







TRƯỜNG ĐẠI HỌC NHA TRANG
KHOA CÔNG NGHỆ THÔNG TIN



KIỂM THỬ PHẦN MỀM

Giảng viên: Bùi Thị Hồng Minh
Bộ môn: Kỹ thuật Phần mềm
SĐT: 0983000604
Email: minhbth@ntu.edu.vn



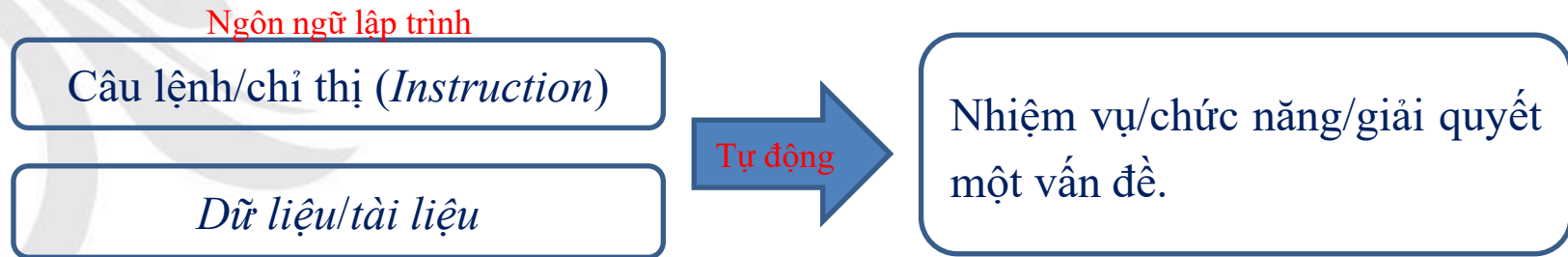
BÀI 1: TỔNG QUAN VỀ PHẦN MỀM VÀ CHẤT LƯỢNG PHẦN MỀM

- Phần mềm
- Quy trình phát triển phần mềm
- Một số mô hình phát triển phần mềm
- Chất lượng phần mềm

- Phần mềm
- Quy trình phát triển phần mềm
- Một số mô hình phát triển phần mềm
- Chất lượng phần mềm

1. Liệt kê các sản phẩm phần mềm được tạo ra trong quá trình phát triển và cho biết cụ thể sản phẩm đó là kết quả của giai đoạn nào của quá trình phát triển phần mềm? Sản phẩm nào cần được kiểm tra/kiểm thử?
2. Các loại kiểm thử đặc trưng nào được thực hiện với các Desktop App, Web App và Mobile App?
3. Cho biết sự khác biệt của giai đoạn kiểm thử phần mềm trong các mô hình Waterfall, V model và AgileScrum?

Khái niệm



Ứng dụng chạy trên máy tính, thiết bị điện tử, cung cấp các chức năng đáp ứng các yêu cầu, công việc của người sử dụng thông qua phần cứng.

Hệ thống xử lý thông tin gồm có 3 phần (giao tiếp, xử lý, lưu trữ) được xây dựng thành các chức năng theo yêu cầu của người sử dụng

Thành phần

- Chương trình máy tính: mã nguồn, mã máy.
- Cấu trúc dữ liệu: cách tổ chức lưu trữ dữ liệu trong/ngoài máy tính.
- Tài liệu liên quan: phát triển, thiết kế, tham khảo kỹ thuật, nghiệp vụ, hướng dẫn sử dụng, bảo trì...

PHẦN MỀM (Software)

Phân loại (*Môi trường thực thi*):

- Phần mềm ứng dụng chạy trên hệ điều hành (*Window App/Desktop App*).
- Phần mềm ứng dụng web (*Web App*)
- Phần mềm ứng dụng mobile (*Mobile App*)



Khái niệm

- Là tập hợp các giai đoạn/hoạt động tham gia vào việc sản xuất/tạo ra một sản phẩm phần mềm.
- **Software Development Life Cycle:** Thời gian tính từ lúc phần mềm được *sinh ra* ra đến khi *chết đi*.

