

BÀI TẬP BUỔI 4

I. Trắc Nghiệm

1. Mục tiêu chính của kiểm thử phần mềm là gì?

Đáp án: B. Đảm bảo phần mềm đáp ứng đúng yêu cầu

2. Nhóm nào chịu trách nhiệm đảm bảo chất lượng phần mềm trong dự án?

Đáp án: C. Nhóm SQA

3. Kiểm thử các sản phẩm phi thực thi bao gồm hoạt động nào dưới đây?

Đáp án: B. Walkthrough và review tài liệu

4. Lập kế hoạch trong tiến trình phát triển phần mềm nhằm mục đích gì?

Đáp án: A. Xác định phạm vi công việc và ước tính thời gian hoàn thành

5. Loại kiểm thử nào sau đây tập trung vào việc phát hiện các lỗi chức năng của phần mềm?

Đáp án: B. Kiểm thử chức năng

6. Quản lý phiên bản tài liệu có mục tiêu gì?

Đáp án: A. Đảm bảo tài liệu luôn được cập nhật và có thể truy xuất phiên bản cũ khi cần

7. Nhóm SQA có vai trò gì trong kiểm thử phần mềm?

Đáp án: B. Đánh giá và đảm bảo quy trình phát triển phần mềm tuân thủ tiêu chuẩn chất lượng

8. Loại kiểm thử nào thường được thực hiện cuối cùng trước khi phần mềm được bàn giao cho khách hàng?

Đáp án: D. Kiểm thử chấp nhận

9. Đây là một trong những công cụ phổ biến dùng để quản lý phiên bản tài liệu?

Đáp án: B. Git

10. Hoạt động lập tài liệu trong mỗi pha phát triển phần mềm nhằm mục đích gì?

Đáp án: B. Hỗ trợ quá trình bảo trì và nâng cấp phần mềm sau khi triển khai

II. Câu hỏi ngắn

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

Nhóm SQA (Software Quality Assurance) chịu trách nhiệm đảm bảo rằng phần mềm được phát triển tuân thủ các tiêu chuẩn chất lượng và đáp ứng yêu cầu của khách hàng. Vai trò chính của nhóm SQA bao gồm:

- Xây dựng và thực thi quy trình kiểm thử: Nhóm SQA thiết lập các quy trình kiểm thử phần mềm nhằm phát hiện lỗi sớm và đảm bảo chất lượng.
- Giám sát và đánh giá quy trình phát triển phần mềm: Họ đảm bảo rằng các phương pháp phát triển phần mềm được thực hiện theo chuẩn quốc tế như ISO 9001, CMMI.
- Đánh giá chất lượng sản phẩm: Họ thực hiện kiểm thử phần mềm ở nhiều cấp độ, bao gồm kiểm thử đơn vị, tích hợp, hệ thống, chấp nhận.
- Xác minh và thẩm định (Verification & Validation): Đảm bảo rằng phần mềm được phát triển đúng yêu cầu và hoạt động ổn định trong môi trường thực tế.
- Cải tiến quy trình: Nhóm SQA đề xuất các thay đổi giúp cải thiện hiệu suất, giảm lỗi trong sản phẩm.

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 tiêu	Kiểm tra từng đơn vị (module) của phần mềm hoạt động đúng không.	Kiểm tra sự tương tác giữa các module khi kết hợp với nhau.
Phạm vi	Kiểm tra riêng lẻ từng chức năng, lớp, phương thức.	Kiểm tra kết nối giữa các module và luồng dữ liệu giữa chúng.
AI thực hiện	Lập trình viên.	Nhóm kiểm thử (QA/QC).
Thời điểm thực hiện	Ngay sau khi lập trình xong một module.	Sau khi hoàn thành kiểm thử đơn vị và tích hợp các module.
Công cụ hỗ trợ	JUnit, NUnit, TestNG (cho Java, C#).	Selenium, Postman (nếu kiểm thử API), JMeter.

