Software Development Life Cycle

SDLC

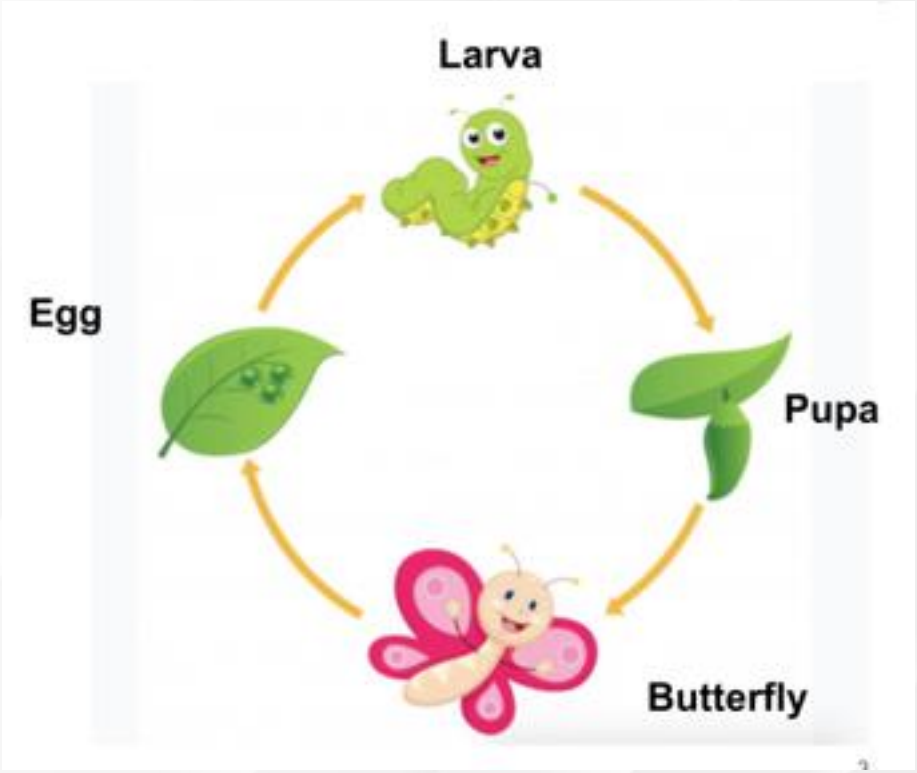
Module CS5002NP

Weekly Objective

- Demonstrate an understanding system development life cycle.
- Explain in brief the different phases of the SDLC.
- Understand the different type of methodologies.
- Understand Waterfall model, prototype model, spiral model.

Software Development Life Cycle

Life cycle of butterfly



Software Development Life Cycle

Software development life cycle is a systematic process of developing any software.

Basically, SDCL helps to:

- Develop the software
- Track the progress of software development
- Manage the changes appeared in the system
- Minimize the risk of system failure

Stages of software development

1. **Initiation**

When a project owner or sponsor identifies a need or an opportunity, they create a concept proposal for developing a software.

Stages of software development

2. **System Concept Development**

- The scope or boundary of the concepts is defined.
- Required different feasibility studies, system boundary documents, cost-benefit analysis, risk management plan, and so on..

Stages of software development

3. **Planning**

- Different project management plans and other planning documents such as Gantt chart are developed.
- Provides the basic for acquiring the resources needed to develop a solution.

Stages of software development

4. **Requirement Analysis**

- The requirements of the project are collected and analyzed.
- SRS(Software Requirement Specification) document is produced.

