# UNIT 1

# Introduction to System and Software Development

# System

- A system is a collection of software components, hardware, and users that work together to perform specific tasks or functions.
- A system typically includes various interconnected parts that cooperate to achieve a common goal, such as managing data, controlling processes, or providing services to users.
- Systems are all around us, in our everyday life, there are many systems like educational system, computer system, solar system, transportation system, communication system, accounting system, production system and so on.

# Characteristics of the System

There are FIVE characteristics of a system are:

1. Basic components
2. Interaction and structure
3. Goal
4. Behavior
5. Life cycle

# Information systems

- **What is Information System ?**

"An information system is a set of interrelated components that works together to collect, process, store and breakdown the information to support decision making."

- **Informal name of Information system is Software or Software System.**

# Information systems categories

An information system can be categorized based upon activity into strategic planning system, tactical information system and operational information system.

1. **Strategic planning** in software development involves setting objectives, focusing on the architecture and design, deciding on the development timeline and concentrating resources and efforts (setting clear goals and deciding how to achieve them).

# Information systems categories

**2. Tactical information** Systems are a type of decision support system that not only help in making decisions but can also mimic the expertise of the decision-maker. They are useful for more specific, short-term decisions within an organization. These systems can analyze data, offer recommendations, and even replicate the decision-making process of an expert in that field.

# Information systems categories

**3. Operational information** Systems are used to handle the day-to-day activities of an organization. These systems are designed to process routine transactions efficiently, ensuring that everyday tasks like processing orders, managing inventory, or handling customer service requests are completed smoothly. Their main goal is to keep the organization running efficiently on a daily basis.

# Need of information system development

- Necessary for businesses to grow
- Better data storage and access
- Better decision making
- Data Management and Security
- Customer and Client Management
- Scalability and Growth
- Cost Management

# What is Software?

Software is:
(1) instructions (computer programs) that provide desired features, function, and performance,
(2) data structures that enable the programs to manipulate data or information and
(3) documentation that describes the operation and use of the programs.

# Basic Components of Software

Four components of software:

- Computer programs (the “code”)
- Procedures
- Documentation
- Data necessary for operating the software system.
- User Interface (UI)
- Database
- APIs (Application Programming Interfaces)

# Software Characteristics

1. **Functionality**
   - Functionality refers to the set of features and capabilities that the software provides to meet the user's needs. It defines what the software can do and how well it performs those functions.
   - Key Aspects: Completeness, correctness, and appropriateness of features.

2. **Reliability**
   - Reliability measures how consistently the software performs its functions without failure. It indicates the software's ability to operate under specified conditions for a given period.
   - Key Aspects: Stability, fault tolerance, and error handling.

# Software Characteristics

### 3. Usability

- Usability refers to how easy the software is for users to understand and operate. It includes aspects such as user interface design, ease of learning, and overall user experience.
- **Key Aspects**: User-friendly interfaces, accessibility, and support for user needs.

### 4. Efficiency

- Efficiency relates to the software's ability to perform its functions with optimal use of resources, such as memory, processing power, and network bandwidth.
- **Key Aspects**: Performance, responsiveness, and resource utilization.

# Software Characteristics

**5. Maintainability**

- Maintainability is the ease with which software can be modified to correct defects, improve performance, or adapt to changing requirements. It includes aspects like code readability and modularity.
- **Key Aspects**: documentation,hard coded, and code complexity.

**6. Scalability**

- Scalability refers to the software's ability to handle increasing amounts of work or to be extended with additional features or resources without compromising performance.
- **Key Aspects**: Performance under load, system architecture, and flexibility to accommodate growth.

