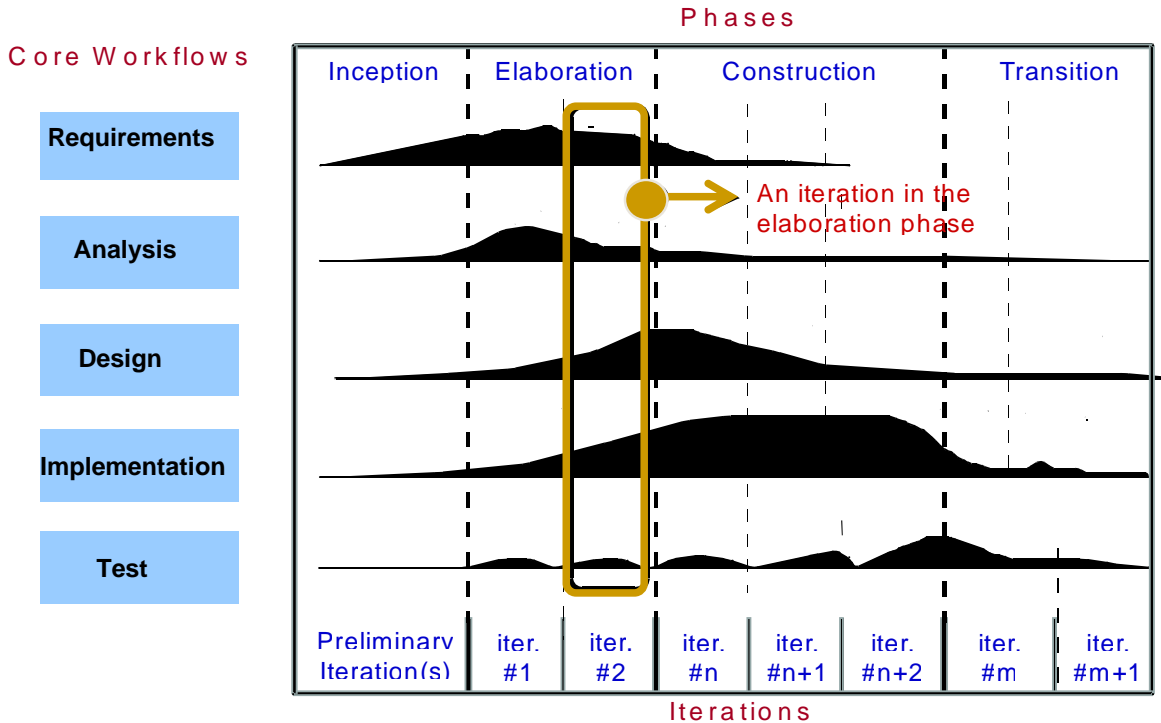


Tổng quát về kiểm thử phần mềm

1.1 Qui trình phát triển phần mềm RUP



Chu kỳ phần mềm được tính từ lúc có yêu cầu (mới hoặc nâng cấp) đến lúc phần mềm đáp ứng đúng yêu cầu được phân phối.

Trong mỗi chu kỳ, người ta tiến hành nhiều công đoạn : khởi động, chi tiết hóa, hiện thực và chuyển giao.

Mỗi công đoạn thường được thực hiện theo cơ chế lặp nhiều lần để kết quả ngày càng hoàn hảo hơn.

Trong từng bước lặp, chúng ta thường thực hiện nhiều workflows đồng thời (để tận dụng nguồn nhân lực hiệu quả nhất) : nắm bắt yêu cầu, phân tích chức năng, thiết kế, hiện thực và kiểm thử.

Sau mỗi lần lặp thực hiện 1 công việc nào đó, ta phải tạo ra kết quả (artifacts), kết quả của bước/công việc này là dữ liệu đầu

vào của bước/công việc khác. Nếu thông tin không tốt sẽ ảnh hưởng nghiêm trọng đến kết quả của các bước/hoạt động sau đó.

