

TS. Trần Đức Minh

Khóa học: Kiểm thử phần mềm

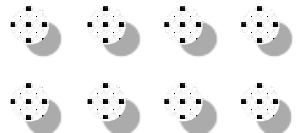
Bài 1

Nội dung trình bày

- Các khái niệm cơ bản
- Mô hình vòng đời kiểm thử
- Các hành vi chương trình
- Kiểm thử trên nền tảng đặc tả và code
- Phân loại lỗi
- Các mức độ kiểm thử



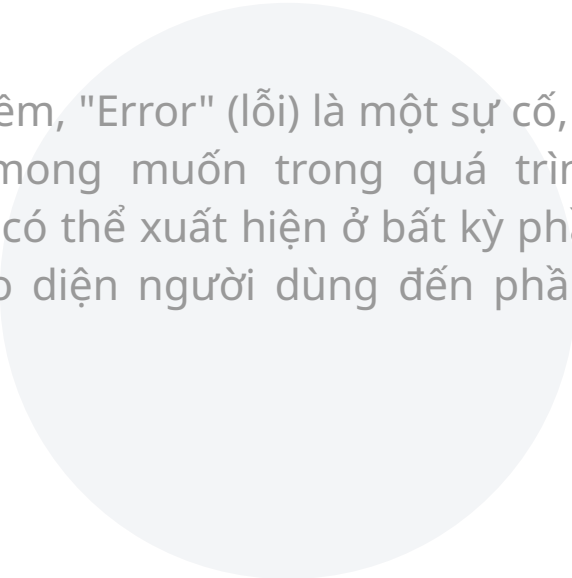
Illustrations by Pixeltrue on [icons8](#)



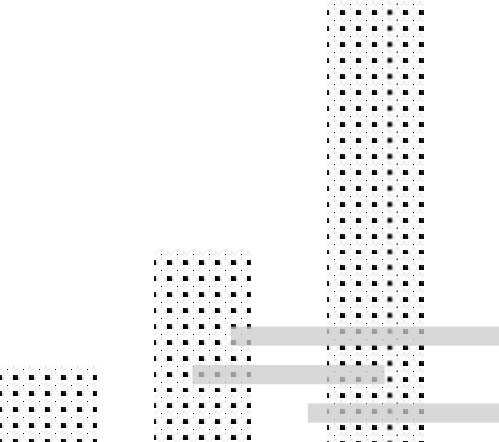


Các khái niệm cơ bản

Error



Trong kiểm thử phần mềm, "Error" (lỗi) là một sự cố, sai sót hoặc hành vi không mong muốn trong quá trình thử nghiệm phần mềm. Lỗi có thể xuất hiện ở bất kỳ phần nào của phần mềm, từ giao diện người dùng đến phần logic hoặc cơ sở dữ liệu.



Lỗi logic (Logic Error)

Đây là loại lỗi xảy ra khi một phần của mã không hoạt động như dự kiến. Điều này có thể do sai sót trong quá trình thiết kế hoặc triển khai mã.

Lỗi cú pháp (Syntax Error)

Đây là loại lỗi xảy ra khi mã không tuân thủ cú pháp của ngôn ngữ lập trình. Nó thường xảy ra khi bạn viết mã sai cú pháp, chẳng hạn như viết sai tên hàm hoặc thiếu dấu chấm phẩy.

Lỗi thời gian chạy (Runtime Error)

Đây là loại lỗi xảy ra khi chương trình đang chạy và gặp phải một tình huống không mong muốn, gây ra việc dừng chương trình hoặc hoạt động không đúng.

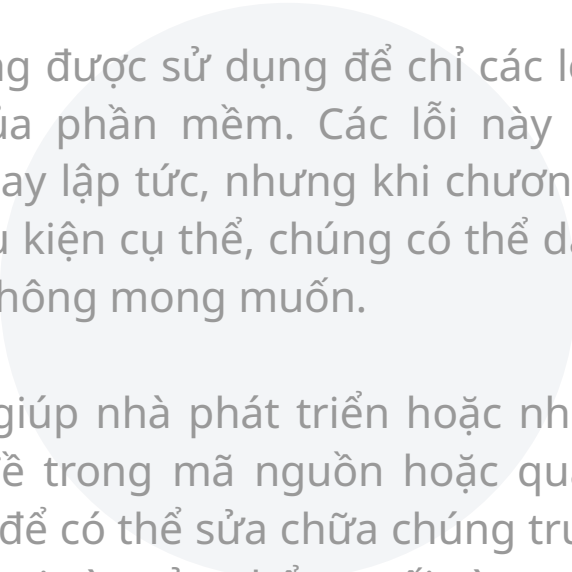
Lỗi giao diện người dùng (User Interface Error)

Đây là loại lỗi liên quan đến giao diện người dùng, như các lỗi trong việc hiển thị thông tin hoặc tương tác người dùng không hoạt động đúng cách.



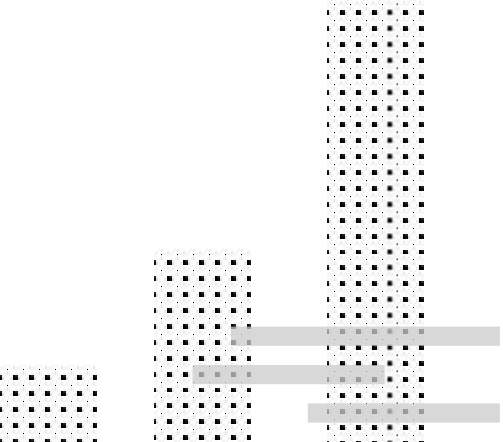
Các khái niệm cơ bản

Fault



Khái niệm "Fault" thường được sử dụng để chỉ các lỗi tiềm ẩn trong mã nguồn của phần mềm. Các lỗi này có thể không gây ra vấn đề ngay lập tức, nhưng khi chương trình thực hiện trong các điều kiện cụ thể, chúng có thể dẫn đến sự cố, lỗi hoặc hành vi không mong muốn.


