CÔNG NGHỆ PHẦN MỀM

BÀI 4
ĐẢM BẢO CHẤT LƯỢNG PHẦN MỀM

Giảng viên: TS. Lê Nguyễn Tuấn Thành
Email: thanhlnt@tlu.edu.vn

Bộ Môn Công Nghệ Phần Mềm – Khoa CNTT
Trường Đại Học Thủy Lợi

Nội dung

1. Quản lý chất lượng phần mềm
2. Kiểm thử phần mềm
3. Bảo trì phần mềm

3

PHẦN 1. QUẢN LÝ CHẤT LƯỢNG PHẦN MỀM

Quality Management

Quản lý chất lượng phần mềm

- Liên quan đến việc đảm bảo một sản phẩm phần mềm đạt được mức độ yêu cầu về chất lượng
- Liên quan đến việc định nghĩa các **tiêu chuẩn chất lượng** phù hợp và đảm bảo rằng những điều này được tuân thủ

Chất lượng là gì?

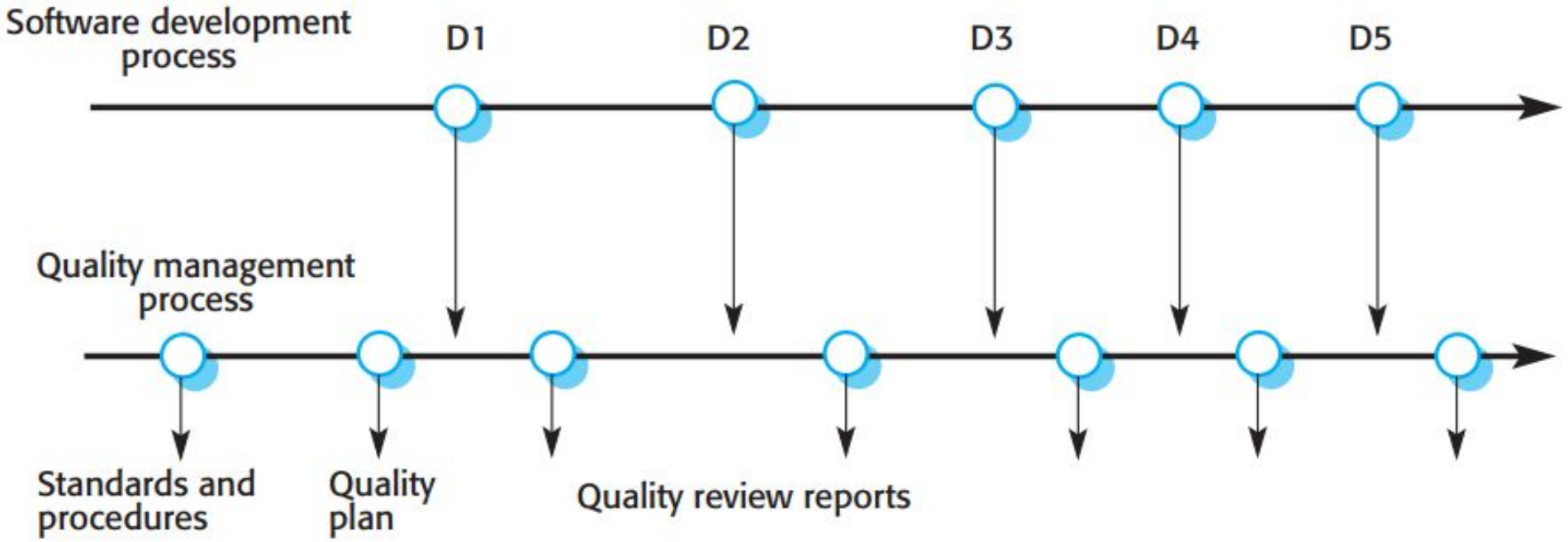
*Chất lượng nghĩa là một sản phẩm phải **đáp ứng các đặc tả** của nó*

Hoạt động quản lý chất lượng

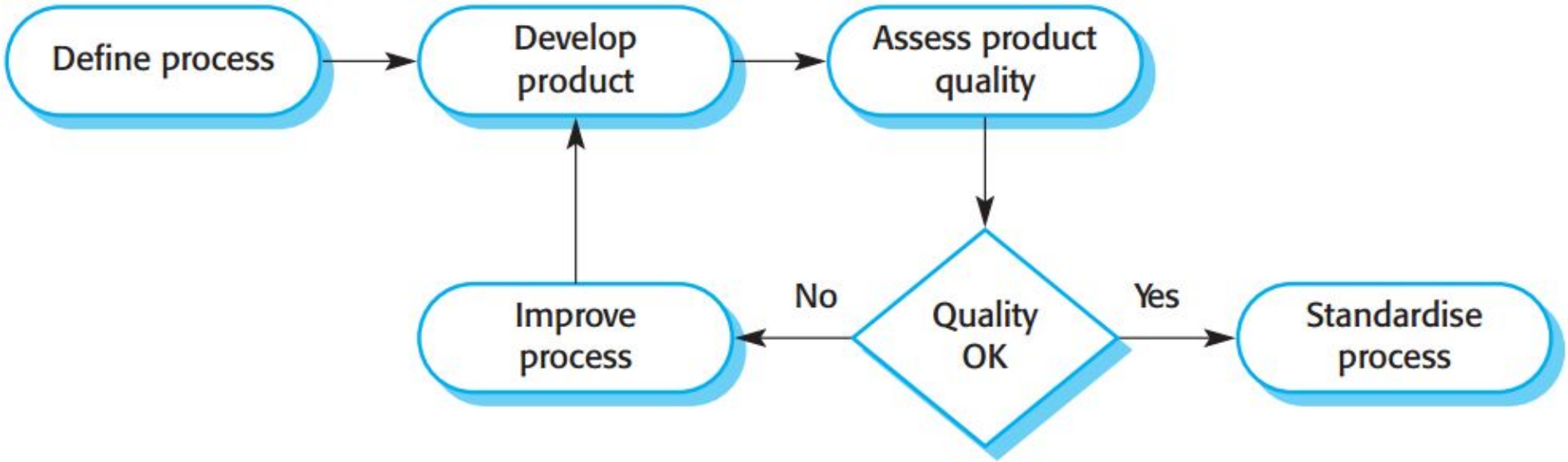
1. **Q**uality **A**ssurance (QA)
   - Thiết lập quy trình và tiêu chuẩn chất lượng
2. **Q**uality **P**lanning (QP)
   - Lựa chọn các thuộc tính chất lượng
3. **Q**uality **C**ontrol (QC)
   - Đảm bảo các quy trình và tiêu chuẩn được tuân thủ

Quản lý chất lượng nên tách biệt với quản lý dự án để đảm bảo sự độc lập

Quản lý chất lượng & Phát triển phần mềm



1.1. Quality Assurance (QA)



Tiêu chuẩn chất lượng

- **Tiêu chuẩn** là đóng gói các kinh nghiệm thực tiễn tốt nhất, giúp tránh lặp lại những sai lầm trong quá khứ
- **Mức độ**: *tiêu chuẩn quốc tế, quốc gia, tổ chức, dự án*

Tiêu chuẩn quốc tế: ISO 9001

- Một bộ tiêu chuẩn quốc tế về quản lý chất lượng.
- Có thể áp dụng cho một khoảng các tổ chức từ sản xuất đến ngành công nghiệp dịch vụ.

