

SOFTWARE ENGINEERING-I

BY

SAID UL ABRAR

LECTURER IN ICS/IT

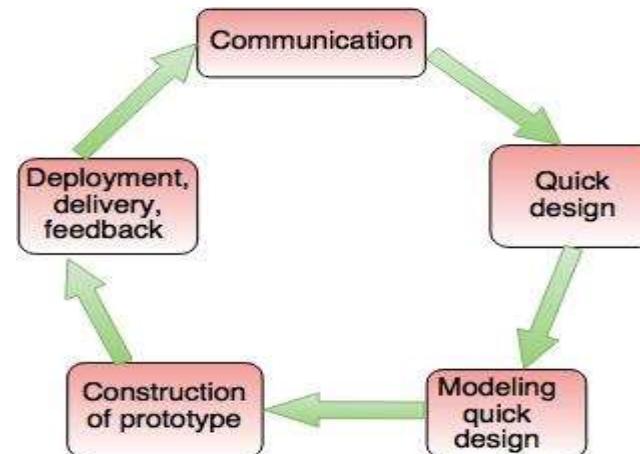
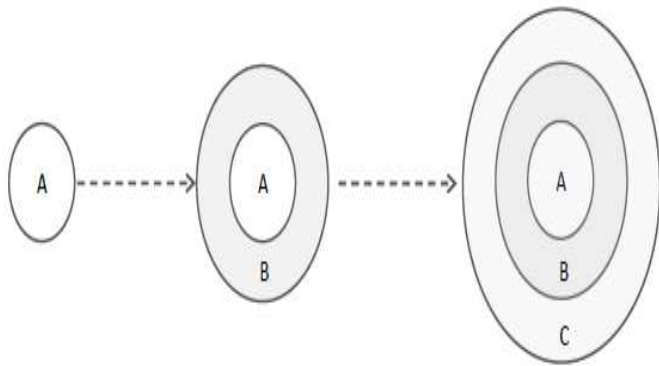
THE UNIVERSITY OF AGRICULTURE PESHAWAR.

Week-4 Evolutionary Process Models

- Prototyping
- Spiral Process Model

Evolutionary Process Models

- Evolutionary model is a **combination of Iterative and Incremental model of software development life cycle.**
- 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.



advantages

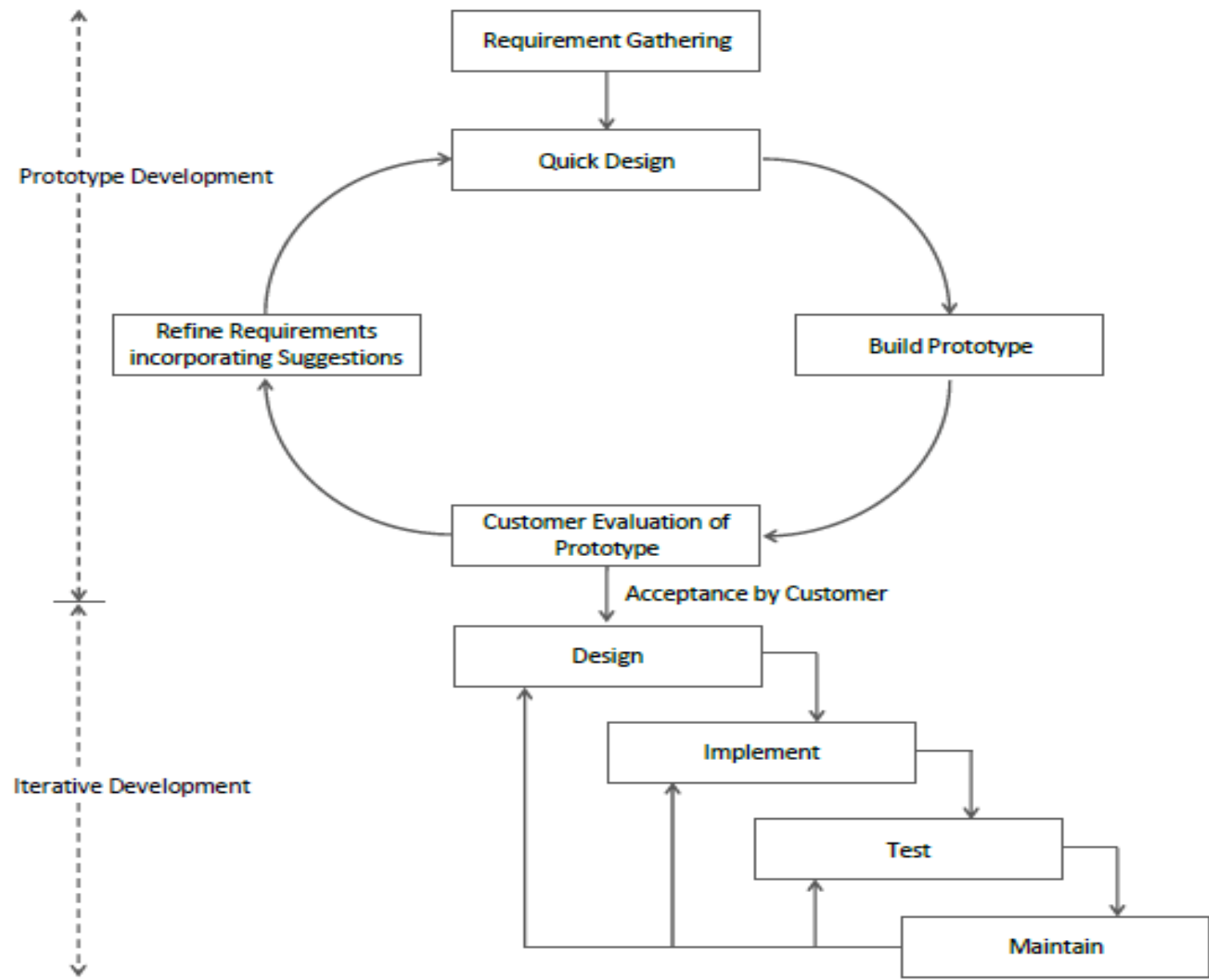
- **Large project:** Evolutionary model is normally useful for very large products.
- User gets a **chance to experiment with a partially developed software** much before the complete version of the system is released.
- Evolutionary model helps to accurately **elicit user requirements** during the delivery of different versions of the software.
- The core modules get tested thoroughly, thereby **reducing the chances of errors** in the core modules of the final products.
- Evolutionary model **avoids the need to commit large resources** in one go for development of the system.