Việc phát hiện "faults" giúp nhà phát triển hoặc nhà kiểm thử xác định các vấn đề trong mã nguồn hoặc quá trình thực thi của phần mềm để có thể sửa chữa chúng trước khi phần mềm được triển khai vào sản phẩm cuối cùng.





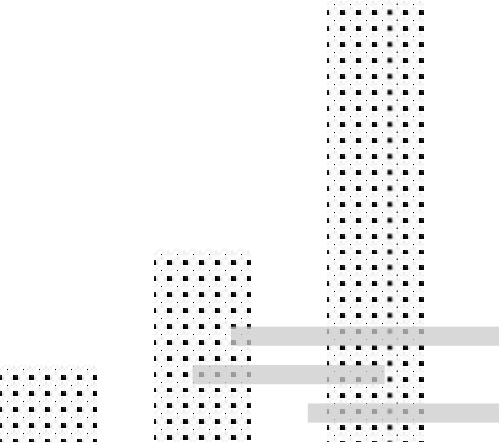
Các khái niệm cơ bản

Failure



Trong kiểm thử phần mềm, "Failure" (sự cố hoặc lỗi) là sự không thỏa mãn của một hệ thống hoặc phần mềm đối với yêu cầu hoặc kỳ vọng của người dùng. Nó là kết quả của việc một hoặc nhiều "faults" (lỗi) trong mã nguồn hoặc quá trình thực thi dẫn đến các vấn đề cụ thể mà người dùng có thể phát hiện được.

Một "Failure" xảy ra khi một hoặc nhiều thành phần của phần mềm không hoạt động như dự kiến, gây ra các vấn đề trong hoạt động của chương trình.



Chức năng không hoạt động

Một phần của phần mềm không thực hiện chức năng mong đợi của nó hoặc không thực hiện đúng cách.

Hiệu suất kém

Phần mềm hoạt động chậm hơn so với dự kiến hoặc không đáp ứng đúng với yêu cầu về hiệu suất.

Lỗi giao diện người dùng

Giao diện người dùng không hoạt động đúng cách hoặc không dễ sử dụng.


Sự cố hệ thống

Phần mềm gây ra sự cố hoặc crash hệ thống.

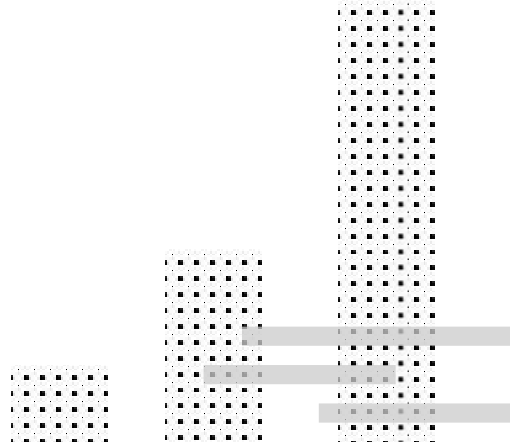


Các khái niệm cơ bản

Incident



Trong kiểm thử phần mềm, "Incident" (sự cố) là một sự kiện hoặc tình huống không mong muốn xảy ra trong quá trình thử nghiệm phần mềm. Điều này có thể là kết quả của việc phát hiện lỗi, sự cố hoặc hành vi không mong muốn trong phần mềm. Một "Incident" có thể là một "Failure" (sự cố) hoặc một vấn đề khác có thể ảnh hưởng đến quá trình thử nghiệm hoặc hoạt động của phần mềm.



Chức năng không hoạt động

Một phần của phần mềm không thực hiện chức năng mong đợi của nó hoặc không thực hiện đúng cách.

Hiệu suất kém

Phần mềm hoạt động chậm hơn so với dự kiến hoặc không đáp ứng đúng với yêu cầu về hiệu suất.

Lỗi giao diện người dùng

Giao diện người dùng không hoạt động đúng cách hoặc không dễ sử dụng.


Sự cố hệ thống

Phần mềm gây ra sự cố hoặc crash hệ thống.

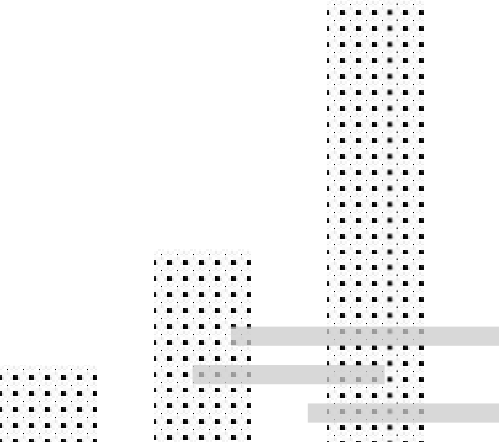
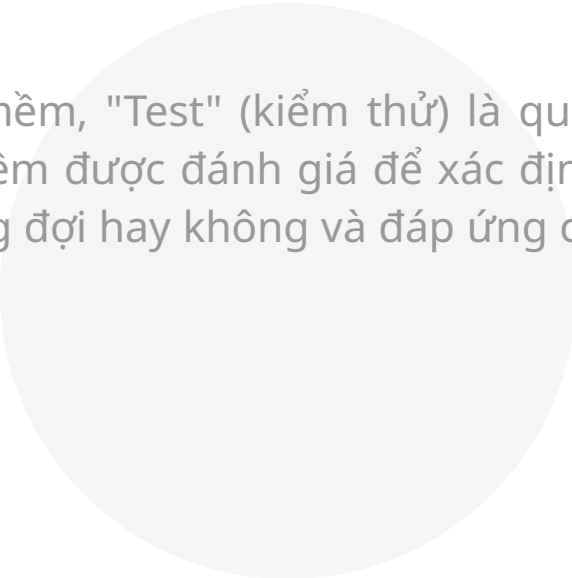


Các khái niệm cơ bản

Test



Trong kiểm thử phần mềm, "Test" (kiểm thử) là quá trình hệ thống hoặc phần mềm được đánh giá để xác định xem nó hoạt động như mong đợi hay không và đáp ứng các yêu cầu được đặt ra.



