

Môn học

**KIỂM THỬ PHẦN MỀM**

---

**Chương VI**

**Quản lý kiểm thử**

# Nội dung

---

1. Tổ chức kiểm thử
2. Quản lý cấu hình
3. Lập kế hoạch và Ước lượng kiểm thử

## VI.1 Tổ chức kiểm thử

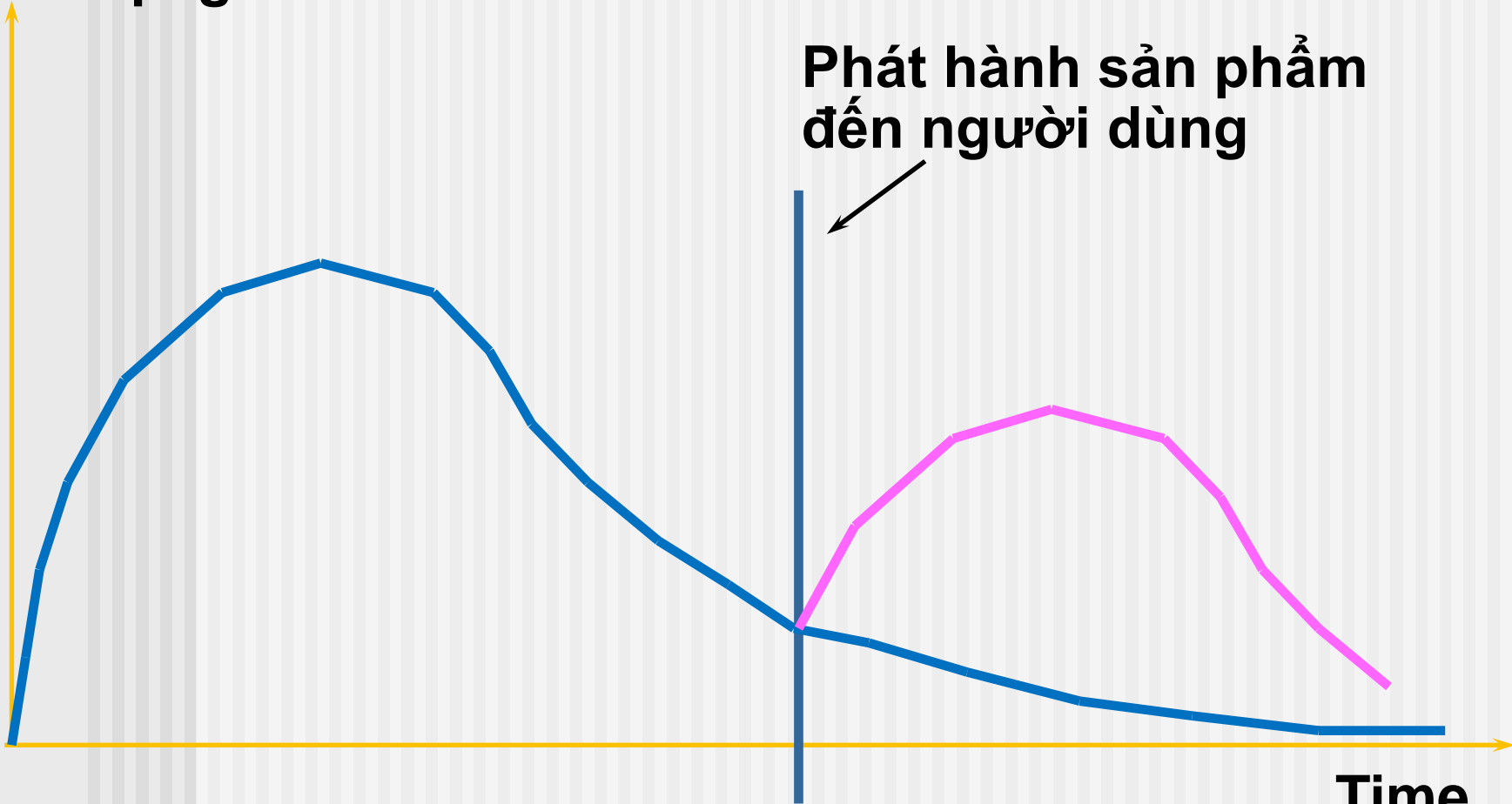
# Tầm quan trọng của tính độc lập trong kiểm thử

