

# Chương 1

## Tổng quan về Kiểm thử phần mềm

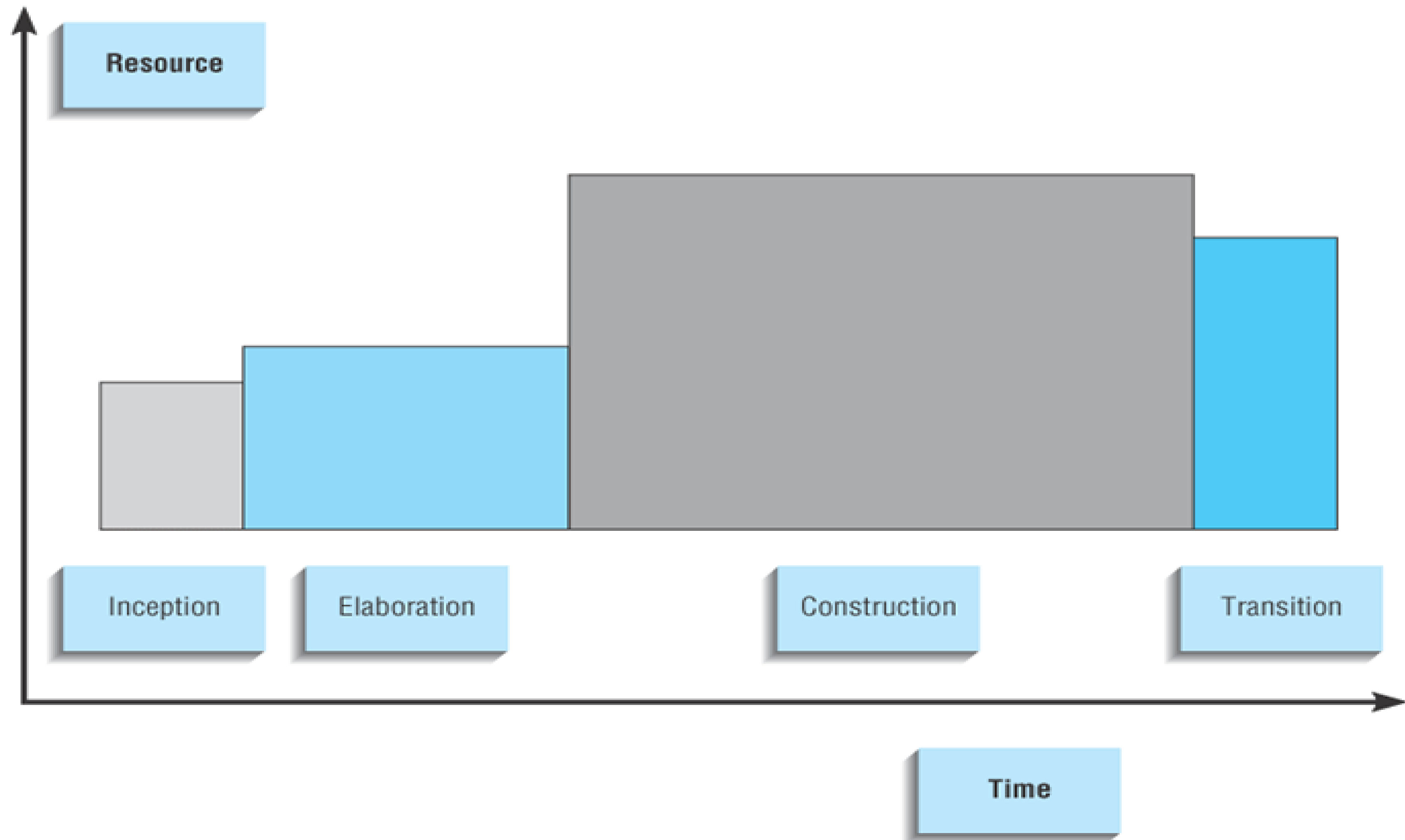
(Buổi 1, 2)

