UNIT – II

SYSTEM DEVELOPMENT LIFE CYCLE

System Development Life Cycle is a set of activities carried out by the systems analysts and system designers to develop an information system.

The concept of System Development Life Cycle (SDLC) model comes from the system life cycle which is defined as the period of time that starts with the conceptualization of a system and ends when the functioning of the system is over. In other words, SDLC models represent descriptive and diagrammatic model of a software product with a series of identifiable phases through which the software product goes during its lifetime.

The fundamental need for using a life cycle model is to provide a basis that helps in controlling the various activities required for developing and maintaining a system. In addition, it establishes a precedence ordering among the various activities and helps the development team in facilitating and understanding the activities involved in the project. The primary task of the development team, while developing a system, is to identify a suitable life cycle model and follow the model throughout the life cycle of the product. Adhering to the life cycle model helps them in developing the system in a systematic and disciplined manner.

Another need for using a life cycle model is that it helps in defining the entry and exit criteria for every phase of the system life cycle. This enables the system project managers to easily monitor the progress of the system development process and also helps in controlling and systematically organizing the various activities of its life cycle. Apart from these uses, SDLC also provides following advantages which help in supporting the need for the life cycle models in the life cycle of various system products:

• Enables effective communication: SDLC enhances the understanding of the system life cycle and provides a specific basis for the execution of software development life cycle.

• Facilitates process reuse: System development is a time-consuming and expensive activity; therefore, the system development team utilizes the existing processes for different projects.

• Effective: The reusable system life cycle models provide an effective means for system development. Some of the well-known SDLC models include the following:

• Classic waterfall model

The ***Waterfall Model*** was the first Process Model to be introduced. It is very simple to understand and use. In a Waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

### ***What are the different phases of the waterfall model?***

* ***Requirements***: The first phase involves understanding what needs to design and what is its function, purpose, etc. Here, the specifications of the input and output or the final product are studied and marked.
* ***System Design***: The requirement specifications from the first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The software code to be written in the next stage is created now.
* ***Implementation***: With inputs from system design, the system is first developed in small programs called units, which are integrated into the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.
* ***Integration and Testing***: All the units developed in the implementation phase are integrated into a system after testing of each unit. The software designed, needs to go through constant [***software testing***](https://www.toolsqa.com/software-testing/software-testing/) to find out if there are any [***flaws or errors***](https://www.toolsqa.com/software-testing/istqb/error-defect-failure/). Testing is done so that the client does not face any problem during the installation of the software.
* ***Deployment of System***: Once the [***functional and non-functional testing***](https://www.toolsqa.com/software-testing/functional-and-non-functional-testing/) is done, the product is deployed in the customer environment or released into the market.
* ***Maintenance***: This step occurs after installation, and involves making modifications to the system or an individual component to alter attributes or improve performance. These modifications arise either due to change requests initiated by the customer, or defects uncovered during live use of the system. The client is provided with regular maintenance and support for the developed software.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals is achieved for the previous phase and it is signed off, so the name "***Waterfall Model***".

## What are the pros and cons of the waterfall methodology?

### ***Advantages of the Waterfall Model***

* The advantage of waterfall development is that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.
* The waterfall model progresses through easily understandable and explainable phases and thus it is easy to use.
* It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
* In this model, phases are processed and completed one at a time and they do not overlap. The waterfall model works well for smaller projects where requirements are very well understood.

### ***Disadvantages of Waterfall Model***

* It is difficult to estimate time and cost for each phase of the development process in waterfall model.
* Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought-out in the concept stage.
* Not a good model for complex and object-oriented projects.
* Not suitable for the projects where requirements are at a moderate to high risk of changing.

• Prototyping model

The **prototype model** is a model where the prototype (trial product/trial design) is built in advance of the actual design. Then this prototype discusses with customers. If the customer is not agreeing with the prototype then it is revised by the expert/experts and again discusses with customers.

The following phases are giving a detailed description of the prototype model.

