SDLC (Software Development Life Cycle) Tutorial: What is, Phases, Model

What is SDLC?

The Software Development Lifecycle is a systematic process for building software that ensures the quality and correctness of the software built. SDLC process aims to produce high-quality software which meets customer expectations. The software development should be complete in the pre-defined time frame and cost.

SDLC consists of a detailed plan which explains how to plan, build, and maintain specific software. Every phase of the SDLC lifecycle has its own process and deliverables that feed into the next phase.

In this Software Development Lifecycle tutorial, you will learn

- What is SDLC?
- Why SDLC?
- SDLC Phases
- Phase 1: Requirement collection and analysis
- Phase 2: Feasibility study
- Phase 3: Design
- Phase 4: Coding
- Phase 5: Testing
- Phase 6: Installation/Deployment
- Phase 7: Maintenance
- Popular SDLC models

Why SDLC?

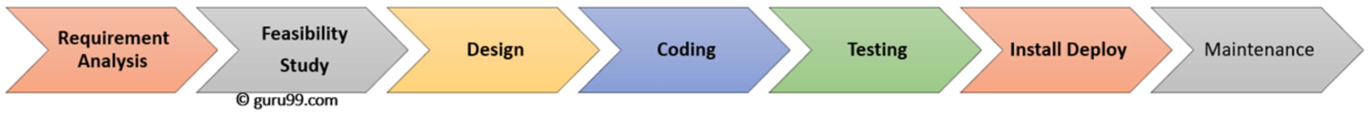
Here, are prime reasons why SDLC is important for developing a software system.

- It offers a basis for project planning, scheduling, and estimating
- Provides a framework for a standard set of activities and deliverables
- It is a mechanism for project tracking and control

- Increases visibility of project planning to all involved stakeholders of the development process
- Increased and enhance development speed
- Improved client relations
- Helps you to decrease project risk and project management plan overhead

SDLC Phases

The entire SDLC process divided into the following stages:



(/images/1/080118_0641_SDLCSoftwar1.png)

- Phase 1: Requirement collection and analysis
- Phase 2: Feasibility study:
- Phase 3: Design:
- Phase 4: Coding:
- Phase 5: Testing:
- Phase 6: Installation/Deployment:
- Phase 7: Maintenance:

In this tutorial, I have explained all these phases

Phase 1: Requirement collection and analysis:

The requirement is the first stage in the SDLC process. It is conducted by the senior team members with inputs from all the stakeholders and domain experts in the industry. Planning for the quality assurance requirements and recognization of the risks involved is also done at this stage.

This stage gives a clearer picture of the scope of the entire project and the anticipated issues, opportunities, and directives which triggered the project.

Requirements Gathering stage need teams to get detailed and precise requirements. This helps companies to finalize the necessary timeline to finish the work of that system.

Phase 2: Feasibility study:

Once the requirement analysis phase is completed the next step is to define and document software needs. This process conducted with the help of 'Software Requirement Specification' document also known as 'SRS' document. It includes everything which should be designed and developed during the project life cycle.

There are mainly five types of feasibilities checks:

- **Economic:** Can we complete the project within the budget or not?
- **Legal:** Can we handle this project as cyber law and other regulatory framework/compliances.
- **Operation feasibility:** Can we create operations which is expected by the client?
- **Technical:** Need to check whether the current computer system can support the software
- **Schedule:** Decide that the project can be completed within the given schedule or not.

Phase 3: Design:

In this third phase, the system and software design documents are prepared as per the requirement specification document. This helps define overall system architecture.

This design phase serves as input for the next phase of the model.

There are two kinds of design documents developed in this phase:

High-Level Design (HLD)

- Brief description and name of each module
- An outline about the functionality of every module
- Interface relationship and dependencies between modules
- Database tables identified along with their key elements
- Complete architecture diagrams along with technology details

Low-Level Design(LLD)

- Functional logic of the modules
- Database tables, which include type and size
- Complete detail of the interface
- Addresses all types of dependency issues
- Listing of error messages
- Complete input and outputs for every module

Phase 4: Coding:

Once the system design phase is over, the next phase is coding. In this phase, developers start build the entire system by writing code using the chosen programming language. In the coding phase, tasks are divided into units or modules and assigned to the various developers. It is the longest phase of the Software Development Life Cycle process.

In this phase, Developer needs to follow certain predefined coding guidelines. They also need to use programming tools like compiler, interpreters, debugger to generate and implement the code.

Phase 5: Testing:

Once the software is complete, and it is deployed in the testing environment. The testing team starts testing the functionality of the entire system. This is done to verify that the entire application works according to the customer requirement.

During this phase, QA and testing team may find some bugs/defects which they communicate to developers. The development team fixes the bug and send back to QA for a re-test. This process continues until the software is bug-free, stable, and working according to the business needs of that system.

Phase 6: Installation/Deployment:

Once the software testing phase is over and no bugs or errors left in the system then the final deployment process starts. Based on the feedback given by the project manager, the final software is released and checked for deployment issues if any.

Phase 7: Maintenance:

Once the system is deployed, and customers start using the developed system, following 3 activities occur

- Bug fixing - bugs are reported because of some scenarios which are not tested at all
- Upgrade - Upgrading the application to the newer versions of the Software
- Enhancement - Adding some new features into the existing software

The main focus of this SDLC phase is to ensure that needs continue to be met and that the system continues to perform as per the specification mentioned in the first phase.

Popular SDLC models

Here, are some most important phases of SDLC life cycle:

Waterfall model

The waterfall is a widely accepted SDLC model. In this approach, the whole process of the software development is divided into various phases. In this SDLC model, the outcome of one phase acts as the input for the next phase.

This SDLC model is documentation-intensive, with earlier phases documenting what need be performed in the subsequent phases.

