

MÔ HÌNH PHÁT TRIỂN PHẦN MỀM CƠ BẢN

Nguyễn Minh Hiền

Ngày 4 tháng 3 năm 2024

Mục lục

Giới thiệu mô hình phát triển phần mềm	3
1 Mô hình thác nước (Waterfall model)	4
1.1 Định nghĩa	4
1.2 Ưu điểm	5
1.3 Nhược điểm	6
1.4 Khi nào sử dụng	7
2 Mô hình chữ V (V model)	8
2.1 Định nghĩa	8
2.2 Ưu điểm	9
2.3 Nhược điểm	9
2.4 Khi nào sử dụng	9
3 Mô hình bản mẫu (Prototype model)	10
3.1 Định nghĩa	10
3.2 Ưu điểm	11
3.3 Nhược điểm	11

3.4	Khi nào sử dụng	11
4	Mô hình xoắn ốc (Spiral model)	12
4.1	Định nghĩa	12
4.2	Ưu điểm	13
4.3	Nhược điểm	13
4.4	Khi nào sử dụng	13

Giới thiệu mô hình phát triển phần mềm

Mô hình phát triển phần mềm là một quy trình phát triển phần mềm có hiệu quả được đúc rút từ kinh nghiệm thực tế. Mô hình này bao gồm các pha, các nguyên tắc, nguyên lý để hướng đến mục tiêu cuối cùng là phát triển phần mềm một cách hiệu quả nhất.

Mỗi mô hình phát triển phần mềm đều có những ưu nhược điểm riêng, và phù hợp với các dự án có một số đặc điểm nhất định. Do đó, việc lựa chọn mô hình phát triển phần mềm phù hợp với dự án là một vấn đề quyết định đến thành công của dự án.

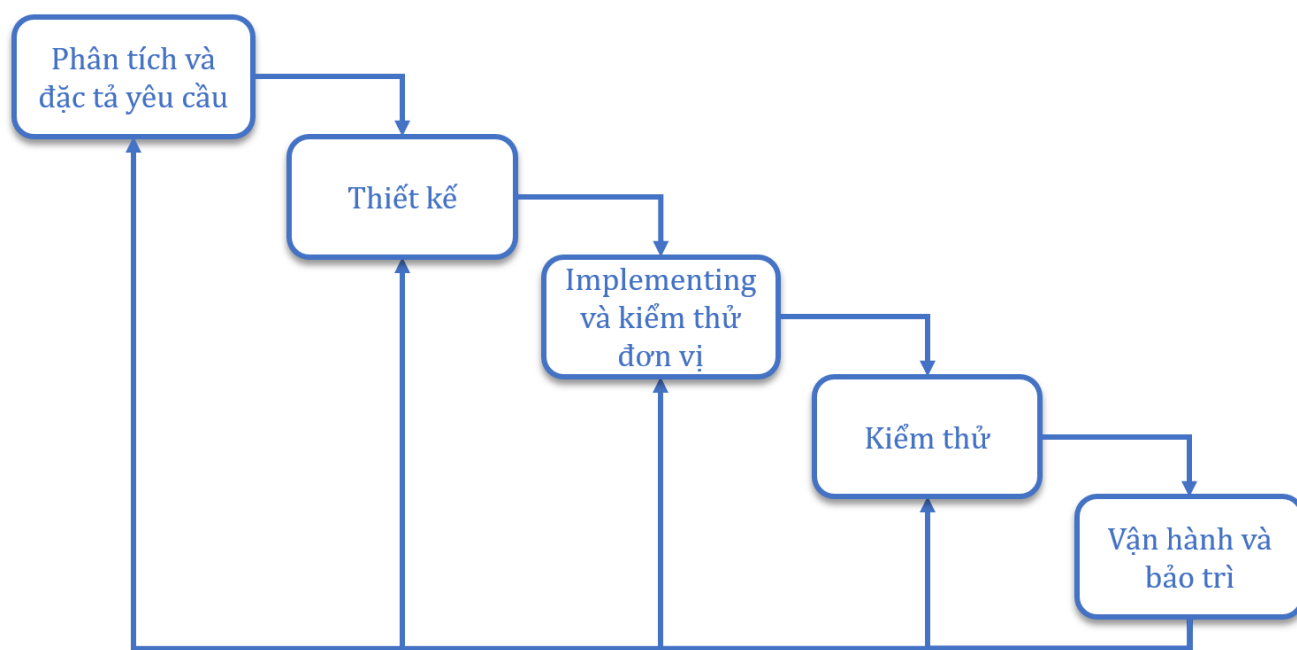
1 Mô hình thác nước (Waterfall model)