Số lượng lỗi

Phát hành sản phẩm đến người dùng

Time

Kiểm thử phần mềm



# Cấu trúc tổ chức kiểm thử

---

1. Trách nhiệm của mỗi developer
2. Trách nhiệm của cả nhóm phát triển(Development team )
3. Tester nằm trong nhóm phát triển
4. Nhóm testers riêng biệt
5. Cố vấn cho kiểm thử
6. Nhóm testers thuộc công ty thứ 3

# Mỗi developer thực hiện kiểm thử

## Ưu điểm

- Hiểu rõ code
- Tìm được các lỗi kĩ thuật mà tester có thể bỏ lỡ
- Tìm lỗi và sửa lỗi ít tốn ngân sách

## Nhược điểm

- Đánh giá chủ quan
- Khó phá hủy công việc của chính họ
- Xu hướng thấy được kết quả mong đợi hơn là kết quả thực tế

# Nhóm phát triển thực hiện kiểm thử

## Ưu điểm

- Có sự độc lập nhất định
- Tìm được các lỗi kỹ thuật mà tester có thể bỏ lỡ
- Thân thiện, dễ dàng giao tiếp

## Nhược điểm

- Thiếu kỹ năng test
- Bị áp lực công việc
- Quan điểm kỹ thuật, không phải quan điểm nghiệp vụ

# Tester nằm trong nhóm phát triển

## Ưu điểm

- Tester chỉ có trách nhiệm kiểm thử
- Độc lập trong kiểm thử
- Tester là thành viên của nhóm → có chung 1 mục đích

## Nhược điểm

- Ít được tôn trọng
- Làm việc một mình
- Quan điểm đưa ra đơn lẻ



# Đội ngũ tester độc lập

## Ưu điểm

- Gồm những tester có kinh nghiệm, chỉ làm nhiệm vụ kiểm thử
- Kiểm thử khách quan hơn và nhất quán hơn
- Thường có ngân sách riêng để đầu tư cho đào tạo tester, công cụ/thiết bị kiểm thử

## Nhược điểm

- Phụ thuộc vào đội ngũ tester → test thiếu
- Cô lập/đối đầu với nhóm phát triển

# Chuyên gia tư vấn kiểm thử nội bộ

## Ưu điểm

- Gồm các chuyên gia trong lĩnh vực kiểm thử → hỗ trợ giúp cải tiến quá trình kiểm thử
- Vạch kế hoạch, ước lượng, kiểm soát quá trình kiểm thử tốt hơn

## Nhược điểm

- Mức chuyên gia đã đủ chưa??
- Lời khuyên, hướng dẫn chỉ ảnh hưởng đến thử tục test, chứ không quyết định hoàn toàn
- Một số chuyên gia cũng phải thực thi test

# Nhóm testers thuộc công ty thứ 3

## Ưu điểm

- Gồm các chuyên gia trong lĩnh vực kiểm thử
- Độc lập với các chính sách nội bộ

## Nhược điểm

- Chi phí đắt??
- Thiếu kiến thức về sản phẩm và công ty
- Kinh nghiệm kiểm thử có được trong dự án thuộc về công ty thứ 3

# Chọn lựa phù hợp?

---

- Miền ứng dụng, mức độ rủi ro, kích cỡ, độ phức tạp của dự án
- Tính độc lập trong các mức kiểm thử
  - Component testing: dev, hoặc dev team
  - Integration testing: ít ai thực hiện tốt
  - System testing: đội ngũ tester độc lập
  - Acceptance testing: user