Quá trình kiểm thử phần mềm thường bao gồm việc thực hiện các bước:

1. Lập kế hoạch kiểm thử

Xác định mục tiêu của kiểm thử, xác định phạm vi và tài nguyên cần thiết, và phát triển kế hoạch kiểm thử.

2. Thiết kế kiểm thử

Xác định các ca kiểm thử, tạo ra các bộ dữ liệu kiểm thử, và lập kế hoạch cho các bước kiểm thử cụ thể.

3. Thực hiện kiểm thử

Thực hiện các ca kiểm thử theo kế hoạch, thực hiện các bộ dữ liệu kiểm thử và ghi lại kết quả.

4. Phân tích kết quả

Phân tích kết quả kiểm thử để xác định xem có bất kỳ lỗi nào được phát hiện không và mức độ hoàn thiện của phần mềm.

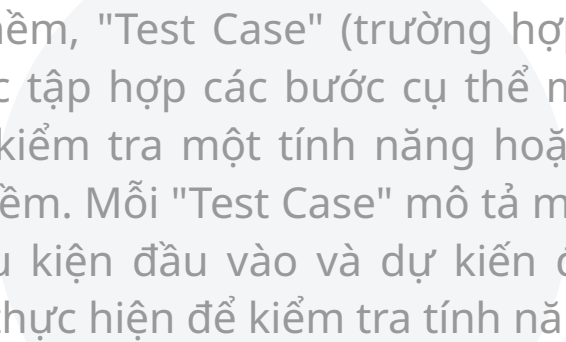
5. Báo cáo kết quả

Tạo báo cáo về kết quả kiểm thử, bao gồm các lỗi phát hiện, tình trạng của các yêu cầu kiểm thử, và đề xuất các biện pháp khắc phục.

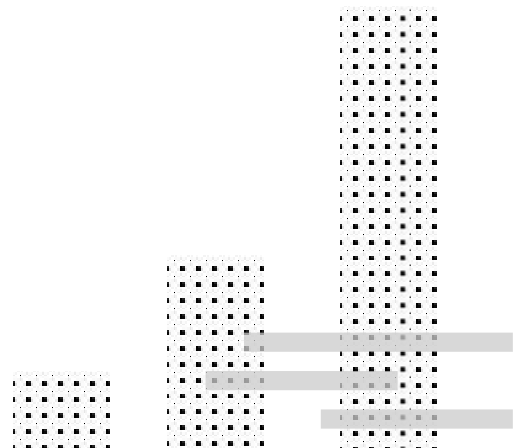


Các khái niệm cơ bản

Test Case



Trong kiểm thử phần mềm, "Test Case" (trường hợp kiểm thử) là một tài liệu hoặc tập hợp các bước cụ thể mà nhà kiểm thử thực hiện để kiểm tra một tính năng hoặc chức năng cụ thể của phần mềm. Mỗi "Test Case" mô tả một tình huống cụ thể hoặc điều kiện đầu vào và dự kiến đầu ra, cũng như các bước cần thực hiện để kiểm tra tính năng đó.



Một "Test Case" thường bao gồm các yếu tố:

Mô tả

Mô tả ngắn gọn về mục tiêu của trường hợp kiểm thử và tính năng hoặc chức năng cụ thể được kiểm tra.

Điều kiện đầu vào

Các điều kiện hoặc dữ liệu cần thiết để thực hiện trường hợp kiểm thử.

Bước thực hiện

Các bước cụ thể mà nhà kiểm thử phải thực hiện để kiểm tra tính năng hoặc chức năng. Bước này bao gồm cả các hành động và các kỳ vọng về kết quả.

Kỳ vọng đầu ra

Kết quả mong đợi của việc thực hiện trường hợp kiểm thử, bao gồm cả các giá trị mong đợi hoặc hành vi của phần mềm.

Ví dụ "Test Case" cho ứng dụng đăng nhập

Mô tả

Kiểm tra tính năng đăng nhập của ứng dụng.

Điều kiện đầu vào

Trang đăng nhập của ứng dụng được hiển thị.

