* 1. **Overview of System Analysis and Design**

System development is systematic process which includes phases such as planning, analysis, design, deployment and maintenance. Here, in this, we will primarily focus on:

* System Analysis
* System Design

**System Analysis: -**

It is a process of collecting and interpreting facts, identifying the problems and decomposition of a system into its components. System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficient to accomplish their purpose.

**System Design: -**

It is a process of planning a new business system or replacing an existing system by defining its components or modules to satisfy the specific requirements. Before planning you need to understand the old system thoroughly and determine how we can operate the system efficiently.

**1.2 Information Systems and Its Types**

” An information system is a set of interrelated components that works together to collect, process, store and breakdown the information to support decision making. ”

Types Of Information System and Its Types: -

* Transaction Processing System
* Management Information System
* Decision Management System
* Experts System

1. Transaction Processing System (TPS):

Transaction Processing System are information system that processes data resulting from the occurrences of business transactionsTheir objectives are to provide transaction in order to update records and generate reports i.e to perform store keeping functionThe transaction is performed in two ways: Batching processing and Online transaction processing.

* Example: Bill system, payroll system, Stock control system.

1. Management Information System (MIS):

Management Information System is designed to take relatively raw data available through a Transaction Processing System and convert them into a summarized and aggregated form for the manager, usually in a report format. It reports tending to be used by middle management and operational supervisors.Many different types of report are produced in MIS. Some of the reports are a summary report, on-demand report, ad-hoc reports and an exception report.

* Example: Sales management systems, Human resource management system.

1. Decision Support System (DSS):

Decision Support System is an interactive information system that provides information, models and data manipulation tools to help in making the decision in a semi-structured and unstructured situation.Decision Support System comprises tools and techniques to help in gathering relevant information and analyze the options and alternatives, the end user is more involved in creating DSS than an MIS.

* Example: Financial planning systems, Bank loan management systems.

1. Experts System:

Experts systems include expertise in order to aid managers in diagnosing problems or in problem-solving. These systems are based on the principles of artificial intelligence research.Experts Systems is a knowledge-based information system. It uses its knowledge about a specify are to act as an expert consultant to users.Knowledgebase and software modules are the components of an expert system. These modules perform inference on the knowledge and offer answers to a user’s question.

**1.3 Stakeholders of Information Systems**

A stakeholder is any person who has an interest in an existing or proposed information system. Stakeholders can be technical or nontechnical workers. They may also include both internal and external workers. Information workers are those workers whose jobs involve the creation, collection, processing, distribution, and use of information. Knowledge workers are a subset of information workers whose responsibilities are based on a specialized body of knowledge.

* System Owner
* System User
* Project Manager
* System Analyst
* System Designer
* System Builders
* External Service Provider (ESP)

**Systems analyst**: -

A specialist who studies the problems and needs of an organization to determine how people, data, processes, and information technology can best accomplish improvements for the business. A programmer/analyst (or analyst/programmer) includes the responsibilities of both the computer programmer and the systems analyst. A business analyst focuses on only the non-technical aspects of systems analysis and design.

* By "Problems" that need solving, we mean:
  + **Problems** either real or anticipated, that require corrective action.
  + **Opportunities** to improve a situation despite the absence of complaints.
  + **Directives** to change a situation regardless of whether anyone has complained about the current situation.

**Skills needed by System Analyst: -**

* Working knowledge of information technology
* Computer programming experience and expertise
* General business knowledge
* General problem-solving skills
* Good interpersonal communication skills
* Good interpersonal relations skills
* Flexibility and adaptability
* Character and ethics

**1.4 Systems Development Life Cycle and**

**Life Cycle Models**

Definition: -

The Software Development Life Cycle (SDLC) refers to a methodology with clearly defined processes for creating high-quality software. in detail, the SDLC methodology focuses on the following phases of software development:

