University of Computer Studies, Yangon
Faculty of Information Science
2019-2020 Academic Year

Subject Code	CS-304	Subject Name	Software Engineering
Semester	First	Course Coordinator	Dr. Khine Khine Oo
Credit	3		
Credit Hours	37.5 hours		
Weeks	15 Weeks		
Period	45 period (1 period :50 Mins) 3 period per week		

Course Description

This course will examine the two parts. The first part presents the rapid software developing techniques and the second part presents the fundamental software testing and related program analysis techniques. In particular, the agile software development methods and techniques are described and the important phases of testing will be reviewed, emphasizing the significance of each phase when testing different types of software. The course will also include concepts such as test generation, test coverage, regression testing, mutation testing, program analysis (e.g., program-flow and data-flow analysis), and test prioritization.

Course Objective

- To know the rapid application software development
- To study fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods.
- To discuss various software testing issues and solutions in software unit test; integration, regression, and system testing.
- To learn how to planning a test project, design test cases and data, conduct testing operations, manage software problems and defects, generate a testing report.
- To gain software testing experience by applying software testing knowledge and methods to practice-oriented software testing projects.
- To understand software test automation problems and solutions.
- To learn how to write software testing documents, and communicate with engineers in various forms.
- To gain the techniques and skills on how to use modern software testing tools to support software testing projects.

Learning Outcomes

At the end of this course student will:

- Gain the knowledge about the rapid software application development methods
- Explain about a range of different software testing techniques and strategies
- Have an ability to apply software testing knowledge and engineering methods to the project
- Apply modern software testing processes in relation to software development and project management.
- Create test strategies and plans, design test cases, prioritize and execute them.

Prerequisites

- Basic understanding of the software development life cycle (SDLC).
- Basic understanding of software programming using any programming language.

Major topic covered in the course

- Rapid Software Development
- Verification and Validation Testing
- Software Testing

TextBook

- Software Engineering (8th Edition) ,Ian Sommerville

Reference Book

- Introduction to Software Testing, Paul Ammann and Jeff Offutt, 2008

Software

- Java JDK 1.5 and above
- Eclipse JDK Environment Europa and Above
- JUnit Testing Tools 10.4

Learning Assessments

Exam	: 60%
Practical and Class Room Participation	: 10%
Tutorial Test	: 10%
Assignment	: 10%
Quiz	: 10%

Course Policy

Participation

Attendance is a prerequisite, not a substitute for class participation. Participation mechanisms include: (1) responding to questions asked in class, (2) initiating discussions on new points in class and (3) discussing cases and offering solutions to problems .

Tutorial Test and Quizzes

The student is expected to complete the tutorial tests and Quizzes at the scheduled time. If a tutorial test or quiz is missed, there will be no make-ups tutorial or quiz for missing student. No make

–ups test or resubmission and extra credit test are not available in this course. Tutorial tests and quizzes are based upon all learning objectives to be reached before the scheduled date.

Assignment

There will be theory and practical assignments which must be submitted. The assignment may be individual or Group. The individual assignment is individual work and tests the ability of each student. Group assignment is team work and tests the ability of collaboration of student to complete the given work.

The due dates for the given assignments are going to be declared by the instructor and there will be no make-ups or individual extensions. No make –ups Assignment or resubmission and extra credit assignment are not available in this course.

In addition to the hardcopies of assignments, electronic (and certifiably virus free) copies should be e-mailed to instructor on the date they are due.

Intellectual Honesty

By departmental policy, the discovery of plagiarism (i.e. copying from another's assignment paper or practical solution or tutorial paper) will result in a reduction of result marks of relevant students.

Lecture Plan

CS-304 **: Software Engineering** **First Semester**

Text Book **: Software Engineering (8th Edition) Ian Sommerville**

Period : 45 Periods for 15 Weeks (3 Periods * 15 Weeks)

No.	**Chapter**	**Page**	**Period**	**Remark**
Chapter 17 : Rapid System Development				
1	Introduction	391-395	1	Detail
2	17.1 Agile Methods	396-398	2	Detail
3	17.2 Extreme Programming	398-405	2	Detail
4	17.3 Rapid Application Development	405-409	2	Detail
5	17.4 Software Prototyping	409-412	1	Detail
6	Review / Tutorial/ Discussion		2	
Chapter 22 : Verification and Validation				
7	Introduction	515-519	1	Detail
8	22.1 Planning verification and validation	519-521	2	Detail
9	22.2 Software Inspections	521-527	2	Detail **Software Inspections** **Ref.[1] - Chapter 1: Introduction** Section 1.2 Software testing limitation and Terminology Section Exercise: 1.2
10	22.3 Automated Static Analysis	527-530	2	Detail
11	22.4 Verification and Formal Method	530-535	2	Detail
12	Review / Tutorial/ Discussion		2	

Chapter 23 : Software Testing				
13	Introduction	537-540	1	Detail
14	23.1 System Testing	541-547	5	Use Testing Tools (JUnit)
15	23.2 Component Testing	547-551	5	Use Testing Tools (JUnit)
16	23.3 Test case Design	551-561	5	How to design test case (**23.3.4 Path Testing** **Ref.[1]: Chapter 2 – Graph Coverage** **Section 2.1: Overview** Section 2.2: Graph Coverage Criteria Section 2.2.1: Structural Coverage Criteria Exercise 4 and 5 Section 2.3: Graph Coverage for Source Code Section 2.3.1: Structural Graph Coverage for Source Code Section 2.3.2: Data flow Graph Coverage for Source Code Section Exercises **23.3.2 Partition Testing** **Ref.[1]: Chapter 4 – Introduction Input space Partitioning** **Section 4.1: Input Domain Modeling** Section 4.1.1: Interface based Input Domain Modeling Section 4.1.2: Functionality based Input Domain Modeling Section 4.1.3: Identifying Characteristics Section 4.1.4: Choosing blocks and values Section Exercise)
17	23.4 Test Automation	561-563	3	Detail
18	Review / Tutorial/ Discussion		2	
19	Revision for all Chapters		3	
		Total	45	

Text References:

[1] **Introduction to Software Testing**by Paul Ammann, Jeff Offutt,