

## BÀI TẬP Củng Cố Lí Thuyết LECTURE05- TESTING

### 1) Câu hỏi trắc nghiệm

- C1. B - Đảm bảo phần mềm đáp ứng đúng yêu cầu (mục tiêu kiểm thử nêu rõ “đúng như yêu cầu của khách hàng”).
- C2. C - Nhóm SQA (chịu trách nhiệm bảo đảm quy trình/sản phẩm tuân chuẩn chất lượng).
- C3. B - Walkthrough & review tài liệu là dạng kiểm thử phi thực thi.
- C4. A - Lập kế hoạch để xác định công việc, phân bổ nguồn lực, ước tính thời gian.
- C5. B - Kiểm thử chức năng tập trung bắt lỗi chức năng. (Slide liệt kê nhóm kiểm thử thực thi: functional/performance/security).
- C6. A - Quản lý phiên bản nhằm cập nhật, theo dõi và truy xuất phiên bản khi cần (mục tiêu: quản lý chặt chẽ, nhất quán).
- C7. B - SQA đánh giá và bảo đảm quy trình tuân chuẩn chất lượng.
- C8. D - Kiểm thử chấp nhận thường là bước cuối trước bàn giao (được nêu cùng kế hoạch kiểm thử hệ thống & chấp nhận).
- C9. B - Git là công cụ phổ biến để quản lý phiên bản tài liệu.
- C10. B- Tài liệu giúp hỗ trợ phát triển & bảo trì/nâng cấp sau triển khai.

### 2) Câu hỏi trả lời ngắn

#### 1. SQA là gì & vai trò?

Nhóm đảm bảo chất lượng phần mềm, thiết lập/giám sát quy trình, đánh giá kết quả kiểm thử để sản phẩm đáp ứng tiêu chuẩn & yêu cầu.

#### 2. Kiểm thử đơn vị (Unit test):

Kiểm tra phần nhỏ nhất của chương trình (hàm/module) do dev viết, nhằm phát hiện lỗi sớm ở mức thấp.

#### 3. Mục tiêu kiểm thử chấp nhận:

Xác nhận hệ thống đáp ứng yêu cầu người dùng/khách hàng để chấp nhận bàn giao (được kế hoạch chung nêu kèm “acceptance testing”).

#### 4. Hoạt động chính của kiểm thử phi thực thi:

Walkthrough, Inspection, Review (soát lỗi trên tài liệu: yêu cầu, thiết kế, HDSD).

5. **Vì sao phải làm tài liệu mỗi pha?**

Để rõ ràng, dễ phát triển, thuận tiện bảo trì về sau.

6. **Quản lý phiên bản tài liệu là gì?**

Theo dõi thay đổi, lưu trữ, bảo mật & nhất quán giữa các phiên bản; dùng công cụ như Git/SVN.

7. **Các loại kiểm thử chính trong thực thi:**

Functional, Performance, Security.

8. **Kiểm thử tích hợp (Integration test):**

Kiểm tra sự tương tác/giao tiếp giữa các module đã qua unit test, phát hiện lỗi giao diện tích hợp.

9. **Lập kế hoạch cho các pha gồm gì?**

Phân tích yêu cầu/phạm vi, xác định công việc & ưu tiên, quản lý rủi ro; nêu rõ theo từng pha: yêu cầu/thiết kế/cài đặt/kiểm thử/bảo trì.

10. **Làm tài liệu kiểm thử gồm gì?**

Kế hoạch kiểm thử và báo cáo kết quả kiểm thử.

