

01. CÂU HỎI TRẮC NGHIỆM

1. Câu hỏi 1: Workflow nào trong tiến trình phát triển phần mềm chịu trách nhiệm thu thập yêu cầu từ khách hàng?

- A. Workflow thiết kế
- B. Workflow lấy yêu cầu**
- C. Workflow kiểm thử
- D. Workflow triển khai

2. Câu hỏi 2: Pha nào trong tiến trình thống nhất (Unified Process) tập trung vào việc phân tích rủi ro và xây dựng kiến trúc ban đầu?

- A. Pha khởi đầu
- B. Pha làm rõ**
- C. Pha xây dựng
- D. Pha chuyển giao

3. Câu hỏi 3: Mô hình CMM mức nào yêu cầu quy trình phát triển phần mềm phải được quản lý định lượng?

- A. Mức 2
- B. Mức 3
- C. Mức 4**
- D. Mức 5

4. Câu hỏi 4: Các pha trong tiến trình thống nhất bao gồm:

- A. Lấy yêu cầu, phân tích, thiết kế, kiểm thử
- B. Khởi đầu, làm rõ, xây dựng, chuyển giao**
- C. Lập kế hoạch, thiết kế, phát triển, bảo trì
- D. Phân tích, kiểm thử, triển khai, bảo trì

5. Câu hỏi 5: Trong tiến trình thống nhất, workflow nào thực hiện sau cùng?

- A. Workflow phân tích
- B. Workflow thiết kế

C. Workflow cài đặt

D. Workflow kiểm thử 49

6. Câu hỏi 6: Mô hình CMM mức 1 có đặc điểm gì?

A. Quy trình được định nghĩa rõ ràng

B. Quy trình được kiểm soát và đo lường

C. Quy trình không ổn định, phụ thuộc vào cá nhân

D. Quy trình liên tục được tối ưu hóa

7. Câu hỏi 7: Tiến trình thống nhất là một ví dụ của mô hình nào?

A. Mô hình vòng đời thác nước

B. Mô hình lặp và tăng trưởng

C. Mô hình mã nguồn mở

D. Mô hình Agile

8. Câu hỏi 8: Trong mô hình CMM mức 5, quy trình phát triển phần mềm có đặc điểm gì?

A. Quy trình được cải tiến liên tục

B. Quy trình chỉ định nghĩa cơ bản

C. Quy trình chưa được quản lý

D. Quy trình chỉ tập trung vào bảo trì

9. Câu hỏi 9: Workflow thiết kế bao gồm việc thực hiện hoạt động nào?

A. Thu thập yêu cầu

B. Lập kế hoạch dự án

C. Thiết kế kiến trúc và chi tiết hệ thống

D. Kiểm thử tích hợp

10. Câu hỏi 10: CMM viết tắt của cụm từ nào?

A. Configuration Management Model

B. Capability Maturity Model

C. Continuous Maintenance Model

02. CÂU HỎI NGẮN

1. Pha khởi đầu trong tiến trình thống nhất là gì?

Pha khởi đầu trong tiến trình thống nhất: Pha này tập trung vào việc xác định các yêu cầu và các yếu tố cơ bản của dự án, thiết lập phạm vi, phân tích tính khả thi, và phát triển một kế hoạch tổng thể. Nó cũng giúp xác định kiến trúc ban đầu của hệ thống và lập kế hoạch tài nguyên.

2. Mục tiêu của workflow lấy yêu cầu là gì?

Mục tiêu của workflow lấy yêu cầu: Mục tiêu của workflow này là thu thập thông tin về yêu cầu của khách hàng, xác định các yêu cầu chức năng và phi chức năng của hệ thống, và đảm bảo rằng tất cả các yêu cầu cần thiết đã được ghi nhận và hiểu rõ.

3. Tiến trình thống nhất gồm bao nhiêu pha chính?

☐ Tiến trình thống nhất gồm bao nhiêu pha chính: Tiến trình thống nhất gồm 4 pha chính, đó là:

- Pha khởi đầu (Inception):

Mục tiêu chính là hiểu rõ các yêu cầu cơ bản và đánh giá tính khả thi của dự án.

- Pha làm rõ (Elaboration):

Tập trung vào phân tích và thiết kế hệ thống, giải quyết các rủi ro lớn.

- Pha xây dựng (Construction):

Triển khai các module phần mềm và kiểm thử chúng.

- Pha chuyển giao (Transition):

Phần mềm được triển khai và bàn giao cho khách hàng sử dụng.

