

## Systems Development Life Cycle

### Systems Life Cycle – Stages and Activities

**Systems Development Life Cycle (SDLC)**, also called waterfall development, consists of well-defined phases that are followed methodically. Most of the larger organizations use the SDLC process in their development to check if the requirements are followed, including its progress.

**Systems Life Cycle** is the oldest method when building information systems. It is a phased approach in creating or building a system by dividing systems development into formal stages. It is still used for large complex systems that require a formal requirements analysis, predefined specifications, and tight control in the process of creating a system.

Every step or stage in the system life cycle has a corresponding activity. These activities are sequential and happen during the process of developing a system. These are the following stages and activities:

- a. **Systems Analysis** – This is the first stage in a system life cycle that has certain steps to come up on the data. It requires analyzing the data, identifying the problems, and determining the information requirements and objectives of the system.

These are the steps and sample of questions for identifying the possible needs to create a system:

1. **Investigation**
    - What is the business process needed to support a system?
    - What problems do you want to solve?
  2. **Technical Feasibility Study**
    - Is the technology available to create the system?
    - Which of the available technologies should be used?
  3. **Economic Feasibility Study**
    - What are the resources needed to implement the system?
    - Will the system's benefits outweigh its costs?
  4. **Operational Feasibility Study**
    - Will the system be used to its full capacity?
    - Will the system be used appropriately by its intended users?
  5. **Requirements Definition**
    - Does it support other systems such as web and mobile technologies?
- b. **System Design** – In this phase, the system design shows how the system will fulfill the objectives based on the requirements that have come up. The designer of the system will be responsible for giving the details of the system specification, which will deliver all the functions that have been identified in the system analysis phase.

This phase comprises three (3) steps: description of the functions or components and how they will work, construction, and testing.

To communicate ideas about the data that have been gathered, systems analysts and programmers use convention symbols, such as the Data Flow Diagram (DFD). DFD symbols are suitable for describing any information systems (IS) even if this is not computer-based. It helps pinpoint the weaknesses in an existing system. While DFDs are easy to learn and use, these still have limitations like any diagramming method and cannot describe a system completely.

Another way to communicate ideas is the Unified Modeling Language (UML). UML is a graphical standard for visualizing, specifying, and documenting software. This tool helps developers to communicate and logically validate desired features in the design phases of software development projects. It consists of diagrams that describe the use case, class, interaction, activity, and physical components.

- c. **Programming** – In this stage, the system specifications that were prepared during the design stage are translated into program code or software.
- d. **Testing** – This phase is the most respected in systems development. Testing each module in every simulation does not mean that the functionalities are in good condition; rather, it is important to test the entire system. It should include attempts to get the system to fail and to violate processes and security controls.

There are three (3) testing that should be performed: **unit testing**, **systems testing**, and **acceptance testing**.

- e. **Implementation** – In the new IS, **implementation is referred to as delivery**. This consists of **conversion and training**. Conversion **takes place when an operation in an organization switches from using an existing or old system to the new system**.
- f. **Production and Maintenance** – When the **new system is installed, it is said to be in production**. In this stage, the **system will be reviewed by both users and technical specialists** to check if it meets the original objectives. If the system is fine and meets all the objectives, **then it should be maintained**.

When the system is in production, the errors will be corrected, meet all the requirements, or improve processing efficiency. If there are some changes or new requirements in software or hardware, or **if there is an improvement of the process**, it is called **maintenance**.

### Tasks in the Development Process Life Cycle

The system developers and their clients share a common goal for developing the information systems that effectively support the objectives of the business process. In order to make sure the cost-effective, develop quality systems that will address the organization's business needs. The usual activities performed include the following:

- System conceptualization
- System requirements and benefits analysis
- Scope and adoption of the project
- Software requirement specifications
- Detailed and architectural design
- Software integration and testing
- System integration and testing
- Installation at site
- Site testing acceptance
- Training and documentation
- Implementation
- Maintenance

### Different Approaches in System Development

The different approaches to developing a system help for faster development and meeting the requirement. Here are the common approaches in system development:

- **Ad-Hoc Development** – This **approach relies on the skills and experience of the individual staff**. It only **refers to the small projects** at which the staff is skilled.
- **Waterfall Model** – This **method is the most common and the earliest structured system development** and is still widely used. This model consists of the following steps:
  - Systems analysis
  - System design
  - Programming
  - Testing
  - Deployment
  - Maintenance
- **Prototyping** – It is **developed on the assumption that it knows all the requirements** at the beginning of the project. When using this approach, the **developer only builds a simplified version of the proposed system** and presents it to the client. After presenting, the client will give feedback for revisions or adjustments based on the requirements.

Like the waterfall model, the prototyping also has various steps:

- Requirements definition – This is where the **client needs are gathered** for developing the software. The requirements of the system are defined in detail.
- Design – A **simple design of the software is created**, which gives a brief idea of the system to the user.
- Prototype creation – An **actual prototype is designed based on the information gathered** from the design phase.
- Assessment – A **proposed system is presented to the client for assessment**. The **client will then look for the strength and weaknesses** of the created software.
- Prototype refinement – If **the client is not satisfied, the software needs to be refined** according to the client's feedback and suggestion.
- System implementation – The **final system is developed based on the final prototype**. It is **already tested and deployed to production**. Then, it **undergoes routine maintenance**.
- **Spiral Model** – This approach is **designed to include the best features from the waterfall and the prototype models**.

It describes the process that is followed as the development takes place. These are the steps in a spiral model:

- Project objectives – Requirements are gathered from the customers, and the objectives are identified.
  - Risk assessment – During the second quadrant, all the possible solutions are evaluated to select the best possible solution. Then the risks associated with that solution are identified and resolved using the best possible strategy.
  - Production – The identified features are developed and verified through testing.
  - Planning and management – Customers evaluate the developed version of the software and check for improvements. They also manage the developed software while looking for possibilities of errors.
- **Agile development** – This approach focuses on the fast delivery of the working software by dividing a large project into a series of small subprojects, which will be completed in short periods using iteration and continuous feedback.

Since the project is divided, each team works on a small project. It helps minimize the overall risk and allows the project to adapt to changes more quickly. In addition, there is face-to-face communication over written documents, so it encourages the members to collaborate and make decisions quickly.

#### REFERENCES:

- Laudon, K. C., & Laudon, J. P. (2014). *Management information systems: Managing the digital firm* (13th Ed.). USA: Pearson Education Limited.
- Sousa, K. J., & Oz, E. (2014). *Management information systems* (7th Ed.). Stamford, USA: Cengage Learning.