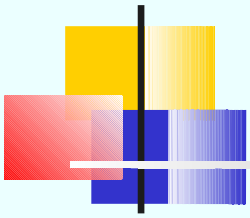
Công nghệ phần mềm

Kiểm thử phần mềm

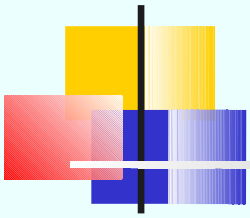
Giảng viên: ***TS. Nguyễn Mạnh Hùng***

Học viện Công nghệ Bưu chính Viễn thông (PTIT)



Nội dung tham khảo từ

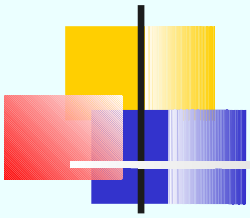
Stephen R. Schach. *Object-Oriented and Classical Software Engineering.* Seventh Edition, WCB/McGraw-Hill, 2007



Kiểm thử

"V & V":

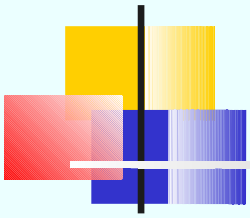
- Verification: kiểm thử xem một workflow hoàn thành có đúng và tương thích với các kết quả trước đấy hay không
- Validation: kiểm thử xem sản phẩm cuối cùng có đúng như yêu cầu của khách hàng hay không



Chất lượng phần mềm

- Không có khái niệm phần mềm « tuyệt hảo » một cách chung chung
- Chất lượng phần mềm thể hiện ở khả năng nó đáp ứng yêu cầu của khách hàng
- Người làm phần mềm phải có trách nhiệm đảm bảo sản phẩm tạo ra là chạy đúng

→ chất lượng phần mềm phải đảm bảo ngay từ đầu giai đoạn phát triển



Quản lí chất lượng PM (1)

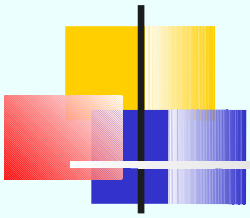
Nhóm SQA đảm bảo chất lượng PM:

- Đảm bảo rằng mọi thành viên phát triển PM đều làm việc với chất lượng cao:
  - Tại cuối mỗi workflow
  - Khi hoàn thành sản phẩm
- Ngoài ra, nhóm SQA còn phải đảm bảo rằng bản thân qui trình phát triển PM cũng đạt chất lượng cao (theo đúng chuẩn)

Quản lí chất lượng PM (2)

Quản lí nhóm SQA:

- Nhóm SQA nên độc lập về quản lí đối với nhóm phát triển PM
- Không nên để nhóm này có quyền cấp trên của nhóm kia
- Chỉ quản lí cấp cao của cả hai nhóm mới được quyết định:
  - Bàn giao sản phẩm đúng hẹn, nhưng còn lỗi
  - Test thêm và bàn giao trễ hẹn
- Việc quyết định còn phải tính đến ý kiến của khách hàng và khía cạnh công ty



Kiểm thử

Có 2 dạng kiểm thử:

- Kiểm thử các sản phẩm phi thực thi
- Kiểm thử các sản phẩm thực thi

Kiểm thử phi thực thi

Dựa trên nguyên tắc:

- Không ai tự kiểm thử sản phẩm của chính mình
- Dựa trên sức mạnh tổng hợp của nhóm

Có hai phương pháp:

- Walkthroughs: rà soát
- Inspections: kiểm tra, thẩm định

Walkthroughs (1)

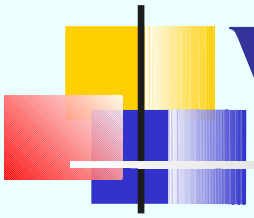
Tổ chức:

- Nhóm có 4 – 6 người
- Có đại diện của workflow hiện tại
- Có đại diện của workflow tiếp theo
- Có đại diện của nhóm SQA
- Người phụ trách nhóm walkthroughs là đại diện của nhóm SQA