* **Problem identification** - This is the first and initial phase of the waterfall model. This phase is used to identify the problem.
* **Requirement analysis** - In this phase, the experts gather requirement and thinks about the solution of a problem.
* **System design** - System design is a blueprint of the solution. The process follows the steps i.e. first creates logical design and physical design.
* **Build a prototype** - It is a stage where a pre planned work converts into a realistic prototype. This prototype has all the necessary requirements of the proposed system.
* **System redesign** - After making the prototype, it has been discussed with the customers and finds the missing points, if any. If the proposed prototype is fulfilling all the necessities then its send to the engineers for implementing the system but if requires any changes then it redesigns as per requirements.
* **Implementation** - Implementation shows the final integration of the system.
* **Testing** - Testing is a process to validate the objective of the system through various steps. It is a mechanism that makes sure that the system is working properly or not.
* **Deployment** - After successful testing of the developed system, it is deployed to the concerned candidate or company for whom it has designed.
* **Maintenance** - After a successful deployment of the system at the clients' end, it is compulsory to maintain it from time to time, so that system can successfully run for a long period.

**Advantages of Prototype Model**

* The customer takes an active involvement during the whole process
* Since customer involvement begins with an initial stage which eliminates possible ambiguity about the need or interpretation of any feature.
* The test model decreases production costs and time because all the possible flaws are detected before system implementation.
* Missing feature or functionality or a requirement adjustment may be found during the assessment process and may be incorporated in the refined prototype.
* All the possible modifications are done on paper which saves wastage of resources and money.

**Disadvantages of Prototype Model**

Due to customer involvement begins with an initial stage on this protocol causes following disadvantages -

* The customer may adjust the end product specification which raises the complexity.
* Product delivery time can extend.

 • Spiral model

## What is Spiral Model?

**Spiral Model** is a risk-driven software development process model. It is a combination of waterfall model and iterative model. Spiral Model helps to adopt software development elements of multiple process models for the software project based on unique risk patterns ensuring efficient development process.

Each phase of spiral model in software engineering begins with a design goal and ends with the client reviewing the progress. The spiral model in software engineering was first mentioned by Barry Boehm in his 1986 paper.

The development process in Spiral model in SDLC, starts with a small set of requirement and goes through each development phase for those set of requirements. The software engineering team adds functionality for the additional requirement in every-increasing spirals until the application is ready for the production phase.

**Spiral Model Phases**

| **Spiral Model Phases** | **Activities performed during phase** |
| --- | --- |
| **Planning** | * It includes estimating the cost, schedule and resources for the iteration. It also involves understanding the system requirements for continuous communication between the system analyst and the customer
 |
| **Risk Analysis** | * Identification of potential risk is done while risk mitigation strategy is planned and finalized
 |
| **Engineering**  | * It includes testing, coding and deploying software at the customer site
 |
| **Evaluation**  | * Evaluation of software by the customer. Also, includes identifying and monitoring risks such as schedule slippage and cost overrun
 |

## When to use Spiral Model?

* A Spiral model in software engineering is used when project is large
* When releases are required to be frequent, spiral methodology is used
* When creation of a prototype is applicable
* When risk and costs evaluation is important
* Spiral methodology is useful for medium to high-risk projects
* When requirements are unclear and complex, Spiral model in [SDLC](https://www.guru99.com/software-development-life-cycle-tutorial.html) is useful
* When changes may require at any time
* When long term project commitment is not feasible due to changes in economic priorities

## Spiral Model Advantages and Disadvantages

| **Advantages** | **Disadvantages** |
| --- | --- |
| Additional functionality or changes can be done at a later stage | Risk of not meeting the schedule or budget |
| Cost estimation becomes easy as the prototype building is done in small fragments | Spiral development works best for large projects only also demands risk assessment expertise |
| Continuous or repeated development helps in risk management | For its smooth operation spiral model protocol needs to be followed strictly |
| Development is fast and features are added in a systematic way in Spiral development | Documentation is more as it has intermediate phases |
| There is always a space for customer feedback | Spiral software development is not advisable for smaller project, it might cost them a lot |

• Rapid Application Development (RAD) model

**RAD Model** or Rapid Application Development model is a software development process based on prototyping without any specific planning. In RAD model, there is less attention paid to the planning and more priority is given to the development tasks. It targets at developing software in a short span of time.

