Object Oriented Software Engineering (OOSE)

22CS017

SDLC Models

Index

Evolutionary Model

CHITKARA UNIVERSITY

- **Evolutionary model** is a combination of **Iterative and Incremental model** of software development life cycle. Delivering your system in a big bang release, delivering it in incremental process over time is the action done in this model.
- Some initial requirements and architecture envisioning need to be done. It is better for software products that have their feature sets redefined during development because of user feedback and other factors.
- The Evolutionary development model divides the development cycle into smaller, incremental waterfall models in which users are able to get access to the product at the end of each cycle.
- Feedback is provided by the users on the product for the planning stage of the next cycle and the development team responds, often by changing the product, plan or process.
- Therefore, the software product evolves with time. All the models have the disadvantage that the duration of time from start of the project to the delivery time of a solution is very high. Evolutionary model solves this problem in a different approach.

Evolutionary Model

CHITKARA UNIVERSITY

- Evolutionary model suggests breaking down of work into smaller chunks, prioritizing them and then delivering those chunks to the customer one by one.
- The number of chunks is huge and is the number of deliveries made to the customer. The main advantage is that the customer's confidence increases as he constantly gets quantifiable goods or services from the beginning of the project to verify and validate his requirements.
- The model allows for changing requirements as well as all work is broken down into maintainable work chunks.

Application of Evolutionary Model

1. It is used in large projects where you can easily find modules for incremental implementation. Evolutionary model is commonly used when the customer wants to start using the core features instead of waiting for the full software.
2. Evolutionary model is also used in object oriented software development because the system can be easily portioned into units in terms of objects.

