Data Driven Testing of CNTR-195C

File type: docx

Files Size: Less than 10 MB

Prepared By: Anamul

Date: 12-Oct-2016

Basic Testcase Concepts

A testcase is simply a test with formal steps and instructions; testcases are valuable because they are repeatable, reproducible under the same environments, and easy to improve upon with feedback. A testcase is the difference between saying that something *seems* to be working okay and proving that a set of specific tasks are *known* to be working correctly.

Some tests are more straightforward than others. For example, say you need to verify that all the links in your web site work. There are several different approaches to checking this:

- you can read your HTML code to see that all the link code is correct
- you can run an HTML DTD validator to see that all of your HTML syntax is correct, which would imply that your links are correct
- you can use your browser (or even multiple browsers) to check every link manually
- you can use a link-checking program to check every link automatically
- you can use a site maintenance program that will display graphically the relationships between pages on your site, including links good and bad
- you could use *all* of these approaches to test for any possible failures or inconsistencies in the tests themselves

Verifying that your site's links are not broken is relatively unambiguous. You simply need to decide which one of more of these tests best suits your site structure, your test resources, and your need for granularity of results. You run the test, and you get your results showing any broken links.

Notice that you now have a list of broken links, not of *incorrect* links. If a link is valid syntactically, but points at the incorrect page, your link test won't catch the problem. My general point here is that you must understand *what* you are testing. A testcase is a series of explicit actions and examinations that identifies the "what".

A testcase for checking links might specify that each link is tested for functionality, appropriateness, usability, style, consistency, etc. For example, a testcase for checking links on a typical page of a site might include these steps:

Link Test: for each link on the page, verify that

1. the link works (i.e., it is not broken)
2. the link points at the correct page
3. the link text effectively and unambiguously describes the target page
4. the link follows the approved style guide for this web site (for example, closing punctuation is or is not included in the link text, as per the style guide specification)
5. every instance of a link to the same target page is coded the same way

As you can see, this is a detailed testing of many aspects of the link, with the result that on completion of the test, you can say definitively what you know works. However, this is a simple example: testcases can run to hundreds of instructions, depending on the types of functionality being tested and the need for iterations of steps.

Defining Test and Testcase Parameters

A testcase should set up any special environment requirements the test may have, such as clearing the browser cache, enabling JavaScript support, or turning on the warnings for the dropping of cookies.

In addition to specific configuration instructions, testcases should also record browser types and versions, operating system, machine platforms, connection speeds — in short, the testcase should record any parameter that would affect the reproducibility of the results or could aid in troubleshooting any defects found by testing. Or to state this a little differently, specify what platforms this testcase should be run against, record what platforms it is run against, and in the case of defects report the exact environment in which the defect was found.

An Example Testcase

For a more realistic example of a testcase, I have built the following sample testcase for philosophe.com.

```
0. On a Windows 95 box, load Netscape Navigator 3.04. Clear browser
   caches, and clear history.
1. In the URL field type in www.philosophe.com and hit the
   ENTER key
2. On the home page, for each link
   a. examine the link text
   b. click on the link, and verify that the link works
   c. verify that the new page is the correct page, as
      indicated by the link text
   d. verify that the little black arrow correctly indicates
      the current page
   e. click on the browser's BACK button
   f. verify that the link's color changed to the vlink color
3. Once all links on the home page have been tested, click on the first
   link in the left navigation column.
4. for this page, repeat steps 2a - 2f.
5. repeat steps 3 and 4 until every page has been tested
```

As you can see, even the simplest of testcases can go into great detail; for a site that has a stable structure, this is a good thing.

Some Issues to Consider About Testcases

The quality assurance process will find defects uncovered by formal testcases, just like quality control, but in addition will find defects in usability or performance that wouldn’t be caught by even the most detailed testcases. This is the major shortcoming of quality control testing, that the very design of the testcases will tend to limit the scope of attention paid to how things come together with the code. Furthermore, with a web site that changes rapidly, intricate testcases will become outdated quickly, requiring a great deal of work just to keep them current. My point here is not that testcases are not useful, but that you must seek a balance between using scarce resources for creating testcases and using them for actual *testing*.