4. Sự khác nhau giữa CMM mức 2 và mức 3 là gì?

- Mức 2 (Quản lý quy trình) yêu cầu quy trình phát triển phần mềm phải được quản lý và kiểm soát, đảm bảo rằng các dự án được triển khai đúng cách và theo lịch trình.
- Mức 3 (Mô hình chuẩn hóa) yêu cầu quy trình phát triển phần mềm được chuẩn hóa và cải tiến liên tục trong toàn bộ tổ chức.

5. Workflow kiểm thử có nhiệm vụ gì?

Workflow kiểm thử đảm nhận nhiệm vụ kiểm tra và xác nhận rằng hệ thống phát triển hoạt động đúng theo yêu cầu đã thu thập. Nó bao gồm kiểm thử chức năng, kiểm thử tích hợp, kiểm thử hiệu suất và kiểm thử bảo mật.

6. Mô hình CMM có bao nhiêu mức?:

Mô hình CMM có 5 mức:

- Mức 1: Initial (Ban đầu) :

Quy trình không ổn định, phụ thuộc nhiều vào cá nhân.

Phần lớn dự án thành công nhờ vào nỗ lực cá nhân thay vì tổ chức.

- Mức 2: Managed (Quản lý):

Quy trình được quản lý ở mức cơ bản.

Các hoạt động như lập kế hoạch, quản lý rủi ro được thực hiện.

- Mức 3: Defined (Định nghĩa)

Quy trình được định nghĩa rõ ràng và nhất quán trong toàn tổ chức.

Các tiêu chuẩn quy trình được xây dựng và áp dụng.

- Mức 4: Quantitatively Managed (Quản lý định lượng) :

Quy trình được đo lường và kiểm soát bằng dữ liệu định lượng.

Tập trung vào việc giảm thiểu sai sót và cải tiến quy trình.

- Mức 5: Optimizing (Tối ưu hóa):

Quy trình liên tục được cải tiến dựa trên phản hồi và dữ liệu.

Mục tiêu là đạt được sự hoàn hảo trong phát triển phần mềm.

7. Khác biệt giữa mô hình thác nước và mô hình lặp là gì?

- Mô hình thác nước là một mô hình phát triển phần mềm tuyến tính, trong đó mỗi pha phải hoàn thành trước khi chuyển sang pha tiếp theo.
- Mô hình lặp là một mô hình phát triển phần mềm linh hoạt, trong đó các pha được lặp lại và cải tiến qua từng vòng, cho phép thay đổi yêu cầu trong suốt quá trình phát triển.

8. Tiến trình thống nhất có phải là mô hình lặp không?

Có, tiến trình thống nhất là một mô hình lặp và tăng trưởng, trong đó các pha sẽ được lặp lại qua nhiều vòng để cải tiến và hoàn thiện sản phẩm.

9. Mục đích của workflow thiết kế là gì?

Mục đích của workflow thiết kế: Mục đích của workflow thiết kế là tạo ra các bản thiết kế kiến trúc và chi tiết cho hệ thống, đảm bảo rằng các yêu cầu đã được chuyển thành các giải pháp kỹ thuật khả thi và dễ triển khai.

10.CMM mức 5 tập trung vào điều gì?

CMM mức 5 tập trung vào việc cải tiến quy trình phát triển phần mềm liên tục, thông qua việc thu thập dữ liệu, phân tích và áp dụng các biện pháp cải tiến nhằm nâng cao chất lượng và hiệu suất của quy trình.

3. CÂU HỎI THẢO LUẬN NHÓM

1. Thảo luận về vai trò của từng workflow trong tiến trình phát triển phần mềm.

Trong tiến trình phát triển phần mềm, workflow đóng vai trò quan trọng trong việc đảm bảo từng bước của dự án được thực hiện có hệ thống và đạt hiệu quả cao. Các workflow chính bao gồm:

- Workflow lấy yêu cầu: Thu thập, phân tích và xác định yêu cầu từ khách hàng, giúp định hướng phát triển phần mềm theo đúng mong muốn của người dùng.
- Workflow phân tích: Xác định cách thức hệ thống sẽ hoạt động dựa trên yêu cầu đã thu thập, giúp xây dựng mô hình logic cho phần mềm.
- Workflow thiết kế: Chuyển đổi các yêu cầu và mô hình phân tích thành bản thiết kế kiến trúc và chi tiết, đảm bảo hệ thống có cấu trúc hợp lý, dễ bảo trì.
- Workflow cài đặt: Hiện thực hóa thiết kế bằng cách lập trình, viết mã nguồn, và tích hợp các thành phần của hệ thống.
- Workflow kiểm thử: Đảm bảo hệ thống hoạt động đúng yêu cầu bằng cách thực hiện các kiểm thử đơn vị, tích hợp, hệ thống và nghiệm thu.
- Workflow triển khai: Đưa sản phẩm vào môi trường thực tế và cung cấp hướng dẫn sử dụng, bảo trì phần mềm sau khi bàn giao.

