SDLC

Software Development Life Cycle

Explained

The Software Development Life Cycle (SDLC) model is a process by which application development takes place. The SDLC was borne out of the Waterfall methodology of project management where one phase of the development process must be completed prior to proceeding to the next phase. All phases of this model are essential for the successful delivery of a software application. While the SDLC model can be used with hardware implementations, it is typically used with application development, which will be the focus of this guide. This guide is intended to be a very basic overview of the SDLC.

This guide is organized by the six phases of the Software Development Life Cycle: Plan, Design, Develop, Test, Implement, and Maintain as illustrated in the diagram below.

Phases of the Software Development Life Cycle

Plan

The Plan phase is sometimes called the Requirements Analysis phase, however, more goes into planning a software development project than just gathering and analyzing requirements. The following is typically involved in the Planning phase of the SDLC:

- **Why?** Determine what concern this software will address. For what reason will you use this application?
- **Should We?** A feasibility study may be conducted during this phase
- **What?** Requirements are gathered from the customer, analyzed, and discussed among the project team
- **What If?** A risk assessment is typically performed during this phase; this assessment helps to plan for possible threats to the project
- **When?** A project plan, or project schedule, is developed based on requirements and availability of the development team

Deliverables: Requirements document, project schedule and plan, scope of work

Gathering and analyzing requirements is the most important task in the Plan phase

Design

The Design phase is where the system specifications are determined and documented in a design document. Think of this phase as building the system on paper.

Functional requirements gathered in the Plan phase are analyzed and detailed system architecture is documented.

The following elements of the application are determined and documented in this phase:

- The architecture of the application is determined. An example of this is the design of the graphical user interface (GUI)
- How the application will interact with other software and systems is determined and documented
- Processes determine how the application will be used and by whom
- Diagrams are an important part of this phase

Deliverables: Design document, processes and procedures, diagrams of application integration

Utilizing stakeholder requirements from the Plan phase is vital to the success of the Design phase

Develop

The Develop phase is where the application is coded and built. The Systems Engineer or Project Manager will work closely with the development team during this phase to ensure customer requirements are met.

Using the Waterfall methodology, customer involvement is minimal during this phase.

The following elements will be developed during this phase:

- A beta version of the application will be coded
- Processes and procedures will be documented
- Training materials will be developed
- Test scrips will be developed to prepare for the test phase
- Test resources will be determined
- This phase could include changes to the design document and application as a whole

Deliverables: Beta version of the application, test scripts, training materials

This is typically the longest phase of the SDLC

SDLC

Software Development Life Cycle

Explained

Test

The Test phase is where a beta version of the application is tested by end users. This type of testing is called User Acceptance Testing (UAT).

This phase could be repeated during the SDLC depending on the outcome of initial testing.

The following events take place during this phase:

- A beta version of the application is moved from the development server to the test server
- Test scripts are distributed to UAT testers
- UAT testing takes place
- Server load testing is part of the test phase to ensure performance
- Processes and procedures are also tested
- Test results are documented
- Changes, if any, are documented and made to the application and retested

Deliverables: Test results, documented changes made to system based on testing

Change management during this phase is important once initial testing is completed and documented

Implement

The Implement phase is where the application is deployed after successful testing has been concluded. If the application is integrated with other software, it will take place during this phase.

Another important element of this phase is end user training using training materials developed in the Develop phase.

The following events take place during this phase:

- Training should take place approximately one week prior to roll out
- The application is deployed to end users
- User feedback is collected and documented
- Members of the project team are on standby for any end user questions
- A close out meeting takes place to determine lessons learned during the project
- Customer sign off on the final product

Deliverables: Application, training, customer sign-off, Lessons Learned document

Communication with the end user community is vital during this phase

Maintain

The Maintain phase takes place after successful deployment of the application.

If there is an onsite program team for a specified timeframe, they will likely be responsible for maintaining the application going forward. Otherwise, the customer's own IT team will take over maintenance of the application with minimal involvement from the Project Manager or Systems Engineer.

Customer follow-up is essential if the application is taken over by the customer's IT team to ensure the application is working as intended.

An annual assessment (or gap analysis) should also take place to determine if any updates need to be made to the application based on changes to the customer's business.

Deliverables: A satisfied customer that would be happy to do business with you in the future.

Maintaining communication with the customer after the project concludes is important to future business!

Agile vs. Waterfall

Although the SDLC process is based off of the Waterfall methodology of application development, more and more organizations are using this model with the Agile development methodology. The Agile methodology is iterative and simply involves more customer interaction throughout the design and development phase while elements of the SDLC are repeated until the final product is delivered.

Waterfall = Predictive approach Agile = Adaptive approach