TP.HCM

# Nội dung

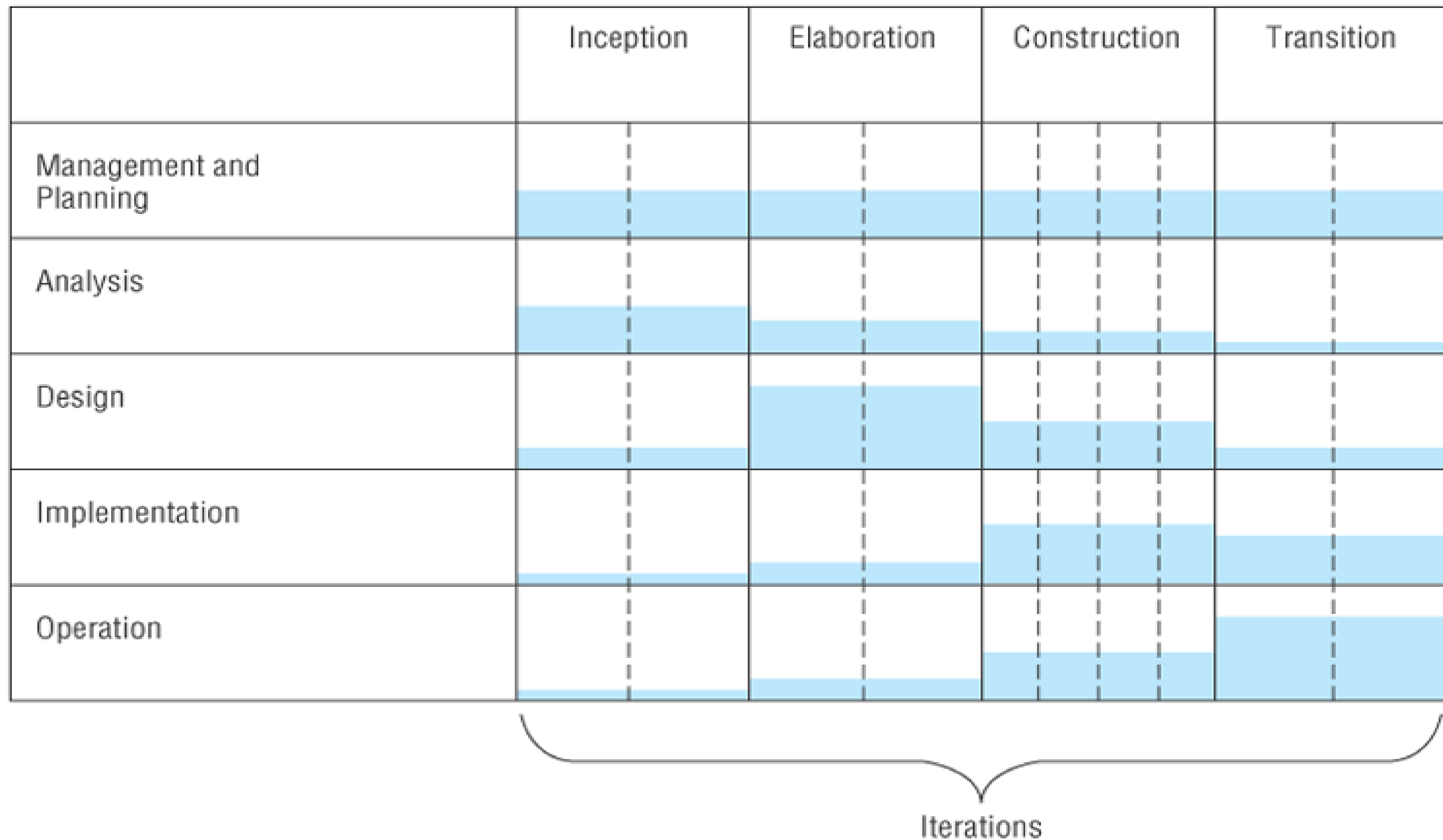
- ❖ Qui trình phát triển phần mềm RUP (*Rational Unified Process*)
- ❖ Kiểm thử phần mềm là gì?
- ❖ Các nguyên tắc kiểm thử phần mềm
- ❖ Các hạn chế của kiểm thử phần mềm
- ❖ Sơ đồ tổ chức của nhóm kiểm thử
- ❖ Mô hình V
- ❖ Qui trình kiểm thử phần mềm

# Quy trình phát triển phần mềm RUP



Hình 1.1. Các giai đoạn phát triển RUP (*Rational Unified Process*)

# Quy trình phát triển phần mềm RUP



Hình 1.2. Iterative development: Mapping of phases to workflows

# Quy trình phát triển phần mềm RUP

## ❖ Chu kỳ phần mềm (*software cycle*)

- ▶ Bắt đầu: các yêu cầu của người sử dụng
- ▶ Kết thúc: phần mềm đáp ứng đúng các yêu cầu của người sử dụng.

## ❖ Các giai đoạn của chu kỳ phần mềm

- ▶ Bắt đầu (*inception*)
- ▶ Phát triển (*elaboration*)
- ▶ Xây dựng (*construction*)
- ▶ Chuyển giao (*transition*)