Disadvantages

- **Difficult to divide the problem into several versions** that would be acceptable to the customer and which can be incrementally implemented and delivered.

prototype Models

- **Prototyping Model** is a software development model in which prototype is built, tested, and reworked until an acceptable prototype is achieved. It also creates base to produce the final system or software. It works best in scenarios where the project's requirements are not known in detail. It is an iterative, trial and error method which takes place between developer and client.



Prototyping Model has following six SDLC phases as follow:

- **Step 1: Requirements gathering and analysis**
- A prototyping model starts with requirement analysis. In this phase, the requirements of the system are defined in detail. During the process, the users of the system are interviewed to know what is their expectation from the system.
- **Step 2: Quick design**
- The second phase is a preliminary design or a quick design. In this stage, a simple design of the system is created. However, it is not a complete design. It gives a brief idea of the system to the user. The quick design helps in developing the prototype.
- **Step 3: Build a Prototype**
- In this phase, an actual prototype is designed based on the information gathered from quick design. It is a small working model of the required system.

- **Step 4: Initial user evaluation**

- In this stage, the proposed system is presented to the client for an initial evaluation. It helps to find out the strength and weakness of the working model. Comment and suggestion are collected from the customer and provided to the developer.

- **Step 5: Refining prototype**

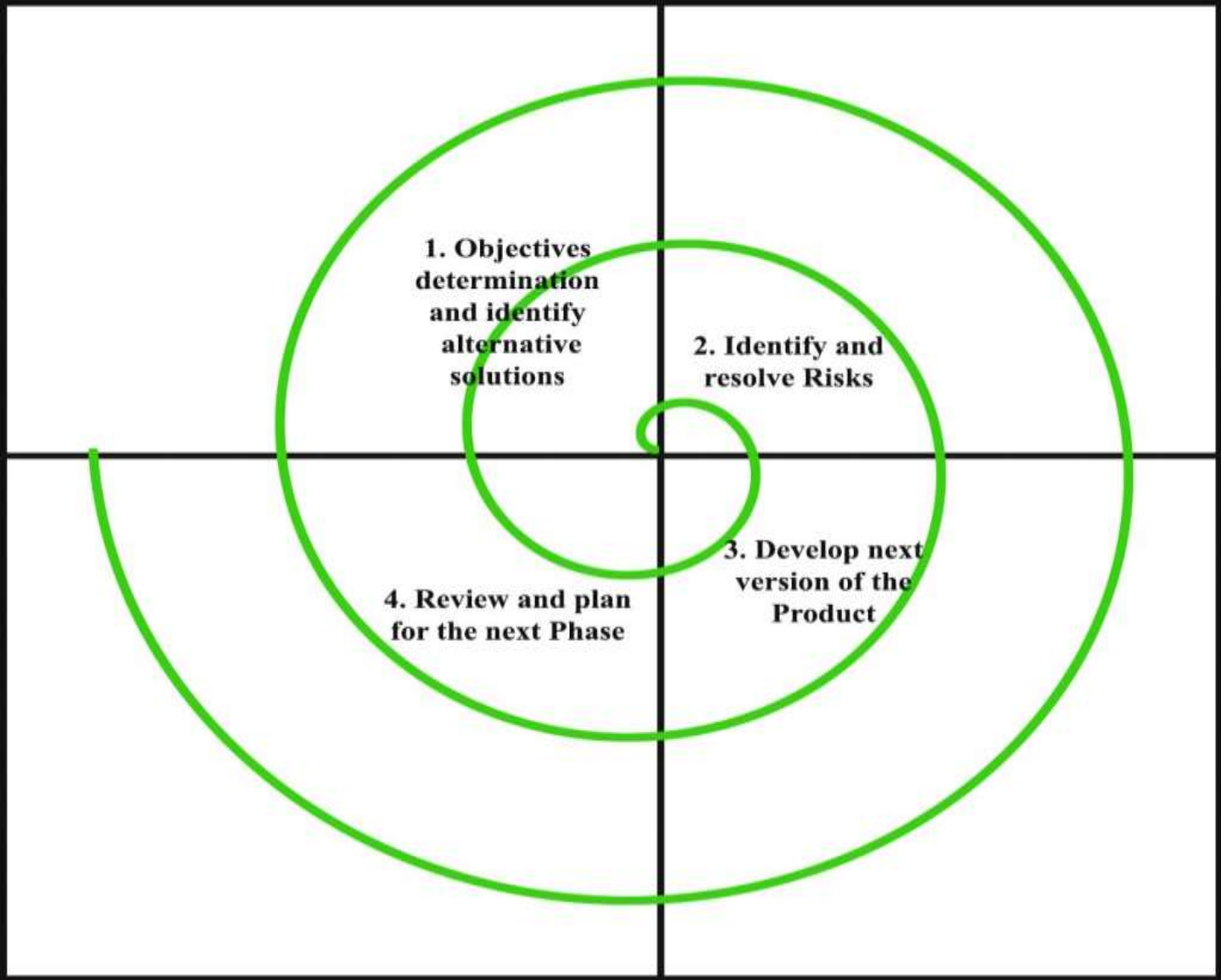
- If the user is not happy with the current prototype, you need to refine the prototype according to the user's feedback and suggestions.
- This phase will not over until all the requirements specified by the user are met. Once the user is satisfied with the developed prototype, a final system is developed based on the approved final prototype.

- **Step 6: Implement Product and Maintain**

- Once the final system is developed based on the final prototype, it is thoroughly tested and deployed to production. The system undergoes routine maintenance for minimizing downtime and prevent large-scale failures.

