University of Pretoria
Software Engineering - COS 301

Testing Policy

Team Singularity
3 May 2019

Team Members:

Richard McFadden	**u17026662**
Adrian le Grange	**u17056782**
Jarrod Goschen	**u17112631**
Alessio Rossi	**u14137934**
Kyle Olivier	**u15001319**

Contents

1 Definition of Testing

We as a team see testing as a way to ensure that the software meets the requirements of the client and the COS 301 lecturers. We also believe testing can help reduce time and effort during the development stage of the project.

In this regard the tests will assure that we accomplish building and integrating software that meets the requirements set out beforehand.

It is also an efficient and effective way of making sure you do not modify the system and stray away from the requirements.

2 Description of the testing process

The testing process involves 3 levels of testing namely Unit-, Integration- and System-testing. Testing will follow a bottom-up methodology approach.

Level	Branch	Objective	Key Areas
Unit	On feature branches	To test the functions of the feature and ensure it works.	Functionality
System	Master	To ensure that the system as a whole functions and flows together nicely	Perfomance Functionality Usability Maintainability Scalability
Integration	Development	Test to ensure that system works after integrating with other subsystems	Fucntionality Compatbility

Tests should be written before any development starts and it should be based on the requirements of the system. It should happen one iteration before development and members should understand what the tests entail. This will ensure that members develop the system to work with the tests and not make the tests work with the system.

Karma and Jasmin will be used for unit tests on Angular.js and applicable Node.js projects. Unittest is a library that will be used to do unit tests on python. All of these tests are run on Travis CI on every commit.Every new feature will undergo regression testing once implemented.

Once the system starts integrating together tests will be reconsidered and changed as need be to accommodate the new additions.

3 Testing evaluation

Tests will be evaluated based on outcome and via static-code analysis. Our project includes a lot of intercommunication between subsystems and these outcomes can inform us whether or not all subsystems are communication correctly. Thus a test will only be deemed as Passed when the expected outcome matches the actual outcome.

All tests are of type functional and due to this we, as a team, will use a Pass/Fail method for evaluation. A success rate of 97% is required between all tests.

Tests will have certain priorities tied to them.

Priority	Tests	Objective
High	Any tests that involves API functionality or Server Functionality	These functionalities may not be broken and at all times be working.
Medium	Tests involving generating or encoding of necessary data needed.	These functions should always be up but if they fail it won't be severe
Low	All the nice-to-have features	These tests are only for when we start feature adding

4 Testing improvements

Testing will be evaluated every week when the team meets up. Just to ensure the development is in-line with the tests. Bi-weekly meetings with clients will also be used to determine whether the tests follow the requirements and to change as needed based on our iterative design process.