Walkthroughs (2)

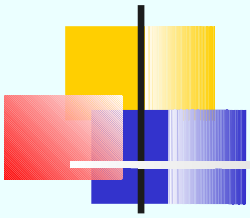
Hoạt động:

- Mỗi người phải chuẩn bị một danh sách:
  - Các vấn đề không hiểu
  - Các vấn đề có vẻ không đúng
- Phát hiện lỗi, nhưng không sửa lỗi:
  - Sửa lỗi bởi cả nhóm sẽ không đảm bảo chất lượng, mà chi phí lại cao
  - Không phải mọi vấn đề bị nêu lên đều thực sự là lỗi
  - Mỗi phiên walkthroughs không quá 2h → không đủ thời gian sửa lỗi



Walkthroughs (3)

- Walkthrough hướng tài liệu, chứ không hướng vào người tham gia
- Mục tiêu là phát hiện lỗi
- Kết quả walkthrough không bao giờ được dùng để đánh giá hiệu năng người thực hiện



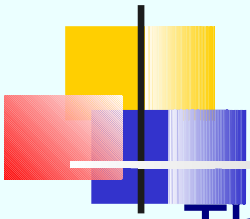
Inspections (1)

Bao gồm 4 thành viên:

- Người quản lí chung (moderator)
- 1 đại diện của workflow hiện tại
- 1 đại diện của workflow tiếp theo
- 1 đại diện của nhóm SQA

Có 3 vị trí khác nhau:

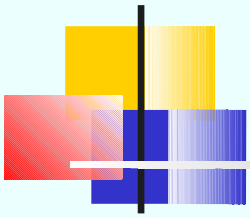
- Moderator
- Reader
- Recorder



Inspections (2)

Thực hiện theo 5 bước:

- Oveview: toàn bộ tài liệu cần thẩm định được phát đến các thành viên
- Preparation: so sánh lỗi với tỉ lệ lỗi của các dự án trước
- Inspection: rà soát lại tất cả các tài liệu (tương tự walkthroughs)
- Rework: mỗi thành viên sửa lỗi trên tài liệu mình rà soát
- Follow-up: trưởng nhóm kiểm duyệt lại lần cuối để các lỗi đều được sửa hoặc làm rõ các vấn đề gây khó hiểu

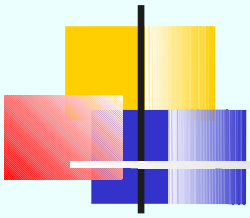


Inspections (3)

Thống kê lỗi:

- Thống kê lỗi theo mức độ, số lượng
  - Ví dụ: *phần lớn* hay *không nhiều*

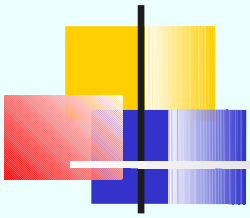
- Thống kê lỗi theo kiểu lỗi
  - Ví dụ: tham số tham chiếu và tham số thực tế không thống nhất



Inspections (4)

Thống kê lỗi:

- Lỗi được so sánh với tỉ lệ lỗi của các dự án trước
- Nếu số lỗi tăng đột biến thì phải nghĩ đến việc: thiết kế lại từ đầu
- Lỗi được bảo lưu cho workflow kế tiếp
- Kết quả thống kê không dùng để đánh giá hiệu năng người lao động



Kiểm thử thực thi (1)

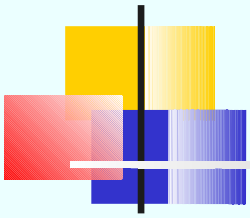
Thực tế:

- Thường người ta dành 50% chi phí dự án cho pha kiểm thử thực thi

→ Nhưng mà sản phẩm khi bàn giao vẫn chưa tin cậy được

Nguyên lí Dijkstra (1972):

