

Lý thuyết Kiểm Tra Phần Mềm

Bài 01:

Giới thiệu Kiểm tra Phần mềm

GV: Nguyễn Ngọc Tú

Email: nntu@hoasen.edu.vn

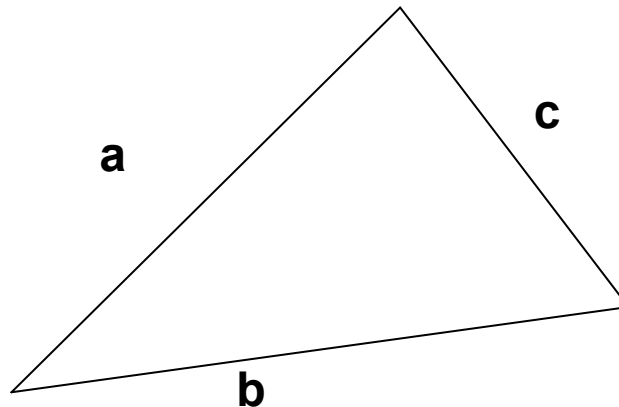
Bộ môn: Kỹ thuật Phần mềm

Nội dung

- Tại sao phải kiểm tra ?
 - Các hệ thống phần mềm
 - Nguyên nhân các khiếm khuyết phần mềm
 - Vai trò của việc kiểm tra trong quá trình phát triển phần mềm
 - Quan hệ Kiểm tra và chất lượng phần mềm
 - Kiểm tra đến mức nào là đủ ?
- Kiểm tra phần mềm là gì ?
- Các nguyên lý tổng quát của việc kiểm tra phần mềm
- Quy trình kiểm tra cơ bản
 - Lập kế hoạch và điều khiển
 - Phân tích và thiết kế
 - Hiện thực và việc thực thi
 - Đánh giá các tiêu chuẩn và lập báo cáo
 - Kiểm tra các hoạt động cuối
- Triết lý của việc kiểm tra phần mềm

Tại sao phải kiểm tra ?

- Xét ví dụ đơn giản
 - Xác định dạng tam giác từ độ dài ba cạnh: a, b, c .



Tại sao phải kiểm tra ?

■ Xét ví dụ đơn giản

□ Xác định ngày kế tiếp

- Nhập vào 3 số mô tả ngày/tháng/năm hiện tại
- Đánh giá dữ liệu đầu vào và xác định ngày kế tiếp là ngày nào ?



Tại sao phải kiểm tra ?

- Xét ví dụ đơn giản
 - ❑ Bài toán tính huê hồng bán hàng
 - ❑ Hệ thống ATM
 - ❑ Chuyển đổi tiền tệ
 - ❑ Điều khiển lau kính xe hơi
 - ❑ ...

Các hệ thống phần mềm

- Gia tăng tiện nghi của cuộc sống
 - Ngân hàng – ATM
 - Sản phẩm tiêu dùng – xe, thiết bị di động
 - ...
- Qua Thực tế
 - Không như kỳ vọng
 - Hoạt động không đúng – gây ra **nhiều vấn đề**
 - Tiền bạc
 - Thời gian
 - Tổn hại con người

Các hệ thống phần mềm

- “Lỗi phần mềm là chuyện hiển nhiên của cuộc sống. Chúng ta dù cố gắng đến mức nào thì thực tế là ngay cả những lập trình viên xuất sắc nhất cũng không có thể lúc nào cũng viết được những đoạn mã không có lỗi. Tính trung bình, ngay cả một lập trình viên loại tốt thì cũng có từ 1 đến 3 lỗi trên 100 dòng lệnh. Người ta ước lượng rằng việc kiểm tra để tìm ra các lỗi này chiếm phân nửa khối lượng công việc phải làm để có được một phần mềm hoạt động được”.

(Software Testing Techniques, Second Edition, by Boris Beizer, Van Nostrand Reinhold, 1990, ISBN 1850328803).