2. Phân biệt mô hình vòng đời thác nước và tiến trình thống nhất.

Tiêu chí	Mô hình thác nước	Tiến trình thống nhất
Cách tiếp cận	Tuyến tính, từng bước, không quay lại pha trước	Lặp lại các pha để cải tiến liên tục
Tính linh hoạt	Cứng nhắc, khó thay đổi yêu cầu	Linh hoạt, có thể thay đổi trong suốt quá trình phát triển
Quy trình phát triển	Yêu cầu → Thiết kế → Cài đặt → Kiểm thử → Triển khai	Khởi đầu → Làm rõ → Xây dựng → Chuyển giao
Ứng dụng	Phù hợp cho dự án có yêu cầu cố định	Phù hợp cho dự án lớn, phức tạp, yêu cầu thay đổi

3. Thảo luận về các ưu và nhược điểm của mô hình lặp và tăng trưởng.

Ưu điểm:

Linh hoạt, dễ thích nghi với thay đổi của khách hàng.

Giúp phát hiện lỗi sớm, giảm rủi ro.

Dễ dàng cải tiến sản phẩm qua từng vòng lặp.

Nhược điểm:

Có thể kéo dài thời gian phát triển do liên tục thay đổi.

Cần sự phối hợp tốt giữa các nhóm để đảm bảo tiến độ.

Khó xác định chi phí và thời gian hoàn thành chính xác.

4. Vì sao mô hình CMM được sử dụng rộng rãi trong quản lý chất lượng phần mềm?

- Chuẩn hóa quy trình phát triển phần mềm, giúp tăng hiệu suất và đảm bảo chất lượng.
- Cung cấp khung làm việc chi tiết giúp các tổ chức kiểm soát và cải tiến quy trình.
- Tạo điều kiện đo lường hiệu suất, giúp đánh giá và nâng cao năng suất đội nhóm.
- Nâng cao độ tin cậy của phần mềm, giảm lỗi và chi phí bảo trì.

5. Thảo luận về các khó khăn khi áp dụng mô hình CMM trong thực tế.

- Chi phí triển khai cao, đòi hỏi nguồn lực lớn.
- Cần sự cam kết từ lãnh đạo để thực hiện thay đổi trong tổ chức.
- Khó áp dụng cho các nhóm nhỏ, vì yêu cầu quy trình phức tạp.
- Cần đào tạo và tuân thủ nghiêm ngặt, làm chậm quá trình phát triển ban đầu.

6. Đề xuất các giải pháp để cải tiến quy trình phát triển phần mềm.

Áp dụng DevOps để tự động hóa kiểm thử và triển khai.

Kết hợp Agile với CMM để đảm bảo linh hoạt nhưng vẫn có quy trình chặt chẽ.

Sử dụng công cụ quản lý dự án như Jira, Trello để theo dõi tiến độ.

Xây dựng văn hóa cải tiến liên tục, thu thập phản hồi từ khách hàng và nhóm phát triển.

7. Phân tích ưu điểm của việc áp dụng tiến trình thống nhất trong các dự án lớn.

Linh hoạt, giúp điều chỉnh yêu cầu trong suốt quá trình phát triển.

Quản lý rủi ro tốt, nhờ vào việc lặp lại nhiều pha.

Dễ mở rộng và tích hợp, phù hợp với phần mềm có quy mô lớn.

Hỗ trợ kiểm soát tiến độ và chất lượng, nhờ vào workflow rõ ràng.

8. Thảo luận về sự cần thiết của việc kiểm thử trong từng pha của tiến trình thống nhất.

- Pha khởi đầu: Xác minh yêu cầu có đầy đủ và hợp lệ không.
- Pha làm rõ: Kiểm tra thiết kế kiến trúc, đảm bảo đáp ứng yêu cầu.
- Pha xây dựng: Thực hiện kiểm thử đơn vị và kiểm thử tích hợp.

- Pha chuyển giao: Kiểm thử hệ thống, kiểm thử chấp nhận để đảm bảo phần mềm sẵn sàng triển khai.

9. So sánh giữa mô hình CMM mức 4 và mức 5.

Tiêu chí	Mức 4 (Quản lý định lượng)	Mức 5 (Tối ưu hóa)
Quy trình	Được đo lường và kiểm soát định lượng	Được cải tiến liên tục
Mục tiêu	Đảm bảo sự ổn định và kiểm soát hiệu suất	Không ngừng tối ưu và đổi mới
Cách tiếp cận	Dựa trên dữ liệu thực tế để quản lý quy trình	Tích cực tìm cách nâng cao hiệu suất hệ thống
Mức độ cải tiến	Cải tiến dựa trên số liệu thu thập	Chủ động tối ưu bằng công nghệ mới

10. Đề xuất cách tổ chức hoạt động nhóm trong workflow lấy yêu cầu

Bước 1 - Xác định yêu cầu sơ bộ:

- Nhóm BA (Business Analyst) làm việc với khách hàng để thu thập nhu cầu.