# Những kĩ năng cần thiết trong đội ngũ kiểm thử

---

- Chuyên gia kỹ thuật
- Chuyên gia về kiểm thử tự động
- Chuyên gia database
- Kỹ năng business
- Hiểu biết tốt về tính khả dụng
- Chuyên gia về môi trường test
- Người quản lý test

# Vai trò của Test leader

---

- Lập kế hoạch, giám sát và kiểm soát các hoạt động kiểm thử
- Lập các mục tiêu, tổ chức chính sách, chiến lược kiểm thử
- Ước tính test cần thực hiện và đàm phán với quản lý để có được các nguồn lực cần thiết
- Nhận biết thời điểm cần test tự động, chọn lựa tools, tổ chức training
- Tham khảo ý kiến với nhóm khác

# Vai trò của Test leader

---

- Hướng dẫn, giám sát việc phân tích, thiết kế thực thi các TCs, thủ tục kiểm thử, kịch bản kiểm thử
- Đảm bảo quản lý cấu hình testware
- Đảm bảo thực hiện môi trường test
- Lập lịch thực hiện kiểm thử
- Theo dõi, đo lường, kiểm soát và báo cáo về tiến độ kiểm thử, tình trạng chất lượng sản phẩm và kết quả kiểm thử
- Viết báo cáo tóm tắt về trạng thái test

# Vai trò của tester

---

- Rà soát và góp ý cho kế hoạch kiểm thử
- Phân tích, đánh giá các yêu cầu của người dùng, các đặc tả thiết kế.
- Xác định điều kiện test, thiết kế TCs, thủ tục test, hỗ trợ test tự động
- Cài đặt môi trường test



# Vai trò của tester

---

- Thực hiện kiểm thử trên tất cả các mức test
- Thực thi, ghi nhận, đánh giá kết quả test, lập tài liệu báo cáo lỗi
- Rà soát đặc tả test, các báo cáo lỗi, kết quả test.
- Theo dõi quá trình kiểm thử, môi trường test sử dụng các tools

