

Bài tập chương 4

CÂU HỎI TRẮC NGHIỆM

- Mục tiêu chính của kiểm thử phần mềm là gì?
B. Đảm bảo phần mềm đáp ứng đúng yêu cầu
(Kiểm thử giúp xác nhận phần mềm hoạt động đúng như yêu cầu đặt ra, không chỉ tìm lỗi.)
- Nhóm nào chịu trách nhiệm đảm bảo chất lượng phần mềm trong dự án?
C. Nhóm SQA (Software Quality Assurance)
(Nhóm SQA chịu trách nhiệm giám sát và đảm bảo quy trình phát triển tuân thủ tiêu chuẩn chất lượng.)
- Kiểm thử các sản phẩm phi thực thi bao gồm hoạt động nào dưới đây?
B. Walkthrough và review tài liệu
(Các tài liệu như đặc tả yêu cầu, thiết kế cũng cần được kiểm tra để đảm bảo tính chính xác.)
- Lập kế hoạch trong tiến trình phát triển phần mềm nhằm mục đích gì?
A. Xác định phạm vi công việc và ước tính thời gian hoàn thành
(Lập kế hoạch giúp xác định phạm vi, lịch trình và tài nguyên cần thiết cho dự án.)
- Loại kiểm thử nào sau đây tập trung vào việc phát hiện các lỗi chức năng của phần mềm?
B. Kiểm thử chức năng
(Kiểm thử chức năng tập trung vào việc phần mềm có thực hiện đúng chức năng theo yêu cầu hay không.)
- Quản lý phiên bản tài liệu có mục tiêu gì?
A. Đảm bảo tài liệu luôn được cập nhật và có thể truy xuất phiên bản cũ khi cần
(Quản lý phiên bản giúp theo dõi các thay đổi và khôi phục tài liệu khi cần.)
- Nhóm SQA có vai trò gì trong kiểm thử phần mềm?
B. Đánh giá và đảm bảo quy trình phát triển phần mềm tuân thủ tiêu chuẩn chất lượng
(Nhóm SQA giám sát quy trình, không trực tiếp viết code hay triển khai phần mềm.)
- Loại kiểm thử nào thường được thực hiện cuối cùng trước khi phần mềm được bàn giao cho khách hàng?
D. Kiểm thử chấp nhận
(Kiểm thử chấp nhận (UAT) là bước cuối cùng để xác nhận phần mềm đáp ứng yêu cầu trước khi bàn giao.)
- Đâu là một trong những công cụ phổ biến dùng để quản lý phiên bản tài liệu?
B. Git
(Git không chỉ dùng để quản lý mã nguồn mà còn có thể dùng để quản lý phiên bản tài liệu.)
- Hoạt động lập tài liệu trong mỗi pha phát triển phần mềm nhằm mục đích gì?
B. Hỗ trợ quá trình bảo trì và nâng cấp phần mềm sau khi triển khai
(Tài liệu giúp hiểu rõ hệ thống, hỗ trợ bảo trì và nâng cấp dễ dàng hơn.)

CÂU HỎI NGẮN

- Nhóm SQA là gì và vai trò của nhóm này trong phát triển phần mềm?

Nhóm SQA (Software Quality Assurance) là nhóm chịu trách nhiệm giám sát, đánh giá và đảm bảo chất lượng phần mềm trong suốt vòng đời phát triển.

Vai trò chính:

- Đảm bảo quy trình phát triển phần mềm tuân thủ tiêu chuẩn chất lượng (ISO, CMMI, Agile, v.v.).
- Xây dựng và thực thi kế hoạch kiểm thử, review tài liệu, kiểm tra mã nguồn.
- Giám sát việc tuân thủ các quy trình phát triển phần mềm.
- Đánh giá hiệu suất phần mềm và đưa ra khuyến nghị cải tiến.

2. Kiểm thử đơn vị là gì?

Kiểm thử đơn vị (Unit Testing) là quá trình kiểm thử từng thành phần nhỏ nhất của phần mềm (hàm, module, class) để đảm bảo chúng hoạt động đúng.

