



# SOFTWARE DEVELOPMENT

## Software Development Life Cycle (SDLC)

### 1. Planning

- **Objective:** Define the scope and purpose of the project.
- **Activities:**
  - Identify stakeholders.
  - Conduct feasibility studies.
  - Develop project plans and timelines.

### 2. Requirements Gathering and Analysis

- **Objective:** Collect and document the requirements of the software.
- **Activities:**
  - Conduct interviews, surveys, and workshops.
  - Create requirement specifications.
  - Validate requirements with stakeholders.

### 3. Design

- **Objective:** Outline how the software will function and how it will be structured.
- **Activities:**
  - Create architectural designs and system models.
  - Develop UI/UX designs.
  - Prepare design specifications for developers.

### 4. Implementation (Coding)

- **Objective:** Convert design documents into actual software.
- **Activities:**
  - Write code according to specifications.
  - Conduct unit tests to ensure each component functions correctly.
  - Integrate components into a complete system.

### 5. Testing

- **Objective:** Identify and fix defects before deployment.
- **Activities:**
  - Perform various testing types (e.g., functional, integration, performance).
  - Validate that the software meets requirements.
  - Document and address bugs.

## 6. Deployment

- **Objective:** Release the software to users.
- **Activities:**
  - Deploy the software in the production environment.
  - Conduct user training and documentation.
  - Gather user feedback.

## 7. Maintenance

- **Objective:** Ensure the software continues to function and meets user needs.
- **Activities:**
  - Fix bugs and issues reported by users.
  - Implement updates and new features.
  - Monitor performance and make optimizations.

# Different between Desktop, Native, Hybrid, and Web-based software.

## 1. Desktop Software

- **Definition:** Software installed directly on a user's computer or laptop.
- **Examples:** Microsoft Office, Adobe Photoshop.
- **Advantages:**
  - High performance and speed.
  - Full access to system resources.
  - Often more feature-rich.
- **Disadvantages:**
  - Limited to the device it's installed on.
  - Requires installation and updates.

## 2. Native Software

- **Definition:** Software developed specifically for a particular platform or operating system (OS).
- **Examples:** iOS apps, Android apps.
- **Advantages:**
  - Optimized for performance on the specific OS.
  - Seamless integration with device features (camera, GPS, etc.).
- **Disadvantages:**
  - Requires separate codebases for different platforms.
  - Higher development and maintenance costs.

### 3. Hybrid Software

- **Definition:** Combines elements of both native and web applications, typically using web technologies within a native container.
- **Examples:** Apps built with frameworks like React Native or Flutter.
- **Advantages:**
  - Cross-platform compatibility with a single codebase.
  - Faster development compared to fully native apps.
- **Disadvantages:**
  - May not fully leverage device capabilities.
  - Potential performance issues compared to native apps.

### 4. Web-Based Software

- **Definition:** Applications that run in a web browser and are accessed via the internet.
- **Examples:** Google Docs, Trello.
- **Advantages:**
  - Accessible from any device with a web browser.
  - No installation required; updates are automatic.
- **Disadvantages:**
  - Requires internet access.
  - May have limited functionality compared to desktop or native apps.

### Summary Table

| Type      | Installation Required | Performance | Accessibility           | Development Cost |
|-----------|-----------------------|-------------|-------------------------|------------------|
| Desktop   | Yes                   | High        | Limited to device       | Higher           |
| Native    | Yes                   | Very High   | Platform-specific       | Highest          |
| Hybrid    | Yes                   | Moderate    | Cross-platform          | Moderate         |
| Web-Based | No                    | Variable    | Any device with browser | Lower            |