Bước 2 - Phân tích yêu cầu chi tiết:

- Nhóm kỹ thuật và QA xác minh tính khả thi của yêu cầu.

Bước 3 - Xây dựng tài liệu yêu cầu:

- Nhóm BA tổng hợp và tạo tài liệu SRS (Software Requirement Specification).

Bước 4 - Xác nhận yêu cầu:

- Trình bày với khách hàng để xác nhận và điều chỉnh nếu cần.

Bước 5 - Chuyển giao cho nhóm phát triển:

- Chuyển tài liệu yêu cầu cho nhóm phát triển và kiểm thử.

4. CÂU HỎI TÌNH HUỐNG

1. Một công ty phát triển phần mềm gặp khó khăn khi yêu cầu của khách hàng liên tục thay đổi trong pha xây dựng. Đội phát triển nên làm gì để giải quyết vấn đề này?

Linh hoạt áp dụng Agile/Scrum, tổ chức các cuộc họp ngắn (daily standup) để cập nhật yêu cầu từ khách hàng. Đồng thời, sử dụng backlog để quản lý thay đổi và ưu tiên công việc sao cho hợp lý.

2. Trong pha chuyển giao của tiến trình thống nhất, khách hàng yêu cầu bổ sung thêm tính năng mới. Đội phát triển nên xử lý ra sao?

Đánh giá tác động của yêu cầu mới, xem xét ảnh hưởng đến tiến độ và ngân sách. Nếu thay đổi lớn, thương lượng với khách hàng để triển khai trong phiên bản sau. Nếu thay đổi nhỏ, có thể xem xét bổ sung mà không ảnh hưởng đến tiến độ chính.

3. Dự án phát triển phần mềm bị trễ tiến độ do lỗi phát sinh liên tục trong quá trình kiểm thử. Là trưởng dự án, bạn sẽ làm gì?

Tăng cường kiểm thử tự động và kiểm thử sớm (shift-left testing) để phát hiện lỗi ngay từ giai đoạn đầu. Đồng thời, tổ chức các cuộc họp đánh giá lỗi, phân tích nguyên nhân gốc rễ để ngăn chặn lỗi lặp lại, giúp kiểm soát tiến độ dự án.

4. Trong workflow thiết kế, kiến trúc sư phần mềm muốn thay đổi thiết kế ban đầu để cải thiện hiệu suất. Đội phát triển nên xử lý thế nào?

Thảo luận kỹ lưỡng với nhóm phát triển, đánh giá lợi ích của việc thay đổi thiết kế so với rủi ro có thể phát sinh. Nếu thay đổi giúp tối ưu hiệu suất đáng kể, có thể đề xuất khách hàng phê duyệt và cập nhật lại tài liệu thiết kế.

5. Khách hàng yêu cầu rút ngắn thời gian phát triển dự án mà không thay đổi yêu cầu. Đội phát triển nên phản ứng ra sao?

Tối ưu hóa quy trình phát triển bằng cách sử dụng phát triển song song, tự động hóa, và loại bỏ các công đoạn không cần thiết. Nếu vẫn không đủ thời gian, thảo luận với khách hàng về việc cắt giảm phạm vi hoặc bổ sung nhân sự.

6. Một công ty nhỏ muốn áp dụng mô hình CMM nhưng gặp khó khăn do thiếu nguồn lực. Hãy đề xuất giải pháp.

Bắt đầu với CMM ở mức thấp như mức 2 hoặc 3, tập trung vào các quy trình quan trọng trước, sau đó dần dần cải tiến theo thời gian. Sử dụng các công cụ hỗ trợ quản lý quy trình như Jira, Trello để giảm gánh nặng tài nguyên.

7. Trong workflow lấy yêu cầu, khách hàng cung cấp thông tin không rõ ràng. Đội phát triển cần làm gì?

Sử dụng phương pháp phỏng vấn, khảo sát, workshop hoặc xây dựng prototype để giúp khách hàng mô tả yêu cầu rõ ràng hơn. Nếu khách hàng vẫn chưa xác định rõ nhu cầu, có thể đề xuất phát triển theo mô hình Agile để dễ dàng điều chỉnh sau này.

