**PHẦN 1: Các khái niệm căn bản về kiểm thử phần mềm**

**Bài 1: Kiểm thử phầm mềm là gì?**

**Các cách nhìn khác nhau về chất lượng (quality) của một sản phẩm phần mềm**

* Trước khi nói về kiểm thử phần mềm, chúng ta phải nói về chất lượng sản phẩm phần mềm:
  + Đứng ở các góc nhìn khác nhau, định nghĩa về một sản phẩm phần mềm chất lượng cũng sẽ khác nhau
    - Từ phía người dùng: sản phẩm phù hợp với mục đích sử dụng của người đó
    - Từ phía nhà phân bổ, cung cấp sản phẩm (product view): một sản phẩm đạt chất lượng là sản phẩm có được các đặc tính về sản phẩm, đạt được những tiêu chú đánh giá do nhà cung cấp sản phẩm đề ra
      * Tnhs chính xác
      * Tính bảo mật
      * Tính dễ sử dụng
      * …
    - Từ phía nhà sản suất ra sản phẩm phân mềm: đạt được chất lượng cần thiết để đưa ra dùng hay sử dụng phải được xây dựng đúng như những yêu cầu đặc tả (do người sử dụng hoặc nhà cung cấp sản phẩm đề ra)
  + Từ những góc nhìn trên, tam giác chất lượng của sản phẩm phần mềm sẽ được xây dựng như sau:
    - Đỉnh 1: view của nhà cung cấp sản phẩm (Product’s view)
    - Đỉnh 2: view của người sử dụng (user’s view)
    - Đỉnh 3: view của nhà sản xuất (manufacture’s view)
* Một sản phẩm phần mềm sẽ được xây dựng dựa trên yêu ầu người dùng + yêu cầu đặc tả
  + Thông thường, trong quy trình sản xuất phần mềm, nhà cung cấp sản phẩm phần mềm sẽ theo những yêu cầu của user để viết ra một customer requirement (hay có lúc được gọi là high level requirement)
  + Sau đó nhà sản xuất sẽ phát triển thành một tài liệu là requirement specification (yêu cầu đặc tả) mô tả chi tiết yêu cầu về các chức năng, tính năng của sản phẩm 🡪 Developers sẽ dựa vào đó để xẩy dựng / phát triển sản phẩm phần mềm
  + Khoảng cách giữa các đỉnh của tam giác giữa sản phẩm thực tế và yêu cầu chức năng, tính năng của sản phẩm càng lớn, nghĩa là **chất lượng (quality)** của sản phẩm càng kém và ngược lại.

**Kiểm thử phần mềm là gì?**

* Định nghĩa:
  + Theo IEEE 829: Testing là một quá trình thực hiện đánh giá một hệ thống, một sản phẩm tự động bằng tay để xác nhận rằng sản phẩm có thỏa mãn đặc tả hay không
  + Theo ISTQB: Testing là một quá trìn bao gồm một vòng lặp các hoạt động cả tĩnh và động, từ khi lập kế hoạch, chuẩn bị và đánh giá sản phẩm để xác định sản phẩm có thõa mãn yêu cầu đặc tả không, chứng minh sản phẩm đó có phù hợp với mục đích sử dụng không và để phát hiện lỗi.
* Mục tiêu cyả kiểm thử phần mềm:
  + Xác định sản phẩm có thõa mãn yêu cầu đặc tả hay không
  + Xác định sản phẩm có thõa mãn những cái người dùng cần không
  + Có được tự tin để cung cấp một sản phẩm có chất lượng không
  + Xác định các lỗi, nguyên nhân để đưa ra các hành động để hạn chế, ngăn chặn các lỗi trong tương lai
  + Có thể đưa ra các cải tiến quy trình, tìm thêm các công cụ để phục vụ công việc testing hiệu quả hơn

**Sự khác nhau giữa lý thuyết và thực tế:**

1. Testing là một công việc đơn giản, dễ làm: thực tế, testing là 1 công việc thách thức, có những trường hợp kiểm thử đòi hỏi kỹ năng phân tích cao
2. Ai cũng có thể làm manual testing: thực tế testing yêu cầu nhiều kỹ năng cần có và phải có để làm nghề tester
3. Testing để đảm bảo phần mềm sau khi kiểm thử không có lỗi, nhưng thức tế không phải lúc nào việc testing cũng có thể phát hiện được tất cả các lỗi của phần mềm
4. Test tự động sẽ tốt hơn, hiệu quả hơn việc test bằng tay: không thể tự động test hoàn toàn tất cả các trường hợp của phần mềm
5. Testing và debugging là giống nhau: thực tế là hoàn toàn khác nhau:
   1. Testing là tìm lỗi, debugging là hành động để sửa lỗi
   2. Testing thực hiện bởi tester, còn debugging thực hiện bởi developers
   3. Testing là để tìm được càng nhiều lỗi càng tốt, còn debugging là để loại bỏ những lỗi đó

