NITK Surathkal

Department of Computer Science & Engineering

Course Plan

Course Type: PSE

Year & Semester : VIII Sem B.Tech (CSE)

Course Name: Software Testing

Course No: CO421

Credits (L-T-P): 3 (1-0-3)

Academic Session: December 2020 - May 2021

Prerequisites (if any): CO251 (Software Engineering) and
CO255 (Software Engineering Lab)

Name of Course Instructor: Prof. P. Santhi Thilagam

Contact Details : Room number 410,
Computer Science and Engineering Building.

Email-ID : evensemester2021@gmail.com

(All the course related queries will be addressed only through this e-mail id)

Evaluation Scheme:

Midsem Examination : 20%
Project Evaluation : 50% [Test plan: 10%, Test Cases: 15%, Test Execution: 25%]
Endsem Examination : 30%

Assessment Pattern : Various rubrics are designed to evaluate the performance of the students

Level No	Knowledge Level	Evaluation Component			Assessment (%)
		Midsem Exam (20%)	Project (50%)	Endsem Exam (30%)	
K1	Remember	25%	0%	40%	17%
K2	Understand	25%	20%	20%	21%
K3	Apply	20%	20%	20%	20%
K4	Analyze	20%	20%	20%	20%
K5	Evaluate	10%	20%	0%	12%
K6	Create	0%	20%	0%	10%
					100%

Course Objectives:

1. Understand the concepts and theory related to software testing.
2. Understand different testing techniques used in designing test plans, developing test suites, and evaluating test suite coverage
3. Apply software testing knowledge and methods on practice-oriented software testing projects.
4. Learn to use modern software testing tools to support software testing projects.

Course (Learning) Outcomes (COs):

CO1: To instruct students on the importance of software testing,
CO2: To expose them to various testing-related tools and techniques, and
CO3: To explain and exercise the decision process and situations in which certain tests should and should not be used.

Mapping of CO's with PO's:

(Strength of Correlation: S: Strong, M: Medium, W: Weak)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	M	S	S	M	W	M	M	S	M	S	W	M
CO2	S	W	W	M	S	S	M	W	M	S	M	M
CO3	M	S	M	S	M	S	S	W	M	W	S	S

Project Instructions:

1. Students should form a team of two members (self selected; team name list due on 5th January 2021) .
2. Groups can select any one type of open source applications [Refer the open source applications section].
   a. Object Oriented
   b. Component based
   c. Concurrent
   d. Distributed
   e. GUI
   f. Web based or Mobile based applications
3. Students are free to choose any of the Open source software testing tools [Refer the open source tools section].
4. Students should submit all reports or code (as applicable) on the due date.
5. No extension of due dates is permitted.

Project Stages

1. Team Formation — Last Date: 5th January 2021

Students are advised in the team creation process by choosing which other student he/she would like to be partnered with. Once the teams have been formed, their first step is to choose an application to be tested. Students are asked to select medium-sized open source applications for testing. They must be large enough to provide an adequately sized test bed. Applications with roughly 100-200 classes have been found to be appropriate for this project, and students are not limited by programming language. The final step in the application selection process is to have the program approved by the instructor.

2. Selection of Open source Application and Types of Testing

Last Date: 14th January 2021

Once the teams have been formed, their first step is to choose an application to be tested. Students are asked to select medium-sized open source applications for testing. They must be large enough to provide an adequately sized test bed, but not so large as to overwhelm the teams. Applications with roughly 100-200 classes have been found to be appropriate for this project, and students are not limited by programming language. Projects with a high level of documentation are preferred, as this provides a basis for acceptance tests and the documentation itself can also be tested. The final step in the application selection process is to have the program approved by the instructor.

Students can select a wide range of open source applications, including both traditional and mobile applications, applications written for Unix, Windows, and OSX operating systems, and applications written by both small groups and organizations the size of Mozilla. For example, A few of the applications include JBrick , tuxGuitar , Notepad++ and components of Firefox. Open source applications are given in the references section of this document. Selection of Testing types: Each Group is asked to choose two types of Functional testing and two types of Non-functional Testing for the chosen application.

3. Preparation of Test Plan — Last Date: 11th February 2021

The next major deliverable is the project test plan [Refer the sample Test Plan Template. Test plans are expected to significantly differ from one another due to the wide variety of applications selected. Groups are encouraged to make the best test plans possible. Students are allowed to build on to and deviate from this test plan as necessary.