Stages of software development

5. **Design**

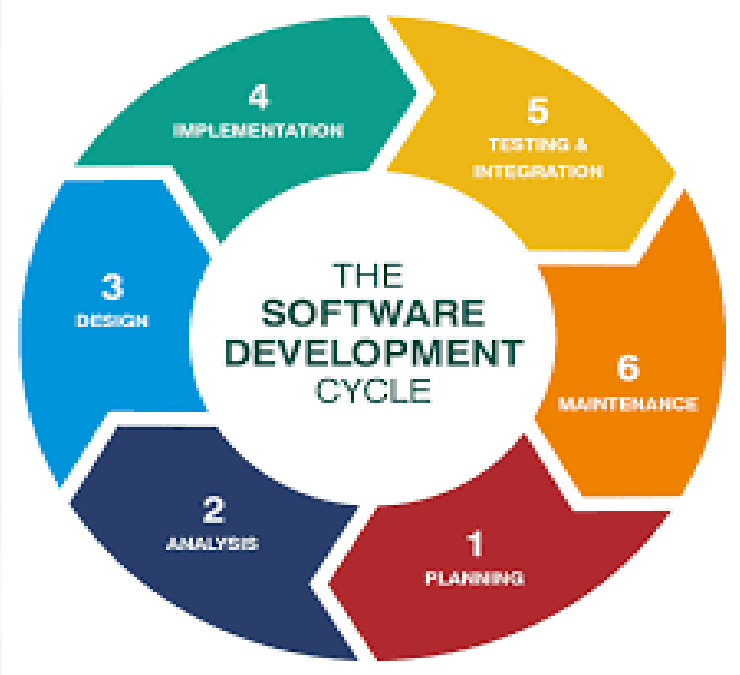
- The analyzed requirements are transformed into a complete system design.
- Determine the suitable architecture.

Stages of software development

6. **Development**

The system design is converted into a completed information system that includes:

- Acquiring and installing systems environment
- Creating and testing databases
- Preparing test case procedures
- Preparing test files
- Coding
- Compiling
- Refining programs
- Performing test readiness reviews



Stages of software development

7. **Integration and Testing**

- The developed programs are integrated into a system
- The quality assurance team and users ensures that the developed system is working as per the requirements specified in the functional requirements documents.

Stages of software development

8. Deployment and Maintenance

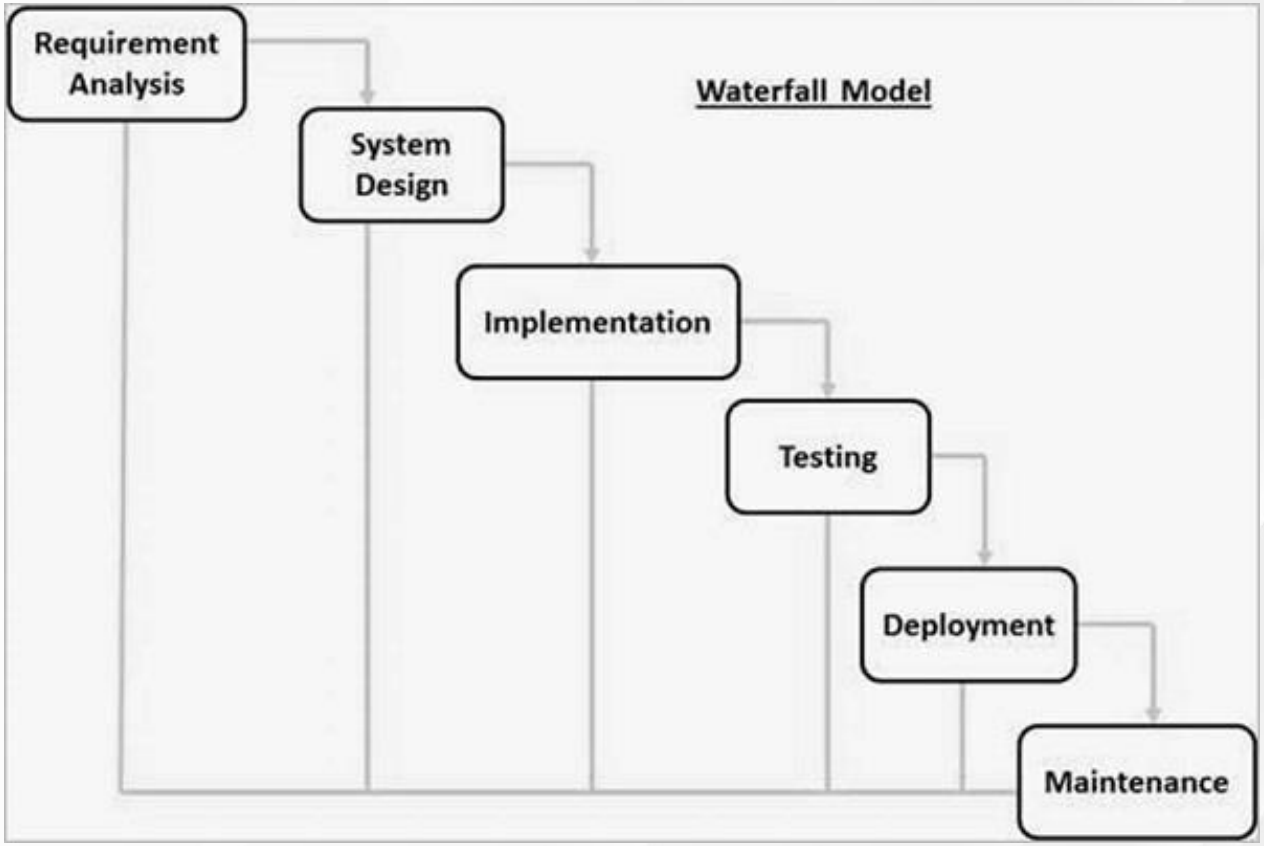
- The tested system is carried out for the implementation into a working environment.
- The problems identified in the integration and testing phase are also resolved.
- It is required to operate and provide service for the maintenance of the system.
- The updates and maintenance tasks regarding the information system are described.

