## ****Spiral model (SDLC)****

The spiral model is a systems development lifecycle (SDLC) method used for risk management that combines the iterative development process model with elements of the Waterfall model. The spiral model is used by software engineers and is favored for large, expensive and complicated projects.

**Risk handling in spiral model**

The spiral model analyzes all proposed solutions and identifies, analyzes and addresses all potential risks. Following this, methods such as prototyping, simulation, benchmark testing, analytical models, and user research are used to develop the lowest-risk, most cost-effective strategy.

**Spiral model phases**

The different phases of spiral model are-

**1. Planning**

This phase begins by gathering  business requirements into a baseline spiral. In the subsequent spiral as the product matures, all system, subsystem  and unit requirements are identified at this stage.

This phase also includes understanding  system requirements through ongoing communication between the customer and  system analysts. At the end of the spiral, the product will be deployed in the identified market. This includes iteration cost, schedule, and resource estimates. This includes understanding  system requirements for ongoing communication between  system analysts and customers.

**2. Risk Analysis**

After the “plan” phase, the team prepares for the “risk” phase. The “risk” phase is designed to take into account the variability in the rate at which a given product might fail. It is designed to account for the uncertainty in the rate at which a given product might fail. During the “risk” phase, the team evaluates various aspects of the current state of the product, such as the state of its code, the state of its design, and the state of its prototype. The team then makes adjustments to the current state of the product based on the changes made in the “plan” phase, and then follows up with a “sales” phase to collect customer feedback.

Once  risks are identified, risk mitigation strategies are planned and completed.

Briefly, risk analysis involves identifying, estimating and monitoring  technical feasibility and management risks such as: schedule slippage and cost overrun. After testing the build, at the end of the first iteration, customers rate the software and provide feedback.

**3. Product development**

In the next quadrant, prototypes are built and tested. This step includes architectural design, module design, physical product design and  final design. Convert the proposals made in the first two quadrants  into usable software.

This phase also includes the actual implementation of features in a project which are verified by performing testing.

4**. Next phase planning**

In this phase ,the software is evaluated by the customer and feedback is given. The team prepares for the next phase of the planning process. The next phase of the planning process is known as the “spiral” phase. During the “spiral” phase, the team determines the order of events in the current state of the product and then follows these events up with a “revision” phase to “Revise” the current state of the product so that it is ready for production. The “revision” phase is also called the “reproduction” phase, and it is one of the most important aspects of the planning process.



The advantages of the Spiral SDLC Model are as follows −

* Changing requirements can be accommodated.
* Allows extensive use of prototypes.
* Requirements can be captured more accurately.
* Users see the system early.
* Development can be divided into smaller parts and the risky parts can be developed earlier which helps in better risk management.

The disadvantages of the Spiral SDLC Model are as follows −

* Management is more complex.
* End of the project may not be known early.
* Not suitable for small or low risk projects and could be expensive for small projects.
* Process is complex
* Spiral may go on indefinitely.
* Large number of intermediate stages requires excessive documentation.