

Nhập môn kiểm thử phần mềm

Chương 8 - Kiểm thử tĩnh



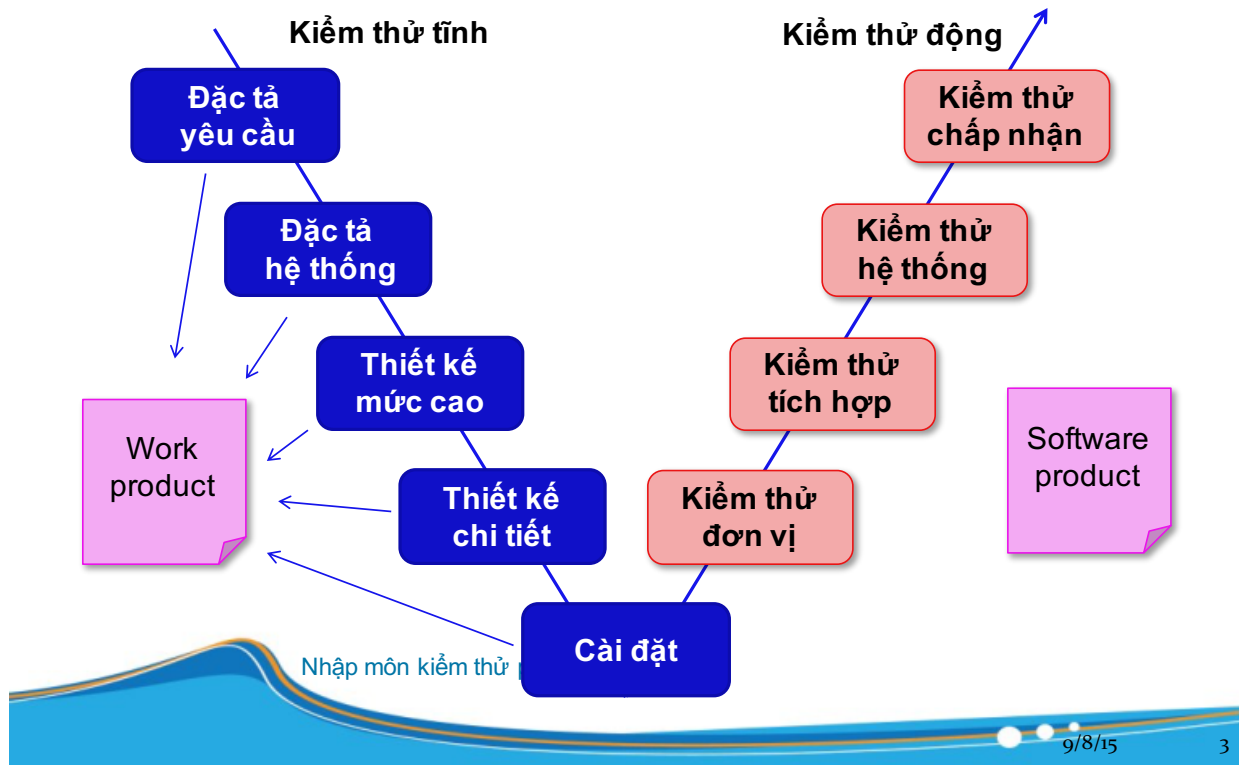
KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN



Nội dung

- ☐ Kiểm thử tĩnh
- ☐ Rà soát – Reviews
- ☐ Phân tích tĩnh – Static Analysis

Kiểm thử tĩnh và kiểm thử động



Kiểm thử tĩnh và kiểm thử động

Kiểm thử tĩnh

- ❑ Không cần chạy chương trình
- ❑ Kiểm tra work product để tìm sai sót (errors, defects)
- ❑ Phòng ngừa lỗi

Kiểm thử động

- ❑ Biên dịch và chạy chương trình
- ❑ Sử dụng sản phẩm để tìm lỗi (failures)
- ❑ Phát hiện lỗi

Lợi ích của kiểm tra tĩnh

- ☐ Nguyên lý kiểm thử sớm nhất có thể
- ☐ Đối với phát triển
 - ☐ Tăng năng suất, giảm thời gian phát triển
- ☐ Đối với kiểm thử
 - ☐ Giảm chi phí, thời gian cho kiểm thử
- ☐ Đối với nhân lực
 - ☐ Tăng hiệu quả giao tiếp, hiểu rõ yêu cầu