Ví dụ:

- Kiểm thử đơn vị: Kiểm tra một hàm tính tổng hai số có trả về kết quả đúng không.
- Kiểm thử tích hợp: Kiểm tra xem khi hệ thống nhận dữ liệu từ module nhập liệu, module xử lý có thể xử lý chính xác và chuyển dữ liệu đến module hiển thị hay không.

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

Lập tài liệu kiểm thử là quá trình ghi lại kế hoạch kiểm thử, các trường hợp kiểm thử (test case), kết quả kiểm thử để đảm bảo quá trình kiểm thử có hệ thống và có thể lặp lại. Lợi ích của việc lập tài liệu kiểm thử:

- Hỗ trợ bảo trì phần mềm: Khi phần mềm cần nâng cấp hoặc sửa lỗi, tài liệu kiểm thử giúp các kỹ sư hiểu rõ các kịch bản kiểm thử trước đó.
 - Tạo quy trình chuẩn hóa: Giúp nhóm kiểm thử làm việc nhất quán, tránh kiểm thử thiếu sót.
 - Hỗ trợ phát hiện lỗi sớm: Xác định rõ tiêu chí kiểm thử giúp phát hiện lỗi ngay từ giai đoạn đầu.
 - Giúp đánh giá chất lượng phần mềm: Báo cáo kiểm thử cung cấp bằng chứng về mức độ sẵn sàng của phần mềm.
 - Tuân thủ quy định: Một số ngành như ngân hàng, y tế yêu cầu tài liệu kiểm thử để đáp ứng các tiêu chuẩn chất lượng.
-

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

- Thay đổi yêu cầu từ khách hàng: Khách hàng có thể thay đổi yêu cầu trong quá trình phát triển, làm ảnh hưởng đến kế hoạch ban đầu.
 - Ước lượng sai thời gian và nguồn lực: Nếu không tính toán đúng, dự án có thể bị trễ hạn hoặc thiếu nhân lực.
 - Quản lý rủi ro không hiệu quả: Không xác định trước các vấn đề có thể phát sinh dẫn đến khó khăn trong triển khai.
 - Khó khăn trong việc phối hợp giữa các nhóm: Các nhóm phát triển, kiểm thử, khách hàng cần làm việc chặt chẽ với nhau để tránh hiểu sai yêu cầu.
 - Thiếu công cụ hỗ trợ: Nếu không có công cụ quản lý tiến độ như Jira, Trello, dễ xảy ra tình trạng mất kiểm soát kế hoạch.
-

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

- Giúp theo dõi thay đổi: Khi phần mềm phát triển, tài liệu cần cập nhật liên tục. Hệ thống quản lý phiên bản giúp theo dõi ai thay đổi gì và khi nào.
 - Tránh nhầm lẫn giữa các phiên bản: Đảm bảo rằng mọi người đều làm việc trên tài liệu mới nhất, tránh sử dụng tài liệu lỗi thời.
 - Hỗ trợ khôi phục thông tin cũ: Nếu cần quay lại phiên bản trước do lỗi, có thể dễ dàng tìm lại tài liệu cũ.
 - Tăng cường bảo mật: Hạn chế quyền chỉnh sửa để tránh sửa đổi không mong muốn.
 - Công cụ quản lý phiên bản phổ biến: Git, SVN, Google Docs với tính năng lịch sử chỉnh sửa.
-

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

Loại kiểm thử	Kiểm thử sản phẩm phi thực thi	Kiểm thử sản phẩm thực thi
Định nghĩa	Kiểm thử tài liệu, thiết kế trước khi mã nguồn chạy được.	Kiểm thử trực tiếp trên phần mềm đã lập trình.
Ví dụ	Walkthrough, Inspection, Review tài liệu.	Kiểm thử đơn vị, kiểm thử hệ thống, kiểm thử hiệu suất.
Mục tiêu	Phát hiện lỗi sớm trong giai đoạn thiết kế.	Đảm bảo phần mềm chạy đúng yêu cầu.
Thời điểm thực hiện	Trước khi lập trình.	Sau khi có sản phẩm phần mềm.