Spiral Process Model

- **Spiral Model** is a risk-driven software development process model.
- It is a combination of waterfall model and iterative model.
- Spiral Model helps to adopt software development elements of multiple process models for the software project based on unique risk patterns ensuring efficient development process.
- Each phase of spiral model in software engineering begins with a design goal and ends with the client reviewing the progress.
- The development process in Spiral model in SDLC, starts with a small set of requirement and goes through each development phase for those set of requirements.
- The software engineering team adds functionality for the additional requirement in every-increasing spirals until the application is ready for the production phase.



- **Objectives determination and identify alternative solutions:** Requirements are gathered from the customers and the objectives are identified, elaborated, and analyzed at the start of every phase. Then alternative solutions possible for the phase are proposed in this quadrant.
- **Identify and resolve Risks:** During the second quadrant, all the possible solutions are evaluated to select the best possible solution. Then the risks associated with that solution are identified and the risks are resolved using the best possible strategy. At the end of this quadrant, the Prototype is built for the best possible solution.
- **Develop next version of the Product:** During the third quadrant, the identified features are developed and verified through testing. At the end of the third quadrant, the next version of the software is available.
- **Review and plan for the next Phase:** In the fourth quadrant, the Customers evaluate the so far developed version of the software. In the end, planning for the next phase is started.

Advantages of Spiral Model:

- **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 later phase can be incorporated accurately by using this model.
- **Customer Satisfaction:** Customer 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.

Disadvantages of Spiral Model:

- **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, so time estimation is very difficult.