8. Một dự án gặp rủi ro cao trong pha khởi đầu do thiếu tài liệu yêu cầu rõ ràng. Đội phát triển nên làm gì?

Xây dựng tài liệu yêu cầu tối thiểu (Minimum Viable Documentation) dựa trên các thông tin có sẵn. Kết hợp với khách hàng và các bên liên quan để làm rõ các yêu cầu quan trọng nhất, tránh rủi ro trong quá trình phát triển.

9. Dự án phần mềm lớn có nhiều nhóm phát triển ở các địa điểm khác nhau. Làm thế nào để đảm bảo các nhóm phối hợp hiệu quả?

Sử dụng công cụ quản lý dự án như Jira, Azure DevOps hoặc Trello, tổ chức các cuộc họp định kỳ giữa các nhóm để đảm bảo sự phối hợp nhịp nhàng. Đồng thời, thiết lập quy trình làm việc chung để tránh hiểu nhầm trong quá trình phát triển.

10. Một công ty phát triển phần mềm gặp khó khăn trong việc quản lý quy trình do không có chuẩn hóa. Hãy đề xuất giải pháp.

Áp dụng một framework quản lý quy trình như CMMI, ISO hoặc Agile, xây dựng bộ tài liệu chuẩn cho các quy trình quan trọng. Đào tạo nhân sự để đảm bảo mọi người tuân theo quy trình, giúp công ty vận hành hiệu quả hơn.

05. CÂU HỎI TRẮC NGHIỆM

1. 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?

A. Pha thiết kế

B. Pha lấy yêu cầu

C. Pha phân tích

D. Pha cài đặt

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

năng?

A. Mô hình thác nước

B. Mô hình lặp và tăng trưởng

C. Mô hình bản mẫu nhanh

D. Mô hình tiến trình linh hoạt

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

A. Viết mã nguồn

B. Sửa lỗi và cập nhật tính năng mới

C. Gỡ bỏ phần mềm

D. Phân tích yêu cầu

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

A. Dự án nhỏ, đơn giản

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

C. Dự án yêu cầu linh hoạt cao

D. Dự án mã nguồn mở

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

A. Một pha kiểm thử

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

C. Một phiên bản phần mềm nhỏ

D. Một lần phân tích rủi ro

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

A. Khó phát triển nhanh

B. Tốn nhiều chi phí bảo trì

C. Khó kiểm soát chất lượng

D. Không phù hợp với dự án nhỏ

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

A. Mô hình lặp và tăng trưởng

B. Mô hình bản mẫu nhanh

C. Mô hình xoắn ốc

D. Mô hình tiến trình linh hoạt

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

A. Pha bảo trì

B. Pha cài đặt

C. Pha giải thể

D. Pha thiết kế

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

- A. Mô hình thác nước lặp lại nhiều lần
- B. Mô hình lặp và tăng trưởng phát triển theo từng đợt nhỏ**
- C. Mô hình lặp và tăng trưởng không có giai đoạn kiểm thử
- D. Mô hình thác nước không có giai đoạn bảo trì

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

- A. Mô hình thác nước
- B. Mô hình xoắn ốc
- C. Mô hình xây và sửa
- D. Mô hình tiến trình linh hoạt**

06. 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 (Requirement Gathering) là giai đoạn đầu tiên trong vòng đời phát triển phần mềm. Nó tập trung vào việc thu thập, phân tích và xác định các yêu cầu của người dùng cũng như doanh nghiệp. Vai trò của nó là giúp đảm bảo rằng phần mềm đáp ứng đúng nhu cầu, tránh việc phát triển sai hướng và giảm thiểu rủi ro trong các giai đoạn sau.

2. Mô hình thác nước hoạt động như thế nào?

Mô hình thác nước (Waterfall Model) hoạt động theo quy trình tuần tự, mỗi pha phải hoàn thành trước khi chuyển sang pha tiếp theo. Các pha chính bao gồm:

1. Phân tích yêu cầu
2. Thiết kế hệ thống
3. Triển khai (Coding)
4. Kiểm thử (Testing)
5. Triển khai và bảo trì

Mô hình này phù hợp với các dự án có yêu cầu rõ ràng, ít thay đổi.

3. Mô hình lặp và tăng trưởng khác gì so với mô hình thác nước?

Mô hình lặp và tăng trưởng (Iterative and Incremental Model) phát triển phần mềm theo từng vòng lặp nhỏ, mỗi vòng lặp có thể bổ sung và cải tiến tính năng của sản phẩm.

Khác với mô hình thác nước, mô hình này linh hoạt hơn, cho phép phản hồi và điều chỉnh trong quá trình phát triển, giúp giảm rủi ro nếu yêu cầu thay đổi.