**When to use RAD Methodology?**

* When a system needs to be produced in a short span of time (2-3 months)
* When the requirements are known
* When the user will be involved all through the life cycle
* When technical risk is less
* When there is a necessity to create a system that can be modularized in 2-3 months of time
* When a budget is high enough to afford designers for modelling along with the cost of automated tools for code generation

**Rapid Application Development Advantages and Disadvantages**

| **Advantages of RAD Model** | **Disadvantages of RAD Model** |
| --- | --- |
| * Flexible and adaptable to changes
 | * It can’t be used for smaller projects
 |
| * It is useful when you have to reduce the overall project risk
 | * Not all application is compatible with RAD
 |
| * It is adaptable and flexible to changes
 | * When technical risk is high, it is not suitable
 |
| * It is easier to transfer deliverables as scripts, high-level abstractions and intermediate codes are used
 | * If developers are not committed to delivering software on time, RAD projects can fail
 |
| * Due to code generators and code reuse, there is a reduction of manual coding
 | * Reduced features due to time boxing, where features are pushed to a later version to finish a release in short period
 |
| * Due to prototyping in nature, there is a possibility of lesser defects
 | * Reduced scalability occurs because a RAD developed application begins as a prototype and evolves into a finished application
 |
| * Each phase in RAD delivers highest priority functionality to client
 | * Progress and problems accustomed are hard to track as such there is no documentation to demonstrate what has been done
 |
| * With less people, productivity can be increased in short time
 | * Requires highly skilled designers or developers
 |

• Incremental model

• V Model

• Build and fix model

• Evolutionary development model

• Synchronize and Stabilize Model

SDLC follows a phased approach for the development of the system. In the phased development process, the process that is used to develop information system is divided into phases. Each phase ends with a defined output. The division of the development process into phases lowers the overall cost of the development because each phase handles a different part of the entire software development process. In addition, the phased development process allows you to check the quality of the information system and progress of the development process at the end of each phase.

Phases of SDLC

A general process model consists the following phases:

 • Requirement analysis

• Software design

• Coding

• Testing

• Implementing and evaluating

Requirement Analysis

The requirement analysis phase determines the functionality of information system and constraints under which the system will work. The requirement analysis phase is divided into two activities, problem analysis and requirement specification. In problem analysis, the problem that will be solved by the information system is determined. After the problem is analysed, the requirements are specified in the requirements specification document. The requirements specification document specifies all the functional and non-functional requirements, input and output format of all the phases of the information system development process and the constraints under which the system will work.

Software Design

In the software design phase, solution to the problem that is analysed in the requirement analysis phase is planned. This phase is the most crucial phase as it affects all the later phases of the information system development process.

Software design involves specifying the methods that can be used for dealing with the problems that can occur during the system development process. Some of the problems that can occur in the system development process are as follows:

• Problem of scale: The problem of scaling arises because different sets of methods are used to develop a large software project as compared to the methods used to develop a small software project.

• Problem of quality: Problem of quality arises if the quality of software cannot be determined using single parameter and the parameters, on which the quality depends, are specific to a software project. The goal of a software project should be to satisfy the quality that is specified before developing the project process. Reliability of a software project determines the quality of the software and the unreliability of software is due to the defects in the software. A defect in software is defined as a problem that causes software crash or incorrect output.

• Problem of productivity: Productivity of a software project is determined by the manpower employed for developing the software. If the productivity of the manpower employed to develop software is high, the cost of the software will be low.

• Problem of cost: The cost of developing software includes the cost of resources involved in the software development process such as hardware and manpower. The cost of software is measured in terms of total number of persons-months spent in the software project including the cost of overheads such as hardware and tools. Persons-months are defined as the amount of work done by a person in one month.

• Problem of scheduling: The scheduling of a software project is affected by the market competition which specifies that the time involved in the development of software must be small as compared to the other software in the market. The software projects where the schedule and cost of projects exceed the predicted limits are categorized as runaway projects. To reduce the problem of unreliability and delays, organizations need to take advice from the consultants of the organizations to improve the performance of Information system of the organization.

• Problem of consistency: Problem of consistency is one of the major challenges that affect the software development process. The problem of consistency arises if a software project does not repeat the successful results of previous software projects and the consistency in the quality of the previous software projects. The main aim of an organization involved in software development is to develop high quality products consistently with high productivity.

Coding

 In the coding phase, the modules designed in the software design phase are implemented using a programming language. The purpose of this phase is to implement the modules in an optimized manner. In addition, the code should be simple to read and understand. An optimized code reduces the testing and maintenance effort. The output of the coding phase is the software that satisfies the information system requirements.