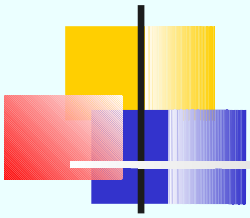
- Kiểm thử một chương trình rất dễ chỉ ra rằng chương trình có lỗi, nhưng rất khó để chứng tỏ rằng chương trình không còn lỗi



Kiểm thử thực thi (2)

Nội dung kiểm thử thực thi:

- Tính hữu dụng (utility)
- Tính tin cậy (reliability)
- Tính ổn định (robustness)
- Tính hiệu quả (performance)
- Tính đúng đắn (correctness)



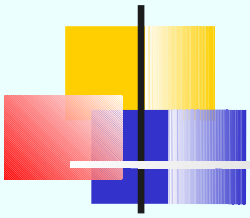
Kiểm thử thực thi (3)

Tính hữu dụng:

- Liệu sản phẩm có giúp ích cho khách hàng hay không?

Ví dụ:

- Dễ dàng sử dụng
- Chức năng cần thiết cho khách hàng
- Giá chấp nhận được



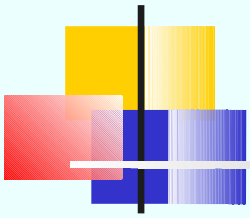
Kiểm thử thực thi (4)

Tính tin cậy:

- Liệu tần suất xuất hiện lỗi và mức độ lỗi có chấp nhận được hay không?

Thước đo:

- Thời gian trung bình giữa hai lần lỗi liên tiếp
- Thời gian trung bình để sửa một lỗi
- Chi phí và thời gian để khắc phục hậu quả của lỗi



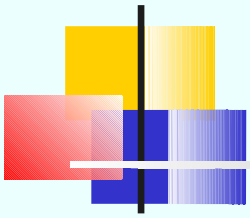
Kiểm thử thực thi (5)

Tính ổn định:

- Khả năng chống lỗi/sự cố của sản phẩm?

Thước đo:

- Pham vi của môi trường hoạt động ổn định của sản phẩm rộng hay hẹp
- Khả năng cho kết quả sai khi có đầu vào sai
- Tác động / hậu quả của việc cho đầu vào sai



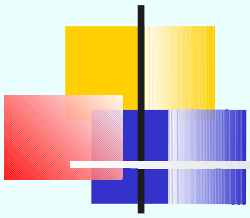
Kiểm thử thực thi (6)

Tính hiệu quả:

- Mức độ tiêu tốn tài nguyên thực thi (dung lượng bộ nhớ, thời gian) của sản phẩm?

Thước đo:

- Dung lượng bộ nhớ
- Thời gian thực hiện



Kiểm thử thực thi (7)

Tính đúng đắn:

- Các chức năng của sản phẩm có thực hiện đúng?

Thước đo:

- Unit test
- Black-box test

Kiểm thử thực thi (8)

Ví dụ kiểm thử đoạn đặc tả:

Input specification:	p : array of n integers, $n > 0$.
Output specification:	q : array of n integers such that $q[0] \leq q[1] \leq \cdots \leq q[n-1]$

Đoạn code sau sẽ thỏa mãn đặc tả trên:

```
void trickSort (int p[ ], int q[ ])
{
    int i;
    for (i = 0; i < n; i++)
        q[i] = 0;
}
```

Kiểm thử thực thi (9)

Ví dụ đặc tả sai:

Input specification:	p : array of n integers, $n > 0$.
Output specification:	q : array of n integers such that $q[0] \leq q[1] \leq \cdots \leq q[n-1]$

Đặc tả đúng:

Input specification:	p : array of n integers, $n > 0$.
Output specification:	q : array of n integers such that $q[0] \leq q[1] \leq \cdots \leq q[n-1]$ The elements of array q are a permutation of the elements of array p, which are unchanged.