QUY TRÌNH PHÁT TRIỂN PHẦN MỀM

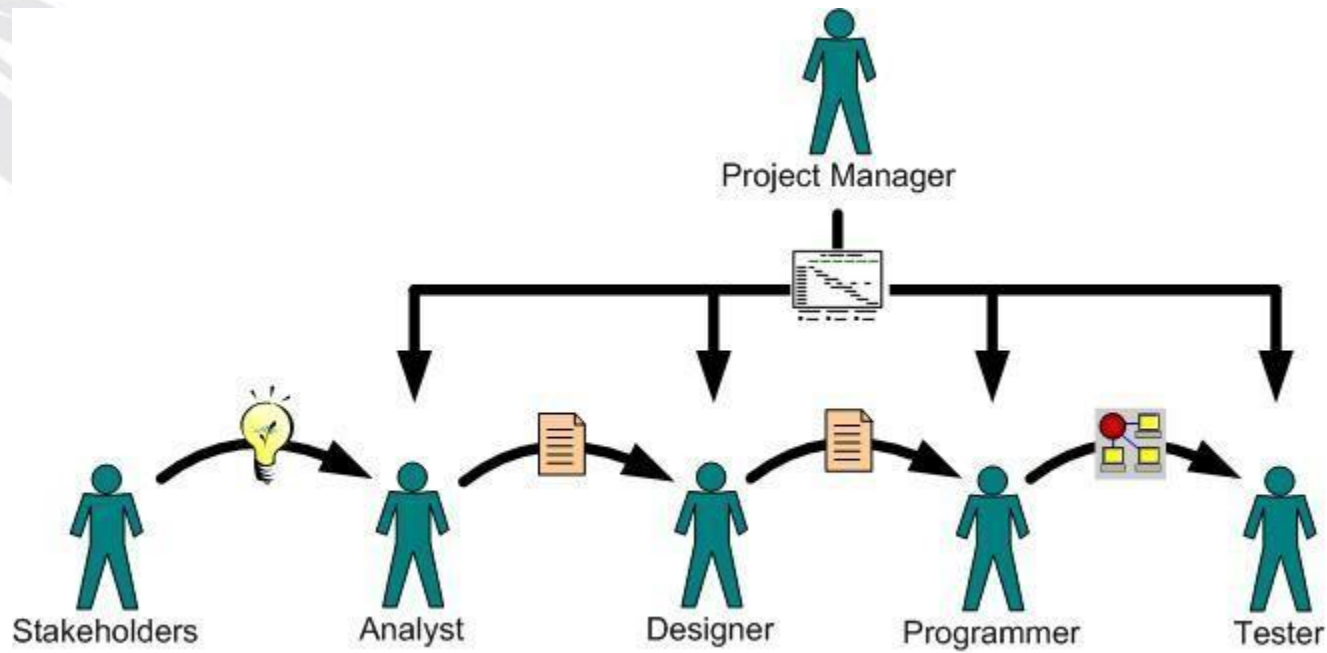
Các hoạt động

- Xác định yêu cầu (*Requirements*)
- Phân tích và đặc tả yêu cầu (*Analysis*)
- Thiết kế (*Design*)
- Thực hiện (*Development*)
- Kiểm thử (*Testing*)
- Bảo trì (*Maintenance*)
- Loại bỏ.



QUY TRÌNH PHÁT TRIỂN PHẦN MỀM

Organization



Các hoạt động

- Xác định yêu cầu (*Requirements –BA, PM – Project Plan*)
 - Gặp khách hàng để tìm hiểu yêu cầu, thu thập các biểu mẫu liên quan, tìm hiểu nghiệp vụ.
 - → Xác định, hiểu và làm rõ yêu cầu của khách hàng → xác định phạm vi của phần mềm.

Các hoạt động

- Phân tích và đặc tả yêu cầu (*Analysis – BA - SRS*)
 - Phân tích yêu cầu của khách hàng → định nghĩa các chức năng và dịch vụ cần thiết.
 - Mô tả các công việc cần triển khai cho từng chức năng đã phân tích.

→ Phần mềm sẽ làm gì?

Các hoạt động

- **Thiết kế (*Design –Designer– Design Docs*)**
 - *Thiết kế tổng quát: xác định kiến trúc tổng thể của hệ thống gồm các hệ thống con nào và mối quan hệ giữa chúng.*
 - *Thiết kế chi tiết: xác định cách thực hiện các hệ thống con và chức năng của hệ thống con.*

→ *Phần mềm được tổ chức và làm như thế nào?*

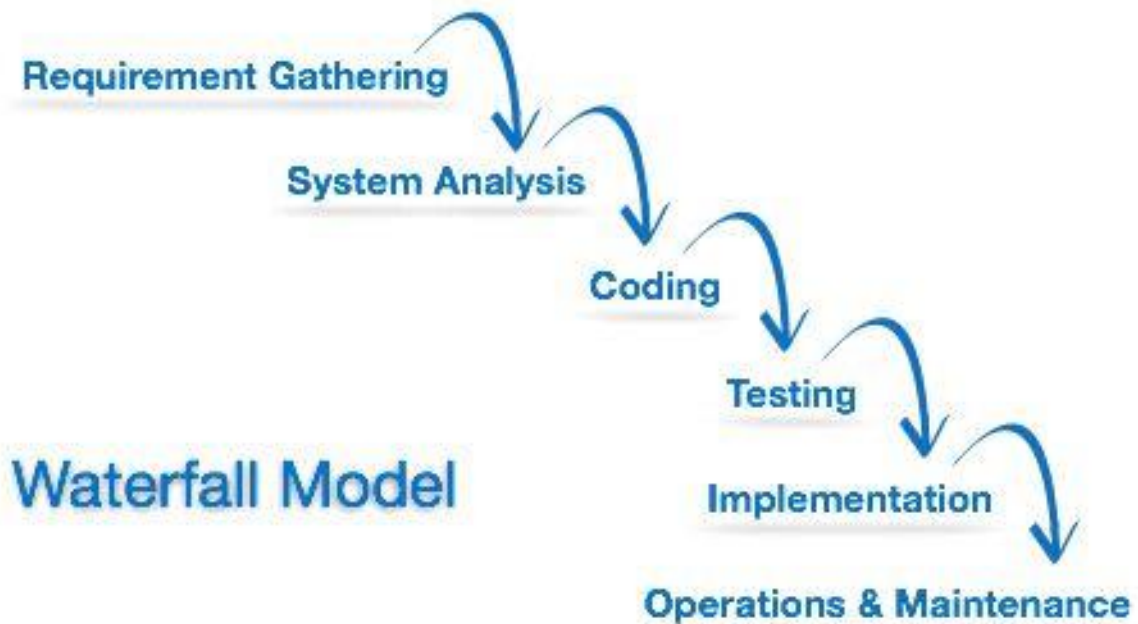


Các hoạt động

- **Hiện thực hóa (*Code – Dev/Coder/Programmer – Source Code/Build*)**
 - Dùng ngôn ngữ lập trình để cài đặt chương trình.
- **Kiểm thử (*Testing – Tester/QC – Program*)**
 - Kiểm tra chương trình để tìm lỗi đảm bảo PM hoạt động đúng.
- **Triển khai và bảo hành, bảo trì (*Maintenance*)**
 - Cài đặt.
 - Đảm bảo chương trình vận hành tốt.
 - Hỗ trợ KH xử lý các lỗi khi vận hành.
 - Nâng cấp, cập nhật phần mềm

Mô hình thác nước (*Waterfall*):

Như một dòng chảy, với các pha/giai đoạn được thực hiện tuần tự theo trật tự nghiêm ngặt và không có sự quay lui hay nhảy vượt pha.