Đặc điểm:

- Thường được thực hiện bởi lập trình viên.
- Sử dụng framework như Jest (JavaScript), JUnit (Java), PyTest (Python), v.v.
- Giúp phát hiện lỗi ngay từ giai đoạn đầu, tiết kiệm chi phí sửa lỗi sau này.

3. Mục tiêu chính của kiểm thử chấp nhận là gì?

Mục tiêu:

- Xác nhận rằng phần mềm đáp ứng đúng yêu cầu nghiệp vụ và mong đợi của khách hàng.
- Đảm bảo phần mềm sẵn sàng để triển khai.
- Phát hiện các lỗi hoặc vấn đề từ góc nhìn của người dùng.

Các loại kiểm thử chấp nhận:

- UAT (User Acceptance Testing): Kiểm thử do khách hàng thực hiện.
- Alpha Testing: Kiểm thử nội bộ trước khi phát hành.
- Beta Testing: Kiểm thử với người dùng thật trước khi chính thức triển khai.

4. Các hoạt động chính trong kiểm thử sản phẩm phi thực thi là gì?

Sản phẩm phi thực thi bao gồm tài liệu yêu cầu, thiết kế, mô hình dữ liệu, v.v.

Các hoạt động chính:

- Review tài liệu (Requirement Review, Design Review).

- Walkthrough (trình bày tài liệu để nhận phản hồi).
- Inspection (kiểm tra chính thức tài liệu để phát hiện lỗi).
- Phân tích tĩnh (Static Analysis) giúp tìm lỗi trong mã nguồn mà không cần chạy chương trình.

5. Tại sao việc lập tài liệu cho mỗi pha phát triển phần mềm lại quan trọng?

Lợi ích của việc lập tài liệu:

- Hỗ trợ bảo trì phần mềm (giúp lập trình viên hiểu hệ thống để sửa lỗi, nâng cấp dễ dàng hơn).
- Lưu trữ thông tin yêu cầu, thiết kế, kiểm thử (đảm bảo nhất quán giữa các đội nhóm).
- Hỗ trợ đào tạo nhân viên mới (có tài liệu rõ ràng giúp nhân viên mới dễ dàng tiếp cận dự án).
- Giảm rủi ro khi thay đổi đội ngũ phát triển.

6. Quản lý phiên bản tài liệu là gì?

Quản lý phiên bản tài liệu là quá trình theo dõi, kiểm soát các thay đổi của tài liệu trong suốt vòng đời phần mềm.

Mục tiêu:

- Giữ lịch sử thay đổi tài liệu.
 - Cho phép khôi phục phiên bản cũ khi cần.
 - Đảm bảo tất cả thành viên sử dụng tài liệu mới nhất.
- Công cụ phổ biến: Git, SVN, Confluence, Google Docs.

7. Các loại kiểm thử chính trong kiểm thử sản phẩm thực thi là gì?

Kiểm thử sản phẩm thực thi liên quan đến việc kiểm tra phần mềm đang chạy.

Các loại chính:

- Kiểm thử đơn vị (Unit Testing) – kiểm tra từng module riêng lẻ.
- Kiểm thử tích hợp (Integration Testing) – kiểm tra sự kết hợp giữa các module.
- Kiểm thử hệ thống (System Testing) – kiểm tra toàn bộ hệ thống phần mềm.
- Kiểm thử chấp nhận (Acceptance Testing) – xác nhận phần mềm đáp ứng yêu cầu nghiệp vụ.
- Kiểm thử hiệu suất (Performance Testing) – đo tốc độ, độ ổn định phần mềm.
- Kiểm thử bảo mật (Security Testing) – kiểm tra khả năng bảo vệ dữ liệu.

8. Kiểm thử tích hợp là gì?

Kiểm thử tích hợp (Integration Testing) là quá trình kiểm thử sự tương tác giữa các module trong phần mềm.

