

Chương 5

Câu hỏi trắc nghiệm

Câu 1 : Kỹ thuật nào sau đây thường được sử dụng để thu thập yêu cầu khách hàng?

Đáp án: B. Phỏng vấn khách hàng

Câu 2 : Trong bước tổng hợp kết quả yêu cầu, việc nào sau đây là cần thiết?

Đáp án: B. Phân loại yêu cầu và loại bỏ vòng lặp thông tin trùng lặp

Câu 3 : Use case mô tả:

Đáp án: B. Cách người dùng tương tác với hệ thống

Câu 4 : Tại sao cần xây dựng danh sách từ khóa chuyên môn khi tìm hiểu lĩnh vực ứng dụng?

Đáp án: C. Để đảm bảo đội phát triển và khách hàng hiểu các thuật ngữ

Câu 5: Quan hệ nào sau đây trong use case mô tả một use case có thể mở rộng thêm chức năng trong một số điều kiện nhất định.

Đáp án: B. Extend

Câu 6 : Khi mô tả yêu cầu bằng ngôn ngữ tự nhiên, điều quan trọng nhất là gì?

Đáp án: C. Viết rõ ràng, dễ hiểu và tránh từ đa nghĩa

Câu 7: Mục tiêu chính của việc trích các use case là gì?

Đáp án: B. Mô tả các chức năng mà hệ thống cung cấp cho người dùng

Câu 8 : Quan hệ nào giữa các use case thể hiện việc một use case phải gọi một use case khác để thực hiện chức năng đầy đủ?

Đáp án: A. Include

Câu 9 : Hoạt động nào sau đây là bước đầu tiên trong việc xây dựng mô hình nghiệp vụ?

Đáp án: B. Trích các use case

Câu 10 : Một yêu cầu chức năng là gì?

Đáp án: A. Một yêu cầu mô tả cách hệ thống xử lý dữ liệu

Câu hỏi ngắn

1. Kỹ thuật phỏng vấn khách hàng là gì?

- Có ba loại phỏng vấn chính:
 - **Phỏng vấn không cấu trúc:** Khách hàng tự do trình bày các yêu cầu.
 - **Phỏng vấn có cấu trúc:** Câu hỏi được chuẩn bị trước, khách hàng trả lời theo các chủ đề rõ ràng.
 - **Phỏng vấn bán cấu trúc:** Kết hợp cả hai loại trên, vừa đặt câu hỏi theo trình tự, vừa tạo không gian để khách hàng bổ sung thông tin.

2. Tại sao cần tổng hợp kết quả yêu cầu?

- Giúp phân loại yêu cầu thành yêu cầu chức năng và phi chức năng.
- Loại bỏ yêu cầu trùng lặp hoặc mâu thuẫn.
- Sắp xếp các yêu cầu theo mức độ ưu tiên.

3. Use case là gì?

- Use case mô tả các chức năng mà hệ thống cung cấp cho người dùng và cách người dùng tương tác với hệ thống.

4. Mục tiêu của việc xây dựng danh sách từ khóa chuyên môn là gì?

- Giúp đội phát triển hiểu rõ thuật ngữ chuyên ngành mà khách hàng sử dụng.
- Thống nhất ý nghĩa và cách sử dụng thuật ngữ trong suốt quá trình phát triển dự án.

5. Quan hệ Include giữa các use case là gì?

- Một use case bao gồm một use case khác để hoàn thành chức năng của nó.

6. Quan hệ Extend giữa các use case là gì?

- Một use case có thể mở rộng thêm chức năng cho một use case khác trong một số điều kiện nhất định.

7. Yêu cầu phi chức năng là gì?

- Là các yêu cầu không liên quan trực tiếp đến chức năng chính nhưng ảnh hưởng đến hiệu suất, bảo mật, khả năng mở rộng, tính thân thiện với người dùng, v.v..

8. Mô hình nghiệp vụ là gì?

- Là mô hình mô tả các quy trình, hoạt động của doanh nghiệp, giúp hiểu rõ các yêu cầu của hệ thống phần mềm.

9. Điều kiện trước và điều kiện sau của use case là gì?

- **Điều kiện trước:** Các điều kiện phải được thỏa mãn trước khi use case có thể thực hiện.
- **Điều kiện sau:** Kết quả mong muốn hoặc trạng thái hệ thống sau khi use case hoàn thành.

10. Một số lưu ý khi trích các use case là gì?

- Đảm bảo mỗi use case thể hiện một chức năng rõ ràng.
- Tránh viết quá chi tiết hoặc quá chung chung.
- Kiểm tra sự nhất quán giữa các use case.