## VI.2 Quản lý cấu hình

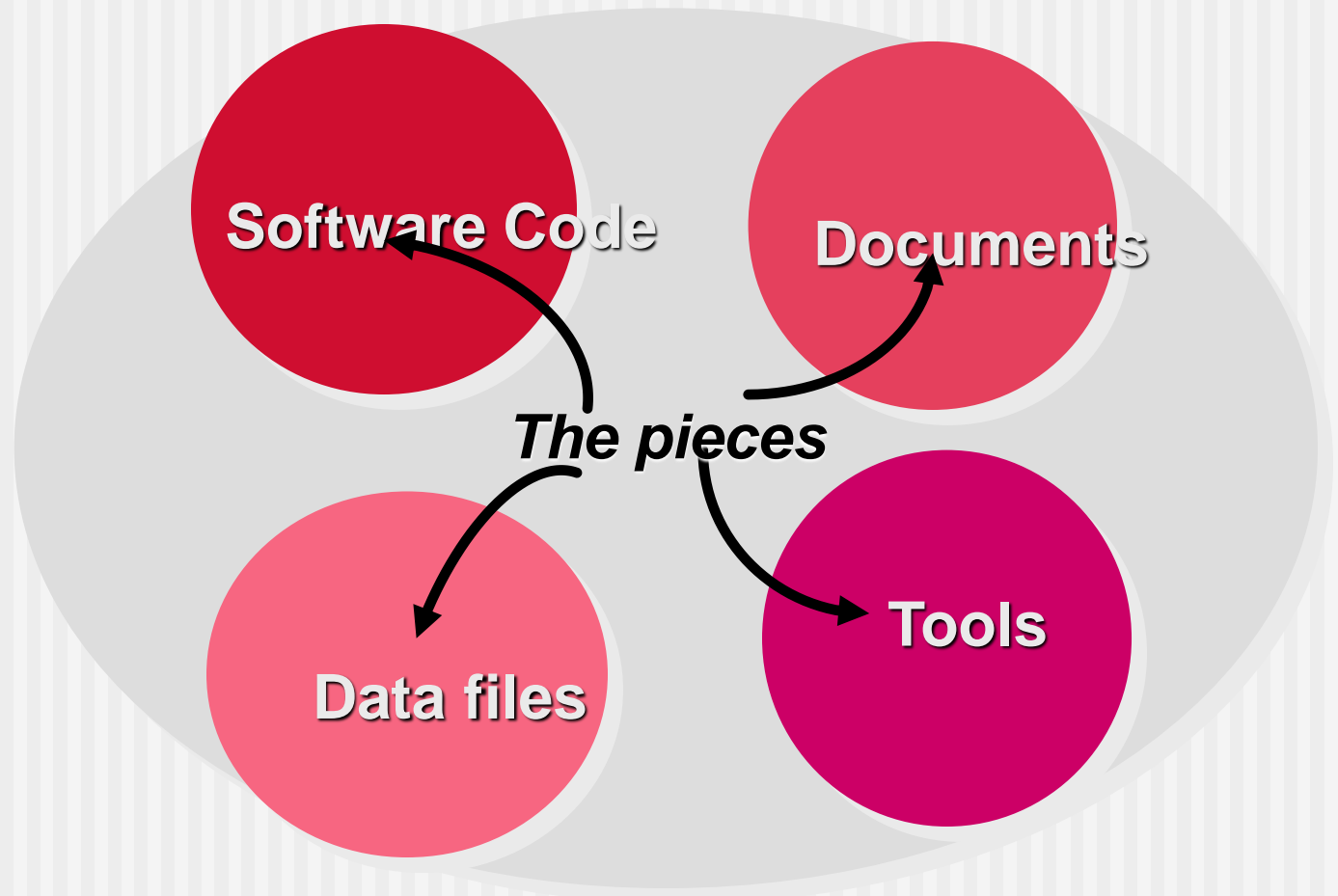
# Vấn đề do quản lý cấu hình kém

---

- Không tái tạo lỗi được báo cáo bởi KH
- Không quay lại hệ thống con trước đó
- Một thay đổi có thể ghi đè lên cái khác
- Lỗi đã được fix có thể xuất hiện trở lại
- Code thay đổi thuộc version nào?
- ...

# Khoản mục cấu hình PM (Software Configuration Item – SCI)

---



# Các loại SCI phổ biến

---

- Documents
  - Software development plan (SDP)
  - Software requirements document (SRD)
  - Interface design specifications
  - Preliminary design document (PDD)
  - Database description
  - Software test plan (STP)
  - Software test report (STR)
  - User manuals
  - Maintenance manuals
  - Software change requests (SCRs)
  - ...

# Các loại SCI phổ biến

---

- Software Code
  - Source code
  - Object code
  - Prototype software
- Data files
  - Test cases: test scripts, test data, expected results
  - Actual results
- Software development tools
  - Compilers and debuggers
  - Application generators
  - CASE tools
  - Test tools
  - ...

# Cấu hình phần mềm

---

- Là tập hợp các SCIs tạo thành phần mềm
- Sắp xếp các SCIs theo phiên bản, tính chất, đặc trưng

# Ví dụ: Cấu hình phần mềm

SCI version	Release and release date	
	PMT Version 6.0 January 6, 2002 SCI version in the release	PMT Version 7.0 January 22, 2003 SCI version in the release
SRD	Ver. 1	Ver. 1
CDD	Ver. 3	Ver. 4
STP	Ver. 3	Ver. 4
SIP	Ver. 2	Ver. 2
VDD	Ver. 6	Ver. 7
Code Module 1	Ver. 3	Ver. 5
Code Module 2	Ver. 8	Ver. 8
Code Module 3	Ver. 2	Ver. 2
Test cases file	Ver. 3	Ver. 4
CL compiler	Ver. 5	Ver. 7
Software user manual	Ver. 6	Ver. 7



# Định nghĩa Quản lý cấu hình

---

- Là quá trình xác định các khoản mục cấu hình trong một hệ thống
- Kiểm soát việc phát hành và thay đổi của các khoản mục này cũng như các phiên bản của nó trong suốt vòng đời hệ thống
- Ghi nhận và báo cáo tình trạng các khoản mục cấu hình và yêu cầu thay đổi
- Kiểm tra tính đầy đủ và đúng đắn của các khoản mục cấu hình.

