**DANH SÁCH ĐỀ TÀI**

**MÔN ĐẢM BẢO CHẤT LƯỢNG VÀ KIỂM THỬ PHẦN MỀM**

# YÊU CẦU

* SV viết báo cáo cho đề tài (thang điểm 8).
* Riêng đề tài tìm hiểu tool cần có thêm demo và được chấm trực tiếp (thang điểm 10).
* Định dạng file báo cáo: font chữ Times New Roman; 13pt; First line: 1.2cm; Spacing-before: 6pt; Line spacing: Multiple 1.3
* Cấu trúc của file báo cáo:

- Trang bìa cần ghi: tên đề tài, nhóm, họ tên sv, bảng phân công công việc (có ngày bắt đầu thực hiện).

- Danh mục các bảng, đồ thị, sơ đồ (nếu có)

- Mục lục

- Phần nội dung: gồm các mục và tiểu mục, được đánh số bằng các nhóm hai hoặc ba số, cách nhau một dấu chấm. Ví dụ:

1...

1.1. ...

1.1.1. ...

1.1.2. ...

1.2. ...

- Phần kết luận: ghi nhận xét/ý nghĩa mà SV cảm nhận được sau khi làm đề tài này.

- Danh mục tài liệu tham khảo

* SV có phần nào chưa rõ hoặc cần hướng dẫn thêm thì email hoặc trao đổi trực tiếp với GV.
* SV có thể chọn đề tài khác ngoài danh sách này, khi đó SV phải gởi yêu cầu đề tài cho GV duyệt trước.
* Số SV/nhóm tùy thuộc vào từng đề tài.

**Khi đăng ký đề tài, chỉ cần trưởng nhóm đại diện gởi email cho GV. Email có subject ghi theo định dạng “LOPHP\_DangKyDeTai” (LOPHP tuỳ theo lớp học phần SV của sinh viên, vd/ DHKTPM13A\_DangKyDeTai). Nội dung email gồm: danh sách thành viên trong nhóm, tên đề tài (*chú ý cách hành văn khi email cho GV*).**

Nhóm trưởng nộp bài qua email cho GV với subject email là **LOPHP\_STTNhom\_NopBaoCaoDeTai.** Thời gian nộp: **tuần LT10, riêng đề tài tìm hiểu tool chấm trực tiếp vào tuần TH9**. Nội dung nộp cần có: file báo cáo, các tài liệu tham khảo khác có sử dụng trong đề tài (nếu có). Tên file báo cáo đặt theo mẫu: **STTNhom\_TenDeTai**.

Các lỗi SV hay gặp khi làm đề tài:

* Bài dịch quá sát nghĩa, không dùng thuật ngữ chuyên ngành (đọc xong không ai hiểu gì ☹).
* Không dùng thống nhất thuật ngữ (vd/ kiểm tra, kiểm thử, thử nghiệm, kiểm nghiệm…).
* Lấy thông tin trên các forum, các nguồn không đáng tin cậy…
* Không làm theo yêu cầu đề tài.
* Làm nhóm nhưng không review với nhau; hoặc người làm phần sau không kế thừa kết quả của phần trước.

