

Testing and Its Importance

Angela Dini

CSCI3190

Mr. James Fleming

June 13, 2016

When it comes to web development, there are many different steps that a designer must take to have a good website. A more formal list of these steps is given by the Software Development Life Cycle or SDLC, according to SDLC – Overview (2016). The SDLC has multiple phases, including obtaining a list of what the client needs for their website, formalizing this list into a list of what the developers need to do to meet the requirements set up by the client, making a prototype of the website, testing it, and finally releasing the final product. There are many different types of testing that designers can do to make sure their website is the best it can be. These different types of testing are very important in the SDLC and in our daycare website project.

There are two main ideas behind software testing, according to Zafar (2012): verification and validation. Verification is about making sure the website behaves in the way it is supposed to and validation is about meeting the needs of the client. These two ideas must be present in any software development project and can be achieved through testing. There are two main types of testing: black box testing and white box testing. Black box testing is mostly concerned with the output of a website given some input while white box testing is more concerned with the inner workings of the website. Black box testing is usually used for validation while white box testing is usually for verification. There is also a third main type of testing called grey box testing, which is a combination of white box testing and black box testing. In grey box testing, the tester only has a small amount of knowledge about the inner workings of the website. In black box testing, the tester does not have any knowledge about the inner workings and in white box testing, the tester only focuses on the inner workings. Grey box testing also works with different architectural models such as UML diagrams. According to Grey box testing (2016), this type of testing is ideal for websites. Other than these, there are many other different types of testing, which are discussed below.

One type of testing, according to Palani (2011), is usability testing. This type of testing is about the user interfaces and if they follow universally recognized standards. They also need to be accessible to everyone, including people with disabilities and people with varying degrees of colour-blindness. Some aspects of this testing include easy navigation between pages, no overcrowded pages with too much content, and being intuitive for every user, including beginners and experts. According to Nielsen (2012), usability can be divided into five aspects: “learnability” (Nielsen, 2012) or how straightforward a user finds doing a task for the first time with the design, “efficiency” (Nielsen, 2012) or how long the user takes to do a task, “memorability” (Nielsen, 2012) or how much the user remembers how to do a task the next time they have to do it using the design, “errors” (Nielsen, 2012) or how many mistakes the user makes and how quickly they can recover from those mistakes, and “satisfaction” (Nielsen, 2012) or how much they enjoyed using the design. This type of testing is usually done by observing how participants interact with a website design without interference from the researchers. Many researchers involved in human-computer interaction argue that five participants are enough to conduct a proper usability test although there is some debate. This type of testing should be done at every step of development.

Another type of testing, according to Palani (2011) is user acceptance testing. This type of testing is about if the website does what the user expects it to do. Some aspects of this testing include making sure that the users can open the website on their computers and servers, that input fields are big enough for the information that will need to be inputted into them, and that the fields have an appropriate way to input data, such as using a radio button to select your gender instead of having to type it out. There are two stages in user acceptance testing: alpha

testing and beta testing. Alpha testing is done by developers at their workplace setting and beta testing is done by a select number of users in their own settings. There are many steps that the tester needs to follow in order to do user acceptance testing according to Seela (2016). The tester first needs to think about the acceptance criteria, which is a set of conditions that need to be met in order for the website to be considered a success. These acceptance criteria are usually discussed with the client, users, and/or the developers. Next, the tester determines how involved the Quality Assurance or QA team will be in the testing. This team consists of people who make sure that the website meets all the technical requirements and is in good condition. Usually the QA team help the tester by, for example, teaching the tester how to use a website and be around in case the tester has any questions. Both of these steps happen before the actual user acceptance testing. The next step is to plan the roles of everyone involved, the setting of the testing, and how to analyze the results. Then test cases make sure that each acceptance criterion is tested and satisfied. After this, the testers and QA team all test each test case in the website and the results are written down. Finally, the decision is made about whether all the acceptance criteria were met and if the website was successful or not.

Another type of testing Palani (2011) discusses is performance testing. This type of testing has to do with how well the system does in different situations. There are many different kinds of performance tests. For example, the stress test makes a maximum limit to the functioning of the website and tests what the website does when the amount of data exceeds this maximum limit, the scalability test tries to uncover how the website can change when it is used in different software and hardware, and the load test determines how the website uses its resources such as memory space when it has a lot of data. Performance testing can be done by many different applications that can be installed onto a computer. These applications include Jmeter, OpenSTA, Load Runner, and Web Load, according to Performance testing (2016). WordPress can also do performance testing. According to WPExplorer.com (2016), it does this using Pingdom, a free plugin that checks every page of a website and analyzes it for different aspects, including the time it takes to load and its size.