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

- Áp dụng phương pháp Agile để tăng tính linh hoạt.
- Sử dụng mô hình quản lý rủi ro như SWOT, FMEA để dự báo và xử lý rủi ro sớm.
- Phân chia dự án thành các giai đoạn nhỏ để kiểm soát tốt hơn.
- Thực hiện các buổi họp đánh giá định kỳ để cập nhật tiến độ và điều chỉnh kế hoạch.

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

- Lập tài liệu kiểm thử: TestRail, Zephyr, Xray.
- Quản lý phiên bản: Git, SVN, Microsoft SharePoint.
- Quản lý dự án: Jira, Trello, Asana.

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

- Xác nhận rằng phần mềm đáp ứng yêu cầu khách hàng.
- Là bước cuối cùng trước khi triển khai, giúp tránh các lỗi nghiêm trọng sau khi đưa vào sử dụng.
- Tăng độ tin cậy của phần mềm, giảm nguy cơ phải sửa lỗi sau triển khai.

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

- Áp dụng các tiêu chuẩn như ISO 9001, CMMI.

- Tích hợp kiểm thử tự động để giảm thời gian kiểm thử thủ công.
- Thực hiện kiểm thử bảo mật để đảm bảo an toàn thông tin.
- Thực hiện kiểm thử liên tục (Continuous Testing) để phát hiện lỗi sớm.

III. 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) đảm bảo rằng các quy trình phát triển phần mềm tuân theo các tiêu chuẩn chất lượng.
 - Họ giám sát, đánh giá các hoạt động phát triển, thiết lập quy trình kiểm thử, và đề xuất cải tiến để giảm lỗi phần mềm.
2. **Sự khác nhau giữa kiểm thử đơn vị và kiểm thử tích hợp?**
 - **Kiểm thử đơn vị (Unit Testing):** Kiểm tra từng mô-đun hoặc thành phần riêng lẻ của phần mềm.
 - **Kiểm thử tích hợp (Integration Testing):** Kiểm tra sự tương tác giữa các mô-đun để đảm bảo chúng hoạt động đúng khi kết hợp lại.
3. **Tại sao việc lập tài liệu kiểm thử lại quan trọng trong mỗi dự án phần mềm?**
 - Hỗ trợ quá trình kiểm thử có hệ thống và có thể lặp lại.
 - Giúp bảo trì và cập nhật phần mềm dễ dàng hơn.
 - Cung cấp thông tin cần thiết khi cần khắc phục lỗi hoặc mở rộng phần mềm.
4. **Các thách thức khi lập kế hoạch cho các pha phát triển phần mềm?**
 - **Xác định yêu cầu không rõ ràng:** Khách hàng có thể thay đổi yêu cầu liên tục.
 - **Ước lượng thời gian và nguồn lực:** Rất khó xác định chính xác.
 - **Quản lý rủi ro:** Có thể xuất hiện lỗi bảo mật, thay đổi công nghệ...
5. **Quản lý phiên bản tài liệu ảnh hưởng đến bảo trì phần mềm như thế nào?**
 - Giúp theo dõi thay đổi trong tài liệu và quay lại phiên bản trước khi cần.
 - Hỗ trợ bảo trì dễ dàng khi cần cập nhật hoặc sửa lỗi phần mềm.
 - Tránh sai sót do sử dụng tài liệu lỗi thời.
6. **So sánh kiểm thử sản phẩm phi thực thi và kiểm thử sản phẩm thực thi?**
 - **Sản phẩm phi thực thi:** Kiểm thử tài liệu, review thiết kế, walkthrough.
 - **Sản phẩm thực thi:** Kiểm thử đơn vị, kiểm thử hệ thống, kiểm thử hiệu suất.
7. **Cách cải thiện quy trình lập kế hoạch để giảm rủi ro trong dự án phần mềm?**
 - Áp dụng phương pháp Agile để linh hoạt hơn.
 - Sử dụng công cụ quản lý dự án như Jira, Trello.
 - Dự báo rủi ro và có kế hoạch dự phòng.
8. **Các công cụ hỗ trợ lập tài liệu kiểm thử và quản lý phiên bản?**
 - **Lập tài liệu kiểm thử:** TestRail, Zephyr.
 - **Quản lý phiên bản:** Git, SVN.
9. **Tại sao kiểm thử chấp nhận lại quan trọng?**
 - Đảm bảo sản phẩm phù hợp với yêu cầu khách hàng.
 - Giúp phát hiện lỗi cuối cùng trước khi bàn giao.
 - Tăng độ tin cậy và giảm rủi ro triển khai.
10. **Các phương pháp quản lý chất lượng phần mềm trong dự án lớn?**

