20AI501 SOFTWARE ENGINEERING AND TESTING METHODOLOGIES

Unit1: SOFTWARE PROCESS MODELS

Evolutionary process model

- ✓ Spiral Model
- ✓ WIN WIN Spiral Model
- ✓ Prototyping Model
- ✓ Concurrent development Model

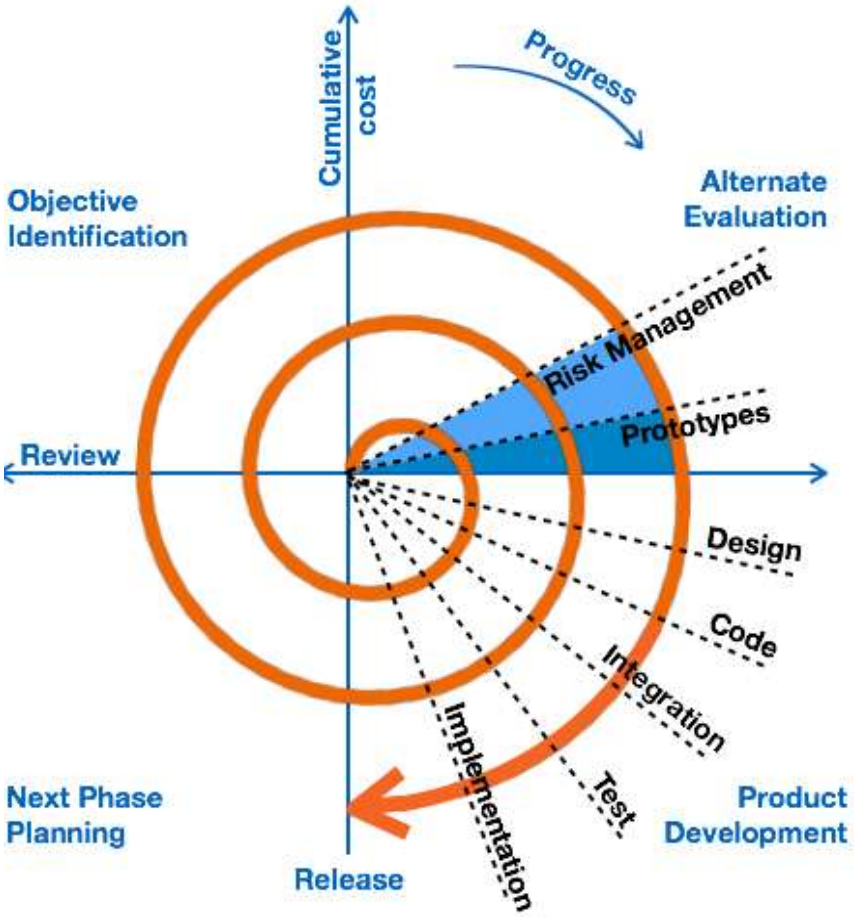
Spiral Model:

- **Spiral Model** is a **risk-driven** software development process 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.(waterfall)**
- It allows **incremental releases** of the product or incremental refinement through each iteration around the spiral.

Phases of Spiral Model

A software project repeatedly passes through these phases in iterations called **Spirals**.

- Identification
- Design
- Construct or Build
- Evaluation and Risk Analysis



Identification

- **Gathering the business requirements** in the baseline spiral.
- Understanding the system requirements by **continuous communication** between the customer and the system analyst.

Design

- **Conceptual design** in the baseline spiral
- Involves **architectural design, logical design of modules, physical product design and the final design** in the subsequent spirals.

Construct or Build

- **Production of the actual software product at every spiral**.
- In the baseline spiral, when the **product is just thought of and the design is being developed a POC** (Proof of Concept) is developed in this phase to get customer feedback.
- Then in the **subsequent spirals with higher clarity on requirements and design details** a working model of the software called **build** is produced with a version number.
- These builds are sent to the customer for feedback.

Evaluation and Risk Analysis

- Risk Analysis includes **identifying, estimating and monitoring the technical feasibility and management risks.**
- After testing the build, at the **end of first iteration, the customer evaluates the software** and provides **feedback**.

When to use Spiral Model?

