

# BÀI TẬP BUỔI 3

## I. Trắc nghiệm

Câu hỏi 1: Pha nào trong mô hình lý thuyết vòng đời phát triển phần mềm chịu trách nhiệm chuyển đổi yêu cầu thành đặc tả kỹ thuật?

Đáp án: C. Pha phân tích

Câu hỏi 2: Mô hình vòng đời nào phát triển phần mềm bằng cách tạo các phiên bản nhỏ và tăng dần tính năng?

Đáp án: B. Mô hình lặp và tăng trưởng

Câu hỏi 3: Pha bảo trì trong vòng đời phát triển phần mềm bao gồm hoạt động nào?

Đáp án: B. Sửa lỗi và cập nhật tính năng mới

Câu hỏi 4: Mô hình thác nước phù hợp nhất với loại dự án nào?

Đáp án: B. Dự án có yêu cầu rõ ràng và ít thay đổi

Câu hỏi 5: Trong mô hình xoắn ốc, mỗi vòng xoắn tương ứng với?

Đáp án: B. Một chu kỳ lặp của toàn bộ quy trình phát triển

Câu hỏi 6: Điểm yếu lớn nhất của mô hình xây và sửa là gì?

Đáp án: C. Khó kiểm soát chất lượng

Câu hỏi 7: Mô hình nào tập trung vào việc tạo các nguyên mẫu nhanh để thu thập phản hồi từ khách hàng?

Đáp án: B. Mô hình bản mẫu nhanh

Câu hỏi 8: Pha nào kết thúc vòng đời phát triển phần mềm?

Đáp án: C. Pha giải thể

Câu hỏi 9: Điểm khác biệt chính giữa mô hình lặp và tăng trưởng với mô hình thác nước là gì?

Đáp án: B. Mô hình lặp và tăng trưởng phát triển theo từng đợt nhỏ

Câu hỏi 10: Mô hình nào có khả năng thích nghi tốt nhất với sự thay đổi của yêu cầu khách hàng?

Đáp án: D. Mô hình tiến trình linh hoạt

## II. Câu hỏi ngắn

1. Pha lấy yêu cầu là gì và có vai trò gì trong vòng đời phát triển phần mềm?

- Pha lấy yêu cầu là giai đoạn thu thập và phân tích các yêu cầu từ khách hàng để hiểu rõ những gì phần mềm cần thực hiện.
  - Vai trò: Xác định các yêu cầu chức năng và phi chức năng, lập tài liệu yêu cầu phần mềm (SRS)
2. Mô hình thác nước hoạt động như thế nào?
    - Phát triển phần mềm theo quy trình tuyến tính, từng pha hoàn thành trước khi chuyển sang pha tiếp theo.
    - Gồm các pha: Lấy yêu cầu → Phân tích → Thiết kế → Cài đặt → Kiểm thử → Bảo trì
  3. Mô hình lặp và tăng trưởng khác gì so với mô hình thác nước?
    - Mô hình lặp và tăng trưởng phát triển theo từng đợt lặp lại, cải tiến dần phần mềm, giúp giảm thiểu rủi ro và thích nghi tốt với thay đổi yêu cầu.
    - Mô hình thác nước phát triển theo trình tự cố định, khó thay đổi khi yêu cầu thay đổi
  4. Mục tiêu của pha bảo trì là gì?
    - Đảm bảo phần mềm tiếp tục hoạt động ổn định và đáp ứng các yêu cầu mới.
    - Hoạt động chính: Sửa lỗi, cải tiến tính năng, điều chỉnh phần mềm để tương thích với môi trường mới
  5. Mô hình xây và sửa có nhược điểm gì?
    - Phát triển nhanh nhưng không có quy trình rõ ràng, dễ phát sinh lỗi khi mở rộng.
    - Nhược điểm chính: Khó bảo trì và nâng cấp
  6. Mô hình bản mẫu nhanh là gì?
    - Là mô hình phát triển phần mềm bằng cách tạo các nguyên mẫu nhanh để thu thập phản hồi từ khách hàng.
    - Ưu điểm: Giúp khách hàng hình dung rõ hơn về sản phẩm.
    - Nhược điểm: Tốn thời gian và chi phí xây dựng nhiều bản mẫu
  7. Pha giải thể là gì?
    - Là pha kết thúc vòng đời phần mềm khi phần mềm không còn giá trị sử dụng.
    - Hoạt động chính: Gỡ bỏ phần mềm, lưu trữ tài liệu và dữ liệu cần thiết, đảm bảo bảo mật dữ liệu
  8. Mô hình xoắn ốc là gì?
    - Kết hợp giữa mô hình thác nước và lặp, mỗi vòng xoắn là một pha phát triển.
    - Ưu điểm: Tập trung vào quản lý rủi ro và thích nghi tốt với thay đổi.
    - Nhược điểm: Phức tạp và tốn kém
  9. Tại sao mô hình tiến trình linh hoạt được đánh giá cao?

