Mô hình xoắn ốc (*Spiral-Model*) là mô hình có sự kết hợp giữa mô hình thác nước (*Waterfall-Model*) và mô hình tiếp cận lặp (*Iterative-Model*) và nó có nhiều điểm giống nhau với mô hình gia tăng (*Incremental-Model*).

Chú trọng vào phân tích rủi ro dự án. Mỗi giai đoạn trong mô hình được bắt đầu với yêu cầu/mục tiêu thiết kế và kết thúc với việc khách hàng kiểm tra tiến độ của từng giai đoạn. Mô hình xoắn ốc lần đầu tiên được ***Barry Boehm*** đề cập trong bài báo của ông vào năm 1986.

**✤ Các giai đoạn của mô hình xoắn ốc:**

* Lập kế hoạch - Planning phase.
* Phân tích rủi ro - Risk analysis phase.
* Thực thi kỹ thuật - Engineering phase.
* Đánh giá - Evaluation phase.

**❈ Lập kế hoạch - Planning phase:**

Thu thập, phân tích yêu cầu từ của dự án từ phía khách hàng. Bao gồm các công việc: Ước lượng chi phí (*estimating cost*), lên lịch trình thực hiện dự án (*shedule-master*), xác định số lượng nhân lực, môi trường làm việc (*identifying necessary resources and work environment*), tìm hiểu yêu cầu hệ thống (*requirements*) từ đó đưa ra các tài liệu đặc tả (*Bussiness Requirement Specifications và System Requirement specifications*) để phục vụ cho việc trao đổi giữa khách hàng và phân tích hệ thống sau này.

**❈ Phân tích rủi ro - Risk analysis phase:**

Một quá trình phân tích sẽ được thực hiện để xác định rủi ro và đưa ra các giải pháp thay thế. Một prototype sẽ được tạo ra ở cuối giai đoạn phân tích rủi ro. Nếu có bất kỳ rủi ro nào được tìm thấy trong quá trình này thì các giải pháp thay thế sẽ được đề xuất và thực hiện.

**❈ Thực thi kỹ thuật - Engineering phase:**

Đây là giai đoạn mà dự án được các dev tiến hành code, các tester tiến hành test và deploying software trên trang web của khách hàng.

**❈ Đánh giá - Evaluation phase:**

Khách hàng sẽ tham gia vào giai đoạn này để đánh giá công việc, sản phẩm và đảm bảo rằng sản phẩm đáp ứng tất cả các yêu cầu đã đặt ra trước đó. Nếu có bất kỳ yêu cầu thay đổi nào từ khách hàng, các giai đoạn sẽ được lặp lại. Đây là giai đoạn quan trọng vì cần có sự phản hồi của khách hàng về sản phẩm trước khi sản phẩm được release.

**2. Khi nào nên sử dụng mô hình xoắn ốc?**

* Khi dự án có quy mô lớn.
* Khi việc đánh giá (phân tích) các chi phí và các rủi ro là quan trọng.
* Bất cứ lúc nào cũng có thể có yêu cầu thay đổi từ phía khách hàng.
* Khi dự án được yêu cầu release thường xuyên.
* Khi yêu cầu không rõ ràng và phức tạp.
* Đối với các dự án có độ rủi ro từ trung bình đến cao.
* Những người sử dụng không chắc chắn về các nhu cầu của họ.
* Các yêu cầu phần mềm phức tạp và lớn.
* Cần phát triển một dòng sản phẩm mới (*New product line*).
* Khi có các thay đổi quan trọng (cần nghiên cứu và khảo sát cẩn thận).

**3. Ưu điểm**

* Lượng phân tích rủi ro cao. Do đó việc tránh rủi ro được tăng cường.
* Ước lượng chi phí dễ dàng như việc hoàn thành một prototype trong một fragment nhỏ.
* Ứng dụng tốt đối với các dự án lớn và quan trọng.
* Kiểm soát tài liệu và phê duyệt chặt chẽ.
* Chức năng bổ sung hoặc thay đổi có thể được thêm vào những giai đoạn sau.
* Phần mềm sẽ được sản xuất sớm trong vòng đời của phần mềm.
* Ứng dụng được phát triển nhanh và các tinh năng được thêm vào một cách có hệ thống.
* Luôn có thời gian cho khách hàng để phản hồi về sản phẩm.