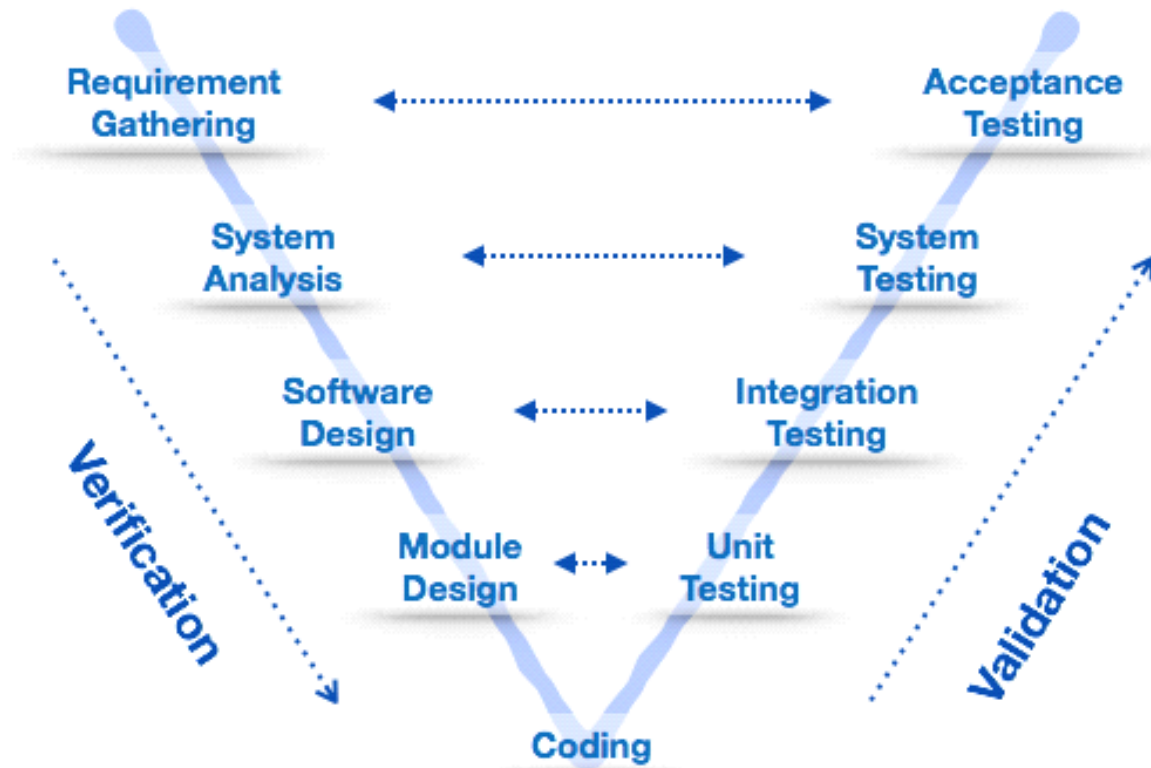


MỘT SỐ MÔ HÌNH PHÁT TRIỂN PHẦN MỀM



Mô hình chữ V (*Verification process*):

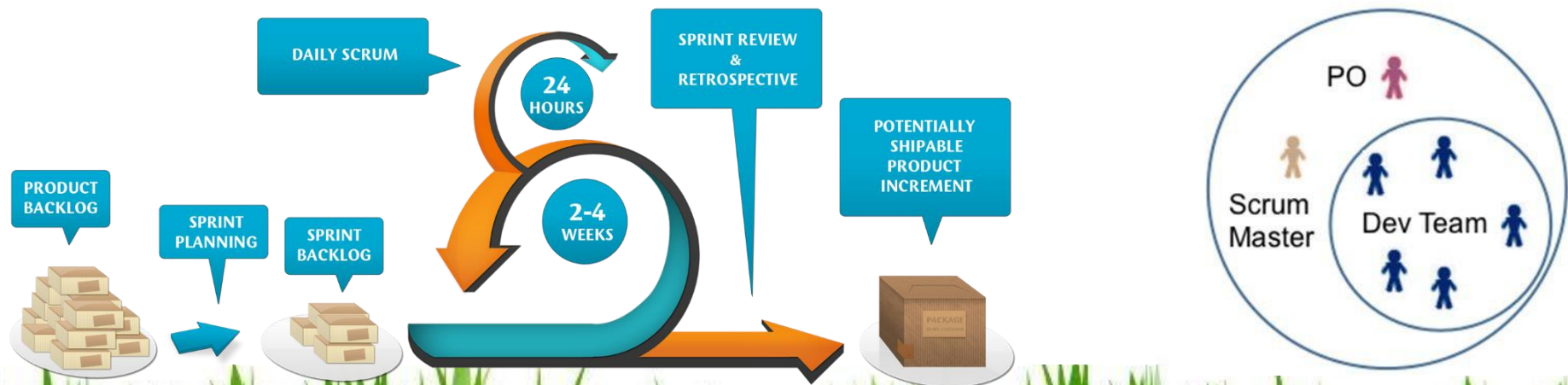
Chú trọng các hoạt động kiểm thử ở mỗi giai đoạn phát triển



MỘT SỐ MÔ HÌNH PHÁT TRIỂN PHẦN MỀM

Quy trình Agile/Scrum:

- Xây dựng phần mềm theo hướng linh hoạt
- Dựa trên mô hình lặp và tăng trưởng.
- Chia phần mềm thành các phần nhỏ (sprint) để phát triển.
- Mỗi phần nhỏ được phát triển theo mô hình lặp
- Có sự tham gia và tương tác của khách hàng → luôn được validation.



