

Lời mở đầu

Cũng như mọi ngành sản xuất khác, qui trình là một trong những yếu tố cực kỳ quan trọng đem lại sự thành công cho các nhà sản xuất phần mềm, nó giúp cho mọi thành viên trong dự án từ người cũ đến người mới, trong hay ngoài công ty đều có thể xử lý đồng bộ công việc tương ứng vị trí của mình thông qua cách thức chung của công ty, hay ít nhất ở cấp độ dự án.

Có thể nói qui trình phát triển/xây dựng phần mềm (Software Development/Engineering Process - SEP) có tính chất quyết định để tạo ra sản phẩm chất lượng tốt với chi phí thấp và năng suất cao, điều này có ý nghĩa quan trọng đối với các công ty sản xuất hay gia công phần mềm củng cố và phát triển cùng với nền công nghiệp phần mềm đầy cạnh tranh.

Để sản xuất cùng một sản phẩm phần mềm người ta có thể dùng các mô hình khác nhau. Tuy nhiên không phải tất cả các mô hình đều thích hợp cho mọi ứng dụng. Chúng ta sẽ đi chi tiết hơn về mô hình xoắn ốc (spiral model)_ một trong những mô hình được sử dụng khá phổ biến tới thời điểm hiện tại.

I. Tổng quát về lịch sử hình thành và sơ lược về mô hình xoắn ốc.

Chức năng chính của một mô hình phát triển phần mềm là để xác định thứ tự các giai đoạn liên quan đến phát triển phần mềm, sự tiến hóa và thiết lập các tiêu chuẩn của mỗi giai đoạn. Chúng bao gồm các tiêu chuẩn hoàn thiện các tiêu chí của giai đoạn cộng với sự lựa chọn và tiêu chuẩn hiện hành cho giai đoạn tiếp theo.



Vòng đời phần mềm của mỗi sản phẩm nhiều khi có sự khác biệt rất lớn. Có phần mềm dùng một vài năm cho việc khảo sát, tìm hiểu vấn đề. Những trường hợp như thế này thường xảy ra do chưa có phần cứng phù hợp để xây dựng phần mềm, hoặc cần phải tiến hành khá nhiều nghiên cứu để tìm ra thuật toán hiệu quả. Có sản phẩm được

thiết kế và viết chương trình rất nhanh nhưng lại tốn hàng năm để bảo trì do phải sửa đổi chương trình cho phù hợp với các yêu cầu của khách hàng. Cũng có những sản phẩm phần mềm sau một thời gian sử dụng người ta nhận thấy rằng có lẽ nên viết hẳn một sản phẩm mới hoàn toàn thì sẽ tốt hơn là bảo trì sản phẩm cũ. Cho đến nay có rất nhiều mô hình vòng đời phần mềm được sử dụng. Tuy nhiên mỗi mô hình lại có những ưu và nhược điểm khác nhau.

Mô hình xoắn ốc được trình bày trong bài viết này là một trong những ứng cử viên cho việc cải thiện các mô hình phát triển phần mềm hiện tại. Các tính năng phân biệt chủ yếu của mô hình xoắn ốc là nó tạo ra một cách tiếp cận theo định hướng rủi ro để phát triển phần mềm. Nó kết hợp nhiều trong những thế mạnh của các mô hình khác và giải quyết những khó khăn của những mô hình trước.

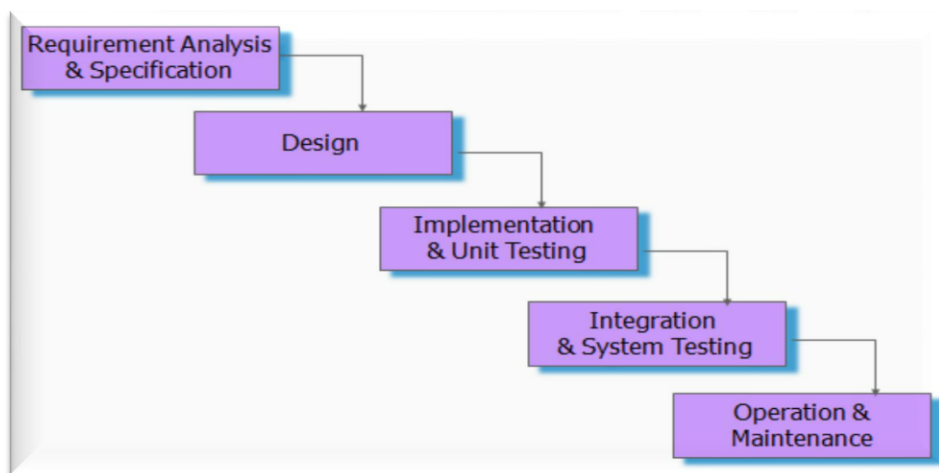
Trong khi mô hình thác nước(waterfall model) trình bày một cái nhìn đơn giản về vòng đời phần mềm, phương pháp tiếp cận của mô hình này giúp loại bỏ rất nhiều khó khăn trước đây gặp phải trên các dự án phần mềm. Trở thành cơ sở cho các tiêu chuẩn của hầu hết các phần mềm trong chính phủ và ngành công nghiệp, sự ra đời này đã cung cấp 2 cải tiến chính của các mô hình trước:

(1) Cập nhật thông tin phản hồi giữa các giai đoạn, và một hướng dẫn để giới hạn phản hồi các giai đoạn kế tiếp để giảm thiểu sự tổn kém liên quan đến thông tin phản hồi qua nhiều giai đoạn .

