hardware components जैसे Computer, Printer |
  2. **Abstract System:** Theoretical models या software designs |
  3. **Open System:** Environment के साथ interact करने वाला system |
    - Example: College system (students, society से जुड़ा हुआ) |
  4. **Closed System:** Environment से isolated system (कम interaction) |
    - Example: Nuclear plant का core system |
  5. **Man-Made System:** Human द्वारा बनाया गया system |
    - Example: Railway Reservation System |
  6. **Natural System:** Nature द्वारा बने systems |
    - Example: Solar System |
- 

## 5. System Concepts in Software Engineering

Software Engineering में **system concept** का उपयोग software को design और analyze करने में होता है।

- Software system = Input (user data) → Process (program execution) → Output (result) → Feedback (error correction) |
  - इससे software structured और efficient बनता है।
- 

## 6. Importance of System Concepts

1. Complex projects को समझने में मदद मिलती है।
2. System के हर component की role clear होती है।
3. Requirement analysis आसान हो जाता है।
4. Software को modular design किया जा सकता है।
5. System approach से errors कम होते हैं और efficiency बढ़ती है।

## Types of Systems (Open, Closed, Static and Dynamic Systems)

### 1. Open System

#### Definition:

Open system वह होता है जो अपने environment के साथ लगातार data, information और resources exchange करता है।

### **Characteristics:**

- Environment से interact करता है।
- Feedback लेकर adapt करता है।
- Dynamic और flexible होता है।
- Continuous input और output का flow होता है।

### **Examples:**

- College system (students, teachers, society interaction)
- Banking system (customers, govt. rules से जुड़ा)
- Hospital management system

## **2. Closed System**

### **Definition:**

Closed system ऐसा होता है जो **अपने environment से isolated** रहता है और **external interaction** बहुत कम या नहीं करता।

### **Characteristics:**

- Environment से कोई data exchange नहीं करता।
- Self-contained और rigid होता है।
- Predictable और controlled behavior।
- Feedback external environment से नहीं लेता।

### **Examples:**

- Nuclear reactor का core
- Mathematics models
- कुछ security-based systems

## **3. Static System**

### **Definition:**

Static system वह होता है जो **time के साथ change** नहीं होता और हमेशा **fixed** रहता है।

### **Characteristics:**

- Behavior time के साथ नहीं बदलता।
- Stable और constant रहता है।
- Easy to analyze और simple structure होता है।

### **Examples:**

- Railway time table (fixed schedule)
- Mathematical formulas
- Printed books

## **4. Dynamic System**

**Definition:**

Dynamic system वह होता है जो **time** के साथ बदलता रहता है और **continuously evolve** करता है।

**Characteristics:**

- Time dependent behavior।
- Feedback और environment से influence होता है।
- Unpredictable changes possible।
- Complex और adaptive होता है।

**Examples:**

- Stock market system
- Weather forecasting system
- Human body (biological system)

## **Importance of these Types in Software Engineering**

1. Software system को categorize करने में मदद मिलती है।
2. सही design और testing approach select की जा सकती है।
3. Dynamic और Open systems software development में ज्यादा practical होते हैं।
4. Closed और Static systems predictable होते हैं, इसलिए easily analyze होते हैं।

## **Exam KeyWords (Quick Revision)**

- **Open System** – Interacts with environment, feedback oriented
- **Closed System** – Isolated, no external interaction
- **Static System** – Fixed, unchanging with time
- **Dynamic System** – Changing, adaptive, time dependent

## **Basics of Software Engineering**

### **1. Definition of Software Engineering**

Software Engineering Computer Science की एक branch है जिसमें **systematic, disciplined** और **quantifiable approach** से software को design, develop, test और maintain किया जाता है।

 **Simple Words** में:

Software Engineering = Software Development + Maintenance + Quality Assurance in systematic way.