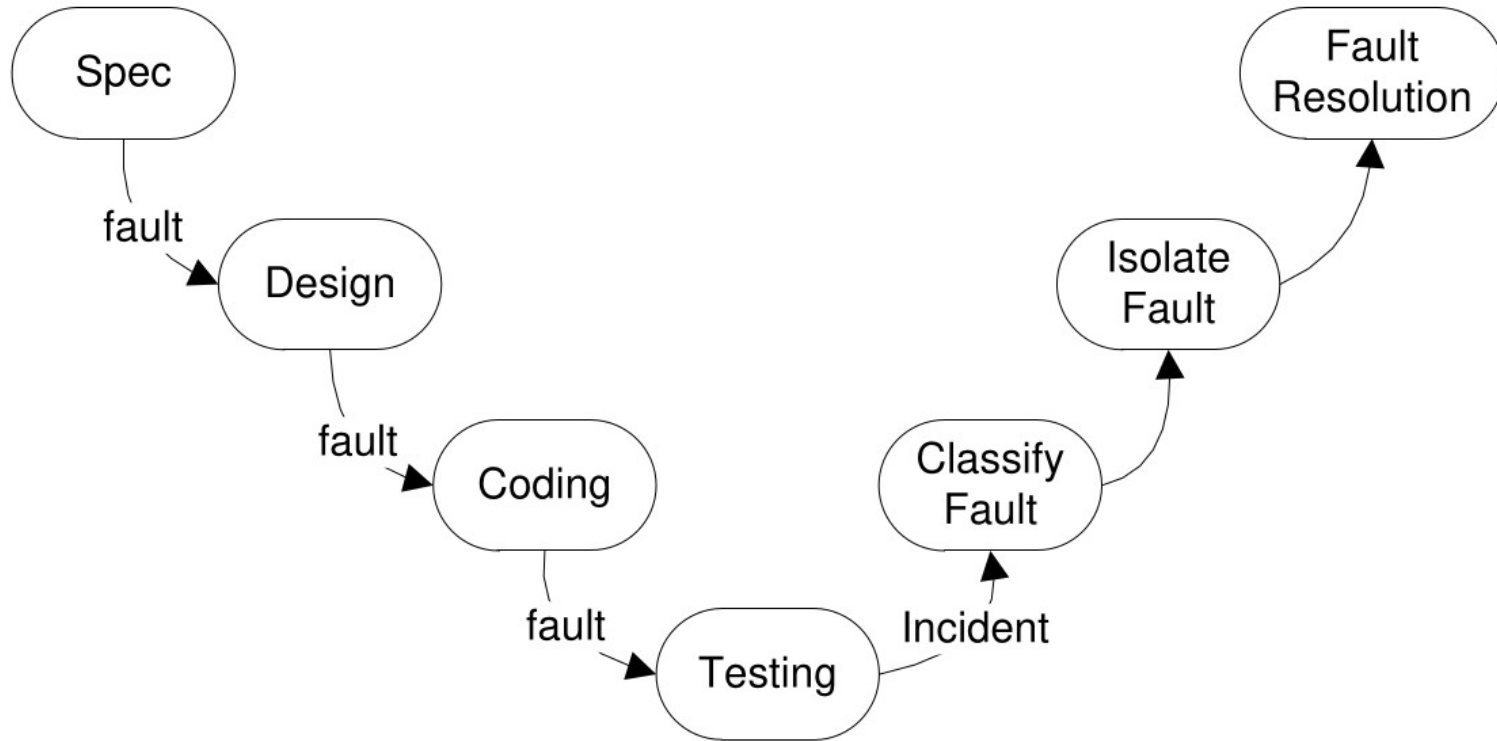
Bước thực hiện

- Nhập tên người dùng và mật khẩu.
- Nhấn nút "Đăng nhập".

Kỳ vọng đầu ra

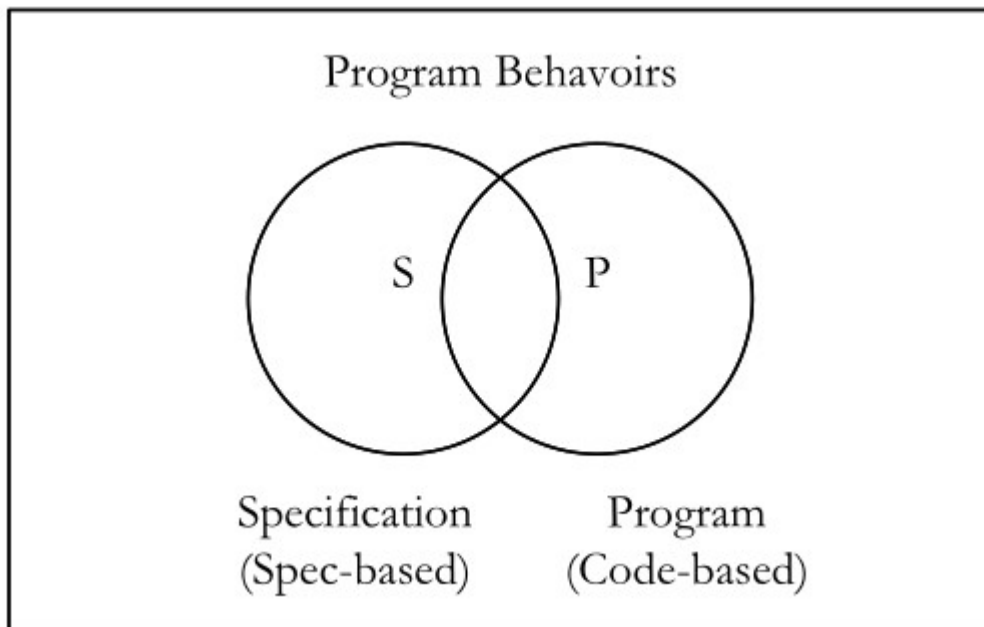
Ứng dụng chuyển hướng đến trang chính sau khi đăng nhập thành công.

Một mô hình vòng đời kiểm thử



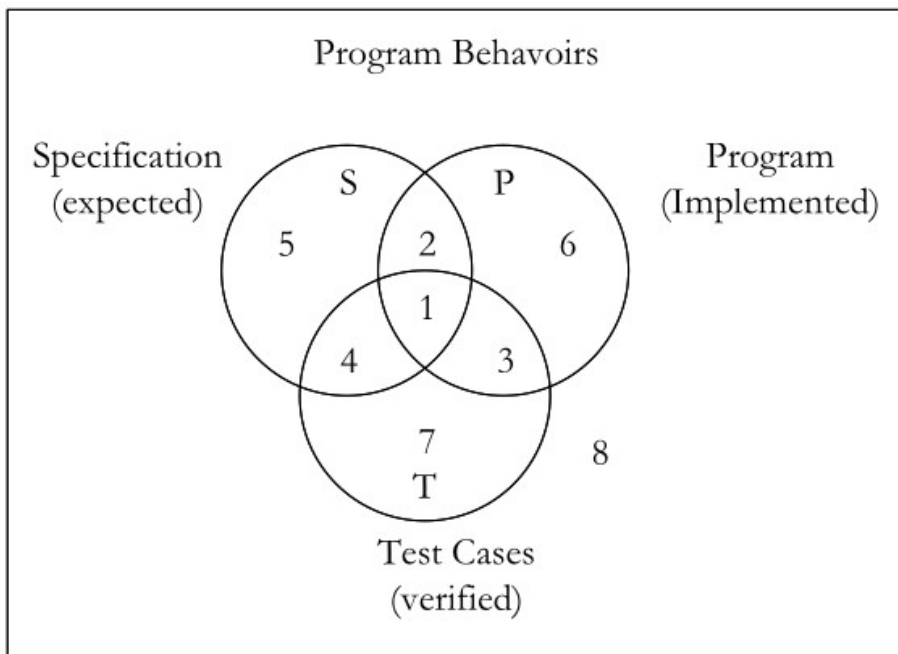
Các hành vi chương trình được đặc tả và thực hiện

