

Lý thuyết Kiểm Tra Phần Mềm

Bài 02:

Vai trò Kiểm tra trong vòng đời phần mềm

GV: Nguyễn Ngọc Tú

Email: nntu@hoasen.edu.vn

Bộ môn: Kỹ thuật Phần mềm

Nội dung

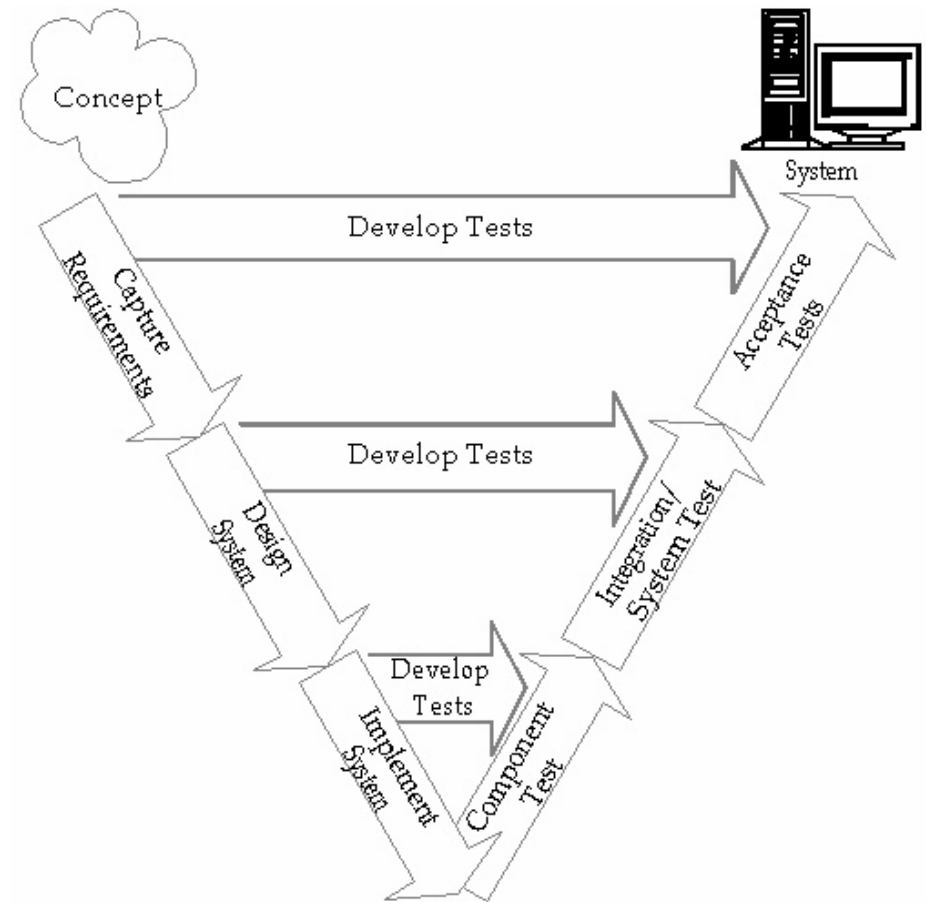
- Những mô hình phát triển phần mềm
 - Mô hình chữ V
 - Mô hình phát triển lặp gia tăng
- Kiểm tra trong mô hình vòng đời phần mềm
- Kiểm chứng và chứng thực (V & V)
- Các mức kiểm tra
- Các kiểu kiểm tra