(2) Một kết hợp ban đầu các bản mẫu trong vòng đời phần mềm, thông qua một bước " "build it twice " chạy song song với các yêu cầu phân tích và thiết kế.

Tuy nhiên quan điểm này chỉ thích hợp cho những lớp chắc chắn trong sự phát triển phần mềm. Cụ thể, mô hình thác nước hoạt động tốt khi hiểu rõ được các yêu cầu phần mềm (như trình biên dịch hoặc hệ điều hành) và bản chất của sự phát triển phần mềm liên quan đến việc thỏa thuận hợp đồng. (Hợp đồng được ký kết tại giai đoạn đầu của tiến trình làm cho sự thay đổi các yêu cầu của khách sạn sau đó sẽ rất khó khăn. Sau khi khách hàng yêu cầu công việc, trên cơ sở đó để có căn cứ ký hợp đồng về thời gian và kinh phí, nếu thay đổi thì thời gian và kinh phí làm sẽ tăng lên đáng kể, mà làm khối lượng công việc lớn rất khó khăn.). Mô hình thác nước có một vị trí quan trọng vì nó đưa ra một hình mẫu về các bước mà một phần mềm cần phải trải qua là: phân tích hệ thống, thiết kế, cài đặt, tích hợp và bảo trì .

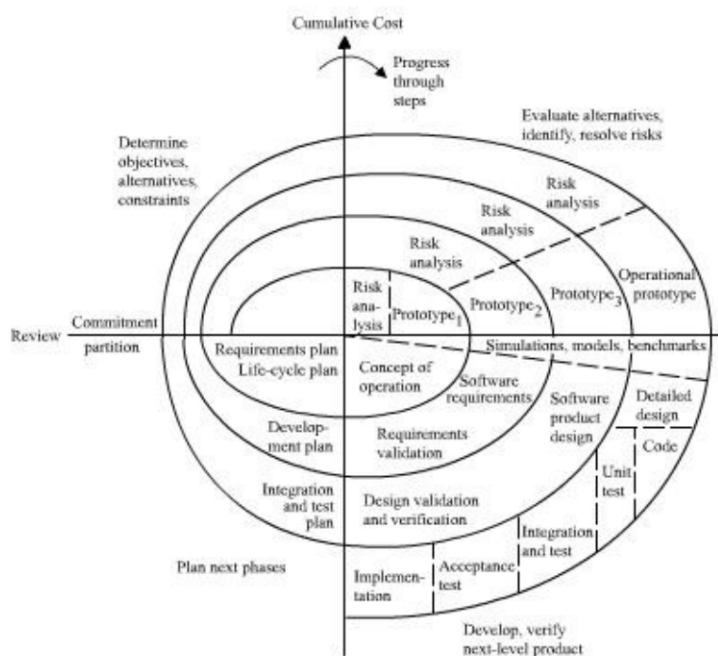
Tuy nhiên theo Boehm (*), mô hình này "không làm việc tốt cho sự phát triển phần mềm, đặc biệt là tương tác với các ứng dụng của người dùng cuối. *Khách hàng thường khó có thể phát biểu mọi yêu cầu một cách tường minh ngay từ đầu.* Đồng thời khách hàng phải kiên nhẫn chờ đợi, vì bản làm việc của chương trình chỉ có được vào thời gian cuối của dự án. Hơn nữa, nếu đến lúc đó chương trình làm việc mới phát hiện ra thì có thể là một thảm họa.

**Mô hình thác nước**

Hoặc với việc phân tích một số dự án hiện tại, thấy rằng *bản chất tuyến tính của vòng đời cổ điển dẫn tới "các trạng thái tắc nghẽn"*, nghĩa là có một số thành viên của nhóm phát triển phải chờ đợi sự chuyển giao từ nhóm khác hoàn thành công việc ở pha trước. Trong thực tế, thời gian chờ đợi có thể vượt quá thời gian sản xuất. Trạng thái nghẽn có xu hướng xảy ra vào thời gian đầu và cuối của quy trình phần mềm.

Ngoài những thiếu sót này, mô hình thác nước không cung cấp phương tiện để đánh giá những rủi ro và quản lý vòng đời.

Sự ra đời của các mô hình tiến hoá để giải quyết một số vấn đề mà các mô hình trước mắc phải, phát triển mô hình tiến hóa lý tưởng phù hợp với một ngôn ngữ thể hệ thứ tư và cũng phù hợp với các tình huống trong đó người dùng nói rằng: "Tôi không thể cho bạn biết những gì tôi muốn, nhưng tôi sẽ biết điều đó khi tôi nhìn thấy nó." Nó mang đến cho người dùng một khả năng tiếp cận nhanh chóng lúc đầu và cung cấp một cơ sở hoạt động thực tế để xác định cải tiến sản phẩm tiếp theo.



The Spiral Model

Năm 1988, Barry Boehm đã đề xuất một mô hình vòng đời toàn diện hơn gọi là mô hình xoắn ốc để giải quyết những bất cập của mô hình thác nước (trong bài viết "A Spiral Model of Software Development and Enhancement "). Tính năng phân biệt chủ yếu của mô hình xoắn ốc là nó tạo ra một cách tiếp cận theo định hướng rủi ro để xây dựng phần mềm chứ không phải là một quá trình chủ yếu là document-driven

hoặc driven-code kết hợp nhiều trong những thế mạnh của các mô hình khác và giải quyết nhiều khó khăn của họ.

Tiểu sử Boehm: ()*

Barry W. Boehm (sinh 1935) cử nhân toán học của Đại học Harvard vào năm 1957, thạc sĩ khoa học vào năm 1961, và tiến sĩ từ UCLA vào năm 1964.