### **2. Why Software Engineering is Needed? (Necessity)**

1. **Complexity:** आज के software बहुत बड़े और complex हैं।
2. **Cost Control:** Proper planning से development cost कम होती है।

3. **Time Management:** Deadline पर software deliver करने में मदद करता है।
  4. **Quality:** Error-free और reliable software बनाने के लिए।
  5. **Team Work:** बड़े project कई लोगों द्वारा develop होते हैं, coordination जरूरी है।
  6. **User Satisfaction:** User की बदलती needs fulfill होती हैं।
- 

### 3. Main Activities in Software Engineering

1. **Requirement Analysis** – User से जरूरतें collect करना।
  2. **System Design** – Architecture और models बनाना।
  3. **Implementation (Coding)** – Program लिखना।
  4. **Testing** – Errors और bugs remove करना।
  5. **Deployment** – User के system में software install करना।
  6. **Maintenance** – Updates और improvements करना।
- 

### 4. Characteristics of Good Software

1. **Correctness:** सही result देना चाहिए।
  2. **Reliability:** हर situation में काम करना चाहिए।
  3. **Efficiency:** कम resources (CPU, Memory) use करना चाहिए।
  4. **Usability:** Easy to use हो।
  5. **Maintainability:** Future में changes आसानी से हो सकें।
  6. **Portability:** Different platforms पर चल सके।
  7. **Security:** Unauthorized access से protected हो।
- 

### 5. Goals of Software Engineering

1. High-quality software बनाना।
  2. Low cost और कम समय में delivery करना।
  3. Maintenance आसान बनाना।
  4. Customer satisfaction achieve करना।
  5. Software को reliable और secure बनाना।
- 

### 6. Advantages of Software Engineering

- Systematic और organized process मिलता है।
  - Large और complex projects handle किए जा सकते हैं।
  - Errors और bugs कम होते हैं।
  - Software की performance और security improve होती है।
  - Maintenance और future upgrades आसान होते हैं।
- 

## 🌟 Exam KeyWords (Quick Revision)

- **Definition** – Systematic, disciplined approach
- **Need** – Complexity, Cost, Time, Quality, Team Work
- **Activities** – Requirement, Design, Coding, Testing, Deployment, Maintenance
- **Good Software** – Correct, Reliable, Efficient, Usable, Maintainable, Portable, Secure

- **Goals** – High quality, Low cost, On-time delivery, Easy maintenance, User satisfaction



# Need for Software Engineering

## 1. Definition of Software Engineering

Software Engineering Computer Science की वह branch है जिसमें software को develop, design, test और maintain करने के लिए **systematic, disciplined** और **quantifiable approach** का उपयोग किया जाता है।

👉 Simply:

**Software Engineering = Systematic way of software development + Maintenance + Quality Assurance.**

## 2. Need for Software Engineering

Software Engineering की आवश्यकता क्यों?

1. **Increasing Complexity:** आज के software बहुत बड़े और complex हैं।
2. **Quality Demand:** Reliable, secure और error-free software चाहिए।
3. **Cost Management:** बिना planning software development बहुत महंगा हो सकता है।
4. **Time Bound Delivery:** Software को deadline पर deliver करना जरूरी है।
5. **User Satisfaction:** Changing user needs fulfill करनी होती है।
6. **Team Coordination:** बड़े projects में कई developers मिलकर काम करते हैं।
7. **Maintenance:** Software को future में update करना जरूरी है।

## 3. Programs vs Software Products

Aspect	Program (छोटा code)	Software Product (बड़ा package)
Size	छोटा और simple code	बड़ा और complex system
Purpose	एक specific task के लिए	Multiple tasks और commercial use
Development	Individual programmer बना सकता है	Large team और proper process से बनता है
Documentation	Documentation जरूरी नहीं	Proper documentation और manuals जरूरी
Maintenance	Maintenance की जरूरत नहीं	Regular maintenance जरूरी

## 4. Software Characteristics

**Software की विशेषताएँ (Important Properties):**

1. **Intangible (अमूर्त):** Software को देखा या छुआ नहीं जा सकता, केवल use किया जा सकता है।
2. **Developed, not Manufactured:** Software को factory में नहीं बनाया जाता, इसे develop किया जाता है।
3. **Customizable:** User की need के अनुसार बदला जा सकता है।
4. **High Development Cost:** Development में human effort और cost ज्यादा लगती है।
5. **Easy Reproduction:** एक बार बनने के बाद इसे कई बार copy किया जा सकता है।
6. **Evolves with Time:** User की changing requirements के साथ software भी change होता है।
7. **Quality Dependent:** Software की quality उसके design और testing पर depend करती है।
8. **Documentation Oriented:** Proper documentation software का हिस्सा होता है।

## ★ Exam KeyWords (Quick Revision)

- **Definition** – Systematic, disciplined approach
- **Need** – Complexity, Cost, Time, Quality, Team work, Maintenance
- **Program vs Product** – Size, Purpose, Development, Documentation, Maintenance
- **Characteristics** – Intangible, Developed not Manufactured, Customizable, High Cost, Reusable, Evolves, Quality dependent