Nguyên nhân các khiếm khuyết

- Con người tạo ra lỗi (*error – mistake*)
- Hệ quả là xuất hiện khiếm khuyết (*sai lầm / sai sót - fault , rồi - bug*)
 - mã, hệ thống, phần mềm, tài liệu
 - đưa ra dư thừa – chúng ta đưa một vài thứ không chính xác vào mô tả yêu cầu phần mềm, ...
 - do bỏ sót, thiếu một số phần đáng ra phải có trong mô tả yêu cầu phần mềm, ...
- Khi mã thực thi, hệ thống thực hiện công việc sai sót → thực hiện không mong đợi (*failure – hỏng hóc*)
- Hệ quả không mong đợi (*Incident*)
- *Khiếm khuyết có thể gây ra hiện tượng không mong đợi*
 - *không phải tất cả khiếm khuyết đều như vậy*

Nguyên nhân các khiếm khuyết

- Hành vi con người
 - → Error
- Biến đổi logic chương trình
 - → Fault
- Hành vi của chương trình
 - → Failure
- Hệ quả của chuỗi quá trình không mong đợi
 - → Incident

Nguyên nhân các khiếm khuyết

- Khiếm khuyết có thể xảy ra bởi
 - Áp lực về thời gian
 - Mã phức tạp
 - Hạ tầng phức tạp
 - Thay đổi công nghệ
 - Tương tác nhiều hệ thống

Nguyên nhân các khiếm khuyết

- Thực thi không mong đợi có thể dẫn tới từ điều kiện môi trường
 - Sự bức xạ
 - Năng lượng
 - Trường điện từ
 - Ô nhiễm môi trường
 - ...
- Phần dẻo (*firmware*) bị ảnh hưởng
- Tác động tới SW do thay đổi điều kiện HW

Vai trò của việc kiểm tra

- Kiểm tra nghiêm ngặt
 - ❑ Khiếm khuyết tìm thấy được chỉnh sửa trước khi đưa ra sử dụng
 - ❑ Giảm thiểu rủi ro trong suốt quá trình thực hiện và phân phối đối với chất lượng phần mềm
- Có thể được **yêu cầu để đáp ứng** các
 - ❑ Yêu cầu trong hợp đồng
 - ❑ Yêu cầu của luật
 - ❑ Chuẩn công nghiệp, ...

Quan hệ Kiểm tra và chất lượng

- Với sự trợ giúp của việc kiểm tra, tìm các khiếm khuyết
 - Yêu cầu – đặc tính phần mềm
 - Chức năng
 - Phi chức năng
 - Tin cậy, khả dụng, hiệu quả, khả năng bảo trì, khả chuyển
- Tạo sự **tin tưởng chất lượng** của phần mềm

Quan hệ Kiểm tra và chất lượng

- Rút ra bài học từ các dự án thực hiện
- Hiểu “gốc rễ” của nguyên nhân các khiếm khuyết
 - Cải tiến quá trình với dự án mới
 - Tránh tái diễn sai lầm cũ
- Tích hợp như một phần của các hoạt động QA

Kiểm tra đến mức nào là đủ ?

- Việc quyết định **bao nhiêu là đủ** lên dựa vào bảng miêu tả các rủi ro
 - rủi ro kỹ thuật, nghiệp vụ, dự án
 - các ràng buộc về thời gian lẫn ngân sách.
- Việc kiểm tra lên cung cấp thông tin đủ “thẩm quyền” để **đảm bảo ra quyết định**
 - phiên bản xuất xưởng của phần mềm
 - hệ thống được kiểm tra
 - bước phát triển kế
 - bàn giao phần mềm

Kiểm tra phần mềm là gì ?

■ Các nhận thức sai lầm

- ❑ Chỉ bao gồm việc thực thi các mẫu thử
 - Đây chỉ là 1 phần công việc
- ❑ Quá trình chạy mẫu mà lỗi không xuất hiện
- ❑ Mục đích chỉ cho thấy chương trình thực hiện đúng các chức năng
- ❑ Quá trình tạo sự tin tưởng chương trình thực thi những gì chương trình “hỗ trợ”

Kiểm tra phần mềm là gì ?

- Vậy kiểm tra phần mềm là gì ?

Kiểm tra phần mềm là gì ?

Việc kiểm tra là quá trình thực thi một chương trình với mục đích là tìm ra lỗi.

(Glen Myers)

Kiểm tra phần mềm là gì ?