- Deliverance is required to be frequent.
- Project is large
- Requirements are unclear and complex
- Changes may require at any time
- Large and high budget projects

Advantages:

- User gets a chance to **experiment partially developed system**.
- Reduces the error because the core modules get tested thoroughly.
- Useful for large and mission-critical projects.
- Changing requirements can be accommodated.
- Allows extensive use of prototypes.

- Development can be divided into smaller parts and the **risky parts can be developed earlier**.

Disadvantages:

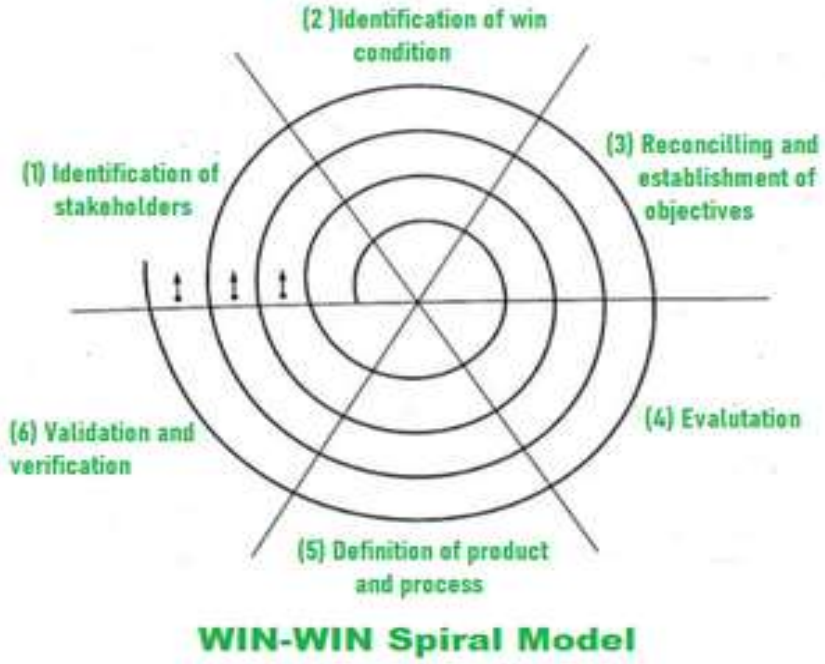
- Doesn't work well for smaller projects.
- Can be a costly model to use.
- Risk analysis needed highly particular expertise

WIN WIN Spiral Model

- For obtaining the project requirements, **customer communication** is very important and essential in the spiral model.
- At the beginning of **each pass of the spiral, the negotiation activities are carried** out in a WIN-WIN spiral model.
- In actual practice, the process of negotiation which simply means to compromise has to be faced by the customers and developers. When **both sides agree**, only then successful negotiation occurs. This is called the **WIN-WIN situation.**

Customer's win: Obtaining the system that **fulfill most of the requirements of customers**.
Developer's win: Getting the work done by **fulfilling the realistic requirements of customers** in a given deadline and achievable budgets.

WIN-WIN spiral model various activities:



WIN-WIN Spiral Model

1. **Identification** of stakeholders.
2. **Determination** of stakeholders to make great efforts to **achieve win condition**.
3. **Negotiations** of stakeholders struggling vigorously for **win condition**.
4. **Evaluation** of the process and product and then analyzing and resolving or reducing the risks to make it easy.
5. **Defining** the **next level of product and process** for proper working.
6. **Validation** of **process** and **product** definitions is must.
7. **Reviewing** of the product and giving the necessary and **important comments** on it.

Three anchor points in WIN-WIN spiral model:

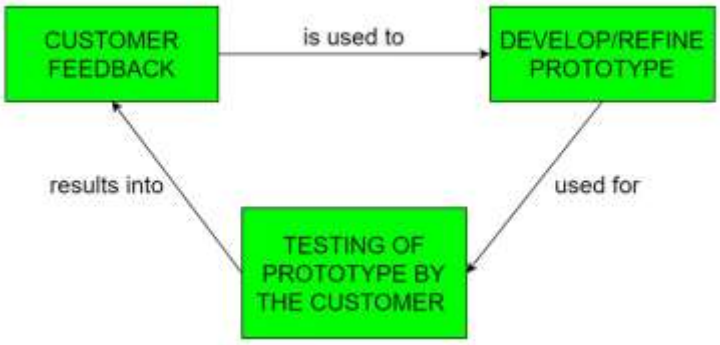
1. **Life Cycle Objective (LCO):** LCO defines the **objectives** that are **essential for activities** of software engineering.