### 3) Gợi ý thảo luận nhóm

1. **Vai trò SQA:** đảm bảo quy trình tuân chuẩn; độc lập đánh giá chất lượng; thúc đẩy cải tiến quy trình.
2. **Unit vs Integration:** Unit cô lập hàm/lớp; Integration kiểm tra giao tiếp giữa các module; phạm vi & dữ liệu test khác nhau.
3. **Vì sao tài liệu kiểm thử quan trọng:** thống nhất cách test, truy vết kết quả, hỗ trợ bảo trì & tái sử dụng ca kiểm thử.
4. **Thách thức khi lập kế hoạch:** yêu cầu thay đổi, ước lượng khó, rủi ro ẩn; cách khắc phục: quản lý rủi ro & ưu tiên rõ ràng.
5. **Quản lý phiên bản ảnh hưởng bảo trì:** giúp truy xuất lịch sử, so sánh thay đổi, tránh nhầm lẫn giữa các bản tài liệu.
6. **Phi thực thi vs thực thi:** phi thực thi rà soát tài liệu (Walkthrough/Inspection/Review), thực thi chạy phần mềm (functional/performance/security).
7. **Cải thiện lập kế hoạch để giảm rủi ro:** thêm buffer, quản lý thay đổi, hợp mốc (milestone), rà soát định kỳ.

8. **Công cụ hỗ trợ tài liệu & phiên bản:** Mẫu test plan/test report + Git/SVN, wiki nội bộ.
9. **Vì sao UAT quan trọng:** xác nhận phù hợp nghiệp vụ & chấp nhận bàn giao.
10. **Quản lý chất lượng cho dự án lớn:** SQA độc lập, checklist chuẩn hoá, tự động hoá test, quản lý rủi ro liên tục.

#### 4) Câu hỏi tình huống

1. **Trước bàn giao, KH yêu cầu kiểm tra lại toàn bộ tài liệu yêu cầu/thiết kế:**
  - Lập danh sách tài liệu & phiên bản; khoá phiên bản để rà soát.
  - Tổ chức review/inspection tập trung.
  - Ghi biên bản thay đổi & cập nhật kho phiên bản (Git/SVN).
2. **Phát hiện lỗi nghiêm trọng khi sắp đến hạn:**
  - Báo cáo ngay mức độ nghiêm trọng & rủi ro.
  - Ưu tiên fix, tách nhánh hotfix, bổ sung test hồi quy.
  - Thương lượng phạm vi/hạn nếu cần, có kế hoạch giảm rủi ro.
3. **Khó quản lý phiên bản tài liệu vì thay đổi liên tục:**
  - Chuẩn hoá đặt tên & workflow (branch/tag).
  - Dùng Git/SVN, PR/Review bắt buộc.
  - Lập changelog.
4. **KH yêu cầu thay đổi lớn ở pha cài đặt:**
  - Phân tích tác động (thiết kế, code, test, lịch).
  - Cập nhật kế hoạch pha liên quan & rủi ro.
  - Trình CR (change request) và thương lượng phạm vi/thời gian/chi phí.
5. **Tester nói “lỗi”, Dev nói “tính năng”:**
  - Dựa trên tài liệu yêu cầu/test case & tiêu chí chấp nhận.
  - Hợp phân xử với PO/BA/SQA, cập nhật tài liệu nếu cần.
6. **Muốn thêm tính năng mới sau khi xong tích hợp:**
  - Đưa vào backlog/phiên bản sau; nếu bắt buộc, tạo CR, đánh giá tác động & điều chỉnh kế hoạch test hệ thống/chấp nhận.

7. **Cty nhỏ muốn lập nhóm SQA nhưng hạn chế ngân sách:**

- Bắt đầu nhỏ: 1–2 người kiêm nhiệm + checklist chuẩn hoá.
- Dùng công cụ miễn phí/mã mở (Git, wiki).
- Tập trung quy trình cốt lõi & review tài liệu định kỳ.