# Thủ tục quản lý cấu hình

---

- Gồm:
  - Định danh SCI
  - Kiểm soát thay đổi
  - Kiểm soát phiên bản
  - Kiểm toán cấu hình
  - Báo cáo trạng thái

# Định danh SCI

---

- ❑ Mỗi khoản mục cần được **định danh**: mô tả tên, đánh số, hoặc đánh dấu đặc trưng
- ❑ Xây dựng sơ đồ phân cấp định danh của khoản mục để thể hiện mối quan hệ giữa chúng

# Kiểm soát thay đổi

---

- ❑ Thay đổi không được kiểm soát sẽ dẫn đến hỗn loạn
- Cần tạo ra sự thay đổi cho phù hợp với hệ thống phần mềm
- Cần một tập các công cụ và thủ tục để kiểm soát thay đổi

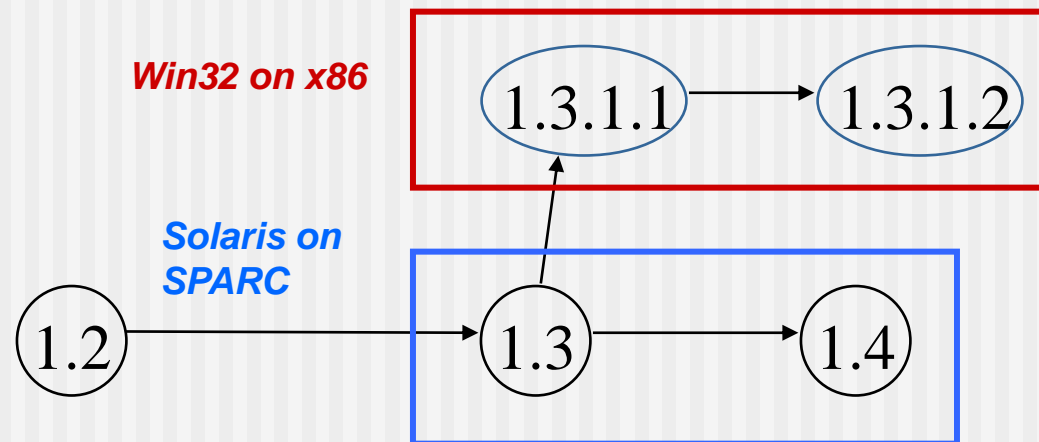
# Versions/variants/releases của hệ thống

---

- ❑ **Version**: là một thể hiện của hệ thống. Phân biệt các phiên bản của hệ thống dựa vào:
  - Chức năng khác nhau
  - Tính thực thi được nâng cao
  - Lỗi của phần mềm đã được khắc phục
- ❑ **Variant**: tương đương với **version** về chức năng, nhưng được thiết kế cho những thiết lập khác nhau

# Versions/variants/releases của hệ thống

- ❑ **Release:** là một phiên bản được phân phối tới khách hàng



# Kiểm soát phiên bản

---

- ❑ Kết hợp các thủ tục và công cụ để quản lí các phiên bản khác nhau của SCI
- ❑ Hệ thống kiểm soát phiên bản có 4 tính năng chính:
  - Repository(project database)
  - Quản lí phiên bản
  - Tạo phiên bản mới của PM
  - Theo dõi vấn đề (lỗi) liên quan đến SCI
- ❑ Công cụ hỗ trợ cho việc kiểm soát phiên bản: Visual Source Safe của Microsoft, ClearCase của Rational, CVS

