

# PHÁT TRIỂN VẬN HÀNH BẢO TRÌ PHẦN MỀM

ThS. NGUYỄN THỊ THANH TRÚC

# Nội dung (Chương 8)

---



**CÔNG CỤ BẢO TRÌ**



**CÔNG CỤ HỖ TRỢ KIỂM THỬ**



**CÔNG CỤ HỖ TRỢ QUẢN LÝ CẤU HÌNH**



**SEMINAR**



---

## **Chương 8:** **CÁC CÔNG CỤ BẢO TRÌ**

### **8.1 CÔNG CỤ BẢO TRÌ**

#### **8.1 CÔNG CỤ HỖ TRỢ KIỂM THỬ**

#### **8.2 CÔNG CỤ HỖ TRỢ QUẢN LÝ CẤU HÌNH**

# CÁC CÔNG CỤ

---

## ❑ CÔNG CỤ BẢO TRÌ

- Giới thiệu & Định nghĩa
- Điều kiện cho chọn lựa công cụ

## ❑ Taxonomy of tools

## ❑ Công cụ đọc hiểu và reverse engineering

- Program Slicer
- Static Analyser
- Dynamic Analyser
- Data Flow Analyser
- Cross-Referencer
- Dependency Analyser
- Transformation Tool

## ❑ CÔNG CỤ HỖ TRỢ KIỂM THỬ

- Công cụ mô phỏng giả lập (Simulator)
- Bộ phát sinh test case (Generator)
- Bộ phát sinh Test Paths (Generator)

## ❑ CÔNG CỤ ĐỂ HỖ TRỢ QUẢN LÝ CẤU HÌNH

- Source Code Control System
- Other Utilities

# Criteria for Selecting Tools

---

- ☐ *Capability*
- ☐ *Features*
- ☐ *Cost and benefits*
- ☐ *Platform*
- ☐ *Programming language*
- ☐ *Ease of use*
- ☐ *Openness of architecture*
- ☐ *Stability of vendor*
- ☐ *Organizational culture*

# Tiêu chí chọn lựa công cụ

---

- ❑ Có một vài nhà cung cấp phát triển mở rộng thị trường các công cụ rất đa dạng hỗ trợ bảo trì phần mềm. Một số yếu tố khi xem xét chọn lựa
  - Khả năng: hỗ trợ tác vụ thực thi (tính tự động, hay làm tay)
  - Chức năng: xem xét tính năng tự động
  - Chí phí và lợi ích:
  - Platforms: Win, Linux, ...
  - Ngôn ngữ lập trình: hỗ trợ ngôn ngữ Java, Ada, C, C++, Cobol, Fortran, Modula-2, Lisp and Prolog, ...
  - Tính dễ dụng: ví dụ: command line or menu-driven
  - Tính mở của kiến trúc: tính mở rộng và khả chuyển của CASE-tools
  - Tính ổn định của nhà cung cấp
  - Văn hoá tổ chức: a working culture và work patterns. Để tăng cơ hội công cụ được chấp nhận bởi người dùng cuối, cần thiết xem xét đến văn hoá và mẫu công việc

# Taxonomy of Tools

---

- ☐ Program understanding and reverse engineering
- ☐ Testing
- ☐ Configuration management
- ☐ Documentation and measurement.

# *Taxonomy of Tools*

---

- ❑ Phân loại tác vụ cho công cụ được thảo luận dựa trên :
  - Khả năng nắm bắt chương trình và reverse engineering
  - Kiểm thử
  - Quản lý cấu hình
  - Sưu liệu và độ đo.
- ❑ Đọc thêm tài liệu giới thiệu về **Taxonomy of Tools**



# Công cụ đọc hiểu và reverse engineering

---

- ❑ Program Slicer
- ❑ Static Analyser
- ❑ Dynamic Analyser
- ❑ Data Flow Analyser
- ❑ Cross-Referencer
- ❑ Dependency Analyser
- ❑ Transformation Tool
- ❑ **Yêu cầu các nhóm**
  - Xem định nghĩa các công cụ này ở ebook
  - Tìm hiểu các công cụ trên → tìm phần mềm **nguồn mở** hỗ trợ các tính năng công cụ này.
  - Xem xét các CASE-tools có sẵn hỗ trợ tính năng này

# Tools for Comprehension and Reverse Engineering

---

## ❑ *Program Slicer*

## ❑ *Static Analyser*

- allows general viewing of the program text - serves as a **browser**; generates summaries of contents and usage of selected elements.

