Andrew Arnopoulos

Professor Boese

Software Engineering

13 March 2015

Homework 3

ID: 317152

Avatar: lion

* 1. Unit testing uses simple tests that check a small part of someone’s code. These tests are helpful because they automate mundane parts of testing code. It is also great for testing corner cases of a particular part of someone’s code. This also makes writing code easier, after writing the test cases, because it ensures that a person’s code works without manually checking the output. Integration testing tests larger code modules that have, hopefully, been unit tested, with other code modules. It is used to verify functional parts or a person’s code and major design themes. This is useful when testing things, such as, a custom objects, SDK’s, frameworks, etc. For example someone can do integration testing that a GPS framework is working correctly between class found in both the framework and the user’s app. Functional testing is testing whether or not a particular user feature works correctly. Functional tests ensure that an application will respond, correctly to a user’s action or, in the unavoidable case, fails gracefully. Finally, acceptance testing is when someone polls users about an application, a particular change or a feature that’s been or will be added. This ensures that a group of user’s remains happy with a particular product or feature.
  2. A company with a large amount of unit tests can make sure that all there code is more robust and can make changes to their code without worrying too much about breaking another person’s code, as long as they are passing all of the corresponding unit tests. Even though unit tests may take a long time to write it is time well spent because a programmer can take much longer to test these bugs, when their code breaks. The only real disadvantage to unit testing is that the tests are only as good as the imagination of the test writer. However if I were to choose a company based on whether or not hey wrote unit tests I would have to choose the company with the unit tests.