Một số vấn đề thường gặp trong phát triển phần mềm :

- tính toán không đúng, hiệu chỉnh sai dữ liệu.
- trộn dữ liệu không đúng.
- Tìm kiếm dữ liệu sai yêu cầu.
- Xử lý sai mối quan hệ giữa các dữ liệu.
- Coding/hiện thực sai các qui luật nghiệp vụ.
- Hiệu suất của phần mềm còn thấp.
- Kết quả hoặc hiệu suất phần mềm không tin cậy.
- Hỗ trợ chưa đủ các nhu cầu nghiệp vụ.
- Giao tiếp với hệ thống khác chưa đúng hay chưa đủ.
- Kiểm soát an ninh phần mềm chưa đủ.

1.2 Vài định nghĩa về kiểm thử phần mềm

- Kiểm thử phần mềm là qui trình chứng minh phần mềm không có lỗi.
- Mục đích của kiểm thử phần mềm là chỉ ra rằng phần mềm thực hiện đúng các chức năng mong muốn.
- Kiểm thử phần mềm là qui trình thiết lập sự tin tưởng về việc phần mềm hay hệ thống thực hiện được điều mà nó hỗ trợ.
- Kiểm thử phần mềm là qui trình thi hành phần mềm với ý định tìm kiếm các lỗi của nó.
- Kiểm thử phần mềm được xem là qui trình cố gắng tìm kiếm các lỗi của phần mềm theo tinh thần "hủy diệt".

Các mục tiêu chính của kiểm thử phần mềm :

- Phát hiện càng nhiều lỗi càng tốt trong thời gian kiểm thử xác định trước.
- Chứng minh rằng sản phẩm phần mềm phù hợp với các đặc tả yêu cầu của nó.
- Xác thực chất lượng kiểm thử phần mềm đã dùng chi phí và nỗ lực tối thiểu.
- Tạo các testcase chất lượng cao, thực hiện kiểm thử hiệu quả và tạo ra các báo cáo vấn đề đúng và hữu dụng.

Kiểm thử phần mềm là 1 thành phần trong lĩnh vực rộng hơn, đó là Verification & Validation (V &V), ta tạm dịch là Thanh kiểm tra và Kiểm định phần mềm.

Thanh kiểm tra phần mềm là qui trình xác định xem sản phẩm của 1 công đoạn trong qui trình phát triển phần mềm có thoả mãn các yêu cầu đặt ra trong công đoạn trước không (Ta có đang xây dựng đúng đắn sản phẩm không ?)

Thanh kiểm tra phần mềm thường là hoạt động kỹ thuật vì nó dùng các kiến thức về các artifacts, các yêu cầu, các đặc tả rời rạc của phần mềm.

Các hoạt động Thanh kiểm tra phần mềm bao gồm kiểm thử (testing) và xem lại (reviews).

Kiểm định phần mềm là qui trình đánh giá phần mềm ở cuối chu kỳ phát triển để đảm bảo sự bằng lòng sử dụng của khách hàng (Ta có xây dựng phần mềm đúng theo yêu cầu khách hàng ?).

Các hoạt động kiểm định được dùng để đánh giá xem các tính chất được hiện thực trong phần mềm có thoả mãn các yêu cầu khách hàng và có thể theo dõi với các yêu cầu khách hàng không ?

Kiểm định phần mềm thường phụ thuộc vào kiến thức của lĩnh vực mà phần mềm xử lý.

Verification

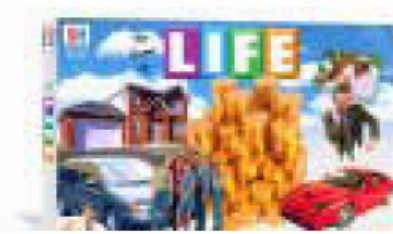
Are we building the product right?



"I landed on "Go" but didn't get my \$200!"