Câu hỏi tình huống

1. Khách hàng không hiểu rõ các thuật ngữ kỹ thuật trong tài liệu yêu cầu. Là trưởng nhóm phát triển, bạn sẽ làm gì?

- **Giải pháp:**
 - Sử dụng ngôn ngữ đơn giản, dễ hiểu khi giao tiếp và trình bày tài liệu.
 - Tổ chức các buổi giải thích, demo trực quan hoặc mô phỏng để khách hàng dễ hình dung.
 - Chuẩn bị một bảng thuật ngữ (Glossary) giải thích các khái niệm kỹ thuật chính.
 - Duy trì sự tương tác thường xuyên với khách hàng để kiểm tra mức độ hiểu rõ.
-

2. Trong quá trình lấy yêu cầu, khách hàng liên tục thay đổi ý kiến về chức năng cần thiết. Đội phát triển nên xử lý ra sao?

- **Giải pháp:**
 - Áp dụng phương pháp **Agile/Scrum** để linh hoạt cập nhật và thích ứng với thay đổi.
 - Xây dựng **Product Backlog** để quản lý các yêu cầu và sắp xếp độ ưu tiên.
 - Xác định rõ phạm vi dự án (Scope) và yêu cầu khách hàng phê duyệt các thay đổi trước khi triển khai.
 - Thực hiện **MoSCoW** (Must have, Should have, Could have, Won't have) để ưu tiên hóa các chức năng cần thiết.
-

3. Một dự án phần mềm quản lý bán hàng gặp vấn đề khi các yêu cầu được mô tả quá chung chung, khó thực hiện. Đội phát triển cần làm gì để khắc phục?

- **Giải pháp:**
 - Áp dụng phương pháp **phân rã yêu cầu** (Decomposition) thành các yêu cầu cụ thể, chi tiết hơn.
 - Thực hiện **phỏng vấn sâu (Interview)** hoặc **workshop** với các bên liên quan để làm rõ yêu cầu.
 - Xây dựng **Use Case Diagram** và **User Story** để minh họa các chức năng rõ ràng.
 - Yêu cầu khách hàng cung cấp các tình huống sử dụng thực tế (Case Study) để hiểu rõ nhu cầu.

4. Khách hàng đưa ra yêu cầu không rõ ràng và chỉ có thể mô tả một cách sơ lược. Làm thế nào để thu thập yêu cầu đầy đủ từ khách hàng?

- **Giải pháp:**

- Tổ chức các buổi **phỏng vấn chi tiết** với các bộ phận liên quan để thu thập thông tin từ nhiều góc độ.
- Sử dụng **kỹ thuật quan sát** quy trình làm việc thực tế để xác định các yêu cầu ngầm.
- Áp dụng **Prototyping** (xây dựng mô hình mẫu) để khách hàng dễ hình dung và bổ sung yêu cầu.
- Sử dụng **Bảng hỏi (Questionnaire)** hoặc **checklist** để đảm bảo không bỏ sót thông tin quan trọng.

5. Trong quá trình xây dựng mô hình nghiệp vụ, một số use case bị trùng lặp về chức năng. Bạn sẽ xử lý tình huống này như thế nào?

- **Giải pháp:**

- Rà soát và đối chiếu các **Use Case** để xác định mức độ trùng lặp và hợp nhất các chức năng tương đồng.
- Phân loại các **Use Case** theo mức độ chi tiết (Main flow, Alternative flow, Exception flow).
- Sử dụng **Extend/Include** trong **UML** để xử lý các kịch bản phụ và chia sẻ chức năng chung.
- Làm việc với khách hàng và các bên liên quan để xác nhận yêu cầu thực sự cần thiết, tránh dư thừa.

Câu 6: Khách hàng yêu cầu bổ sung một tính năng mới khi phần mềm đã hoàn thành pha kiểm thử tích hợp. Điều phát triển nên làm gì?

Khi khách hàng yêu cầu bổ sung một tính năng mới sau khi phần mềm đã hoàn thành pha kiểm thử tích hợp, đội phát triển cần thực hiện các bước sau:

1. **Đánh giá yêu cầu:** Phân tích yêu cầu mới để hiểu rõ phạm vi, tác động đến hệ thống hiện tại, và các tài nguyên cần thiết để triển khai.
2. **Thảo luận với khách hàng:** Làm việc với khách hàng để xác định mức độ ưu tiên của tính năng mới và khả năng tích hợp nó vào lộ trình phát triển.
3. **Đánh giá rủi ro:** Xem xét tác động của việc bổ sung tính năng mới đến lịch trình, chi phí, và chất lượng của phần mềm.
4. **Cập nhật kế hoạch:** Nếu quyết định triển khai, cần cập nhật kế hoạch dự án, bao gồm thiết kế, phát triển, kiểm thử, và triển khai tính năng mới.
5. **Thực hiện và kiểm thử:** Tiến hành phát triển tính năng mới, sau đó kiểm thử kỹ lưỡng (bao gồm kiểm thử tích hợp và hồi quy) để đảm bảo không ảnh hưởng đến các chức năng hiện có.
6. **Giao tiếp minh bạch:** Thông báo cho khách hàng về tiến độ và bất kỳ thay đổi nào trong lịch trình hoặc chi phí.