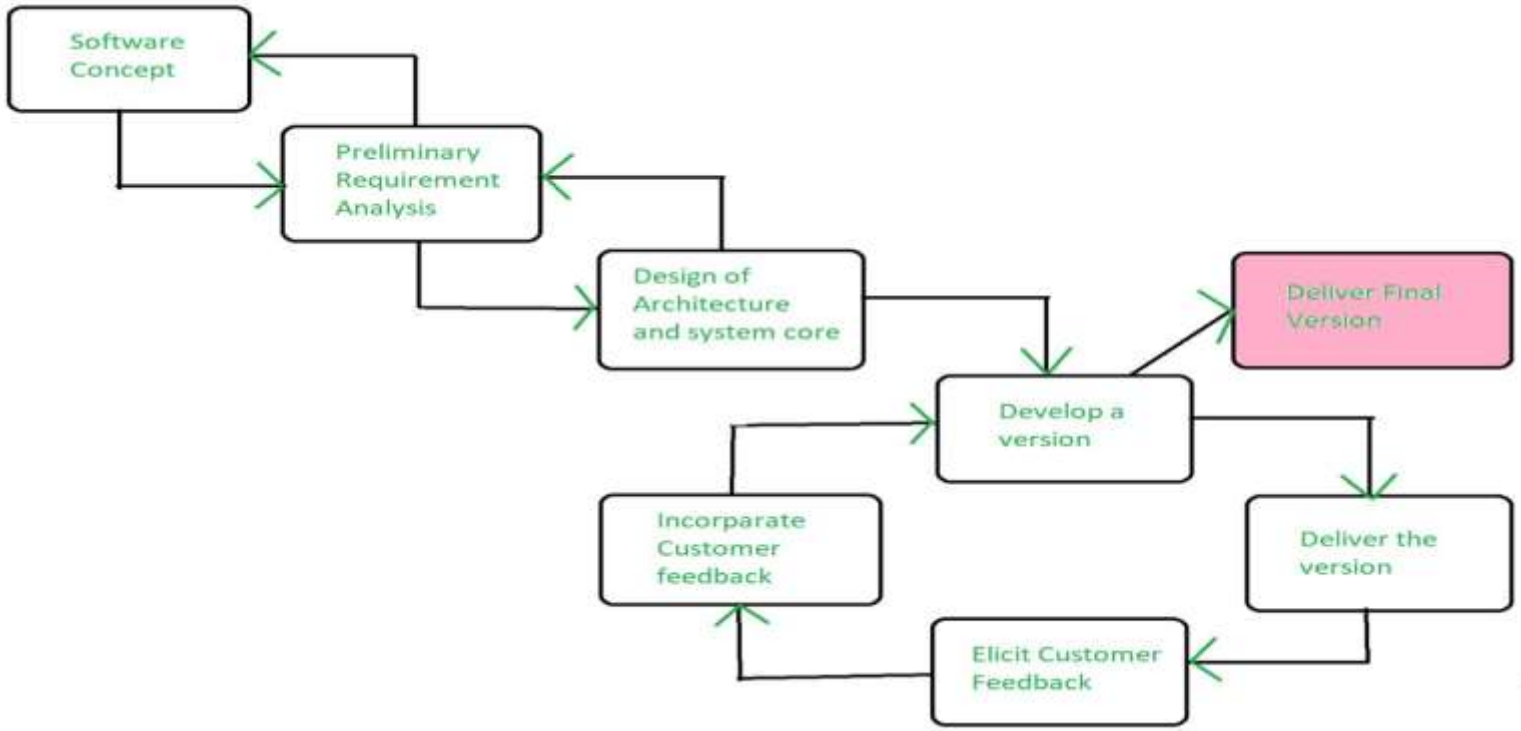

Figure 1: - Evolutionary Model

Iterative (Water Fall)Model

CHITKARA UNIVERSITY

- The classical waterfall model is hard to use. So, the Iterative waterfall model can be thought of as incorporating the necessary changes to the classical waterfall model to make it usable in practical software development projects. It is almost the same as the classical waterfall model except some changes are made to increase the efficiency of the software development.

- The iterative waterfall model provides feedback paths from every phase to its preceding phases, which is the main difference from the classical waterfall model. When errors are detected at some later phase, these feedback paths allow for correcting errors committed by programmers during some phase.

- The feedback paths allow the phase to be reworked in which errors are committed and these changes are reflected in the later phases. But, there is no feedback path to the stage – feasibility study, because once a project has been taken, does not give up the project easily.

Iterative (Water Fall)Model

CHITKARA UNIVERSITY

- It is good to detect errors in the same phase in which they are committed. It reduces the effort and time required to correct the errors.

- The Iterative Waterfall Model is a software development approach that combines the sequential steps of the traditional Waterfall Model with the flexibility of iterative design. It allows for improvements and changes to be made at each stage of the development process, instead of waiting until the end of the project.

- **Real-life example:** Iterative Waterfall Model could be building a new website for a small business. The process might look like this:

Iterative (Water Fall)Model

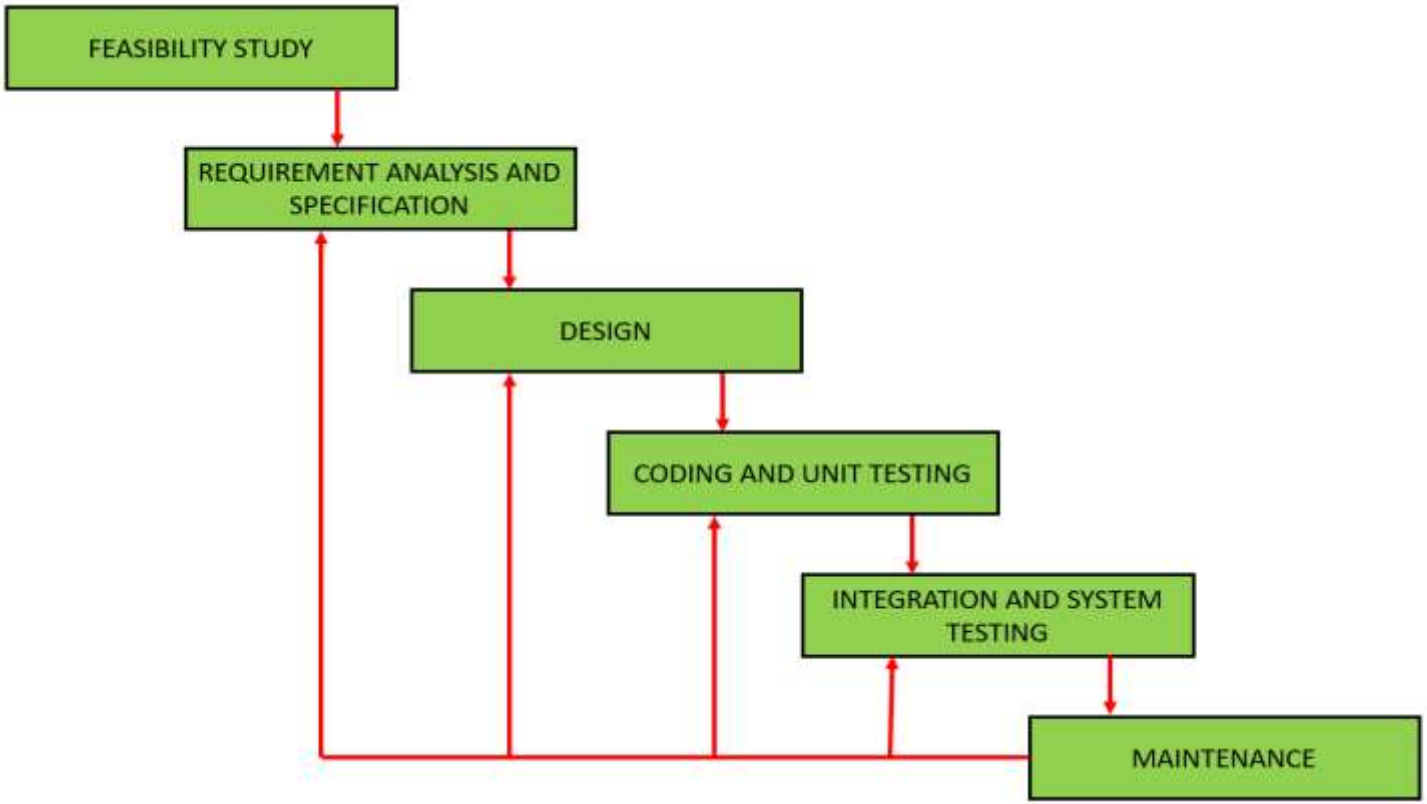

Figure 2: - Iterative(Water Fall) Model

Incremental Process Model

CHITKARA UNIVERSITY

- The incremental process model is also known as the Successive version model.
- First, a simple working system implementing only a few basic features is built and then that is delivered to the customer. Then thereafter many successive iterations/ versions are implemented and delivered to the customer until the desired system is released. A, B, and C are modules of Software Products that are incrementally developed and delivered.
- **Life cycle activities:** Requirements of Software are first broken down into several modules that can be incrementally constructed and delivered. At any time, the plan is made just for the next increment and not for any kind of long-term plan. Therefore, it is easier to modify the version as per the need of the customer. The Development Team first undertakes to develop core features (these do not need services from other features) of the system.
- Once the core features are fully developed, then these are refined to increase levels of capabilities by adding new functions in Successive versions. Each incremental version is usually developed using an iterative waterfall model of development.