## ❖ Mỗi giai đoạn thường được thực hiện lặp nhiều lần để kết quả ngày càng tốt hơn.

# Quy trình phát triển phần mềm RUP

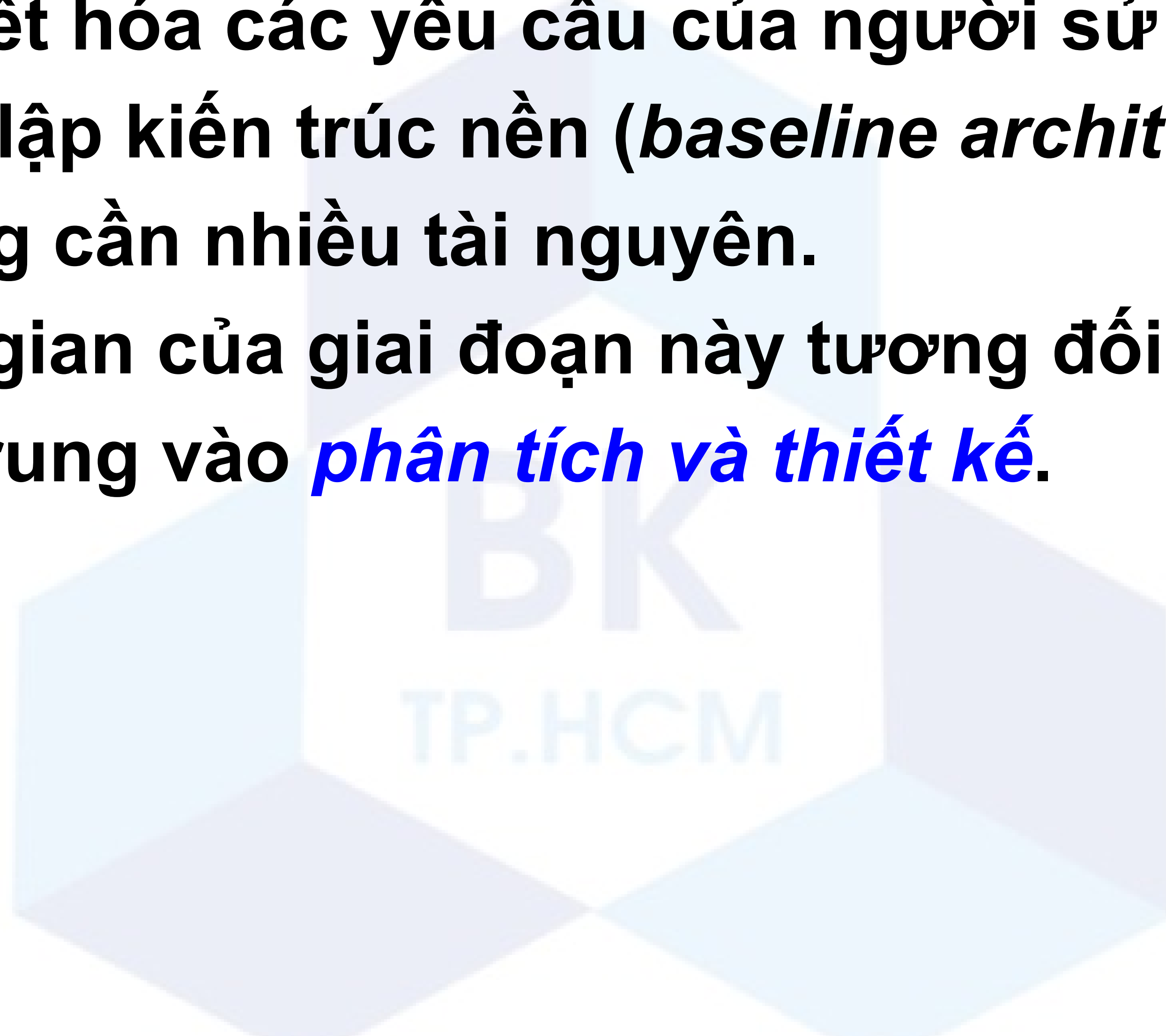
## ❖ Giai đoạn 1. Bắt đầu (*inception*)

- ▶ Định nghĩa phạm vi (*scope*).
- ▶ Xác định tính khả thi (*feasibility*).
- ▶ Hiểu các yêu cầu của người sử dụng (*user requirement*).
- ▶ Chuẩn bị kế hoạch phát triển phần mềm (*software development plan*).
- ▶ Các yêu cầu về tài nguyên tương đối ít.
- ▶ Thời gian của giai đoạn này ngắn.
- ▶ Tập trung vào việc ***lập kế hoạch và phân tích***.

# Quy trình phát triển phần mềm RUP

## ❖ Giai đoạn 2. Phát triển (*elaboration*)

- ▶ Chi tiết hóa các yêu cầu của người sử dụng.
- ▶ Thiết lập kiến trúc nền (*baseline architecture*).
- ▶ Không cần nhiều tài nguyên.
- ▶ Thời gian của giai đoạn này tương đối dài.
- ▶ Tập trung vào *phân tích và thiết kế*.