Những mô hình phát triển PM

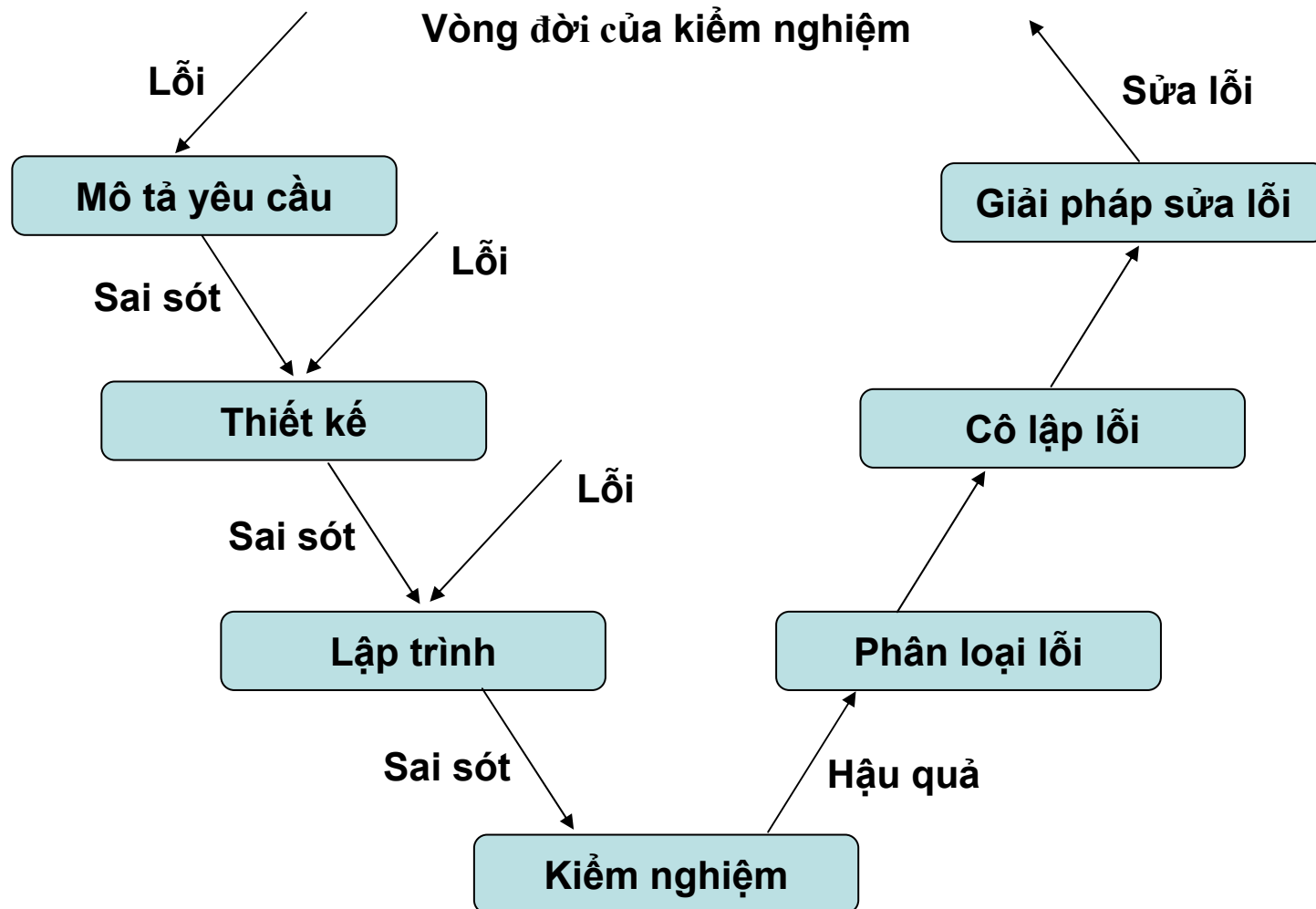
- Quan hệ giữa phát triển và kiểm tra
- Thích hợp giữa mô hình đến ngữ cảnh
- Lý do của các mức kiểm thử khác nhau

Mô hình chữ V

- Thường lập lịch, hướng tới rủi ro ngân sách
- Thực hiện các mức sâu hơn, xây dựng và kiểm tra
- Mô hình quen thuộc
- Bất ổn
- △ Khó lập kế hoạch xa
- △ Khi đổ vỡ kế hoạch, khâu kiểm thử hứng chịu cuối cùng