Incremental Approach

The incremental model is not a separate model. It is essentially a series of waterfall cycles. The requirements are divided into groups at the start of the project. For each group, the SDLC model is followed to develop software. The SDLC process is repeated, with each release adding more functionality until all requirements are met. In this method, every cycle act as the maintenance phase for the previous software release. Modification to the incremental model allows development cycles to overlap. After that subsequent cycle may begin before the previous cycle is complete.

V-Model

In this type of SDLC model testing and the development, the phase is planned in parallel. So, there are verification phases on the side and the validation phase on the other side. V-Model joins by Coding phase.

Agile Model

Agile methodology is a practice which promotes continue interaction of development and testing during the SDLC process of any project. In the Agile method, the entire project is divided into small incremental builds. All of these builds are provided in iterations, and each iteration lasts from one to three weeks.

Spiral Model

The spiral model is a risk-driven process model. This SDLC model helps the team to adopt elements of one or more process models like a waterfall, incremental, waterfall, etc.

This model adopts the best features of the prototyping model and the waterfall model. The spiral methodology is a combination of rapid prototyping and concurrency in design and development activities.

Big bang model

Big bang model is focusing on all types of resources in software development and coding, with no or very little planning. The requirements are understood and implemented when they come.

This model works best for small projects with smaller size development team which are working together. It is also useful for academic software development projects. It is an ideal model where requirements is either unknown or final release date is not given.

Conclusion

- The SDLC is a systematic process for building software that ensures the quality and correctness of the software built
- SDLC process provides a framework for a standard set of activities and deliverables
- Seven different SDLC stages are 1) Requirement collection and analysis 2) Feasibility study: 3) Design 4) Coding 5) Testing: 6) Installation/Deployment and 7) Maintenance
- The senior team members conduct the requirement analysis phase
- Feasibility Study stage includes everything which should be designed and developed during the project life cycle
- In the Design phase, the system and software design documents are prepared as per the requirement specification document
- In the coding phase, developers start build the entire system by writing code using the chosen programming language
- Testing is the next phase which is conducted to verify that the entire application works according to the customer requirement.
- Installation and deployment face begins when the software testing phase is over, and no bugs or errors left in the system
- Bug fixing, upgrade, and engagement actions covered in the maintenance face
- Waterfall, Incremental, Agile, V model, Spiral, Big Bang are some of the popular SDLC models
- SDLC consists of a detailed plan which explains how to plan, build, and maintain specific software

Next (/what-is-sdlc-or-waterfall-model.html)

YOU MIGHT LIKE:

SOFTWARE TESTING

(/functional-testing-vs-non-functional-testing.html) (/functional-testing-vs-non-functional-testing.html)

Functional Testing Vs Non-Functional Testing: What's the Difference?

(/functional-testing-vs-non-functional-testing.html)

SDLC

(/code-coverage-tools.html) (/code-coverage-tools.html)

13 Best Code Coverage Tools for Java, C, C++, C#, Python in 2019

(/code-coverage-tools.html)

SOFTWARE TESTING

(/quality-management-plan-template.html) (/quality-management-plan-template.html)

Quality Management Plan Template: Download with Sample Example

(/quality-management-plan-template.html)

SDLC

(/functional-programming-tutorial.html) (/functional-programming-tutorial.html)

What is Functional Programming? Tutorial with Example

(/functional-programming-tutorial.html)

AGILE TESTING

(/waterfall-vs-agile.html) (/waterfall-vs-agile.html)

Waterfall Vs. Agile: Must Know Differences

(/waterfall-vs-agile.html)

SOFTWARE TESTING

(/iot-testing-challenges-tools.html) (/iot-testing-challenges-tools.html)

IoT Testing Tutorial: What is, Process, Challenges & Tools

(/iot-testing-challenges-tools.html)

Software Engineering Tutorial

1) SDLC Tutorial (/software-development-life-cycle-tutorial.html)

2) What is Waterfall Model in SDLC? (/what-is-sdlc-or-waterfall-model.html)

3) Incremental Model in SDLC (/what-is-incremental-model-in-sdlc-advantages-disadvantages.html)

4) What is Spiral Model? (/what-is-spiral-model-when-to-use-advantages-disadvantages.html)

5) What is RAD Model? (/what-is-rad-rapid-software-development-model-advantages-disadvantages.html)

6) Prototyping Model (/software-engineering-prototyping-model.html)

7) Waterfall vs. Incremental vs. Spiral vs. Rad Model (/compare-waterfall-vs-incremental-vs-spiral-vs-rad.html)

8) Capability Maturity Model (CMM) (/capability-maturity-model-cmm-cmm-levels-a-fool-s-guide.html)

9) N Tier(Multi-Tier) Architecture (/n-tier-architecture-system-concepts-tips.html)

10) What is Full Stack Developer? (/full-stack-developer.html)

11) Functional Programming Tutorial (/functional-programming-tutorial.html)

(https://www.facebook.com/guru99com/)
(https://twitter.com/guru99com)
(https://www.youtube.com/channel/UC19i1XD6k88KqHlET8atqFQ)
(https://forms.aweber.com/form/46/724807646.htm)

About

About Us (/about-us.html)
Advertise with Us (/advertise-us.html)
Write For Us (/become-an-instructor.html)
Contact Us (/contact-us.html)

Career Suggestion

SAP Career Suggestion Tool (/best-sap-module.html)
Software Testing as a Career (/software-testing-career-complete-guide.html)

Interesting

Books to Read! (/books.html)
Blog (/blog/)
Quiz (/tests.html)
eBook (/ebook-pdf.html)

Execute online

Execute Java Online (/try-java-editor.html)
Execute Javascript (/execute-javascript-online.html)