# Quy trình phát triển phần mềm RUP

## ❖ Giai đoạn 3. Xây dựng (*construction*)

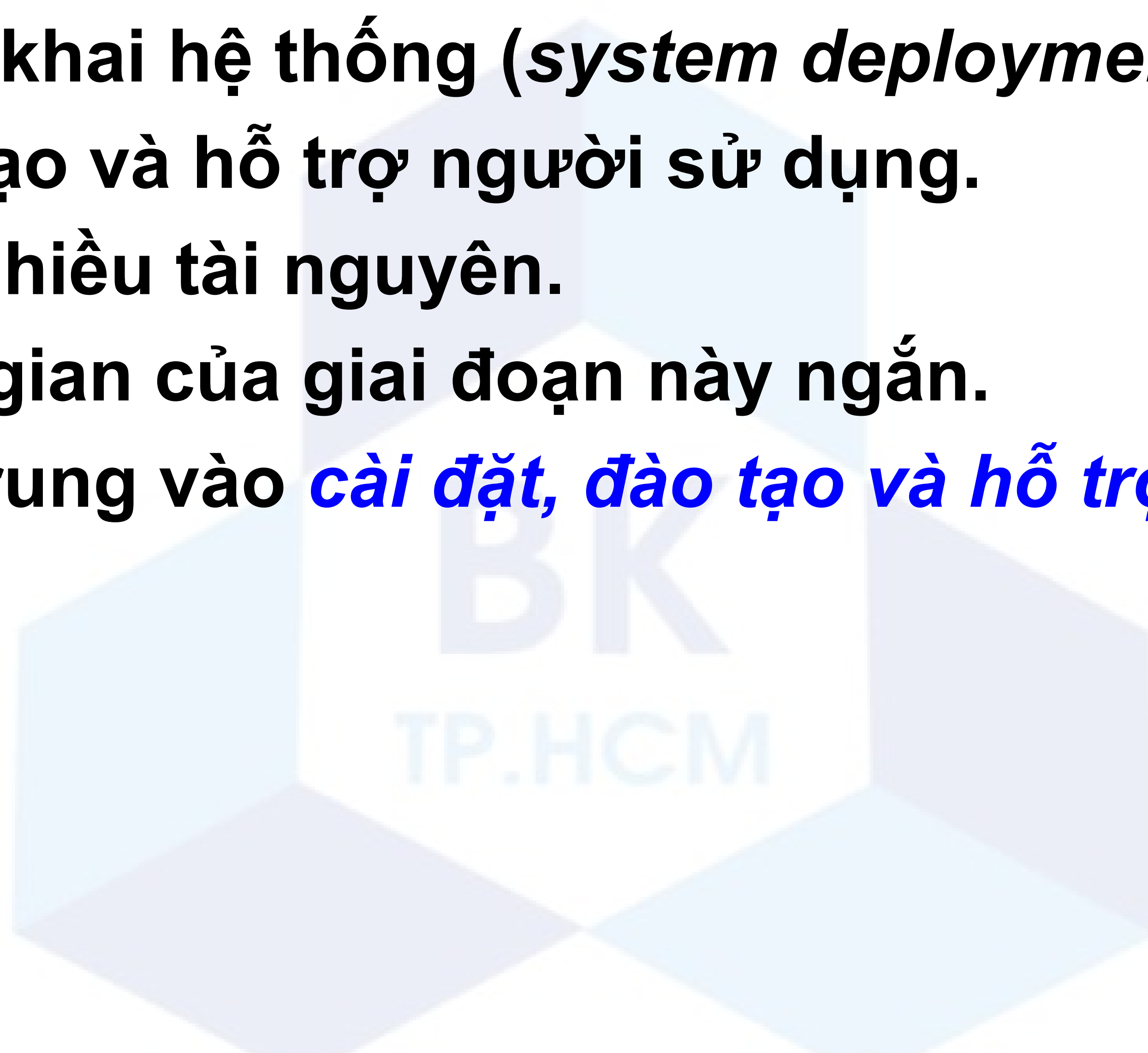
- ▶ Lập trình (*coding*).
- ▶ Kiểm tra (*testing*).
- ▶ Lập tài liệu lập trình.
- ▶ Cần nhiều tài nguyên.
- ▶ Thời gian của giai đoạn này là dài nhất.
- ▶ Tập trung vào các **công việc thực hiện** (*implementation task*).



# Quy trình phát triển phần mềm RUP

## ❖ Giai đoạn 4. Chuyển giao (*transition*)

- ▶ Triển khai hệ thống (*system deployment*).
- ▶ Đào tạo và hỗ trợ người sử dụng.
- ▶ Cần nhiều tài nguyên.
- ▶ Thời gian của giai đoạn này ngắn.
- ▶ Tập trung vào *cài đặt, đào tạo và hỗ trợ*.



# Các vấn đề thường gặp

## ❖ Các vấn đề thường gặp trong phát triển phần mềm

- ▶ Tính toán bị sai.
- ▶ Hiệu chỉnh bị sai dữ liệu.
- ▶ Kết hợp các dữ liệu bị sai.
- ▶ Tìm kiếm dữ liệu bị sai so với yêu cầu.
- ▶ Không thỏa mãn các ràng buộc của nghiệp vụ.
- ▶ Hiệu suất của phần mềm còn thấp.
- ▶ Kết quả không đáng tin cậy.
- ▶ Chưa đáp ứng đầy đủ các yêu cầu nghiệp vụ.
- ▶ Chưa giao tiếp được với các hệ thống khác.
- ▶ Bảo mật chưa tốt.

# Kiểm thử phần mềm là gì?

- ❖ Kiểm thử phần mềm là qui trình chứng minh phần mềm không có lỗi sai.
- ❖ Mục đích của kiểm thử phần mềm: phần mềm thực hiện đúng các yêu cầu của người sử dụng.
- ❖ Kiểm thử phần mềm là qui trình chạy phần mềm để tìm kiếm các lỗi sai.

# Kiểm thử phần mềm là gì?

- ❖ Kiểm thử phần mềm cho thấy phần mềm này thực hiện đúng các yêu cầu của nó và phát hiện các khiếm khuyết của nó trước khi đưa vào sử dụng.
- ❖ Kiểm tra các kết quả của lần chạy thử để phát hiện lỗi sai, các bất thường hoặc thông tin về các thuộc tính phi chức năng của chương trình.

