

Introduction to Software Development

Introduction to Software Development

Key concepts, phases, roles, and best practices

1. What Is Software Development?

Aspect	Description
Definition	The systematic process of designing, creating, testing, and maintaining software applications or systems.
Goal	Transform user needs or business requirements into reliable, functional, and maintainable software.
Scope	Includes everything from a simple script to large, distributed enterprise systems.
Disciplines	Requirements engineering, design, coding, testing, deployment, maintenance, and project management.

2. Core Concepts & Terminology

Term	Meaning
Algorithm	Step-by-step procedure for solving a problem.
Data Structure	Organized way to store and access data (e.g., arrays, linked lists, trees).
API (Application Programming Interface)	Set of rules that allows different software components to communicate.
Framework	Reusable, semi-complete platform that provides generic functionality (e.g., React, Spring).
Version Control	System for tracking changes to code (Git, Subversion).
CI/CD	Continuous Integration / Continuous Delivery – automated build, test, and deployment pipelines.
Technical Debt	Future cost incurred by shortcuts or sub-optimal solutions taken today.

3. Software Development Life Cycle (SDLC)

Introduction to Software Development

Phase	Primary Activities	Typical Deliverables
1. Planning / Feasibility	Stakeholder interviews, market analysis, risk assessment.	Project charter, feasibility report.
2. Requirements Gathering	Elicit functional & non-functional requirements, create user stories or use cases.	Requirements Specification (SRS), backlog.
3. Design	Architecture selection, UI/UX mockups, database schema, component diagrams.	Architecture diagram, UI prototypes, design docs.
4. Implementation (Coding)	Write source code, adhere to coding standards, perform peer reviews.	Source code repository, build scripts.
5. Testing	Unit, integration, system, acceptance, performance, security testing.	Test plans, test cases, defect reports.
6. Deployment	Release to production environment, configure infrastructure, roll-back plan.	Deployment scripts, release notes.
7. Maintenance	Bug fixes, enhancements, performance tuning, security patches.	Updated documentation, versioned releases.

Note: Models such as **Waterfall**, **Iterative**, **Spiral**, and **Agile** define how these phases are organized and revisited.

4. Popular Development Methodologies

Methodology	Key Characteristics	Typical Use Cases
Waterfall	Linear, sequential phases; heavy documentation.	Regulated industries, fixed-scope projects.
Agile (Scrum, Kanban)	Incremental delivery, frequent feedback, adaptable scope.	Rapidly changing requirements, product-centric teams.
DevOps	Combines development & operations; emphasizes automation & monitoring.	Cloud-native services, continuous delivery pipelines.
Lean	Eliminate waste, focus on value-stream mapping.	Startups, startups, teams seeking efficiency.

Introduction to Software Development

5. Typical Roles in a Development Team

Role	Primary Responsibilities
Product Owner / Business Analyst	Define product vision, prioritize backlog, liaise with stakeholders.