Incremental Process Model

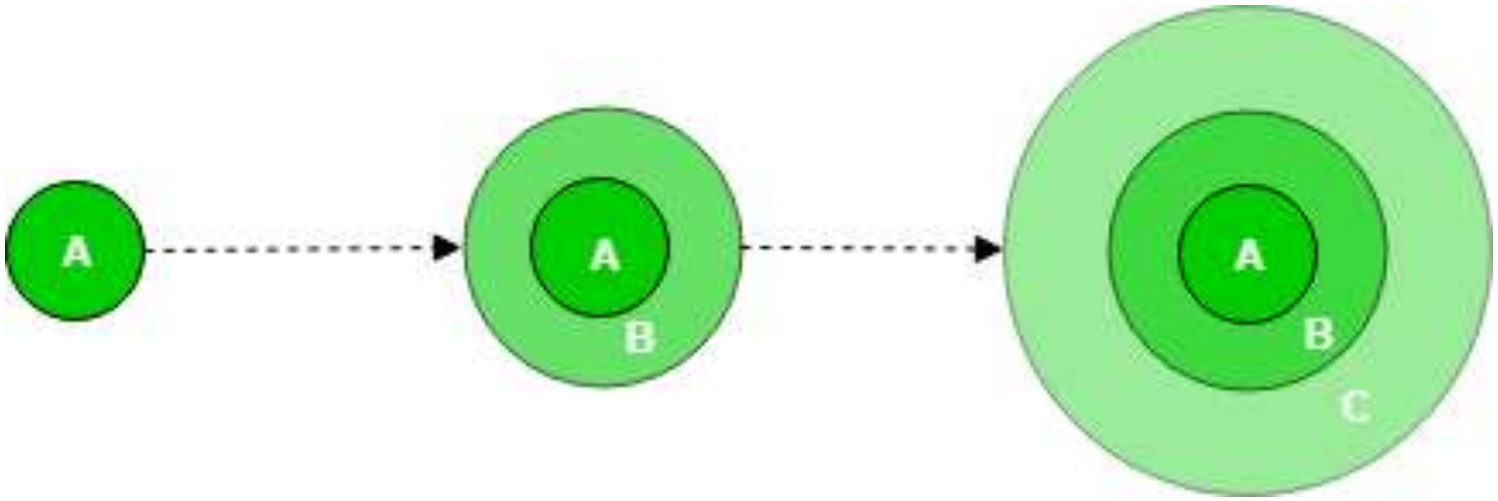

Figure 3 : - Incremental Process Model

Increment & iterative model

Increment development

Iterative development

Figure 4 : - Comparison Iterative and Incremental Process Model

INCREMENT & ITERATIVE MODEL

CHITKARA UNIVERSITY

- **ADVANTAGES:**
  - It is easier to test and debug during a smaller iteration.
  - In this model customer can respond to each built.
  - This model is less costly compared to others
  - Initial product delivery is faster.

- **DISADVANTAGES:**
  - Needs good planning and design.
  - Needs a clear and complete definition of the whole system before it can be broken down and built incrementally.
  - More management attention is required.

The Spiral Model

CHITKARA UNIVERSITY

- The Spiral Model is a **Software Development Life Cycle (SDLC)** model that provides a systematic and iterative approach to software development. In its diagrammatic representation, looks like a spiral with many loops. The exact number of loops of the spiral is unknown and can vary from project to project.
- Each loop of the spiral is called a **Phase of the** software development process.
  1. The exact number of phases needed to develop the product can be varied by the project manager depending upon the project risks.
  2. As the project manager dynamically determines the number of phases, the project manager has an important role in developing a product using the spiral model.
  3. It is based on the idea of a spiral, with each iteration of the spiral representing a complete software development cycle, from requirements gathering and analysis to design, implementation, testing, and maintenance.