# Mục tiêu của kiểm thử phần mềm

- ❖ Chứng minh cho người phát triển và khách hàng rằng phần mềm thoả mãn các yêu cầu của nó.
  - ▶ **Đối với phần mềm của khách hàng**, điều này có nghĩa là phải có ít nhất một lần kiểm tra cho mọi yêu cầu trong tài liệu các yêu cầu.
  - ▶ **Đối với các sản phẩm phần mềm nói chung**, điều này có nghĩa là phải có nhiều lần kiểm tra cho tất cả các tính năng của hệ thống, các tổ hợp của các tính năng này mà chúng kết hợp chặt chẽ với nhau trong phiên bản sản phẩm.



# Mục tiêu của kiểm thử phần mềm

- ❖ Phát hiện các trường hợp mà phần mềm chạy sai, không mong muốn hoặc không đúng theo đặc tả của nó.
  - ▶ **Kiểm tra khiếm khuyết** (*defect testing*) liên quan đến việc trừ khử cách hoạt động không mong muốn của hệ thống, ví dụ như hệ thống không chạy, các tương tác không mong muốn với các hệ thống khác, các tính toán bị sai và sai lệch dữ liệu.

# Mục tiêu của kiểm thử phần mềm

- ❖ Mục tiêu đầu tiên là **kiểm tra kiểm chứng** (*validation testing*)
  - ▶ Mong muốn hệ thống thực hiện đúng bằng cách sử dụng một tập các trường hợp kiểm tra để phản ánh hệ thống thực hiện đúng như mong muốn.
- ❖ Mục tiêu thứ hai là **kiểm tra khiếm khuyết** (*defect testing*)
  - ▶ Thiết kế các trường hợp kiểm tra để phát hiện các khiếm khuyết. Các trường hợp kiểm tra trong việc kiểm tra khiếm khuyết có thể không rõ ràng và không cần phản ánh cách hệ thống thường được sử dụng.