Kết luận:

- Với mỗi quy trình phát triển khác nhau → vị trí, vai trò, thời điểm kiểm thử khác nhau.
- Kiểm thử viên cần nắm vững được quy trình công ty sử dụng để làm tốt công việc của mình.



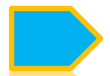
Phần mềm



Quy trình phát triển phần mềm



Một số mô hình phát triển phần mềm



Chất lượng phần mềm

1. Theo IEEE và ISO chất lượng phần mềm được xem xét ở góc độ (view) nhìn nhận của ai? *(Người dùng, người phát triển, nhà quản lý dự án hay nhà cung cấp)*
2. Bộ tiêu chí đánh giá chất lượng phần mềm của McCall xem xét ở những khía cạnh nào và cụ thể các khía cạnh đó gồm những tiêu chí nào được đặt ra để xem xét chất lượng của phần mềm?
3. Giữa các bộ tiêu chí đánh giá của McCall, ISO 9126/ISO25010, CMM/CMMI có sự khác biệt gì?

CHẤT LƯỢNG PM (Software quality)

Thị trường phần mềm:



CHẤT LƯỢNG PM (Software quality)

- "Software quality is the degree to which a system, component, or process meets specified requirements and user/customer needs or expectations." (*IEEE*)
- "The degree to which software satisfies stated and implied needs when used under specified conditions." (*ISO*).

View

Khách hàng: đáp ứng được yêu cầu sử dụng và kỳ vọng, phù hợp với người dùng

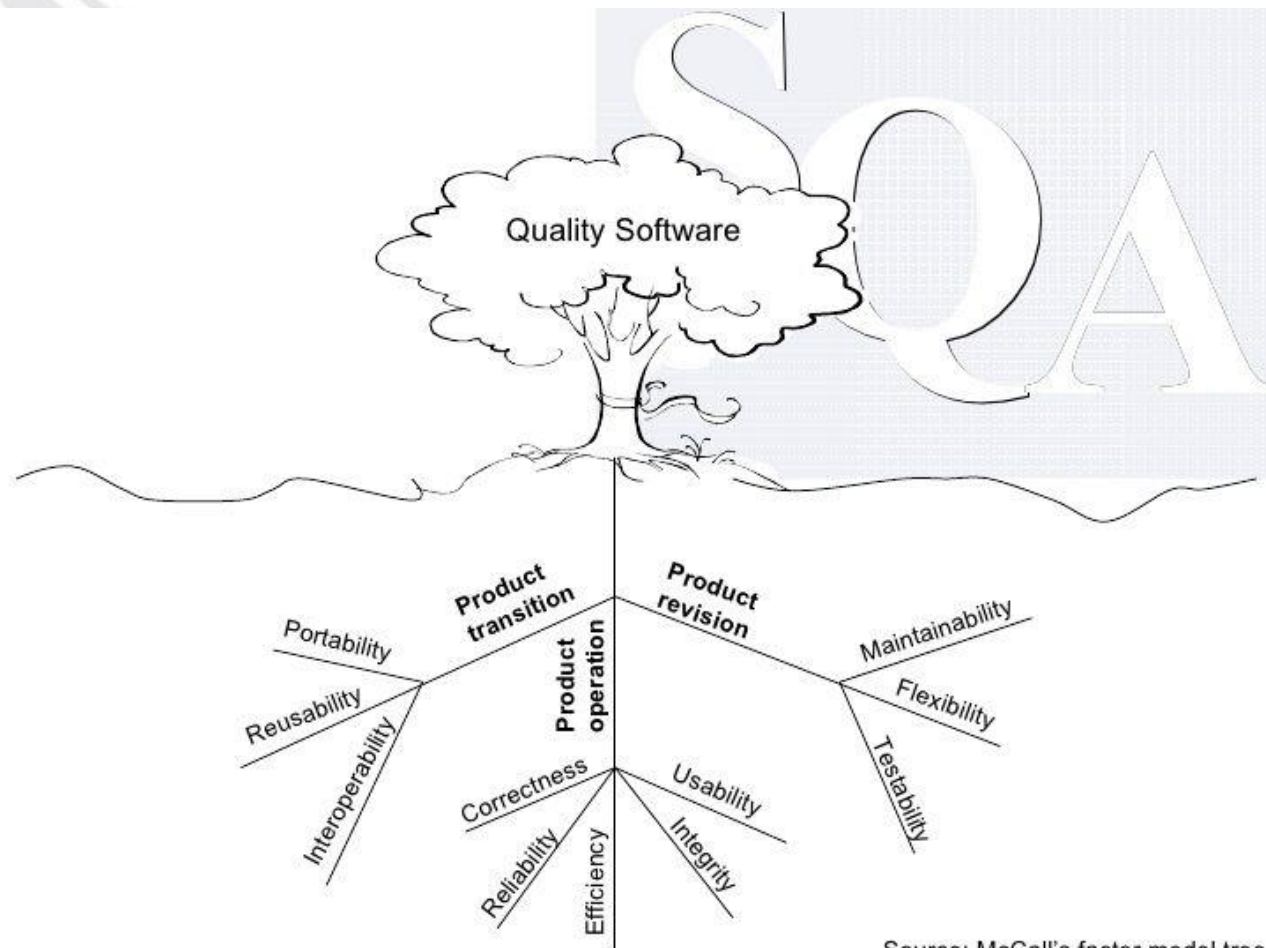
Người phát triển PM: xây dựng theo đúng đặc tả và tuân thủ đúng quy trình.