- Tập S gồm các hành vi được đặc tả và tập P gồm các hành vi được lập trình
- Giao điểm của S và P là phần "correct", tức là các hành vi vừa được đặc tả vừa được lập trình.



Các hành vi được đặc tả, lập trình và kiểm thử

- Vùng 2 và 5: Các hành vi được đặc tả nhưng không được kiểm thử.
- Vùng 1 và 4: Các hành vi được đặc tả và được kiểm thử.
- Vùng 3 và 7: Vùng kiểm thử các hành vi chưa được đặc tả.

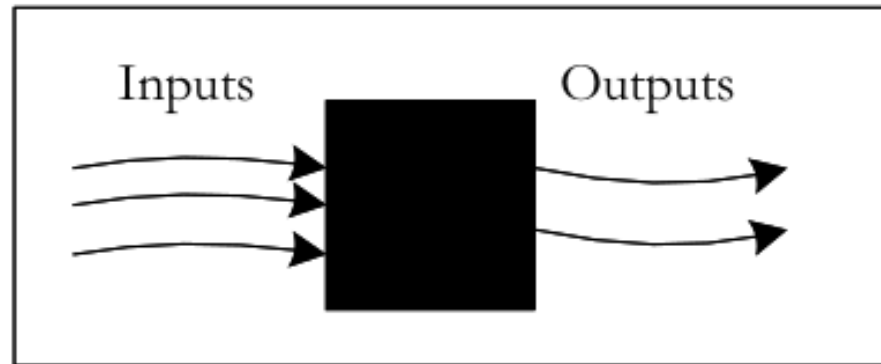


- Vùng 2 và 6: Các hành vi được lập trình nhưng không được kiểm thử.
- Vùng 1 và 3: Các hành vi được lập trình và được kiểm thử.
- Vùng 4 và 7: Vùng kiểm thử các hành vi chưa được lập trình.

Vùng 1 ?

Kiểm thử trên nền tảng đặc tả (1)

- Thử nghiệm trên nền tảng đặc tả còn được gọi là “thử nghiệm chức năng” hay “thử nghiệm hộp đen”.
- Nội dung (việc triển khai) của hộp đen không được biết đến và chức năng của hộp đen được hiểu hoàn toàn theo các đầu vào và đầu ra của nó.



Kiểm thử trên nền tảng đặc tả (2)

- Ưu điểm:

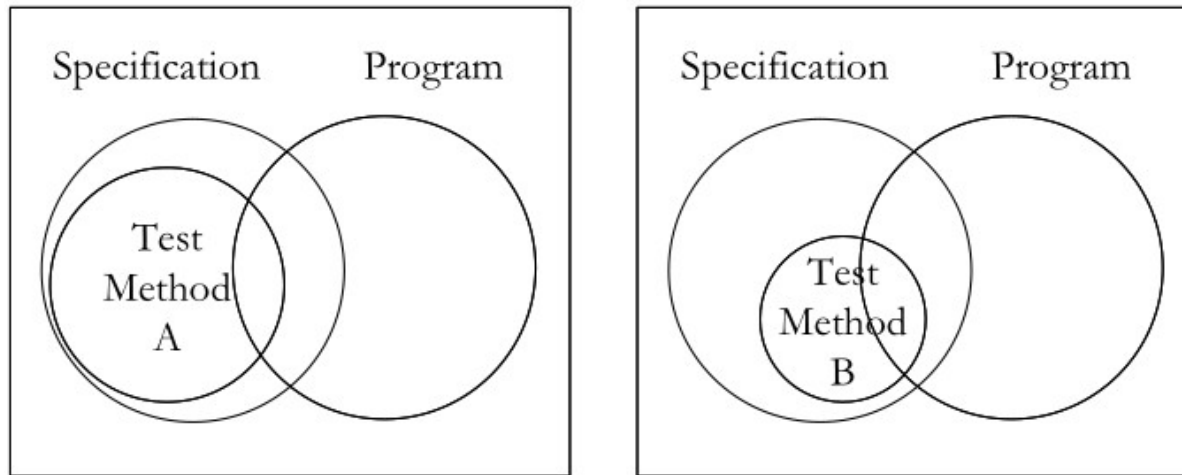
- Độc lập với cách triển khai phần mềm, vì vậy nếu cách triển khai thay đổi thì các ca kiểm thử vẫn hữu ích.
- Việc phát triển test case có thể diễn ra song song với việc triển khai, do đó làm giảm thời gian phát triển dự án tổng thể.

- Nhược điểm:

- Sự dư thừa đáng kể có thể tồn tại giữa các Test case.
- Có thể những lỗi hỏng của phần mềm chưa được kiểm thử

So sánh các phương pháp xác định Test case dựa trên nền tảng đặc tả

- Đối với cả hai phương thức A và B, tập hợp các Test case hoàn toàn nằm trong tập hợp các hành vi được chỉ định.
- Phương thức A xác định tập hợp các Test case lớn hơn Phương thức B.