Tiêu chuẩn Việt Nam

- TCVN là viết tắt của cụm từ **Technical Commit of Vietnam**
- Hiện nay đã có hàng nghìn TCVN: *cơ bản, thuật ngữ, yêu cầu kỹ thuật, ghi nhãn, bao gói, …*
- Một số tiêu chuẩn CNTT có ảnh hưởng rộng rãi:
  - **TCVN 5712** định nghĩa chuẩn cho bộ mã ABC với cách nhập liệu Telex
  - **TCVN 6909** định nghĩa chuẩn mã hóa tiếng Việt như là một tập con của bộ mã Unicode 3.1
  - **TCVN ISO 9001** (tương đương với ISO 9001) về các yêu cầu đối với hệ thống quản lý chất lượng

Tiêu chuẩn tổ chức, dự án: Code conventions

- Tập hợp những nguyên tắc chung khi lập trình nhằm làm cho mã nguồn dễ đọc, dễ hiểu, do đó dễ quản lý, bảo trì hơn
- Ví dụ: quy tắc đặt tên
  - **Cú pháp lạc đà (camelCase)**
    - Ký tự đầu tiên của từ đầu tiên viết thường, những ký tự đầu tiên của những từ tiếp theo viết hoa, ví dụ: **productName, productPrice, …**
  - **Cú pháp Pascal (PascalCase)**
    - Viết hoa chữ cái đầu tiên của mỗi từ, ví dụ: **ProductName, ProductPrice, …**
  - **Cú pháp con rắn (snake_case)**
    - Tất cả các chữ cái đều viết thường, và các từ cách nhau bởi dấu gạch dưới, ví dụ: **product_name, product_price, …**

1.2. Quality Planning (QP)

- Tài liệu kế hoạch trong đó đưa ra các thuộc tính chất lượng sản phẩm mong muốn và cách đánh giá, định nghĩa các thuộc tính chất lượng này
- Cấu trúc kế hoạch chất lượng:
  - Giới thiệu sản phẩm
  - Các kế hoạch sản phẩm
  - Miêu tả quy trình
  - Các mục tiêu chất lượng
  - Rủi ro và quản lý rủi ro

Các thuộc tính chất lượng phần mềm

Tính an toàn	Tính có thể hiểu	Tính di động
Tính bảo mật	Khả năng kiểm tra	Tính khả dụng
Độ tin cậy	Tính tương thích	Khả năng tái sử dụng
Khả năng phục hồi	Tính mô-đun	Tính hiệu quả
Độ bền	Độ phức tạp	Khả năng học

1.3. Quality Control (QC)

- Liên quan việc kiểm tra quy trình phát triển phần mềm để đảm bảo rằng các quy trình và tiêu chuẩn đang được tuân thủ.
- Hai phương pháp để kiểm soát chất lượng:
  1. **Duyệt/Rà soát lại chất lượng**
  2. **Đo lường phần mềm thông qua các độ đo**

QC1. Duyệt lại chất lượng

- Là phương pháp chính để xác thực chất lượng của một quy trình hoặc một sản phẩm.
- Một nhóm sẽ kiểm tra một phần hoặc toàn bộ quy trình (*mã chương trình, thiết kế, đặc tả, kế hoạch kiểm thử, tiêu chuẩn*, v.v.) hoặc hệ thống và tài liệu của nó để tìm ra các vấn đề tiềm ẩn.

Các kiểu duyệt lại

1. **Duyệt lại để loại bỏ khiếm khuyết** (*sản phẩm*)
2. **Duyệt lại về tiến độ** (*sản phẩm và quy trình*)
3. **Duyệt lại chất lượng** (*sản phẩm và tiêu chuẩn*)

Kết quả duyệt lại

1. **Không cần sửa chữa**
2. **Tham khảo để sửa chữa**
3. **Xem lại thiết kế tổng thể**

QC2. Đo lường Phần mềm qua các độ đo

- Đo lường phần mềm liên quan đến việc tìm ra một giá trị **bằng số** cho một thuộc tính của sản phẩm phần mềm hoặc quy trình.
  - Cho phép so sánh khách quan các kỹ thuật và quy trình khác nhau.

Độ đo phần mềm

- Là **bất kỳ loại phép đo** nào liên quan đến hệ thống phần mềm, quy trình hoặc tài liệu
  - Số dòng mã trong chương trình, chỉ số Fog, số người/ngày cần thiết để phát triển một thành phần.
  - Cho phép lượng hóa phần mềm và quy trình phần mềm.

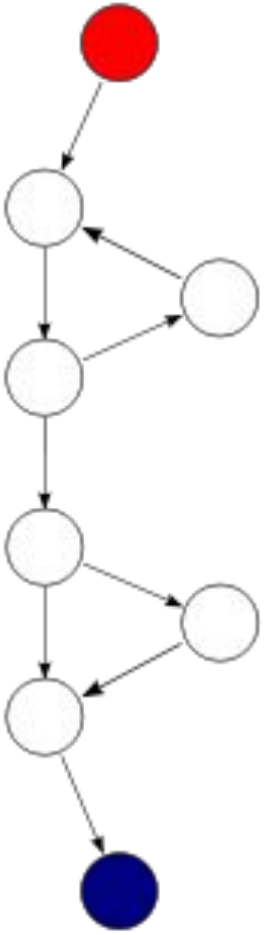
Độ đo chung

Metrics
Fan-in/Fan-out
Chiều dài đoạn mã (LOC – Line of Code**)**
Chiều dài của định danh (LI – Length of identifiers**)**
Độ sâu của các lệnh lồng có điều kiện
Chỉ số Fog (Fog index)
Độ phức tạp chu kỳ (CC - Cyclomatic complexity**)**

$$CC = E - N + 2*P$$

where:

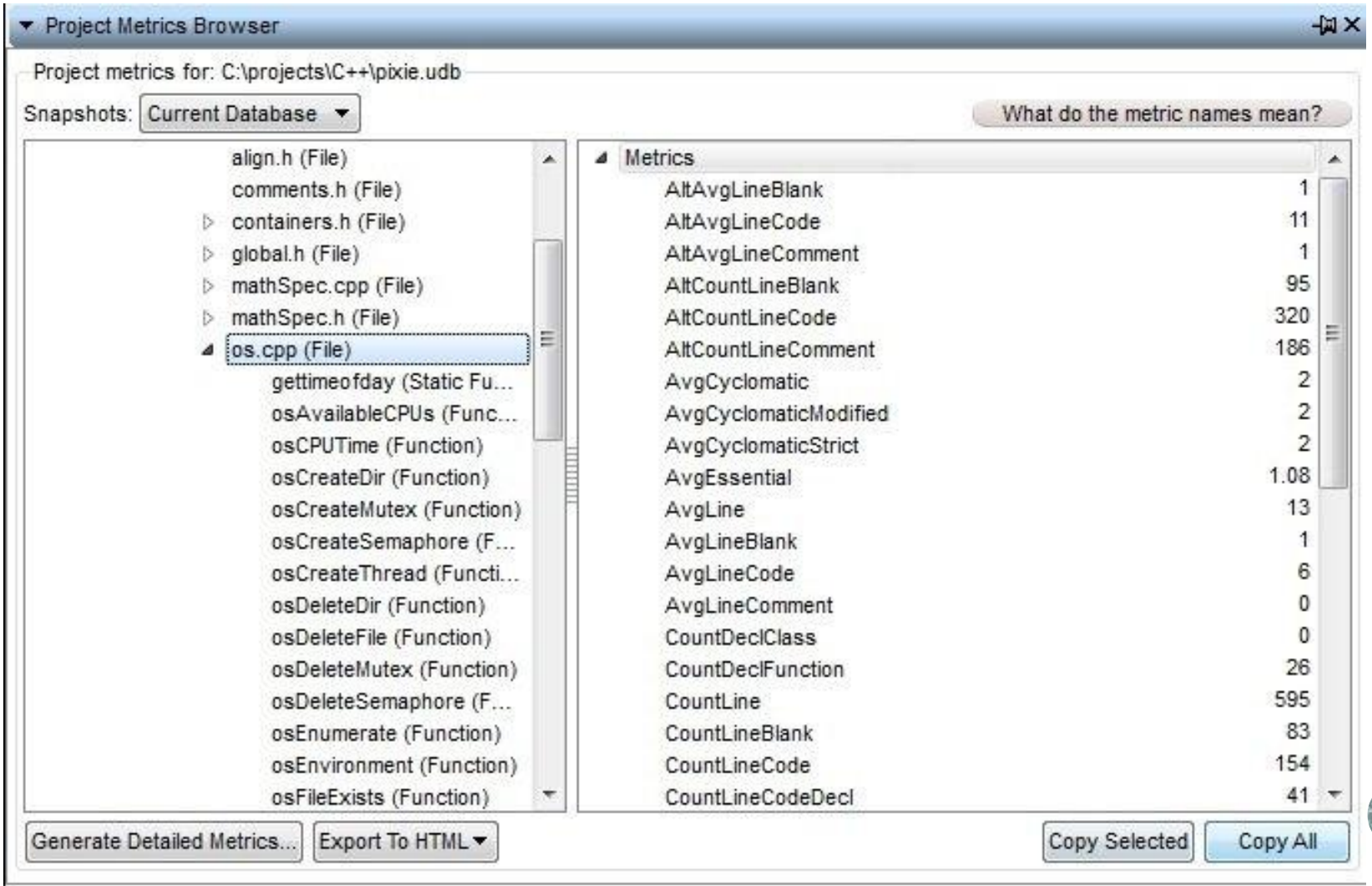
- *E* = number of **edges** of the graph
- *N* = number of **nodes** of the graph
- *P* = number of **connected components**