- Các hoạt động của kiểm tra bao gồm các công việc trước và sau khi “chạy” chương trình
 - Lập kế hoạch – điều khiển
 - Chọn điều kiện kiểm tra
 - Thiết kế kịch bản / Mẫu (kiểm) thử (*Test-Case*)
 - Kiểm tra kết quả
 - Đánh giá các tiêu chuẩn
 - Lập báo cáo quá trình Kiểm tra – hệ thống được kiểm tra

Kiểm tra phần mềm là gì ?

- Cả hai hướng kiểm tra động và tĩnh
 - Điều **hướng tới** cùng mục tiêu
 - Cung cấp thông tin ***cải tiến*** hệ thống lẫn quy trình

Kiểm tra phần mềm là gì ?

■ Mục tiêu

- Tìm khiếm khuyết
- Ngăn ngừa khiếm khuyết
- Chứng minh được phần mềm hoạt động đúng như đã thiết kế.
- Chứng minh được phần mềm đáp ứng yêu cầu của *user*

⇒ Góp phần **chứng minh chất lượng** của phần mềm.

⇒ Tăng tin tưởng mức chất lượng và thông tin cung cấp

Kiểm tra phần mềm là gì ?

- Khác biệt giữa Gỡ rối (*Debug*) và kiểm tra (*Testing*)
 - Kiểm tra:
 - cho thấy các trường hợp không mong đợi do khiếm khuyết phần mềm
 - Testers
 - Gỡ rối:
 - Hoạt động phát triển
 - Xác định nguyên nhân
 - Sửa chữa mã
 - Kiểm tra lại khiếm khuyết có được sửa đúng
 - Developers

Kiểm tra phần mềm là gì ?

- Kiểm tra sớm từ lúc thiết kế ngăn ngừa khiếm khuyết khi phát triển mã
- Rà soát lại tài liệu yêu cầu phần mềm cũng giúp ngăn ngừa các khiếm khuyết

Kiểm tra phần mềm là gì ?

- Các góc nhìn khác nhau về kiểm tra tạo ra các *mục tiêu khác nhau*
 - Kiểm tra phát triển
 - Kiểm tra thành phần, tích hợp hệ thống
 - Xác định nguyên nhân các hoạt động lỗi nhằm sửa khiếm khuyết
 - Kiểm tra chấp nhận
 - Chứng thực hệ thống làm việc như kỳ vọng/yêu cầu của khách hàng
 - Nhằm ước lượng chất lượng phần mềm → có thông tin xác định rủi ro phát hành hệ thống với thời gian đã xác định

Kiểm tra phần mềm là gì ?

- ❑ Kiểm tra bảo trì
 - Bao gồm các kiểm tra để đảm bảo không có những **khiếm khuyết mới** qua việc mở rộng phát triển khi có yêu cầu thay đổi
- ❑ Kiểm tra tác vụ (*operational Testing*)
 - Ước lượng **các đặc tính** của hệ thống: độ tin cậy – tính sẵn sàng

Kiểm tra phần mềm là gì ?

- Quá trình kiểm nghiệm phần mềm là **tốt** khi
 - Có khả năng ***tìm ra lỗi cao***.
 - Không dư thừa.
 - Biết chọn lọc
 - chỉ kiểm nghiệm những phần nào có khả năng tìm ra lỗi đặc trưng.
 - Không quá phức tạp cũng không quá đơn giản.

Các nguyên lý tổng quát

- Đúc kết thông qua quá trình phát triển và qua các hướng dẫn tổng quát nhất cho việc kiểm tra
- Có 7+ nguyên lý chính

Các nguyên lý tổng quát

- #1: phơi bày biểu hiện của khiếm khuyết
 - Cho thấy các khiếm khuyết đang tồn tại
 - Không thể chứng minh : không có khiếm khuyết
 - Giảm xác suất xuất hiện khiếm khuyết chưa phát hiện trong phần mềm
 - Ngay cả khi không có khiếm khuyết : vẫn không thể chứng minh nó chính xác.