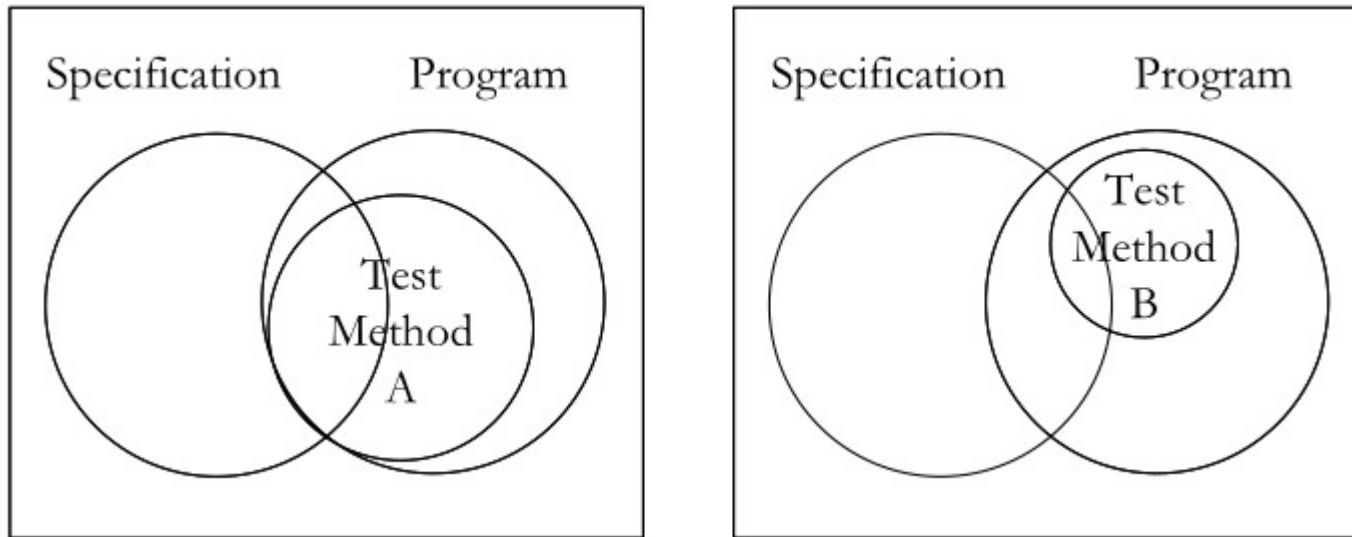


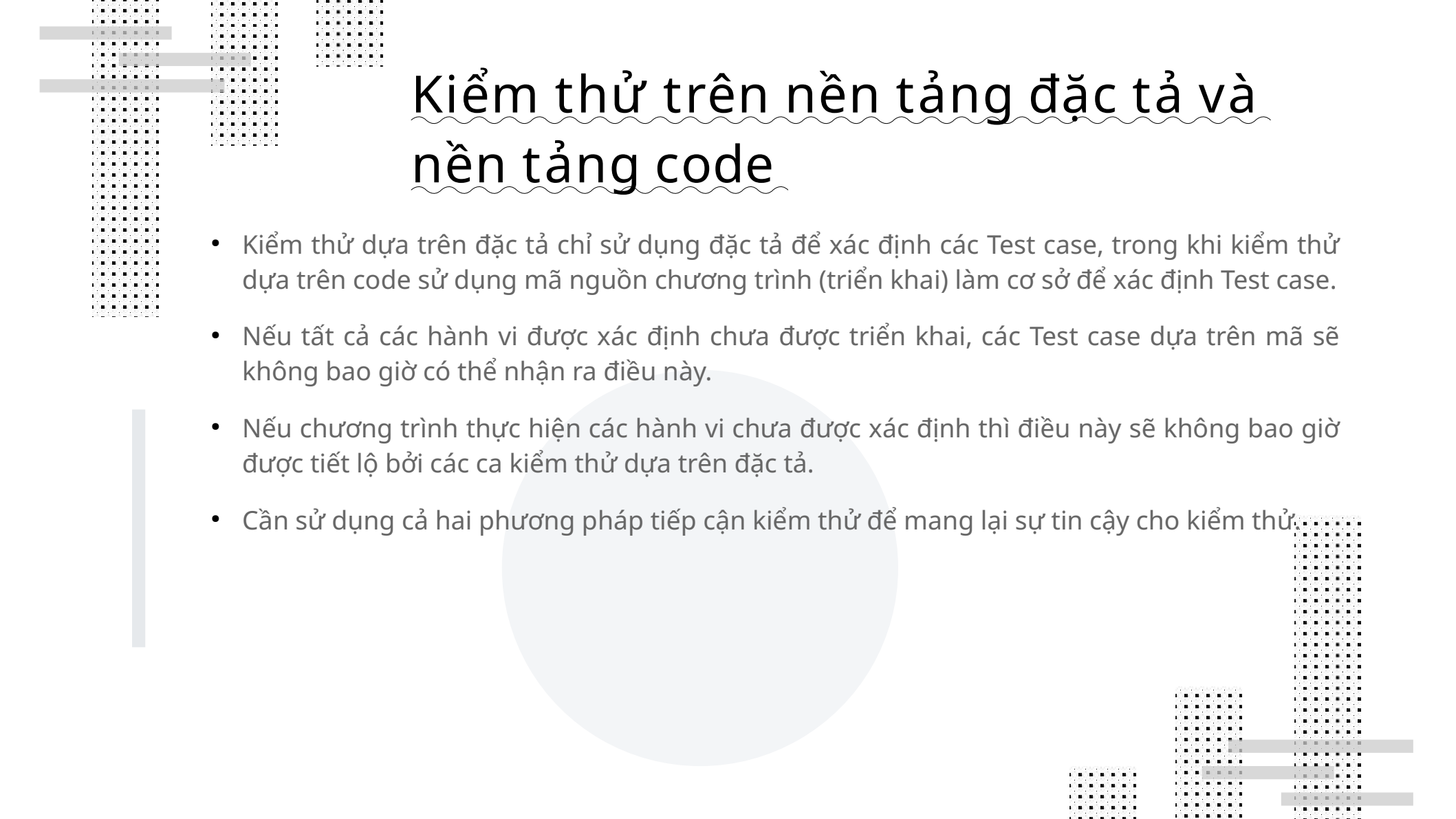
Kiểm thử trên nền tảng code (1)