Mục tiêu:

- Đảm bảo các thành phần hoạt động cùng nhau mà không gây lỗi.
 - Kiểm tra dữ liệu luân chuyển giữa các module đúng cách.
- Các phương pháp kiểm thử tích hợp:
- Top-down: Kiểm thử từ module cao nhất xuống module con.
 - Bottom-up: Kiểm thử từ module con trước rồi đến module cao hơn.
 - Big Bang: Kết hợp tất cả các module rồi kiểm thử cùng lúc.
-

9. Hoạt động lập kế hoạch cho các pha phát triển phần mềm bao gồm những gì?

Các hoạt động chính trong lập kế hoạch:

- Xác định phạm vi dự án (Scope).
- Phân tích yêu cầu và rủi ro.
- Lập lịch trình phát triển phần mềm (timeline, milestones).
- Ước tính nguồn lực (nhân sự, công cụ, ngân sách).
- Xác định chiến lược kiểm thử và triển khai.

10. Làm tài liệu kiểm thử bao gồm những gì?

Tài liệu kiểm thử giúp theo dõi và quản lý quá trình kiểm thử phần mềm.

Các thành phần chính:

1. Test Plan (Kế hoạch kiểm thử) – Xác định phạm vi, phương pháp kiểm thử.
2. Test Cases (Kịch bản kiểm thử) – Mô tả từng trường hợp kiểm thử chi tiết.
3. Test Scripts – Mã kiểm thử tự động (nếu có).
4. Test Report (Báo cáo kiểm thử) – Kết quả kiểm thử, lỗi tìm thấy, trạng thái phần mềm.

CÂU HỎI THẢO LUẬN NHÓM

1. Vai trò của nhóm SQA trong việc đảm bảo chất lượng phần mềm là gì?

Nhóm SQA (Software Quality Assurance) có nhiệm vụ giám sát, đánh giá và đảm bảo phần mềm đáp ứng các tiêu chuẩn chất lượng trong suốt quá trình phát triển.

Vai trò chính:

- Xây dựng quy trình đảm bảo chất lượng dựa trên các tiêu chuẩn như ISO 9001, CMMI, Agile.
- Giám sát và đánh giá quy trình phát triển phần mềm để phát hiện sai sót.
- Định nghĩa chiến lược kiểm thử và hỗ trợ nhóm kiểm thử trong việc đảm bảo chất lượng sản phẩm.
- Xác định và theo dõi các lỗi, rủi ro có thể ảnh hưởng đến chất lượng phần mềm.
- Thực hiện kiểm tra tài liệu, mã nguồn, quy trình để đảm bảo tính nhất quán và đúng chuẩn.

2. Sự khác nhau giữa kiểm thử đơn vị và kiểm thử tích hợp

Tiêu chí	Kiểm thử đơn vị (Unit Testing)	Kiểm thử tích hợp (Integration Testing)
Mục đích	Kiểm tra từng module riêng lẻ	Kiểm tra sự kết hợp giữa các module
Ai thực hiện?	Lập trình viên	Nhóm kiểm thử hoặc nhóm phát triển
Giai đoạn thực hiện	Giai đoạn đầu của kiểm thử	Sau khi các module đã được kiểm thử đơn vị
Công cụ hỗ trợ	Jest, JUnit, PyTest, NUnit	Selenium, Postman, TestNG, SoapUI
Ví dụ	Kiểm tra hàm calculateTotalPrice() có trả về kết quả đúng không	Kiểm tra API giao tiếp giữa backend và frontend có hoạt động đúng không

3. Tại sao việc lập tài liệu kiểm thử lại quan trọng trong mỗi dự án phần mềm?

Lợi ích của tài liệu kiểm thử:

- Đảm bảo tính minh bạch về chiến lược kiểm thử, phạm vi kiểm thử.
- Tạo nền tảng cho bảo trì và nâng cấp khi có thay đổi trong tương lai.
- Hỗ trợ giao tiếp giữa các nhóm (QA, Dev, Khách hàng) về yêu cầu chất lượng.
- Giúp theo dõi lỗi và hiệu suất phần mềm qua từng giai đoạn.
- Chuẩn hóa quy trình kiểm thử, dễ dàng đào tạo nhân viên mới.