Các nguyên lý tổng quát

- #2: Không thể vét cạn hết các trường hợp
 - Kiểm tra mọi thứ không khả thi
 - Tổ hợp của tất cả các trạng thái dữ liệu đầu vào và các điều kiện thực thi
 - Không thể **kiểm nghiệm triệt để** một phần mềm.
 - Ngoại trừ trường hợp không đáng kể
 - Thay vào đó:
 - Phân tích rủi ro.
 - Ưu tiên tập trung vào nỗ lực kiểm tra
 - Việc kiểm nghiệm nên hướng về yêu cầu của khách hàng

Các nguyên lý tổng quát

■ #3: Kiểm tra sớm

- Nên bắt đầu sớm nhất có thể trong chu kỳ phát triển
- Tập trung vào các mục tiêu đã xác định

Các nguyên lý tổng quát

- #4: Gom nhóm các khiếm khuyết
 - Thông thường một nhóm nhỏ các module phần mềm gây ra phần lớn các khiếm khuyết của hệ thống
 - Áp dụng nguyên lý Pareto: 80% lỗi có nguyên nhân từ 20% các *module*
 - ⇒ cô lập và kiểm tra những *module* khả nghi nhất.

Các nguyên lý tổng quát

- #5: Nghịch lý thuốc trừ sâu (*Pesticide paradox*)
 - Lặp lại cùng mẫu kiểm tra:
 - Không tìm thấy khiếm khuyết mới
 - Kịch bản cần rà soát và xem xét lại đều đặn
 - Kiểm tra mới-khác cần được viết cho các phần khác để có khả năng tìm nhiều khiếm khuyết

Các nguyên lý tổng quát

- #6: Phụ thuộc ngữ cảnh
 - Thực hiện khác nhau trong những ngữ cảnh khác nhau

- #7: Ảo tưởng “không lỗi” (*Absence-of-errors fallacy*)
 - Tìm và khắc phục khiếm khuyết là vô nghĩa nếu hệ thống xây dựng
 - Vô dụng
 - Không đáp ứng yêu cầu cần thiết
 - Không đáp ứng kỳ vọng người sử dụng