# DANH MỤC ĐỀ TÀI

| Mã | Tên đề tài | Nội dung | Tài liệu tham khảo | Số SV/nhóm |
| --- | --- | --- | --- | --- |
|  | Software quality infrastructure components ([xem chi tiết](#DT01)) | * Quality support devices * Configuration management | - Galin. *Software Quality Assurance*. Chapter 15  - Ian Sommerville. *Software engineering*. Ninth Edition. Addison-Wesley. Chapter 25  - Slide bài giảng | 3 SV |
|  | Management components of software quality ([xem chi tiết](#DT02)) | * Project progress control * Software quality metrics * Software quality costs | - Galin. *Software Quality Assurance*. Chapter 20, 21, 22  - Ian Sommerville. *Software engineering*. Ninth Edition. Addison-Wesley  - Slide bài giảng | 3 SV |
|  | Standards, certification and assessment ([xem chi tiết](#DT03)) | * Quality management standards * SQA project process standards | - Galin. *Software Quality Assurance*. Chapter 23, 24  - Ian Sommerville. *Software engineering*. Ninth Edition. Addison-Wesley  - Slide bài giảng | 2 SV |
|  | GUI testing (Aplication) ([xem chi tiết](#DT06)) | * GUI testing | Paul zGerrard. *Testing GUI applications,* 1997 | 2 SV |
|  | Usability testing ([xem chi tiết](#DT07)) | * Usability testing | J. Rubin. *Handbook of Usability testing*, Wiley Technical Communication Library, 2008 | 4 SV |
|  | User interface and Functional testing in web-based application ([xem chi tiết](#DT08)) | * User interface tests * Functional tests | Hung Q. Nguyen et al. *Testing Applications on the Web*. Chapter 10, 11 | 2 SV |
|  | Database testing ([xem chi tiết](#DT09)) | * Testing methods | Hung Q. Nguyen et al. *Testing Applications on the Web*. Chapter 14 | 2 SV |
|  | Web security testing ([xem chi tiết](#DT10)) | * Security solution basics * Common vulnerabilities and attacks * Testing for security | Hung Q. Nguyen et al. *Testing Applications on the Web*. Chapter 18 | 3 SV |
|  | Performance testing ([xem chi tiết](#DT11)) | * Performance testing concepts * Performance testing key factors * Three phases of performance testing | Hung Q. Nguyen et al. *Testing Applications on the Web*. Chapter 19 | 2 SV |
|  | Testing mobile web applications ([xem chi tiết](#DT12)) | * Mobile web application platform * Testing mobile versus desktop web applications * Various types of tests * Mobile testing support tools | Hung Q. Nguyen et al. *Testing Applications on the Web*. Chapter 6, 20 | 3-4 SV |
|  | Test tools ([xem chi tiết](#DT13))  Nhóm đăng ký tool cần ghi rõ chọn tool nào sau đây:   * NUnit * JUnit * CPPUnit * TestComplete * Mantis * [Bugzilla](http://www.testingvn.com/viewforum.php?f=76) * [Quick Test Pro](http://www.testingvn.com/viewtopic.php?f=61&t=4384) * Selenium * Watir * Robotium * Cucumber * Jmeter * SoapUI * TestLink * SilkTest * … | * Giới thiệu * Tính năng * Cài đặt * Demo | [www.google.com](http://www.google.com)  www.youtube.com | 2-3 SV/tool  (trừ 3 tool đầu 1SV/tool) |
|  | Software testing process in Agile development ([xem chi tiết](#DT14)) | * Background * Problem definition/goals * Software testing process in Agile development | Ahsan Nawaz, Kashif Masood Malik. *Software testing process in Agile development* | 3 SV |
|  | Exploratory testing ([xem chi tiết](#DT15)) | * Khái niệm * Cách thức thực hiện * Ví dụ * Lợi ích * Khó khăn | - Lee Copeland. A *Practitioner's Guide to Software Test Design*. Chapter 13  Tham khảo thêm:  - Cem Kaner. *A Tutorial in Exploratory Testing*, 2008  - Bach, James. *Exploratory Testing Explained.* v.1.3 16 April 2003 | 1 SV |

# MÔ TẢ CHI TIẾT ĐỀ TÀI

## DT 01. Software quality infrastructure components (Các thành phần cơ bản của chất lượng phần mềm)

* Nêu khái niệm, ví dụ, lợi ích, cách quản lý template, checklist.
* Tại sao cần phải quản lý cấu hình? Nêu và giải thích các hoạt động quản lý cấu hình phần mềm.
* Demo một công cụ về quản lý cấu hình (SVN, GitHub,…)

## DT 02. Management components of software quality (Các thành phần quản lý của chất lượng phần mềm)

* Nêu mục tiêu của kiểm soát tiến độ dự án (project progress control). Nêu các thành phần khi kiểm soát tiến độ dự án. Demo một công cụ bất kỳ trong quản lý tiến độ dự án.
* Nêu mục tiêu chung khi thu thập các độ đo. Phân loại các độ đo chất lượng phần mềm. Hiểu và giải thích được 3-5 công thức cụ thể của 2 loại độ đo.
* Nêu mục tiêu quản lý chi phí chất lượng phần mềm. Nêu và giải thích các mô hình chi phí chất lượng phần mềm (mô hình cổ điển và mở rộng).

## DT 03. Standards, certification and assessment (Các chuẩn, chứng chỉ và đánh giá)

* Lợi ích của việc sử dụng chuẩn
* Phân loại các lớp chuẩn (chuẩn quản lý, chuẩn tiến trình)
* Khái niệm ISO, ISO 9000, ISO 9001, ISO 9000-3
* Quy trình cấp chứng nhận ISO (tìm trên google)
* CMM/CMMI và các level
* So sánh ISO và CMM/CMMI
* Cấu trúc của chuẩn IEEE
* Sơ lược về:
  + IEEE/EIA Std 12207 – Software Life Cycle Processes
  + IEEE 829 — Software Test Documentation
  + IEEE Std 1012 – Verification And Validation
  + IEEE Std 1028 – Reviews

## DT 04. GUI Applications testing (Kiểm thử ứng dụng có giao diện)

* Testing GUI Applications (Paul Gerrard)
* Minh hoạ bằng các ví dụ thực tế về những GUI tốt và chưa tốt theo tiêu chí của tài liệu.