Năm 1955, ông bắt đầu làm việc như một nhà Programmer-Analyst . Năm 1959, ông làm việc tại tổng công ty RAND , và trở thành trưởng phòng Khoa học Thông tin cho đến năm 1973. Từ 1973 đến 1989, ông là Giám đốc khoa học của Tập đoàn Hệ thống Quốc phòng TRW Inc . Từ 1989 đến 1992 ông phục vụ trong các Cục Quốc phòng của Mỹ (DoD) và là Giám đốc của các phần mềm DDR & E và Văn phòng Công nghệ máy tính. Từ 1992 Giáo sư danh dự của Công nghệ phần mềm tại Sở Khoa học Máy tính trường Đại học Nam California, và Giám đốc trung tâm Hệ thống và Công nghệ phần mềm USC, trước đây là Trung tâm Công nghệ phần mềm. Ông được biết đến với nhiều đóng góp của mình cho công nghệ phần mềm : mô hình chi

phí xây dựng (COCOMO), mô hình xoắn ốc của quá trình phát triển phần mềm...

II . Tiếp cận mô hình xoắn ốc:

1 . Định nghĩa.

Mô hình xoắn ốc là một quá trình phát triển phần mềm kết hợp các yếu tố của cả thiết kế và tạo mẫu trong mỗi giai đoạn. Còn được gọi là vòng đời mô hình xoắn ốc (hoặc hình xoắn ốc phát triển), nó là một phương pháp phát triển hệ thống (SDM) được sử dụng trong công nghệ thông tin. Đây là mô hình phát triển kết hợp các tính năng của mô hình bản mẫu và thác nước. Mô hình xoắn ốc được sử dụng phổ biến cho các dự án lớn, đắt tiền và phức tạp, đặc biệt áp dụng cho các dự án phần mềm lớn của chính phủ.

Đây không phải là mô hình đầu tiên thảo luận về phát triển lặp, nhưng nó là mô hình đầu tiên để giải thích lý do tại sao lặp lại vấn đề. Như hình dung ban đầu, lặp đi lặp lại thường từ 6 tháng đến 2 năm. Mỗi giai đoạn bắt đầu với một mục tiêu thiết kế và kết thúc đáp ứng được khách hàng và là mô hình tiến bộ cho đến thời điểm hiện tại. Sự cố gắng và những nỗ lực phân tích và công nghệ được áp dụng ở từng giai đoạn của dự án, với một tầm nhìn hướng tới mục tiêu cuối cùng của dự án.

2. Đặc điểm mô hình xoắn ốc:

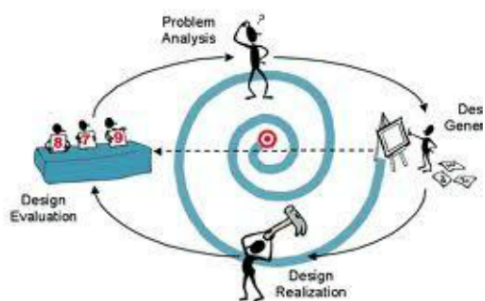
Về bản chất, mô hình mô tả sự phát triển của phần mềm qua các giai đoạn tiến hoá, mỗi giai đoạn được coi như một mô hình thác đổ. Ban đầu người ta chưa định nghĩa hệ thống một cách chi tiết, mà chỉ chú ý đến những đặc trưng nổi bật nhất. Sau đó phần đặc trưng này được xây dựng và đưa cho khách hàng xem xét, có ý kiến (cũng không hẳn là sử dụng cho công việc như trong mô hình tăng dần). Cùng những thông tin phản hồi từ khách hàng, người phát triển trở lại thực hiện các đặc trưng với mức độ chi tiết hơn. Bản chất mô hình xoắn ốc như tên gọi của nó, là bắt đầu từ những cái khái quát nhất rồi đi dần đến chi tiết.

Quá trình xây dựng phần mềm thường chứa đựng những rủi ro. Các nguy cơ như vượt chi phí dự án, yêu cầu thay đổi (ví dụ, hệ thống hành lý DIA), hay công ty chế tạo phần cứng mà phần mềm sẽ cài đặt bị phá sản. Sau khi chi phí hàng trăm nghìn đô la cho sự phát triển phần

mềm bỗng có một bước thay đổi đột phá trong công nghệ làm cho phần mềm trở nên vô dụng, phải thiết kế lại hoàn toàn. Công ty có thể nghiên cứu và phát triển hệ quản trị cơ sở dữ liệu, nhưng trước khi sản phẩm được hoàn thành và đưa ra thị trường thì một công ty khác lại quảng cáo một hệ tương đương có giá rẻ hơn. Có thể công ty sử dụng mô hình tăng dần đồng thời, nhưng sau đó các thành phần không thể tích hợp được với nhau để được phần mềm như yêu cầu đặt ra... Nói tóm lại, các nhà phát triển phần mềm thường có thể gặp rất nhiều rủi ro và họ muốn giảm thiểu các khả năng rủi ro đến mức có thể.

Ý tưởng làm giảm thiểu rủi ro thông qua việc sử dụng các bản mẫu và một số công cụ khác dẫn đến một mô hình mới mang tên: mô hình xoắn ốc (spiral model). Cách đơn giản nhất để xem xét mô hình này chính là mô hình thác đổ trong đó mỗi pha (trừ pha bảo trì) được bổ sung phần phân tích rủi ro ở trước. Trước khi bắt đầu một pha nào đó người ta phân tích các khả năng rủi ro và cách thức giải quyết có thể. Nếu không có cách nào để giải quyết được các rủi ro quan trọng thì dự án sẽ kết thúc.

