Dalton Murray

Systems Analysis and Design

INT 4202 - 1952-202310\_INT4203\_M

Dr. Anthony Padalino

November 29, 2022

**Week 13 Assignment**

**1) Briefly discuss Unit Testing, Integration Testing and System Testing. How are they different?**

Unit Testing – Unit testing is performed when you are testing an individual program or module (Tilley, S. p. 370). When performing unit testing, the goal is to be able to identify and eliminate any execution errors which will cause the program to improperly terminate as well as look for any logic errors that may not show up while testing for errors as the program will still be able to run but not have the correct output (Tilley, S. p. 370). When performing unit testing it is important to have test which data has the correct data as well as erroneous data which will demonstrate that the program is operating correctly (Tilley, S. p. 371). It is also important while unit testing to use stub testing which allows programmers to simulate every program as well as the outcomes and results and to show if the program executed successfully or not, while doing so each stub has an entry or exit point that links to other programs or data files (Tilley, S. p. 371). It is also a good idea with unit testing to create test plans which shows the procedures needed to be done and specify how and when the testing will be performed as well as who will be participants of the testing and what test data will be used when testing (Tilley, S. p. 371).

Integration Testing – Integration testing is done when you are testing two or more programs which depend on each other (Tilley, S. p. 372). An example with integration testi9ng would be if you have a program that checks and validates a customers credit card or credit score or credit status, we then have another program which connects to this and updates data in the customers database, with this system the output from the credit card checks are the inputs of the code to update the database, however, we cannot test these independently because it will not prove that the systems work together and the data passed is correct, which is why they must be tested together (Tilley, S. p. 372). When performing integration testing it is also important to have test data so that we ensure normal data works as well as invalid data is rejected, and we can figure out if there are any problems with the system and the data being passed such as blank records or errors or text not being passed correctly (Tilley, S. p. 372).

System Testing – System testing is performed after integration testing has been completed (Tilley, S. p. 372). When performing system testing it involves testing the entire information system such as including all likely processing situations like having regular users using it, developers using it, managers using it, as well as regular employees (Tilley, S. p. 372). This allows us to be sure that the program meets all requirements and is working properly (Tilley, S. p. 372). When system testing the people who test it enter in data such as samples of actual data, live data, or old data, as well as perform queries, and produce reports, the goal of a system test is to use it like a live system, as it very well could be soon (Tilley, S. p. 372). During this testing outputs are logged, and accuracy is constantly being verified by users and the project development team to make sure everything is working properly (Tilley, S. p. 372). System testing often has a few major objectives, “Perform a final test of all programs, verify that the system will handle all input data properly, both valid and invalid, ensure that the IT staff has the documentation and instructions needed to operate the system properly and that backup and restart capabilities of the system are adequate, demonstrate that users can interact with the system successfully” (Tilley, S. p. 372). However, there are still a few more objectives other than those listed (Tilley, S. p. 372). System testing is how we get approval that the systems are correct and assists in management approval, however, final acceptance tests are what gives us the go ahead to put the system into production (Tilley, S. p. 372). It is also important to know how much testing needs to be done on the system and ensure that amount is done to it (Tilley, S. p. 372).

To simplify, unit testing tests individual programs or modules and has the goal to identify and eliminate/correct any execution errors and logic errors, integration testing is to verify that the program works with other programs and systems, and system testing is to be able to test the entirety of the system and work on getting approvals before moving to final acceptance testing.

**2) Briefly discuss the testing environment and operational environment.**

Testing environment – The testing environment is used as a place for programmers and analysts to be able to develop and maintain programs. It is necessary to have a separate environment to test, develop, and maintain programs so that system security and integrity are able to be properly done (Tilley, S. p. 378). It is also important to use a test environment simply to protect the operational environment, thus limiting potential damage done in the testing environment to only the testing environment and not outside of it, meaning if something were to happen in the testing environment it would be limited to that and not be able to damage the operational environment. The test environment also typically has limited access to who can use it and should be in an extremely controlled environment, preventing anyone from being able to access it who shouldn’t be able to as it may have systems in it which are not done and may be insecure until further testing has taken place, it is also important for access to be controlled so information has less chance to be leaked to the public (Tilley, S. p. 378). The testing environment also often contains a copy of all programs, procedures, and test data files on it, and when making changes to the operational environment it must be done on the test environment first to ensure that everything works correctly (Tilley, S. p. 378).

Operational environment – The operational environment is the environment which is used as the production environment, in other words it is what has the actual systems which are implemented and are in use stored on. The operational environment is also an extremely controlled environment with many less people compared to the test environment has access to, this is because the operational environment should not have frequent changes and should only have changes when critical systems are at stake, or corrections to a system needs to be done, or other authorized modifications or enhancements (Tilley, S. p. 378). The operational environment is usually where day-to-day operations take place on and is what most people are able to see (Tilley, S. p. 378). When making changes to the operational environment it is important to have approval before doing so (Tilley, S. p. 378). The operational environment also often contains hardware and software configurations, system utilities, telecommunications resources, and any other component necessary for it to be ran (Tilley, S. p. 379).

The primary differences between the testing environment and the operational environment is that the testing environment is where every change must be done to first and fully tested and then can be, with approval, moved onto the operational environment. The test environment may have a large amount of frequent changes, however, the operational environment should not have too many. Both should have a controlled access as to who can access what, however, the operational environment typically has a more extreme amount of control.

References

Tilley, S. (2020). Systems analysis and design (12th ed.). Cengage.

I have neither given nor received unauthorized aid in completing this work, nor have I presented someone else's work as my own.

*Dalton Murray*