- Thử nghiệm trên nền tảng code còn được gọi là “thử nghiệm hộp trắng”.
- Khả năng “nhìn thấy bên trong” hộp đen cho phép người kiểm tra xác định các Test Case dựa trên cách chức năng được triển khai thực tế.

Kiểm thử trên nền tảng code (2)

- Đối với cả hai phương thức A và B, tập hợp các Test case hoàn toàn nằm trong tập hợp các hành vi được chỉ định.
- Phương thức A xác định tập hợp các Test case lớn hơn Phương thức B.





Kiểm thử trên nền tảng đặc tả và nền tảng code

- Kiểm thử dựa trên đặc tả chỉ sử dụng đặc tả để xác định các Test case, trong khi kiểm thử dựa trên code sử dụng mã nguồn chương trình (triển khai) làm cơ sở để xác định Test case.
- Nếu tất cả các hành vi được xác định chưa được triển khai, các Test case dựa trên mã sẽ không bao giờ có thể nhận ra điều này.
- Nếu chương trình thực hiện các hành vi chưa được xác định thì điều này sẽ không bao giờ được tiết lộ bởi các ca kiểm thử dựa trên đặc tả.
- Cần sử dụng cả hai phương pháp tiếp cận kiểm thử để mang lại sự tin cậy cho kiểm thử.

Phân loại lỗi (1)

- Đảm bảo chất lượng phần mềm (SQA - Software Quality Assurance) thường cố gắng cải thiện sản phẩm bằng cách cải tiến quy trình.
 - SQA quan tâm nhiều hơn đến việc giảm các lỗi phổ biến trong quá trình phát triển.
- Kiểm thử định hướng sản phẩm nhiều hơn.
 - Kiểm thử quan tâm nhiều hơn đến việc phát hiện ra các lỗi trong sản phẩm.

Phân loại lỗi (2)

- Phân loại tiêu chuẩn IEEE về các bất thường của phần mềm (IEEE, 1993) cho biết cách xử lý toàn diện các lỗi.

Table 1.1 Input/Output Faults

Type	Instances
Input	Correct input not accepted
	Incorrect input accepted
	Description wrong or missing
	Parameters wrong or missing
Output	Wrong format
	Wrong result
	Correct result at wrong time (too early, too late)
	Incomplete or missing result
	Spurious result
	Spelling/grammar
	Cosmetic

Table 1.2 Logic Faults

Missing case(s)
Duplicate case(s)
Extreme condition neglected
Misinterpretation
Missing condition
Extraneous condition(s)
Test of wrong variable
Incorrect loop iteration
Wrong operator (e.g., < instead of \leq)

Phân loại lỗi (3)

- Phân loại tiêu chuẩn IEEE về các bất thường của phần mềm (IEEE, 1993) cho biết cách xử lý toàn diện các lỗi.

Table 1.3 Computation Faults

Incorrect algorithm
Missing computation
Incorrect operand
Incorrect operation
Parenthesis error
Insufficient precision (round-off, truncation)
Wrong built-in function

Table 1.4 Interface Faults

Incorrect interrupt handling
I/O timing
Call to wrong procedure
Call to nonexistent procedure
Parameter mismatch (type, number)
Incompatible types
Superfluous inclusion

Phân loại lỗi (4)

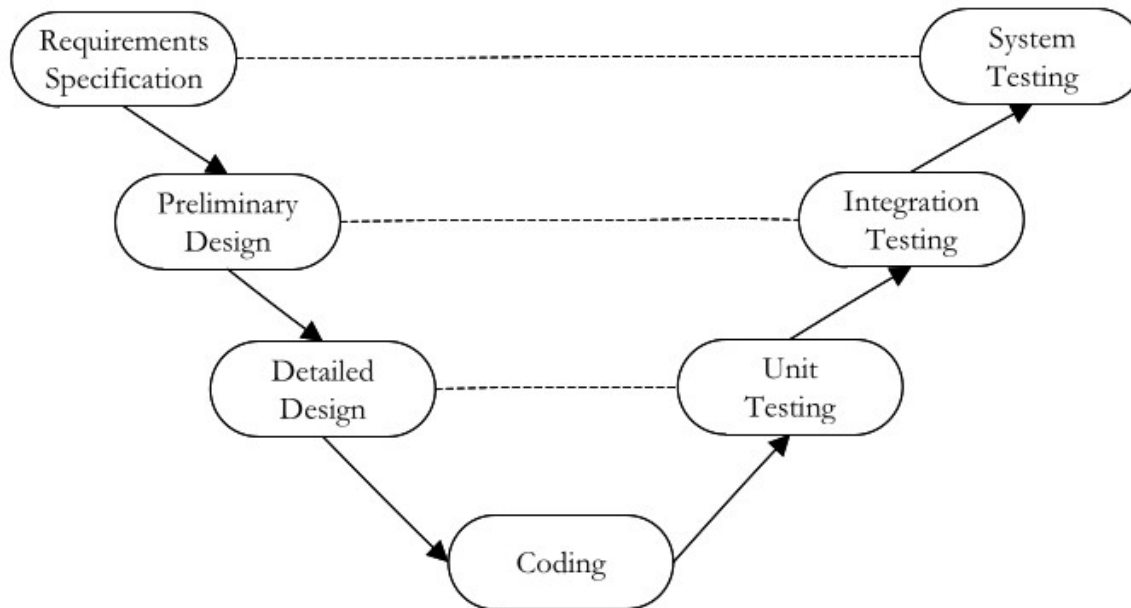
- Phân loại tiêu chuẩn IEEE về các bất thường của phần mềm (IEEE, 1993) cho biết cách xử lý toàn diện các lỗi.