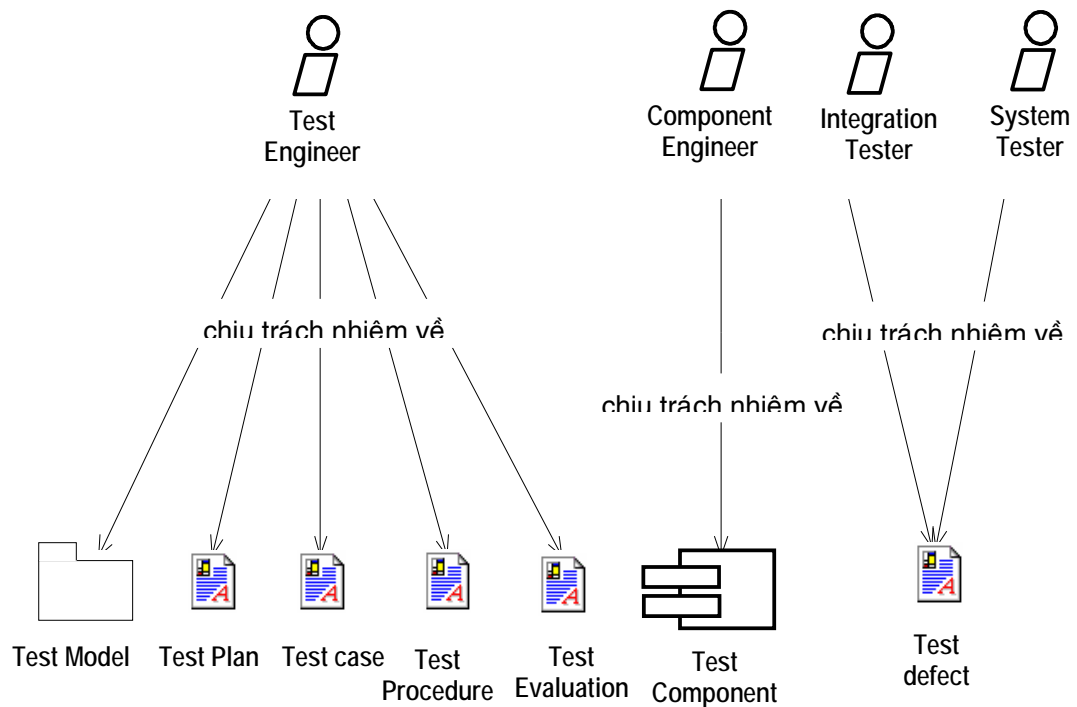
Validation

Are we building the right product?



"I know this game has money and players and "Go" – but this is not the game I wanted."

1.3 Kiểm thử : các worker và qui trình



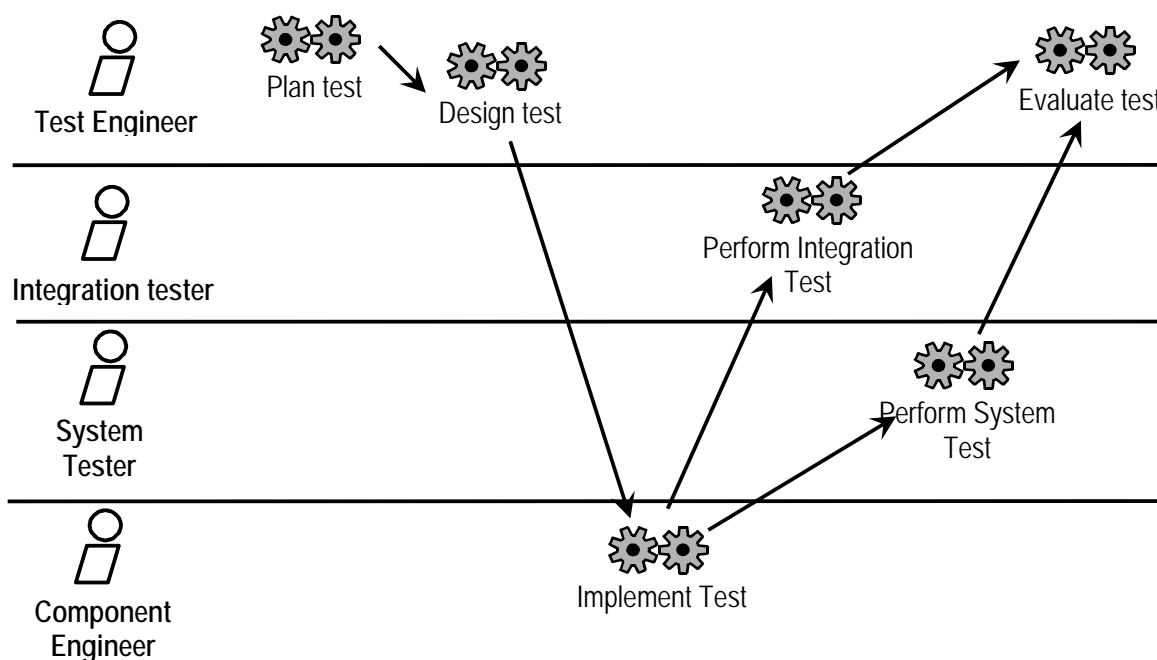
Kỹ sư kiểm thử :