## ❑ *Dynamic Analyser*

- Tracer, execute path

# Tools for Comprehension and Reverse Engineering

---

- ❑ ***Data Flow Analyser***
- ❑ ***Cross-Referencer***
- ❑ ***Dependency Analyser***
- ❑ ***Transformation Tool***

# *Công cụ hỗ trợ kiểm thử*

---

- ☐ ***Simulator***
- ☐ ***Test Case Generator***
- ☐ ***Test Paths Generator***
- ☐ ***Xem lại bài đọc thêm về kiểm thử***

# *Công cụ hỗ trợ quản lý cấu hình*

---

- ❑ ***Hệ thống kiểm soát mã nguồn (Source Code Control System)***
  - ***Xem thống kê các tools tại***  
***<http://www.daveeaton.com/scm/CMTools.html>***

# *Công cụ hỗ trợ sưu liệu và độ đo*

---

- ☐ ***Sưu liệu (Documentation)***
- ☐ ***Đánh giá độ phức tạp (Complexity Assessment)***

# Tài liệu tham khảo

---

- ❑ Configuration Management Yellow Pages :  
[http://www.cmtoday.com/yp/configuration\\_management.html](http://www.cmtoday.com/yp/configuration_management.html)
- ❑ <http://www.sei.cmu.edu/legacy/scm>
- ❑ CM Community : [www.cmcrossroads.com](http://www.cmcrossroads.com)
- ❑ Configuration Management Principles and Practice, Anne Mette Jonassen Hass, Addison Wesley.
- ❑ Configuration Management with CVS and Open Source Tools, Derek Clifford

# Thảo luận Checklist (1/2)

---

## ❑ Chương trình

### ○ Source code

- ✓ Độ phức tạp
- ✓ Tính dễ đọc/ dễ hiểu
- ✓ Dễ bảo trì
- ✓ Code standard
- ✓ Cài đặt chương trình

### ○ Chạy chương trình:

- ✓ Loại ứng dụng, Platforms ?
- ✓ Chạy tốt, ổn định
- ✓ Chức năng đầy đủ, giao diện thân thiện người dùng
- ✓ Lỗi chương trình
- ✓ Hiệu năng chương trình

## ❑ Sơu liệu

- Độ phức tạp
- Tính dễ đọc/dễ hiểu
- Document standard
- Đặc tả chức năng, hệ thống, dữ liệu, giao diện ...
- Hướng dẫn sử dụng (User guide), cài đặt



## Thảo luận Checklist (2/2)

---

### ☐ Kiểm thử

- Vận dụng các loại kiểm thử
- Công cụ

### ☐ Cài đặt: tools?

### ☐ Vận dụng các loại **Công cụ (Tools)** : phát triển, dự án, tương tác phối hợp, cấu hình

### ☐ Qui trình

### ☐ Phối hợp làm việc (Co-ordination)

### ☐ FeedBack (phản hồi) Khách hàng

## ***Đánh giá và thực hiện checklist***

---

- ☐ Upload chương trình vào ngày **THI** sau khi thi xong 23h
  - ☐ Vì vậy các nhóm phải upload source code cho nhóm chấm trước ngày thi từ 10 ngày, trong khoảng thời gian này các nhóm sẽ chấm rồi đính kèm checklist nộp chung với source code + Báo cáo lên course
  - ☐ Test vòng quanh 3 nhóm
    - o Nhóm 1 test :2,3,4
    - o Nhóm 2 test: 3,4,5
    - o Nhóm 3: test 4,5,6
    - o ...
    - o Nhóm 18 test: 19,20,1
  - ☐ Các nhóm thực hiện đánh giá và gửi các checklist: vd: checklist1\_234.xls (2 sheet 1, 3 sheet 2, ...)
  - ☐ Các nhóm chuẩn bị báo cáo thuyết trình:
  - ☐ Evaluation: 50% customer group+ 30% lecturer + 10% other group + 10% diligent
- 

☐ Lưu ý: customer group đánh giá phải có checklist