

Overview of Software Development Lifecycle

Overview of the Software Development Lifecycle (SDLC)

A concise, structured guide to the phases, activities, and best-practice considerations for building quality software.

1. What Is the SDLC?

- **Definition** – A systematic, repeatable process that guides a software project from idea to delivery and maintenance.
 - **Purpose** – Improves predictability, quality, and stakeholder alignment while controlling cost and risk.
 - **Typical Models** – Waterfall, Iterative, Incremental, V-Model, Spiral, Agile (Scrum, Kanban, XP), DevOps-oriented pipelines.
-

2. Core Phases (Common to Most Models)

Phase	Primary Goal	Key Activities	Typical Deliverables
1. Planning & Feasibility	Validate the idea and define scope.	• Stakeholder interviews • Business case & ROI analysis • Risk assessment • High-level schedule & budget • Elicitation (workshops, surveys)	Project charter, feasibility report, high-level roadmap
2. Requirements Analysis	Capture what the system must do.	• Functional & non-functional requirements • Prioritization (MoSCoW, Kano) • Traceability matrix	Requirements Specification (SRS), use-case diagrams, user stories

Overview of Software Development Lifecycle

Phase	Primary Goal	Key Activities	Typical Deliverables
3. Design	Translate requirements into a technical blueprint.	• Architecture selection (monolith, microservices, serverless) • Data modeling (ERD, schemas) • UI/UX mockups • Component diagrams, API contracts	Architecture Document, Design Specification, UI prototypes, API specs
4. Implementation (Coding)	Build the solution.	• Set up development environment • Follow coding standards & guidelines • Version control (Git) • Unit testing, code reviews • Test planning (test cases, test data)	Source code repository, build scripts, unit test suite
5. Testing & Verification	Ensure the product meets quality criteria.	• Types of testing: unit, integration, system, acceptance, performance, security • Automated regression testing • Defect tracking • Release management (staging ? production)	Test Plan, Test Report, Defect Log, Release Candidate
6. Deployment	Deliver the product to the target environment.	• Configuration management • Continuous Integration/Continuous Delivery (CI/CD) pipelines • Rollback strategy • Incident handling	Deployment scripts, release notes, environment configuration
7. Maintenance & Support	Keep the system functional and relevant.	• Bug fixes & patches • Enhancements & new features • Performance monitoring • Technical debt management	Patch releases, service level agreements (SLAs), maintenance backlog

Overview of Software Development Lifecycle

3. Popular SDLC Models – When to Use Them

Model	Characteristics	Ideal Scenarios
Waterfall	Linear, sequential; each phase must finish before the next begins.	Fixed, well-understood requirements; regulatory projects; small teams.
Iterative/Incremental	Repeats cycles; each iteration adds functional increments.	Projects with evolving requirements; need early user feedback.
Spiral	Combines iteration with risk analysis at each loop.	High-risk, large-scale systems; heavy compliance constraints.
V-Model		