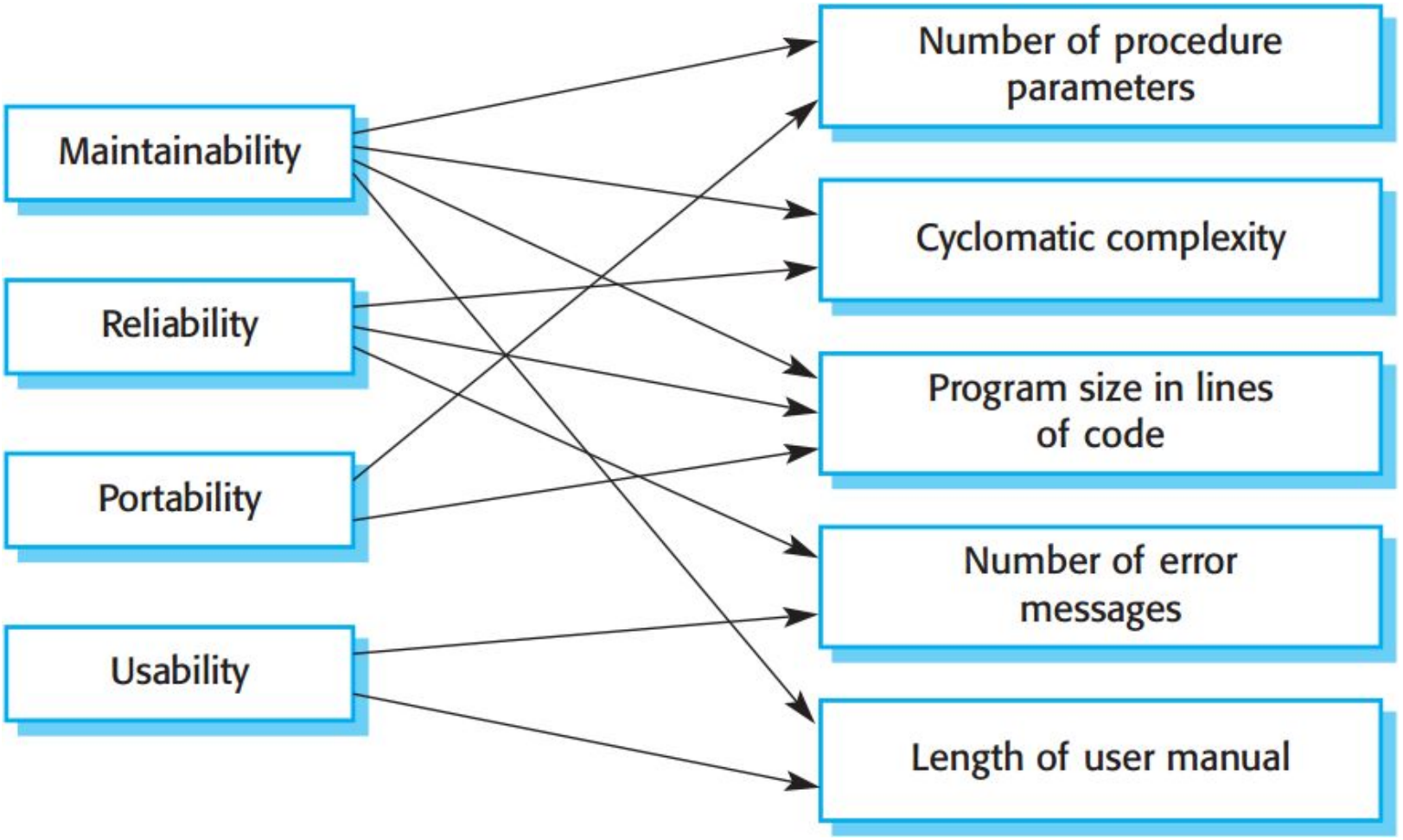
Độ đo hướng đối tượng

OO Metrics
Độ sâu của cây thừa kế (DIT – Depth of Inheritance Tree**)**
Số lượng lớp con trực tiếp (NOC – Number of Children**)**
Phương thức trọng số cho lớp (**WMC** – Weighted methods per class)
Coupling between object classes (**CBO** – Coupling between object classes)
Số lượng phương thức ghi đè (**NOM** – Number Of Overrided Methods)

Công cụ Understand



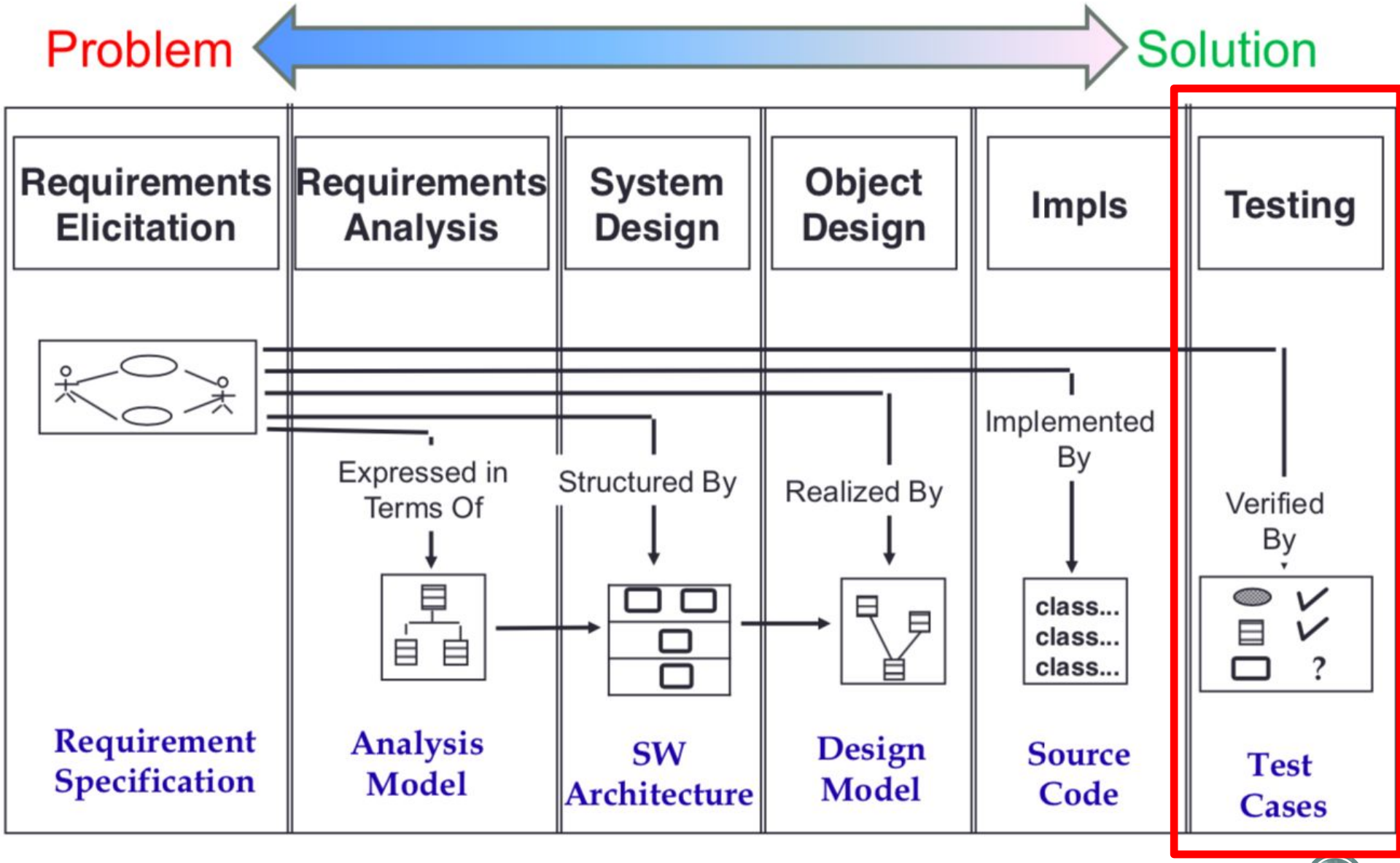
Độ đo và Thuộc tính



26

PHẦN 2. KIỂM THỬ PHẦN MỀM

Software Testing


Problem
Solution
Requirements Elicitation
Requirements Analysis
System Design
Object Design
Impls
Testing
Expressed in Terms Of
Structured By
Realized By
Implemented By
Verified By
class...
class...
class...
?
Requirement Specification
Analysis Model
SW Architecture
Design Model
Source Code
Test Cases