1.1 Định nghĩa

Mô hình thác nước là mô hình phát triển phần mềm đầu tiên, sơ khai nhất, tự nhiên nhất và vẫn còn phổ biến đến hiện nay.

Mô hình này chia quá trình phát triển phần mềm thành các giai đoạn tuần tự (các pha), nghĩa là hoàn thành pha này mới tài liệu để chuyển sang được pha tiếp theo. Điều này khá giống với dòng thác nước, nước đổ từ bậc trên xuống bậc dưới vậy, bậc trên phải đầy thì nước mới tràn xuống bậc dưới.

Từ vòng đời của một phần mềm (gồm các bước: phân tích và đặc tả yêu cầu, thiết kế, implementing và kiểm thử đơn vị, kiểm thử, vận hành và bảo trì), ta xây dựng mô hình thác nước như sau:



Hình 1: Mô hình thác nước

Tuy nhiên, không phải pha nào cũng có thể thực hiện một cách hoàn hảo. Lỗi

có thể xảy ra ở bất kỳ pha nào, tại bất kỳ thời điểm nào, và giả sử lỗi được phát hiện là của pha trước đó thì ta cần phải quay lại pha trước đó để sửa. Một điều cần chú ý là để đảm bảo tính tuần tự của mô hình thác nước, khi gặp lỗi, ta phải dừng pha đang làm lại (nói cách khác là dừng cả quy trình lại), lật lại pha trước đó để xem vấn đề ở đâu và sửa lỗi. Điều này có thể gây đến nhiều hệ lụy sẽ được đề cập ở phần nhược điểm bên dưới.

1.2 Ưu điểm

Dễ hiểu, dễ học

Có thể thấy, mô hình thác nước là mô hình cơ bản nhất: chỉ là việc triển khai tuần tự các pha theo các vòng đời phát triển của một phần mềm. Việc triển khai tuần tự nghĩa là các pha được thực hiện một cách rõ ràng, lần lượt, không có sự chồng chéo giữa các pha. Điều này giúp cho việc hiểu và học mô hình này trở nên dễ dàng hơn.

Đảm bảo được phần mềm có chất lượng cao

Khi đã xác định sử dụng mô hình thác nước để phát triển phần mềm thì người ta cũng xác định việc triển khai tuần tự các pha. Nhận thức được điều đó, họ luôn phải cẩn thận trong từng bước bởi rất ít cơ hội để quay lại và sửa lỗi. Mỗi lần làm cần kiểm tra lại, cần có tài liệu đầy đủ. Và mỗi pha cũng cần làm sao cho việc quay lại và sửa lỗi tốn ít thời gian và kinh phí nhất có thể.

Thêm vào đó, mô hình thác nước cũng đảm bảo quy trình dễ quản lý do việc triển khai theo từng pha nên dễ quản lý tiến độ. Pha sau cũng luôn có cái nhìn tổng thể về hệ thống của các pha trước nên mô hình thác nước sẽ có chất lượng cao hơn các mô hình làm từng phần.

1.3 Nhược điểm

Chỉ phù hợp với dự án vừa và nhỏ

Vấn đề này sẽ được phân tích ở bên dưới

Chi phí cao và thời gian lâu

Về chi phí cao, do việc triển khai tuần tự và làm cẩn thận từng pha để phần mềm đầu ra có chất lượng cao nên chi phí sẽ tăng lên.

Về thời gian, do việc triển khai tuần tự nên mỗi thời điểm chỉ có một pha được thực hiện nên thời gian sẽ kéo dài hơn so với các mô hình khác.

Chưa kể nếu có lỗi xảy ra ở pha nào đó, việc dừng quy trình để quay lại pha trước đó sẽ làm tăng thêm thời gian và chi phí.