Quản lý dự án: đúng thời hạn và ngân sách, đạt mục tiêu đề ra

Nhà cung cấp: sản phẩm đạt được các tiêu chí đánh giá do nhà cung cấp đề ra.

CHẤT LƯỢNG PM (Software quality)

Một số tiêu chí đánh giá chất lượng PM (McCall – 1977, 11)



Source: McCall's factor model tree

Một số tiêu chí đánh giá chất lượng PM (McCall - 11)

- **Tiêu chí vận hành sản phẩm (*product operation*) - 5**
 - *Tính đúng đắn (Correctness): Đặc tả về độ chính xác, tính toàn vẹn của Outputs.*
 - *Tính tin cậy (Reliability): Định ra tỉ lệ lỗi của từng chức năng hoặc của cả hệ thống. Hệ thống chạy ổn định?*
 - *Tính hiệu quả (Efficiency): Tài nguyên phần cứng cần để thực hiện các chức năng của phần mềm.*
 - *Tính toàn vẹn (Integrity): Bảo vệ hệ thống, ngăn các truy cập trái phép.*
 - *Tính khả dụng (Usability): Tính dễ đọc, dễ dùng và hiệu quả.*

Một số tiêu chí đánh giá chất lượng PM (McCall - 11)

- **Tiêu chí sửa đổi sản phẩm (*product revision*) - 3**
 - *Tính bảo trì được (Maintainability): Mức công sức để tìm ra nguyên nhân + sửa + xác nhận đã sửa được lỗi*
 - *Tính linh hoạt (Flexibility): Bảo trì cải tiến dễ dàng*
 - *Tính kiểm thử được (Testability): Có lưu lại kết quả trung gian để hỗ trợ kiểm thử hay không? Có tạo file log, backup?*

Một số tiêu chí đánh giá chất lượng PM (McCall - 11)

- **Tiêu chí chuyển giao sản phẩm (*product transition*)**
 - *Tính khả chuyển (Portability): cài đặt trong môi trường mới (phần cứng, hệ điều hành, ...) vẫn hoạt động tốt. Dễ dàng cài đặt/gỡ bỏ.*
 - *Khả năng tái sử dụng (Reusability): có thể tái sử dụng các phần của phần mềm cho các ứng dụng khác.*
 - *Khả năng tương thích (Interoperability): phần mềm có khả năng giao tiếp, tương tác được với các hệ thống đã có.*

CHẤT LƯỢNG PM (Software quality)

Một số bộ tiêu chí đánh giá:

- Chất lượng phần mềm: ISO9126 (1991), ISO25010 (2011)
- Chất lượng quy trình PTPM: CMM (1986 - 1993), CMMI (2000).

Lỗi phần mềm

- Xác định các mức lỗi Error, Bug/Fault và Failure trong các đoạn mã sau?

```
void Sum(int a, int b)
{
    int kq = a + b;
    long sum = a + b;
    printf("Sunary %d + %d = %ld\n", a, b, sum);
}
int main()
{
    Sum(10,5);
    return 1;
}
```

Lỗi phần mềm

- Xác định các mức lỗi Error, Bug/Fault và Failure trong các đoạn mã sau?

```
#include <stdio.h>
int HB(int r1, float tb)
{
    if((r1>=80) || (tb>=7.5))
        return 1;
    return 0;
}

int main()
{
    HB(80, 8.3);
    HB(80, 4);
}
```

Rút ra nhận xét về mối quan hệ giữa *Error*, *Fault* và *Failure*.

Lỗi phần mềm

Là sự không khớp giữa kết quả thực tế của phần mềm với bản đặc tả hay mong đợi của khách hàng.

- PM không thực hiện đúng những gì mà đặc tả định nghĩa → *Sai*
- PM không thực hiện những gì mà đặc tả không đề cập đến nhưng lẽ ra nên thực hiện → *Thiếu*
- PM thực hiện những gì mà đặc tả không đề cập đến → *Thừa*.
- PM khó hiểu hay khó sử dụng → *Thảm mỹ/Thuận tiện*

CHẤT LƯỢNG PM (Software quality)

Bug

đầu tiên


được

tìm thấy

9/9

0800 Antan started
1000 stopped - antan ✓ { 1.2700 9.032 847 025
1300 (032) MP-MC 2.130476415 (032) 4.615925059 (-2)
033 PRO 2 2.130476415
conv 2.130676415
Relays 6-2 in 033 failed special input test
in relay 11.00 test.
Relays changed

1100 Started Cosine Tape (Sine check)
1525 Started Multi-Adder Test.

1545  Relay #70 Panel F
(moth) in relay.

First actual case of bug being found.

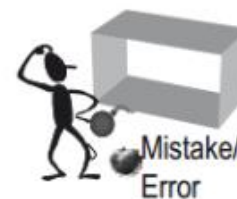
1630 Antan started.
1700 closed down.

