

# Test levels – Test methods

Công cụ kiểm chứng phần mềm

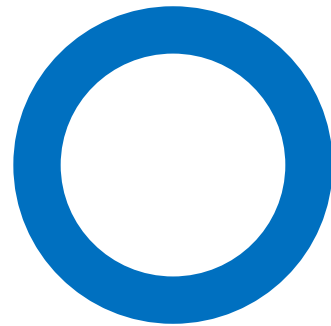
Hồ Tuấn Thanh

[htthanh@fit.hcmus.edu.vn](mailto:htthanh@fit.hcmus.edu.vn)

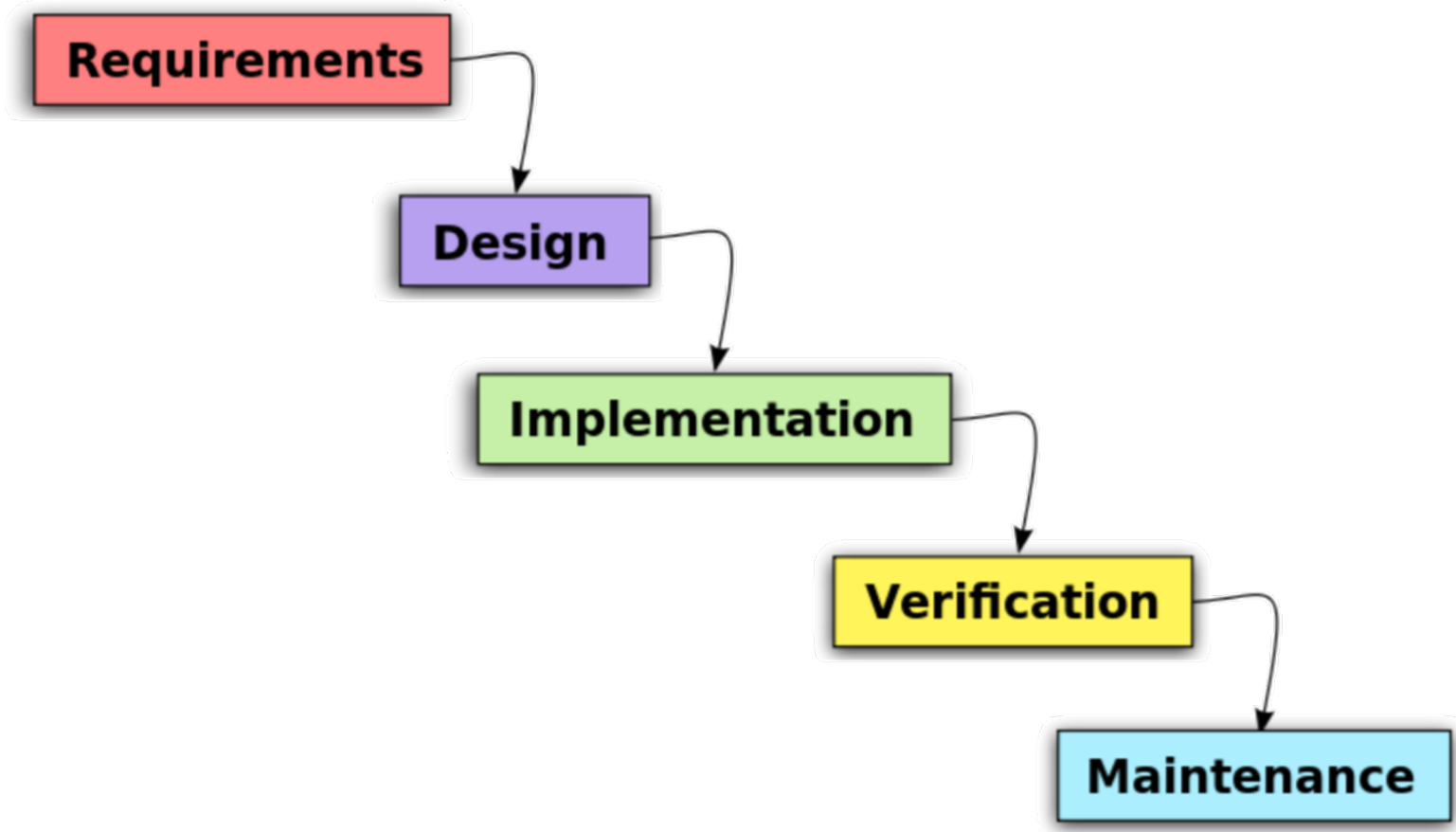
# Nội dung

- ☐ Life cycle
- ☐ Test level