2. **Life Cycle Architecture (LCA)**: LCA defines the **software architectures** that can **be produced with all the objectives** that are set.

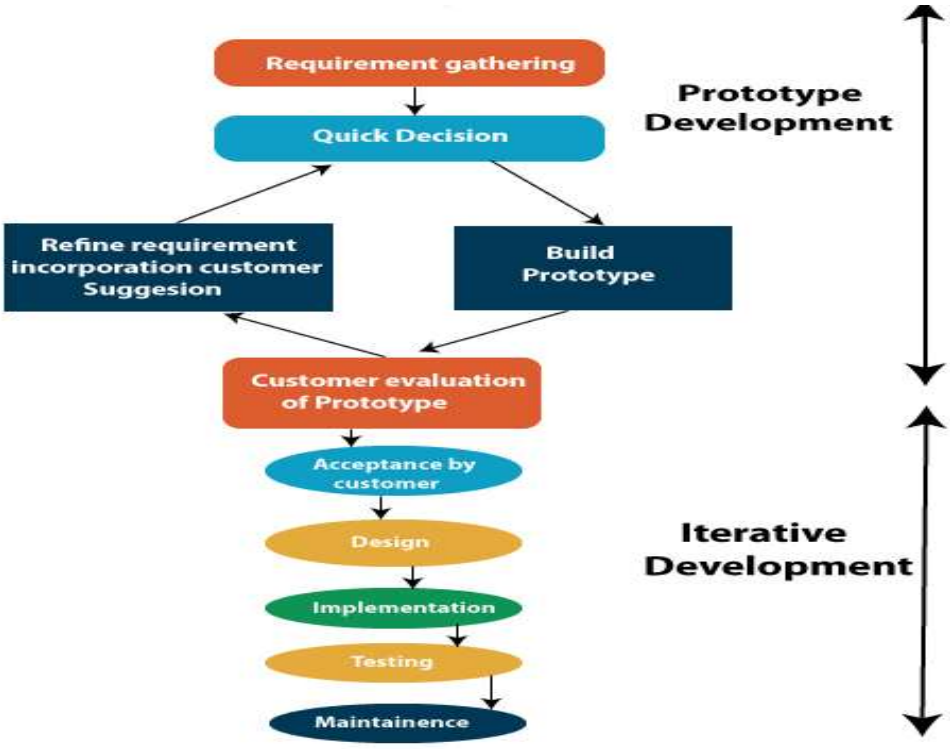
3. **Initial Operational Capability (IOC):** IOC represent the **software with all the initial required operational capabilities**.

Prototyping Model:

- Prototyping is defined as the **process of developing a working replication of a product or system that has to be engineered**.



- This model is used when the **customers do not know the exact project requirements** beforehand.
- A **prototype** of the end product is **first developed, tested and refined** as per **customer feedback repeatedly till a final acceptable prototype is achieved** which forms the basis for developing the final product.
- The **system is partially implemented** before or during the analysis phase thereby giving the **customers an opportunity to see the product early** in the life cycle.
- The process starts by **interviewing the customers** and developing the **incomplete high-level paper model.** This document is used to **build the initial prototype** supporting only the **basic functionality** as desired by the customer.
- Once the customer figures out the problems, the **prototype** is further **refined** to eliminate them.
- The **process continues** until the user **approves the prototype** and finds the **working model to be satisfactory.**



Four Types of Prototyping Model:

1. Rapid Throwaway prototypes
2. Evolutionary prototype
3. Incremental prototype
4. Extreme prototype

Rapid Throwaway Prototype

- Rapid throwaway is based on the **preliminary requirement**.
- It is quickly developed to **show how the requirement will look visually**.
- This technique offers a useful method of **exploring ideas and getting customer feedback** for each of them.
- In this method, a **developed prototype need not necessarily** be a part of the ultimately accepted **prototype**.
- **Customer feedback** helps in **preventing unnecessary design faults** and hence, **the final prototype developed is of better quality.**

Evolutionary Prototyping

- Prototype developed is **incrementally refined based on customer's feedback** until it is finally accepted.
- It helps you to **save time** as well as effort. That's because developing a prototype from scratch for every interaction of the process can sometimes be very frustrating.
- This model is **helpful for a project which uses a new technology that is not well understood**, **complex project** and the **requirement is not stable or not understood clearly at the initial stage**.