Relay
3145
032, 1171

CHẤT LƯỢNG PM (Software quality)

Lỗi phần mềm - các mức lỗi:

- **Error:** là sai sót của con người trong quá trình phát triển phần mềm (*phân tích, thiết kế, lập trình sai*)
- **Defect/Fault:** là kết quả của sai sót (error) ...Khi sai sót thể hiện trên các sản phẩm sẽ gây ra lỗi Defect khiến phần mềm chạy không chính xác.
- **Failure:** Thất bại của phần mềm khi fault được kích hoạt, là kết quả khi vận hành/chạy sản phẩm



CHẤT LƯỢNG PM (Software quality)

Lỗi phần mềm - các mức lỗi:

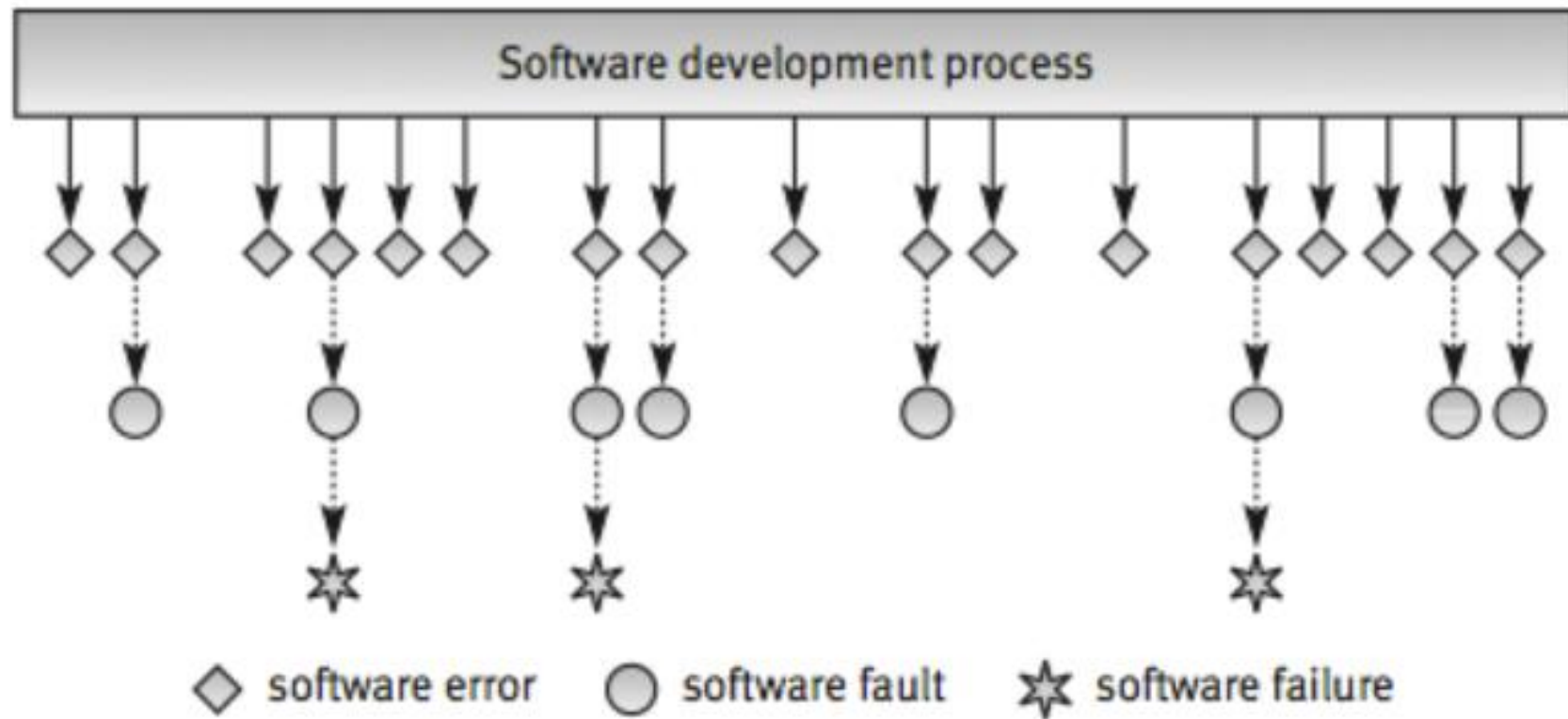


Figure 2.1: Software errors, software faults and software failures

CHẤT LƯỢNG PM (Software quality)

Lỗi phần mềm

```
void Sum(int a, int b)
{
    int kq = a + b;
    long sum = a + b;
    printf("Sunary %d + %d = %ld\n", a, b, sum);
}

int main()
{
    Sum(10,5);
    return 1;
}
```


CHẤT LƯỢNG PM (Software quality)

Lỗi phần mềm

```
#include <stdio.h>
int HB(int r1, float tb)
{
    if((r1>=80) || (tb>=7.5))
        return 1;
    return 0;
}

int main()
{
    HB(80, 8.3);
    HB(80, 4);
}
```

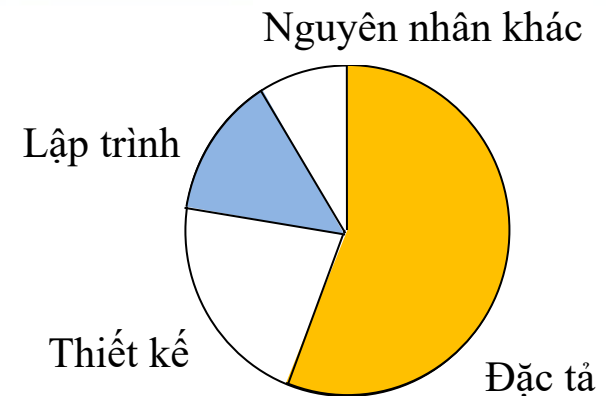
CHẤT LƯỢNG PM (Software quality)

