



INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X

Impact factor: 4.295

(Volume 4, Issue 2)

Available online at: www.ijariit.com

Techniques to test web application system based on scenarios

Reddy Sangeetha DM

sangeethad1@gmail.com

ABSTRACT

Software Testing is the most significant stage in Software Development Life Cycle. The overall aim of any software industry is to provide quality to the end user. To ensure high quality software it is required to test. Testing ensures all the requirements are met and the software/application is bug free. As, we are aware web sites are rapidly increasing day by day and testing needs to be done to ensure if the site is user friendly to the end user and there are no bugs in the production environment. This paper mainly focuses on Web application testing, types of web application testing and black box testing techniques. There are many testing techniques used to optimize the test cases, this paper mainly explains the most commonly used techniques used for the web application testing.

Keywords: *Web application, software testing, testing techniques, quality, optimization, functional testing.*

1. INTRODUCTION

Software Testing is one of the major and primary techniques for achieving high quality software. Software testing is done to detect the presence of faults, which can cause failure. Software testing can also be defined as process of verifying and validating software to ensure software meets technical and business requirements as expected.

Websites has become a major delivery platform for variety of complex and enterprise application in several domains. In addition to this web applications exhibit complex behavior and unique behavior in terms of usability, performance, quality and reliability. The scope and complexity of current web applications vary widely from small scale to large scale enterprise applications. There are various categories of web applications are Informational (Online Newspaper, Electronic books etc), Interactive (Registration Forms, Online Games), Transactional (Online banking), Online communities (Chat groups, face book) and Web portals (E-commerce sites) .

Web applications are multidisciplinary and they are built in constantly changing environment where requirements are unstable. Quality web application needs to be usable, functional, reliable, maintainable, scalable and secure. Web application testing has many dimensions in addition to conventional software testing. Each unit of Web application such as page, code, site, navigation, standards, and business requirements needs to be tested. However, software testers need to prepare an explicit testing strategy that covers all the tests. This paper mainly focuses on techniques used to test various categories of web application.

This paper is divided in 4 sections. Section 2 describes briefly about various testing types in Web application. In section 3 various black box testing techniques are explained with examples. Section 4 concludes the paper.

2. WEB APPLICATON TESTING TYPES

Web testing in simple terms is checking your web application for potential bugs before its made live or moving the code to production environment.

During this stage issues such as that of web application security, functioning of site and business workflow are tested. The end purpose of Website testing is to make user comfortable in learning, understanding, using and navigating the website.

Depending on the requirement, the following testing types needs to be performed.

1) *Functional Testing:*

This is used to check if the website is as per the specifications intended and functional requirements are met.

Testing activities include:

- a) *Test all links in your webpages are working correctly and make sure there are no broken links. Links to check will include outgoing links, internal links and anchor links.*

Example:



Figure 1: Link Example

Clicking on Terms of Service link, it should redirect to appropriate page, if it's not redirecting it is termed as defect.

- b) *Test forms are working as expected, this will include*

- i) Mandatory fields should be validated

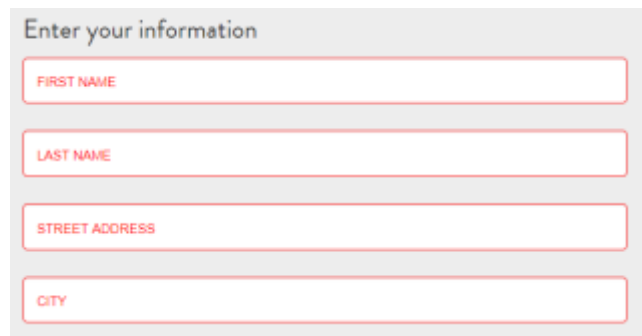


Figure 2: Mandatory field example

- ii) Verify if default values are being populated
- iii) Once submitted, data should be saved in live database.
- iv) Verify the workflow of the test form.

2) *User Interface Testing*

- i) Checking for font size, alignment etc.
- ii) Readable color schemas

3) *Compatibility Testing*

This testing is performed based on the context of the application. Testing is performed to ensure if web application is compatible in browser, operating system and various devices like notebook, mobile etc.

4) *Performance Testing*

Testing is performed to verify the application load and response time.