Incremental Prototyping

- In incremental Prototyping, the **final product is decimated into different small prototypes** and developed **individually**.
- Eventually, the **different prototypes are merged** into a single **product**.
- It's a very efficient approach that **reduces the complexity** of the development process
- The time interval between the **project's beginning and final delivery is reduced** because all parts of the system are prototyped and tested simultaneously.

Extreme Prototyping:

- Mostly used for **web development**.

Three sequential phases:

1. Basic prototype with all the **existing page** is present in the **HTML** format.
2. **Functional screens** are made with a simulated data process using a prototype **services layer**.
3. The **services** are implemented and integrated into the **final prototype**.

When to use Prototyping Model:

- When the requirements of the product are not clearly understood or changing quickly.
- Used for developing user interfaces, high technology software-intensive systems, and systems with complex algorithms and interfaces.

Advantages

- The customers get to see the partial product early in the life cycle.
- New requirements can be easily accommodated.
- Missing functionalities can be easily figured out.
- Errors can be detected much earlier.
- The developed prototype can be reused
- Flexibility in design

Disadvantages

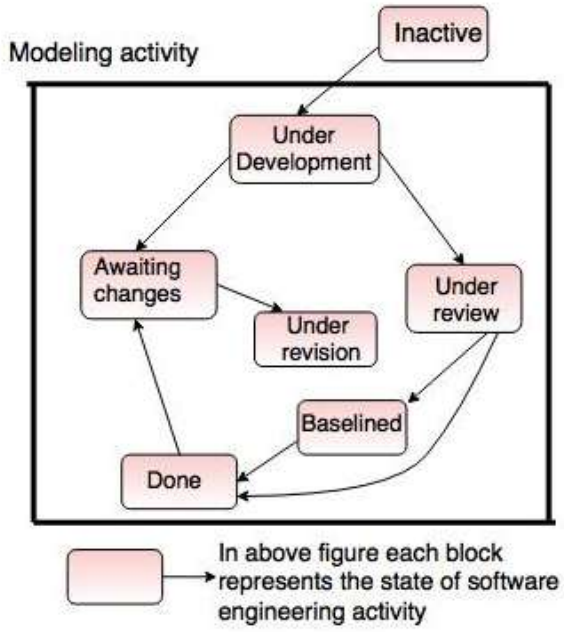
- Costly w.r.t time as well as money.
- The cost of developing a prototype is a total waste as the prototype is ultimately thrown away.
- It is very difficult for developers to accommodate all the changes demanded by the customer.
- After seeing an early prototype, the customers sometimes demand the actual product to be delivered soon.
- Developers in a hurry to build prototypes may end up with sub-optimal solutions.
- The customer might lose interest in the product if he/she is not satisfied with the initial prototype.

Concurrent Model:

- The concurrent development model, sometimes called **concurrent engineering**, can be represented **schematically as a series of framework activities, actions of tasks, and their associated states.**
- The concurrent model is often more appropriate where **different engineering teams** are involved.

Transitions:

- The communication activity has completed in the first iteration and exits in the awaiting changes state.
- The modeling activity completed its initial communication and then go to the underdevelopment state.
- If the customer specifies the change in the requirement, then the modeling activity moves from the under development state into the awaiting change state.
- The concurrent process model activities moving from one state to another state.



Advantages

- Applicable to all types of software development processes.
- Easy for understanding and use.
- Gives immediate feedback from testing.
- Provides an accurate picture of the current state of a project.

Disadvantages

- Needs Better communication between the team members.
- Requires to remember the status of the different activities.