Lỗi phần mềm

```
1  #include <stdio.h>
2  void Tong(int n)
3  {
4      int S=0,i,dem;
5      for(int i=2;i<n;i++)
6      {
7          dem=0;
8          for(int j=2;j<=i-1;j++)
9              if(i%j==0)
10             {
11                 dem++;
12                 break;
13             }
14             if(dem==0)
15                 S=S+i;
16     }
17     printf("\nTong la: %d",S);
18 }
19 int main()
20 {
21     Tong(10);    //TestCase1
22     Tong(12);    //TestCase2
23     Tong(7);     //TestCase3
24     Tong(11);    //TestCase4
25     return 0;
26 }
```

CHẤT LƯỢNG PM (Software quality)

Lỗi phần mềm – thời điểm xuất hiện:



Request 1

Request 2

Request 3

Request 4

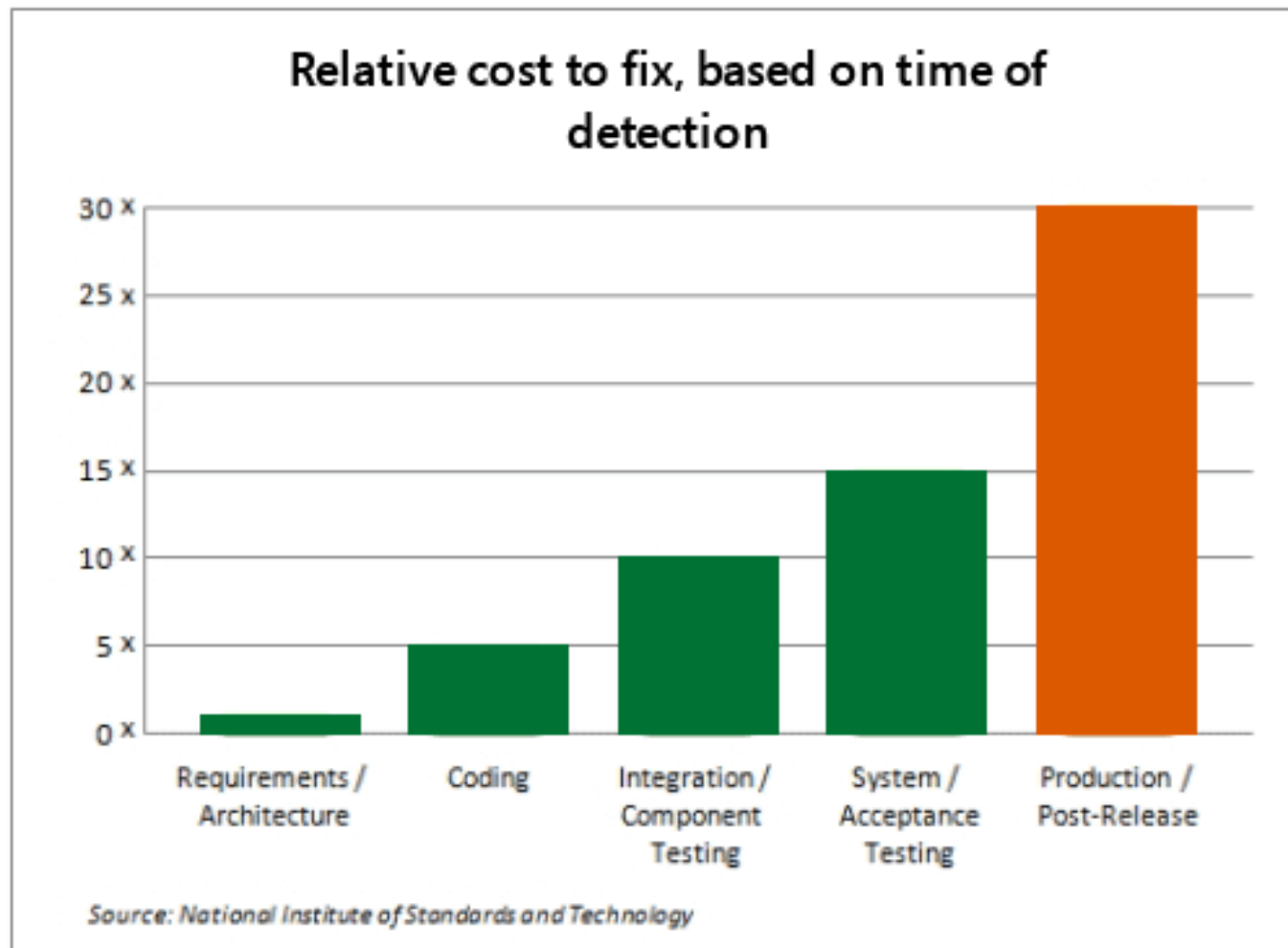
Correct Requirement	Correct Requirement	Correct Requirement	Incorrect Requirement
Designed as per requirement	Designed as per requirement	Mistakes made in design	Designed as per requirement
Developed as per design	Mistakes in development	Developed as per design	developed as per design
Correct Product	Product has coding defects	Product has design defects	Wrong product.

Lỗi phần mềm – nguyên nhân gây lỗi:

- Incorrect requirements
- Wrong design
- Poor coding
- Complex business logic/technology
- Work pressure
- Frequently changing requirements.
- Shorten software testing process