- người chuyên về IT, chịu trách nhiệm về nhiều hoạt động kỹ thuật liên quan đến kiểm thử.

- định nghĩa các testcase, viết các đặc tả và thủ tục kiểm thử.
- phân tích kết quả, báo cáo kết quả cho các người phát triển và quản lý biết.
- ...

Người quản lý kiểm thử :

- Thiết lập chiến lược và qui trình kiểm thử, tương tác với các người quản lý về các hoạt động khác trong project, giúp đỡ kỹ sư kiểm thử thực hiện công việc của họ.



Tự động 1 số hoạt động kiểm thử

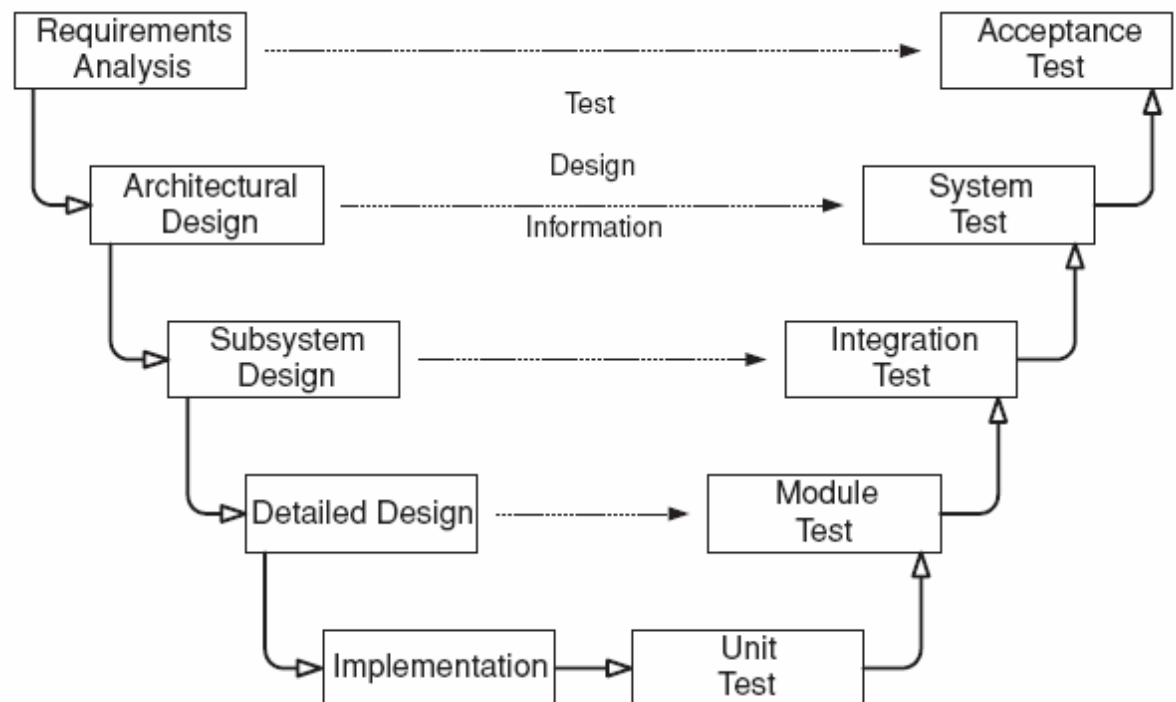
Kiểm thử phần mềm tốn nhiều chi phí nhân công, thời gian. Trong 1 số dự án, kiểm thử phần mềm tiêu hao trên 50% tổng giá phát triển phần mềm. Nếu cần ứng dụng an toàn hơn, chi phí kiểm thử còn cao hơn nữa.