Câu 7: Một công ty phát triển phần mềm nhỏ muốn xây dựng nhóm SQA nhưng gặp khó khăn về ngân sách. Hãy đề xuất giải pháp.

Để xây dựng nhóm SQA (Software Quality Assurance - Đảm bảo Chất lượng Phần mềm) với ngân sách hạn chế, công ty có thể áp dụng các giải pháp sau:

1. **Tận dụng nhân sự hiện có:** Đào tạo các thành viên trong nhóm phát triển (developer) hoặc nhóm kiểm thử (tester) để đảm nhận vai trò SQA. Điều này giúp tiết kiệm chi phí tuyển dụng.
2. **Thuê ngoài (Outsourcing):** Cân nhắc thuê ngoài một nhóm SQA từ các công ty dịch vụ bên thứ ba trong giai đoạn đầu để giảm chi phí cố định.
3. **Tự động hóa kiểm thử:** Đầu tư vào công cụ tự động hóa kiểm thử để giảm khối lượng công việc thủ công, từ đó giảm nhu cầu về nhân sự SQA.
4. **Xây dựng quy trình SQA nội bộ:** Thiết lập các quy trình kiểm tra chất lượng đơn giản nhưng hiệu quả, như peer review (đánh giá chéo) hoặc checklist kiểm tra, để đảm bảo chất lượng mà không cần đội ngũ SQA lớn.
5. **Tăng dần quy mô:** Bắt đầu với một nhóm SQA nhỏ (1-2 người) và mở rộng dần khi ngân sách cho phép.
6. **Sử dụng công cụ miễn phí:** Tận dụng các công cụ SQA mã nguồn mở như Selenium, JIRA (phiên bản miễn phí), hoặc TestRail để hỗ trợ công việc mà không tốn chi phí lớn.

Câu 8: Trong quá trình làm tài liệu kiểm thử, nhóm phát triển không thống nhất được về nội dung cần đưa vào tài liệu. Là trưởng nhóm, bạn sẽ giải quyết vấn đề này như thế nào?

Là trưởng nhóm, tôi sẽ giải quyết vấn đề này bằng các bước sau:

1. **Tổ chức cuộc họp:** Tập hợp tất cả các thành viên trong nhóm để thảo luận và lắng nghe ý kiến của từng người về nội dung tài liệu kiểm thử.
2. **Xác định mục tiêu:** Làm rõ mục tiêu của tài liệu kiểm thử (ví dụ: tài liệu dùng để làm gì, đối tượng sử dụng là ai – khách hàng, đội phát triển, hay nhóm kiểm thử).
3. **Tham khảo tiêu chuẩn:** Sử dụng các tiêu chuẩn hoặc hướng dẫn phổ biến (như IEEE 829 – tiêu chuẩn về tài liệu kiểm thử phần mềm) để làm cơ sở thống nhất nội dung. Một tài liệu kiểm thử thường bao gồm: phạm vi kiểm thử, mục tiêu, chiến lược kiểm thử, tài nguyên, lịch trình, và các trường hợp kiểm thử (test cases).
4. **Phân công trách nhiệm:** Phân công từng thành viên chịu trách nhiệm cho một phần cụ thể của tài liệu để đảm bảo tất cả ý kiến đều được xem xét.
5. **Thống nhất và phê duyệt:** Sau khi hoàn thành bản nháp, tổ chức một buổi đánh giá chung để thống nhất nội dung cuối cùng và phê duyệt tài liệu.
6. **Ghi nhận bài học:** Lưu lại các vấn đề phát sinh và cách giải quyết để cải thiện quy trình làm tài liệu trong tương lai.

Câu 9: Dự án phát triển phần mềm cho một ngân hàng yêu cầu bảo mật cao. Đề xuất cách lập kế hoạch kiểm thử cho dự án này.