- ☐ Test technique

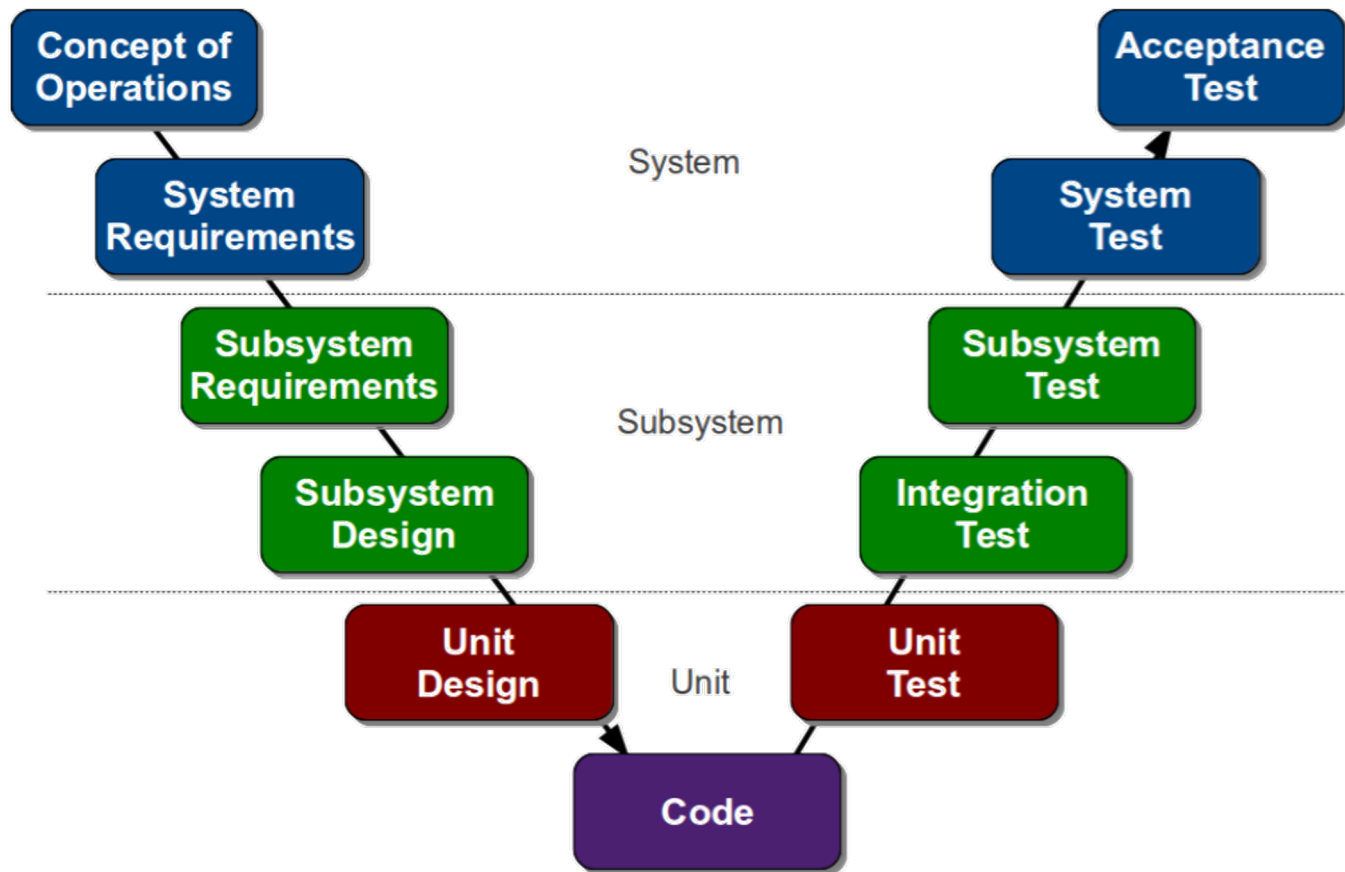
# LIFE CYCLE



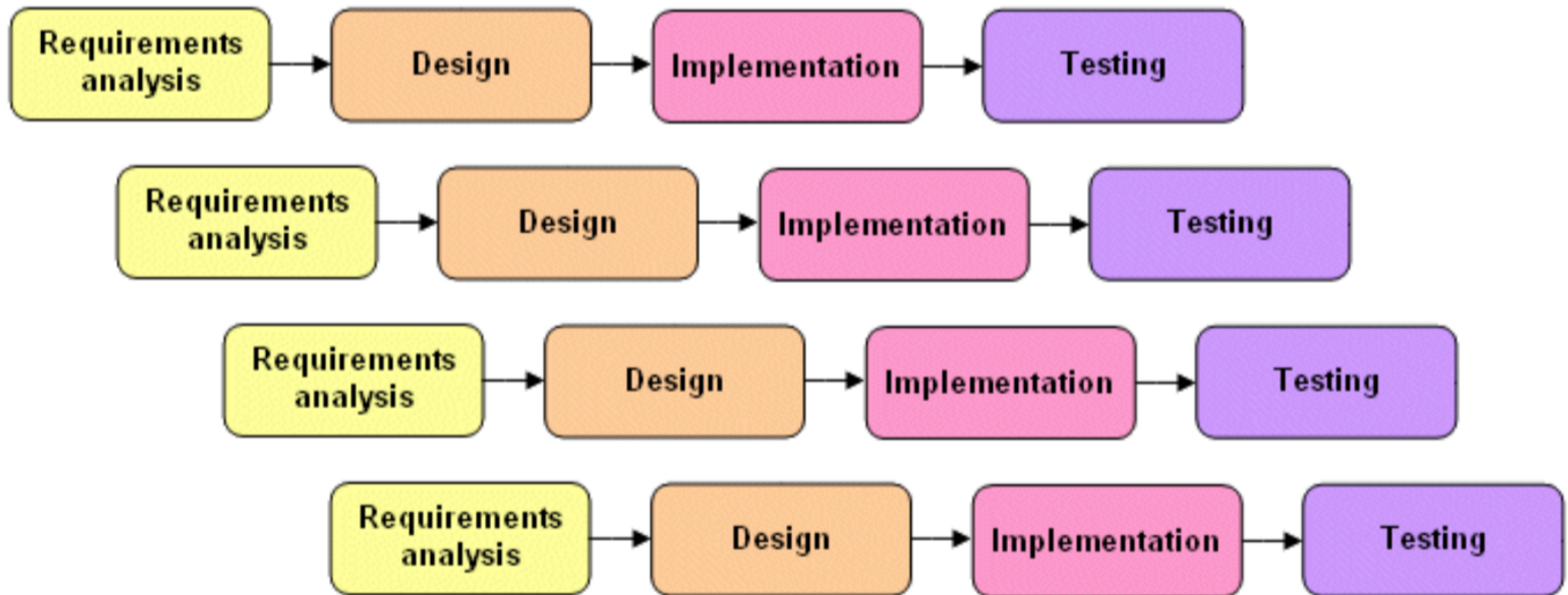
# Waterfall model (Mô hình thác nước)



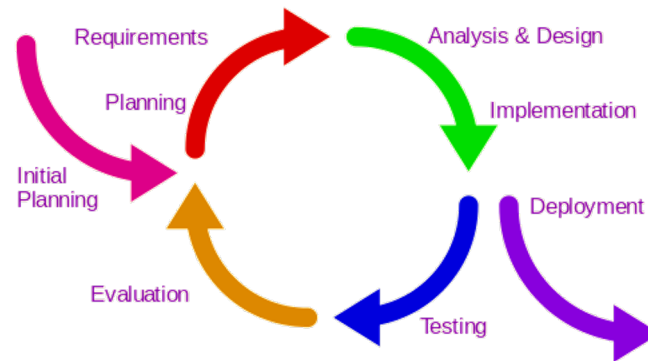
# V-model



# Incremental model (Mô hình tăng dần)



# Iterative model (Mô hình lặp)



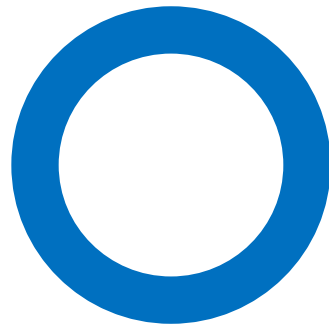
# Đặc tính chung của kiểm thử tốt

- ☐ Kiểm thử cho mỗi giai đoạn, mỗi thành phần phát triển
- ☐ Các mức kiểm tra phối hợp liên tục, tránh trùng lặp
- ☐ Phân tích, thiết kế bắt đầu sớm, ngăn ngừa lỗi