Software Development Models

- **Waterfall Model**
- **Prototype Model**
- **Spiral Model**

Software Development Model

1. Waterfall Model



Waterfall Model

Advantages of waterfall model:

- This model is simple and easy to understand and 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. Phases do not overlap.
- Waterfall model works well for smaller projects where requirements are very well understood.

Waterfall Model

Disadvantages of 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.
- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing.

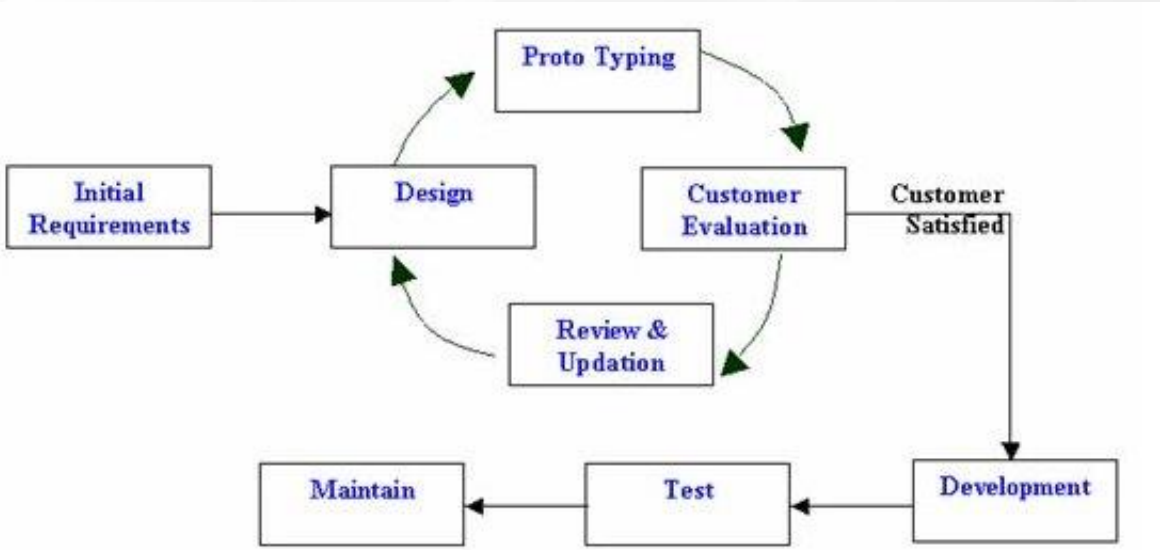
Waterfall Model

When to use the waterfall model:

- This model is used only when the requirements are very well known, clear and fixed.
- Product definition is stable.
- Technology is understood.
- There are no ambiguous requirements
- Ample resources with required expertise are available freely.
- The project is short.