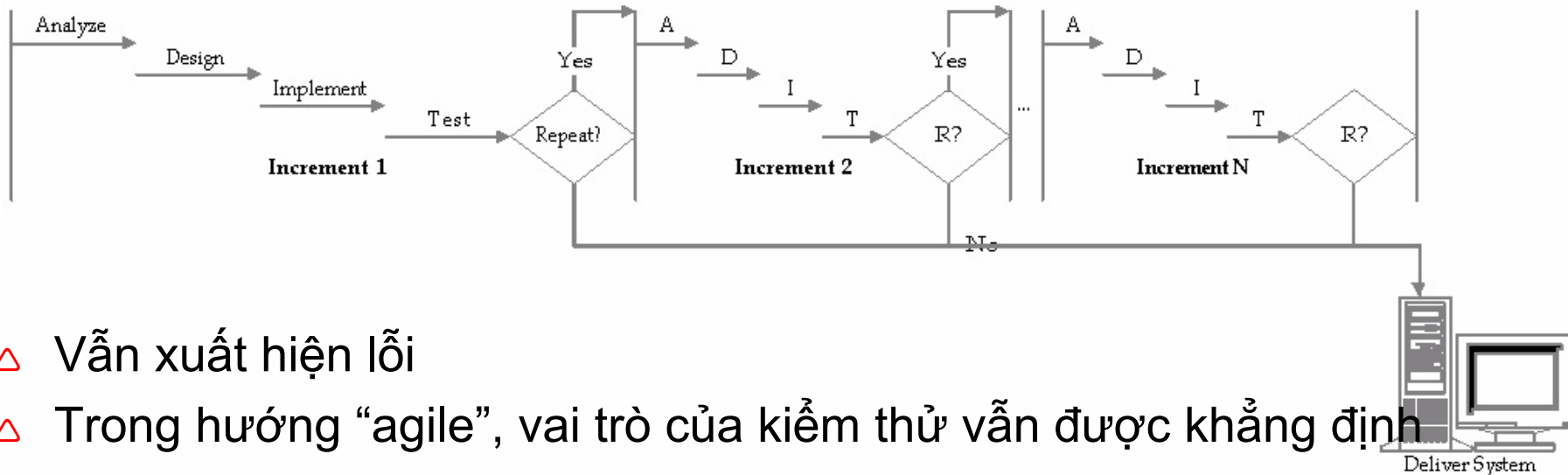


Mô hình chữ V



Mô hình phát triển lặp gia tăng

- Hướng tới rủi ro kế hoạch: tiếp cận thị trường hoặc theo hạn định
- Tập các đặc tính phát triển từ chức năng cốt lõi
- Có thể chuyển sử dụng một khi “khung” đã xong
- Trở thành cách tiếp cận phổ biến
- Extreme Programming , RAD , RUP



- △ Vẫn xuất hiện lỗi
- △ Trong hướng “agile”, vai trò của kiểm thử vẫn được khẳng định

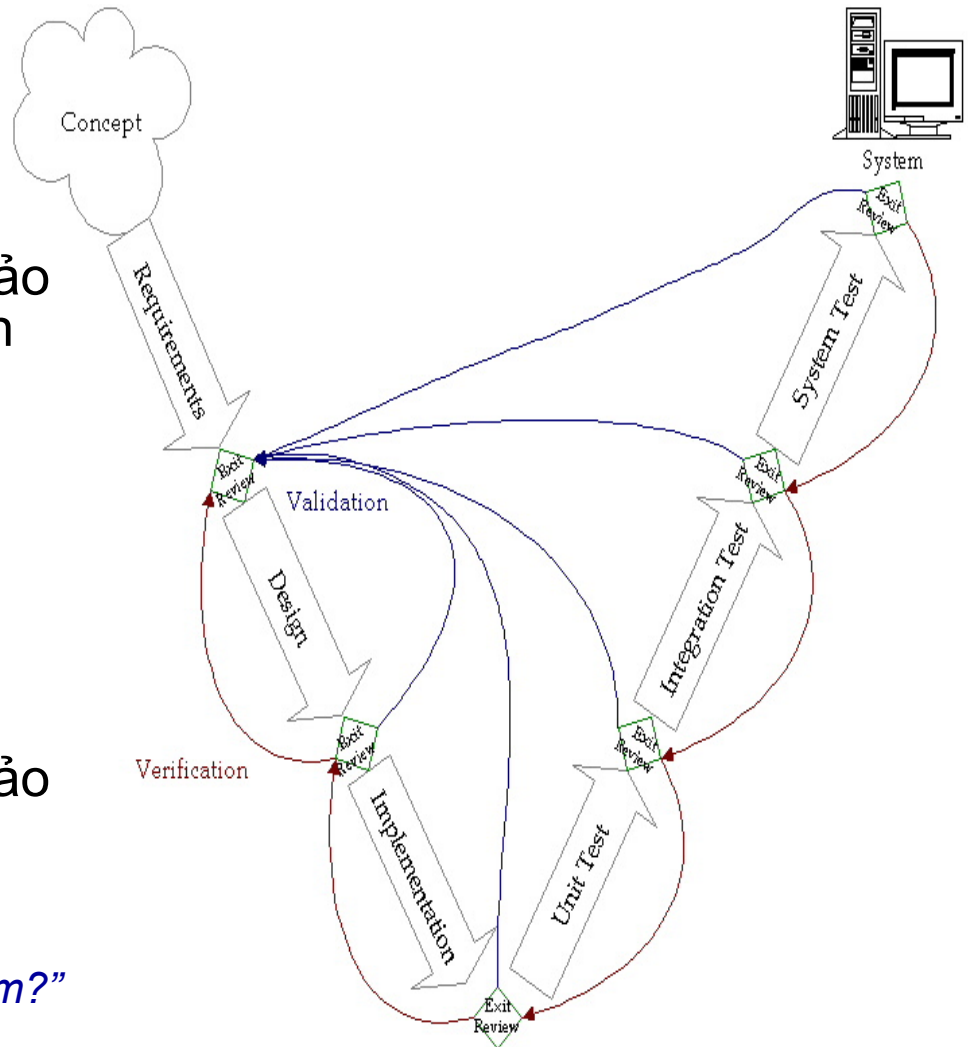
Kiểm tra trong mô hình vòng đời phần mềm