4. Các thách thức khi lập kế hoạch cho các pha phát triển phần mềm

Các thách thức chính:

1. Xác định yêu cầu không đầy đủ – Khách hàng thay đổi yêu cầu liên tục.
2. Ước tính thời gian và nguồn lực khó khăn – Dự án có thể mất nhiều thời gian hơn dự kiến.
3. Rủi ro kỹ thuật – Công nghệ mới hoặc lỗi bảo mật có thể ảnh hưởng tiến độ.
4. Quản lý nhóm phát triển – Sự phối hợp giữa các team đôi khi không hiệu quả.
5. Tích hợp và kiểm thử muộn – Nếu không lập kế hoạch tốt, có thể gặp lỗi ở giai đoạn cuối.

5. Quản lý phiên bản tài liệu có ảnh hưởng như thế nào đến quá trình bảo trì phần mềm?

Ảnh hưởng của quản lý phiên bản tài liệu:

- Dễ dàng theo dõi thay đổi trong tài liệu (yêu cầu, thiết kế, kiểm thử).
- Hỗ trợ quay lại phiên bản cũ nếu phát hiện lỗi hoặc cần phục hồi dữ liệu.
- Tăng tính minh bạch và cộng tác, đặc biệt trong nhóm lớn hoặc phân tán.
- Giúp bảo trì phần mềm hiệu quả bằng cách lưu trữ thông tin chi tiết về kiến trúc và thay đổi phần mềm.

Công cụ phổ biến: Git, SVN, Confluence, Google Docs, SharePoint.

6. So sánh kiểm thử sản phẩm phi thực thi và kiểm thử sản phẩm thực thi

Tiêu chí	Kiểm thử sản phẩm phi thực thi	Kiểm thử sản phẩm thực thi
Đối tượng	Tài liệu, thiết kế, yêu cầu	Mã nguồn, ứng dụng thực tế
Mục đích	Phát hiện lỗi trong tài liệu, thiết kế trước khi phát triển	Kiểm tra tính đúng đắn của phần mềm khi chạy
Ví dụ	Walkthrough, Review tài liệu, Static Analysis	Kiểm thử đơn vị, kiểm thử hệ thống, kiểm thử hiệu suất

7. Cách cải thiện quy trình lập kế hoạch để giảm thiểu rủi ro trong dự án phần mềm

Cách cải thiện:

- Xác định rủi ro ngay từ đầu và có kế hoạch dự phòng.
- Lập lịch trình linh hoạt, sử dụng phương pháp Agile hoặc Scrum để giảm thiểu thay đổi.
- Thường xuyên họp nhóm, cập nhật tiến độ và điều chỉnh kế hoạch.
- Sử dụng công cụ quản lý dự án như Jira, Trello, Asana để theo dõi công việc.

8. Công cụ hỗ trợ việc lập tài liệu kiểm thử và quản lý phiên bản

Công cụ lập tài liệu kiểm thử:

- TestRail, Zephyr – Quản lý test case, test plan.
- Google Docs, Confluence – Viết và lưu trữ tài liệu.

Công cụ quản lý phiên bản:

- Git, GitHub, GitLab – Theo dõi lịch sử thay đổi mã nguồn và tài liệu.
- SVN (Subversion) – Quản lý phiên bản tài liệu truyền thống.

9. Tại sao kiểm thử chấp nhận lại là một giai đoạn quan trọng trong phát triển phần mềm?

Tầm quan trọng của kiểm thử chấp nhận (UAT - User Acceptance Testing):

- Xác minh rằng phần mềm đáp ứng yêu cầu thực tế của khách hàng.
- Phát hiện lỗi từ góc nhìn người dùng trước khi triển khai.
- Giảm thiểu rủi ro thất bại sau khi phát hành sản phẩm.
- Tạo sự tin tưởng từ khách hàng, đảm bảo tính sẵn sàng của hệ thống.