**4. Nhược điểm**

* Đối với rủi ro, ở giai đoạn phân tích cần một chuyên gia có chuyên môn cao để thực hiện việc phân tích.
* Không hữu ích với dự án có quy mô nhỏ.
* Thời gian và chi phí cho dự án có thể là vô hạn vì đặc tính xoắn ốc của mô hình.
* Tài liệu cho dự án có thể rất dài vì có các giai đoạn trung gian.
* Rủi ro có thể không đáp ứng được tiến độ hoặc ngân sách.
* Sự thành công của dự án phụ thuộc rất nhiều vào giai đoạn phân tích rủi ro.

**5. Kết luận**

* Mô hình xoắn ốc là một cách tiếp cận thực tế để phát triển các sản phẩm phần mềm quy mô lớn bởi vì phần mềm phát triển khi quá trình tiến triển (*the software evolves as the process progresses*). Ngoài ra, nhà phát triển và khách hàng hiểu rõ hơn và phản ứng với các rủi ro ở mỗi cấp độ (*level*) phát triển.
* Mô hình sử dụng prototyping như một cơ chế giảm rủi ro và cho phép phát triển các prototype ở bất kỳ giai đoạn nào của quá trình phát triển.
* Nó duy trì cách tiếp cận có tính hệ thống, giống như mô hình vòng đời (*Life Cycle - Model*) nhưng kết hợp nó thành một framework lặp lại và được phản ánh nhiều hơn từ thế giới thực.
* Nếu được sử dụng đúng cách, mô hình này sẽ giảm thiểu rủi ro trước khi chúng trở thành vấn đề. Vì các rủi ro kỹ thuật được xem xét ở tất cả các giai đoạn.

**Mô hình tiến hóa.**

**Ưu điểm:**

Chú trọng việc tái sử dụng mẫu. Một phần của hệ thống có thể được phát triển ngay trong các giai đoạn phân tích phát triển yêu cầu và thiết kế.

Cho phép thay đổi yêu cầu và khuyến khích người sử dụng tham gia trong suốt chu kỳ của dự án.

**Nhược điểm:**

Làm chậm quá trình phát triển yêu cầu và có thể ảnh hưởng sự chú ý đến các công việc trung gian như kiểm tra mã nguồn, thực hiện kiểm thử cấp thấp...

Dễ dẫn đến kết cấu của hệ thống kém.

Thường thì với mô hình này, tính chặt chẽ, minh bạch của qui trình kém.

**Ứng dụng:**

Hệ thống tương tác nhỏ và vừa; phần GUI của những hệ thống lớn; những hệ thống cần chu kỳ phát triển ngắn.

Đội ngũ phát triển không quen thuộc với lĩnh vực của dự án.

**Mô hình lặp và tăng dần**

**Ưu điểm:**

Giảm rủi ro sớm trong chu kỳ phát triển phần mềm. Những yêu cầu quan trọng thường được phát triển và chuyển đến người sử dụng sớm.

Phản hồi của nguời sử dụng về những vấn đề phát sinh trong phiên bản trước được dùng để cải tiến và ngăn ngừa những vấn đề tương tự xảy ra trong những phiên bản tiếp theo.

**Nhược điểm:**

Tổng chi phí lập kế hoạch phát triển cho toàn hệ thống có thể cao hơn. Lưu ý, ở đây chỉ đề cập chi phí lập kế hoạch ban đầu, không bao gồm tất cả chi phí phát sinh. Trong thực tế, nếu ứng dụng hợp lý, toàn bộ chi phí và thời gian cho đến khi sản phẩm được nghiệm thu có thể thấp hơn so với mô hình khác.

Các yêu cầu về kế hoạch và hoạt động trong qui trình cụ thể sẽ phức tạp hơn.

**Ứng dụng:**

*Mô hình lặp:*

Đội ngũ phát triển quen thuộc với lĩnh vực dự án nhưng không có nhiều kinh nghiệm, nhất là về công nghệ được dùng phát triển dự án.

Có nhiều rủi ro về mặt kỹ thuật

*Mô hình tăng dần:*

Rủi ro được phân tích và xác định ngay từ đầu.

Giao tiếp giữa các module cũng được xác định rõ ràng từ đầu.

Đội ngũ phát triển quen thuộc với lĩnh vực của dự án và có nhiều kinh nghiệm.

Hệ thống lớn được phát triển trong thời gian dài, khách hàng cần triển khai sớm một số phần của hệ thống.