Systems Development Life Cycle Phases

Systems Development Life Cycle (SDLC)

There are seven main stages in the SDLC.

1. Project initiation and planning

The first stage of planning is when the initial idea is generated. The idea is formed with a business need in mind and the way in which the new system will address it. The development team must consider the new system's objectives, the strategies they will use, the scope of the new system, costs associated with new system, and the time scale. The idea is documented with information regarding all the considerations and drawn up into a project proposal for management's approval.

It is also important to include security considerations in this initial phase. Security mechanisms should also be integrated at each phase in the SDLC. Security professionals have to take into several security aspects into consideration during the planning phase in order to develop and integrate appropriate security. These considerations include:

- how valuable or sensitive the information is before and after so that they know how much protection is required
- if the information has been valued by the system owner, and if any classifications have been defined
- if the running of the application poses a threat of access control to the information
- whether resultant data will be produced in public places, and whether control mechanisms are needed for displaying the resultant data
- will operation of the application need protected areas, and how will it affect the company operations and culture, and
- whether the company will require the resultant data for operation, and if the application will need maintenance support

The risk management process usually incorporates these questions.

2. Defining functional objectives

The second phase of defining the functional objectives is when the development team put together a list of what they want the system to be able to do. For example, what end-user needs the system should meet and what functions it requires to meet them. Possible future needs should be considered as well as current needs.

As the functional objectives are being defined, the development team occasionally go on to plan from the initial phase and make adaptations where necessary. All security plans need to be finalized at this phase.

If the ongoing project is slightly smaller, this phase is sometimes integrated as part of the initial planning phase.

3. Defining system requirements

In the third phase, the development team needs to work out how the system will operate in order to perform the intended functions. The system architecture, output, and interfaces all need to be created. The development team needs to determine data flow, and the requirements of how the output needs to operate.

The finalized security plans now need to be integrated into the system design. The security mechanisms must take into account the whole company's security.

4. Developing and implementing

After all the planning and strategies are in place, the system is ready to enter the developing and implementing phase. The system is created according to all the planning and defined system requirements and is implemented using test cases. The new system is implemented and checked that it is running appropriately and meeting the requirements.

In this phase everything is documented so that if the development team need to come back later and adjust for maintenance, corrections, or improvements, they have access to everything that took place in this phase. Security mechanisms are integrated with the development of the system and are also checked with the test runs. As well checking the security of system operation, the code should also be checked so that any inconsistencies become evident. This means risks are reduced by dealing with highlighted inconsistencies early on.

5. Recording and documenting

The development team needs to make sure that the new system is recording appropriate data, and logging operations correctly. There should be a logging system that keeps track of different versions. For example, for each version logged, the date it was last modified, the person who last modified it, and the changes that were made can all be recorded. Common Program Controls need to be put into place to achieve this. Several checks can be used including:

- checks for application, operating, utility, privileged functions, and system documentation
- component, and restart and recovery procedure checks
- checks for process integrity using counting such as totals, subtotals, and balances
- data integrity checks internally from start to finish, and
- valid address references and data types

6. Testing and evaluating

After the system is fully developed and documented, it has to be evaluated. The system should be tested using test data. There are several guidelines that should be followed while selecting or creating the test data. For example, some test data should be selected to test the range limits of

the system, as well as data that is known to be beyond the system range limits. Some data should be completely random, and some data should be known to be valid data. Data that is selected for testing should never be the company's actual data that will eventually be what the system is used for. This is because of the risk of data exposure if a fault becomes evident. Therefore, all test data should be cleared of any kind of sensitive information.

All test data and results should be recorded. In case an error does occur, the design team can study the data that caused the error to determine the reason and resolve the problem.

Security testing is extremely important at this phase. All security mechanisms need to be thoroughly checked. If there is an inconsistency, the company risks exposure, distortion, or loss of important data. The security of the system must also be compliant with the company's security policies. The development team needs to test whether the new system fulfills the security plans that were developed and designed in the initial phases of the project.

Once the security has been evaluated, a Security Officer can certify that the system meets all the security requirements and standards. Only once the system has been certified, is management able to authorize the system to go into production.

7. Producing and installing

The seventh phase is when the system is ready to be implemented and put into production. The new system is installed and put into use. Users are trained during this phase, and any data that needs to be converted is converted at this time.

System Life Cycle (SLC)

The SLC has two addition phases that can be added onto the previous seven development phases.

8. Maintaining and operation

When the system is in operation, it needs to be constantly monitored. The performance of the new system can be followed to make sure that it stays constant. If any inconsistencies become evident, the system can then be adapted to resolve them. By following the system operation, the development team can also identify areas where improvements could be made.

Security is also monitored and maintained, including the backup and recovery processes. During this phase, the development team can make sure that the data is being appropriately handled and protected, and that the security mechanisms that were implemented are operating correctly. If any changes are made at this phase, a risk assessment and an additional certification process can be conducted to make sure the systems remain secure. Compliance is also monitored during this phase.

9. Revisions and replacements

Timely evaluations and audits should be conducted, and any errors or inconsistencies identified can be dealt with in the revisions phase. Also, if there has been an area identified where improvements can be made, the changes can be made at this phase.

Any changes that are made need to follow the SDLC and be appropriately documented. Documenting inconsistencies that occurred, and the actions that were performed to resolve them, creates a strong case for making improvements in the future.

(Software Development Security)