Table 1.5 Data Faults

Incorrect initialization
Incorrect storage/access
Wrong flag/index value
Incorrect packing/unpacking
Wrong variable used
Wrong data reference
Scaling or units error
Incorrect data dimension
Incorrect subscript
Incorrect type
Incorrect data scope
Sensor data out of limits
Off by one
Inconsistent data

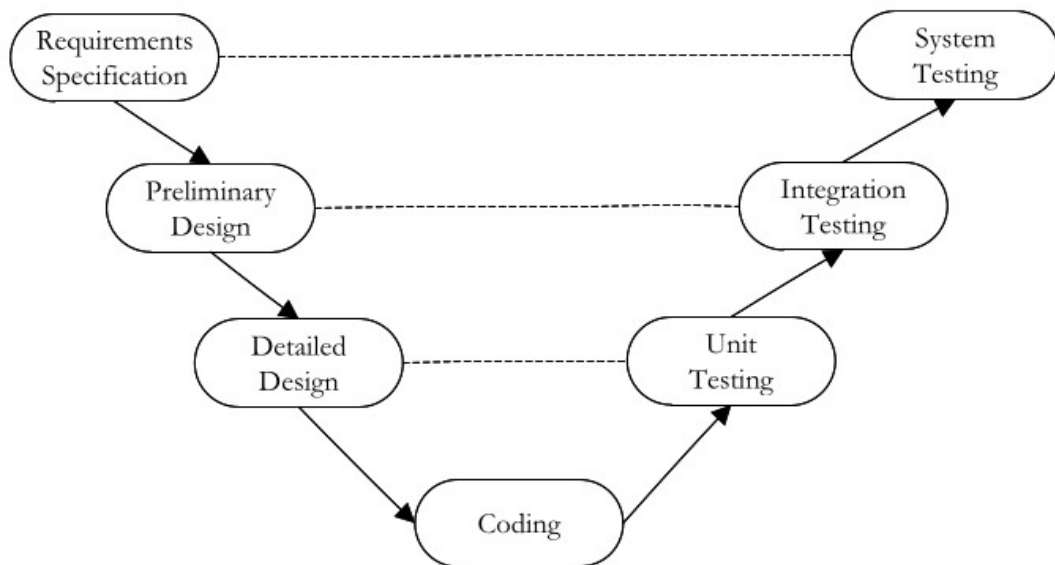
Các mức độ kiểm thử (1)

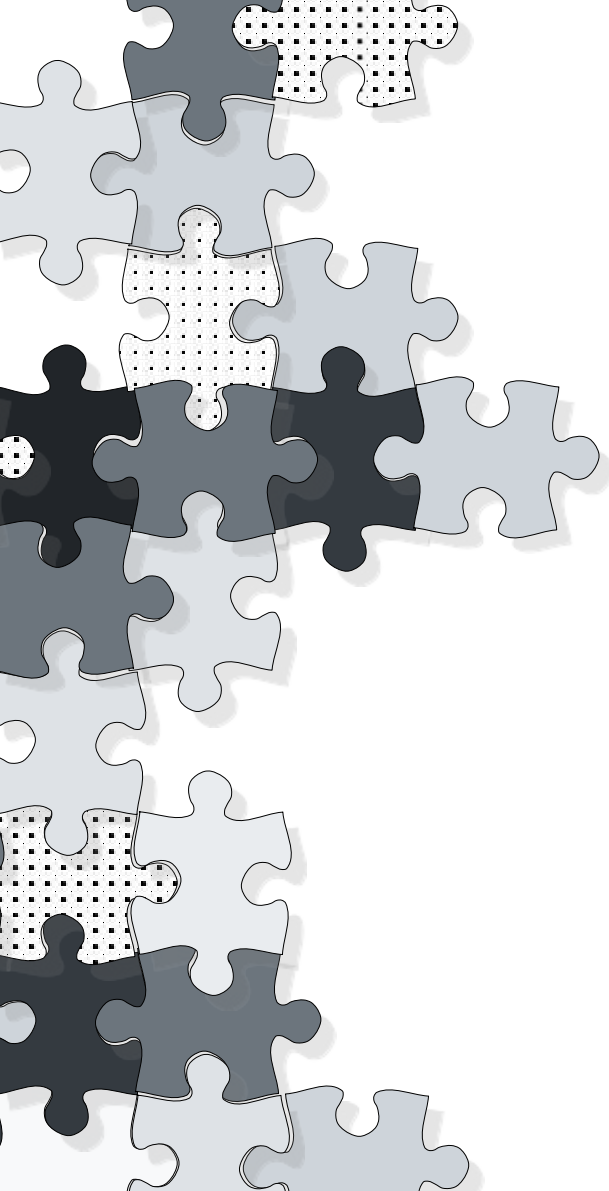
- Các mức độ kiểm thử phản ánh các mức độ trừu tượng được tìm thấy trong Mô hình thác nước (Waterfall) của vòng đời phát triển phần mềm.



Các mức độ kiểm thử (2)

- Kiểm tra dựa trên nền tảng code phù hợp nhất ở cấp độ kiểm thử đơn vị.
- Kiểm tra dựa trên nền tảng đặc tả phù hợp nhất ở cấp độ hệ thống.





“

Kết thúc Bài 1

”