4.Mục tiêu của pha bảo trì là gì?

Mục tiêu của pha bảo trì là đảm bảo phần mềm hoạt động ổn định, sửa lỗi, cập nhật tính năng mới, cải thiện hiệu suất và bảo mật.

5.Mô hình xây và sửa có nhược điểm gì?

Nhược điểm của mô hình xây và sửa (Build-and-Fix Model) gồm:

- Khó kiểm soát chất lượng do thiếu kế hoạch chi tiết.
- Tốn nhiều chi phí bảo trì vì sản phẩm có thể không tối ưu ngay từ đầu.
- Dễ dẫn đến việc phát triển không có định hướng rõ ràng.

6.Mô hình bản mẫu nhanh là gì?

Mô hình bản mẫu nhanh (Rapid Prototyping Model) tập trung vào việc tạo ra các nguyên mẫu (prototype) nhanh chóng để thu thập phản hồi từ khách hàng. Sau khi nhận phản hồi, nguyên mẫu được cải tiến dần để tạo ra sản phẩm hoàn chỉnh.

7.Pha giải thể là gì?

Pha giải thể (Retirement Phase) là giai đoạn kết thúc vòng đời phần mềm khi hệ thống không còn được sử dụng. Nó bao gồm việc ngừng hỗ trợ, chuyển đổi dữ liệu, và loại bỏ phần mềm một cách an toàn.

8.Mô hình xoắn ốc là gì?

Mô hình xoắn ốc (Spiral Model) kết hợp các yếu tố của mô hình thác nước và mô hình lặp. Nó tập trung vào quản lý rủi ro bằng cách chia dự án thành nhiều chu kỳ xoắn, mỗi chu kỳ bao gồm các bước: lập kế hoạch, phân tích rủi ro, phát triển, và đánh giá.

9.Tại sao mô hình tiến trình linh hoạt được đánh giá cao?

Mô hình tiến trình linh hoạt (Agile) được đánh giá cao vì:

- Linh hoạt với sự thay đổi của yêu cầu khách hàng.
- Cho phép phát triển phần mềm theo từng phần nhỏ (incremental).
- Tăng cường giao tiếp giữa nhóm phát triển và khách hàng.
- Cải thiện tốc độ phát triển và chất lượng sản phẩm.

10.Điểm khác biệt chính giữa mô hình mã nguồn mở và các mô hình khác là gì?

Mô hình mã nguồn mở (Open Source Model) khác với các mô hình khác ở chỗ:

- Mã nguồn được công khai và có thể được đóng góp bởi cộng đồng.
- Phát triển dựa trên sự hợp tác giữa nhiều lập trình viên trên toàn cầu.
- Không bị giới hạn bởi một công ty hay tổ chức duy nhất.
- Thường có sự phát triển liên tục với nhiều bản cập nhật từ cộng đồng.

7. CÂU HỎI THẢO LUẬN NHÓM

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

Tiêu chí	Mô hình thác nước (Waterfall)	Mô hình xoắn ốc (Spiral)
Ưu điểm	<ul style="list-style-type: none">- Dễ hiểu, dễ quản lý do có quy trình rõ ràng.- Phù hợp với các dự án có yêu cầu cố định.- Kiểm tra chất lượng ở từng giai đoạn.	<ul style="list-style-type: none">- Giảm thiểu rủi ro nhờ phân tích và kiểm tra liên tục.- Dễ thích nghi với yêu cầu thay đổi.- Cung cấp sản phẩm sớm trong từng chu kỳ phát triển.
Nhược điểm	<ul style="list-style-type: none">- Không linh hoạt khi yêu cầu thay đổi.- Tốn thời gian để quay lại sửa lỗi ở các giai đoạn trước.	<ul style="list-style-type: none">- Đòi hỏi chi phí cao hơn do quá trình lặp lại nhiều lần.- Quản lý phức tạp hơn so với mô hình thác nước.

2. Thảo luận về tình huống thực tế có thể áp dụng mô hình lặp và tăng trưởng.

Mô hình lặp và tăng trưởng phù hợp với các dự án có yêu cầu thay đổi theo thời gian. Ví dụ:

- Ứng dụng thương mại điện tử: Ban đầu chỉ phát triển các chức năng cơ bản như đăng ký, đăng nhập, đặt hàng. Sau đó, bổ sung giỏ hàng, thanh toán trực tuyến, và các tính năng nâng cao khác.
- Phần mềm quản lý doanh nghiệp: Doanh nghiệp có thể cần các tính năng cơ bản ban đầu (quản lý nhân sự, kế toán) và mở rộng dần theo nhu cầu (CRM, phân tích dữ liệu).