1 B

2 D

3 D

4 D

5 A

**--------------- --------------- --------------- --------------- --------------- ---------------**

**Bài 2: Tại sao phải thực hiện kiểm thử phần mềm?**

**Tại sao việc kiểm thử phần mềm lại cần thiết**

1. Sản phẩm phần mềm nào cũng chắc chắn có lỗi, do đó sẽ luôn cần một người, nhóm người hoặc tổ chức độc lập đứng ra kiểm thử xem sản phẩm có lỗi hay không
2. Kiểm thử giúp tìm hiểu độ tin cậy của phần mềm
3. Các lỗi trong quá trình phần mềm được đưa vào sử dụng có thể rất tốn chi phí để khắc phụ và có thể gây ra những hậu quả nghiêm trọng
4. Tránh được sự kiện tụng của khách hàng
5. Doanh nghiệp muốn phát triển trong một lĩnh vực nào đó luôn cần đảm bảo sản phẩm do mình cung cấp đến người dùng có chất lượng tốt 🡪 cần kiểm thử là đương nhiên

**Lỗi trong sản phẩm phần mềm**

* Lỗi trong phần mềm có thể dẫn đến:
  + Mất tiền để xử lý, khắc phục
  + Thời gian để xử lý, khắc phục
  + Mất sự tin tưởng từ phía khách hàng
  + Thương tổn về tài chính, con người.
* Các khái niệm về lỗi trong sản phẩm, dự án phần mềm:
  + Error: là chỉ hành động của con người gây ra một kết quả không đúng
  + Fault: là một bước sai, quy trình sai hay định nghĩa dữ liệu sai, fault là kết quả của error
  + Failure: là lỗi tìm được trong quá trình sản phẩm phần mềm đã được đưa vào sử dụng, đó chính là độ lệch (GAP) giữa sản phẩm và mong muốn của người sử dụng
* Lỗi xuất hiện khi nào:
  + Bất cứ khi nào tạo ra một sản phẩm thì đều có khả năng sản phẩm đó chứa lỗi
* Chi phí khắc phục lỗi thay đổi như thế nào?
  + Chi phí khắc phục, xử lý lỗi sẽ tăng dần theo thời gian
  + Những lỗi mà ở giai đoạn live use mới được phát hiện, nếu chiếu theo thuật ngữ về lỗi thì sẽ là failure, là những lỗi tốn nhiều chi phí nhất để sửa chữa và khắc phục

1 C

2 C

3 A

4 B

5 C

**--------------- --------------- --------------- --------------- --------------- ---------------**

**Bài 3: các chân lý trong kiểm thử phần mềm**

**Chân lý thứ nhất: testing shows presence of defects (kiểm thử sẽ chỉ ra các lỗi hiện diện hay việc kiểm thử sẽ tìm ra lỗi)**

* Testing sẽ chỉ ra có lỗi tồn tại trong hệ thống, chứ không chứng minh rằng sản phẩm phần mềm đó không có lỗi, hay chứng minh rằng sản phẩm phần mềm đó hoạt động đúng
* Testing làm giảm xác suất lỗi chưa tìm thấy, nhưng ngay cả khi không tìm thấy lỗi, nó cũng không chứng minh được rằng sản phẩm phần mềm đó hoạt động chính xác và không có lỗi

**Chân lý thứ hai: kiểm thử tất cả các trường hợp là không khả thi**

* Kiểm thử mọi thứ là không thể vì thời gian và chi phí sẽ tăng theo cấp số nhân
* Thay vì kiểm thử toàn bộ, việc phân tích rủi ro và dựa trên mức độ ưu tiên chúng ta có thể tập trung kiểm thử vào một số điểm cần thiết