10. Các phương pháp hiệu quả để quản lý chất lượng phần mềm trong các dự án lớn

Các phương pháp chính:

1. Áp dụng mô hình phát triển phù hợp (Agile, DevOps, Spiral).
2. Xây dựng quy trình kiểm thử tự động để giảm thời gian và công sức.
3. Thực hiện kiểm thử sớm và liên tục (Continuous Testing).
4. Quản lý lỗi hiệu quả bằng các công cụ như JIRA, Bugzilla.
5. Đánh giá và kiểm tra mã nguồn định kỳ để tránh lỗi tiềm ẩn.
6. Đào tạo nhóm phát triển và QA về các tiêu chuẩn chất lượng phần mềm.

CÂU HỎI TÌNH HUỐNG

1. Khách hàng yêu cầu kiểm tra lại toàn bộ tài liệu yêu cầu và thiết kế trước khi bàn giao phần mềm

Cách xử lý:

- Rà soát lại tài liệu: Đội phát triển và nhóm SQA kiểm tra các tài liệu yêu cầu và thiết kế để đảm bảo không có sai sót.
- Tổ chức buổi họp với khách hàng để xác nhận lại nội dung cần kiểm tra và giải thích nếu có sự khác biệt giữa phần mềm và tài liệu gốc.
- So sánh tài liệu với phần mềm hiện tại, cập nhật nếu cần để phản ánh đúng thực tế.
- Cung cấp báo cáo giải thích nếu có sự thay đổi hợp lý trong quá trình phát triển mà chưa được cập nhật vào tài liệu.

2. Phát hiện lỗi nghiêm trọng trong kiểm thử hệ thống khi thời hạn bàn giao sắp đến

Cách xử lý:

- Đánh giá mức độ ảnh hưởng: Xác định lỗi có ảnh hưởng đến chức năng chính của phần mềm hay không.
- Thảo luận với khách hàng: Nếu lỗi nghiêm trọng, cần báo ngay để thỏa thuận về phương án sửa lỗi và khả năng gia hạn thời gian bàn giao.
- Tạm thời tìm giải pháp thay thế (workaround) để khách hàng có thể sử dụng phần mềm mà không bị ảnh hưởng nặng nề.
- Lập kế hoạch sửa lỗi nhanh chóng và kiểm thử lại toàn bộ hệ thống trước khi bàn giao.

3. Khó khăn trong quản lý phiên bản tài liệu do tài liệu liên tục thay đổi

Giải pháp đề xuất:

- Sử dụng hệ thống quản lý phiên bản như Git, Confluence, SharePoint, Google Docs để lưu trữ và theo dõi thay đổi.
- Xác định quy trình quản lý tài liệu: Ai có quyền chỉnh sửa? Khi nào cần cập nhật? Ai phê duyệt thay đổi?
- Đánh số phiên bản tài liệu rõ ràng (VD: v1.0, v1.1, v2.0...) để dễ theo dõi.
- Dùng công cụ theo dõi thay đổi (Track Changes trong Word, Google Docs Suggesting mode) để kiểm soát chỉnh sửa.

4. Khách hàng yêu cầu thay đổi lớn trong pha cài đặt

Cách xử lý:

- Đánh giá tác động: Xác định thay đổi này ảnh hưởng đến phạm vi, thời gian và chi phí dự án như thế nào.

- Thương lượng với khách hàng: Nếu thay đổi có tác động lớn, cần đề xuất phương án như kéo dài thời gian hoặc bổ sung ngân sách.
- Chia nhỏ thay đổi thành các giai đoạn để giảm rủi ro.
- Cập nhật lại tài liệu yêu cầu để đảm bảo tính nhất quán.