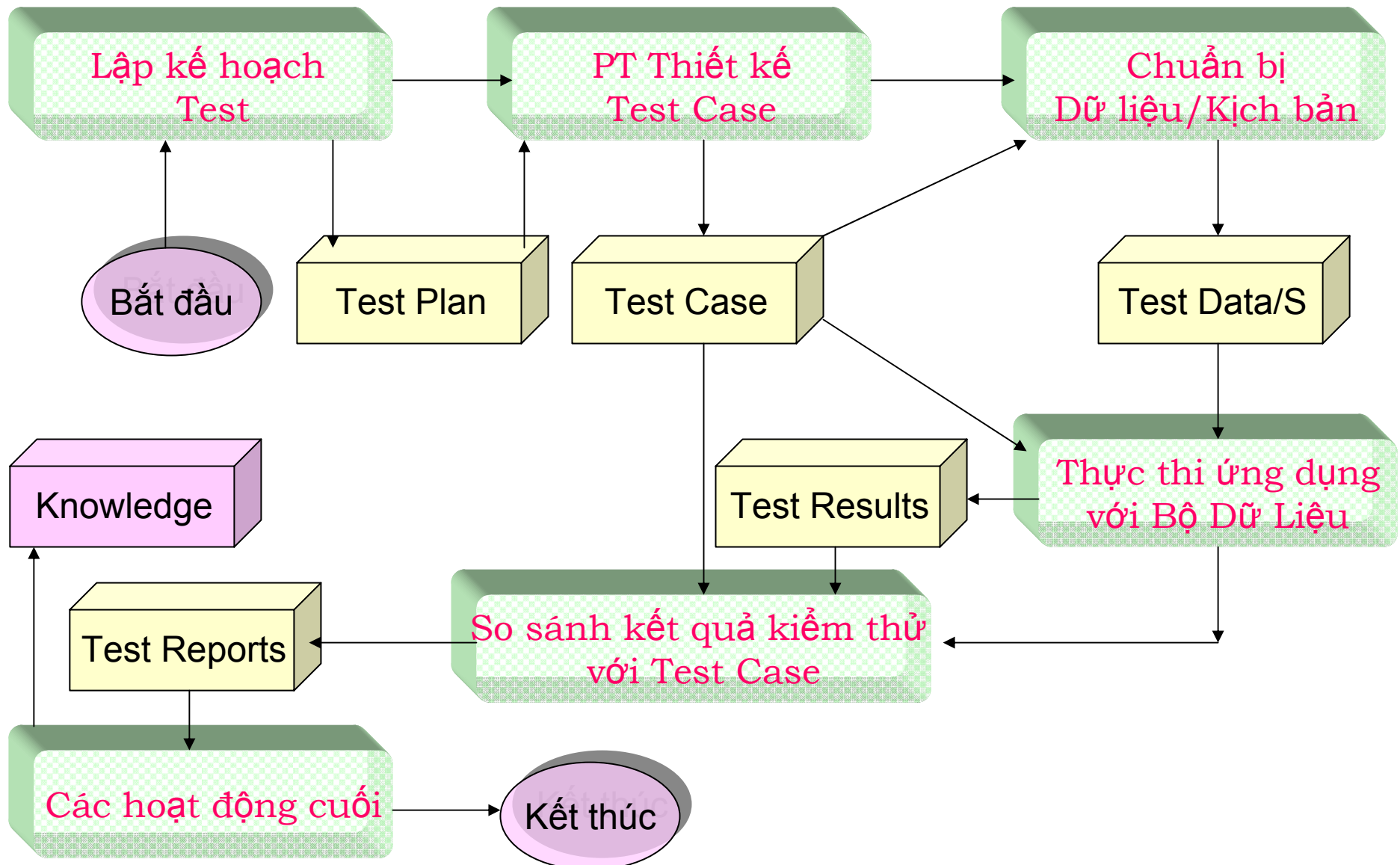
Các nguyên lý tổng quát

- Nên được hoạch định trước một thời gian dài.
- Nên tiến hành từ nhỏ đến lớn: bắt đầu từ những *module* riêng biệt rồi sau đó tích hợp các *module* lại.
- Nên được thực hiện bởi những đối tượng KHÔNG tham gia vào quá trình phát triển phần mềm.
- Nguyên tắc ngẫu nhiên:
 - dữ liệu và chức năng được chọn, tuy có chủ đích nhưng không phải xuất hiện theo thứ tự nhất định.
- Nguyên tắc "người sử dụng kém":
 - Người có trình độ thấp mức chấp nhận được dùng thử.
 - có thể gây các sự cố không lường trước được
- Nguyên tắc "kẻ phá hoại":
 - hệ thống rơi vào tay có trình độ nghiệp vụ cao, chủ ý phá hoại.

Quy trình kiểm tra cơ bản

- Lập kế hoạch và điều khiển
- Phân tích và thiết kế
- Hiện thực và việc thực thi
- Đánh giá các tiêu chuẩn và lập báo cáo
- Các hoạt động cuối

Quy trình kiểm tra cơ bản



Lập kế hoạch và điều khiển

■ Hoạt động

- Xác nhận các nhiệm vụ kiểm tra
- Định nghĩa mục tiêu
- Đặc tả các hoạt động kiểm tra đáp ứng mục tiêu và nhiệm vụ

■ Điều khiển kiểm tra

- Liên tục **đối chiếu** tiến trình thực sự với kế hoạch
- Lập báo cáo **tình trạng**
 - **Cả việc lệch hướng**
- Giữ các **hoạt động cần thiết** để đáp ứng nhiệm vụ/mục tiêu dự án
- Cần theo dõi thông qua thực hiện dự án

Phân tích và thiết kế

- Các mục tiêu kiểm tra chuyển đổi thành
 - Các điều kiện kiểm tra rõ ràng
 - Kịch bản
- Các nhiệm vụ chính
 - Xem xét cơ sở cho việc kiểm tra
 - yêu cầu, kiến trúc, thiết kế, giao tiếp, ...
 - Đánh giá khả năng kiểm tra của các cơ sở, mục tiêu
 - Xác định và lập ưu tiên các điều kiện kiểm tra
 - Thiết kế và lập ưu tiên các kịch bản
 - Xác định tập dữ liệu cần thiết hỗ trợ điều kiện kiểm tra và kịch bản
 - Thiết kế việc thiết lập môi trường; hạ tầng; công cụ

Hiện thực và việc thực thi

- Các thủ tục hoặc các đoạn mã script kiểm tra được đặc tả từ việc kết hợp các kịch bản kiểm tra theo thứ tự cụ thể
- Bao gồm các thông tin khác cần cho kiểm tra
 - Môi trường được thiết lập
 - Kiểm tra được thực hiện

