Week 2 Learning Activity - Software Testing Levels

Shaun Hoadley

CST313 Software Testing

Professor Michael Hayden

06/14/2021

This week’s learning activity concerns the different software testing levels. These software testing levels include the general V-model, component testing, integration testing, system testing, and acceptance testing. These tests are performed throughout the software development life cycle (SDLC) to ensure quality products are developed. Testing, when properly preformed, can help reduce costs, improve code maintainability and quality, find and correct faults and flaws earlier, increase the robustness of the product, and ensure the customer receives the product that they need (Spillner, Linz, & Schaefer, 2014).

The general V-model is used to show the relationship between software testing and the development process. The general V-model derives it’s name from the “V” shape formed by the representation of it’s two branches, the development branch (left side) and the testing branch (on the right) and the direction of their flow down from the left and then up the right. The general V-model begins at the top left with Requirements Engineering and moves down to the functional systems design. It then proceeds onward to the technical systems design, followed by component specification. After the designs and specifications are ready, it moves into programming, and finally starts up the right branch into the various testing phases below.

The first phase of testing, following the general V-model up the right branch, is component testing. Component testing consists of testing the individual parts of the product the be sure they perform their tasks as intended, without any external influence. This phase of testing is done by the author as they complete each function of the component. If it is possible, component testing should also be conducted by one to five of the author’s peers, as this will help in locating any faults that may have escaped the author’s notice.

After the component testing is deemed passed, it is moved on to integration testing. The integration testing phase is responsible for taking the components and making sure that they work together as intended. This is where the first external influences to each component begins, though still under controlled conditions (i.e. will not inadvertently tamper with vital real-world data). The components are not only tested to see if they work with the other components they are supposed to, but also to be sure they do not affect, or can be affected by, components that they are not intended to work with but are still a part of the system as a whole.

The next phase of testing is the system test. In the system test, the product is tested in the environment for which it is itended to be used, checked for compliance with the requirements of the product, and matched up with all associated, necessary documentation included in the project. This is the final stage of testing BEFORE taking the product to the customer for final approval and acceptance.

Finally, we move up to the top right of the general V-model, acceptance testing. Acceptance testing puts the product in the customers hands. The customer ensures that all requirements have been met, both functional and non-functional. The client verifies that everything works as intended in their environment and the documentation meets their needs. At this point, the client either accepts or rejects the product as completed. If rejected, the product goes back into development to accommodate the clients requests or revisions. Otherwise, the client accepts the product and the project is deemed completed, at least for the moment.

**References**

Spillner, A., Linz, T., & Schaefer, H. (2014). *Software testing foundations: A study guide for the certified tester exam* (4th ed.). Rocky Nook.