Testing

The purpose of the testing phase is to detect defects in the information system. This phase detects defects in all the previous phases of a system development life cycle. Testing of information system involves:

 • Unit testing: It involves testing of different components of the information system to ensure whether or not the components are functioning correctly.

• Module testing: It involves testing of modules to ensure whether or not the modules are functioning properly.

• Subsystem testing: It involves testing of subsystems to ensure whether or not the subsystems are functioning properly. A subsystem is a collection of dependent modules.

• System testing: It involves testing of the complete information system. System testing detects the errors that occur in the software due to interaction of subsystems and modules.

 • Acceptance testing: It involves testing of the software with the user-supplied, real-time data to demonstrate the use of software to the user.

Implementing and Evaluating the System

Implementation is a process of ensuring that the information system is operational and then allowing the users to use and evaluate it. Implementing a system also involves training the users to handle the system. The systems analyst needs to plan for a smooth conversion from the old system to the new one by converting files from old formats to new ones. The total evaluation of the system is done in this final phase. After the system is installed, it must be maintained for the following two reasons:

 • To find out system errors

• To enhance the system’s capabilities

Training

Training is a process of providing knowledge about the system to the users. Systems analyst engages in educational processes with the users through training. The trainer and the trainee determine training strategies. The systems analyst ensures that all the end users affected by the new information system are properly trained.

Possible training sources include the following:

• Vendors

 • Systems analysts

• External paid trainer

• In-house trainers

• Other system users

Guidelines for training are:

• Establishing measurable objectives

• Using appropriate training methods

• Selecting suitable training locations

• Employing understandable training materials

Conversion

Conversion is the method to replace the old system with the new one. A conversion plan includes a description of all the activities that must occur during the implementation of a new system. Following are the five strategies for converting the old information system into the new one:

• Direct changeover

• Parallel conversion

• Gradual (Phased) conversion

• Modular Prototype conversion

• Distributed conversion

Security of computer facilities, stored data and information generated are part of a successful conversion. The three interrelated aspects of security are:

• Physical Security

• Logical Security

• Behavioural Security

Evaluation

 Many different evaluation approaches are available for evaluating information systems including cost-benefit analysis, the revised decision evaluation approach and user involvement evaluation. The information system utility approach is a comprehensive and useful technique for evaluating and measuring the success of a developed system. Following are the utilities that are used to evaluate the system:

• Possession utility: It answers the questions who should receive the output.

• Form utility: It answers the question what kind of output is provided to the decision maker. • Place utility: It answers the question where the information is distributed.

• Time utility: It answers the question when the information is delivered.

• Actualization utility: It involves how the information is introduced and used by the decision maker. • Goal utility: It answers the question whether the output has value in obtaining the objectives of an organization.

MIS IN ORGANISATIONS

1. [**Decision making**](https://www.geektonight.com/what-is-mis/#decision-making)
2. [**Coordination among the department**](https://www.geektonight.com/what-is-mis/#coordination-among-the-department)
3. [**Finding out Problems**](https://www.geektonight.com/what-is-mis/#finding-out-problems)
4. [**Comparison of Business Performance**](https://www.geektonight.com/what-is-mis/#comparison-of-business-performance)
5. [**Strategies for an Organization**](https://www.geektonight.com/what-is-mis/#strategies-for-an-organization)

**Role of MIS in Business Organization with particular reference to Management Levels**

The type of information being utilized by each level of management is in accordance with the nature of jobs performed by the managers at their respective level. To facilitate the management decision making at all levels of company, the MIS must be integrated. MIS units are companywide. MIS is available for the Top management. The top management of company

**Data processing group’s responsibility**

* The data processing functions are data collection, manipulation, and storage as used to report and analyze business activities. The data processing system is oriented primarily to processing transactions for day-to-day operations.
* The transactions include sales orders, shipping orders, inventory orders, and payroll data. For most of these transactions, routine procedures can be established and carried out repetitively to do the processing required. The procedures become part of the data processing system.
* An MIS that function properly processes and analyses data to provide, in particular planning and control information that supports the decision-making role of management. A management information system (MIS) performs substantial functions beyond those of a data processing system.
* The MIS involves a man/machine system that provides information for managers to use as they perform their managerial functions of planning, organising, staffing, directing, and controlling. Such a system supports basic transaction processing as does a data processing system. It also provides information about the past, present, and future (forecasts) as each relates to the operations within the organisation and within its environment.