The Spiral Model

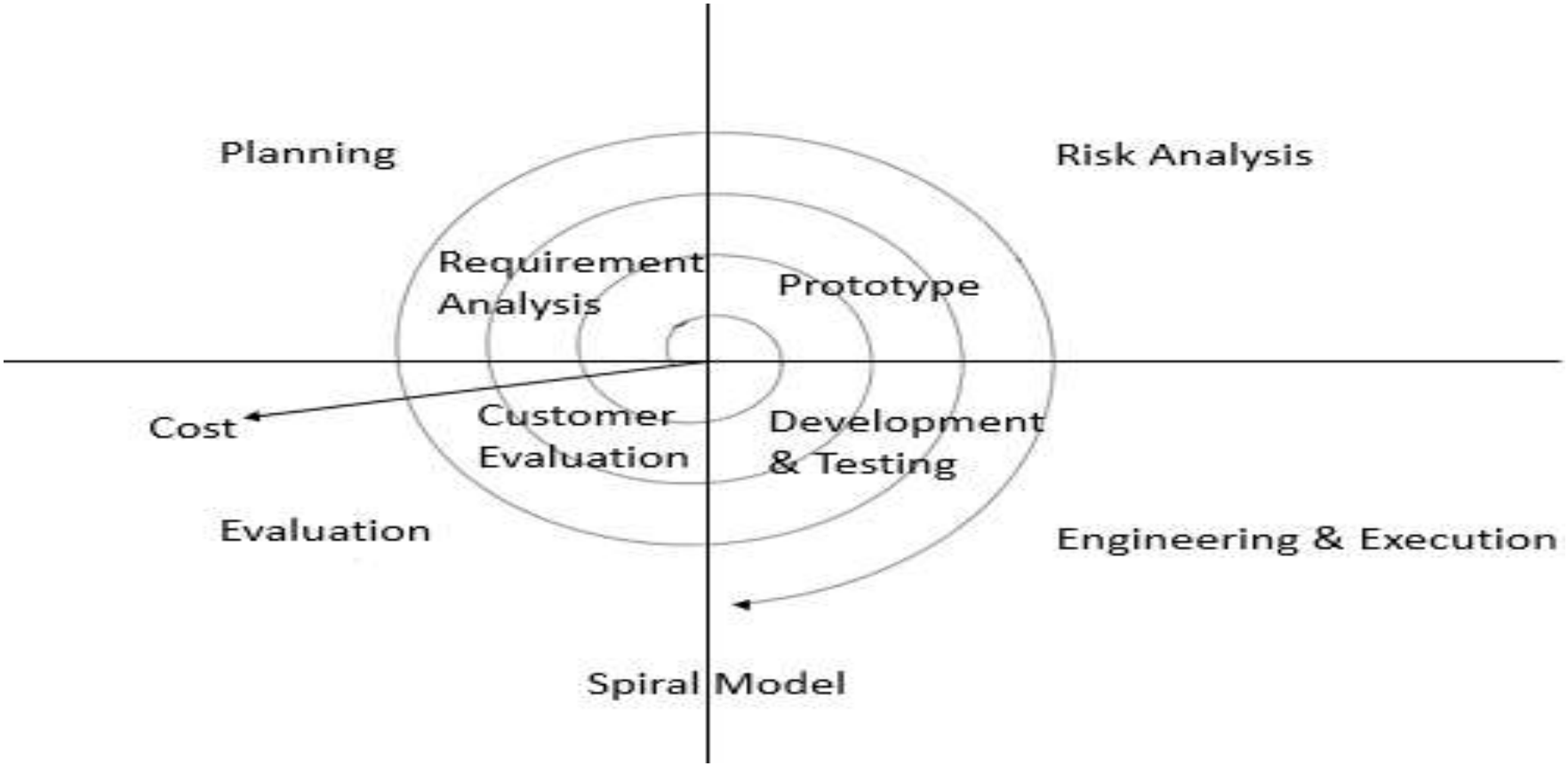

Figure 5 : The Spiral Model

The Spiral Model

CHITKARA UNIVERSITY

What Are the Phases of Spiral Model?

The Spiral Model is a risk-driven model, meaning that the focus is on managing risk through multiple iterations of the software development process. It consists of the following phases:

1. Planning

The first phase of the Spiral Model is the planning phase, where the scope of the project is determined and a plan is created for the next iteration of the spiral.

2. Risk Analysis

In the risk analysis phase, the risks associated with the project are identified and evaluated.

3. Engineering

In the engineering phase, the software is developed based on the requirements gathered in the previous iteration.

4. Evaluation

In the evaluation phase, the software is evaluated to determine if it meets the customer's requirements and if it is of high quality.

The Spiral Model

CHITKARA UNIVERSITY

ADVANTAGES:

- **Risk Handling:** The projects with many unknown risks that occur as the development proceeds, in that case, Spiral Model is the best development model to follow due to the risk analysis and risk handling at every phase.
- **Good for large projects:** It is recommended to use the Spiral Model in large and complex projects.
- **Flexibility in Requirements:** Change requests in the Requirements at a later phase can be incorporated accurately by using this model.
- **Customer Satisfaction:** Customers can see the development of the product at the early phase of the software development and thus, they habituated with the system by using it before completion of the total product.
- **Iterative and Incremental Approach:** The Spiral Model provides an iterative and incremental approach to software development, allowing for flexibility and adaptability in response to changing requirements or unexpected events.
- **Emphasis on Risk Management:** The Spiral Model places a strong emphasis on risk management, which helps to minimize the impact of uncertainty and risk on the software development process.
- **Improved Communication:** The Spiral Model provides for regular evaluations and reviews, which can improve communication between the customer and the development team.
- **Improved Quality:** The Spiral Model allows for multiple iterations of the software development process, which can result in improved software quality and reliability.