Mô hình xoắn ốc cung cấp cách thức làm phần mềm bằng cách đưa ra các phiên bản tăng dần. Sự tăng dần ở đây không phải là bổ sung thêm các thành phần mới như mô hình tăng dần, mà sự tăng ở đây là sự tiến hóa, tức là cũng là các đặc trưng ấy nhưng được làm mịn hơn, chi tiết hơn. Phiên bản sau cùng chính là phần mềm hoàn chỉnh có thể chuyển giao cho khách hàng sử dụng.



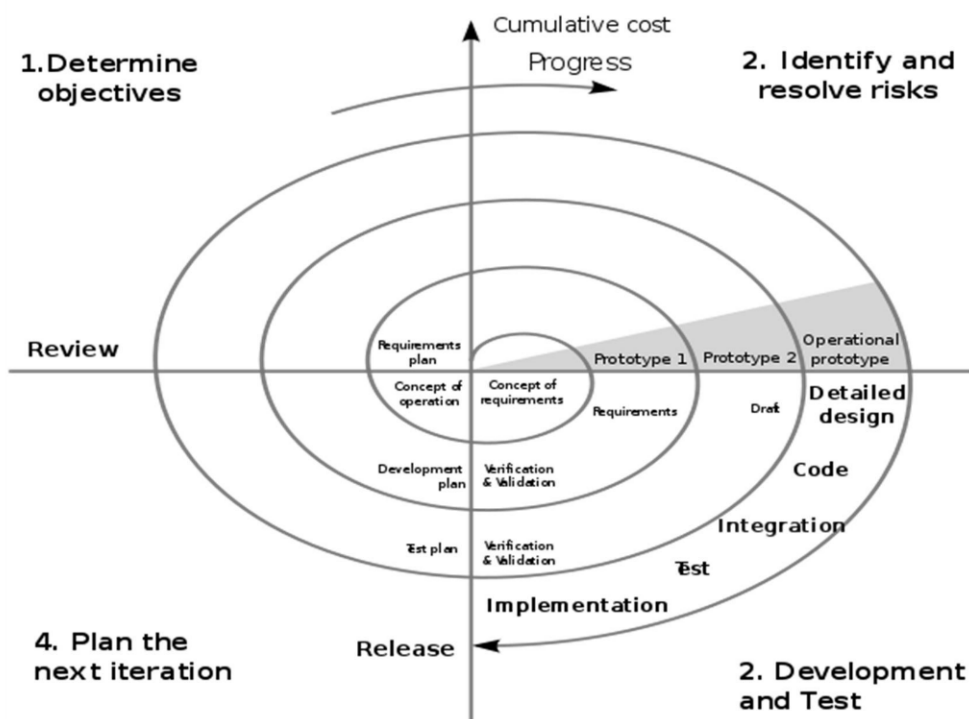
Kích thước xuyên tâm trong hình đại diện cho sự tích lũy chi phí phát sinh

3. Quy trình

trong việc hoàn thành các bước cho đến khi hoàn thành, kích thước góc đại diện cho sự tiến bộ đạt được trong việc hoàn thành mỗi chu kỳ xoắn ốc. (Mô hình phản ánh khái niệm cơ bản mà mỗi chu kỳ sẽ liên quan đến một sự tiến triển để giải quyết cùng một trình tự các bước, cho mỗi phần của sản phẩm và cho mỗi mức độ lặp, từ một khái niệm tổng thể của tài liệu hoạt động mã hóa của mỗi cá nhân chương trình.

Quá trình phát triển được chia thành nhiều bước lặp lại, mỗi bước bắt đầu bằng việc lập kế hoạch, phân tích rủi ro, rồi tạo bản mẫu, hoàn thiện và phát triển hệ thống, duyệt lại, và cứ thế tiếp tục. Nội dung gồm 4 hoạt động chính:

- ✚ **Lập kế hoạch** :Xác định mục tiêu, các giải pháp khác nhau để đạt được mục tiêu, các ràng buộc.
- ✚ **Phân tích rủi ro** : Phân tích những rủi ro và khả năng giải quyết (thường là xây dựng bản mẫu).
- ✚ **Phát triển và kiểm tra.**
- ✚ **Lập kế hoạch cho pha tiếp theo.**



Với mỗi lần lặp vòng xoắn ốc (bắt đầu từ tâm), các phiên bản được hoàn thiện dần. Tại một vòng xoắn ốc, phân tích rủi ro phải đi đến quyết định “ tiến hành tiếp hay dừng “. Nếu rủi ro quá lớn, thì có thể đình chỉ dự án hay thay đổi yêu cầu đặt ra cho thích hợp.

3.1 Lập kế hoạch:

Xác định được mục tiêu dựa trên các yêu cầu của khách hàng, nêu ra giải pháp thực hiện và các ràng buộc. Nhiệm vụ này đòi hỏi việc định nghĩa các tài nguyên, hạn thời gian và các thông tin liên quan tới dự án. Mỗi chu kỳ xoắn ốc bắt đầu với việc xác định:

- các mục tiêu của các phần của sản phẩm được xây dựng (hiệu suất, tính năng, khả năng thích ứng với sự thay đổi, vv).
- các thay thế nghĩa là thực hiện phần này của sản phẩm bằng cách thay thế nào (thiết kế A, thiết kế B, tái sử dụng, mua, lập kế hoạch, kiểm soát nhân sự ...)
- các hạn chế đối với việc áp dụng các lựa chọn thay thế (chi phí, thời gian...).