Phát hiện lỗi ở những bước cuối là thảm họa

Do việc triển khai tuần tự nên nếu có lỗi xảy ra ở pha nào đó, cần dừng quy trình để quay lại pha trước đó. Giả sử lỗi bắt nguồn từ những pha đầu tiên thì ta cần rất nhiều thời gian, chi phí để rework lại các bước trước. Hệ quả là chúng ta có thể hết thời gian giao phần mềm cho khách hàng, hoặc hết kinh phí để hoàn thành dự án.

Kiểm thử muộn và chung chung

Kiểm thử bao gồm có 4 mức: kiểm thử đơn vị, kiểm thử tích hợp, kiểm thử hệ thống và kiểm thử chấp nhận. Việc kiểm thử chỉ thực hiện khi có sản phẩm hay một phần sản phẩm cuối cùng.

Có thể thấy trong mô hình thác nước, lần đầu tiên khách hàng nhìn thấy sản phẩm (ở bước kiểm thử chấp nhận) cũng là thời điểm họ sử dụng phần mềm luôn.

Một nguyên tắc cơ bản của phát triển phần mềm là cố gắng đưa người dùng vào test sớm nhất có thể. Tuy nhiên, với đặc điểm trên thì mô hình thác nước không thể triển khai được nguyên tắc này. Điều đó có thể dẫn đến các hệ lụy sau:

- Nếu phần mềm chưa đáp ứng được yêu cầu của khách hàng hay UI/UX không tốt, thì ta sẽ không kịp sửa cho khách hàng được.

- Thời gian đào tạo người dùng ngắn.

Khi phát triển phần mềm, có nhiều lỗi phát sinh là do khách hàng không biết cách sử dụng phần mềm, chứ không phải do lỗi của phần mềm. Do đó, ta cần đào tạo khách hàng để họ có thể dùng thạo phần mềm. Mức hài lòng của khách hàng cũng phụ thuộc rất nhiều vào việc họ có thể sử dụng phần mềm một cách thành thạo hay không.

Tuy nhiên, trong mô hình thác nước, lần đầu nhìn thấy sản phẩm cũng là lúc khách hàng phải học cách sử dụng nó luôn nên họ rất dễ vấp khi dùng. Điều này khó hơn hẳn với việc họ được liên tục hỏi ý kiến hay nhìn thấy, dùng thử phần mềm trước đó.

1.4 Khi nào sử dụng

2 Mô hình chữ V (V model)

2.1 Định nghĩa

Mô hình chữ V là mô hình phát triển phần mềm được cải tiến từ mô hình thác nước. Về cơ bản, mô hình chữ V cũng triển khai tuần tự các pha như mô hình thác nước. Tuy nhiên, mô hình chữ V cải tiến từ mô hình thác nước ở hai điểm sau:

- **Đẩy việc sinh test lên sớm hơn**

Trong mô hình thác nước, kiểm thử chỉ thực hiện sau pha *implementing* và *kiểm thử đơn vị*. Nói chính xác hơn là sau pha này mới thực hiện việc sinh test và kiểm thử. Đầu ra của các pha *phân tích và đặc tả yêu cầu*, *thiết kế* chỉ có tài liệu cho các pha sau.