Đối tượng kiểm tra tĩnh

- ☐ Mọi tài nguyên có thể thẩm tra được
 - ☐ Đặc tả yêu cầu
 - ☐ Tài liệu thiết kế
 - ☐ Mã nguồn
 - ☐ Lịch trình
 - ☐ Test plans, test cases, test reports





Mục đích của kiểm thử tĩnh

- ☐ Sự sai lệch với các chuẩn
- ☐ Sai sót trong yêu cầu
- ☐ Sai sót trong thiết kế
- ☐ Thiếu sót khả năng nâng cấp
- ☐ Sai sót trong đặc tả giao diện



Nội dung

- ☐ Kiểm thử tĩnh
- ☐ **Rà soát – Reviews**
- ☐ Phân tích tĩnh – Static Analysis



Rà soát - Reviews

- Rà soát là hoạt động mà work product được phân chia cho reviewer để kiểm tra và phản hồi
- Các kiểu rà soát
 - Không hình thức – Informal reviews
 - Trao đổi ngoài lề, kiểm tra cho bạn, lập trình theo cặp
 - Ngang hàng – Peer/Technical reviews
 - Lần bước – Walkthroughs
 - Thăm tra – Inspections

Semi formal reviews

Formal reviews



Quy trình rà soát hình thức (Formal Reviews)



- Quy trình
 1. Lập kế hoạch – Planning
 2. Khởi đầu – Kick-off
 3. Chuẩn bị – Preparation
 4. Họp xét duyệt – Review meeting
 5. Làm lại – Rework
 6. Tiếp tục – Follow-up

Lập kế hoạch, đánh giá, phân công, huấn luyện

Lặp lại mỗi lần rà soát một phần việc

Các hoạt động cá nhân: phân tích, cải tiến quy trình, đánh giá loại bỏ lỗi



Vai trò và trách nhiệm

- ☐ Điều phối (Moderator)
 - ☐ Chủ trì các cuộc họp
- ☐ Tác giả (Author)
 - ☐ Mô tả, giải thích, trả lời câu hỏi
- ☐ Thư ký (Scribe/Recorder)
 - ☐ Tập hợp , ghi lại thông tin
- ☐ Người rà soát (Reviewer/Inspector)
 - ☐ Tìm kiếm sai sót
- ☐ Người quản lý (Manager)
 - ☐ Lập kế hoạch, sắp xếp tài nguyên, huấn luyện



Lần bước – Walkthroughs

- ☐ Chủ trì bởi tác giả của tài liệu/mã nguồn
- ☐ Tài liệu/mã nguồn được giải thích từng bước nhằm
 - ☐ Đạt sự hiểu rõ/đồng thuận
 - ☐ Thu thập thông tin/phản hồi
- ☐ Người tham gia
 - ☐ Trong/ngoài nhóm
 - ☐ Trong/ngoài ngành





Ngang hàng – Technical reviews

- ☐ Chủ trì bởi người điều phối (không phải tác giả)
 - ☐ Đồng nghiệp (Peer)/Chuyên gia (Technical expert)
- ☐ Cuộc họp thảo luận nhằm đạt được sự đồng thuận về vấn đề kỹ thuật
- ☐ Người tham gia
 - ☐ Kiến trúc sư/Trưởng thiết kế/Thành viên chủ chốt



Thẩm tra – Inspections

- ☐ Loại rà soát hình thức nhất (Formal Reviews)
 - ☐ → Theo đúng qui trình các bước
- ☐ Chủ trì bởi người điều phối đã được đào tạo
- ☐ Tài liệu/mã nguồn được rà soát theo các luật (rules) và danh sách cần kiểm tra (checklists). Tất cả các sai sót được lập văn bản.
- ☐ Người tham gia
 - ☐ Phân rõ vai trò của người tham gia trong qui trình





Rà soát mã nguồn – Code reviews

- Quá trình rà soát tìm lỗi trên mã nguồn
- Các kỹ thuật
 - ▣ Lập trình cặp – Pair Programming
 - ▣ Lần bước – Walkthroughs
 - ▣ Thẩm tra – Inspections



Nhập môn kiểm thử phần mềm

9/8/15

15



Nội dung

- Kiểm thử tĩnh
- Rà soát – Reviews
- **Phân tích tĩnh – Static Analysis**

Nhập môn kiểm thử phần mềm

9/8/15

16

Static Analysis – Phân tích tĩnh

- Static Analysis = Automated Review
- Một dạng kiểm thử tĩnh tự động
 - ▣ Tự động hóa quá trình rà soát
 - ▣ Kiểm tra sự vi phạm chuẩn
 - ▣ Kiểm tra sai sót trong mã và mô hình
- Có hai kỹ thuật
 - ▣ Phân tích mã nguồn tĩnh – Static code analysis
 - ▣ Phân tích mô hình – Analysis of model