# Mục tiêu của kiểm thử phần mềm

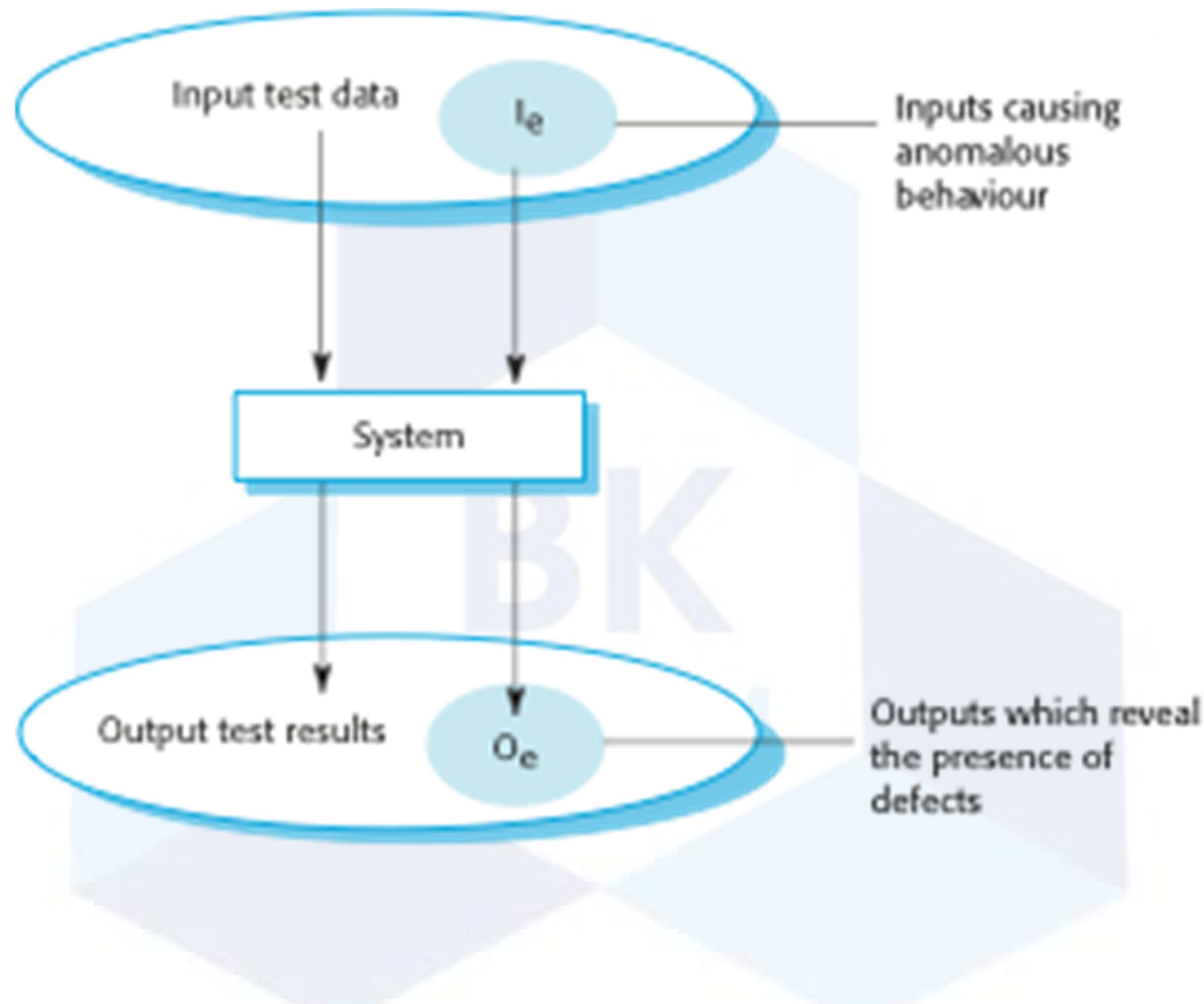
## ❖ Kiểm tra kiểm chứng

- ▶ Chứng minh cho người phát triển và khách hàng rằng phần mềm thỏa mãn các yêu cầu của nó.
- ▶ Một kiểm tra thành công cho thấy hệ thống hoạt động như mong muốn.

## ❖ Kiểm tra khiếm khuyết

- ▶ Phát hiện các lỗi sai hoặc các thiếu sót trong phần mềm mà cách hoạt động của nó bị sai hoặc không đúng với đặc tả của nó.
- ▶ Một kiểm tra thành công là kiểm tra làm cho hệ thống thực hiện sai và bộc lộ khiếm khuyết của hệ thống.

# Mục tiêu của kiểm thử phần mềm



# Kiểm chứng và kiểm định

## ❖ Kiểm định (*verification*)

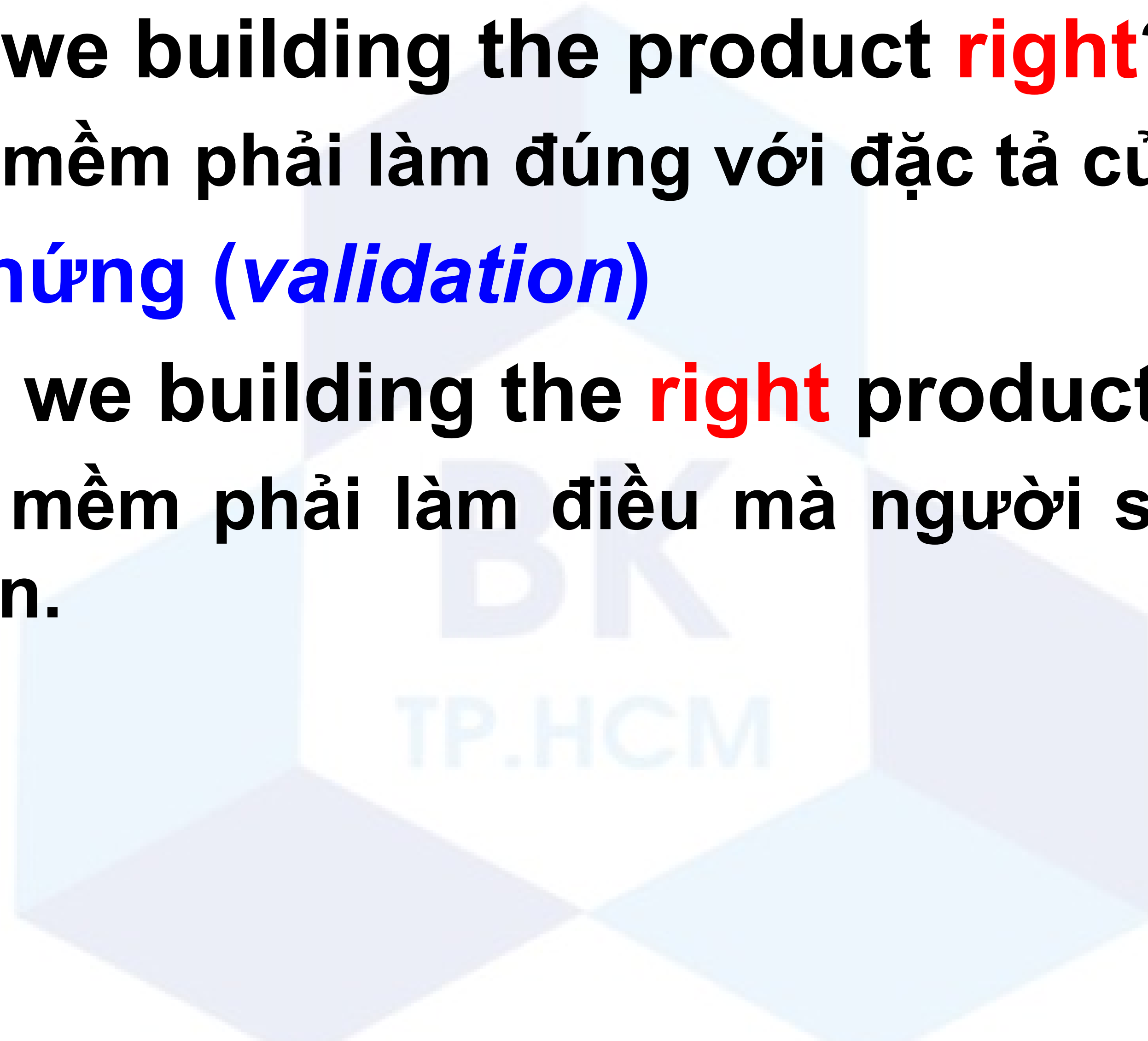
Are we building the product **right**?

