

Tutorial 2: Process Activities

Activities such as budgets, requirements gathering, and documentation writing, are included in the cycle, as well as the more technical elements. The software development life cycle comprises of seven distinct phases:

- Planning
- Analysis
- Design
- Development
- Testing
- Deployment
- Maintenance

Phase 1: Planning

In the planning phase, project goals are determined and a high-level plan for the intended project is established. Planning is the most fundamental and critical organizational phase. The three primary activities involved in the planning phase are as follows:

1. Identification of the system for development
2. Feasibility assessment
3. Creation of project plan

Phase 2: Analysis

In the analysis phase, end user business requirements are analysed and project goals converted into the defined system functions that the organization intends to develop. The three primary activities involved in the analysis phase are as follows:

1. Gathering business requirement
2. Creating process diagrams
3. Performing a detailed analysis

Business requirement gathering is the most crucial part at this level of SDLC. Business requirements are a brief set of business functionalities that the system needs to meet in order to be successful. Technical details such as the types of technology used in the implementation of the system need not be defined in this phase. A sample business requirement might look like “The system must track all the employees by their respective department, region, and the designation”. This requirement is showing no such detail as to how the system is going to implement this requirement, but rather what the system must do with respect to the business.

The requirements engineering process results in the production of a software requirements document that is the specification for the system.

There are 3 main phases in the requirements engineering process:

1. Requirements elicitation and analysis. This is the process of deriving the system requirements through observation of existing systems, discussions with potential users, requirements workshop, storyboarding, etc.

2. Requirements specification. This is the activity of translating the information gathered during the analysis activity into a document that defines a set of requirements. Two types of requirements may be included in this document: user (functional) requirements and system (non-functional) requirements.
3. Requirements validation. It is determined whether the requirements defined are complete. This activity also checks the requirements for consistency.

Phase 3: Design

In the design phase, we describe the desired features and operations of the system. This phase includes business rules, pseudo-code, screen layouts, and other necessary documentation. The two primary activities involved in the design phase are as follows:

1. Designing of IT infrastructure
2. Designing of system model

To avoid any crash, malfunction, or lack of performance, the IT infrastructure should have solid foundations. In this phase, the specialist recommends the kinds of clients and servers needed on a cost and time basis, and technical feasibility of the system. Also, in this phase, the organization creates interfaces for user interaction. Other than that, data models and entity relationship diagrams (ERDs) are also created in the same phase.

A software design is a description of the architecture of the software to be implemented, the data which is part of the system, the interfaces between system components and, sometimes, the algorithms used. The design process activities are the followings:

1. Architectural design. The sub-systems of system and their relationships are identified based on the main functional requirements of software.
2. Abstract specification. For each sub-system, an abstract specification of its services and the constraints under which it must operate is defined.
3. Interface design. Interfaces allow the sub-system's services to be used by other sub-systems. The representation of interface should be hidden. In this activity the interface is designed and documented for each sub-system. The specification of interface must be unambiguous.
4. Component design. Services are allocated to components and the interfaces of these components are designed.
5. Data structure design. The data structures used in the system implementation are designed in detail and specified.
6. Algorithm design. In this activity the algorithms used to provide services are designed in detail and specified.

Phase 4: Development

In the development phase, all the documents from the previous phase are transformed into the actual system. The two primary activities involved in the development phase are as follows:

1. Development of IT infrastructure
2. Development of database and code

In the design phase, only the blueprint of the IT infrastructure is provided, whereas in this phase the organization actually purchases and installs the respective software and hardware in order to support the IT infrastructure. Following this, the creation of the database and actual code can begin to complete the system on the basis of given specifications.

The implementation phase of software development is the process of converting a system specification into an executable system through the design of system.

Phase 5: Testing

In the testing phase, all the pieces of code are integrated and deployed in the testing environment. Testers then follow Software Testing Life Cycle activities to check the system for errors, bugs, and defects to verify the system's functionalities work as expected or not, often. The two primary activities involved in the testing phase are as follows:

1. Writing test cases
2. Execution of test cases

Testing is a critical part of software development life cycle. To provide quality software, an organization must perform testing in a systematic way. Once test cases are written, the tester executes them and compares the expected result with an actual result in order to verify the system and ensure it operates correctly. Writing test cases and executing them manually is an intensive task for any organization, which can result in the success of any business if executed properly.

Software validation or, more generally, verification and validation (V & V) is intended to show that a system conforms to its specification and that the system meets the expectations of the customer buying the system. It involves checking the processes at each stage of the software process. The majority of validation costs are incurred after implementation when the operation of system is tested.

The software is tested in the usual three-stage testing process. The system components, the integrated system and finally the entire system are tested. Component defects are generally discovered early in the process and the interface problems during the system integration. The stages in the testing process are:

1. Component (or unit) testing. Individual components are tested to ensure that they operate correctly. Each component is tested independently, without other system components.
2. System testing. The components are integrated to make up the system. This testing process is concerned with finding errors that result from interactions between components and component interface problems. It is also concerned with validating that the system meets its functional and non-functional requirements.
3. Acceptance testing. It is considered a functional testing of system. The system is tested with data supplied by the system customer.

Phase 6: Deployment

During this next phase, the system is deployed to a real-life (the client's) environment where the actual user begins to operate the system. All data and components are then placed in the production environment. This phase is also called referred to as 'delivery.'

Phase 7: Maintenance

In the maintenance phase, any necessary enhancements, corrections, and changes will be made to make sure the system continues to work, and stay updated to meet the business goals. It is necessary to maintain and upgrade the system from time to time so it can adapt to future needs. The three primary activities involved in the maintenance phase are as follows:

1. Support the system users
2. System maintenance
3. System changes and adjustment

Software evolution, specifically software maintenance, is the term used in software engineering to refer to the process of developing software initially, then repeatedly updating it for various reasons.

The aim of software evolution would be to implement the possible major changes to the system. The existing larger system is never complete and continues to evolve. As it evolves, the complexity of the system will grow. The main objectives of software evolution are ensuring the reliability and flexibility of the system. The costs of maintenance are often several times the initial development costs of software.

Exercises

- 1- List the activities in each phase.
- 2- What are the benefits of (Software specification) **requirement phase**?
- 3- Describe how could **requirement document** be used through the development life cycle?