The Spiral Model

CHITKARA UNIVERSITY

DISADVANTAGES:

- **Complex:** The Spiral Model is much more complex than other SDLC models.
- **Expensive:** Spiral Model is not suitable for small projects as it is expensive.
- **Too much dependability on Risk Analysis:** The successful completion of the project is very much dependent on Risk Analysis. Without very highly experienced experts, it is going to be a failure to develop a project using this model.
- **Difficulty in time management:** As the number of phases is unknown at the start of the project, time estimation is very difficult.
- **Complexity:** The Spiral Model can be complex, as it involves multiple iterations of the software development process.
- **Time-Consuming:** The Spiral Model can be time-consuming, as it requires multiple evaluations and reviews.
- **Resource Intensive:** The Spiral Model can be resource-intensive, as it requires a significant investment in planning, risk analysis, and evaluations.

Prototype Model

CHITKARA UNIVERSITY

- Prototyping is defined as the process of developing a working replication of a product or system that has to be engineered. It offers a small-scale facsimile of the end product and is used for obtaining customer feedback. The Prototyping concept is described below:
- The Prototyping Model is one of the most popularly used Software Development Life Cycle Models (SDLC models). This model is used when the customers do not know the exact project requirements beforehand. In this model, 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.
- In this process model, 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.

Prototype Model

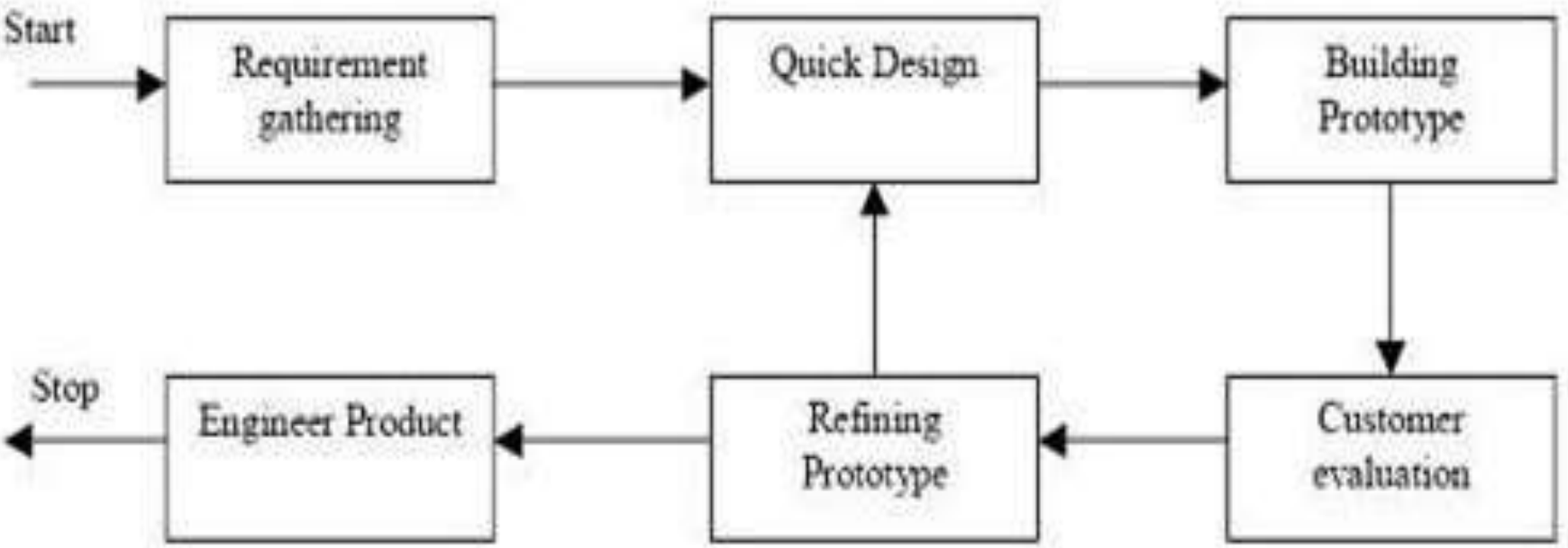

Figure 6 : The Prototype Model

Steps to build Prototype Model

CHITKARA UNIVERSITY

Step 1: Requirement Gathering and Analysis: This is the initial step in designing a prototype model. In this phase, users are asked about what they expect or what they want from the system.

Step 2: Quick Design: This is the second step in Prototyping Model. This model covers the basic design of the requirement through which a quick overview can be easily described.

Step 3: Build a Prototype: This step helps in building an actual prototype from the knowledge gained from prototype design.

Step 4: Initial User Evaluation: This step describes the preliminary testing where the investigation of the performance model occurs, as the customer will tell the strength and weaknesses of the design, which was sent to the developer.

Step 5: Refining Prototype: If any feedback is given by the user, then improving the client's response to feedback and suggestions, the final system is approved.

Step 6: Implement Product and Maintain: This is the final step in the phase of the Prototyping Model where the final system is tested and distributed to production, here program is run regularly to prevent failures.

Prototype model

CHITKARA UNIVERSITY

ADVANTAGES:

- The customers get to see the partial product early in the life cycle. This ensures a greater level of customer satisfaction and comfort.
- New requirements can be easily accommodated as there is scope for refinement.