Do đó 1 trong các mục tiêu của kiểm thử là tự động hóa nhiều như có thể, nhờ đó mà giảm thiểu chi phí rất nhiều, tối thiểu hóa các lỗi do người gây ra, đặc biệt giúp việc kiểm thử hồi qui dễ dàng và nhanh chóng hơn.

Tự động hóa việc kiểm thử là dùng phần mềm điều khiển việc thi hành kiểm thử, so sánh kết quả có được với kết quả kỳ vọng, thiết lập các điều kiện đầu vào, các kiểm soát kiểm thử và các chức năng báo cáo kết quả...

Thí dụ các tiện ích phục vụ tự động kiểm thử như : Stress Test, Selenium, TestComplete, IBM Rational Functional Tester.

1.4 Các mức độ kiểm thử phần mềm



- Kiểm thử đơn vị (Unit Testing) : kiểm thử sự hiện thực chi tiết của từng đơn vị nhỏ (hàm, class,...) có hoạt động đúng không ?
- Kiểm thử module (Module Testing) : kiểm thử các dịch vụ của module có phù hợp với đặc tả của module đó không ?
- Kiểm thử tích hợp (Integration Testing) : kiểm thử xem từng phân hệ của phần mềm có đảm bảo với đặc tả thiết kế của phân hệ đó không ?
- Kiểm thử hệ thống (System Testing) : kiểm thử các yêu cầu không chức năng của phần mềm như hiệu suất, bảo mật, làm việc trong môi trường căng thẳng,...

- Kiểm thử độ chấp nhận của người dùng (Acceptance Testing) : kiểm tra xem người dùng có chấp thuận sử dụng phần mềm không ?
- Kiểm thử hồi qui : được làm mỗi khi có sự hiệu chỉnh, nâng cấp phần mềm với mục đích xem phần mềm mới có đảm bảo thực hiện đúng các chức năng trước khi hiệu chỉnh không ?

1.5 Testcase

Mỗi testcase chứa các thông tin cần thiết để kiểm thử thành phần phần mềm theo 1 mục tiêu xác định. Thường testcase gồm bộ 3 thông tin {tập dữ liệu đầu vào, trạng thái của thành phần phần mềm, tập kết quả kỳ vọng}

Tập dữ liệu đầu vào (Input): gồm các giá trị dữ liệu cần thiết để thành phần phần mềm dùng và xử lý.

Tập kết quả kỳ vọng : kết quả mong muốn sau khi thành phần phần mềm xử lý dữ liệu nhập.

Trạng thái thành phần phần mềm : được tạo ra bởi các giá trị prefix và postfix.