Prototype Model

A prototype is an initial version of a software system that is used to demonstrate concepts, try out design options, and find out more about the problem and its possible solutions.



Proto Type Model

Prototype Model

Advantages of Prototype model:

- Users are actively involved in the development
- Since in this methodology a working model of the system is provided, the users get a better understanding of the system being developed.
- Errors can be detected much earlier.
- Quicker user feedback is available leading to better solutions.
- Missing functionality can be identified easily

Prototype Model

Disadvantages of Prototype model:

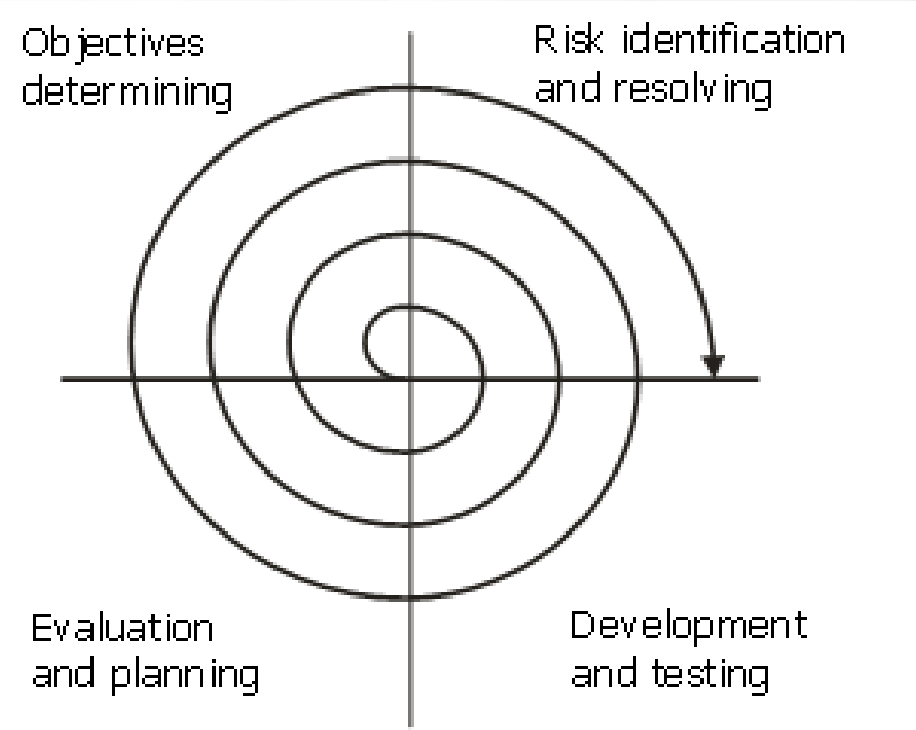
- Leads to implementing and then repairing way of building systems.
- Practically, this methodology may increase the complexity of the system as scope of the system may expand beyond original plans.
- Incomplete or inadequate problem analysis.