Vì có khả năng thích nghi tốt với thay đổi, tập trung vào phản hồi nhanh từ khách hàng và cải thiện chất lượng phần mềm

10. Điểm khác biệt chính giữa mô hình mã nguồn mở và các mô hình khác là gì?
  - Mô hình mã nguồn mở phát triển với sự đóng góp từ cộng đồng, có nguồn lực phát triển dồi dào và chi phí thấp.

- Nhược điểm: Khó kiểm soát chất lượng và tiến độ

### **III. Câu hỏi thảo luận nhóm**

#### **1. So sánh ưu và nhược điểm của mô hình thác nước và mô hình xoắn ốc**

- Mô hình thác nước (Waterfall Model)
  - Ưu điểm:

Quy trình rõ ràng, dễ hiểu và dễ quản lý.

Thích hợp cho các dự án có yêu cầu rõ ràng, ít thay đổi.

Dễ dàng theo dõi tiến độ, mỗi giai đoạn hoàn thành trước khi chuyển sang giai đoạn tiếp theo.

- Nhược điểm:

Rất khó thay đổi yêu cầu khi đã bước sang giai đoạn tiếp theo.

Không phản hồi được nhanh khi khách hàng yêu cầu thay đổi.

Không thích hợp cho dự án có rủi ro cao hoặc yêu cầu phức tạp

- Mô hình xoắn ốc (Spiral Model)
  - Ưu điểm:

Tập trung vào quản lý rủi ro, đánh giá rủi ro ở mỗi vòng xoắn.

Linh hoạt hơn so với thác nước, có thể điều chỉnh dự án sau từng vòng lặp.

Kết hợp cả mô hình thác nước và mô hình lặp, thích hợp cho dự án lớn.

- Nhược điểm:

Tốn kém chi phí do phải thực hiện nhiều vòng lặp.

Đòi hỏi đội ngũ có kỹ năng cao để đánh giá và kiểm soát rủi ro

#### **2. Tình huống thực tế có thể áp dụng mô hình lặp và tăng trưởng**

- Mô hình lặp và tăng trưởng (Iterative and Incremental Model) phù hợp với các dự án cần phát triển theo từng phiên bản nhỏ và cải thiện dần theo phản hồi khách hàng.

- Tình huống thực tế:
  - Phát triển phần mềm thương mại điện tử (E-commerce)
  - Ban đầu chỉ triển khai các tính năng cơ bản như đăng ký, đăng nhập, hiển thị sản phẩm.
  - Qua từng phiên bản, bổ sung các tính năng như thanh toán trực tuyến, gợi ý sản phẩm, chatbot hỗ trợ khách hàng.
  - Mỗi đợt phát triển giúp điều chỉnh phần mềm theo nhu cầu thực tế

### 3. Tại sao mô hình xây và sửa không phù hợp với các dự án lớn?

Mô hình xây và sửa (Build and Fix Model) thường chỉ áp dụng cho các dự án nhỏ, không có quy trình rõ ràng.

- Lý do không phù hợp với dự án lớn:
  - Không có quy trình chặt chẽ → Dự án lớn cần có kế hoạch chi tiết để tránh sai sót.
  - Dễ phát sinh lỗi → Việc sửa lỗi liên tục khiến chi phí bảo trì tăng cao.
  - Không mở rộng được → Khi thêm tính năng mới, hệ thống dễ bị lỗi hoặc không hoạt động như mong đợi

### 4. So sánh giữa mô hình bản mẫu nhanh và mô hình tiến trình linh hoạt

Tiêu chí	Mô hình bản mẫu nhanh (Rapid Prototyping)	Mô hình tiến trình linh hoạt (Agile)
----------	---	--------------------------------------