Another type of testing Palani (2011) discusses is security testing. This type of testing is involved with the security of the website and can be useful if a website needs to be secure because it has confidential information or needs to prevent other people from changing the information on the website. There are two kinds of security testing: static and dynamic. Static testing is done by analyzing the code of the program and checking for hidden weaknesses and security threats and seeing how the security works throughout the code. Dynamic testing is done by seeing if the program runs the appropriate response when a request is made (Palani, 2011). According to Jones (2013), both static security testing and dynamic security testing can be done using different applications that can be installed onto a computer. These applications include Secure Programming Lint or Splint, Pylint, and FindBugs for static security testing, and Valgrind, dmalloc, OpenVAS, and Wireshark for dynamic security testing. The applications that perform static security testing analyze the code and can reveal many hidden issues that could be security threats. The applications that perform dynamic security testing can reveal different memory issues and other issues with the code that could be potential security threats (Jones, 2013).

Another type of testing, according to Palani (2011), is functionality testing. This type of testing is very similar to black box testing. It is about checking to see if a website is functioning correctly or if it has functional errors. For example, this testing should test if all correct input values are accepted and if the incorrect input values are rejected and if an error message is

displayed when an incorrect input value is seen. This testing should also check if the hyperlinks are working properly and that the correct changes are made depending on what is clicked, for example, testing if a database is changed when a person adds a name to the waitlist on the website. There are many different kinds of functionality testing including database testing or checking if a database makes the correct changes depending on the input, configuration testing or checking if the website works on all of its supported software and hardware, compatibility testing or checking if the website can run on any person's machine, and flow testing or checking if the flow of the website and the way you get to different pages is intuitive.

Another type of testing Palani (2011) discusses is interface testing. This type of testing is about the connectedness of the website. This can be about the separate parts on the website or how the website connects to outside sources, such as other websites. This is significantly important when it comes to using the website on more than one module. In interface testing, it is important to make sure that the data from one platform moves effortlessly to another without errors. There are many things to test when doing interface testing. For example, according to Interface Testing (2016), the tester must make sure that the communication between two platforms does not have any errors, the software and hardware that the website supports uses the website correctly, all files from the website can be opened on all devices, encryption has been maintained when there is a connection between two platforms, and the website can manage a network failure (Interface testing, 2016).

Another type of testing, according to Barron (2015), is A/B testing. This type of testing is about comparing two different forms of the same website that differ in certain aspects and testing which one works better. This can be very helpful when a designer is trying to decide between two similar features. There are three kinds of A/B testing: split testing, multivariate testing, and experimental. Split testing is when only one aspect is different between the two websites so you can specifically test that one aspect. Multivariate testing is more intricate because in this type of A/B testing there is more than one aspect that is different between the two websites. Experimental is about checking every element on the website and it allows you to look at everything. Split testing is the most common kind of A/B testing. A designer can do A/B testing by first deciding what the purpose and goals of the website are. This can help the designer to figure out what to test in order to achieve these goals. For example, if the goal was to increase social awareness, the designer may want to make the share button easier to find by moving it to different locations on the website and seeing where the best location for it is. After this, the designer will make different versions of the website, do experiments with participants, and then analyze the data that they have collected to figure out which version of the website is the best. WordPress can do A/B testing using different plugins, as given by Barron (2015). For example, Simple Page Tester can be used for A/B split testing. In this plugin the designer simply needs to click an edit button which will set up two versions of a website. The designer can then edit both to make some variation between them and then choose which one was more successful. Nelio A/B Testing is another common plugin mentioned by Barron (2015) that can be used with heat maps to see which aspects of the website are the most visited. This plugin has a free version.

Another type of testing, presented by McFarlin (2012), is unit testing. This type of testing is about testing small areas or units of the code used to design a website. Black box testing, white box testing, and grey box testing are classified under this type of testing because they observe small portions of the website or system. Designers can do this type of testing by testing their code after they write a small amount of it or by using a plugin in WordPress that can do unit testing. For example, PHPUnit is a common plugin that can be used to do unit testing, according

to Haynes (2014). To use this, one must install it, use a WordPress framework with a testing library, make an xml file, and finally make unit tests for this plugin to use. PHPUnit can also be used to test any PHP code that has been used in developing the WordPress website.

All of these testing types are very important when developing a website. It is also very important to test during each stage of development because, according to Khan (2014) and his colleague, catching errors quicker can lead to fixing them when they are small issues and have not yet developed into larger issues. Testing can also give developers the opportunity to reread the list of requirements, make sure they are doing everything correctly, and to fix errors.

