

Overview of Software Development

Overview of Software Development

(A concise, high-level guide for students, new hires, or anyone looking to understand the software creation process)

1. What Is Software Development?

- **Definition** – The systematic design, creation, testing, deployment, and maintenance of software applications or systems.
 - **Goal** – Transform user needs and business requirements into reliable, efficient, and maintainable code.
 - **Scope** – Ranges from tiny scripts (e.g., automation) to massive, distributed systems (e.g., cloud platforms).
-

2. Core Phases of the Development Lifecycle

Phase	Primary Activities	Typical Deliverables
1. Requirements Gathering	• Stakeholder interviews • Market research • Use case & user story creation	• Requirements specification (SRS) • Prioritized backlog
	• Architecture selection (monolith, microservices, serverless)	• Architecture diagrams
2. Planning & Design	• UI/UX mockups • Technology stack decision	• Wireframes / prototypes • Project plan (timeline, resources)
3. Implementation (Coding)	• Writing source code • Applying coding standards • Version control commits	• Source repository (Git, Mercurial) • Build scripts (Maven, Gradle, npm)
4. Testing & Quality Assurance	• Unit, integration, system, acceptance tests • Static analysis, code reviews • Automated test pipelines	• Test suites & results • Defect reports • Test coverage metrics

Overview of Software Development

Phase	Primary Activities	Typical Deliverables
5. Deployment	• Packaging (containers, installers) • Release to staging/production • Configuration management	• Release notes • Deployment scripts (Ansible, Terraform)
6. Maintenance & Support	• Monitoring & logging • Bug fixing, security patches • Feature enhancements	• Updated documentation • Service level reports

The phases are often iterative (see Agile & DevOps).

3. Popular Development Methodologies

Methodology	Key Characteristics	When to Use
Waterfall	Linear, sequential; each phase must finish before the next begins.	Fixed scope, regulated environments (e.g., aerospace).
Agile (Scrum, Kanban)	Incremental delivery; short sprints, continuous feedback, adaptable scope.	Dynamic requirements, fast-moving markets.
Lean	Focus on waste elimination, delivering value quickly.	Startups, product-centric teams.
DevOps	Merges development & operations; emphasizes automation, CI/CD, monitoring.	Cloud-native, continuous delivery pipelines.
Hybrid (e.g., Scrumban)	Mixes elements from multiple approaches to fit specific contexts.	Large organizations needing flexibility.

4. Essential Roles & Responsibilities

Role	Core Responsibilities
Product Owner / Business Analyst	Define vision, prioritize backlog, translate business needs.
Software Architect	Design high-level structure, choose technologies, ensure scalability.
Developer / Engineer	Write, review, and refactor code; implement features.

Overview of Software Development

Role

Core Responsibilities

QA Engineer / Tester

Create and execute test plans, automate testing, ensure quality.

DevOps Engineer

Build CI/CD pipelines, manage infrastructure as code