Ví dụ

- Đoạn code sau đây có vấn đề gì?

```
n := 0;  
read (x);  
n := 1;  
while x > y do  
  begin  
    read (y);  
    write( n*y);  
    x := x - n;  
  end;
```



Ví dụ

□ Đoạn code sau đây có vấn đề gì?

```
int foo (int iX, int iY)
{
    return iX + iY;
    int iZ = iX*iY;
}
```

```
int a = 0;
while (a < 10)
{
    printf("%d\n", a);
    if (a = 5)
    {
        printf("a equals 5!\n");
    }
    a++;
}
return 0;
```

Nhập môn kiểm thử phần mềm

9/8/15

19

Phân tích mã nguồn tĩnh

□ Lỗi phát hiện

- Tham chiếu đến biến chưa gán giá trị
- Biến chưa bao giờ được sử dụng
- Mã “chết”
- Vi phạm chuẩn lập trình
- Yếu điểm bảo mật
- Vi phạm giá trị biên
- ...

Nhập môn kiểm thử phần mềm

9/8/15

20

Phân tích mã nguồn tĩnh

- Một số hướng tiếp cận
 - ▣ Chuẩn lập trình – Programming standard
 - ▣ Phân tích luồng dữ liệu – Data flow analysis
 - ▣ Phân tích cấu trúc dữ liệu – Data structure analysis
 - ▣ Phân tích luồng điều khiển – Control flow analysis
 - ▣ Độ đo mã nguồn – Code metrics
 - ▣ Độ phức tạp Cyclomatic – Cyclomatic complexity



Phân tích mã nguồn tĩnh

- Chuẩn lập trình – Programming standard
 - ▣ Chứa năng quan trọng nhất của SCA
 - ▣ Kiểm tra sự vi phạm các chuẩn lập trình
 - ▣ Thường tích hợp vào IDE



Phân tích mã nguồn tĩnh

- Phân tích luồng dữ liệu – Data flow analysis
 - ▣ Phân tích luồng dữ liệu của một biến (define, use)
 - ▣ Các sai sót được phát hiện
 - Sử dụng một biến chưa được gán giá trị
 - Biến không được sử dụng

```

n := 0;
read (x);
n := 1;
while x > y do
begin
  read (y);
  write( n*y);
  x := x - n;
end;

```

n được định nghĩa lại mà không được dùng
 y được sử dụng trước khi được định nghĩa

Nhập môn kiểm thử phần mềm

9/8/15

23

Phân tích mã nguồn tĩnh

- Phân tích cấu trúc dữ liệu – Data structure analysis
 - ▣ Liên quan đến cách tổ chức dữ liệu, độc lập với chương trình
 - ▣ Cung cấp thông tin về độ khó chương trình để giải quyết dữ liệu
 - ▣ Chương trình phức tạp bởi cấu trúc dữ liệu phức tạp

Nhập môn kiểm thử phần mềm

9/8/15

24

Phân tích mã nguồn tĩnh

- Phân tích luồng điều kiện – Control flow analysis
 - ▣ Phân tích thứ tự thực hiện của các câu lệnh
 - ▣ Các sai sót được phát hiện
 - Mã “chết”
 - Lặp không dừng
 - Nhảy đến một nhãn không xác định

```
int foo (int iX, int iY)
{
    return iX + iY;
    int iZ = iX*iY;
}
```

```
int a = 0;
while (a < 10)
{
    printf("%d\n", a);
    if (a = 5)
    {
        printf("a equals 5!\n");
    }
    a++;
}
return 0;
```

Nhập môn kiểm thử phần mềm

9/8/15

25

Phân tích mã nguồn tĩnh

- Độ đo mã nguồn – Code metrics
 - ▣ Mã nguồn lớn và phức tạp như thế nào
 - ▣ Quyết định những bản thiết kế thay thế
 - ▣ Quyết định độ rủi ro
- Một số độ đo
 - ▣ LOC – line of code
 - ▣ Operands và Operators
 - ▣ Fan-in & Fan-out
 - ▣ Nesting levels
 - ▣ OO metrics

Nhập môn kiểm thử phần mềm

9/8/15

26

Phân tích mã nguồn tĩnh

- Độ phức tạp Cyclomatic – Cyclomatic complexity
 - ▣ Độ đo phức tạp của lưu đồ
 - ▣ Công thức đơn giản: độ phức tạp = số quyết định + 1



Thảo luận