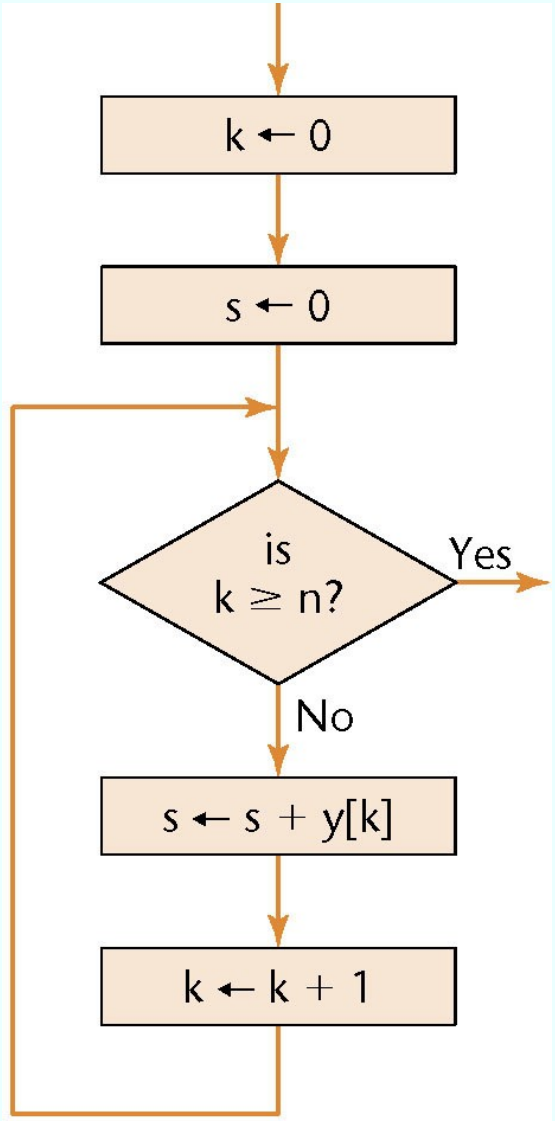
Kiểm thử thực thi (10)

Ví dụ làm sao chứng minh tính đúng đắn của code:

```
int k, s;
int y[n];
k = 0;
s = 0;
while (k < n)
{
    s = s + y[k];
    k = k + 1;
}
```

