Manual QA

Software Development Life Cycle - SDLC

SOFTWARE DEVELOPMENT LIFE CYCLE - SDLC

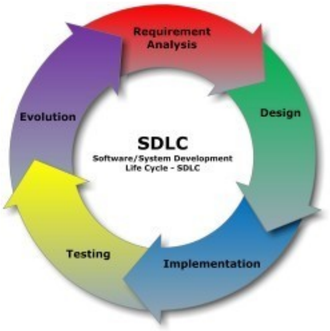
SDLC Phases Explained

What are the SDLC Phases? Software Development Life Cycle or **SDLC** is a process used to develop software. There are different stages or phases within the Software Development Life Cycle and in each phase, different activities take place.

SDLC creates a structure for the development teams to be able to design, create and deliver high quality software by defining various tasks that need to happen the life cycle defines a methodology for improving the quality of software and the overall development process.

The methodology within the SDLC process can vary across organizations, but standards such as ISO/IEC 12207 represent processes that establish a life cycle for software, and provide a standard for building and maintaining software.

The intent of a SDLC process it to help produce a product that is cost-efficient, effective, and of high quality.



Phase 1: Requirement Analysis

Software Development Life Cycle begins with Requirement Analysis phase, where the stakeholders discuss the requirements of the software that needs to be developed to achieve a goal. The aim of the requirement analysis phase is to capture the detail of each requirement and to make sure everyone understands the scope of the work and how each requirement is going to be fulfilled.

It is a normal practice to also discuss how each requirement will be tested and so testers can add great value in participating in requirement analysis meetings.

Depending on which software development methodology is used, different approaches are taken in moving from one phase to another. For example, in the waterfall or V model, the requirement analysis phase are saved in a SRS (Software Requirement Specification) document and needs to be finalized before the next phase can take place.

Phase 2: Design

The next stage of Software Development Life Cycle is the Design phase. During the design phase, developers and technical architects start the high-level design of the software and system to be able to deliver each requirement.

The technical details of the design is discussed with the stakeholders and various parameters such as risks, technologies to be used, capability of the team, project constraints, time and budget are reviewed and then the best design approach is selected for the product.

The selected architectural design, defines all the components that needs to be developed, communications with third party services, user flows and database communications as well as front-end representations and behavior of each components. The design is usually kept in the Design Specification Document (DSD).

Phase 3: Implementation

After the requirements and design activity is completed, the next phase of the Software Development Life Cycle is the implementation or development of the software. In this phase, developers start coding according to the requirements and the design discussed in previous phases.

Database admins create the necessary data in the database, front-end developers create the necessary interfaces and GUI to interact with the back-end all based on guidelines and procedures defined by the company.

Developers also write unit tests for each component to test the new code that they have written, review each other's code, create builds and deploy software to an environment. This cycle of development is repeated until the requirements are met.

Phase 4: Testing

Testing is the last phase of the Software Development Life Cycle before the software is delivered to customers. During testing, experienced testers start to test the system against the requirements.

The testers aim to find defects within the system as well as verifying whether the application behaves as expected and according to what was documented in the requirements analysis phase. Testers can either use a test script to execute each test and verify the results, or use exploratory testing which is more of an experience based approach. It is possible that defects are identified in the testing phase. Once a defect is found, testers inform the developers about the details of the issue and if it is a valid defect, developers will fix and create a new version of the software which needs to be verified again.

This cycle is repeated until all requirements have been tested and all the defects have been fixed and the software is ready to be shipped.

Phase 5: Deployment

Once the software has been fully tested and no high priority issues remain in the software, it is time to deploy to production where customers can use the system.

Phase 6: Maintenance

Once a version of the software is released to production, there is usually a maintenance team that looks after any post-production issues.

If an issue is encountered in the production the development team is informed and depending on how severe the issue is, it might either require a hot-fix which is created and shipped in a short period of time or if not very severe, it can wait until the next version of the software.

Conclusion

All the stages in the Software Development Life Cycle defined above are applicable to any software development methodology, but the duration and the activities in each phase depend on whether you follow the V Model development methodology or Agile.

In Agile, the duration to deliver working software is typically between 2 to 4 weeks and so each of the above phases is shortened. Also in agile, it is a whole team approach where the developers and testers are involved and work together to make a high quality product.