

## **PROCESS AND MODEL ADOPTED**

System development processes is a technique that is used to show how the proposed system will be developed. In this case, the methodology (model) used will be a waterfall model.

## **SPIRAL MODEL**

**Spiral model** is one of the most important Software Development Life Cycle models, which provides support for **Risk Handling**.

In its diagrammatic representation, it looks like a spiral with many loops. The exact number of loops of the spiral is unknown and can vary from project to project.

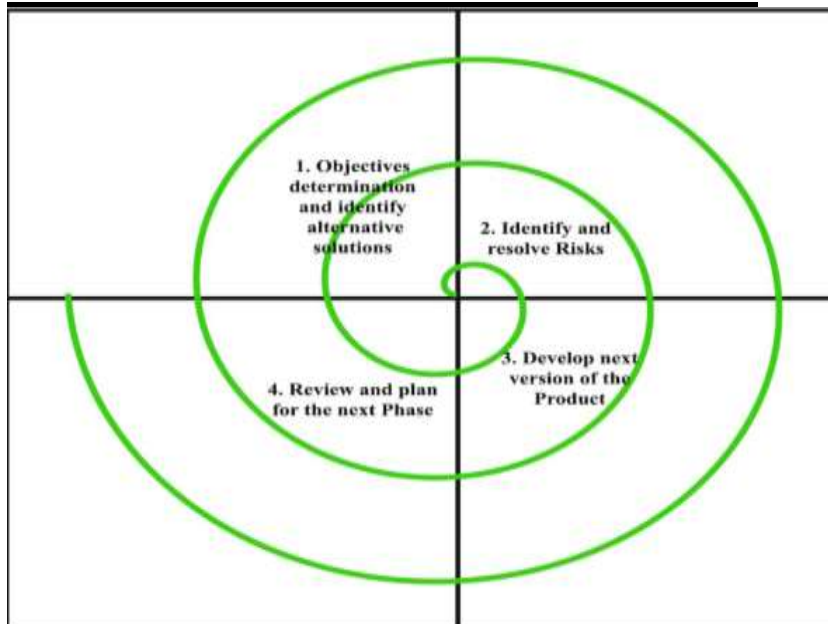
**Each loop of the spiral is called a Phase of the software development process.**

The exact number of phases needed to develop the product can be varied by the project manager depending upon the project risks.

As the project manager dynamically determines the number of phases, so the project manager has an important role to develop a product using spiral model.

The Radius of the spiral at any point represents the expenses(cost) of the project so far, and the angular dimension represents the progress made so far in the current phase.

## **DIAGRAMATIC REPRESTATION**



Each phase of Spiral Model is divided into four quadrants as shown in the above figure.

The functions of these four quadrants are discussed below-

1. **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.
2. **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 is identified and the risks are resolved using the best possible strategy. At the end of this quadrant, Prototype is built for the best possible solution.
3. **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.
4. **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.

## **RISK HANDLING IN SPIRAL MODEL**

A risk is any adverse situation that might affect the successful completion of a software project.

The most important feature of the spiral model is handling these unknown risks after the project has started. Such risk resolutions are easier done by developing a prototype. The spiral model supports coping up with risks by providing the scope to build a prototype at every phase of the software development.

**Prototyping Model** also support risk handling, but the risks must be identified completely before the start of the development work of the project. But in real life project risk may occur after the development work starts, in that case, we cannot use Prototyping Model. In each phase of the Spiral Model, the features of the product dated and analyzed and the risks at that point of time are identified and are resolved through prototyping.

Thus, this model is much more flexible compared to other SDLC models.

## **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 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**

- **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 dependable on Risk Analysis:** The successful completion of the project is very much dependent on Risk Analysis. Without very highly experienced expertise, 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.

## **When to Use Spiral model?**

- When the project is large.
- Where the software needs continuous risk evaluation.
- Requirements are a bit complicated and require continuous clarification.
- Software requires significant changes.
- Where enough time frame is there to get end user feedback.

- Where releases are required to be frequent.

### **Conclusion:**

- Each spiral can be termed as a loop and each loop is a separate development process in a spiral model. The four activities (Planning, Risk analysis, engineering and evaluation) form the intermediary phases of a spiral model and is repeated again for each loop.
- This model is very good to use for larger projects where you can develop and deliver smaller prototypes and can enhance it to make the larger software. The implementation of this model requires experienced resources as risk analysis is a very integral part of this model and risk analysis requires expertise and as a result this model becomes costly.

### **Functional Requirements**

The functional requirements of this system are:

- Register new students.
- Record the attendance of students.
- Record the internal marks of students.
- Record the feed details of students.
- Register a new teacher/employee.
- Register a new user for the system.
- Record the salary details of employees.
- Record the course details and subject information.
- Record the scholarship details and information.
- Generate various reports for all transactions in the system.

### **Non- Functional Requirements**

In this system, the authentication of the user is an important factor. In this system, user authentication will be done by login by user name and password and classified by user type. Users will get access to the system as permissions are classified for that type of user. The system has a consistent interface so that the system is easy to use and in the interface of our system buttons and forms are used to enter data related to a specific module.

