

Introduction to Software Development

Introduction to Software Development

Quick reference guide – concise, organized, and easy to read

1. What Is Software Development?

Aspect	Description
Definition	The systematic process of conceiving, designing, building, testing, and maintaining computer programs or applications.
Goal	Transform user needs or business requirements into functional, reliable, and maintainable software.
Key Stakeholders	• End users (customers) • Product owners / business analysts • Developers (frontend, backend, full stack) • QA/Test engineers • DevOps / operations teams • Project managers

2. Core Phases of the Software Development Lifecycle (SDLC)

Phase	Primary Activities	Typical Deliverables
1. Requirements Gathering	• Interviews, surveys, workshops • Document functional & non-functional needs	Requirements Specification, User Stories, Use case diagrams
2. Analysis & Planning	• Feasibility study • Scope definition • Estimation (time, cost, resources)	Project charter, risk register, high level architecture
3. Design	• System architecture (layers, components) • UI/UX mockups • Database schema	Architecture diagrams, wireframes, API contracts

Introduction to Software Development

Phase	Primary Activities	Typical Deliverables
4. Implementation (Coding)	• Write source code • Follow coding standards & guidelines • Version control (Git, Mercurial)	Source repository, build scripts
5. Testing	• Unit, integration, system, acceptance testing • Automated test suites, manual exploratory tests	Test plans, test reports, defect logs
6. Deployment	• Release packaging • Continuous Integration/Continuous Deployment (CI/CD) pipelines • Environment provisioning (dev, staging, prod)	Release notes, deployment scripts
7. Maintenance & Support	• Bug fixing, performance tuning • Feature enhancements • Monitoring & incident response	Patch releases, service level agreements (SLAs)

Note: Different methodologies (Waterfall, Agile, DevOps, etc.) rearrange or iterate these phases.

3. Popular Development Methodologies

Methodology	Key Characteristics	When to Use
Waterfall	Linear, sequential; each phase completed before the next begins.	Small, well-defined projects with stable requirements.
Agile (Scrum, Kanban)	Incremental delivery, frequent feedback, adaptive planning.	Projects with evolving requirements, cross-functional teams.
Extreme Programming (XP)	Emphasizes test-driven development, pair programming, continuous integration.	High-quality codebases where rapid change is expected.
DevOps	Combines development & operations; automation of build, test, deploy, monitor.	Organizations seeking fast, reliable releases and continuous delivery.

Introduction to Software Development

Methodology	Key Characteristics	When to Use
Lean	Focus on waste elimination, delivering value quickly.	Startups or teams aiming for minimal viable products (MVPs).

4. Essential Tools & Technologies

Category	Examples	Typical Use
Version Control	Git, Subversion (SVN)	Track changes, collaborate, branch/merge workflows.
Integrated Development Environments (IDEs)		