- Đặc tính chung của một kiểm thử tốt
 - Kiểm thử cho mỗi giai đoạn/phần phát triển
 - Các mức kiểm tra nhấn mạnh vào mục tiêu phối hợp liên tục, không trùng lặp
 - Phân tích, thiết kế bắt đầu sớm, ngăn ngừa lỗi

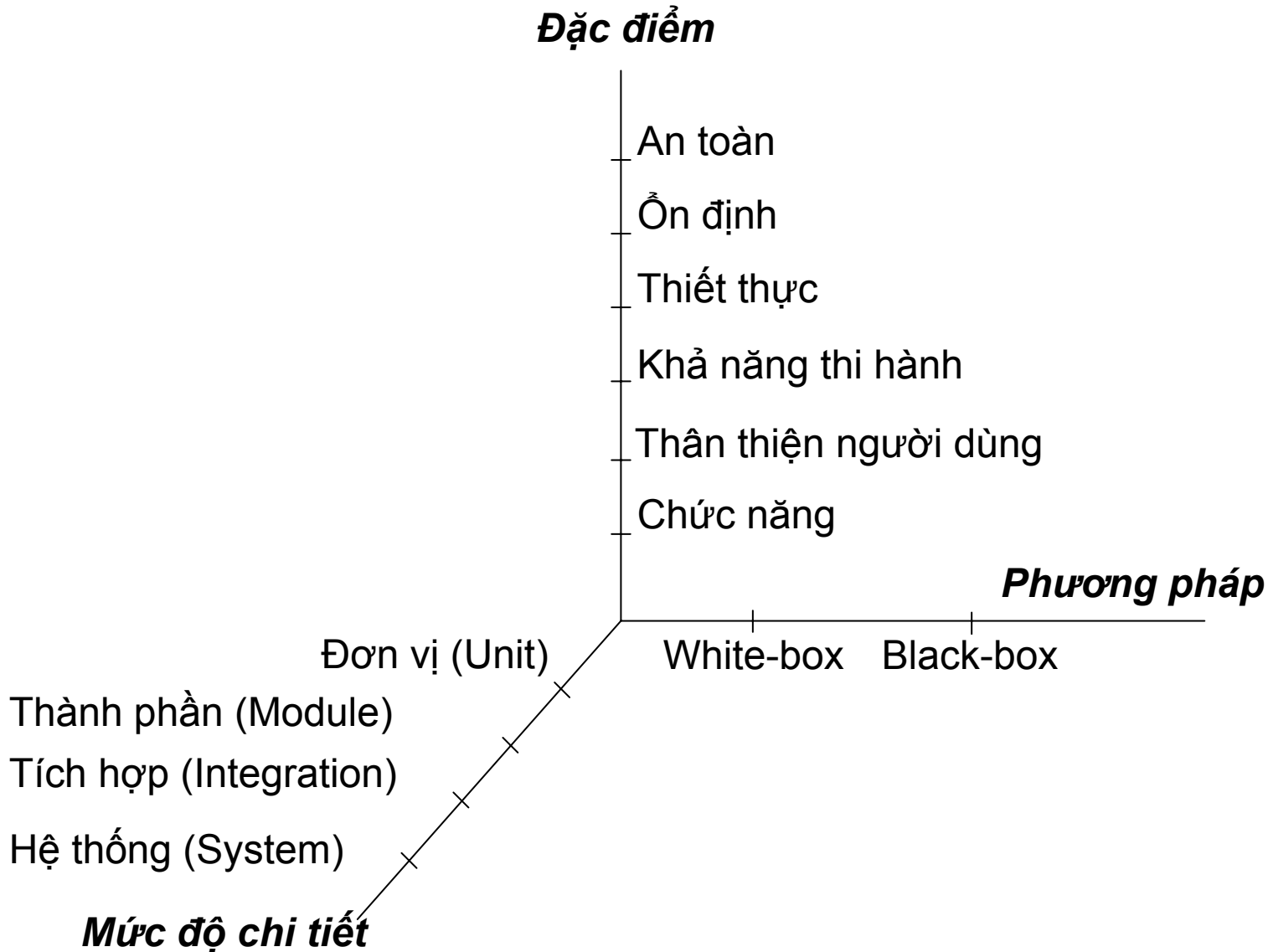
Kiểm chứng và chứng thực

Verification & Validation

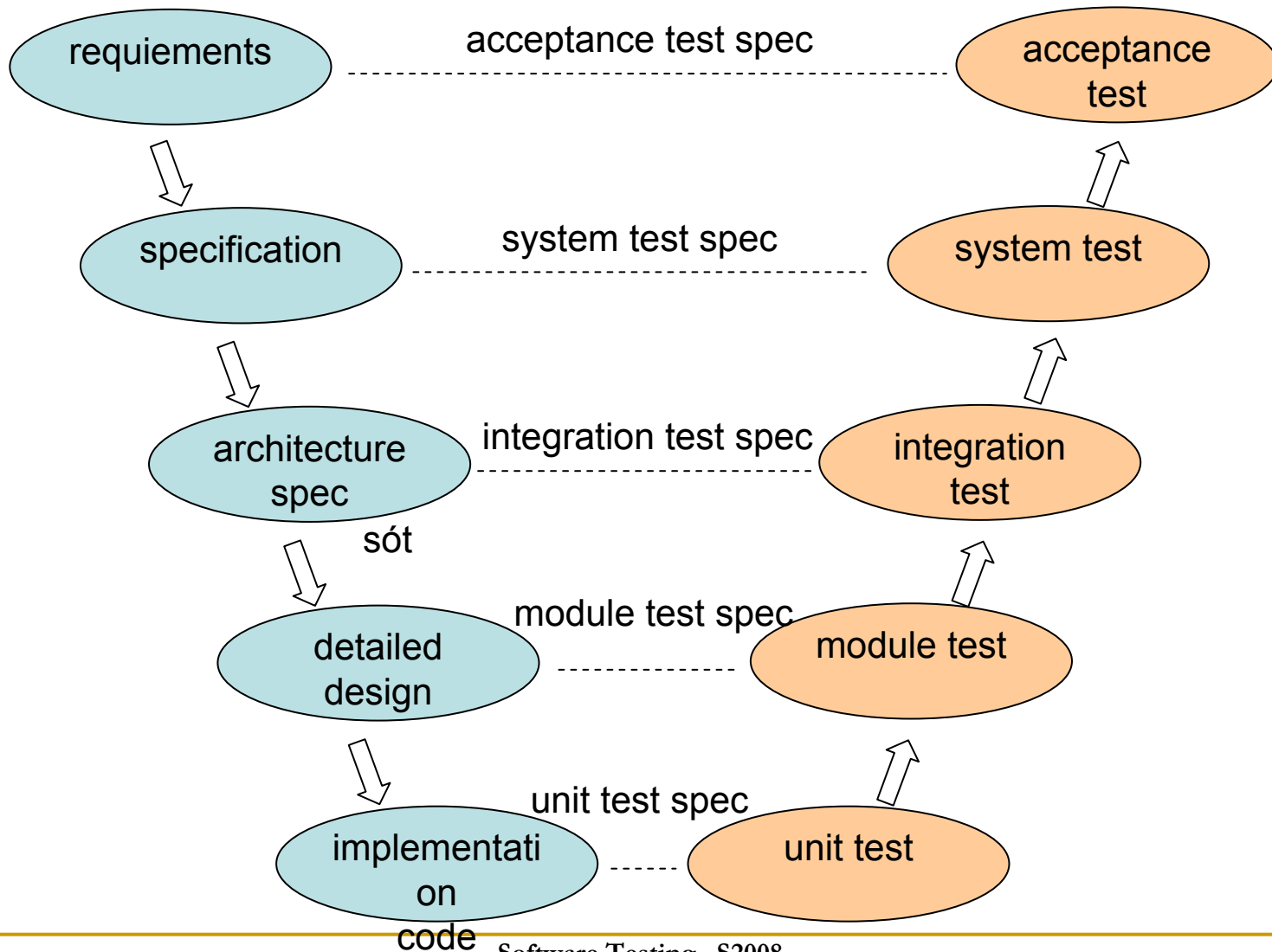
- Kiểm chứng
 - Tìm các lỗi trong từng giai đoạn
 - các hành động để đảm bảo cho phần mềm được hiện thực đúng theo một chức năng cụ thể nào đó
 - *“Are we building the system right?”*
- Chứng thực
 - Tìm lỗi trong hệ thống, phần chuyển giao
 - các hành động để đảm bảo cho phần mềm được xây dựng theo đúng yêu cầu của khách hàng
 - *“Are we building the right system?”*