Khởi đầu và chấm dứt xoắn ốc. Bốn câu hỏi cơ bản phát sinh trong việc xem xét trình bày này của mô hình xoắn ốc:

- (1) Làm thế nào để hình xoắn ốc được bắt đầu ?
- (2) Khi nào thích hợp để chấm dứt một dự án ?
- (3) Tại sao xoắn ốc kết thúc quá đột ngột ?
- (4) Điều gì sẽ xảy ra khi phần mềm được nâng cấp (hoặc bảo trì) ?

Những câu trả lời cho những câu hỏi liên quan đến một quan sát rằng mô hình xoắn ốc áp dụng tốt trong sự phát triển hoặc nâng cấp phần mềm. Trong cả hai trường hợp, mô hình xoắn ốc được bắt đầu bởi một giả thuyết rằng một nhiệm vụ hoạt động cụ thể, có thể được cải thiện bằng một quá trình nỗ lực. Sau đó, quá trình xoắn ốc liên quan đến một thử nghiệm về giả thuyết: bất cứ lúc nào nếu không được kiểm tra (ví dụ, nếu chậm trễ thì một phần mềm, sản phẩm sẽ bỏ lỡ thị trường, hoặc nếu một sản phẩm thương mại cao cấp thì trở nên đã có sẵn trước), thì xoắn ốc sẽ bị chấm dứt. Nếu không, nó chấm dứt với việc cài đặt mới về sửa đổi phần mềm, và giả thuyết được kiểm tra bằng cách quan sát vào hiệu quả hoạt động. Thông thường, kinh nghiệm với nhiệm vụ hoạt động dẫn đến giả thuyết khác về cải tiến phần mềm, và một vòng xoáy bảo trì mới được bắt đầu để kiểm tra giả thuyết.

3.2 Phân tích rủi ro:

Phần này ta sẽ phân tích các rủi ro có thể xảy ra khi thực hiện, nhiệm vụ đòi hỏi xác định những rủi ro kỹ thuật và quản lý. Ý tưởng làm giảm thiểu rủi ro thông qua việc sử dụng các bản mẫu và một số công cụ khác dẫn đến một mô hình mới mang tên: mô hình xoắn ốc (spiral model). Cách đơn giản nhất để xem xét mô hình này chính là mô hình thác đổ trong đó mỗi pha (trừ pha bảo trì) được bổ sung phần phân tích rủi ro ở trước. Trước khi bắt đầu một pha nào đó người ta phân tích các khả năng rủi ro và cách thức giải quyết có thể. Nếu không có cách nào để giải quyết được các rủi ro quan trọng thì dự án sẽ kết thúc.

Để quản lý rủi ro của một giai đoạn trong mỗi vòng xoắn ốc, Boehm sử dụng mẫu dưới đây để đánh giá rủi ro trong quá trình phát triển của một phần mềm. Các hàng đại diện cho các yếu tố quản lý khác nhau của dự án. Đối với mỗi giai đoạn mới, ông đã tạo ra một thể hiện mới của các mẫu để xem xét tình trạng của dự án và quyết định liệu những rủi ro có thể tiếp tục hay là dừng lại.

<i>Template</i>	<i>Explanation</i>	<i>Example Phase</i>
Objectives	The goals of the software project	Significantly improve software quality
Constraints	Limitations which the project must meet	Within three years Without large-scale capital investment Without radical change to company standards
Alternatives	Possible ways to achieve the objectives	Reuse existing certified software Introduce formal specification and verification Invest in testing and validation tools
Risks	Potential risks for this phase	No cost effective quality improvement possible Quality improvements may increase costs excessively New methods might cause existing staff to leave
Risk Resolution	Strategies for reducing the	Literature survey, Pilot project, Survey of potential reusable components, Assessment of available tool support, Staff

	risks	training and motivation seminars
Results	Results of applying risk resolution strategies	Experience of formal methods is limited - very hard to quantify improvements Limited tool support available for company standard development system Reusable components available but little reuse tool support
Plans	Development plans for the next phase	Explore reuse option in more detail Develop prototype reuse support tools Explore component certification scheme
Commitment	Resources needed to achieve the plans	Fund further 18-month study phase

Spiral Model Template

Trong ví dụ công ty phần mềm, có mục tiêu cải thiện đáng kể chất lượng của phần mềm của họ. Để đáp ứng mục tiêu này, công ty đánh giá ba lựa chọn thay thế và ba rủi ro. Tuy nhiên, có thể phải chịu các nguy cơ gây ra bởi nhân viên hiện có để lại kể từ khi họ thích sử dụng phương pháp quen thuộc hơn của phát triển phần mềm. Để giải quyết nguy cơ này, các cuộc hội thảo đào tạo nhân viên được tiến hành cho thấy những lợi ích của những phương pháp mới và xác định mức độ chuyên môn hiện tại của phương pháp chính thức. Là một Công ty, kết quả phát hiện ra rằng các nhân viên biết rất ít về những phương pháp này. Vì vậy, rất khó để ước tính công ty có thể nhận được lợi ích từ việc sử dụng thay thế để đáp ứng mục tiêu của nó. Kể từ khi tùy chọn này có vẻ quá mạo hiểm, các kế hoạch cho giai đoạn tiếp theo tập trung vào một lựa chọn thay thế khác là có triển vọng hơn là tái sử dụng các thành phần của phần mềm.

