

SPIRAL MODEL

Gehad Safwat

AGENDA

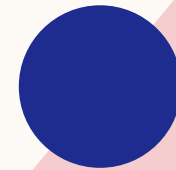
Introduction

Phases of Spiral Model

Risk Handling in Spiral Model

Spiral Model is called Meta Model

Pros & Cons of the Spiral Model



INTRODUCTION

- **The 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 Spiral Model is a **Software Development Life Cycle (SDLC)** model that provides a systematic and iterative approach to software development. 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.

What Is Spiral Model



PHASES OF SPIRAL MODEL

Each phase of the Spiral Model is divided into four quadrants as shown in the above figure. The functions of these four quadrants which are :

- **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 the 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.

RISK HANDLING IN SPIRAL MODEL

- 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 with risks by providing the scope to build a prototype at every phase of software development.
- The **Prototyping Model** also supports 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 the Prototyping Model.

SPIRAL MODEL IS CALLED META MODEL

- The spiral model incorporates the stepwise approach of the **Classical Waterfall Model**.
- The spiral model uses the approach of the **Prototyping Model** by building a prototype at the start of each phase as a risk-handling technique.
- Also, the spiral model can be considered as supporting the **Evolutionary model**, the iterations along the spiral can be considered as evolutionary levels through which the complete system is built.

PROS & CONS OF THE 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 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

PROS & CONS OF THE 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, 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.



THANK YOU