Testing

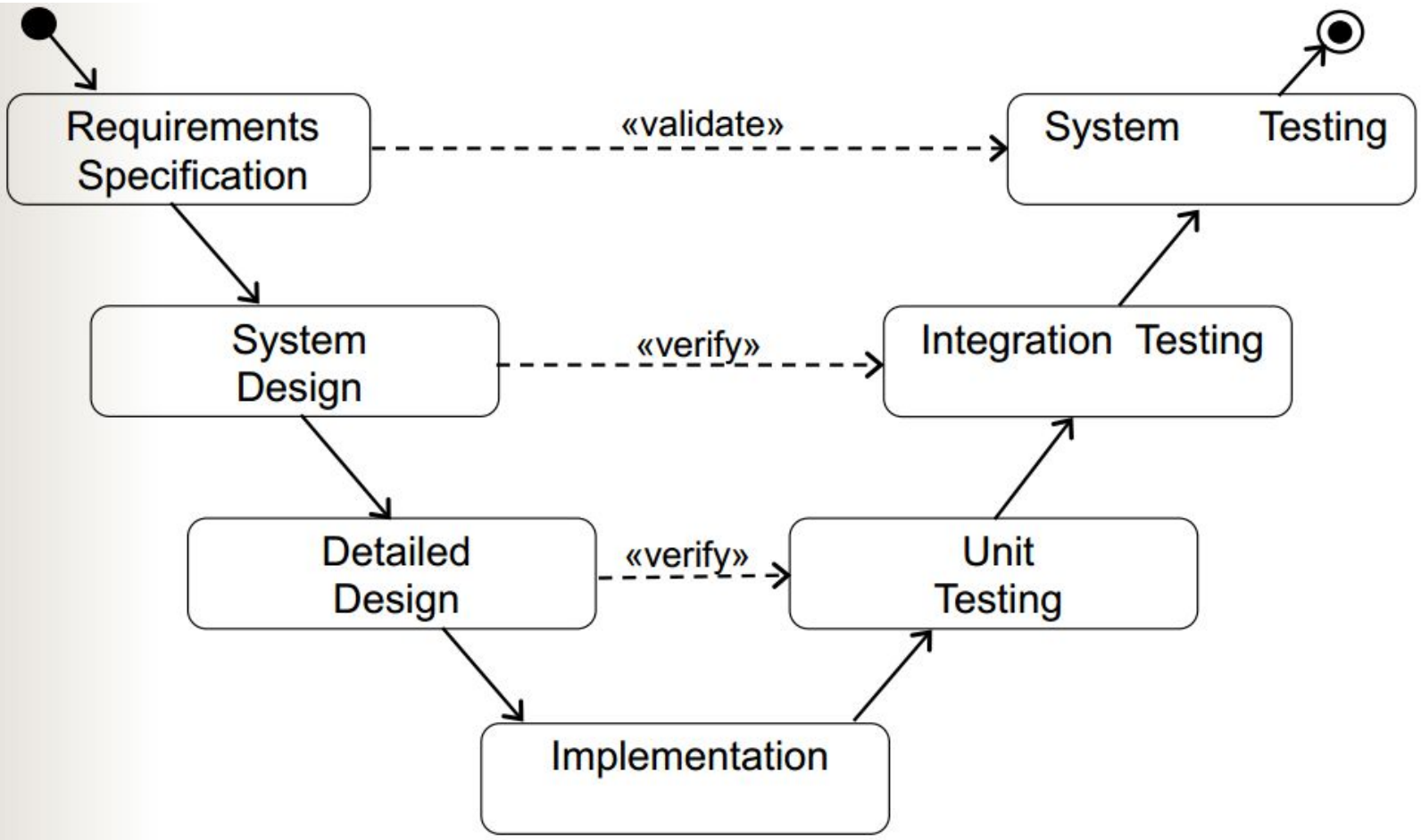
```
int main() {
   …
   return;
}
```

Test case: {i ∈ D, o ∈ O}
Test suite: set of test cases

Mục tiêu kiểm thử

- **Kiểm thử xác thực** (validation testing)
  - Để *chứng minh* cho người phát triển và khách hàng hệ thống rằng phần mềm đáp ứng các yêu cầu của nó
  - Một kiểm thử xác thực thành công cho thấy hệ thống hoạt động như dự định.
- **Kiểm thử khiếm khuyết** (defect testing)
  - Để *phát hiện lỗi hoặc khuyết tật* trong phần mềm có hành xử không đúng hoặc không phù hợp với đặc tả của nó;
  - Một kiểm thử khiếm khuyết thành công là một thử nghiệm làm cho hệ thống thực hiện không đúng và do đó bộc lộ một khiếm khuyết trong hệ thống.