Để lập kế hoạch kiểm thử cho một dự án phát triển phần mềm ngân hàng yêu cầu bảo mật cao, cần thực hiện các bước sau:

1. **Phân tích yêu cầu bảo mật:** Xác định các yêu cầu bảo mật cụ thể của hệ thống ngân hàng, ví dụ: bảo vệ dữ liệu khách hàng, tuân thủ các tiêu chuẩn như PCI DSS, GDPR, hoặc các quy định địa phương.
2. **Xác định phạm vi kiểm thử:** Tập trung vào các khía cạnh bảo mật, bao gồm:
 - Kiểm thử xâm nhập (penetration testing) để phát hiện lỗ hổng.
 - Kiểm thử mã hóa dữ liệu (encryption) để đảm bảo dữ liệu được mã hóa an toàn khi truyền tải và lưu trữ.
 - Kiểm thử xác thực và phân quyền (authentication và authorization) để đảm bảo chỉ người dùng hợp lệ mới truy cập được hệ thống.
 - Kiểm thử chống tấn công (ví dụ: SQL injection, XSS, CSRF).
3. **Lựa chọn công cụ kiểm thử:** Sử dụng các công cụ chuyên dụng như OWASP ZAP, Burp Suite, hoặc Nessus để kiểm tra bảo mật. Ngoài ra, cần các công cụ kiểm thử hiệu năng (performance testing) và kiểm thử chức năng (functional testing).

4. **Lên lịch kiểm thử:** Phân bổ thời gian hợp lý cho từng loại kiểm thử, ưu tiên kiểm thử bảo mật sớm và thường xuyên trong suốt vòng đời phát triển.
5. **Đào tạo đội ngũ:** Đảm bảo đội kiểm thử được đào tạo về bảo mật phần mềm và các mối đe dọa phổ biến trong ngành ngân hàng.
6. **Tích hợp kiểm thử liên tục:** Áp dụng CI/CD (Continuous Integration/Continuous Deployment) để kiểm thử bảo mật tự động trong quá trình phát triển.
7. **Báo cáo và khắc phục:** Ghi nhận chi tiết các lỗi bảo mật, ưu tiên khắc phục theo mức độ nghiêm trọng, và kiểm thử lại để đảm bảo lỗi đã được sửa.
8. **Tuân thủ quy định:** Đảm bảo kế hoạch kiểm thử đáp ứng các yêu cầu pháp lý và tiêu chuẩn bảo mật của ngành ngân hàng.

Câu 10: Sau khi triển khai phần mềm, khách hàng phát hiện ra một số lỗi bảo mật nghiêm trọng. Điều phát triển cần xử lý ra sao để khách phục vấn đề và lấy lại niềm tin từ khách hàng?

Để xử lý các lỗi bảo mật nghiêm trọng sau triển khai và lấy lại niềm tin từ khách hàng, đội phát triển cần thực hiện các bước sau:

1. **Phản hồi nhanh chóng:** Ngay khi nhận được thông báo về lỗi, liên hệ với khách hàng để xác nhận vấn đề và cam kết giải quyết nhanh chóng.
2. **Phân tích lỗi:** Tiến hành phân tích chi tiết để xác định nguyên nhân gốc rễ (root cause) của lỗi bảo mật, đồng thời đánh giá mức độ thiệt hại (nếu có).
3. **Khắc phục lỗi:**
 - Ưu tiên vá lỗi ngay lập tức, tập trung vào các lỗ hổng nghiêm trọng nhất.
 - Triển khai bản vá (patch) hoặc bản cập nhật phần mềm nhanh chóng, đồng thời kiểm thử kỹ lưỡng để đảm bảo không phát sinh lỗi mới.
4. **Báo cáo minh bạch:** Cung cấp cho khách hàng báo cáo chi tiết về lỗi, nguyên nhân, và cách khắc phục. Minh bạch trong giao tiếp giúp xây dựng lại niềm tin.
5. **Tăng cường bảo mật:**
 - Thực hiện kiểm tra bảo mật toàn diện (security audit) trên toàn bộ hệ thống để phát hiện các lỗ hổng khác.
 - Cải thiện quy trình phát triển và kiểm thử, ví dụ: tích hợp kiểm thử bảo mật sớm hơn trong vòng đời phát triển (shift-left testing).
6. **Đào tạo và cải tiến:** Đào tạo đội ngũ về bảo mật phần mềm và cập nhật quy trình phát triển để tránh các lỗi tương tự trong tương lai.
7. **Hỗ trợ khách hàng:** Cung cấp hỗ trợ kỹ thuật liên tục cho khách hàng, đồng thời đề xuất các biện pháp giảm thiểu rủi ro (nếu cần).
8. **Xây dựng niềm tin:** Đưa ra cam kết dài hạn về chất lượng và bảo mật, đồng thời theo dõi hệ thống sau triển khai để đảm bảo không có vấn đề phát sinh.