Mục đích	Tạo mẫu nhanh để thu thập phản hồi	Phát triển theo từng phần nhỏ, linh hoạt
----------	------------------------------------	--

Ưu điểm	Giúp khách hàng hình dung sản phẩm trước khi phát triển	Tích hợp phản hồi nhanh, liên tục cải tiến
---------	---	--

Nhược điểm	Tốn thời gian, chi phí để tạo nhiều bản mẫu	Yêu cầu phối hợp chặt chẽ trong nhóm phát triển
------------	---	---

Ứng dụng	Phần mềm cần thiết kế giao diện phức tạp, ví dụ: app di động, web Dự án yêu cầu thay đổi liên tục, ví dụ: phần mềm quản lý doanh nghiệp	
----------	--	--

## 5. Phân tích vai trò của quản lý rủi ro trong mô hình xoắn ốc

- Quản lý rủi ro là yếu tố quan trọng nhất trong mô hình xoắn ốc.
- Mỗi vòng xoắn trong mô hình giúp đánh giá và kiểm soát rủi ro trước khi bước vào pha phát triển tiếp theo.
- Các hoạt động chính của quản lý rủi ro:
  - Xác định rủi ro: Dự đoán vấn đề có thể xảy ra (ví dụ: thay đổi yêu cầu, công nghệ mới).
  - Lập kế hoạch giảm rủi ro: Đề xuất giải pháp xử lý các rủi ro có thể gặp.
  - Kiểm soát rủi ro: Điều chỉnh kế hoạch phát triển khi phát hiện vấn đề

## 6. Khi nào nên sử dụng mô hình thác nước thay vì mô hình tiến trình linh hoạt?

- Khi yêu cầu dự án rõ ràng ngay từ đầu, ít có thay đổi.
- Khi cần quản lý chặt chẽ về thời gian và ngân sách.
- Khi đội ngũ phát triển chưa có nhiều kinh nghiệm với phương pháp linh hoạt.

Ví dụ: Phát triển phần mềm nhúng trong thiết bị y tế, vì cần sự chính xác cao

## 7. Những khó khăn khi áp dụng mô hình mã nguồn mở

- Khó kiểm soát chất lượng: Do có nhiều người tham gia phát triển từ nhiều nơi khác nhau.
- Thiếu tài chính ổn định: Nhiều dự án mã nguồn mở không có nguồn vốn rõ ràng.
- Quản lý bảo mật kém: Không phải ai cũng có chuyên môn cao về bảo mật

## 8. Mô hình tiến trình linh hoạt giúp cải thiện chất lượng phần mềm như thế nào?

- Tích hợp phản hồi liên tục từ khách hàng → Điều chỉnh phần mềm nhanh chóng.
- Kiểm thử thường xuyên → Giúp phát hiện lỗi sớm hơn.
- Phát triển theo từng phần nhỏ → Mỗi phiên bản đều được tối ưu hóa trước khi đưa vào sử dụng

## 9. Vai trò của pha bảo trì trong vòng đời phát triển phần mềm

- Đảm bảo phần mềm tiếp tục hoạt động ổn định sau khi triển khai.

- Sửa lỗi, cập nhật tính năng mới theo nhu cầu khách hàng.
- Điều chỉnh phần mềm để tương thích với công nghệ mới

10. Mô hình vòng đời phù hợp cho dự án phát triển phần mềm ngân hàng và lý do

Mô hình phù hợp: Mô hình xoắn ốc (Spiral Model)

- Lý do:
  - Yêu cầu bảo mật cao: Ngân hàng xử lý dữ liệu nhạy cảm, cần đánh giá rủi ro liên tục.
  - Cần phát triển nhiều giai đoạn: Ngân hàng có nhiều nghiệp vụ phức tạp (quản lý tài khoản, giao dịch, bảo mật).
  - Dễ dàng mở rộng: Mô hình xoắn ốc cho phép điều chỉnh và bổ sung tính năng mới theo từng giai đoạn

#### **IV. Câu hỏi tình huống**

Tình huống 1: Giải pháp cho yêu cầu thay đổi sau pha thiết kế trong mô hình thác nước:

Đội phát triển cần thảo luận với khách hàng để hiểu rõ yêu cầu thay đổi và đánh giá mức độ ảnh hưởng của chúng đến tiến độ và ngân sách. Nếu thay đổi nhỏ, có thể áp dụng các biện pháp điều chỉnh trong kế hoạch. Nếu thay đổi lớn, có thể phải xem xét lại toàn bộ kế hoạch dự án, bao gồm cả thời gian, chi phí và nguồn lực.