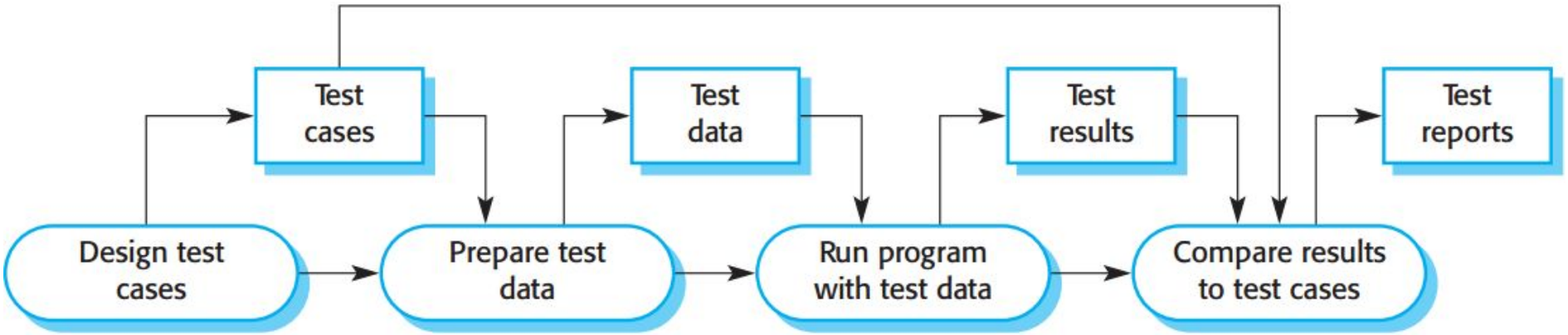
Quy trình kiểm thử



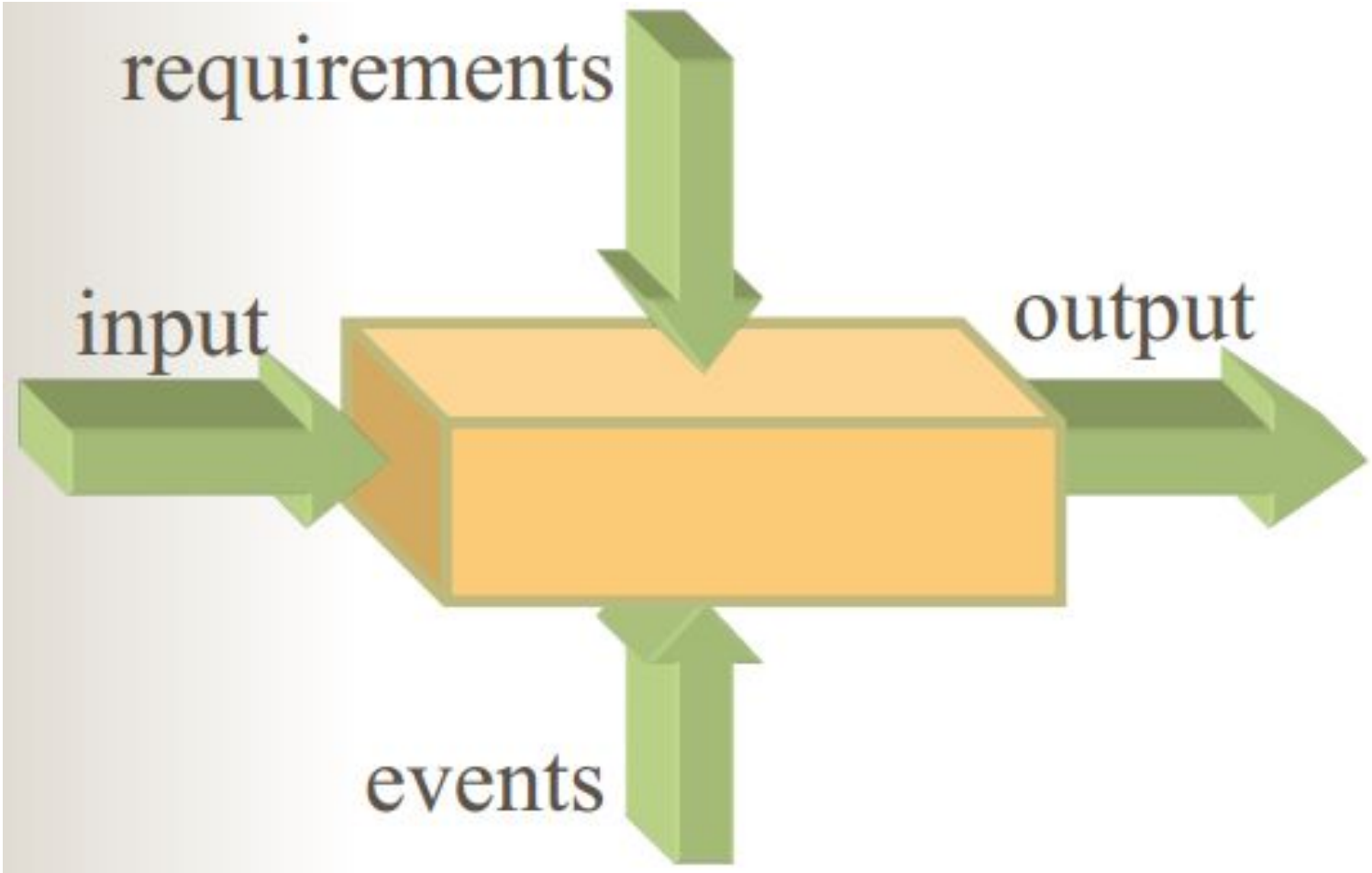
Mức độ kiểm thử

- **Unit testing**
  - Test an individual unit
- **Integration testing**
  - Test units in combination
- **System testing**
  - Test everything (functionality, usability, reliability, performance, portability ...)
- **Acceptance testing**
  - Customer accepts the software and give us their money