Như ta đã thấy, sử dụng bản mẫu trong pha xác định yêu cầu là cách thức tuyệt vời để ngăn ngừa khả năng sản xuất ra một phần mềm không thỏa mãn tất cả các yêu cầu của khách hàng. Trong các pha tiếp theo, người ta cũng có thể xây dựng những bản mẫu thích hợp. Chẳng hạn, công ty điện thoại có thể vừa phát minh ra một thuật toán hiệu quả cho việc phân tuyến các cuộc gọi thông qua mạng diện rộng. Nếu phần mềm được xây dựng nhưng không làm việc được

như mong muốn, thì công ty sẽ bị thiệt hại về kinh phí. Trong trường hợp này khách hàng có thể bực tức và chuyển sang lựa chọn một công ty khác. Tình trạng này có thể được loại trừ nếu ta xây dựng một bản mẫu chỉ dùng cho mục đích phân tuyến các cuộc gọi và được kiểm thử trên thiết bị mô phỏng. Bằng cách này hệ thống thật sẽ không bị ảnh hưởng, và giá của công trình chỉ là thuật toán phân tuyến. Sau khi thử nghiệm, công ty sẽ quyết định được là có nên áp dụng thuật toán mới cho toàn hệ thống của họ hay không.

Tuy nhiên cũng có những rủi ro không thể đánh giá được thông qua bản mẫu. Ví dụ như nếu một thành viên chủ chốt xin thôi việc trước khi sản phẩm hoàn thành thì liệu có thể kiếm được người thay thế kịp thời hay không? Hoặc trình độ của các thành viên trong nhóm phát triển liệu có đáp ứng được việc phát triển phần mềm quy mô lớn hay không? Các thành viên công ty lâu nay vẫn thường xây dựng các phần mềm sử dụng trong gia đình, nay phải xây dựng phần mềm phức tạp sử dụng trong công sở thì có làm được không? Một lĩnh vực khác mà bản mẫu cũng không sử dụng được trong việc đánh giá rủi ro là những hứa hẹn về sự phát triển phần cứng. Phần mềm được phát triển có tính tới sự ra đời của các thiết bị mà các công ty phần cứng hứa hẹn, nhưng thực tế lại không xảy ra như vậy...

3.3 Phát triển và kiểm tra:

Trong giai đoạn này, chúng ta sẽ phát triển sản phẩm theo kế hoạch. Thử nghiệm cũng được thực hiện. Để phát triển, ta sử dụng mô hình thác nước hoặc tiếp cận từng bước có thể được thực hiện.

3.4 Lập kế hoạch cho pha tiếp theo:

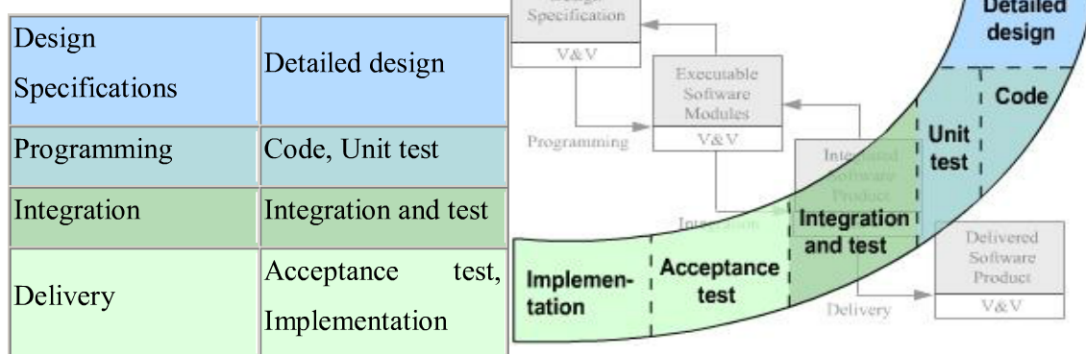
Ở đây, chúng ta xem xét tiến độ và đánh giá, xem xét tất cả các thông số. Các vấn đề cần được giải quyết được, như các yêu cầu thêm của khách hàng và tiếp tục thực hiện các bước trên.

Giai đoạn cuối cùng của mô hình xoắn ốc là tương tự như mô hình thác nước. Tại thời điểm này trong dự án, yêu cầu phần mềm nên được hiểu rõ thông qua sự phát triển của một số nguyên mẫu. Dự án cũng phải giải quyết các rủi ro chính để xây dựng phiên bản cuối cùng của phần mềm. Với những vấn đề được giải quyết, bản thiết kế chi tiết của phần mềm của ba quá trình cuối cùng chính là mô hình thác nước. Mặc dù tên các giai đoạn trong mô hình xoắn ốc có

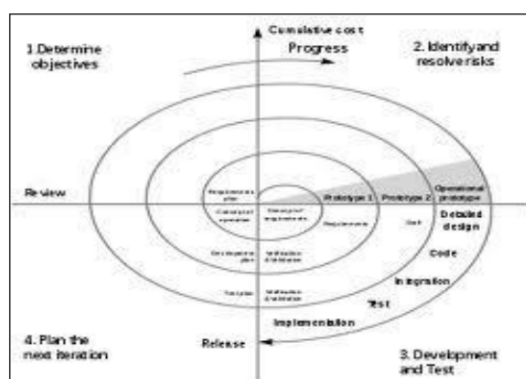
khác nhau với mô hình thác nước nhưng các quá trình này được thực hiện gần như tương tự nhau. Bảng dưới đây cho thấy sự tương ứng giữa giai đoạn cuối cùng của mô hình xoắn ốc và mô hình thác nước.