8. **Không thống nhất nội dung tài liệu kiểm thử:**

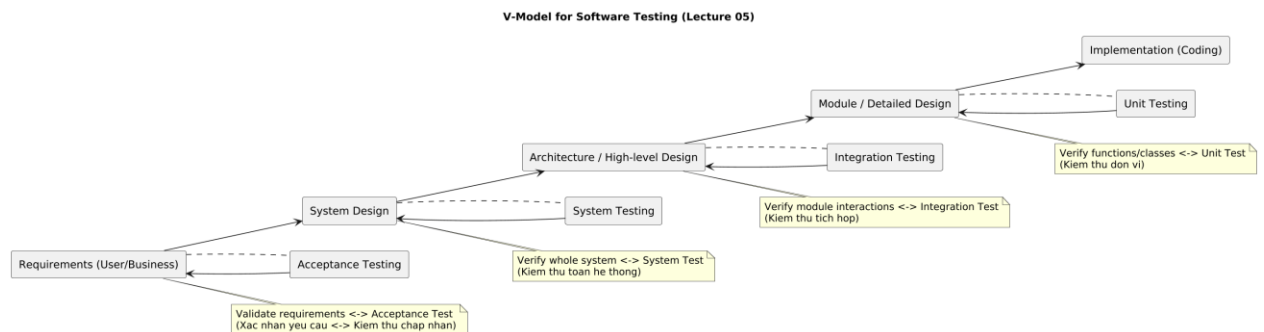
- Chuẩn hoá cấu trúc test plan/test report.
- Hợp đồng thuận tiêu chí bao phủ test.
- SQA điều phối review để chốt phiên bản.

9. **Dự án ngân hàng yêu cầu bảo mật cao , lập kế hoạch kiểm thử:**

- Thêm kiểm thử bảo mật (vét lỗ hổng, kiểm soát truy cập, mã hoá), hiệu năng, tuân thủ.
- Test chấp nhận bảo mật với kịch bản nghiệp vụ nhạy cảm.

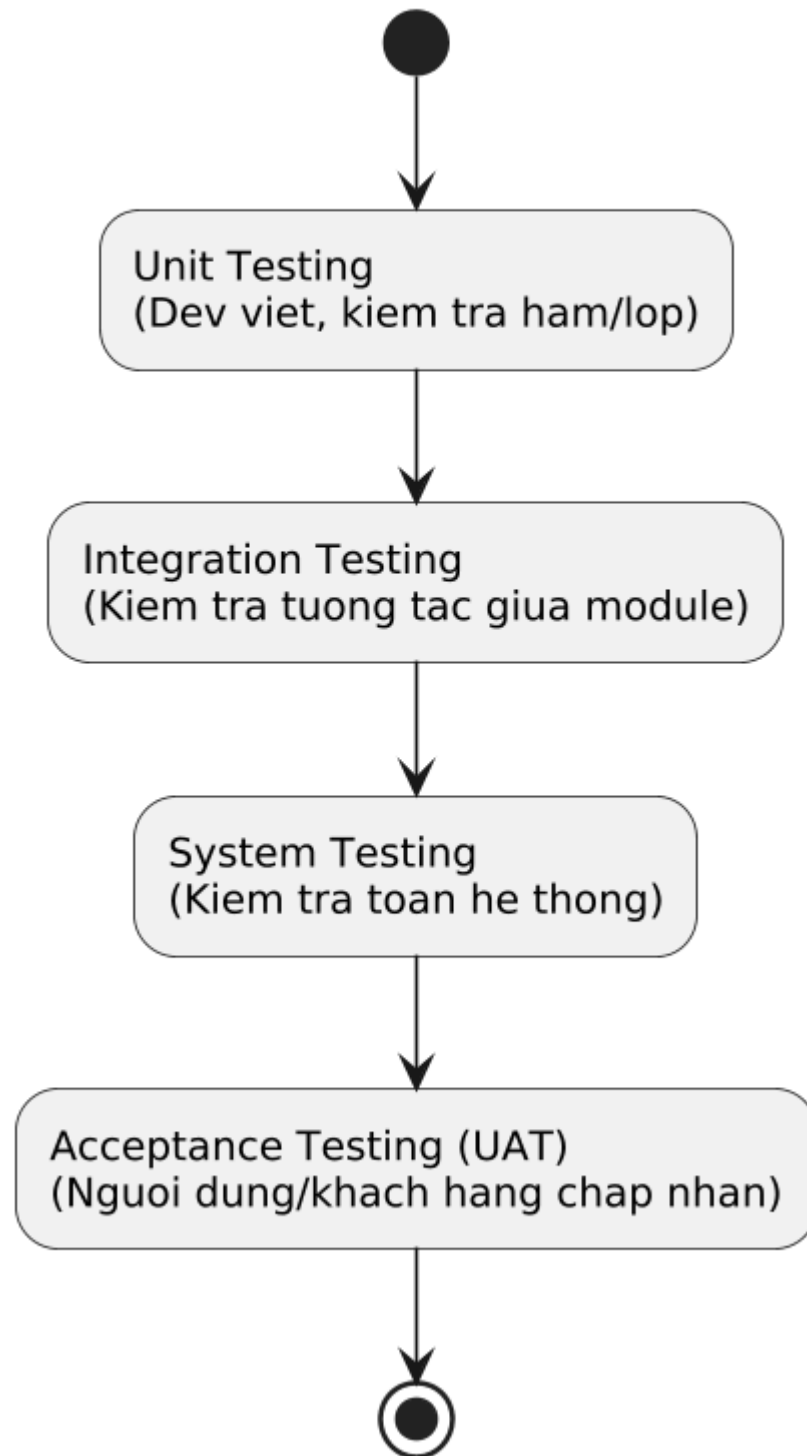
10. **Sau triển khai phát hiện lỗi bảo mật nghiêm trọng:**