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

- Thiếu kế hoạch chi tiết: Dễ dẫn đến lỗi phát sinh và tốn kém trong việc sửa lỗi.
- Khó bảo trì: Không có tài liệu đầy đủ, việc mở rộng hoặc sửa chữa sau này gặp nhiều khó khăn.
- Chi phí cao: Vì sửa đổi liên tục, tổng chi phí phát triển có thể vượt quá dự kiến.
- Không đảm bảo chất lượng: Dự án lớn đòi hỏi kiểm thử chặt chẽ, trong khi mô hình này tập trung vào phát triển nhanh mà không có giai đoạn kiểm thử rõ ràng.

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)
Ưu điểm	<ul style="list-style-type: none"> - Giúp khách hàng dễ hình dung sản phẩm sớm. - Cho phép chỉnh sửa nhanh theo phản hồi. 	<ul style="list-style-type: none"> - Thích ứng nhanh với thay đổi của khách hàng. - Cung cấp sản phẩm có thể sử dụng sớm và cải tiến liên tục.
Nhược điểm	<ul style="list-style-type: none"> - Mất thời gian xây dựng nguyên mẫu có thể bị bỏ đi. - Không phù hợp với dự án lớn, phức tạp. 	<ul style="list-style-type: none"> - Cần sự phối hợp chặt chẽ giữa nhóm phát triển và khách hàng. - Đòi hỏi đội ngũ có kỹ năng cao để quản lý hiệu quả.

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

Quản lý rủi ro là cốt lõi của mô hình xoắn ốc, giúp:

- Phát hiện sớm vấn đề: Ở mỗi vòng lặp, nhóm phát triển đánh giá rủi ro và có biện pháp xử lý.
- Giảm chi phí sửa lỗi: Việc kiểm soát rủi ro ngay từ đầu giúp giảm chi phí khắc phục sau này.
- Tối ưu tài nguyên: Tập trung phát triển các tính năng quan trọng trước, tránh lãng phí nguồn lực vào các phần không cần thiết.

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, ít thay đổi (ví dụ: hệ thống nhúng, phần mềm quản lý nội bộ).
- Khi đội ngũ không có kinh nghiệm với Agile và cần một quy trình chặt chẽ.
- Khi có thời gian và ngân sách hạn chế, cần kiểm soát chặt chẽ từng giai đoạn.

7. Thảo luận về những khó khăn khi áp dụng mô hình mã nguồn mở.

- Khó kiểm soát chất lượng: Nhiều lập trình viên tham gia, nhưng không phải ai cũng có kỹ năng cao.
- Vấn đề bảo mật: Mã nguồn mở có thể bị khai thác nếu không được kiểm tra kỹ.
- Thiếu sự hỗ trợ chính thức: Không có nhà cung cấp chính thức để hỗ trợ khi gặp sự cố.
- Khó tạo lợi nhuận: Do mã nguồn mở miễn phí, doanh nghiệp phải tìm mô hình kinh doanh phù hợp.
-

8. Phân tích cách mô hình tiến trình linh hoạt giúp cải thiện chất lượng phần mềm.

- Phản hồi nhanh từ khách hàng giúp phát hiện lỗi sớm.
- Phát triển theo từng bước nhỏ giúp kiểm thử liên tục và tránh lỗi tích tụ.
- Cộng tác nhóm chặt chẽ giúp cải thiện hiệu suất làm việc.
- Tích hợp liên tục giúp đảm bảo phần mềm luôn hoạt động tốt khi có tính năng mới.

9. Thảo luận về vai trò của pha bảo trì trong vòng đời phát triển phần mềm.

- Sửa lỗi: Khắc phục lỗi phần mềm sau khi triển khai.
- Cập nhật tính năng: Bổ sung chức năng mới theo nhu cầu người dùng.
- Cải thiện hiệu suất: Tối ưu tốc độ xử lý và khả năng mở rộng của hệ thống.
- Đảm bảo bảo mật: Vá lỗi bảo mật để tránh các cuộc tấn công.
-

10. Đề xuất mô hình vòng đời phù hợp cho dự án phát triển phần mềm ngân hàng và giải thích lý do.

Mô hình phù hợp: Mô hình xoắn ốc hoặc mô hình tiến trình linh hoạt.

- Tại sao không chọn mô hình thác nước?
 - Ngành ngân hàng có yêu cầu thay đổi liên tục về chính sách và quy định.
 - Cần kiểm tra bảo mật chặt chẽ ở nhiều giai đoạn.
- Tại sao chọn mô hình xoắn ốc?
 - Ngân hàng có yêu cầu cao về bảo mật và kiểm soát rủi ro.
 - Dự án lớn, cần nhiều vòng lặp để kiểm tra chất lượng trước khi triển khai.
- Tại sao chọn mô hình tiến trình linh hoạt?
 - Các sản phẩm ngân hàng hiện đại (Mobile Banking, Internet Banking) cần phát triển nhanh, liên tục cập nhật theo phản hồi người dùng.
 - Agile giúp triển khai các tính năng quan trọng trước, điều chỉnh linh hoạt theo nhu cầu thực tế.