## DT 05. Usability testing

* Concept? Goal? (p.48)
* Basic elements of usability testing (p.52)
* Types of usability tests (p.54)
* Test plan (p.92): chú ý phần Bảng câu hỏi khảo sát (cho ví dụ) và Chọn người dùng đại diện (participant)
* The process (p.89): nêu ngắn gọn-đầy đủ, có ví dụ

## DT 06. User interface and Functional testing in web-based application (Kiểm thử giao diện và chức năng cho ứng dụng web)

* User interface design testing (p.216): Profiling the Target User, Considering the Design
* Usability and accessibility testing (p.247)
* Testing methods in functional tests (p.257)
* Minh hoạ bằng các ví dụ thực tế tốt và chưa tốt theo tiêu chí của tài liệu.

## DT 07. Database testing (Kiểm thử cơ sở dữ liệu)

* Test methods (p.328)
* Cho các ví dụ cụ thể để minh hoạ kỹ thuật.

## DT 08. Web security testing (Kiểm thử bảo mật web)

* Security solution basics (p.424)
* Common vulnerabilities and attacks (p.435)
* Testing for security (p.449)
* SV tự đọc thêm các phần còn lại trong chương để nắm kiến thức.

## DT 09. Performance testing (Kiểm thử hiệu năng)

* Performance testing concepts (p.481)
* Performance testing key factors (p.487)
* Three phases of performance testing (p.493 - 522)
* SV chọn một công cụ để demo test.
* SV tự đọc thêm các phần còn lại trong chương để nắm kiến thức.

## DT 10. Testing mobile web applications (Kiểm thử web trên thiết bị di động)

* Mobile web application platform (p.149)
* Testing mobile versus desktop web applications (p.528)
* Various types of tests (p.536)
* Mobile testing support tools (p.546)
* Giáo trình có thể đưa ra những công nghệ mobile mà hiện giờ không còn được sử dụng, SV tự tìm hiểu và trình bày những công nghệ mới.

## DT 11. Test tool (Tìm hiểu công cụ hỗ trợ kiểm thử)

Mỗi nhóm SV chọn MỘT TOOL để tìm hiểu. Nội dung trình bày gồm:

* Giới thiệu: Dùng để làm gì? Của công ty nào? Download ở đâu? Bản quyền thế nào? → xong
* Chức năng: Hỗ trợ công việc gì? Có những ưu điểm nổi bật nào? → xong
* Cài đặt: Cài đặt thế nào? Môi trường triển khai ra sao? → xong
* Hướng dẫn sử dụng. Có chức năng gì …→ xong
* Khó khăn, đánh giá, bảng so sánh với những PM cùng nhóm: đánh giá về công cụ tìm hiểu → xong
* Demo: cho ví dụ test các component/ứng dụng tự chọn cho hợp lý. Selenium bao gồm một bộ công cụ IDE, Grid, Remote Control, WebDriver. Dự định làm 1 demo WebDriver. Demo Se IDE and WebDriver

## DT 12. Software testing process in Agile development (Quy trình kiểm thử trong phát triển Agile)

* Chapter 1: Background
* Chapter 2: Problem definition/goals
* Chapter 5: Software testing process in Agile development

## DT 13. Exploratory testing (Kiểm thử thăm dò)

* Khái niệm
* Cách thức thực hiện
* Ví dụ
* Lợi ích - Khó khăn

# Chú thích:

**[1] Same origin policy:** Chính sách cùng nguồn gốc là một cơ chế an ninh trong trình duyệt web để ngăn chặn các trang web hoặc tài nguyên từ nguồn không cùng nguồn gốc truy cập và tương tác với các dữ liệu từ nguồn khác. Nguồn gốc ở đây thường được xác định bằng cách so sánh giao thức, tên miền và cổng của hai trang web.

Ví dụ: Giả sử bạn đang duyệt một trang web A từ miền "https://www.example.com". Trang web A muốn tải dữ liệu từ một trang web B có nguồn gốc từ "https://api.example.com". Nếu trang web B không chia sẻ cùng nguồn gốc với trang web A, thì trình duyệt sẽ áp dụng chính sách cùng nguồn gốc và ngăn chặn trang web A truy cập dữ liệu từ trang web B. Điều này là để đảm bảo rằng trang web A không thể lợi dụng quyền truy cập vào dữ liệu nhạy cảm trên trang web B mà người dùng không được phép truy cập.

* Trong ngữ cảnh của Selenium, vấn đề xuất phát khi người kiểm thử cố gắng kiểm tra ứng dụng web trên một trang web khác nguồn gốc. Ví dụ, nếu bạn đang sử dụng Selenium để kiểm tra một trang web từ miền "https://www.example.com", nhưng Selenium được thực thi từ một miền khác, chẳng hạn "https://test.example". Chính sách cùng nguồn gốc sẽ ngăn chặn Selenium truy cập và tương tác với các thành phần của trang web từ miền "https://www.example.com".