**Chân lý thứ 3: kiểm thử sớm**

* Các hoạt động kiểm thử nên bắt đầu càng sớm càng tốt trong quy trình phát triển và nên tập trung vào các mục tiêu đã định ra từ trước
* Thực hiện kiểm thử sớm giúp phát hiện các lỗi sớm, chi phí để sửa chữa, khắc phục các lỗi đó sẽ rẻ hơn

**Chân lý thứ 4: Defect clustering (sự tập trung lỗi)**

* Trong quá trình thực hiện kiểm thử phần mềm, sẽ có lúc chúng ta thẫy lỗi tập trung ở một số ít các module trong sản phẩm, thông thường là những chức năng chính đòi hỏi xử lý phức tạp, hoặc chức năng do người có ít kinh nghiệm thực hiện

**Chân lý thứ 5: Pesticide Paradox (nghịch lý thuốc trừ sâu)**

* Nếu việc kiểm thử lặp đi lặp lại nhiều lần, thì cuối cùng sẽ có 1 số trường hợp kiểm thử không còn tìm thấy lỗi mới nào 🡺 nghịch lý thuốc trừ sâu. Để khắc phục điều này, cần xem xét sửa đổi các tình huống kiểm thử, các test cases cũng cần được “làm mới” để tìm ra các lỗi tiềm ẩn hơn nữa

**Chân lý thứ 6: Testing is context dependent (kiểm thử theo ngữ cảnh)**

**-** Với mỗi sản phẩm khác nhau, trong những lãnh vực khác nhau thì việc kiểm thử cũng sẽ khác nhau

**Chân lý thứ 7: sự sai lầm về việc không có lỗi**

* Việc tìm và sửa chữa lỗi sẽ là vô nghĩa nếu hệ thống xây dựng xong không thể dùng được và không đáp ứng được nhu cầu và mong đợi của người dùng
* Nghĩa là phần mềm, sau khi đã làm đầy đủ công đoạn để phát triển ra một sản phẩm phần mềm hoàn chỉnh nhưng sản phẩm đưa ra đến tay người dùng không đáp ứng được nhu cầu của người sử dụng thì phần mềm đó coi như thất bại, mặc dù các hoạt động đảm bảo chất lượng phần mềm, kiểm thử vẫn diễn ra đầy đủ

**1 D**

**2 D**

**3 C**

**4 C**

**5 B**

**--------------- --------------- --------------- --------------- --------------- --------------- --------------- --------------- --------------- --------------- --------------- ---------------**

**Phần 2: quy trình kiểm thử phần mềm căn bản**

**Bài 4: lập kế hoạch và kiểm soát hoạt động kiểm thử**

**Quy trình kiểm thử căn bản**

* Một quy trình kiểm thử căn bản được định nghĩa gồm 5 hoạt động chính:
  + Lên kế hoạch và kiểm soát hoạt động kiểm thử sản phẩm, hệ thống phần mềm – Test planning & control
  + Phân tích và thiết kế các tình huống kiểm thử - Test analyst & design
  + Viết các tình huống kiểm thử, chuẩn bị dữ liệu kiểm thử (implementation) và chạy chương trình để kiểm thử - test implementation & execution
  + Đánh giá tiêu chí hoàn thành và báo cáo kết quả kiểm thử - Evaluating exit & reporting
  + Tổng kết quá trình kiểm thử - Test closure activities

