National University of Ho Chi Minh City. HO CHI MINH

UNIVERSITY OF INFORMATION TECHNOLOGY

SYLLABUS

SE113 – SOFTWARE TESTING

1. GENERAL INFORMATION

Subject name (Vietnamese):	Kiểm thử phần mềm
Subject name (English):	Software testing
Subject code:	SE113
Type of course:	Specialization
Faculty:	Faculty of Software Engineering
Editor-in-Chief:	Nguyen Thi Thanh Truc
	Email: trucntt@uit.edu.vn
Credits:	4
Theory:	3
Practice:	1
Self learning:	90 (periods)
Prerequisite:	
Prior subjects:	Introduction to Informatics, Introduction to Software Engineering.

2. COURSE DESCRIPTION

The course introduces students to the basics of software testing. Providing students with the ability to use software testing techniques such as white box testing and black box testing. Students also learn software testing strategies such as unit level testing, integration level testing, system level testing, product acceptance testing, and regression testing.

3. COURSE GOALS

After completing this course, students are able to:

Table 1.

Symbol	Course Objectives	Program outcomes
G1	Have the knowledge of software quality and testing	*1.2.4, 1.2.9*
G2	Understand and apply popular techniques and tools to support Software Testing	*1.2.11, 1.3.5*
G3	Evaluate and analyze software quality	*3.3.2*
G4	Know how to design and execute test cases	
G5	Forming seriousness, systematic thinking and professional style about software testing	*2.4.3, 2.4.4, 2.5.2*
G6	Aware of the importance of individual contribution to the workgroup during testing for a software project.	*3.1.1, 3.1.2, 3.2.4*

4. COURSE learning outcomes

Table 2.

Course outcomes	Description	Level Teaching
G1.1	Gain knowledge of software quality	*I, T, U*
G1.2	Have the knowledge of software testing	*T, U*
G2.1	Understand and apply the software testing techniques	*T, U*

G2.2	Understand and apply common tools that support Software Testing	*T, U*
G3.1	Evaluate software quality	*I, T*
G3.2	Understand and apply software quality analysis	*T, U*
G4.1	Able to design Testcase	*I, T*
G4.2	Ability to execute both manual and automated test cases	*T, U*
G5.1	Know and understand specialized English terms of the subject	*I, T*
G5.2	Read and understand English materials related to lectures	*T, U*
G6.1	Forming and working in groups (from 2-4 students)	*I*
G6.2	Participate in group discussions and debates on the subject's project topic	*I, T*
G6.3	Analyze, design and verify software projects by team collaboration	*I, T*

5. COURSE CONTENTS, TEACHING PLAN

a. Theory

Table 3:

Lesson	Content	Outcomes	Teaching and learning activities	Assessment component
1	Introduction of course content and regulations on subjects, setting up groups.	*G5.1, G5.2*	Lectures, Students read documents and discuss on subject groups	*A1*
2	Overview of software engineering - What is software engineering? - How does the software testing contribute?	*G1.1, G2.1*	Lectures, Exercises, Discussions	*A1*
3	Software development process and software testing	*G1.1, G2.1*	Lectures, Exercises, Discussions	*A1, A5*
4	Black box software testing techniques: test case design	*G2.1, G2.2, G4.1, G4.2*	Lectures, Exercises, Discussions	*A1, A5*
5	Black box testing exercise and seminar: - Develop a program under a specification - Desigin test cases - Test other student's program using test cases	*G2.1, G2.2, G4.1, G4.2 G6.1, G6.2*	Students develop programs under a given specification and design test cases; then test other student's program	*A1, A5*
6	White box software testing techniques	*G2.1, G2.2, G4.1, G4.2*	Lectures, Exercises, Discussions	*A1, A5*

7	White box testing exercise and seminar: - Develop a program under a specification - Desigin test cases - Test other student's program using test cases	*G2.1, G2.2, G4.1, G4.2, G6.1, G6.2*	Students develop programs under a given specification and design test cases; then test other student's program	*A1, A5*
8	Evaluation of testing and reliability: - Estimation of total error count - Software reliablity growth model	*G3.1, G3.2*	Lectures, Exercises, Discussions	*A1, A5*
9	Programming tips and techniques - Coding convention - Code review - Static code analysis - Test-driven development	*G3.1, G3.2*	Lectures, Exercises, Discussions	*A1, A5*
10	Code review exercise and seminar: - Revise poorly-written code - Develop a program under a specification - Review other student's program	*G6.1, G6.2 G4.1, G4.2*	Students revise (given) a poorly-written program. Students develop programs under a given specification; then review other student's program	*A1, A5*
11	Test-driven development exercise and seminar: - Develop a program under a specification - Review other student's test cases and program	*G6.1, G6.2 G4.1, G4.2*	Students develop programs under a given specification in a test-driven style; then review other student's test cases and program	*A1, A5*

12	Software quality management and software metrics - Maintenace types - Foundation of qualicy management - Software metrics	*G3.1, G3.2*	Lectures, Exercises, Discussions	*A1, A5*
13	Bug prediction and test plan - Bug-prone program prediction - Cost-effective testing plan	*G3.1, G3.2*	Lectures, Exercises, Discussions	*A1, A5*
14	Project based discussion (exercise and seminar for testing the specification) - Propose a Web-based system - Discuss what should we test on the system	*G6.1, G6.2*	Students propose a Web-based system (only proposal with usage scenarios and mock-ups); Then, disucss what should we test it	*A1, A5*
15	Review	*G1.1, G1.2, G2.1. G2.2, G3.1, G3.2*		*A1, A5*

b. Practice

Table 4.

Session (3 hours)	Content	Course outcomes	Teaching and learning activities	Assessment component
	Lab1: Writing Blackbox testcase Lab2: Writing Unit testcase Lab3: Test Execution and Test Report Lab4: Measuring programs using software metrics	A1, A5	Students follow the report form and apply and implement the project as required	A1, A5

6. COURSE ASSESSMENT

Table 5.

Review composition	Course outcomes	Ratio (%)
A1. Process (Class tests, assignments, projects, ...)	*G1, G2, G3, G4*	*40%*
A2. Midterm		*0%*
A3. Practice	*G1, G2, G3, G4*	*10%*
A4. End of term	*G3, G4, G5*	*50%*

7. COURSE REQUIREMENTS AND EXPECTATIONS

8. LEARNING DOCUMENTS, REFERENCES

Curriculum

1. Nguyen Cong Hoan. Lecture + Slide of Software Testing. University of Information Technology (internal), 2011.
2. Glenford J. Myers, Corey Sandler, Tom Badgett. The Art of Software Testing, John Wiley & Sons, Inc., 2011

References

1. Dorothy Graham, Mark Fewster. Experiences of Test Automation: Case Studies of Software Test Automation, Pearson Education Inc., 2012
2. Julian Harty. A Practical Guide to Testing Wireless Smartphone Applications, Google Inc., 2009
3. G. Myers. The Art of Software Testing 2nd, John Wiley & Sons Inc., 2004, 234 pages
4. S. Koirala & S. Sheikh. Software Testing Interview Questions, Infinity Science Press LLC, 2008
5. Foundations of software testing istqb certification 3rd edition

9. SOFTWARE OR TOOLS TO AID PRACTICES

1. Gcc and Gcov
2. R and RStudio

Ho Chi Minh City, December 30 2023

Dean/Department	**Compiled by lecturer**
(Sign and write full name)	(Sign and write full name)