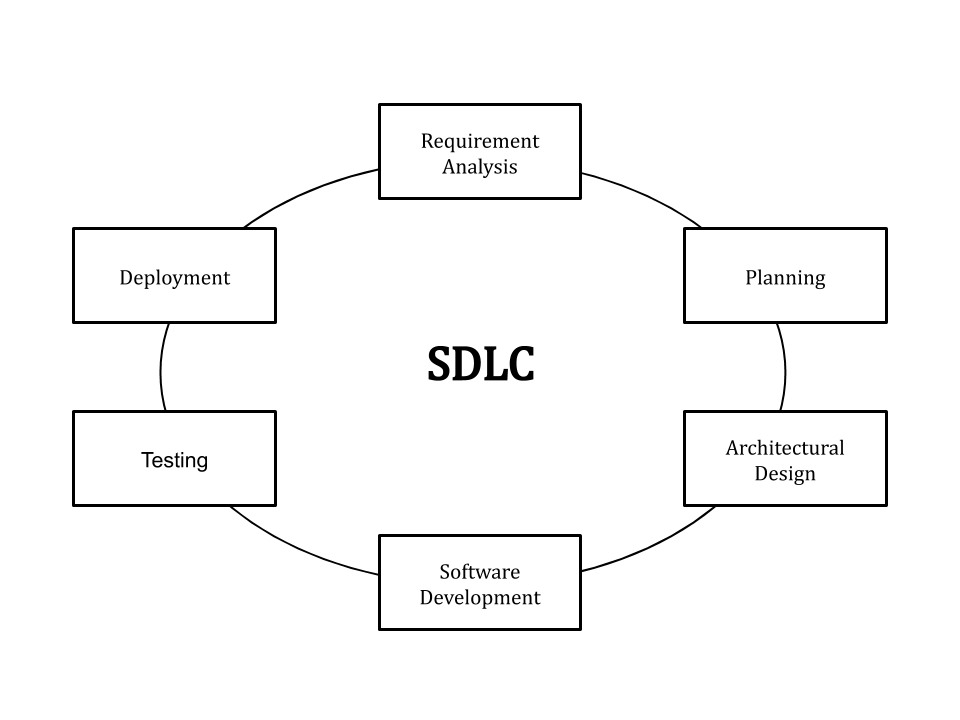
* Requirement analysis
* Planning
* Software design such as architectural design
* Software development
* Testing
* Deployment

Fig. System Development Life Cycle

|  |  |
| --- | --- |
| **Our Simplified System Development Process** | **General Problem-Solving Steps** |
| System initiation | Identify the problem. |
| System analysis | Analyze and understand the problem.  Identify solution requirements or expectations. |
| System design | Identify alternative solutions and choose the “best” course of action.  Design the chosen solution. |
| System construction | Construct the designed system |
| System implementation | Implement the chosen solution.  Evaluate the results. If the problem is not solved, return to step 1 or 2 as appropriate. |

SDLC MODELS: -

* Waterfall Models
* Spiral Models
* Prototype Models

**Waterfall Model: -**

Winston Royce introduced the Waterfall Model in 1970.This model has five phases: Requirements analysis and specification, design, implementation, and unit testing, integration and system testing, and operation and maintenance. The steps always follow in this order and do not overlap. The developer must complete every phase before the next phase begins. This model is named "**Waterfall Model**", because its diagrammatic representation resembles a cascade of waterfalls.

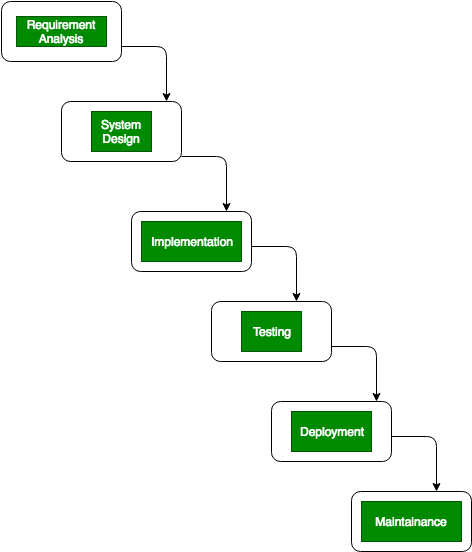
****

Fig. Waterfall Model

**1. Requirements analysis and specification phase:**

The aim of this phase is to understand the exact requirements of the customer and to document them properly. Both the customer and the software developer work together so as to document all the functions, performance, and interfacing requirement of the software. It describes the "what" of the system to be produced and not "how."In this phase, a large document called **Software Requirement Specification (SRS)** document is created which contained a detailed description of what the system will do in the common language.

**2. System Design:** This phase aims to transform the requirements gathered in the SRS into a suitable form which permits further coding in a programming language. It defines the overall software architecture together with high level and detailed design. All this work is documented as a Software Design Document (SDD).

**3. Implementation:** During this phase, design is implemented. If the SDD is complete, the implementation or coding phase proceeds smoothly, because all the information needed by software developers is contained in the SDD. During testing, the code is thoroughly examined and modified. Small modules are tested in isolation initially. After that these modules are tested by writing some overhead code to check the interaction between these modules and the flow of intermediate output.