**Lên kế hoạch kiểm thử sản phẩm**

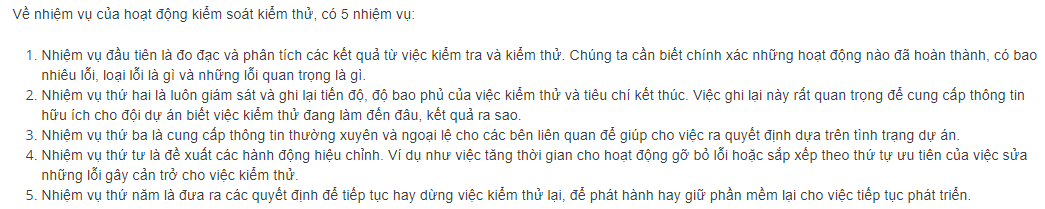
* Bản kế hoạch kiểm thử là một tài liệu mô tả về phạm vi, cách tiếp cận, nguồn lực và kế hoạch thực hiện các hoạt động kiểm thử cho một dự án, sản phẩm phần mềm .
  + **Kế hoạch kiểm thử** (test plan) là một sản phẩm của **quá trình lập kế hoạch kiểm thử** (testing)
  + Việc lập kế hoạch kiểm thử là một hoạt động xây dựng và cập nhật kế hoạch kiểm thử
* Mục tiêu của hoạt động lập kế hoạch kiểm thử:
  + Thiết lập mục tiêu dài và ngắn hạn của việc kiểm thử, cần đạt được cần hiểu các đối tượng chính tham gia vào quá trình sản xuất phần mềm:
    - Khách hàng
    - Các stakeholder
    - Mục tiêu dự án
  + Nhận biết được các rủi ro
  + Xác định được cách tiếp cận và kế hoạch cho việc kiểm thử
* Nhiệm vụ chính trong hoạt động lập kế hoạch:
  + Xác định được phạm vi (scope) kiểm thử
  + Xác định được cách tiếp cận với việc kiểm thử trong dự án
    - Các kỹ thuật kiểm thử
    - Đối tượng kiểm thử
    - Độ bao phủ
    - Người tham gia kiểm thử
    - Các tài liệu dự án, đặc tả phần mềm
  + Thực thi theo chính sách và chiến lược kiểm thử
  + Xác định được nguồn lực cần thiết
    - Con người
    - Môi trường
  + Lên được kế hoạch cho các hoạt động phân tích, thiết kế. thực thi việc kiểm thử
  + Xác định tiêu chí **kết thúc** việc kiểm thử
* 2 điểm quan trọng nhất
  + Tìm được cách tiếp cận tốt nhất cho hoạt động kiểm thử
  + Biết được khi nào việc kiểm thử kết thúc

**Kiểm soát các hoạt động kiểm thử**

Trong mỗi hoạt động, việc quản lý không chỉ kết thúc khi chúng ta lên kế hoạch mà còn kiểm soát nó trong quá trình vận hành xem có đúng với kế hoạch không

Kiểm soát kiểm thử cũng là một hoạt động quản lý và nó diễn ra liên tục

* 2 mục tiêu trong hoạt động kiểm soát
  + Giám sát được tình trạng hiện tại của việc kiểm thử
  + Đưa ra hành động kịp thời kiểm soát được và đạt được mục tiêu chung của dự án



**1 D**

**2 B**

**3 D**

**4 C**

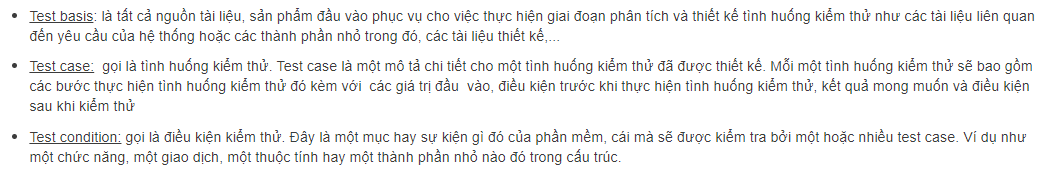
**5 A**

**--------------- --------------- --------------- --------------- --------------- ---------------**

**Bài 5: Phân tích yêu cầu và thiết kế các tình huống kiểm thử**

**Phân tích yêu cầu thiết kế các tình huống kiểm thử**

* Là hoạt động thứ 2 trong quy trình kiểm thử và là hoạt động quan trọng để thực hiện kiểm thử cho hệ thống, sản phẩm phần mềm
* Hoạt dộng này yêu cầu phân tích cụ thể yêu cầu đặc tả của các chức năng trong sản phẩm, từ đó thiết kế các tình huống kiểm thử cho chức năng đó
  + Giai đoạn này rất quan trọng, đảm bảo các mục tiêu đã đề ra sẽ được đưa vào các tình huống kiểm thử trong các quá trình thực thi sau này
  + Mục tiêu: xây dựng được các bộ khung các tình huống cần kiểm thử (test design) dựa trên yêu cầu của khách hàng (test basis), mục tiêu (test obj) và các kỹ thuật để thiết kế tình huống kiểm thử (test techniques)
* Các hoạt động chính trong giai đoạn phân tích và thiết kế tình huống kiểm thử:
  + Ktra lại các tài liệu dự án, bao gồm SRS, design, thiết kế kiến trúc … để hiểu rõ hơn về phần mềm, các sơ hở, không rõ ràng trong bản yêu cầu đặc tả
  + Phân tích, đánh giá khả năng kiểm thử của phần mềm dựa trên yêu cầu của khách hàng
  + Xác định thứ tự ưu tiên cho các điều kiện kiểm thử dựa trên kết quả phân tích các chức năng cần kiểm thử
  + Thiết kế và đặt ưu tiên cho các tình huống kiểm thử mức cao
  + Xác định được các dữ liệu kiểm thử (test data) cho các test conditions và các test cases
  + Thiết kế cho việc thiết lập môi trường kiểm thử, yêu cầu về cơ sở hạ tầng, các công cụ kèm theo
  + Tạo được mối liên hệ giữa test basis và test cases để kiểm soát được việc kiểm thử và sự thay đổi của requirements sau này (nếu có)