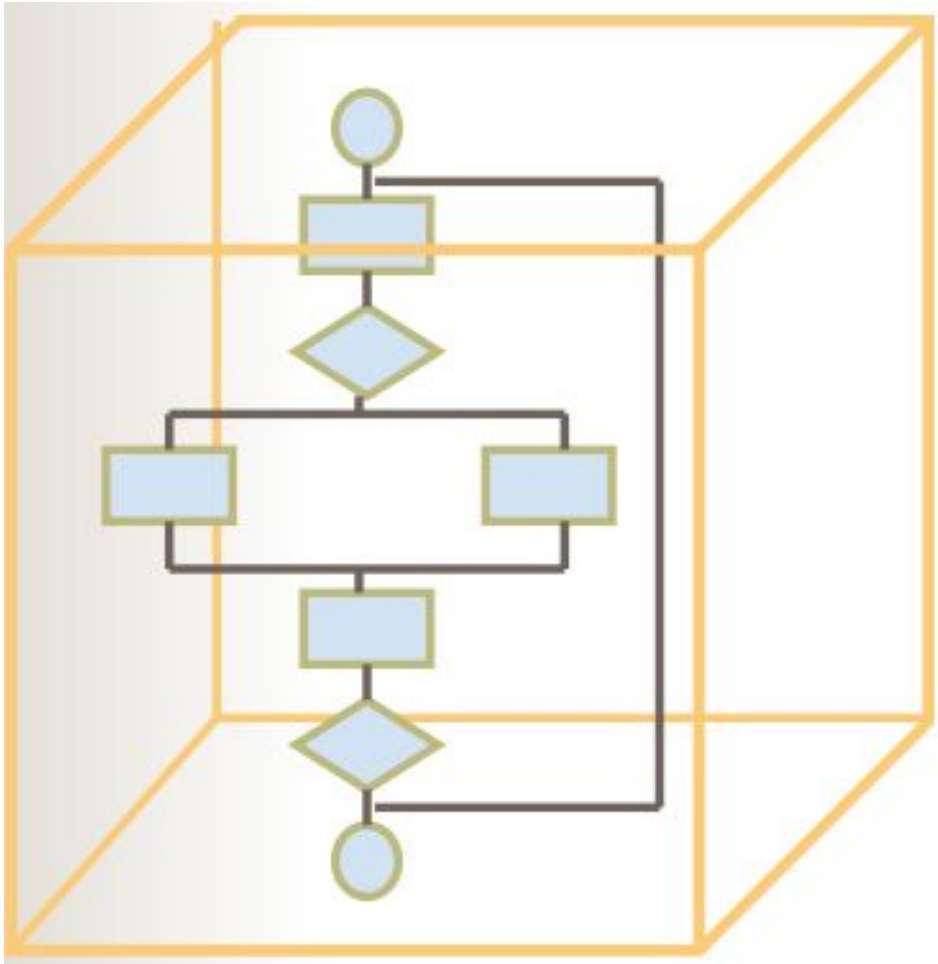
Quy trình kiểm thử phần mềm



Kiểm thử hộp đen

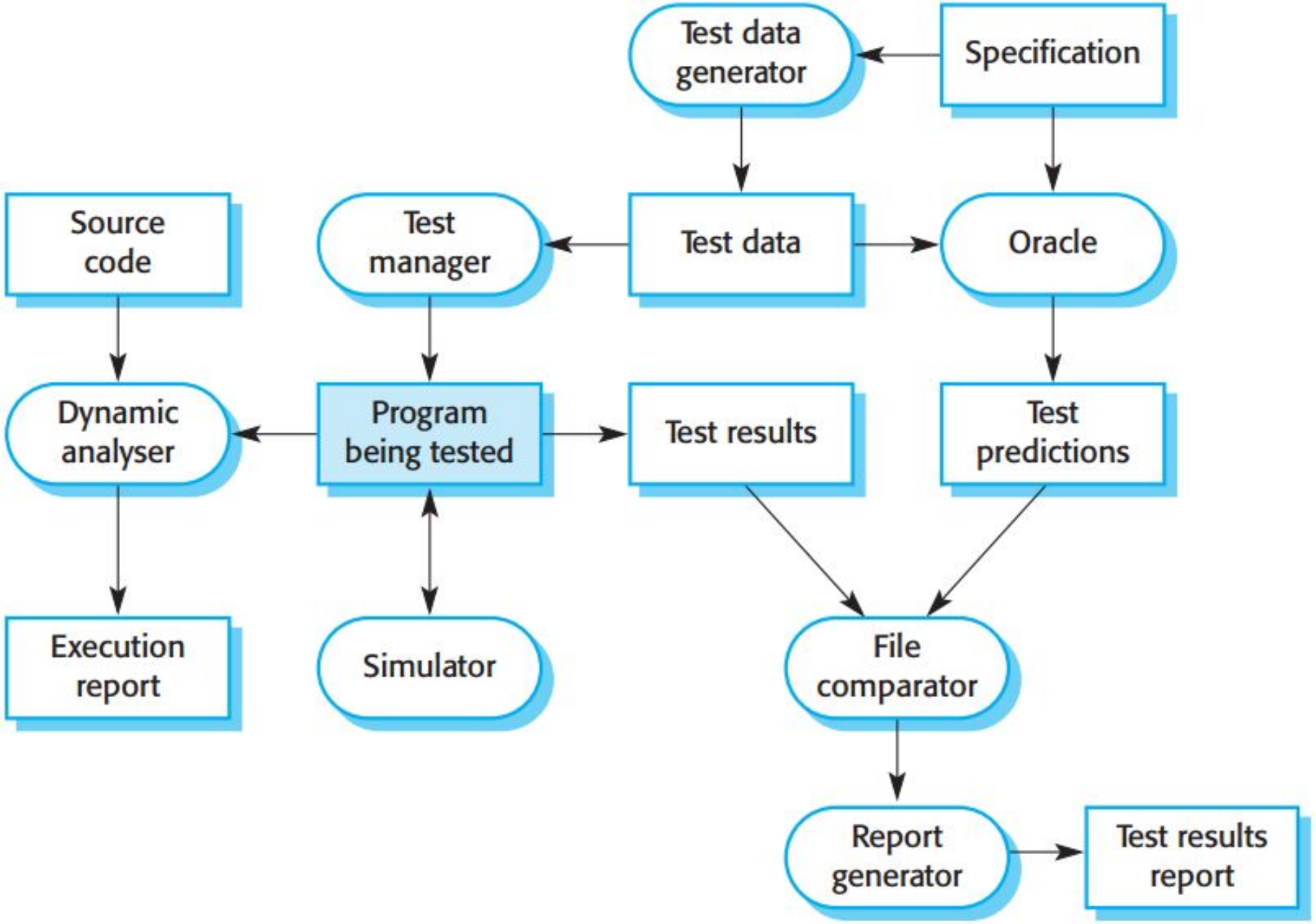


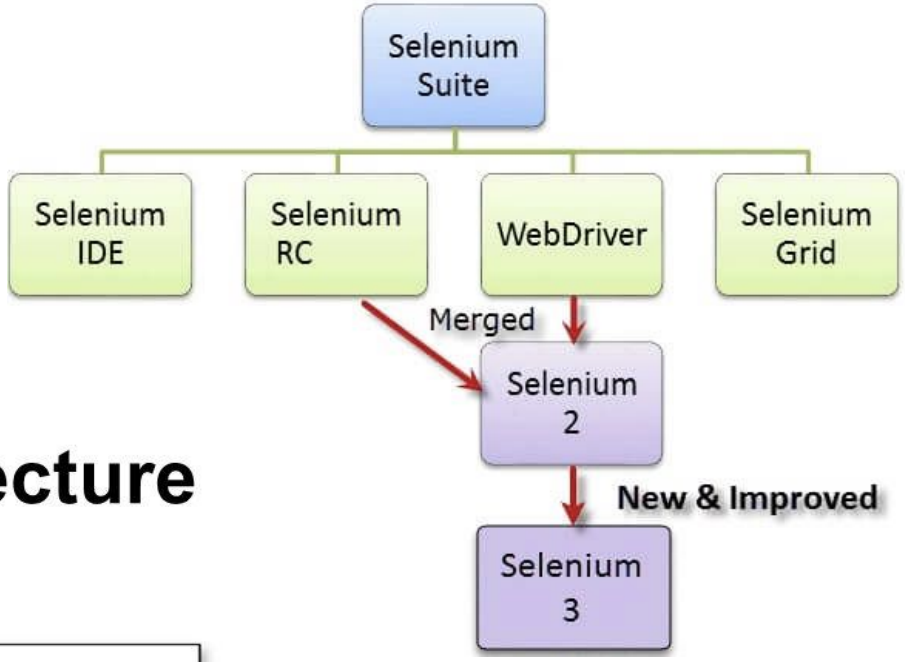
Kiểm thử hộp trắng



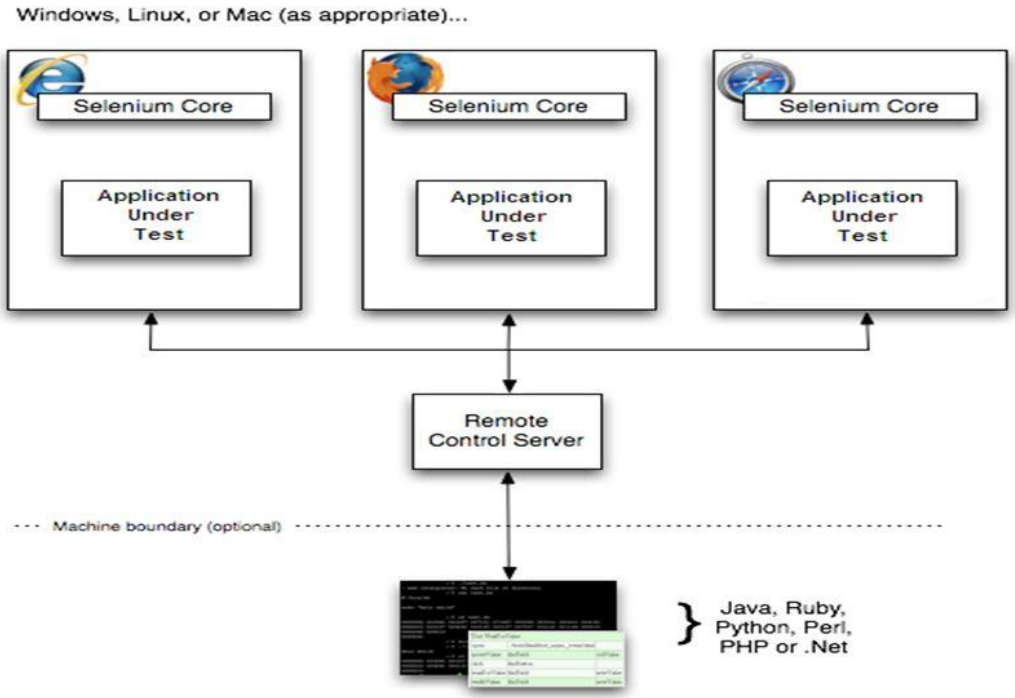
2.5. Tự động hóa kiểm thử

- Kiểm thử là một giai đoạn quy trình tốn kém.
- Các khung kiểm thử cung cấp một loạt các công cụ để giảm thời gian cần thiết và tổng chi phí kiểm thử.
  - VD: **JUnit, NUnit** hỗ trợ thực hiện tự động các bài kiểm thử.