7. CÂU HỎI TÌNH HUỐNG

1. Một công ty phát triển phần mềm theo mô hình thác nước gặp vấn đề khi khách hàng yêu cầu thay đổi sau khi hoàn thành pha thiết kế. Đội phát triển nên xử lý như thế nào?

Trong mô hình thác nước, thay đổi sau pha thiết kế rất khó khăn. Đội phát triển có thể đánh giá mức độ ảnh hưởng, lập kế hoạch bổ sung tài nguyên và điều chỉnh hợp đồng để đáp ứng yêu cầu khách hàng.

2. Dự án phát triển phần mềm theo mô hình lặp và tăng trưởng liên tục bị trễ tiến độ do thiếu nhân lực. Là quản lý dự án, bạn sẽ làm gì?

Cần phân tích nguyên nhân thiếu nhân lực, điều chỉnh phạm vi công việc hoặc tăng cường nhân sự. Có thể áp dụng ưu tiên công việc quan trọng trước hoặc tìm giải pháp tự động hóa để giảm tải.

3. Trong quá trình phát triển phần mềm theo mô hình tiến trình linh hoạt, khách hàng không đưa ra phản hồi kịp thời. Đội phát triển nên xử lý ra sao?

Đội phát triển nên chủ động liên lạc với khách hàng, đặt ra thời hạn phản hồi cụ thể. Nếu cần, có thể đưa ra các giả định hợp lý để tiếp tục phát triển mà không làm chậm tiến độ.

4. Khách hàng yêu cầu bổ sung một số tính năng mới khi phần mềm đã bước vào pha cài đặt. Nên áp dụng mô hình nào để xử lý tốt nhất yêu cầu này?

Mô hình Agile hoặc mô hình xoắn ốc phù hợp vì cho phép thay đổi linh hoạt ngay cả khi phần mềm đã bước vào pha cài đặt mà không ảnh hưởng lớn đến tiến độ.

5. Một công ty nhỏ muốn áp dụng mô hình bản mẫu nhanh nhưng gặp khó khăn do thiếu nguồn lực. Hãy đề xuất giải pháp.

Công ty có thể tối ưu hóa tài nguyên bằng cách sử dụng các công cụ phát triển nhanh, tận dụng nền tảng mã nguồn mở hoặc thuê ngoài một số công đoạn để giảm gánh nặng nhân sự.

6. Trong dự án phần mềm thương mại điện tử, khách hàng liên tục yêu cầu thay đổi giao diện. Mô hình nào sẽ phù hợp nhất?

Mô hình Agile hoặc bản mẫu nhanh sẽ phù hợp vì có khả năng thích ứng nhanh với các yêu cầu thay đổi của khách hàng, đặc biệt là về giao diện người dùng.

7. Dự án phát triển phần mềm lớn với nhiều nhóm phát triển ở các quốc gia khác nhau nên áp dụng mô hình nào?

Mô hình phát triển phân tán hoặc Agile với Scrum có thể phù hợp, giúp điều phối công việc giữa các nhóm ở các quốc gia khác nhau thông qua các công cụ quản lý dự án.

8. Phân tích rủi ro là hoạt động chính trong mô hình xoắn ốc. Đề xuất cách giảm thiểu rủi ro khi áp dụng mô hình này.

Để giảm rủi ro trong mô hình xoắn ốc, cần thực hiện đánh giá rủi ro định kỳ, xây dựng kế hoạch dự phòng và thử nghiệm các giải pháp trước khi triển khai chính thức.

9. Một dự án phát triển phần mềm ngân hàng cần yêu cầu bảo mật cao. Nên áp dụng mô hình nào để đảm bảo yêu cầu này?

Mô hình thác nước hoặc mô hình xoắn ốc là lựa chọn tốt vì có các bước kiểm tra nghiêm ngặt, giúp đảm bảo tính bảo mật cao cho hệ thống ngân hàng.

10. Khi nào nên kết thúc vòng đời phần mềm và thực hiện pha giải thể?

Vòng đời phần mềm kết thúc khi không còn đáp ứng nhu cầu kinh doanh, chi phí bảo trì cao hơn lợi ích hoặc có công nghệ mới thay thế. Lúc này, cần lập kế hoạch giải thể an toàn