4. Generation of Test Cases for each types of testing — Last Date: 18th March 2021

The next major deliverable is the project test cases [Refer the Sample Test Case Template]. Groups are encouraged to make the test cases appropriately. Students are allowed to include more test cases as necessary.

5. Test Execution — Last Date: 1st April 2021

Groups are encouraged to execute the test cases on the chosen application using selected tools and record the observation about the behaviour of the application on each test cases.

6. Preparation of Test Report — Last Date: 8th April 2021

Groups are asked to submit their final Test Report, which details not only the findings of their testing, but also the specific tests used and relevant data. Students are also encouraged to state why they chose and carried out particular tests and provide a rationale for the processes and techniques they employed.

A. Different Types of Software Testing

Each Group is asked to choose two types of Functional testing and two types of Non- functional Testing for the chosen application. Given below is the list of some common types of Software Testing:

Functional testing types	Non-functional testing types
1. Unit testing	1. Performance testing
2. Integration testing	2. Load testing
3. System testing	3. Stress testing
4. Sanity testing	4. Volume testing
5. Smoke testing	5. Security testing
6. Interface testing	6. Compatibility testing
7. Regression testing	7. Install testing
8. End-End testing	8. Recovery testing
9. White box testing	9. Reliability testing
10. Black Box testing	10. Usability testing
11. User Acceptance testing	11. Compliance testing
12. Component testing	12. Localization testing
	13. Penetration testing
	14. Compatibility testing
	15. Maintainability testing
	16. Scalability testing
	17. Performance testing
	18. Documentation testing

B. Test Plan

The test plan serves as a blueprint to conduct software testing activities as a defined process which is minutely monitored and controlled by the test manager. Important aspects like test estimation, test scope, Test Strategy are documented in Test Plan.

In software testing, a test plan gives detailed testing information regarding an upcoming testing effort. Follow the seven steps below to create a test plan as per IEEE 829.

1. Analyze the product
2. Design the Test Strategy
3. Define the Test Objectives
4. Define Test Criteria
5. Resource Planning
6. Plan Test Environment
7. Schedule & Estimation
8. Determine Test Deliverables

Sample Test Plan Template

Items in a Test Plan Template	What do they contain
Scope	Test scenarios/Test objectives that will be validated.
Out of scope	Enhanced clarity on what we are not going to cover
Assumptions	All the conditions that need to hold true for us to be able to proceed successfully
Schedules	Test scenario prep
	Test documentation- test cases/test data/setting up environment
Roles and Responsibilities	Team members are listed
	Who is to do what
Deliverables	What documents(test artifacts) are going to produce at what time frames
	What can be expected from each document
Environment	What kind of environment requirements exist
	Who is going to be in charge
	What to do in case of problems
Tools	For example: JIRA for bug tracking
Defect Management	Who are we going to report the defects to
	How are we going to report
	What is expected- do we provide screenshot?
Risks and Risk Management	Risks are listed
	Risks are analyzed- likelihood and impact is documented
Exit criteria	When to stop testing

C. TEST CASE

The test cases are all about “How we are going to test a requirement”. A TEST CASE is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application.

Fields in test Cases:

a) Precondition – state of the AUT (the state in which the AUT needs to be for us to get started)

b) Input – data entry steps. For these steps it is important to note what kind of input info is required – Test data

c) Validation point/trigger/action – what is causing the validation to happen? (Click of a button or toggle or the link access. Make sure there is at least one validation point to a test case- otherwise it is all going to be data entry with nothing to look for. Also to ensure that we have enough modularity, try not to combine too many validation points into one test case. 1 per test case is optimum.)

d) Output – expected result.

e) Postcondition – This is additional information that is provided for the benefit of the tester, just to make the test case more insightful and informative. This includes an explanation of what happens or what can be expected of the AUT once all the test case steps are done.

Characteristics of a good test case:

1. Accurate: Exacts the purpose.
2. Economical: No unnecessary steps or words.
3. Traceable: Capable of being traced to requirements.
4. Repeatable: Can be used to perform the test over and over.
5. Reusable: Can be reused if necessary.

Sample Test Case Template

A test case can have the following elements. Note, however, that a test management tool is normally used by companies and the format is determined by the tool used.

