

# Software Development Life Cycle (SDLC)

*A Comprehensive Guide to the Five Essential Phases*

## PHASE 1: REQUIREMENTS

<b>Why It Matters:</b>	<b>Foundation of Success:</b> Defines what the software must do. Clear requirements prevent costly changes later and ensure stakeholder alignment.
<b>Key Activities:</b>	<ul style="list-style-type: none"><li>• Gather stakeholder needs</li><li>• Document functional &amp; non-functional requirements</li><li>• Create requirement specifications</li><li>• Define acceptance criteria</li></ul>
<b>Outputs:</b>	Requirements Document, Use Cases

## PHASE 2: DESIGN

<b>Why It Matters:</b>	<b>Blueprint Creation:</b> Transforms requirements into a technical plan. Good design ensures scalability, maintainability, and optimal performance.
<b>Key Activities:</b>	<ul style="list-style-type: none"><li>• Create system architecture</li><li>• Design database schemas</li><li>• Define interfaces &amp; APIs</li><li>• Develop UI/UX mockups</li></ul>
<b>Outputs:</b>	Design Documents, Architecture Diagrams

## PHASE 3: IMPLEMENTATION

<b>Why It Matters:</b>	<b>Bringing Ideas to Life:</b> Converts design into working code. Quality coding practices here determine software reliability and future maintainability.
<b>Key Activities:</b>	<ul style="list-style-type: none"><li>• Write source code</li><li>• Follow coding standards</li><li>• Conduct code reviews</li><li>• Version control management</li></ul>
<b>Outputs:</b>	Source Code, Executables, Libraries

## PHASE 4: TESTING

<b>Why It Matters:</b>	<b>Quality Assurance:</b> Validates that software meets requirements and is defect-free. Early bug detection saves time and money while ensuring user satisfaction.
<b>Key Activities:</b>	<ul style="list-style-type: none"><li>• Unit testing</li><li>• Integration testing</li><li>• System testing</li><li>• User acceptance testing (UAT)</li></ul>

Outputs:	Test Reports, Bug Logs, Fixed Code
----------	------------------------------------

PHASE 5: DEPLOYMENT	
Why It Matters:	<b>Delivery &amp; Maintenance:</b> Releases software to users and ensures ongoing support. Proper deployment minimizes downtime and enables continuous improvement.
Key Activities:	<ul style="list-style-type: none"><li>• Release to production</li><li>• User training &amp; documentation</li><li>• Monitor performance</li><li>• Ongoing maintenance &amp; updates</li></ul>
Outputs:	Live Application, User Manuals, Support System

How the Phases Interconnect

Requirements → Design	Requirements inform design decisions and constraints
Design → Implementation	Design blueprints guide coding structure and patterns
Implementation → Testing	Code is tested against design specs and requirements
Testing → Deployment	Validated software is released; bugs loop back to implementation
Deployment → Requirements	User feedback drives new requirements for next iteration

*Note: Modern SDLC often follows iterative/agile approaches where these phases cycle repeatedly, allowing for continuous refinement and adaptation.*