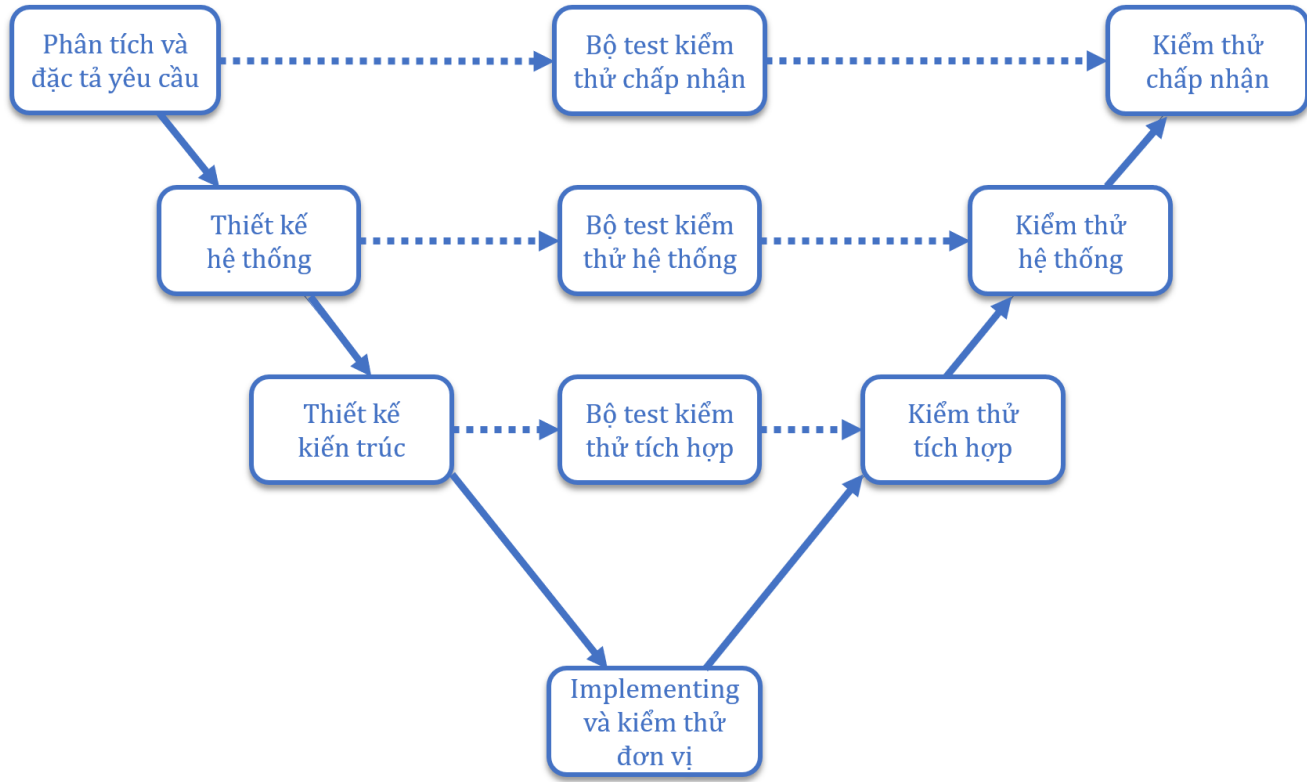
Trong mô hình chữ V, người ta đẩy việc sinh test lên sớm hơn. Ví dụ, với pha *phân tích và đặc tả yêu cầu*, đầu ra sẽ có thêm các bộ test cho pha đó. Sau này khi có sản phẩm hoàn chỉnh để kiểm thử, chỉ cần qua được các bộ test này là sản phẩm đáp ứng được đặc tả yêu cầu. Tương tự với pha *thiết kế*.

- **Phân chia kiểm thử thành nhiều mức**

Trong mô hình thác nước, việc kiểm thử vẫn còn quá chung chung.

Trong mô hình chữ V, người ta phân chia kiểm thử thành nhiều mức:

- Kiểm thử đơn vị
- Kiểm thử tích hợp
- Kiểm thử hệ thống
- Kiểm thử chấp nhận



Hình 2: Mô hình chữ V

2.2 Ưu điểm

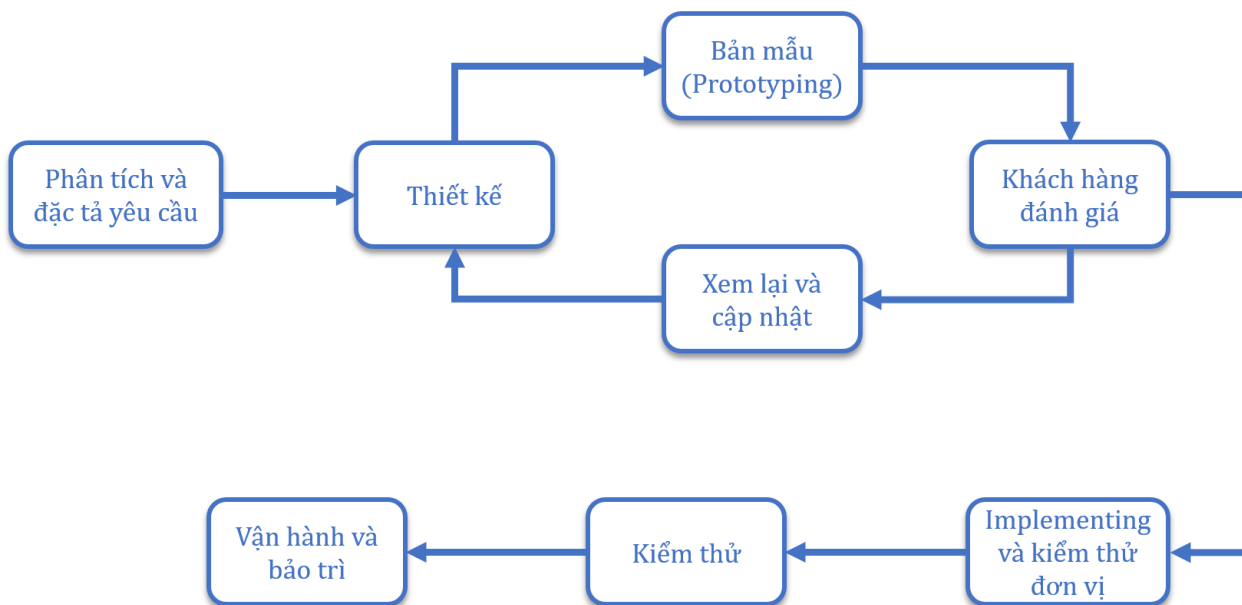
2.3 Nhược điểm

2.4 Khi nào sử dụng

3 Mô hình bản mẫu (Prototype model)

3.1 Định nghĩa

Mô hình bản mẫu là mô hình phát triển phần mềm được xây dựng dựa trên việc tạo ra các bản mẫu (prototype) của phần mềm cho người dùng sử dụng thử.



Hình 3: Mô hình bản mẫu

Khi khách hàng có yêu cầu về phần mềm chưa rõ ràng (dù chỉ là một trong nhiều yêu cầu), thì ta không thể sử dụng mô hình thác nước bởi yêu cầu đó sẽ khiến ta phải dừng pha đặc tả yêu cầu lại cho đến khi yêu cầu rõ ràng.

Ví dụ về một yêu cầu không rõ ràng: Một phần mềm trình chiếu hiệu quả. Nhưng người ta vẫn chưa biết nó là cái gì, nó làm được những gì, nó dùng như thế nào. Để giải quyết vấn đề này, mô hình bản mẫu sẽ liên tục thiết kế các bản mẫu (một giao diện giả của phần mềm) để cho khách hàng xem và đánh giá cho đến khi họ

có thể đưa ra yêu cầu rõ ràng để ta có thể triển khai các pha tiếp theo.

3.2 Ưu điểm

Cách giải quyết yêu cầu ko rõ là trực quan hóa yêu cầu = bản mẫu, giao diện giả để kích thích ngta đưa ra yêu cầu chính xác.