Các mức kiểm tra



Các mức kiểm tra



Các mức kiểm tra

- Kiểm tra thành phần
- Kiểm tra tích hợp
- Kiểm tra hệ thống
- Kiểm tra chấp nhận

Kiểm tra thành phần/đơn vị

Component (Unit) Test

- Mục tiêu:
 - Tìm lỗi trong từng phần đơn lẻ
- Cơ sở:
 - mã, csdl, yêu cầu, thiết kế, các rủi ro chất lượng
- Kiểu kiểm tra:
 - Chức năng, sử dụng tài nguyên, hiệu suất, cấu trúc
- Công cụ:
 - Mức API (drivers , stubs), ...
- Người thực hiện:
 - Người lập trình

Kiểm tra thành phần/đơn vị

Component (Unit) Test

- Tiêu biểu
 - Can thiệp mã
 - Thực thi trong môi trường độc lập
 - Yêu cầu drivers, stubs
 - Được thực hiện bởi người viết chương trình
- Thường được sửa trực tiếp, không lập báo cáo
- Phát triển theo hướng kiểm thử
 - Phát triển tập kiểm thử đơn vị
 - Xây dựng và tích hợp mã
 - Thực thi

Kiểm tra tích hợp

- Mục tiêu:
 - Tìm lỗi trong mối quan hệ và giao diện giữa các cặp, nhóm thành phần liên quan
- Cơ sở:
 - Thiết kế, kiến trúc, giản đồ, lưu đồ. Rủi ro chất lượng
- Kiểu kiểm tra:
 - Chức năng, tài nguyên, hiệu năng
- Công cụ:
 - Mức API , CLI , ..
- Người thực hiện:
 - Cả người kiểm thử và người lập trình

Kiểm tra tích hợp

- Big bang
- Bottom up
- Top down
- Backbone

Kiểm tra tích hợp

- Nhiều hơn 1 mức kiểm tra tích hợp trong dự án
 - Tích hợp thành phần:
 - Tìm lỗi tương tác các thành phần
 - Tích hợp hệ thống:
 - Tìm lỗi tương tác trên toàn hệ thống
- Phức tạp
 - Nhiều tổ chức
 - Tiến trình nghiệp vụ
 - Độ tương thích Hardware/system