**4. System Testing and Deployment:** This phase is highly crucial as the quality of the end product is determined by the effectiveness of the testing carried out. The better output will lead to satisfied customers, lower maintenance costs, and accurate results. Unit testing determines the efficiency of individual modules. However, in this phase, the modules are tested for their interactions with each other and with the system.

**5. Maintenance:** Maintenance is the task performed by every user once the software has been delivered to the customer, installed, and operational.

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| * Easy to explain to the users. * Structures approach. * Stages and activities are well defined. * Helps to plan and schedule the project. * Verification at each stage ensures early detection of errors/misunderstandings. * Each phase has specific deliverables. | * Assumes that the requirements of a system can be frozen. * Very difficult to go back to any stage after it is finished. * A little flexibility and adjusting the scope are difficult and expensive. * Costly and required more time, in addition to the detailed plan. |

**Spiral Model: -**

The spiral model, initially proposed by Boehm, is an evolutionary software process model that couples the iterative feature of prototyping with the controlled and systematic aspects of the linear sequential model. It implements the potential for rapid development of new versions of the software. Using the spiral model, the software is developed in a series of incremental releases. During the early iterations, the additional release may be a paper model or prototype. During later iterations, more and more complete versions of the engineered system are produced.

****

**Fig. Spiral Model**

**Objective setting:** Each cycle in the spiral starts with the identification of purpose for that cycle, the various alternatives that are possible for achieving the targets, and the constraints that exists. **Risk Assessment and reduction:** The next phase in the cycle is to calculate these various alternatives based on the goals and constraints. The focus of evaluation in this stage is located on the risk perception for the project.

**Development and validation:** The next phase is to develop strategies that resolve uncertainties and risks. This process may include activities such as benchmarking, simulation, and prototyping.

**Planning:** Finally, the next step is planned. The project is reviewed, and a choice made whether to continue with a further period of the spiral. If it is determined to keep, plans are drawn up for the next step of the project.

The development phase depends on the remaining risks. For example, if performance or user-interface risks are treated more essential than the program development risks, the next phase may be an evolutionary development that includes developing a more detailed prototype for solving the risks.

The **risk-driven** feature of the spiral model allows it to accommodate any mixture of a specification-oriented, prototype-oriented, simulation-oriented, or another type of approach. An essential element of the model is that each period of the spiral is completed by a review that includes all the products developed during that cycle, including plans for the next cycle. The spiral model works for development as well as enhancement projects.

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| * Estimates (i.e. budget, schedule, etc.) become more realistic as work progressed because important issues are discovered earlier. * Early involvement of developers. * Manages risks and develops the system into phases. | * High cost and time to reach the final product. * Needs special skills to evaluate the risks and assumptions. * Highly customized limiting re-usability |

**Prototype Model: -**

The prototype model requires that before carrying out the development of actual software, a working prototype of the system should be built. A prototype is a toy implementation of the system. A prototype usually turns out to be a very crude version of the actual system, possible exhibiting limited functional capabilities, low reliability, and inefficient performance as compared to actual software. In many instances, the client only has a general view of what is expected from the software product. In such a scenario where there is an absence of detailed information regarding the input to the system, the processing needs, and the output requirement, the prototyping model may be employed.

Steps of Prototype Model

1. Requirement Gathering and Analyst
2. Quick Decision
3. Build a Prototype
4. Assessment or User Evaluation
5. Prototype Refinement
6. Engineer Produc

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| * Reduced time and costs, but this can be a disadvantage if the developer loses time in developing the prototypes. * Improved and increased user involvement. | * Insufficient analysis. User confusion of prototype and finished system. * Developer misunderstanding of user objectives. * Excessive development time of the prototype. * It is costly to implement the prototypes |

**1.5 Introduction to Analysis and Design Tools**

Software analysis and design includes all activities, which help the transformation of requirement specification into implementation. Requirement specifications specify all functional and non-functional expectations from the software. These requirement specifications come in the shape of human readable and understandable documents, to which a computer has nothing to do.

Software analysis and design is the intermediate stage, which helps human-readable requirements to be transformed into actual code.

Let us see few analysis and design tools used by software designers:

* Data Flow Diagram
* Structure Chart
* Hippo Diagram
* Structure English
* Pseudo Code
* E-R diagram
* Decision Table