# **Software Development Life Cycle (SDLC)**

What is SDLC?

The Software Development Life Cycle (SDLC) is a structured process teams use to plan, build, test, and maintain software. Its goal is to deliver high-quality software efficiently while reducing risks, cost, and time to production. Following an SDLC gives clear stages, deliverables, and decision points so teams can measure progress and manage changes.

Common SDLC Phases

1. Requirements (gathering and analysis) — define what the software must do and any constraints.  
 2. Design — architecture, interfaces, and detailed design decisions.  
 3. Implementation (coding) — developers write the software.  
 4. Testing — verify functionality, performance, and security.  
 5. Deployment — release to production environments.  
 6. Maintenance — bug fixes, updates, and enhancements after release.

Popular SDLC Models

Waterfall, Agile (Scrum/Kanban), Iterative/Incremental, Spiral, V-Model, and Prototype/Rapid Application Development (RAD). Each model arranges phases differently: Waterfall is sequential, Agile is iterative and incremental, Spiral focuses on risk-driven iterations, and V-Model emphasizes verification and validation.

Why use an SDLC — Key benefits

• Predictability and planning, helping estimate time and cost.  
 • Better risk management through defined checkpoints and reviews.  
 • Improved quality via explicit testing stages and traceable requirements.  
 • Clear documentation and accountability for handoffs between roles.  
 • Easier maintenance and evolution because of structured deliverables and versioning.

Choosing a model

Choose a model based on project size, risk, clarity of requirements, stakeholder involvement, and delivery cadence. For well-defined, low-change projects Waterfall can work. For evolving requirements or frequent releases, Agile is usually a better fit.