Prototype Model

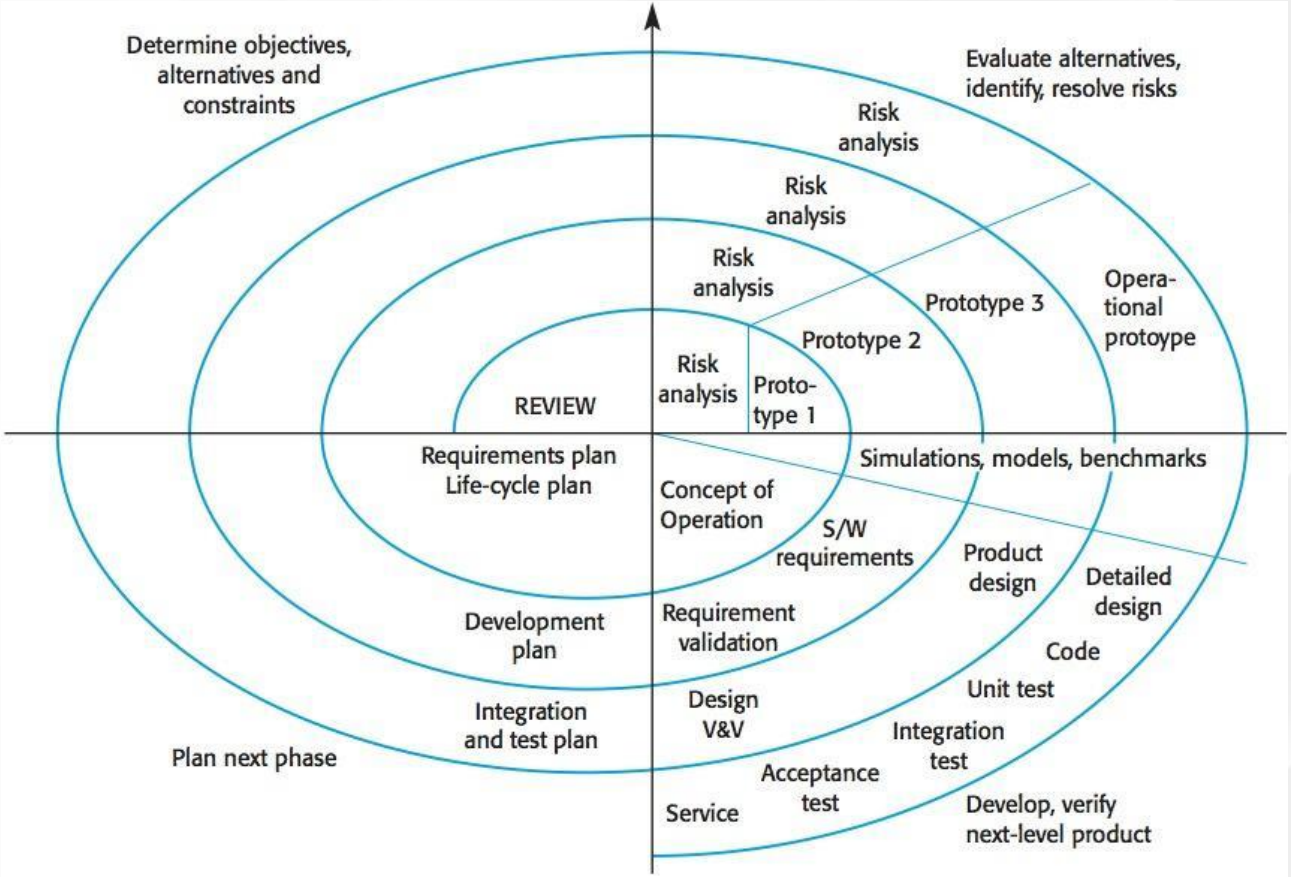
When to use Prototype model:

- Prototype model should be used when the desired system needs to have a lot of interaction with the end users.
- Typically, online systems, web interfaces have a very high amount of interaction with end users, are best suited for Prototype model. It might take a while for a system to be built that allows ease of use and needs minimal training for the end user.

Spiral Model



Spiral Model



Spiral Model

Advantages of Spiral model:

- High amount of risk analysis hence, avoidance of Risk is enhanced.
- Good for large and mission-critical projects.
- Strong approval and documentation control.

Disadvantages of Spiral model:

- Can be a costly model to use.
- Risk analysis requires highly specific expertise.
- Project's success is highly dependent on the risk analysis phase.
- Doesn't work well for smaller projects.

Spiral Model

When to use Spiral model:

- When costs and risk evaluation is important
- For medium to high-risk projects
- Users are unsure of their needs
- Requirements are complex
- New product line
- Significant changes are expected (research and exploration)

Key points

- Software development life cycle or SDLC is a systematic and clearly defined process of developing any software used by the software engineers and system developers.
- SDLC helps to develop the software from beginning to end, track the progress, manage the changes, and minimized the risks.
- The phases of SDLC: initiation, system concept development, planning, requirement analysis, designing, development, integration and testing, deployment and maintenance.
- Different type of software development models and its usage, advantages, and disadvantages.


ANY QUESTION ???