Software Development Life Cycle Models

Spiral Model

The Spiral Model is a combination of the iterative model and the waterfall model. The phases of the Spiral Model are represented as loops in the diagrammatic version. The project manager can dynamically determine the number of phases necessary for the development of the product. The Spiral Model focuses on managing risk through the multiple iterations of the phases. It is commonly regarded to have five phases.

The first phase of the Spiral Model is to clarify the aim of the project, including functional and non-functional requirements. During the second phase, the risk analysis phase, the risks associated with the project are evaluated. Next, the software is developed based on the requirements gathered in the previous iteration in the third phase, the engineering phase. In the fourth phase, the evaluation phase, the software is evaluated to see if it meets the requirements proposed by the customer. In the final fifth phase, the planning phase, the next iteration of the phases begins based off the results of the evaluation 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.

1. Complex: The Spiral Model is much more complex than other SDLC models.
2. Expensive: Spiral Model is not suitable for small projects as it is expensive.
3. 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.
4. Difficulty in time management: As the number of phases is unknown at the start of the project, time estimation is very difficult.
5. Complexity: The Spiral Model can be complex, as it involves multiple iterations of the software development process.
6. Time-Consuming: The Spiral Model can be time-consuming, as it requires multiple evaluations and reviews.
7. Resource Intensive: The Spiral Model can be resource-intensive, as it requires a significant investment in planning, risk analysis, and evaluations.

Extreme Programming