****

**1 B**

**2 A**

**3 C**

**4 A**

**5 B**

**--------------- --------------- --------------- --------------- --------------- ---------------**

**Bài 6: Chuẩn bị và thực hiện kiểm thử**

**Chuẩn bị cho việc kiểm thử**

* Là một quá trình:
  + Phát triển và đặt thứ tự ưu tiên cho các thủ tục kiểm thử, bao gồm các tình huống kiểm thử (test cases) hoặc các bộ tình huống kiểm thử (test suite)
  + Tạo dữ liệu kiểm thử (test data)
  + Chuẩn bị các dụng cụ kiểm thử nếu có (test harness)
  + Viết kịch bản kiểm thử tự động (test script) nếu thực hiện kiểm thử tự động
  + Chuẩn bị MTKT (test environment)
* Mục tiêu:
  + Xây dựng được các thủ tục kiểm thử (test procedure)
  + Cài đặt môi trường, tạo được dữ liệu kiểm thử
* Các nhiệm vụ chính trong thi hành / chuẩn bị kiểm thử:
  + Phát triển và xét thứ tự ưu tiên cho các tình huống kiểm thử, dữ liệu và các công cụ kiểm thử cũng như viết test script nếu có kiểm thử tự động
  + Xây dựng được các bộ kiểm thử từ các tình huống kiểm thử
  + Cài đặt, kiểm tra lại xem môi trường kiểm thử có đúng chưa, có thể chạy kiểm thử cho các trường hợp đặc biệt không

**Thực hiện kiểm thử (test execution)**

* Là quy trình chạy thử một thành phần chức năng hay cả hệ thống dựa trên các tài liệu kiểm thử đã chuẩn bị ở hành động chuẩn bị cho kiêm thử (test implementation) để có kết quả thực tế
* **Mục tiêu:** thực hiện kiểm thử và báo cáo sự bất lường (incident / bug) nếu kết quả thực tế khác mong đợi
* 5 nhiệm vụ chính khi thực hiện kiểm thử:
  + Thực hiện kiểm thử theo một suite hoặc các test cases riêng lẻ với thứ tự đề ra trong thủ tục kiểm thử
  + Ghi lại kết quả của việc kiểm thử, các trạng thái và phiên bản của phần mềm đang kiểm thử, các công cụ kiểm thử (test tool) và test ware
  + Báo cáo sự cố nếu có và phân tích để xác định nguyên nhân
  + Lặp lại việc testing (re-testing) cho mỗi sự cố tìm được, nhằm chắc chắn rằng chúng đã được loại bỏ, và cần đảm bảo lỗi xảy ra sẽ không ảnh hưởng đến các vùng khác của phần mềm (regression testing).

**1 C - SAI / B – SAI / A**

**2 D – SAI / A**

**3 A**

**4 B – SAI / C – SAI / D – SAI / A**

**5 B – SAI / A**

**--------------- --------------- --------------- --------------- --------------- ---------------**

**Bài 7: Đánh giá và đóng các hoạt động kiểm thử (Evaluating exit criteria)**

**Đánh giá các hoạt động kiểm thử**

* Là hoạt động t4 trong quy trình kiểm thử: đánh giá kết quả của việc kiểm thử so với mục tiêu đề ra
  + Là hoạt động để biết được việc kiểm thử như thế nào là đủ cho môi giai đoạn kiểm thử
  + Xác định được hoạt động kiểm thử trong giai đoạn có thể kết thúc để chuyển sang giai đoạn tiếp theo hay chưa
* Nhiệm vụ chính:
  + Kiểm tra các kết quả kiểm thử so với các tiêu chí dừng, tiêu chí thoát khỏi giai đaonj kiểm thử đã được xác định và mô tả trong quá trình lập kế hoạch
  + Đánh giá tình trạng hiện tại để nếu cần sẽ phải xác định lại các tiêu chí để dừng
  + Viết báo cáo tổng kết các hoạt động kiểm thử đã thực hiện trong giai đoạn kiểm thử để gửi các bên liên quan