- ▶ Phần mềm phải làm đúng với đặc tả của nó.

## ❖ Kiểm chứng (*validation*)

Are we building the **right** product?

- ▶ Phần mềm phải làm điều mà người sử dụng thật sự cần.



# Các nguyên tắc kiểm thử phần mềm

## ❖ Trường hợp kiểm tra (*test-case*)

- ▶ Mô tả: đặc tả các điều kiện cần có để tiến hành kiểm tra.
- ▶ Nhập: đặc tả đối tượng hay dữ liệu cần thiết, được sử dụng làm đầu vào để thực hiện việc kiểm tra.
- ▶ Kết quả mong muốn: kết quả trả về từ đối tượng kiểm tra, chứng tỏ đối tượng đạt yêu cầu.

## ❖ Thiết kế đầy đủ các *test-case*.

- ▶ dữ liệu nhập hợp lệ
- ▶ dữ liệu nhập không hợp lệ

# Các nguyên tắc kiểm thử phần mềm

- ❖ **Kiểm tra kết quả.**
  - ▶ **Kết quả mong muốn**
  - ▶ **Kết quả không mong muốn**





# Các nguyên tắc kiểm thử phần mềm

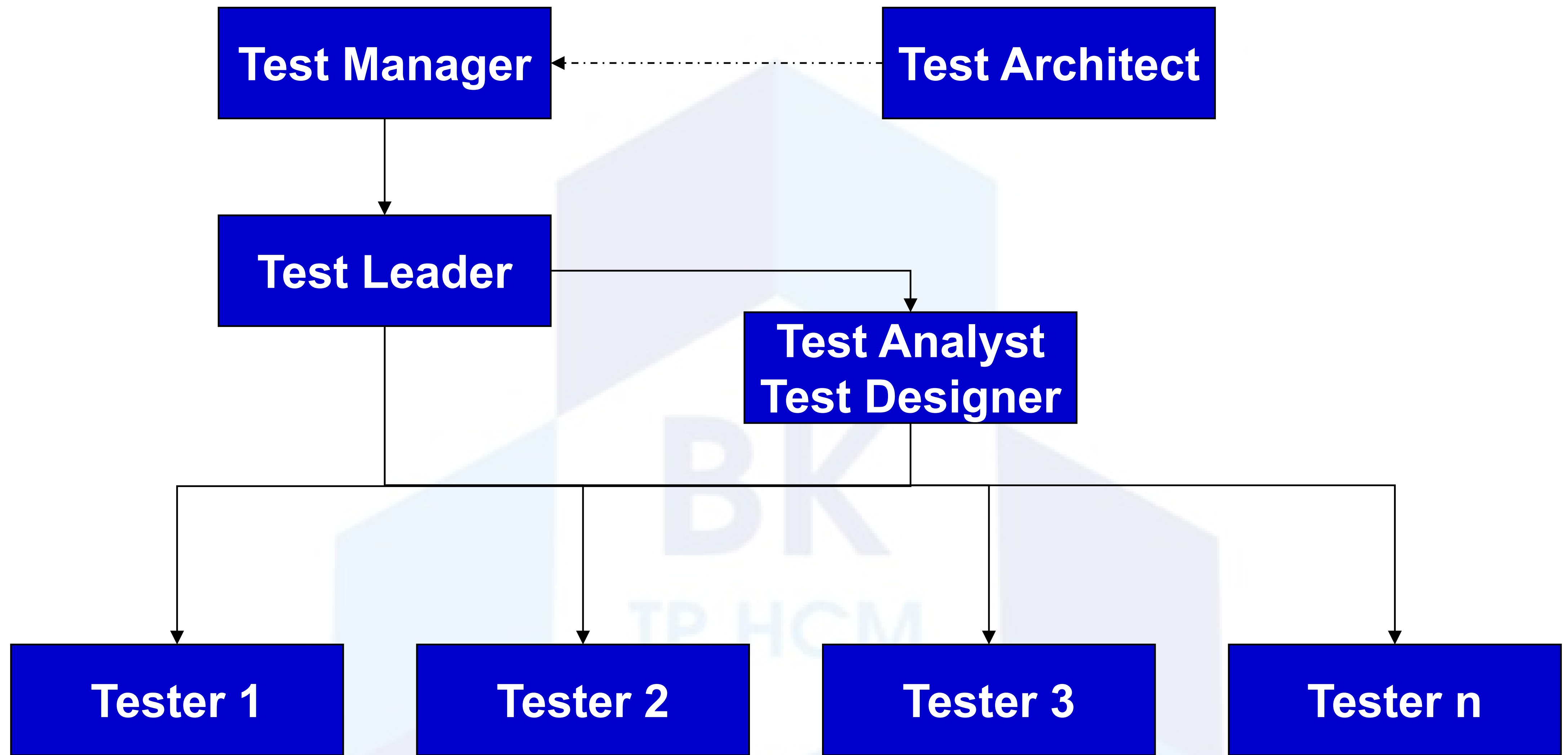
- ❖ Việc kiểm thử đòi hỏi tính độc lập: người lập trình nên tránh việc kiểm thử các chương trình của mình viết.
- ❖ Kiểm thử phần mềm nên bắt đầu từ các thành phần đơn giản, rồi đến các thành phần phức tạp hơn.
- ❖ Nên lập kế hoạch và qui trình kiểm thử trước khi bắt đầu kiểm thử.