5. Nhóm kiểm thử phát hiện lỗi nhưng nhóm phát triển cho rằng đó là tính năng

Cách xử lý:

- Kiểm tra lại tài liệu yêu cầu: Nếu yêu cầu ban đầu không rõ ràng, cần xác minh lại với khách hàng.
- Tổ chức cuộc họp giữa hai nhóm để làm rõ vấn đề. Nếu có tranh cãi, cần xem xét tài liệu và quyết định dựa trên yêu cầu ban đầu.
- Nếu khách hàng coi đó là lỗi, thì phải sửa. Nếu khách hàng chấp nhận tính năng đó, nhóm kiểm thử có thể cập nhật test case.
- Cập nhật tài liệu để tránh hiểu lầm trong tương lai.

6. Khách hàng yêu cầu bổ sung tính năng mới sau khi đã hoàn thành kiểm thử tích hợp

Cách xử lý:

- Đánh giá tác động của tính năng mới: Nếu ảnh hưởng đến kiến trúc hoặc các tính năng khác, cần xem xét kỹ.
- Thương lượng với khách hàng: Có thể thêm tính năng này vào phiên bản tiếp theo thay vì triển khai ngay lập tức.
- Xây dựng kế hoạch phát triển và kiểm thử lại nếu quyết định thêm tính năng vào phiên bản hiện tại.
- Cập nhật tài liệu và thông báo cho các bên liên quan.

7. Công ty phần mềm nhỏ muốn xây dựng nhóm SQA nhưng gặp khó khăn về ngân sách

Giải pháp:

- Thuê ngoài dịch vụ kiểm thử (outsourcing) để tiết kiệm chi phí.
- Sử dụng công cụ kiểm thử tự động miễn phí như Selenium, JMeter, Postman để giảm tải công việc thủ công.
- Đào tạo nội bộ: Chia vai trò kiểm thử cho các lập trình viên thay vì thuê đội SQA riêng biệt.
- Áp dụng quy trình kiểm thử cơ bản như Code Review, Unit Testing để nâng cao chất lượng mà không cần quá nhiều nhân sự.

7. Không thống nhất được nội dung tài liệu kiểm thử trong nhóm phát triển

Cách giải quyết:

- Tổ chức buổi họp thảo luận để làm rõ các tiêu chí cần có trong tài liệu.
- Tham khảo tiêu chuẩn kiểm thử quốc tế như ISTQB, IEEE 829.
- Đưa ra mẫu tài liệu kiểm thử chung để tất cả thành viên thống nhất và sử dụng.
- Ghi nhận ý kiến từ khách hàng hoặc nhóm quản lý để có quyết định khách quan.

9. Lập kế hoạch kiểm thử cho dự án phần mềm ngân hàng yêu cầu bảo mật cao

Kế hoạch kiểm thử:

- Kiểm thử bảo mật (Security Testing):
 - Xác minh xác thực (authentication) và phân quyền (authorization).
 - Kiểm tra mã hóa dữ liệu, bảo vệ API.
 - Kiểm thử tấn công SQL Injection, XSS, CSRF.
- Kiểm thử hiệu suất (Performance Testing):
 - Đánh giá tốc độ xử lý giao dịch.
 - Kiểm tra hệ thống chịu tải cao.
- Kiểm thử chức năng (Functional Testing):
 - Đảm bảo các tính năng hoạt động chính xác theo yêu cầu.
- Kiểm thử hồi quy (Regression Testing):
 - Đảm bảo các thay đổi không làm ảnh hưởng đến chức năng cũ.

10. Xử lý lỗi bảo mật nghiêm trọng sau khi triển khai phần mềm

Cách xử lý:

- Ngay lập tức khắc phục lỗi bằng cách phát hành bản vá bảo mật.
- Thông báo minh bạch với khách hàng, giải thích nguyên nhân và cách khắc phục.
- Thực hiện kiểm thử bảo mật toàn diện để đảm bảo không còn lỗ hổng khác.
- Cập nhật quy trình phát triển phần mềm để tránh lỗi tương tự trong tương lai (VD: Áp dụng DevSecOps, kiểm tra bảo mật sớm hơn).
- Đào tạo nhóm phát triển về bảo mật, đặc biệt là các kỹ thuật phòng chống tấn công.