**Đóng các hoạt động kiểm thử (test closure)**

* Là hoạt động thu thập dữ liệu từ các hoạt động kiểm thử, tổng hợp từ kinh nghiệm dựa trên việc ktra và hoàn thiện bộ sản phẩm kiểm thử. Dữ liệu sau khi thu thập sẽ là căn cứ, cơ sở để phân tích số liệu đưa ra các bài học để áp dụng cho tương lai. Đây là hoạt động cuối trong quy trình kiểm thử
* Mục tiêu của Test Closure này là thu thập dữ liệu để:
  + Cung cấp dữ liệu cho bàn giao sản phẩm phần mềm
  + Lưu trử dữ liệu
  + Phân tích cho hoạt động cải tiến sau này
* Các nhiệm vụ chính (7 nhiệm vụ):
  + Kiểm tra sản phẩm thực tế được bàn giao so với kế hoạch như thế nào, và chắc chắn rằng tất cả lỗi đã được giải quyết hay có kế hoạch giải quyết trong các lần bàn giao sau.
  + Đóng các báo cáo về sự cố lại, hoặc ghi chép các thay đổi cho bất cứ vấn đề nào còn đang mở
  + Viết 1 biên bản chấp nhận phần mềm
  + Lưu trữ các sản phẩm kiểm thử, môi trường kiểm thử và cơ sở hạ tầng để sử dụng lần sau, đặc biệt nếu phải thực hiện kiểm thử bảo trì sản phẩm.
  + Bàn giao các sản phẩm kiểm thử cho bộ phận quản lý dữ liệu, bộ phận bảo trì để tiếp tục hỗ trợ trong việc sửa lỗi nếu có hoặc tiếp tục phát triển theo những yêu cầu mới.
  + Phân tích các bài học để xác định những điểm cần thay đổi cho dự án sau
  + Sử dụng các thông tin thu thập được để cải tiến công việc kiểm thử một cách định kỳ như cải tiến việc thiết kế các test cases, các thực hiện kiểm thử …

1 C

2 D / SAI - C

3 D / SAI - B

4 D

5 C / SAI – D

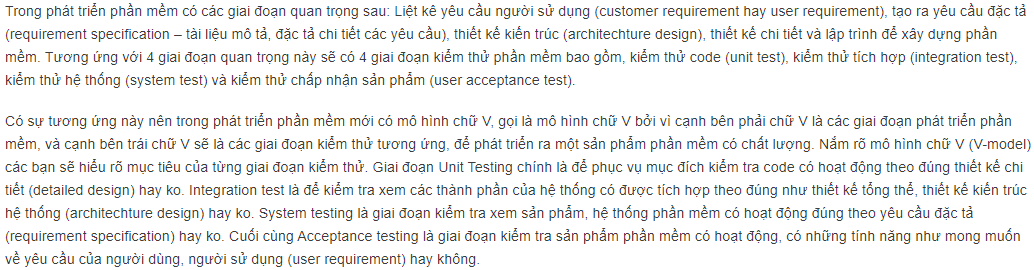
**--------------- --------------- --------------- --------------- --------------- --------------- --------------- --------------- --------------- --------------- --------------- ---------------**