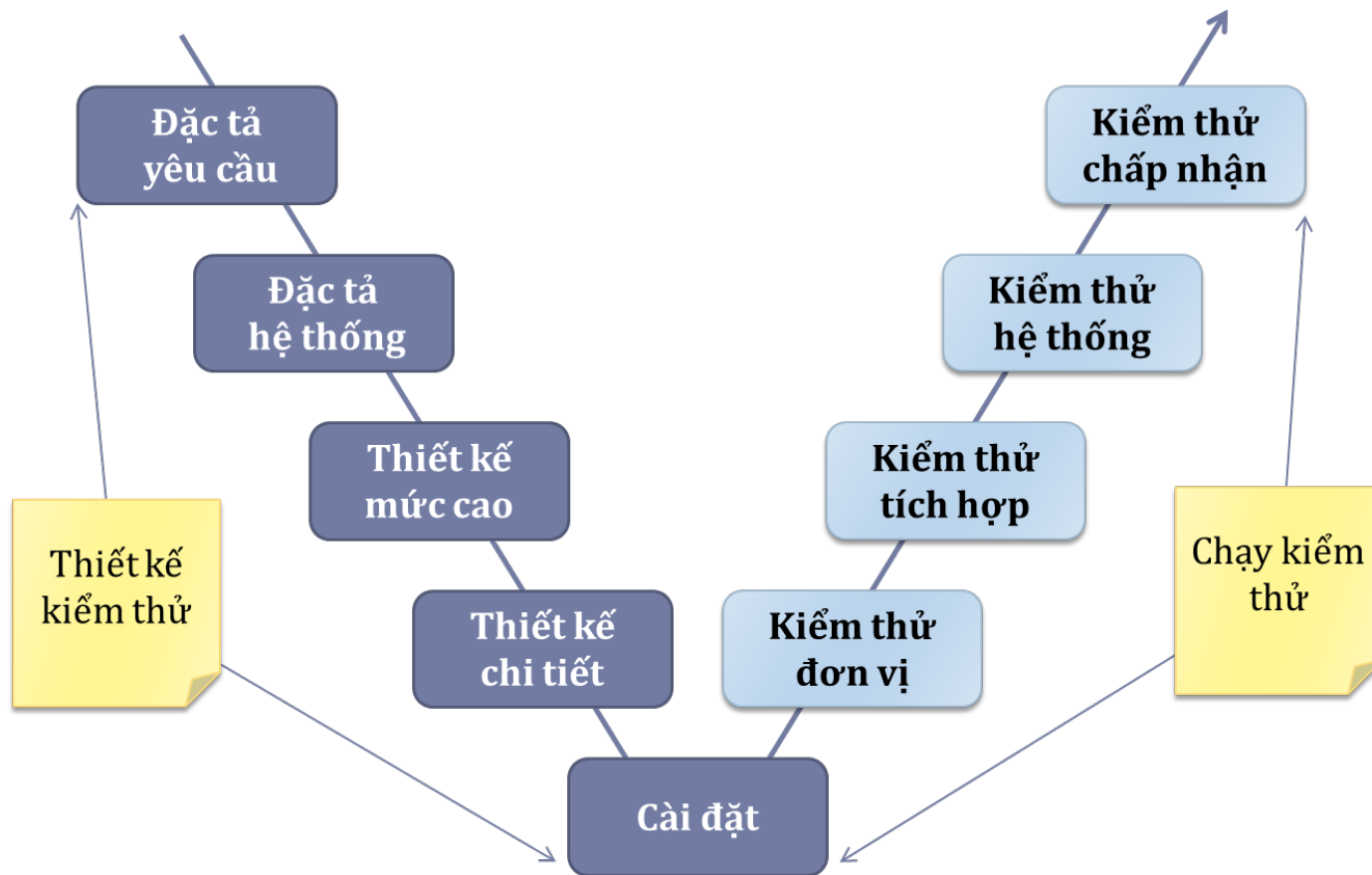




# TEST LEVEL



# V-model



# Kiểm thử đơn vị - Unit testing

- ❑ Người thực hiện
  - Lập trình viên (developer)
- ❑ Báo cáo
  - Lỗi được sửa ngay, ko cần báo cáo
- ❑ Công cụ
  - Viết trực tiếp mã nguồn
  - Unit test framework
  - Mocking framework
  - Dependency Injection & IoC containers

# Unit testing

- ❑ Test driven development
  - Test-first approach
  - Hướng tiếp cận phát triển phần mềm dựa trên unit test
  - Chuẩn bị và tự động hóa test case trước khi coding
  - Lập trình từng phần một → đảm bảo tất cả test case đều đạt

# Kiểm thử tích hợp – Integration testing

- ❑ Kiểm tra hơn 2 đơn vị/hệ thống
- ❑ Mục tiêu:
  - Kiểm thử sự tương tác giữa các đơn vị/hệ thống với nhau
  - Kiểm thử các tập không hoạt động độc lập
  - Kiểm thử chức năng và phi chức năng
- ❑ Dựa trên:
  - Thiết kế phần mềm
  - Kiến trúc phần mềm