Test Suite ID	The ID of the test suite to which this test case belongs.
Test Case ID	The ID of the test case.
Test Case Summary	The summary / objective of the test case.
Related Requirement	The ID of the requirement this test case relates/traces to.
Prerequisites	Any prerequisites or preconditions that must be fulfilled prior to executing the test.
Test Procedure	Step-by-step procedure to execute the test.
Test Data	The test data, or links to the test data, that are to be used while conducting the test.
Expected Result	The expected result of the test.
Actual Result	The actual result of the test; to be filled after executing the test.
Status	Pass or Fail. Other statuses can be ‘Not Executed’ if testing is not performed and ‘Blocked’ if testing is blocked.
Remarks	Any comments on the test case or test execution.
Created By	The name of the author of the test case.
Date of Creation	The date of creation of the test case.
Executed By	The name of the person who executed the test.
Date of Execution	The date of execution of the test.
Test Environment	The environment (Hardware/Software/Network) in which the test was executed.

Refer : https://www.softwaretestinghelp.com/test-case-template-examples/

D. TEST REPORT`

Test Report is a document which contains

1. A summary of test activities and final test results
2. An assessment of how well the Testing is performed.

Based on the test report, the stakeholders can

1. Evaluate the quality of the tested product
2. Make a decision on the software release. For example, if the test report informs that there're many defects remaining in the product, the stakeholder can delay the release until all the defects are fixed.

Test Summary includes the summary of testing activity in general. Information detailed here includes

1. The number of test cases executed
2. The numbers of test cases pass
3. The numbers of test cases fail
4. Pass percentage
5. Fail percentage
6. Comments

One of the most important information in Test Report is defect. The report should contain following information

1. Total number of bugs
2. Status of bugs (open, closed, responding)
3. Number of bugs open, resolved, closed
4. Breakdown by severity and priority

E. References

https://www.softwaretestinghelp.com/free-online-software-testing-qa-training-course/

Open Source Testing Tools:

1. www.opensourcetesting.org
2. www.testingfaqs.org
3. www.qaforums.com
4. https://www.qatestingtools.com/opensource_testing_tools
5. Selenium Tool- https://www.seleniumhq.org/
6. Spin Tool- http://spinroot.com/spin/whatispin.html

a. Functional Testing Tools

1. Ranorex Studio — all-in-one functional test automation for desktop, web, and mobile apps with built-in Selenium WebDriver.
2. Selenium - Popular Open Source Functional Testing Tool
3. QTP - Very user-friendly Functional Test tool by HP
4. JUnit- Used mainly for Java applications and this can be used in Unit and System
5. Testing soapUI - This is an open source functional testing tool, mainly used for Web service testing. It supports multiple protocols such as HTTP, SOAP, and JDBC.
6. Watir - This is a functional testing tool for web applications. It supports tests executed at the web browser and uses a ruby scripting language

b. Non-Functional Testing Tools:

1. Forecast
2. JMeter
3. Load Complete
4. Loadrunner
5. Loadster
6. Loadstorm
7. Loadtracer
8. Neoload
9. vPerformer
10. WebLoad Professional
11. Webserver Stress Tool Forecast

Open source Applications:

1. http://code.google.com/p/jbrick/
2. http://sourceforge.net/projects/tuxguitar/
3. http://notepad-plus-plus.org
4. http://www.mozilla.org/en-US/_refox/
5. http://jmeter.apache.org/
6. http://tsung.erlang-projects.org
7. https://github.com/pcqpcq/open-source-android-apps
8. https://www.datamation.com/osrc/article.php/3694506/50-Open-Source-Desktop-Projects-Good-Downloads.htm
9. https://blogs.oracle.com/pranav/100-great-freewareopen-source-tools-and-applications-for-windows
10. https://www.makeuseof.com/tag/open-source-desktop-apps-android/
11. https://opensource.com/tools/enterprise-resource-planning
12. https://opensource.com/life/13/7/4-text-editors-linux
13. https://opensource.com/life/15/6/eight-open-source-data-visualization-tools
14. https://opensource.com/alternatives/gmail
15. https://digital.com/blog/open-source-business/
16. https://www.creativebloq.com/web-design/10-most-exciting-open-source-projects-web-4132351
17. https://sourceforge.net/directory/os:windows/?q=open+source+php+web+application

Prof. P. Santhi Thilagam
28/12/2020

*******GOOD LUCK*******