Introduction to Application Development

SDLC & Trends

Instructor: Engr. Andrew Gallemit, CpE, MSIT

College of Computing Studies

Learning Objectives

- Define SDLC & its importance
- Describe SDLC phases & methodologies
- Identify current trends in app development
- Apply SDLC knowledge to a case study

What is Application Development?

is the systematic process of designing, creating, testing, and deploying software applications for desktops, mobile devices, or the web to meet specific needs or solve business challenges. It encompasses the entire application lifecycle, from initial planning and requirements gathering to ongoing maintenance and support, and involves various phases like design, coding, testing, and launching.

The Application Development Process

Application development follows a series of phases, often referred to as the Software Development Lifecycle (SDLC):

1. Planning: Identifying the application's purpose, target users, and requirements, as well as outlining the project's goals and scope.
2. Design: Creating blueprints for the application, including user interface (UI) and user experience (UX) design, wireframes, and selecting the necessary tools and technologies.

3. Development/Coding: Building the application by writing the actual code for both the front-end (user-facing parts) and the back-end (server-side logic).
4. Testing: Thoroughly testing the application across various scenarios to find and fix bugs, security vulnerabilities, and usability issues.
5. Deployment: Releasing the application to a production server or marketplace, making it available for end-users.
6. Maintenance: Providing ongoing support, fixing new issues, implementing updates, and managing changes to keep the application functional and secure over time.

Why It Matters

- **Businesses:**

Companies use application development to streamline operations, automate tasks, and improve customer service.

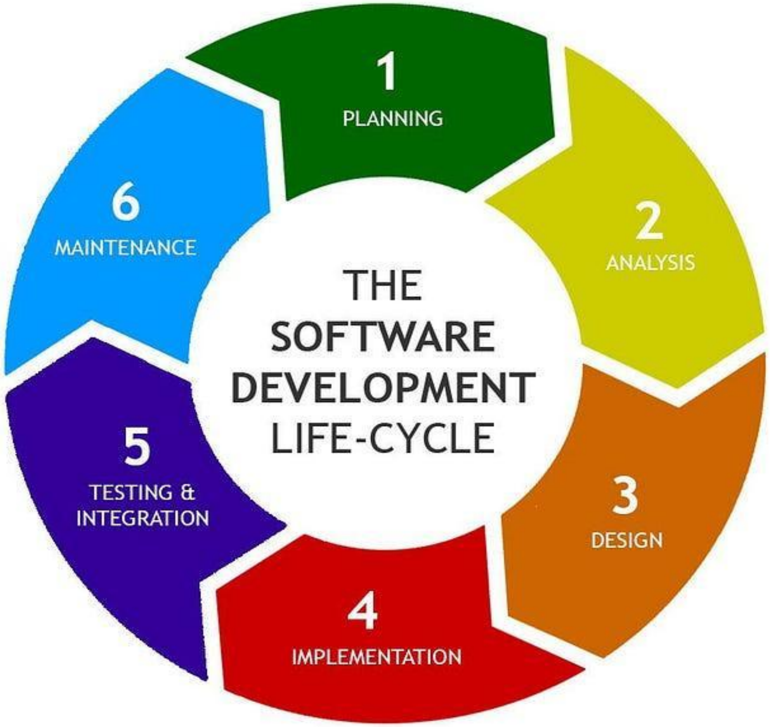
- **Users:**

Applications provide solutions, enhance productivity, and offer convenient access to information and services on various platforms.

- **Innovation:**

It allows for the creation of tailored software solutions that drive innovation and help organizations stay competitive.

What is SDLC?



Software Development Life Cycle, is a structured, step-by-step process that teams use to plan, design, develop, test, deploy, and maintain high-quality software systems. It acts as a framework or roadmap for software projects, breaking them into distinct phases with specific goals and deliverables to ensure the software meets stakeholder needs and customer expectations efficiently and cost-effectively.

Phases of SDLC

1. Planning
2. Design
4. Development/Coding
5. Testing
6. Deployment
7. Maintenance

SDLC Methodologies

- Waterfall
- Agile
- Spiral
- Iterative
- V-Model
- DevOps (CI/CD)

Waterfall Model

- Sequential design process
- Advantages: Clear structure
- Limitations: Inflexible to changes

Agile Model

- Iterative, customer collaboration
- Advantages: Flexible, adaptive
- Limitations: Requires high involvement

Spiral Model

- Focus is on managing potential risks
- The project progresses through multiple cycles (spirals), with each cycle adding functionality and refining the software
- Clients and stakeholders are actively involved throughout the process
- The iterative nature allows for changes in requirements to be incorporated into subsequent spirals, making it adaptable

Iterative Model

- Incremental Development
- It is flexible and can adapt to changing requirements or new features that emerge during development
- Customer and stakeholder feedback is incorporated early and often
- Risks can be identified and mitigated in the early stages of each iteration

V-Model

- Sequential Phases
- Verification on the Left
- Validation on the Right
- Parallel Activities

DevOps Model

- Collaboration
- Automation
- Continuous Improvement
- Shared Responsibility

Case Study Activity

- Scenario: University needs a mobile app for online enrollment.

- Question: Which SDLC model would you use and why?

Current Trends in App Development

- Cloud-based development
- Mobile-first, cross-platform
- Low-code / no-code platforms
- AI/ML integration
- Cybersecurity & DevSecOps

Trends in Action

Examples:

- TikTok → AI/ML integration
- Canva → Cloud-based + collaboration
- Microsoft PowerApps → Low-code platform

Group Activity

- Research sprint: Find examples of apps applying these trends and share with the class.

Wrap-Up Activity

- Propose a simple app idea
- Select an SDLC model
- Apply one current trend

Reflection Questions

1. Which SDLC model fits your working style? Why?

2. How can trends shape the future of software development?

Thank You

- Next session topic

Requirements Analysis, UI/UX