Tình huống 2: Giải pháp cho thiếu nhân lực trong dự án phát triển theo mô hình lặp và tăng trưởng:

Là quản lý dự án, bạn cần phân tích và đánh giá lại tiến độ công việc, ưu tiên các tính năng quan trọng và có thể cắt giảm những phần không cần thiết. Cần nhắc tuyển thêm nhân sự hoặc hợp tác với các đối tác ngoài để tăng cường nguồn lực.

Tình huống 3: Giải pháp cho thiếu phản hồi của khách hàng trong mô hình tiến trình linh hoạt:

Đội phát triển nên chủ động liên lạc với khách hàng, tạo các cuộc họp nhanh để thu thập phản hồi kịp thời. Nếu khách hàng vẫn không phản hồi, đội phát triển có thể đưa ra các giả định hợp lý và tiếp tục phát triển, đồng thời chuẩn bị kế hoạch điều chỉnh khi có phản hồi từ khách hàng.

Tình huống 4: Giải pháp cho yêu cầu tính năng mới trong pha cài đặt:

Trong trường hợp này, mô hình lặp hoặc tiến trình linh hoạt sẽ phù hợp, vì chúng cho phép phát triển và thay đổi linh hoạt theo yêu cầu của khách hàng ngay cả trong các giai đoạn muộn của dự án.

Tình huống 5: Giải pháp cho công ty nhỏ thiếu nguồn lực khi áp dụng mô hình bản mẫu nhanh:

Công ty có thể xem xét sử dụng các công cụ và nền tảng có sẵn để giảm thiểu chi phí phát triển ban đầu, hoặc hợp tác với các công ty khác hoặc thuê ngoài một số phần công việc. Cũng có thể áp dụng mô hình phát triển theo chu kỳ ngắn để tiết kiệm thời gian và chi phí.

Tình huống 6: Mô hình phù hợp cho dự án phần mềm thương mại điện tử yêu cầu thay đổi giao diện liên tục:

Mô hình tiến trình linh hoạt (Agile) sẽ là sự lựa chọn tốt nhất, vì mô hình này cho phép thay đổi giao diện và tính năng liên tục trong suốt quá trình phát triển phần mềm.

Tình huống 7: Mô hình phát triển phần mềm cho dự án lớn với nhiều nhóm phát triển ở các quốc gia khác nhau:

Mô hình xoắn ốc hoặc mô hình phát triển theo quy trình phát triển toàn cầu sẽ phù hợp, vì chúng cho phép các nhóm phát triển ở các quốc gia khác nhau làm việc đồng thời và linh hoạt điều chỉnh các giai đoạn của dự án.

Tình huống 8: Giải pháp giảm thiểu rủi ro khi áp dụng mô hình xoắn ốc:

Để giảm thiểu rủi ro, đội phát triển cần xác định rõ các yếu tố rủi ro ngay từ giai đoạn đầu và xây dựng kế hoạch giảm thiểu. Các bước kiểm tra, thử nghiệm và xác minh cần được thực hiện liên tục trong mỗi chu kỳ để phát hiện và giải quyết vấn đề ngay khi phát sinh.

Tình huống 9: Mô hình phát triển phần mềm cho dự án ngân hàng yêu cầu bảo mật cao:

Mô hình phát triển theo chu trình kiểm thử nghiêm ngặt hoặc mô hình vững chắc như mô hình V-model sẽ đảm bảo việc kiểm tra và xác thực an toàn bảo mật trong suốt quá trình phát triển.

Tình huống 10: Thời điểm kết thúc vòng đời phần mềm và thực hiện pha giải thể:

Vòng đời phần mềm nên kết thúc khi sản phẩm đã đáp ứng đầy đủ các yêu cầu của khách hàng và các mục tiêu dự án. Pha giải thể nên diễn ra sau khi phần mềm đã được triển khai và đưa vào sử dụng, và khi tất cả các vấn đề bảo trì, hỗ trợ và các tài liệu cần thiết đã được hoàn tất.