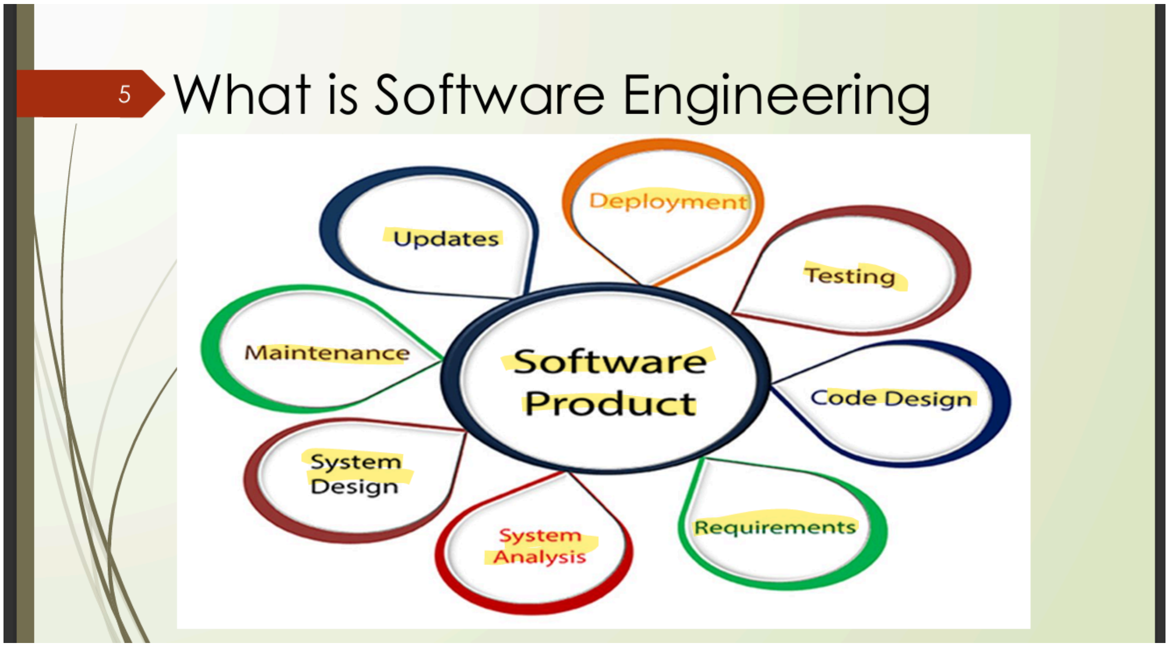
Chapter one



5 What is Software Engineering

Software Engineering :

- **Software**:

  A collection of integrated programs, consisting of organized instructions and code written by developers in various programming languages. It also includes related documentation like requirements, design models, and user manuals.

- **Engineering**:

  The application of scientific and practical knowledge to invent, design, build, maintain, and improve systems and processes.

- **Software Engineering**:

  A branch of engineering focused on the systematic development and evolution of software products, using well-defined principles, techniques, and procedures.

- **Result** :

  The goal of software engineering is to produce effective and reliable software products.

Why is Software Engineering Required?

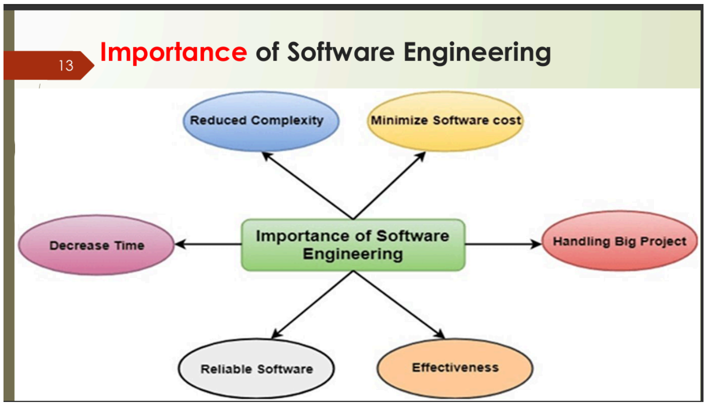
Software Engineering is required due to the following reasons:

- **Manage Large Software**
- **Scalability**
- **Cost Management**
- **To Manage the Dynamic Nature of Software**
- **Better Quality Management**
- **Huge Programming**:
  This means that as software grows, a structured approach is needed to handle its complexity.
- **Adaptability**:
  In other words, engineering principles make it easier to expand software rather than starting over each time.
- **Cost**:
  By following a systematic process, we can keep development affordable and avoid wasteful expenses.
- **Dynamic Nature**:
  This adaptability allows software to stay up-to-date and useful for users as needs change.
- **Quality Management**:
  When we follow structured processes, the end product is more reliable and meets higher standards.

Characteristics of a Good Software Engineer

- Exposure to systematic methods
- Good technical knowledge
- Good programming skills
- Good communication skills
- High motivation
- Sound knowledge of fundamentals of computer science
- Ability to work in a team
- Discipline

Importance of Software Engineering



13

Importance of Software Engineering

Reduced Complexity

Minimize Software cost

Decrease Time

Importance of Software Engineering

Handling Big Project

Reliable Software

Effectiveness

1. **Reduces Complexity**
   - Breaks down large problems into smaller, manageable tasks, which are solved independently and then combined.
2. **Minimizes Software Cost**
   - Provides a structured approach to estimate budgets accurately and allocate resources efficiently.
3. **Decreases Time**
   - Implements processes to ensure timely project completion, reducing scheduling conflicts.
4. **Handles Big Projects**
   - Uses planning, management, and testing to handle long-term, resource-intensive projects smoothly.
5. **Ensures Reliable Software**
   - Uses models to measure and improve software reliability, reducing the likelihood of failures.
6. **Increases Effectiveness**
   - Ensures software meets goals, delivers high quality, and satisfies users by using resources wisely.