# Các hạn chế của kiểm thử phần mềm

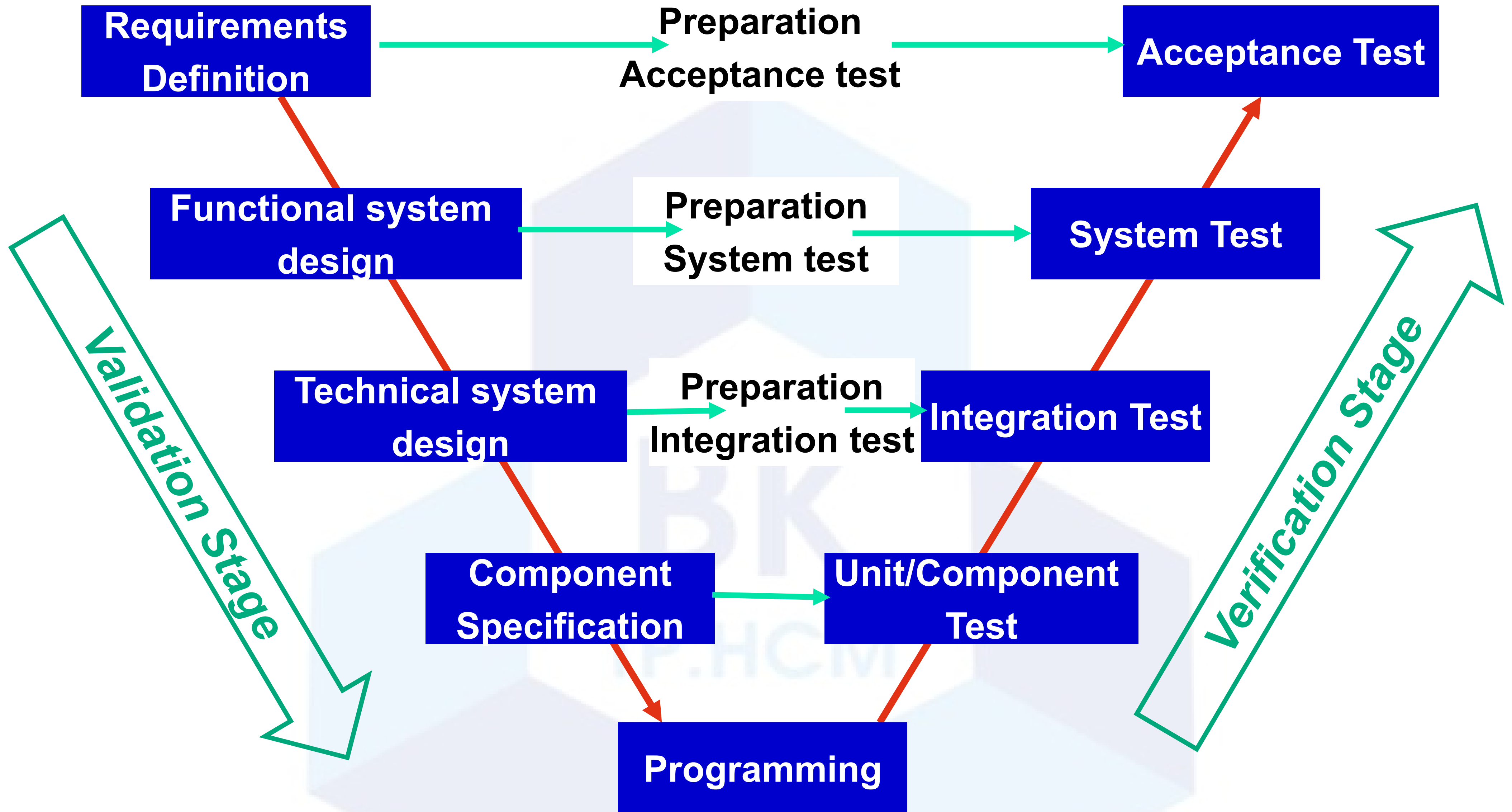
- ❖ Các đặc tả phần mềm có thể chưa đúng.
- ❖ Công cụ kiểm thử có thể chưa chắc đúng.
- ❖ Không có công cụ kiểm thử nào thích hợp cho mọi phần mềm.
- ❖ Kỹ sư kiểm thử có thể chưa hiểu đầy đủ về sản phẩm phần mềm.
- ❖ Không thể thực hiện kiểm thử phần mềm một cách đầy đủ.



# Sơ đồ tổ chức của nhóm kiểm thử



# Mô hình V



# Quy trình kiểm thử phần mềm