- Missing functionalities can be easily figured out.
- Errors can be detected much earlier thereby saving a lot of effort and cost, besides enhancing the quality of the software.
- The developed prototype can be reused by the developer for more complicated projects in the future.
- Flexibility in design.
- Early feedback from customers and stakeholders can help guide the development process and ensure that the final product meets their needs and expectations.
- Prototyping can be used to test and validate design decisions, allowing for adjustments to be made before significant resources are invested in development.
- Prototyping can help reduce the risk of project failure by identifying potential issues and addressing them early in the process.
- Prototyping can facilitate communication and collaboration among team members and stakeholders, improving overall project efficiency and effectiveness.
- Prototyping can help bridge the gap between technical and non-technical stakeholders by providing a tangible representation of the product.

Prototype model

CHITKARA UNIVERSITY

DISADVANTAGES:

- Costly with respect to time as well as money.
- There may be too much variation in requirements each time the prototype is evaluated by the customer.
- Poor Documentation due to continuously changing customer requirements.
- It is very difficult for developers to accommodate all the changes demanded by the customer.
- There is uncertainty in determining the number of iterations that would be required before the prototype is finally accepted 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.
- The focus on prototype development may shift the focus away from the final product, leading to delays in the development process.
- The prototype may give a false sense of completion, leading to the premature release of the product.
- The prototype may not consider technical feasibility and scalability issues that can arise during the final product development.
- The prototype may be developed using different tools and technologies, leading to additional training and maintenance costs.
- The prototype may not reflect the actual business requirements of the customer, leading to dissatisfaction with the final product.

Practice Questions

CHITKARA UNIVERSITY

- **What is the importance of the SDLC process?**
- **Explain the phases in a typical SDLC process briefly.**
- **How does the Waterfall Methodology differ from other SDLC methodologies?**
- **Can you explain each stage of the waterfall methodology?**
- **What are some of the advantages and disadvantages of using the Spiral model?**
- **Is it possible to go back to an earlier phase once the development team moves into the later phases? If yes, then how?**
- **Do all projects require the same level of verification during each phase of the Waterfall Methodology?**

Bibliography

- https://www.tutorialspoint.com/software_engineering/
- https://courses.cs.vt.edu/csonline/SE/Lessons/Qualities/index.html
- https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm
- https://www.geeksforgeeks.org/software-engineering-spiral-model/
- https://www.geeksforgeeks.org/software-engineering-sdlc-v-model/


happy
LEARNING