# Software Application Domains

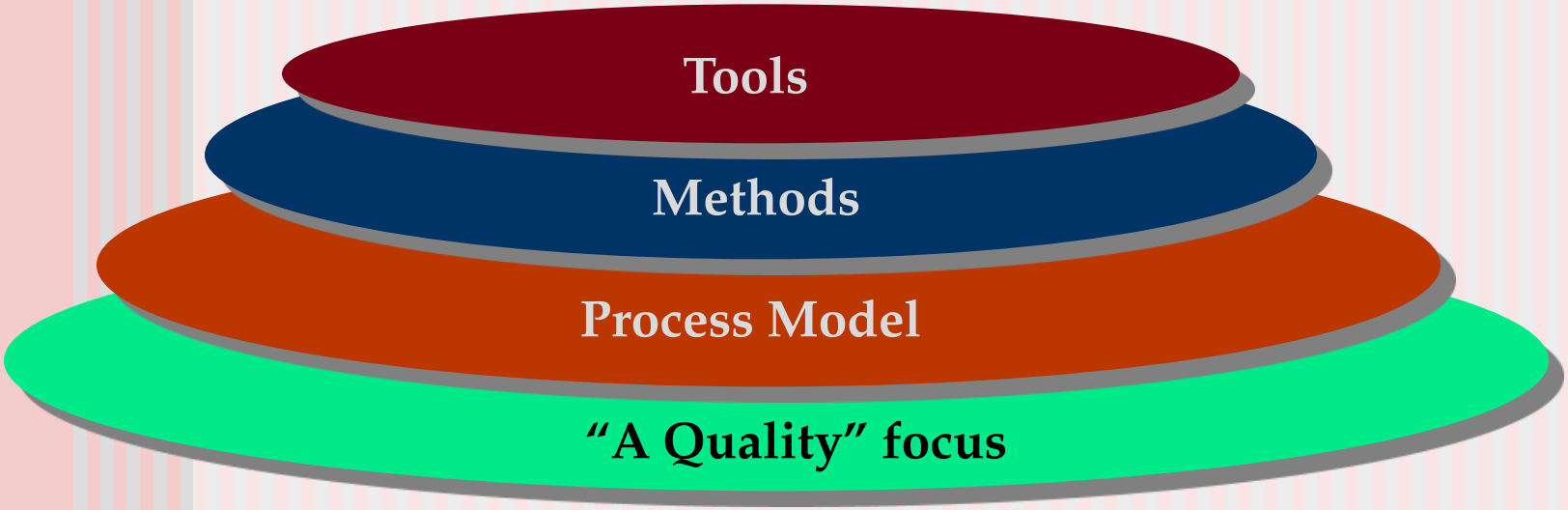
- System Software
- Application Software
- Engineering/Scientific Software
- Embedded Software
- Product-line Software
- WebApps (Web applications)
- AI software

# Software Engineering

**Software engineering** is a systematic, disciplined, and quantifiable approach to design, development, operation, and maintenance of software. It applies engineering principles to software development to ensure that software is reliable, efficient, maintainable, and meets the needs of its users.

# Software Engineering

- Software engineering is a **layered technology**.
- It has 4 layers:
- A quality focus, Process, Method, Tools



# Software Engineering Layers

**1. A Quality Focus:** The foundation of software engineering is a commitment to quality. This layer emphasizes the importance of ensuring that every aspect of the software development process, from requirements to deployment, is focus toward producing high-quality software.

**2. Process:** The process layer provides the framework for developing software in a systematic and disciplined manner. It defines the sequence of activities, tasks, and milestones required to build and deliver software.

# Software Engineering Layers

**3. Method:** The method layer provides the technical “how-to” for building software. It encompasses the approaches, techniques, and procedures that developers use to build software efficiently and effectively.

**4. Tools:** The tools layer includes the software and hardware tools that support the process and method layers. These tools automate or assist in various software engineering tasks, making the development process more efficient and less error software.

# Generic view of software engineering

The process of a software development has three Generic views which are:

1. **Definition Phase :** The experts get the knowledge about "What". This phase defines all the expectations depending on the standard of the software Engineering. It contains three steps.

- Requirement Analysis
- Analysis for system
- Planning of project

# Generic view of software engineering

2. **Development phase** : Focus point of development phase is "How". Three special steps always taken in this phase which are;
   - Design of software
   - Coding
   - testing of software system
3. **Maintenance phase** - The main focus of maintenance phase is change which causes correction of errors and adaption of new idea according to the needs of software after change in customer mood.

# Software Process

- A **process** is a collection of activities, actions, and tasks that are performed when some work product is to be created.
- An **activity** strives to achieve a broad objective (e.g., communication with stakeholders / partners) and is applied regardless of the application domain, size of the project, complexity of the effort.
- An action (e.g., architectural design) encompasses a set of tasks that produce a major work product (e.g., an architectural design model).
- A task focuses on a small, but well-defined objective (e.g., conducting a unit test) that produces a tangible outcome.