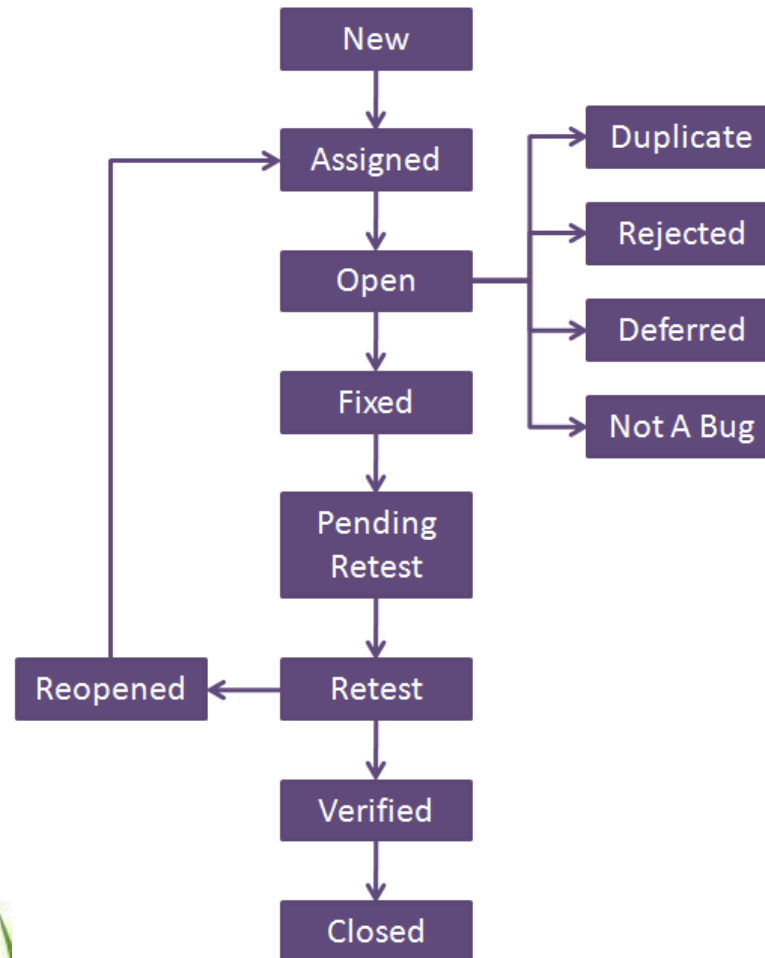
CHẤT LƯỢNG PM (Software quality)

Lỗi phần mềm – *chi phí sửa lỗi*:



CHẤT LƯỢNG PM (Software quality)

Lỗi phần mềm – vòng đời của lỗi:



Bug / Defect Lifecycle

<http://ISTQBExamCertification.com>

MỘT SỐ MÔ CÂU HỎI

1. The word "bug" is synonymous with which of the following words?

A. Error B. Defect C. Failure D. Mistake

2. List activities in SDLC?

MỘT SỐ MÔ CÂU HỎI

Software Requirements Specification : đặc tả yêu cầu phần mềm

3. Who write the SRS?

4. In which model that the testing phase start after the Development phase?

5. Activities involved in the Validation process of the V model?

6. Activities involved in the Verification process of the V model?

- A. Requirement verification
- B. Functional verification
- C. Code validation
- D. Design validation.

7. Ensuring that testing starts during the requirements phase is important to enable which of the following test objectives?

- A. Unit testing
- B. Integration testing
- C. System testing
- D. Acceptance testing

8. Which of the following statements is true?

- A. In waterfall model, customer involved each phase.
- B. In waterfall phases run parallel.
- C. In waterfall, requirement can change frequency.
- D. In waterfall, testing occurs late.

9. Which of the following statements is true?

- A. Verification is: “we build a product right”
- B. Validation is: “we build a right product”
- C. Both
- D. None

10. Unit Testing thuộc về hoạt động nào?

- A. Validation.
- B. Verification.
- C. Development.
- D. Design.

11. Theo định nghĩa IEEE thì PM chất lượng đáp ứng yêu cầu của ai?

- A. Người dùng
- B. Nhà sản xuất/phát triển phần mềm
- C. Quản lý dự án
- D. Một đáp án khác

12. Cho biết các mức kiểm thử được sử dụng trong mô hình chữ V?

- A. Integration Testing, System Testing, Acceptance Testing and Regression Testing
- B. Component Testing, Integration Testing, System Testing and Acceptance Testing
- C. Incremental Testing, Exhaustive Testing, Exploratory Testing, Data Driven Testing
- D. Alpha Testing, Beta Testing, BlackBox Testing, WhiteBox Testing

MỘT SỐ MÔ CÂU HỎI

13. Trong bộ tiêu chí đánh giá chất lượng McCall, tính/tiêu chí **bảo trì được** thuộc nhóm tiêu chí nào sau đây?

- A. Tiêu chí vận hành sản phẩm
- B. Tiêu chí chuyển giao sản phẩm
- C. Tiêu chí sửa đổi sản phẩm
- D. Không có nhóm nào

14. Theo thống kê từ những thách thức đối với công nghệ phần mềm thì lỗi nhiều nhất là do giai đoạn:

- A. Kiểm tra và bảo trì
- B. Phân tích yêu cầu
- C. Thiết kế
- D. Lập trình

MỘT SỐ MÔ CÂU HỎI

15. Cho đoạn mã CT sau, hãy phân tích xác định các mức lỗi error, fault và failure?

(Đoạn CT đếm số lượng kí số chẵn có mặt trong số nguyên n)

```
#include <stdio.h>
int DemKSChan(int n)
{
    int dem=0;
    do
    {
        if(n%10%2==1)
            dem++;
        n/=10;
    }while(n!=0);
    return dem;
}
int main()
{
    printf("So luong ki so chan trong so -123 la: %d\n",DemKSChan(-123));//TestCase 1
    printf("So luong ki so chan trong so 357 la: %d\n",DemKSChan(357));//TestCase 2
    printf("So luong ki so chan trong so 0 la: %d\n",DemKSChan(0));//TestCase 3
    printf("So luong ki so chan trong so 1274 la: %d\n",DemKSChan(1274));//TestCase 4
}
```

Presentation Ending

THANKS FOR ALL
THANKS FOR ALL