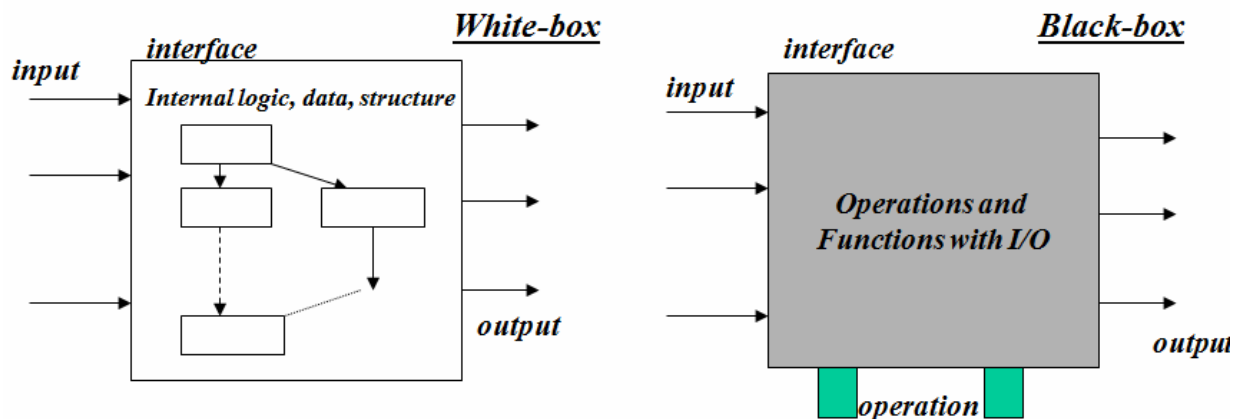
Tập các testcase : tập hợp các testcase mà ta có ý định dùng để kiểm thử thành phần phần mềm để minh chứng rằng TPPM có đúng các hành vi mong muốn.

Các phương pháp thiết kế testcase

Bất kỳ sản phẩm kỹ thuật nào (phần mềm không phải là ngoại lệ) đều có thể được kiểm thử bởi 1 trong 2 cách :

- Kiểm thử hộp đen (Black box testing) : theo góc nhìn sử dụng
 - Không cần kiến thức về chi tiết thiết kế và hiện thực bên trong.

- Kiểm thử dựa trên các yêu cầu và đặc tả sử dụng TPPM.
- Kiểm thử hộp trắng (White box testing) : theo góc nhìn hiện thực
 - cần kiến thức về chi tiết thiết kế và hiện thực bên trong.
 - Kiểm thử dựa vào phủ các lệnh, phủ các nhánh, phủ các điều kiện con,...



Kiểu kiểm thử	Kỹ thuật kiểm thử được dùng
Unit Testing	White Box, Black Box
Integration Testing	Black Box, White Box
Functional Testing	Black Box
System Testing	Black Box
Acceptance Testing	Black Box

1.6 Các nguyên tắc cơ bản về kiểm thử

Thông tin thiết yếu của mỗi testcase là kết quả hay dữ liệu xuất kỳ vọng.

Nếu kết quả kỳ vọng của testcase không được định nghĩa rõ ràng, người ta sẽ giải thích kết quả sai (plausible) thành kết quả đúng bởi vì hiện tượng “the eye seeing what it wants to see.”

=> 1 test case phải chứa 2 thành phần thiết yếu :

- đặc tả về điều kiện dữ liệu nhập.
- đặc tả chính xác về kết quả đúng của chương trình tương ứng với dữ liệu nhập.

Việc kiểm thử đòi hỏi tính độc lập : lập trình viên nên tránh việc kiểm thử các TPPM do mình viết.

Các issues tâm lý :

- Chương trình có thể chứa các lỗi do lập trình viên hiểu sai về đặc tả/phát biểu vấn đề.
- Tổ chức lập trình không nên kiểm thử các chương trình của tổ chức mình viết.
- Thanh tra 1 cách xuyên suốt các kết quả kiểm thử.