For our project, all of these types of testing can be useful. However, some types will be more relevant than others. For example, because we do not have any users to test with, we will not be able to do experiments and therefore the experiment portion of usability testing, beta testing, and A/B testing that requires users will not be feasible. Although we do not have users, we may still be able to do the user acceptance testing using the RPG from the client. We would also still be able to do some parts of the usability testing, for example, making sure the website follows universally recognized standards and is accessible. We can also make sure there are no overcrowded pages with too much content. We could also do some aspects of user acceptance testing that does not involve users, for example, making sure that the fields have the appropriate way to input data. We could mostly do the alpha testing phase of acceptance testing of user acceptance testing. Because the security of the website is out of scope, security testing would not be relevant to our project. Interface testing would also not be relevant because the deployment of the website is out of scope as well. For this reason as well, certain types of functionality testing, such as the compatibility testing and configuration testing, will not be used. Performance testing and interface testing, however, would be relevant because we are redesigning a daycare website and therefore need to check that everything in the website is working properly. In this regard, unit testing would also be useful to check if the code that builds the website is also working. This unit testing would include black box testing, white box testing, and grey box testing.

Testing is a very important step in the SDLC. It can help to reveal small errors before they become big problems and help to make sure the website is running smoothly and quickly. There are many different types of testing which focus on different aspects of the website. All of these tests have different methods and can be used in different ways. For our project, performance testing, some aspects of functionality testing, and unit testing, including black box testing, white box testing, and grey box testing, would be useful to improve the website. Some aspects of usability testing and the alpha testing phase of user acceptance testing would also be useful. However, the experiment parts of usability testing, the beta testing phase of user acceptance testing, and A/B testing would not be feasible because of the lack of participants. Security testing and interface testing would also not be relevant because they test aspects of our project that we have considered to be out of scope. All in all, testing is very helpful because it can help to make a user's visit to a website much more enjoyable.

- Barron, B. (2015). 11 best tools for A/B split testing in WordPress. *wpmudev*. Retrieved from [https://premium.wpmudev.org/blog/ab-split-testing/?utm\\_exp=3606929-67.-MP1PYYzS\\_6mvlSTiltv6Q.0&utm\\_referrer=https%3A%2F%2Fwww.google.ca%2F](https://premium.wpmudev.org/blog/ab-split-testing/?utm_exp=3606929-67.-MP1PYYzS_6mvlSTiltv6Q.0&utm_referrer=https%3A%2F%2Fwww.google.ca%2F)
- Grey box testing. (2016). *www.tutorialspoint.com*. Retrieved from [http://www.tutorialspoint.com/software\\_testing\\_dictionary/grey\\_box\\_testing.htm](http://www.tutorialspoint.com/software_testing_dictionary/grey_box_testing.htm)
- Haynes, W. (2014). Unit testing themes and plugins in WordPress. *INN Nerds*. Retrieved from <https://nerds.inn.org/2014/10/22/unit-testing-themes-and-plugins-in-wordpress/>
- Interface testing. (2016). *www.tutorialspoint.com*. Retrieved from [http://www.tutorialspoint.com/software\\_testing\\_dictionary/interface\\_testing.htm](http://www.tutorialspoint.com/software_testing_dictionary/interface_testing.htm)
- Jones, M. T. (2013). Static and dynamic testing in the software development life cycle. *IBM developerWorks*. Retrieved from <http://www.ibm.com/developerworks/library/se-static/>
- Khan, M. E. & Khan, F. (2014). Importance of software testing in software development life cycle. *International Journal of Computer Science Issues*, 11(2), 120-123. Retrieved from <http://www.ijcsi.org/papers/IJCSI-11-2-2-120-123.pdf>
- McFarlin, T. (2012). The beginner's guide to unit testing: What is unit testing? *envatotutsplus*. Retrieved from <http://code.tutsplus.com/articles/the-beginners-guide-to-unit-testing-what-is-unit-testing--wp-25728>
- Nielsen, J. (2012). Usability 101: Introduction to usability. *Nielsen Norman Group*. Retrieved from <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>
- Palani, G. S. (2011). Summary of web application testing methodologies and tools. *IBM developerWorks*. Retrieved from <http://www.ibm.com/developerworks/library/wa-webapptesting/>
- Performance testing. (2016). *www.tutorialspoint.com*. Retrieved from [http://www.tutorialspoint.com/software\\_testing\\_dictionary/performance\\_testing.htm](http://www.tutorialspoint.com/software_testing_dictionary/performance_testing.htm)
- Seela, S. (2016). What is user acceptance test (UAT) and how to perform it effectively? *Software Testing Help*. Retrieved from <http://www.softwaretestinghelp.com/what-is-user-acceptance-testing-uat/>
- SDLC – Overview. (2016). *www.tutorialspoint.com*. Retrieved from [http://www.tutorialspoint.com/sdlc/sdlc\\_overview.htm](http://www.tutorialspoint.com/sdlc/sdlc_overview.htm)
- WPExplorer.com. (2016). 5 tools for testing your WordPress site's performance and speed. *WPExplorer.com*. Retrieved from <http://www.wpexplorer.com/testing-wordpress-performance-speed/>
- Zafar, R. (2012). What is software testing? What are the different types of testing? *Code Project*. Retrieved from <http://www.codeproject.com/Tips/351122/What-is-software-testing-What-are-the-different-ty>