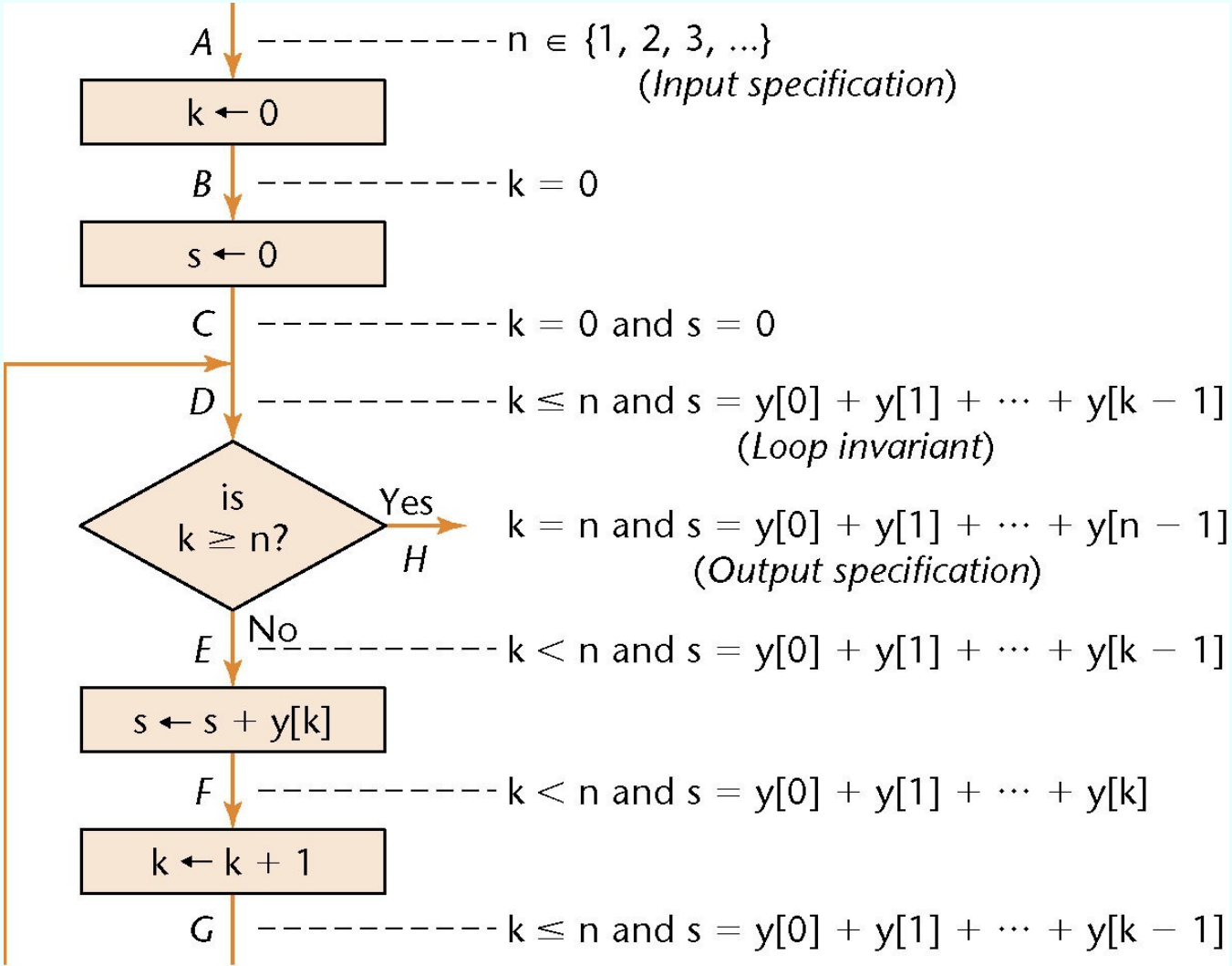
Kiểm thử thực thi (11)

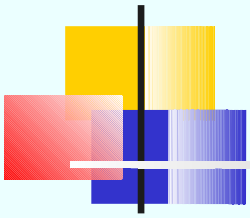
Vẽ sơ đồ khối đoạn code:



Kiểm thử thực thi (12)

Vẽ sơ đồ khối đoạn code:





Kiểm thử thực thi (13)

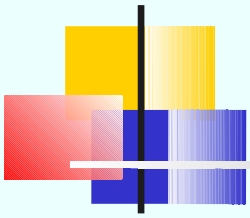
Ai nên kiểm thử thực thi:

- Lập trình là mang tính xây dựng: tạo ra chương trình
- Kiểm thử là mang tính phá hoại: phát hiện ra lỗi

→ Lập trình viên không nên kiểm thử code của chính mình

Giải pháp:

- Lập trình viên chỉ nên kiểm thử tài liệu
- SQA kiểm thử code
- Lập trình viên sẽ sửa code sau khi có kết quả test



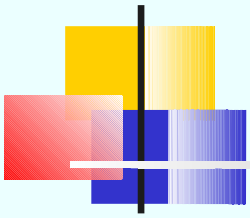
Kiểm thử thực thi (14)

Test case:

- Viết trước khi có code
- Lưu lại sau khi test

Cấu trúc một test case (kịch bản kiểm thử):

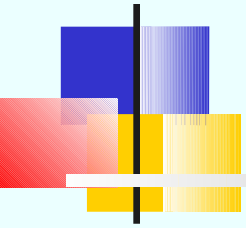
- Tập dữ liệu đầu vào
- Kịch bản các thao tác thực hiện
- Kết quả mong đợi



Kiểm thử thực thi (15)

Khi nào dừng kiểm thử:

- Chỉ dừng lại khi sản phẩm bị xóa bỏ khỏi kế hoạch phát triển cũng như ngoài thị trường!



Questions?