  - Workflows/Use-cases

# Integration testing

## □ Hai cấp độ

- Kiểm thử tích hợp đơn vị
- Kiểm thử tích hợp hệ thống

## □ Người thực hiện

- Người phát triển (developer)
- Người thiết kế (designer)
- Người kiểm thử độc lập (tester)

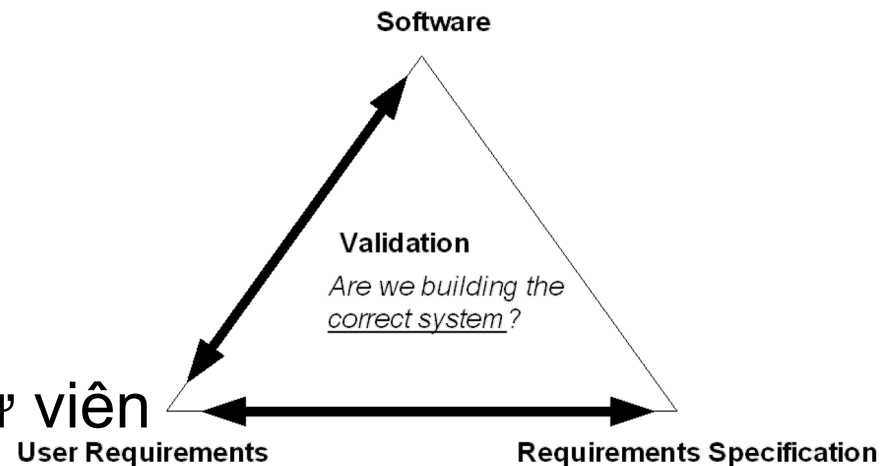
# Kiểm thử hệ thống – System testing

- ❑ Bước cuối cùng của kiểm thử tích hợp
- ❑ Kiểm thử hệ thống như một tổng thể
- ❑ Mục tiêu:
  - Phát hiện sai sót trong toàn bộ hệ thống
  - Kiểm thử chức năng và phi chức năng
- ❑ Dựa trên:
  - Đặc tả yêu cầu phần mềm
  - Use case
  - Tài liệu hướng dẫn sử dụng



# Kiểm thử chấp nhận – Acceptance testing

- ❑ Bước cuối cùng của quá trình validation
- ❑ Mục tiêu:
  - Xác định từ phía người dùng: hệ thống có đáp ứng đúng mong đợi của người dùng ko?
- ❑ Dựa trên:
  - Đặc tả yêu cầu
- ❑ Người thực hiện:
  - Người sử dụng
  - Có thể bao gồm kiểm thử viên



