*Posted on February 25th, 2014*

**COMP 361 – Accuracy of Types of Testing**

The four types of testing listed in the textbook are: system testing, unit testing, integration testing, and usability testing [1]. Each have their place in overall testing of the system. I will look at system and usability testing in more detail here.

First of all, system testing tests the entire system as a whole. When a complete or iterative version of the system is ready for release, the entire system needs to be tested to make sure all the components work together and all the right outputs are achieved. This can be very intricate and difficult to do. However, once the system gets to be very large, the accuracy of the testing will decline because of the sheer number of possibilities that need testing. This means that it is extremely likely that there are still errors present in the system, even after a full system test is complete. These errors can be caught earlier in the development process with careful integration and unit testing, and proper system tests for each iteration, if using iterative development.

As the system development progresses throughout each phase, it becomes more expensive in both time and money to fix the errors. If a major error is caught in the implementation or support phases, it can also be very costly to the reputation of a company, because they may be thought of as providing poor systems. Therefore, to maximize the number of errors caught with systems testing, a careful plan needs to be developed with proper test data. Additionally, proper use of integration and unit testing during earlier phases will help to reduce the number of errors and help provide test data for the systems test.

Secondly, usability testing is also important. Usability testing involves making sure all parts of the system meet user requirements for functionality and quality. It involves performance testing and acceptance testing. Performance testing can be fairly accurate, because it doesn't depend as much on variable data. If good test data is used, performance can be measured well, but there is still no substitute for real-time loads on the system during peak operation. Usability testing is lest accurate. It can be very formal, but since it involves user requirements, it is more sensitive to change and opinion. As with other testing, there are always errors that are missed, but with proper data and a rigid formal plan, the number of errors can be greatly reduced. Especially if the system is designed and planned very well.

Also, similar to system testing, the costliness of each error increases, the later it is found in the SDLC. Performance testing can be done in earlier phases, and can be combined with integration or unit testing [2]. This means that tweaks to performance can be managed in earlier phases of development. However, acceptance testing is one of the last types of testing to be done [3]. This means that changes can be quite costly, as they are mostly caught just before the system is handed over to users in the integration phase. Changes at this point are much more expensive to deal with. Also similar to systems testing, these errors can be reduced with good test planning and data. Heavy involvement of users in earlier phases of development will help to make sure that the system meets user requirements well before formal acceptance testing.

**References**

[1],[2],[3] J. W. Satzinger, R. B. Jackson and S. D. Burd, Systems Analysis and Design in a Changing World, 5th ed. Boston, USA: Thomson Course Technology, 2009.