**UNIT II – SYSTEM DEVELOPMENT LIFE CYCLE**

**Source and Inspiration of a New System Development**

The source and inspiration for a new system development often come from identifying a specific need or problem that existing systems cannot adequately address. This could arise from technological advancements, user feedback, market demands, or gaps in current solutions. Inspiration may also come from innovations in related fields, emerging trends, or opportunities to improve efficiency, usability, and scalability. Ultimately, the goal is to create a system that solves real-world issues or enhances existing processes.

**Recognition and Need**

Recognition and Need phase is the initial stage where the necessity for a new system or an improvement to an existing one is identified.

1. **Recognition**: This involves realizing that there is a problem, inefficiency, or opportunity for enhancement in the current processes, often through stakeholder feedback, market trends, or performance analysis. It highlights the gap between the current and desired state.

2. **Need**: Once the problem or opportunity is recognized, the \*\*need\* for a solution is defined. This involves understanding and documenting the specific requirements, goals, and benefits the new system will provide. It sets the foundation for the rest of the SDLC by ensuring that the development aligns with real business objectives.

Together, they guide the decision-making process on whether to initiate a system development project.

**Linear Approach and Prototype Approach**

In system development, the Linear Approach and Prototype Approach are two distinct methodologies for building software systems:

1. **Linear Approach (Waterfall Model)**

- Description: The linear approach, often referred to as the Waterfall Model follows a sequential, step-by-step process for system development. Each phase must be completed before moving on to the next, with little room for revisiting previous phases once they are finished.

- Phases: Typical phases include Requirements Analysis, System Design, Implementation, Testing,Deployment, and Maintenance.

- Advantages:

* Well-structured and easy to manage.
* Clear milestones and deliverables for each phase.

- Disadvantages:

* Rigid, making it difficult to accommodate changes.
* Limited user feedback during development, leading to potential misunderstandings of requirements.

2. **Prototype Approach:**

- Description: The Prototype Approach is an iterative model where a working model (prototype) of the system is built quickly to demonstrate its functionality. This prototype is refined based on feedback until the final system is developed.

- Phases: Includes Initial Requirement Gathering, Quick Design, Prototype Development, User Evaluation, and Refinement.

- Advantages:

* + Allows for user involvement and feedback early in the process.
  + Better understanding of system requirements through visual and functional prototypes.

- Disadvantages:

* + Can lead to scope creep if not managed well.
  + Initial prototypes may be too simplistic or misinterpreted as the final product.

Both approaches are used based on the project's complexity, requirements, and flexibility needed during development.

**Role of System Analyst**

The system analyst is a person who is thoroughly aware of the system and guides the system development project by giving proper directions. He is an expert having technical and interpersonal skills to carry out development tasks required at each phase.

He pursues to match the objectives of information system with the organization goal.

* Defining and understanding the requirement of user through various Fact finding techniques.
* Prioritizing the requirements by obtaining user consensus.
* Gathering the facts or information and acquires the opinions of users.
* Maintains analysis and evaluation to arrive at appropriate system which is more user friendly.
* Suggests many flexible alternative solutions, pick the best solution, and quantify cost and benefits.
* Draw certain specifications which are easily understood by users and programmer in precise and detailed form.
* Implemented the logical design of system which must be modular.
* Plan the periodicity for evaluation after it has been used for some time, and modify the system as needed.

**System Development Life Cycle (SDLC)**

The **System Development Life Cycle (SDLC)** is a structured process for building and maintaining information systems. It typically involves the following phases:

**1. Planning**

Goal: Define the project’s scope, objectives, and feasibility.

Activities: Project planning, resource allocation, and risk analysis.

Output: Project plan, feasibility report.

2. **Requirement Analysis**

Goal: Gather and analyze the business and user needs for the system.

Activities: Stakeholder interviews, surveys, and documentation of functional and non-functional requirements.

Output: Requirement specification document.

3**. System Design**

Goal: Define the architecture, components, interfaces, and data models.

Activities: Designing system models (e.g., flowcharts, ER diagrams), database structure, and user interface.

Output: System design specifications.

**4. Development**

Goal: Build the actual system based on the design specifications.

Activities: Writing and compiling code, database creation, and system configuration.

Output: Working system modules or components.

**5. Testing**

Goal: Ensure the system works as expected and meets user requirements.

Activities: Unit testing, integration testing, system testing, and user acceptance testing (UAT).

Output: Tested and validated system.

**6. Deployment**

Goal: Release the system for use by end-users.

Activities: Installation, data migration, and user training.

Output: Live system in the production environment.

**7. Maintenance**

Goal: Ensure the system continues to operate effectively and adapt to any necessary changes.

Activities: Bug fixes, updates, system optimization, and performance monitoring.

Output: Stable and updated system.

These phases may vary slightly depending on the methodology used (e.g., Agile, Waterfall), but they provide a general framework for developing systems systematically.