Validation and the Quality Triangle

# Acceptance testing

## □ Alpha testing vs Beta testing

- Giống nhau

  - Khi phần mềm đã ổn định

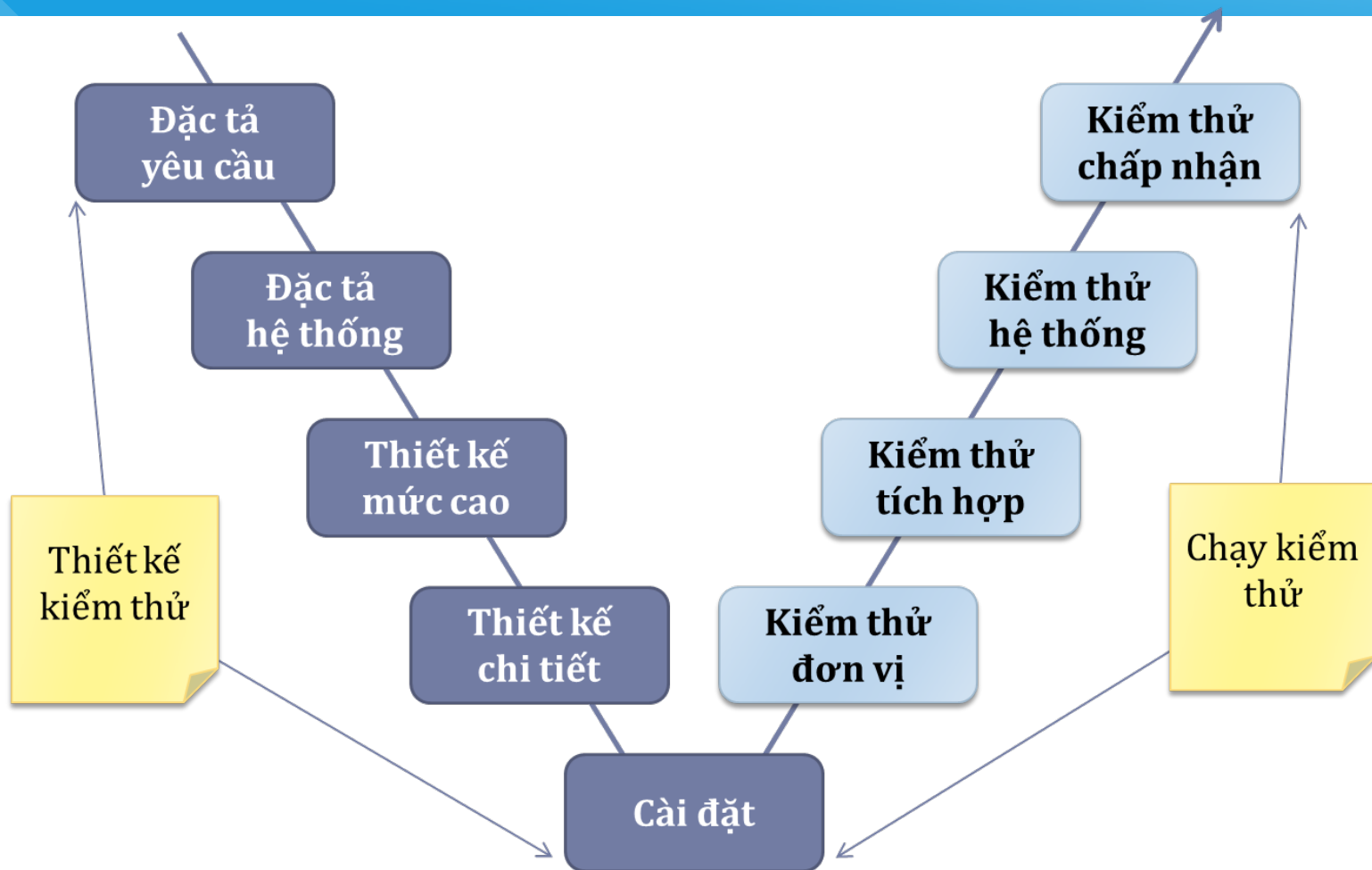
  - Nhận phản hồi về lỗi, mong đợi, đề xuất