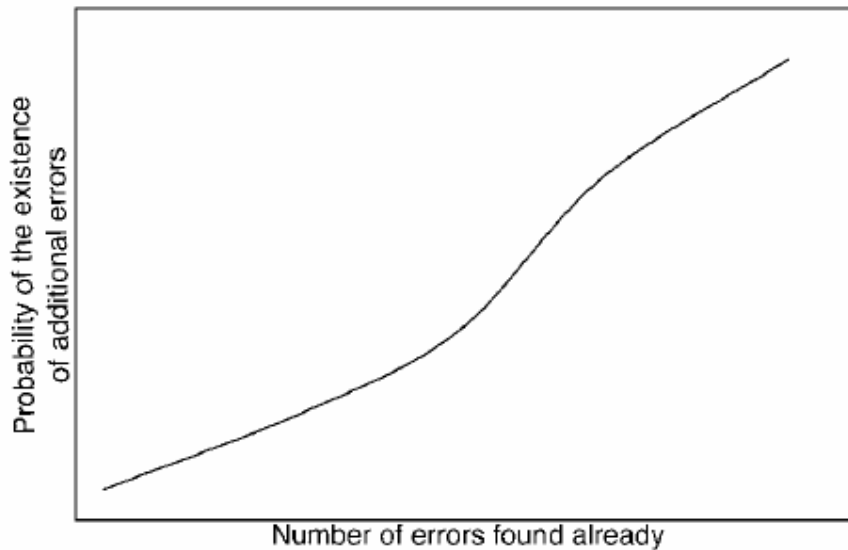
Phải thiết kế đủ các test case cho cả 2 trường hợp : dữ liệu đầu vào hợp lệ và dữ liệu đầu vào không hợp lệ và chờ đợi.

Xem xét chương trình xem nó không thực hiện những điều mong muốn, xem nó có làm những điều không mong muốn ?

Tránh các testcase "throwaway" trừ phi chương trình thật sự là "throwaway".

Không nên lập kế hoạch nỗ lực kiểm thử dựa trên giả định ngầm rằng phần mềm không có lỗi.

Xác xuất xuất hiện nhiều lỗi hơn trong 1 section phần mềm tỉ lệ thuận với số lỗi đã phát hiện được trong section đó.



Kiểm thử là 1 tác vụ rất thách thức đòi hỏi sự sáng tạo và trí tuệ.

Kiểm thử phần mềm nên bắt đầu từ các thành phần nhỏ đơn giản rồi đến các thành phần ngày càng lớn hơn.

Kiểm thử theo kiểu vét cạn là không thể.

Nên hoạch định qui trình kiểm thử trước khi bắt đầu thực hiện kiểm thử.

1.7 Các ý tưởng không đúng về kiểm thử

- Ta có thể kiểm thử phần mềm đầy đủ, nghĩa là đã vét cạn mọi hoạt động kiểm thử cần thiết.
- Ta có thể tìm tất cả lỗi nếu kỹ sư kiểm thử làm tốt công việc của mình.
- Tập các testcase tốt phải chứa rất nhiều testcase để bao phủ rất nhiều tình huống.
- Testcase tốt luôn là testcase có độ phức tạp cao.
- Tự động kiểm thử có thể thay thế kỹ sư kiểm thử để kiểm thử phần mềm 1 cách tốt đẹp.
- Kiểm thử phần mềm thì đơn giản và dễ dàng. Ai cũng có thể làm, không cần phải qua huấn luyện.

1.8 Các hạn chế của việc kiểm thử

- Ta không thể chắc là các đặc tả phần mềm đều đúng 100%.
- Ta không thể chắc rằng hệ thống hay tool kiểm thử là đúng.
- Không có tool kiểm thử nào thích hợp cho mọi phần mềm.
- Kỹ sư kiểm thử không chắc rằng họ hiểu đầy đủ về sản phẩm phần mềm.
- Ta không bao giờ có đủ tài nguyên để thực hiện kiểm thử đầy đủ phần mềm.
- Ta không bao giờ chắc rằng ta đạt đủ 100% hoạt động kiểm thử phần mềm.

1.9 Kết chương

Chương này đã ôn lại qui trình phát triển phần mềm được dùng phổ biến nhất hiện nay, đó là qui trình RUP (Rational Unified Process), từ đó giới thiệu các lý do cần phải kiểm thử phần mềm, các thuật ngữ cơ bản trong hoạt động kiểm thử phần mềm.

Chương này cũng đã giới thiệu vai trò của các worker trong qui trình kiểm thử phần mềm, các mức độ kiểm thử phần mềm khác nhau, các nguyên tắc cơ bản của kiểm thử phần mềm.

Chương này cũng đã giới thiệu 1 số ý tưởng không đúng đắn về kiểm thử phần mềm, những hạn chế của hoạt động kiểm thử phần mềm.