Software Engineering vs. Computer Science :

- **Computer Science**:

  Focuses on theoretical concepts and fundamental principles, dealing with both abstract and practical knowledge.

- **Software Engineering**:

  Applies engineering methods to design, create, and maintain software systems for various uses.

Software Engineering vs. System Engineering :

- **System Engineering**:

  Focuses on the entire development of computer-based systems, including hardware, software, and processes. It covers system specification, design, integration, and deployment.

- **Software Engineering**:
- A subset of system engineering, dedicated to developing the software components within a system, such as applications, databases, and controls.
- **Systems engineering is older than Software Engineering**

Employment in Software Engineering “where to work ”

- **Specialist IT Firms:**

  Includes IT consultancies, software developers, internet providers, and organizations in sectors like retail, law, education, public services, and more.

- **Manufacturing Industry:**

  Includes sectors like automotive, telecommunications, navigation, and construction that rely on software.

- **Financial Services:**

  Involves global investment banks, financial institutions, security market specialists, and the pensions sector.

- **Public Utilities:**

Covers industries like energy and water supply, energy extraction, and transportation, which all depend on software systems.

Software Engineer Job Duties “what you need to work as SWE” :

- **Analyzing User Requirements**:

  Understand and evaluate what users need from the software.

- **Testing and Refining Code**:

  Ensure the code works properly, making necessary changes.

- **Researching and Designing Software**:

  Develop new software programs and solutions.

- **Developing Existing Programs**:

  Improve current programs by identifying areas for enhancement.

- **Integrating Software**:

  Make different software products and platforms work together.

- **Creating Technical Specifications**:

  Define the technical details and requirements of the software.

- **Writing Documentation**:

  Collaborate with technical authors to create operational manuals.

- **Maintaining Systems**:

  Monitor and fix software defects to ensure smooth operation.

- **Collaboration with Staff**:

  Work with project managers, designers, other developers, and professionals in sales and marketing.

- **Consulting Clients**:

  Advise clients on software maintenance, performance, and updates.

- **Investigating New Technologies**:

  Explore and evaluate new technologies for potential use.

Recent Facts About Software Engineer Job Opportunities “Read only”

- **Test and Quality Assurance Engineers**:

  The most gender-diverse group, with women making up about 30% of the workforce.

- **Mobile Engineers**:

  The second-largest specialty, with a younger workforce; 37% have less than 10 years of experience.

- **Front-End Engineers**:

  The largest talent pool, more than double that of the second-largest specialty.

- **Infrastructure and Cloud Computing Engineers**:

  77% have over 10 years of experience, highlighting the expertise in this field.

- **Embedded and Application Engineers**:

  Focus on IoT, with a smaller but highly experienced and less gender-diverse group.

- **Machine Learning and Data Science Engineers**:

  The smallest specialty but the most in-demand, reflecting current industry trends.

Chapter two

Software Process

- **Agenda**
    - Historical Aspects
    - Software Engineering
    - Software Product
    - Software Process Phases
    - Improving Software Process

Historical Aspects

- The term "software engineering" was proposed at NATO conferences in 1968 and 1969 to address the "software crisis."
- The software crisis referred to the challenges in developing large, complex systems during the 1960s, leading to:
    - Late deliveries
    - Over-budget projects
    - Residual faults in the software

Examples of Failures Due to Software

1. **CareFusion's Alaris Pump (2015)**: A software error delayed medication delivery, risking patient suffocation.
2. **Equifax Data Breach (2017)**: Personal data of 143 million consumers, including Social Security numbers, was stolen.
3. **Facebook Outage (2019)**: Users were unable to view or load images from Facebook's newsfeed.

Software Engineering

- Software engineering involves the systematic application of engineering principles, techniques, and procedures to create, maintain, and evolve software products.
- The goal is to produce reliable and effective software.
- Software engineers require a broad range of technical and managerial skills.

Software Product

- Software is a set of integrated items that form a configuration, including:
  - **Programs**: Performing the desired functions and tasks.
  - **Data Structures**: Enabling effective data manipulation.
  - **Documents**: Describing the software's operation and use.

Failure, Error, and Faults

- **Error**: A human action leading to an incorrect result (also called a mistake).
- **Fault**: A defect or bug that results from an error in software.
- **Failure**: A deviation from the software's expected behavior or service.

Relative Cost to Correct Defects

- The cost to fix defects increases as the project progresses.

Software Process

- The software process defines the steps taken to produce software.
- It involves predictable stages (a roadmap) to create high-quality software efficiently.
- **Software Engineering** also includes the methods and tools used throughout the process.
  - **Methods**: Provide guidelines for building software.
  - **Tools**: Offer support for automating or assisting the process.

Software Process Phases

1. **Requirements Analysis Phase** (4 steps)
2. **Specification Phase**
3. **Design Phase**
4. **Implementation Phase**
5. **Integration Phase** (in parallel with Phase 4)
6. **Maintenance Phase**
7. **Retirement**

- Testing and documentation occur throughout all phases.

Software Process Phases - Requirements Analysis Phase

1. **Requirements Gathering (Elicitation)**:
   The team meets with the client to understand and outline detailed requirements.
2. **Requirements Capturing**:
   - **Data Requirements**: Information to be stored.
   - **Functionality Requirements**: Tasks like “add customer” and “print invoice.”
   - **Quality Attributes**: Includes performance, security, availability, and usability.
3. **Validate Requirements (Rapid Prototyping)**:
   A prototype is built to show functionality, allowing the client to test and confirm if it meets their needs.
4. **Deliverable**:
   A **Requirements Document** is created and reviewed by the client, users, and development team