Test data generator
Specification
Source code
Test manager
Test data
Oracle
Dynamic analyser
Program being tested
Test results
Test predictions
Execution report
Simulator
File comparator
Report generator
Test results report



Selenium RC : Architecture



38

PHẦN 3. BẢO TRÌ PHẦN MỀM

Software maintenance

Sự thay đổi / tiến hoá của phần mềm

- Phần mềm thay đổi là không thể tránh khỏi, do:
  - Các yêu cầu mới xuất hiện khi sử dụng phần mềm
  - Môi trường kinh doanh thay đổi
  - Nhiều lỗi phải được sửa chữa
  - Máy tính và thiết bị mới được thêm vào hệ thống
  - Hiệu suất hoặc độ tin cậy của hệ thống có thể phải được cải thiện.
- Một vấn đề quan trọng đối với các tổ chức là quản lý sự thay đổi đối với các hệ thống phần mềm hiện có

Những luật của Lehman về sự tiến hoá phần mềm

- Phải thay đổi liên tục
- Tăng độ phức tạp
- Ổn định về tổ chức
- Tăng trưởng liên tục
- Suy giảm chất lượng
- Hệ thống phản hồi (feedback)

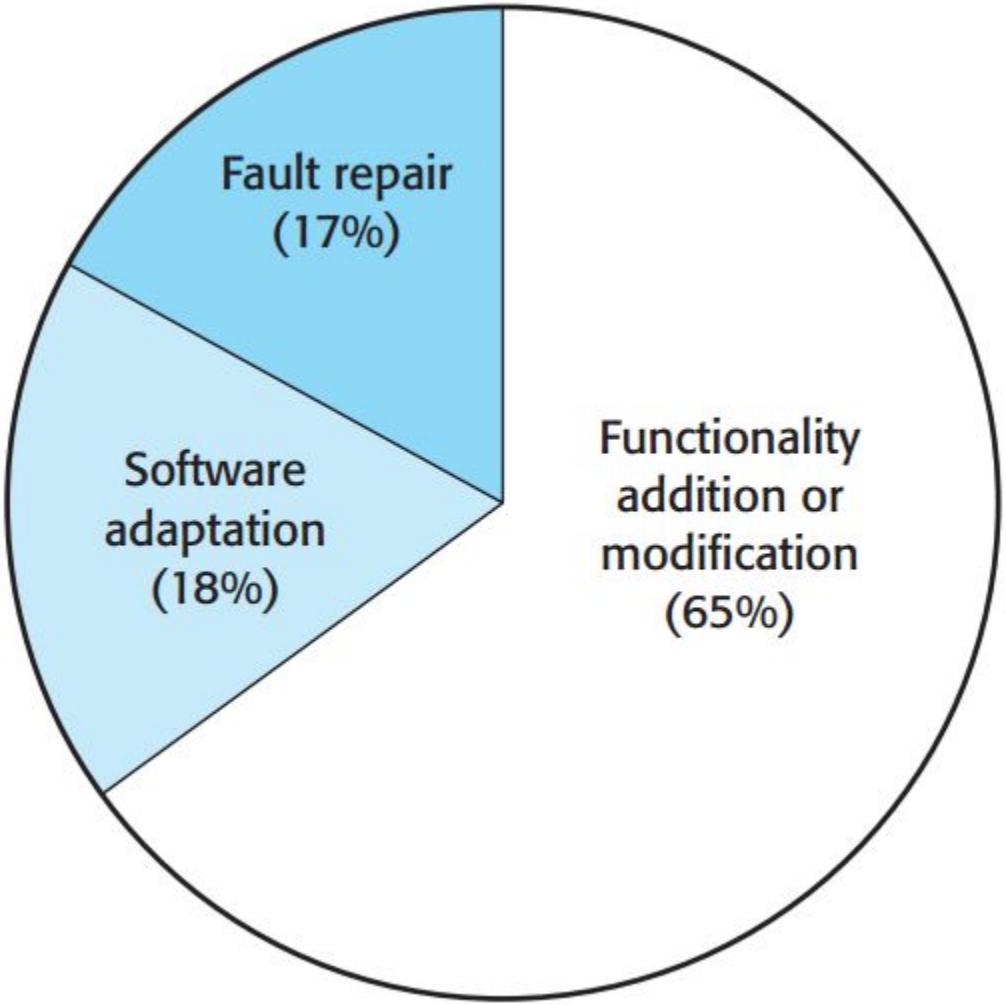
Bảo trì phần mềm

- Liên quan đến sửa đổi một chương trình sau khi nó đã được đưa vào sử dụng.
- Việc bảo trì thường không liên quan đến những thay đổi lớn đối với kiến trúc của hệ thống.

Các kiểu bảo trì

1. Bảo trì để sửa chữa lỗi phần mềm
2. Bảo trì để thích nghi phần mềm với một môi trường hoạt động khác
3. Bảo trì để bổ sung hoặc sửa đổi chức năng của hệ thống

Phân phối của các kiểu Bảo trì



Chi phí bảo trì

- Thông thường lớn hơn chi phí phát triển
- Bị ảnh hưởng bởi cả yếu tố kỹ thuật và phi kỹ thuật.
- Phần mềm có tuổi thọ càng cao thì chi phí bảo trì càng cao (ví dụ: ngôn ngữ, trình biên dịch cũ v.v.).

Các yếu tố chi phí bảo trì

- Sự ổn định của nhóm
- Trách nhiệm hợp đồng
- Kỹ năng nhân viên
- Tuổi thọ và cấu trúc chương trình

Tài liệu tham khảo

- ***Software engineering: A practitioner's approach***, Part 1 & Part 2, Roger S. Pressman, McGraw-Hill Higher Education, 2010. (#000021579)
- *Software Engineering*, Ian Sommerville, 10th Edition, 2016
- *Kỹ nghệ Phần mềm*, TS Lê Văn Phùng, Nhà xuất bản thông tin và truyền thông, 2014
- *Nhập Môn Công Nghệ Phần Mềm*, Phạm Thị Quỳnh