Kiểm tra hệ thống

- Mục tiêu:
 - Tìm lỗi trên toàn bộ và cá biệt về hành vi, chức năng, đáp ứng của hệ thống
- Cơ sở:
 - Yêu cầu, thiết kế mức cao, use cases, rủi ro, kinh nghiệm, môi trường, checklists
- Kiểu kiểm tra:
 - Chức năng, bảo mật, hiệu năng, tin cậy, khả dụng, khả chuyển, ...
- Công cụ:
 - API, CLI, GUI, ...
- Người thực hiện:
 - Người kiểm tra độc lập

Kiểm tra chấp nhận

Acceptance Test

- Mục tiêu:
 - Chạy thử sản phẩm sẵn sàng cho triển khai/xuất xưởng
- Cơ sở:
 - Yêu cầu, hợp đồng, kinh nghiệm
- Kiểu kiểm tra:
 - Chức năng, khả chuyển, hiệu năng
- Công cụ:
 - GUI
- Người thực hiện:
 - Thường là Khách hàng/người sử dụng
 - Người kiểm tra độc lập

Kiểm tra chấp nhận

Acceptance Test

- Kiểm tra chấp nhận với NSD:
 - Người sử dụng nghiệp vụ xác nhận mức phù hợp cho mục cho chức năng,
- Kiểm tra tác vụ (*Operational testing*):
 - Chấp nhận bởi người quản trị
- Hợp đồng và kiểm tra quy tắc (*regulation testing*):
 - Kiểm chứng xác nhận hợp lệ theo hợp đồng.
- Kiểm tra Alpha, Beta, và lĩnh vực (*field testing*):
 - Kiểm thử và xây dựng tin cậy bởi các khách hàng đã có hay tiềm năng
 - Kiểm thử Beta và lĩnh vực được thực hiện trong môi trường thực

Các kiểu kiểm tra

- Chức năng – phi chức năng
- Cấu trúc
- Xác nhận – hồi quy

Kiểm tra chức năng

- Các hoạt động hợp lý, đã yêu cầu không được cung cấp, truy xuất, hư hỏng trầm trọng.
 - *Không thêm chức năng trong bảng tính*
 - *Hiện thực , không làm việc*
 - *Có thể chỉ số nguyên, không số thực*
- Hành động đúng, kết quả sai
 - Hàm ADD: $2+2=5$?
- Hàm đúng, kết quả đúng .. Sai thể hiện
 - Hàm DIV: $2/2=1$

Kiểm tra chức năng – phi chức năng

- Bản địa hóa
(*Localization : user interface, operational*)
- Chuẩn và đúng nguyên tắc (*regulatory compliance*)
- Xử lý lỗi và phục hồi
- Phục hồi tai họa
(*Disaster recovery*)
- Thực hiện và phân bố trên mạng
- Thời gian , sắp xếp
- Chất lượng dữ liệu
- Hoán chuyển dữ liệu
- Tác vụ
- Cài đặt
- Phân giải cài đặt
- Xử lý ngày tháng
- Tài liệu
- ...

Kiểm tra cấu trúc

Structural Tests

- Trên cơ sở hệ thống được xây dựng như thế nào ?
 - Mã
 - Dữ liệu
 - Thiết kế
- Bao phủ Cấu trúc (*white box*) có thể được đánh giá chức năng và phi chức năng non-functional (*black box*)

Kiểm tra hồi quy và xác nhận

Regression and Confirmation

- Kiểm tra hồi quy (*Regression testing*)
 - Kiểm tra ảnh hưởng của thay đổi dù cho nó có nhỏ, cục bộ, biệt lập
- Kiểm tra xác nhận (*Confirmation testing*) xác nhận
 - Thay đổi hệ thống đã có
 - Lỗi đã cố định phải hỗ trợ theo dõi
- Luôn lặp lại kiểm định
- Tự động, kiểm tra sâu, có ích

Bài tập - đọc hiểu

Đọc thêm

- [2]. Chapter 18
- [4]. Chapter 12
- [5]. Chapter 02
- *[3]. Chapter 03,04*

Q/A