- Kích hoạt quy trình sự cố: cô lập rủi ro, hotfix, thông báo minh bạch.
- Điều tra nguyên nhân gốc (RCA), bổ sung test bảo mật & cập nhật tài liệu, quy trình.



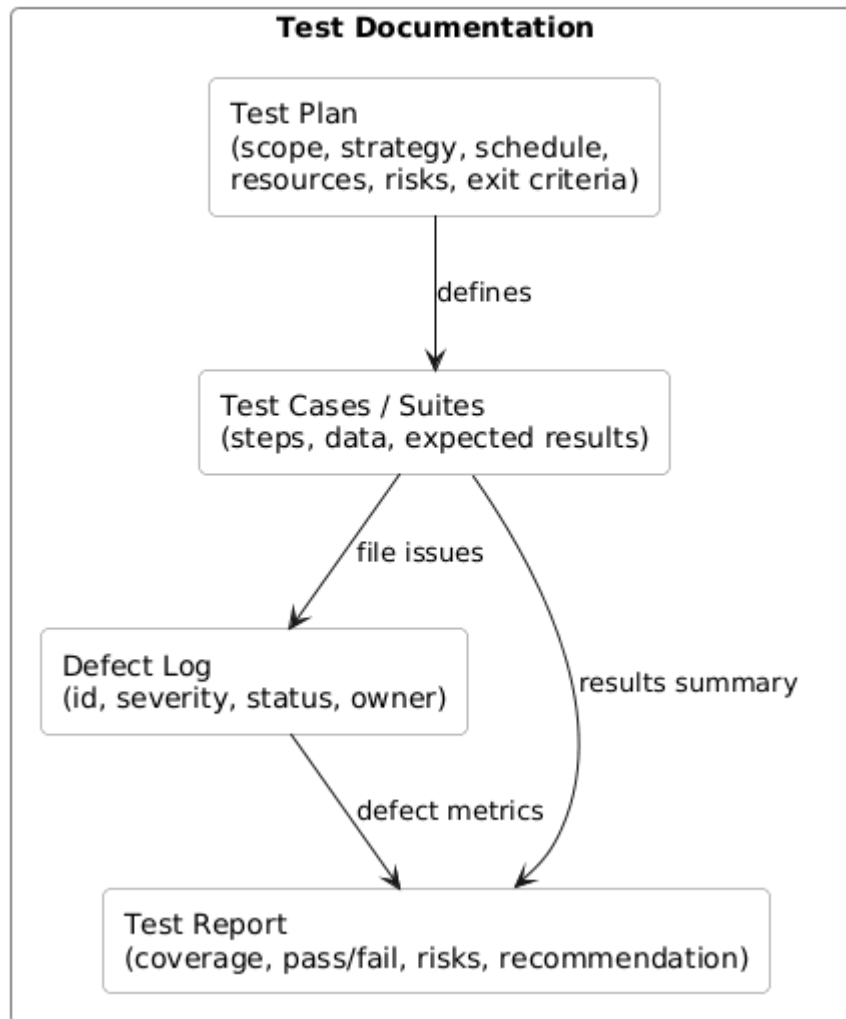
## V-MODEL

## Software Testing Levels



TESTING LEVEL

## Test Documentation (Plan → Cases → Report)



**TEST DOC**