# Kiểm toán cấu hình

---

- ❑ Đảm bảo tất cả các thay đổi thực sự được thực thi
- ❑ Thực hiện rà soát và kiểm tra để đánh giá sản phẩm hoặc quy trình sai lệch với đặc tả, chuẩn, yêu cầu hợp đồng,...
- ❑ Bộ phận QA hoặc customer sẽ thực hiện kiểm toán



# Báo cáo trạng thái

---

- Bao gồm:
  - Ghi nhận và báo cáo tình trạng của các khoản mục cấu hình
  - Tình trạng của các thay đổi được đề xuất, được kiểm duyệt
  - Ước tình tài nguyên để hoàn thành 1 tác vụ
  - Các khiếm khuyết đã được xác định bởi kiểm toán cấu hình
  - ...

# Quản lý cấu hình và kiểm thử

---

- QLCH cho phép tester quản lý các testware và kết quả test
  - Testware là những sản phẩm được tạo ra trong quá trình kiểm thử: test plan, test cases, test scripts,...
- QLCH cho phép định vị được cái gì đang được test, quản lý các lỗi được tìm thấy tương ứng với thành phần nào, thuộc phiên bản nào

## VI.3 Lập kế hoạch và ước lượng kiểm thử

# Kế hoạch kiểm thử (Test plan)

---

## IEEE 829 STANDARD TEST PLAN TEMPLATE

Test plan identifier

Test deliverables

Introduction

Test tasks

Test items

Environmental needs

Features to be tested

Responsibilities

Features not to be tested

Staffing and training needs

Approach

Schedule

Item pass/fail criteria

Risks and contingencies

Suspension and resumption criteria

Approvals

# Ước lượng kiểm thử

---

- Ước lượng:
  - Nỗ lực kiểm thử (effort)
  - Thời gian kiểm thử (schedule)
- Có 2 cách tiếp cận:
  - Dựa trên kinh nghiệm của các dự án trước
  - Ước lượng công việc của từng cá nhân(**Work Breakdown Structures**)
    - Có thể sử dụng công cụ Microsoft Project

# Work Breakdown Structures

---

Điểm giống nhau của ước lượng kiểm với ước lượng các công việc khác:

- Xác định các tác vụ cụ thể
- Thời gian cho mỗi tác vụ
- Ai sẽ thực hiện tác vụ
- Thời gian bắt đầu và kết thúc 1 tác vụ
- Tài nguyên, kỹ năng
- Sự phụ thuộc của các tác vụ
  - Độ ưu tiên của tác vụ
  - Độ ưu tiên kỹ thuật

# Work Breakdown Structures

---

## Điểm khác biệt trong ước lượng kiểm thử

- Sự phụ thuộc:
    - Kiểm thử không phải là một hoạt động độc lập
    - Lịch trình kiểm thử có thể bị thay đổi
    - Phụ thuộc vào môi trường test
  - Vòng lặp kiểm thử
    - kiểm thử sẽ tìm ra lỗi
    - lỗi cần phải được fix
    - fix xong phải test lại
- Vòng lặp này thực hiện bao nhiêu lần?

# Ước lượng vòng lặp kiểm thử

---

- Dựa trên số vòng lặp của phiên bản release trước
- Số lượng lỗi dự kiến
  - Có thể dự đoán từ hiệu quả kiểm thử trước đó và các lỗi được tìm thấy trước đó
  - % lỗi được tìm thấy trong mỗi lần lặp (lỗi lồng nhau)
- Thời gian viết báo cáo lỗi
- Thời gian đợi sửa lỗi



# Các yếu tố ảnh hưởng đến nỗ lực kiểm thử

---

- Yếu tố về Sản phẩm
  - Tài liệu đầy đủ → tester dễ dàng hơn
  - Các đặc trưng chất lượng phi chức năng: tính khả dụng, độ tin cậy,... → tiêu tốn nhiều thời gian
  - Độ phức tạp của sản phẩm
  - Kích thước của sản phẩm
- Yếu tố về quy trình
  - Tính sẵn có của các công cụ test
  - Mô hình phát triển PM