3.3 Nhược điểm

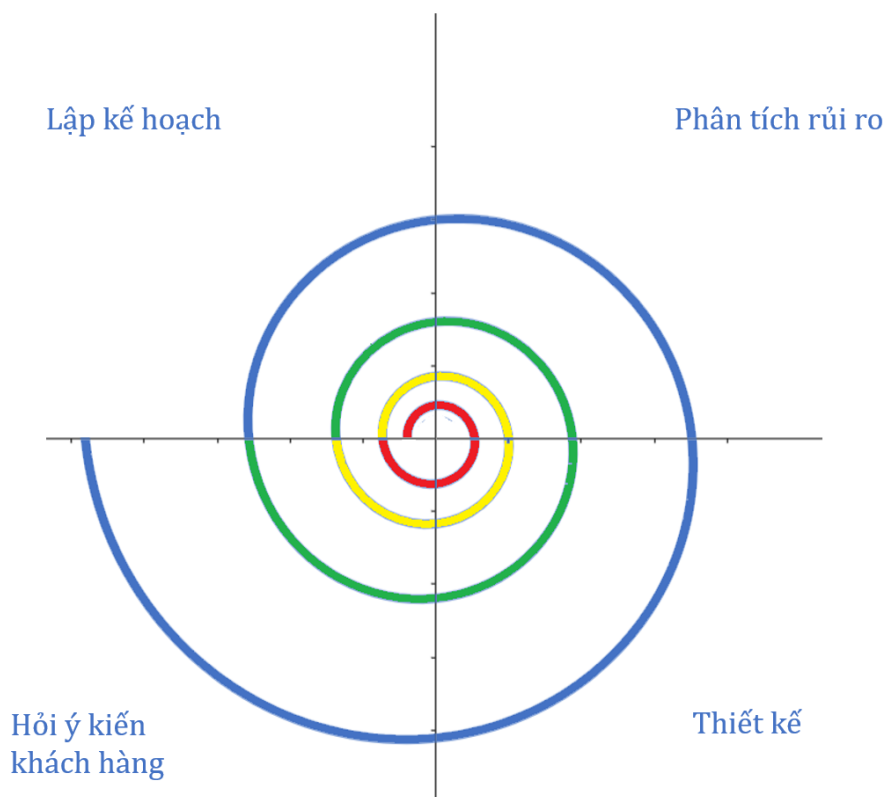
Tuy nhiên, tgian cho bản mẫu nhiều (nên ỏn) -> tgian thiết kế, code, test ít hơn nên CL có vấn đề, tài liệu có thể ko đủ. Ý tưởng hay nhg k đc sử dụng phổ biến vì chỉ cho dự án vừa và nhỏ do chi phí cho bản mẫu lớn. Thực tế ngta dùng ý tưởng bản mẫu -> cách thu thập yêu cầu của họ tg đg vs đi phỏng vấn, questionnaire, quan sát

3.4 Khi nào sử dụng

4 Mô hình xoắn ốc (Spiral model)

4.1 Định nghĩa

Mô hình xoắn ốc là sự kết hợp của mô hình thác nước, mô hình bản mẫu và phân tích rủi ro, được phát triển theo cách tăng dần (incremental).



Hình 4: Mô hình xoắn ốc

Mỗi pha cũng chính là mỗi vòng xoáy bắt đầu bằng việc lập kế hoạch và kết thúc bằng hỏi ý kiến khách hàng. Sau khi kết thúc một vòng xoáy, có thể tiếp tục ở một level cao hơn (vòng xoáy tiếp theo) hoặc dừng lại.

Ý tưởng chính của mô hình này là chia để trị. Chúng ta sẽ làm các tính năng cốt

lõi (core) trước, sau mới đến các tính năng ở mức trung bình (medium level), cuối cùng là các tính năng ít sử dụng.

Mỗi pha là một vòng xoáy, và mỗi vòng xoáy sẽ có các việc sau:

- **Lập kế hoạch**

Lập kế hoạch cho vòng xoáy này, xác định các mục tiêu, rủi ro, chi phí, thời gian, nguồn lực cần thiết.

- **Phân tích rủi ro**

Phân tích rủi ro cho vòng xoáy này, xác định các rủi ro có thể xảy ra và cách xử lý nó. Trả lời câu hỏi có làm không? Làm thì + kế hoạch phòng chống. Dự đoán rủi ro, xác suất, hệ quả. Chống là đưa ra giải pháp. VD khách hàng hay thay đổi yêu cầu (phòng là giảm số lần thay đổi, chống là để hệ thống linh hoạt) -> Khó nhất, cần chuyên gia (rất thâm niên trong quản lý dự án - rất hiếm) -> ít khi dùng, nhg ý tưởng hay, các mô hình sau này có thể thấy hình bóng xoắn ốc.

- **Thiết kế**

- **Hỏi ý kiến khách hàng**

Hỏi ý kiến khách hàng về sản phẩm sau vòng xoáy này.

4.2 Ưu điểm

4.3 Nhược điểm

4.4 Khi nào sử dụng

Hệ thống lớn Cho dự án lớn -> rủi ro cao -> chi phí cho rủi ro lớn, ko còn chi phí phát triển, software fail