Hiện thực và việc thực thi

■ Các nhiệm vụ chính

- ❑ Phát triển, hiện thực, lập thứ tự ưu tiên các kịch bản
- ❑ Phát triển và lập ưu tiên thủ tục kiểm tra, tạo tập dữ liệu; chuẩn bị kiểm tra và viết mã kiểm tra tự động
- ❑ Tạo các bộ kiểm thử từ các thủ tục kiểm tra để thực hiện hiệu quả
- ❑ Xác nhận môi trường được thiết lập đúng
- ❑ Thực thi kiểm tra bằng tay / công cụ
- ❑ Ghi nhận tác động của thực hiện kiểm tra; ghi nhận định danh và phiên bản

Hiện thực và việc thực thi

- ❑ Đối sánh kết quả thực và kỳ vọng
- ❑ Lập báo cáo phần thiếu nhất quán, phân tích xác định nguyên nhân
- ❑ Lặp lại hoạt động kiểm tra như một hệ quả tất yếu với các phần thiếu nhất quán ...
 - Xác nhận lỗi đã sửa
 - Đã sửa lỗi và không phát sinh khiếm khuyết mới

Đánh giá - Lập báo cáo

- Thực hiện khi thực hiện kiểm tra được ước lượng thấy tương phản với mục tiêu đề ra
- Cần thực hiện qua **mỗi mức kiểm thử**
- Các nhiệm vụ chính
 - Kiểm tra phần ghi nhận *tương phản* với tiêu chuẩn đề ra trong bản kế hoạch
 - Ước định
 - nếu cần nhiều kiểm tra
 - Tiêu chuẩn cần được thay đổi
 - Viết báo cáo tổng kết kiểm thử

Hoạt động kết thúc kiểm tra

- Tập hợp dữ liệu từ các hoạt động kiểm tra thành công để củng cố
 - Kinh nghiệm
 - Testware
 - Sự kiện
 - Trị số
- Công việc chính
 - Kiểm tra công việc đã phân theo kế hoạch
 - Testware hoàn tất , cất giữ; môi trường kiểm tra, hạ tầng, ... được sử dụng lại sau
 - Bàn giao testware đến tổ chức bảo trì
 - Phân tích bài học học được

Triết lý của việc kiểm tra PM

- Cách suy nghĩ khi **kiểm thử-rà soát** khác so với khi phát triển phần mềm
 - Người phát triển có suy nghĩ đúng → có thể tự kiểm tra mã mình viết
 - Việc tách riêng cho Tester là đặc thù giúp hỗ trợ tập trung và cung cấp lợi ích gia tăng
- Ở mức độ độ nào đó, thường **hiệu quả trong tìm kiếm khiếm khuyết** và hoạt động không mong đợi

Triết lý của việc kiểm tra PM

■ Các mức độ lập

- ❑ Thiết kế kiểm thử bởi người viết mã
- ❑ Thiết kế kiểm thử bởi người khác
- ❑ Thiết kế kiểm thử bởi người từ tổ chức khác
- ❑ Thiết kế kiểm thử bởi nhóm người từ tổ chức khác

Triết lý của việc kiểm tra PM

- Hướng theo mục tiêu
- Giống theo kế hoạch
 - Mục tiêu lập bởi người quản lý
 - Ký kết
- Xác định lỗi thông qua kiểm thử
 - Hoạt động phê bình tương phản tác giả, tác phẩm

Triết lý của việc kiểm tra PM

- Xảy ra lỗi , khiếm khuyết, hoạt động không mong đợi
 - Truyền thông theo cách tích cực – xây dựng
- Cần kỹ năng giao tiếp, quan hệ tốt
 - Bắt đầu với sự cộng tác hơn là đối đầu
 - Truyền thông những kết quả lỗi theo ý trung lập, tập trung sự kiện, không chỉ trích
 - Cố gắng hiểu người khác cảm nhận ra sao và tại sao họ phản ứng lại
 - Xác nhận lại những gì đã hiểu và chưa hiểu

Bài tập - đọc hiểu

Đọc thêm

- [1]. Chapter 01
- [1]. Chapter 02
- [2]. Chapter 17
- [4]. Chapter 01
- [5]. Chapter 01
- *[3]. Chapter 01*

Q/A