Waterfall Model

Spiral Model



4. Biểu diễn mô hình:



Để biểu diễn sơ đồ cho mô hình xoắn ốc, người ta vẽ hai đường thẳng vuông góc cắt nhau chia mặt phẳng thành 4 vùng. Bốn vùng này tương ứng với 4 vùng công việc: nếu dịch chuyển theo chiều kim đồng hồ và bắt đầu từ góc phần tư phía trên bên trái ta có các vùng tương ứng là 1,2,3,4.

Coi giao điểm của hai đường thẳng là tâm, ta vẽ các đường xoắn ốc đi từ phía trong ra ngoài cũng theo chiều kim đồng hồ. Độ dài đường xoắn ốc sẽ biểu diễn giá tích lũy của phần mềm, Một vòng của đường xoắn ốc sẽ biểu diễn một pha. Nếu đi từ trong ra ngoài ở góc phần tư số 3 ta được mô hình thác đổ. Một pha bắt đầu từ góc phần tư phía trên bên trái (góc 1) bằng việc xác định các mục tiêu của pha, các giải pháp khác nhau để đạt được các mục tiêu này và các ràng buộc cho từng giải pháp. Kết quả của giai đoạn này là chọn được giải pháp thích hợp. Ở góc

phần tư thứ hai là phân tích rủi ro cho giải pháp đã lựa chọn. Một vài biện pháp được đưa ra để khắc phục rủi ro. Biện pháp thường được sử dụng là bản mẫu. Nếu rủi ro lớn và không có biện pháp khắc phục thì dự án phải dừng lại. Trong một số trường hợp, dự án vẫn được tiếp tục nhưng với quy mô nhỏ hơn. Nếu vấn đề rủi ro được giải quyết thì chuyển sang góc phần tư thứ ba là phát triển. Ở góc cuối cùng là kế hoạch cho pha tiếp theo. Đường xoắn ốc sẽ được lặp lại chừng nào sản phẩm chưa đạt mức hoàn chỉnh.

III . Đánh Giá

1. Thuận lợi

Ưu điểm chính của mô hình xoắn ốc là phạm vi của nó có các lựa chọn thích ứng với các tính năng tốt của các mô hình phát triển phần mềm hiện có, trong cách tiếp cận theo định hướng rủi ro tránh nhiều khó khăn mà các mô hình khác gặp phải. Trong những tình huống thích hợp, mô hình xoắn ốc trở nên tương đương với một trong những mô hình quy trình hiện có. Trong tình huống khác, nó cung cấp hướng dẫn trên kết hợp tốt nhất các phương pháp tiếp cận hiện có một dự án nhất định, ví dụ, ứng dụng của nó TRW-SPS cung cấp một kết hợp tạo mẫu, quy định cụ thể và phát triển tiến hóa theo định hướng rủi ro chính điều kiện này mà theo đó mô hình xoắn ốc trở nên tương đương với các mô hình quá trình :

- Nếu một dự án có nguy cơ thấp trong các lĩnh vực như giao diện người dùng sai hoặc không đáp ứng yêu cầu thực hiện nghiêm ngặt, và nếu nó có nguy cơ cao trong ngân sách và khả năng dự báo lịch trình và kiểm soát, sau đó những cân nhắc các nguy cơ và hướng mô hình xoắn ốc vào một tương đương_ mô hình thác nước.
- Nếu yêu cầu một sản phẩm phần mềm là rất ổn định (ngụ ý một rủi ro thấp của thiết kế đắt tiền và vỡ mã do yêu cầu thay đổi trong quá trình phát triển), và nếu sự hiện diện của các lỗi trong các sản phẩm phần mềm tạo thành một nguy cơ cao , sau đó những cân nhắc nguy cơ và hướng mô hình xoắn ốc giống như mô hình two-leg (*Specification Languages: Understanding Their Role in Simulation Model Development- C.Michael Overstreet Richard E.Nance Osman Balci Lynne F.Barger*)
- Nếu một dự án có một rủi ro như mất ngân sách và khả năng dự báo, kiểm soát tiến độ và gặp phải các vấn đề hội nhập vào những hệ thống lớn, hoặc đối phó với xơ cứng thông

tin, và nếu nó có nguy cơ cao trong các lĩnh vực như giao diện người dùng sai hoặc hỗ trợ người dùng yêu cầu quyết định, sau đó những cân nhắc rủi ro và hướng mô hình xoắn ốc vào một tương đương _ evolutionary development model..

- Nếu các yếu tố nguy cơ cao của một dự án liên quan đến một kết hợp của các mục rủi ro được liệt kê ở trên, sau đó các phương pháp tiếp cận xoắn ốc sẽ phản ánh một sự pha trộn thích hợp của các mô hình quá trình ở trên (ví dụ như trong việc áp dụng TRW-SPS). Làm như vậy, các tính năng tránh nguy cơ của nó thường sẽ tránh được những khó khăn của các mô hình khác.