- Khác nhau

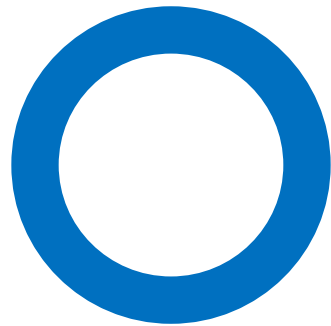
  - Alpha testing thực hiện tại môi trường phát triển

  - Beta testing thực hiện tại môi trường thực tế

# V-model



# TEST TECHNIQUE



# Test technique

1. Kiểm thử chức năng
  - Functional testing/Black-box testing
2. Kiểm thử phi chức năng
  - Non-functional testing
3. Kiểm thử cấu trúc
  - Structural testing/White-box testing
4. Kiểm thử liên quan thay đổi
  - Confirmation testing/Re-testing & Regression testing

# Functional testing vs Non functional testing

FUNCTIONAL TESTING	NON-FUNCTIONAL TESTING
What the system actually does is functional testing	How well the system performs is non-functionality testing
To ensure that your product meets customer and business requirements and doesn't have any major bugs	To ensure that the product stands up to customer expectations
To verify the accuracy of the software against expected output	To verify the behavior of the software at various load conditions
It is performed before non-functional testing	It is performed after functional testing
Example of functional test case is to verify the login functionality	Example of non-functional test case is to check whether the homepage is loading in less than 2 seconds
It can be performed either manual or automated way	It can be performed efficiently if automated

# 1. Kiểm thử chức năng

- ☐ Functional testing/Black-box testing
- ☐ Dựa trên đặc tả chức năng
- ☐ Phát hiện sai sót về chức năng
- ☐ Không quan tâm đến cách cài đặt

# 1. Kiểm thử chức năng

## □ Các kĩ thuật thiết kế

- Phân hoạch tương đương (Equivalence partitioning)
- Phân hoạch giá trị biên (Boundary value analysis)



# 2. Kiểm thử phi chức năng

## ❑ Non-functional testing

1. Kiểm thử hiệu năng – Performance testing
2. Kiểm thử tính tiện dụng – Usability testing
3. Kiểm thử bảo mật – Security testing
4. Kiểm thử cấu hình, cài đặt – Configuration/Installation testing
5. Kiểm thử sao lưu/phục hồi – Backup/Recovery testing

# 2.1. Performance testing

- ❑ Kiểm thử khối lượng – Volume testing
  - Kiểm tra khả năng xử lý dữ liệu lớn của hệ thống
- ❑ Kiểm thử tải/quá tải – Load/Stress testing
  - Kiểm tra yêu cầu về thời gian đáp ứng của hệ thống

## 2.2. Usability testing

- ☐ Dễ học, sử dụng đơn giản
- ☐ Sử dụng hiệu quả
- ☐ Giao diện đơn giản, đồng nhất
- ☐ Hỗ trợ thông tin phản hồi
- ☐ Ngăn ngừa lỗi
- ☐ Liên kết tắt
- ☐ Thông báo lỗi tốt
- ☐ ...

## 2.3. Security testing

- ❑ Kiểm tra tính hợp lệ của việc truy xuất trong và ngoài chương trình

## 2.4. Configuration/Installation testing

- ❑ Kiểm tra cấu hình
  - Phần cứng, môi trường phần mềm khác nhau
  - Cấu hình bản thân phần mềm
- ❑ Kiểm tra cài đặt
  - Gói cài đặt (CD, mạng...)
  - Uninstall

## 2.5. Backup/Restore testing

- ☐ Kiểm tra khả năng sao lưu và khả năng khôi phục hệ thống từ sự cố

# 3. Kiểm thử cấu trúc

- ☐ Structural testing/White-box testing
- ☐ Nghiên cứu mã nguồn
- ☐ Phân tích thứ tự thực hiện các lệnh

# Kiểm thử liên quan thay đổi

- Kiểm tra sau khi lỗi được sửa chữa
- Kiểm tra lại – Re-testing/Confirmation testing
  - Kiểm tra lại chính xác trường hợp kiểm thử đã phát hiện ra lỗi
  - Xác nhận lỗi đã được sửa chữa



# 4. Kiểm thử liên quan thay đổi

- Kiểm thử hồi qui – Regression testing
  - Kiểm tra lại tất cả các trường hợp kiểm thử đã thỏa trước đó
  - Tìm ra lỗi mới phát sinh