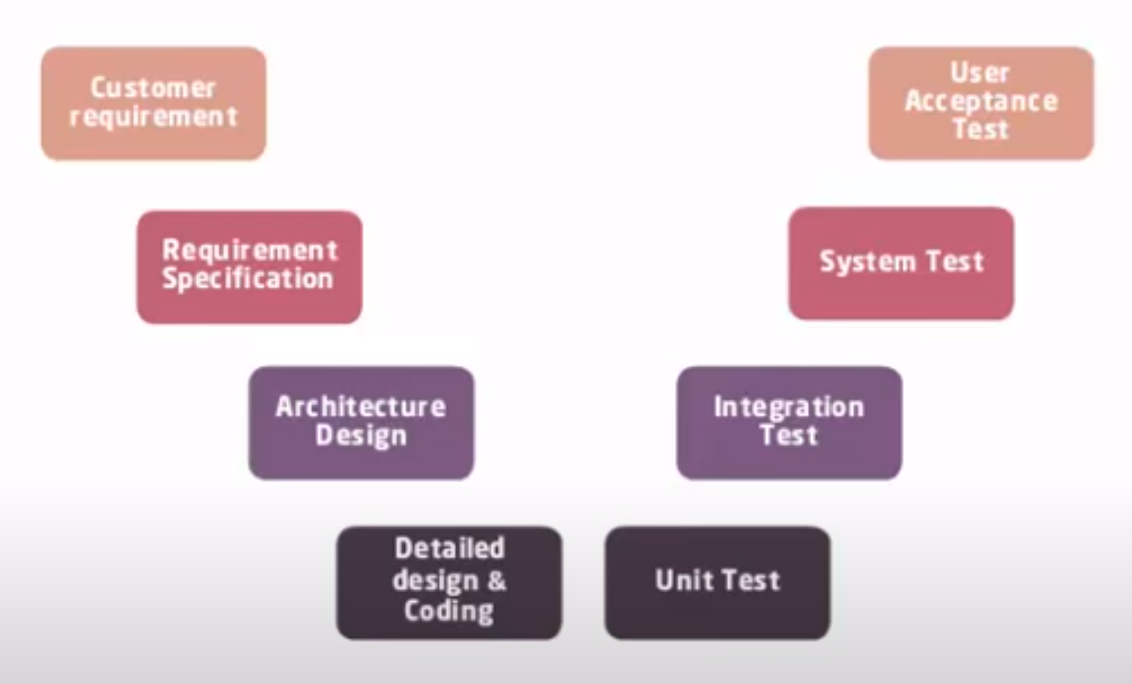
**Phần 3: các mức độ hay các giai đoạn kiểm thử**

**Bài 8: kiểm thử đơn vị**

**Định nghĩa của test levels**

* Là một tập hợp các hoạt động kiểm thử mà được tổ chức, quản lý cùng nhau, phục vụ một mục tiêu kiểm thử nào đó
* Mỗi giai đoạn kiểm thử thường phục vụ cụ thể cho một giai đoạn phát triển phần mềm.
* Các giai đoạn kiểm thử:
  + Unit testing: kiểm thử đơn vị
  + Integration testing: kiểm thử tích hợp
  + System testing: kiểm thử hệ thống
  + Acceptance Testing: kiểm thử chấp nhận (kiểm thử để chấp nhận sản phẩm)

****

****

**Kiểm thử đơn vị (unit test)**

* Là việc kiểm thử từng đơn vị phần mềm xem có hoạt động đúng như thiết kế hay không. Đơn vị cần kiểm thử có thể là:
  + Một dòng lệnh
  + Một hàm
  + Một module
  + Một chương trình
* Mức độ đơn vị to – nhỏ thế nào do người thực hiện làm kiểm thử quyết định, mức độ càng nhỏ thì cơ hội tìm kiếm được lỗi càng lớn
* Các loại kiểm thử đơn vị:
  + Kiểm thử câu lệnh – statement testing: tất cả câu lệnh đều được kiểm thử bằng cách chạy qua ít nhất 1 lần, do đó cần chọn data hợp lý để có thể thực hiện các dòng lệnh viết ra
  + Kiểm thử quyết định hay nhánh (decision or branch testing): tất cả các nhánh đều được chạy qua ít nhất 1 lần
  + Kiểm thử điều kiện – condition testing: kiểm tra sự kết hợp của các điều kiện
  + Kiểm thử đường đi – path testing: kiểm tra tất cả các đường đi từ bắt đầu đến kết thuchs trong 1 hàm ít nhất 1 lần
* Người kiểm thử: lập trình viên, người tạo ra những dòng code
* Sử dụng một hoặc nhiều loại kiểm thử đơn vị để thực hiện việc kiểm thử, cách thức thực hiện có thể như sau:
  + Từ trên xuống, sử dụng các **stubs:** là các đơn vị chưa có mà được gọi bởi 1 đơn vị đã hoàn thành,
  + Từ dưới lên, sử dụng các **drivers:** là các đơn vị chưa có gọi đến các đơn vị đã hoàn thành

**1 A**

**2 B**

**3 A**

**4 A**

**5 C**

**--------------- --------------- --------------- --------------- --------------- ---------------**

**Bài 9: kiểm thử tích hợp**