Tóm lại ,mô hình xoắn ốc có những thuận lợi:

- Spiral Life Cycle Model là một trong những mô hình linh hoạt nhất SDLC tại chỗ . Giai đoạn phát triển có thể được xác định bởi người quản lý dự án, theo sự phức tạp của dự án.
- Giám sát dự án là rất dễ dàng và hiệu quả. Mỗi giai đoạn, cũng như mỗi vòng lặp, yêu cầu xem xét từ những người có liên quan. Điều này làm cho các mô hình minh bạch hơn.
- Quản lý rủi ro là một trong những tính năng trong xây dựng của mô hình, mà làm cho nó thêm hấp dẫn so với các mô hình khác.
- Thay đổi có thể được giới thiệu sau trong vòng đời là tốt. Và đối phó với những thay đổi này không phải là một nhức đầu rất lớn đối với người quản lý dự án.
- Dự đoán dự án về thời hạn, chi phí, vv trở nên nhiều hơn và thực tế hơn như dự án di chuyển về phía trước và trong vòng xoắn ốc được hoàn thành.
- Nó phù hợp đối với các dự án có nguy cơ cao, nơi mà nhu cầu kinh doanh có thể không ổn định.
- Một sản phẩm tùy biến rất cao có thể được phát triển bằng cách sử dụng này

2 . Khó khăn

Mô hình xoắn ốc là cách tiếp cận thực tế cho việc phát triển các phần mềm quy mô lớn. Bởi vì phần mềm được tiến hóa theo đường xoắn ốc, từ tổng quan cho đến chi tiết, nên người

phát triển và khách hàng hiểu rõ hơn và có phản ứng thích hợp với rủi ro tại từng mức tiến hóa. Mô hình này dùng bản mẫu như một cơ chế làm giảm rủi ro. Bản mẫu còn giúp cho khách hàng nhìn rõ từng bước phát triển của phần mềm và có ý kiến góp ý kịp thời để những người phát triển đi đúng hướng, nhanh chóng đưa đến phần mềm hoàn thiện. Mô hình đòi hỏi xem xét trực tiếp các rủi ro kỹ thuật cũng như quản lý tại mọi giai đoạn của dự án, và nếu được áp dụng đúng thì có thể làm giảm rủi ro trước khi những rủi ro này trở thành vấn đề thực sự.

Tuy nhiên mô hình này không phải là sự lựa chọn tốt nhất cho mọi dự án.

- Trước hết, phân tích rủi ro sẽ tốn kém, do đó mô hình chỉ có thể áp dụng cho các dự án lớn, khi mà chi phí phân tích rủi ro là không đáng kể so với tổng chi phí toàn bộ dự án.
- Yêu cầu thay đổi thường xuyên dẫn đến lặp vô hạn, phức tạp, và cần có kỹ năng tốt về phân tích rủi ro.
- Phân tích rủi ro được thực hiện trong suốt quá trình phát triển phần mềm. Tuy nhiên nếu là phần mềm ký hợp đồng mà bị dừng lại thì công ty phát triển sẽ bị phạt. Do đó với các dự án ký hợp đồng thì nhà phát triển và khách hàng phải phân tích rủi ro trước khi hợp đồng được ký, chứ không phải trên đường xoắn ốc như mô hình mô tả.
- Liệu các nhà phát triển đã nhìn thấy hết các rủi ro không? Có thể rủi ro vẫn còn nhưng họ lại chủ quan cho rằng đã hết và có thể mắc sai lầm. Như vậy mô hình này chỉ nên áp dụng nếu công ty phần mềm có một đội ngũ chuyên gia phân tích rủi ro trình độ cao.

3. Ứng dụng:

Mô hình là hướng tiếp cận thực nhất để phát triển các hệ thống lớn .

Trong quân đội mô hình xoắn ốc được áp dụng trong chương trình hệ thống chiến đấu tương lai (FCS) _ hệ thống chiến đấu sử dụng các công nghệ kỹ thuật tiên bộ trong chiến tranh. Theo kế hoạch, FCS bao gồm mạng lưới cảm biến mặt đất không cần giám sát (UGS), xe trên không không người lái (UAV), các phương tiện mặt đất không người lái, và tám người lái xe mặt đất. Công ty Boeing và Khoa học Công ty Cổ phần Quốc tế (SAIC) đã làm việc với nhau như các nhà tích hợp hệ thống đạo, phối hợp hơn 550 nhà thầu và nhà thầu phụ trong 41 tiểu bang. Một mô hình xoắn ốc đã được lên kế hoạch cho FCS phát triển và nâng cấp. Tính đến năm 2004, FCS

trong giai đoạn phát triển hệ thống và trình diễn (SDD), trong đó bao gồm bốn hình xoắn ốc trong hai năm. Tuy nhiên: ngày 05 tháng 10 năm 2005, Lầu Năm Góc đã đề nghị trì hoãn hệ thống Future Combat của quân đội vì chi phí cho cuộc chiến tranh Iraq, bão Katrina, và sự suy giảm trong ngân sách trong dự kiến.

Và dự án FCS đã bị hủy bỏ sau sáu năm (2003-2009), đã có một sự lặp lại hai năm (xoắn ốc). FCS nên có kết quả trong ba nguyên mẫu liên tiếp (một nguyên mẫu cho mỗi chu kỳ xoắn ốc hai năm một lần). Nó đã bị hủy bỏ vào năm 2009. Bộ Quốc Phòng đã ban hành hủy bỏ chương trình Future Combat Systems và thay thế nó bằng các chương trình riêng biệt thuộc quân đội chiến đấu hiện đại hóa đội Lữ đoàn ô để đáp ứng kế hoạch của quân đội.

Ngoài ra, sử dụng các mô hình xoắn ốc là hợp lý trong các dự án mục tiêu kinh doanh không ổn định, nhưng kiến trúc phải được thực hiện cũng đủ để thực hiện tốt và khả năng ứng dụng. Ví dụ, Spiral Architecture Driven Development là xoắn ốc dựa trên Development Life Cycle (SDLC) trong đó cho thấy một trong những cách có thể làm thế nào để giảm nguy cơ của kiến trúc không hiệu quả với sự giúp đỡ của một mô hình xoắn ốc kết hợp với các hoạt động tốt nhất từ các mô hình khác.