- Áp dụng các tiêu chuẩn ISO 9001, CMMI.
- Tích hợp kiểm thử tự động.
- Thực hiện kiểm thử bảo mật và kiểm thử hiệu suất.

IV. 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?**
 - Đánh giá lại tài liệu cùng khách hàng.
 - Xác minh yêu cầu có thay đổi không.
 - Cập nhật nếu cần và xin gia hạn thời gian nếu cần.
2. **Nhóm phát triển phát hiện một lỗi nghiêm trọng gần thời hạn bàn giao?**
 - Đánh giá mức độ ảnh hưởng của lỗi.
 - Nếu nghiêm trọng, thông báo khách hàng và tìm phương án sửa lỗi nhanh.
 - Nếu không nghiêm trọng, lên kế hoạch sửa lỗi sau triển khai.
3. **Nhóm gặp khó khăn khi quản lý phiên bản tài liệu do thay đổi liên tục?**
 - Sử dụng công cụ quản lý phiên bản như Git, Confluence.
 - Xác định quy tắc cập nhật tài liệu và người chịu trách nhiệm.
4. **Khách hàng yêu cầu thay đổi lớn trong pha cài đặt?**
 - Xác định ảnh hưởng của thay đổi.
 - Báo cáo rủi ro và đề xuất giải pháp.
 - Nếu cần, thương lượng về chi phí và thời gian.
5. **Nhóm kiểm thử phát hiện nhiều lỗi nhưng nhóm phát triển cho rằng đó là tính năng?**
 - Xem lại tài liệu yêu cầu để xác định đúng sai.
 - Nếu không rõ ràng, làm việc với khách hàng để xác minh.
 - Cập nhật tài liệu nếu cần.
6. **Khách hàng yêu cầu bổ sung tính năng mới khi đã kiểm thử tích hợp?**
 - Xác định mức độ ảnh hưởng.
 - Nếu quan trọng, đề xuất cập nhật phiên bản sau.
 - Nếu khả thi, kiểm thử lại toàn bộ hệ thống.
7. **Công ty phần mềm nhỏ muốn xây dựng nhóm SQA nhưng gặp khó khăn tài chính?**
 - Bắt đầu với quy trình đơn giản như checklist, review code.
 - Sử dụng công cụ miễn phí như SonarQube, Selenium.
 - Đào tạo nhân viên nội bộ thay vì thuê ngoài.
8. **Nhóm phát triển không thống nhất được nội dung tài liệu kiểm thử?**
 - Tổ chức cuộc họp để thống nhất tiêu chí.
 - Dựa trên yêu cầu dự án để xác định nội dung quan trọng.
 - Ghi nhận ý kiến và thống nhất quy trình cập nhật.
9. **Kiểm thử phần mềm cho ngân hàng yêu cầu bảo mật cao, cần lập kế hoạch thế nào?**
 - Tập trung kiểm thử bảo mật (penetration testing).
 - Sử dụng công cụ như OWASP ZAP, Burp Suite.
 - Kiểm thử tuân thủ ISO/IEC 27001.
10. **Khách hàng phát hiện lỗi bảo mật nghiêm trọng sau khi triển khai?**
 - Nhanh chóng đánh giá và sửa lỗi.

- Cung cấp bản vá khẩn cấp nếu cần.
- Tăng cường kiểm thử bảo mật để tránh tái diễn.