- i) Load Testing: Testing conducted to understand the behavior of the system under a specific load.
- ii) Stress Testing: It is performed to find the upper limit capacity of the system and also to determine how the system performs if the current load goes well above the expected maximum
- iii) Spike Testing: Testing is performed by increasing the number of users suddenly by large amount and measuring the performance of the system

i. Security Testing:

Testing is performed to verify if the application is secured on web. Some of the techniques to perform security level of the system are as follows:

- i) SQL Injection
- ii) Cross-site Scripting (XSS)
- iii) Using components with known vulnerabilities
- iv) Invalidated redirects and forwards

3. BLACKBOX SOFTWARE TESTING TECHNIQUES

Testing techniques refers to the method or a way to test application or particular part of software. Different testing techniques has different approach and each technique can be used to find defect in the software. By using this testing techniques, we can optimize our test cases and ensure all scenarios are covered. Different types of black box testing techniques are described below:

1) Equivalence Partitioning

It is a software testing technique that divides the input data of software into partitions. Partitions are divided into valid and invalid classes. From each partition one test case can be written. This ensures test cases are optimized and 100% test coverage has been met.

Example:

Testing e-commerce site, the scenario will be like

On purchase of 1000 Rs., customer should get 5% discount.

On purchase of 2000 Rs customer should get 10% discount. On purchase of 3000 and above user should get 15% discount.

In this scenario, test cases cannot be written for every value like if user purchases 1200 Rs 5% should be given etc, so we need to use this technique to optimize and make sure test coverage is met.

We need to partition the values now. There are 4 partitions in the below figure:



Figure 3: Equivalence Partition Example

2) Boundary Value Analysis

This technique is used to identify errors at the boundaries rather than finding errors in the center of input domain. More application errors usually occur at the boundaries of the system. Boundary value analysis is the next part of equivalence partitioning for designing test cases where test cases are selected from the edges of equivalence classes. The technique to select test cases for boundary value analysis is min-1,min,min+1, max-1,max, max+1. For example if the input field accepts 1-10 characters, we need to write test cases for 0,1,2,9,10,11.

Scenario: Testing e-commerce site, the scenario will be like

On purchase of 1000 Rs., customer should get 5% discount.

On purchase of 2000 Rs customer should get 10% discount. On purchase of 3000 and above user should get 15% discount.



Figure 4: Boundary Value Analysis Example

3) Decision Table

Testing the system with different combination of inputs which produce different results. It is also cause effect table where cause and effects are captured for better test coverage. A decision table is a tabular representation of inputs/rules/test conditions.

Scenario: Upload Screen, now consider a dialogue box which will ask the user to upload photo with certain conditions like upload only jpeg image, file size less than 20 kb, resolution should be 1024*798

If any one of the condition fails, error message will be shown. If all conditions are met image will be uploaded successfully.

Table 1: Decision Table Example

Conditions	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
Format	png	png	png	png	Not png	Not png	Not png	Not png
Size	Less than 20kb	Less than 20kb	Less than 20kb	Less than 20kb	Less than 20kb	Less than 20kb	More than 20kb	More than 20kb
resolution	1024*798	Not 1024*798	1024*798	Not 1024*798	1024*798	Not 1024*798	1024*798	Not 1024*798
Output	Photo uploaded	Error message resolution mismatch	Error message size mismatch	Error message size and resolution mismatch	Error message for format mismatch	Error message for format and resolution mismatch	Error message for format and size mismatch	Error message for format, size, and resolution mismatch

For this condition we can create 8 test cases and ensure complete test coverage.

- 1) Upload a photo with format '.png', size less than 20kb and resolution 1024*798 and click on upload. Expected result is Photo should upload successfully
- 2) Upload a photo with format '. png ', size less than 20kb and resolution not 1024*798 and click on upload. Expected result is Error message resolution mismatch should be displayed
- 3) Upload a photo with format '. png ', size more than 20kb and resolution 1024*798 and click on upload. Expected result is Error message size mismatch should be displayed
- 4) Upload a photo with format '. png ', size less than 20kb and resolution not 1024*798 and click on upload. Expected result is Error message size and resolution mismatch should be displayed
- 5) Upload a photo with format other than '. png ', size less than 20kb and resolution 1024*798 and click on upload. Expected result is Error message for format mismatch should be displayed
- 6) Upload a photo with format other than '. png ', size less than 20kb and resolution not 1024*798 and click on upload. Expected result is Error message format and resolution mismatch should be displayed
- 7) Upload a photo with format other than '. png ', size more than 20kb and resolution 1024*798 and click on upload. Expected result is Error message for format and size mismatch should be displayed
- 8) Upload a photo with format other than '. png ', size more than 20kb and resolution not 1024*798 and click on upload. Expected result is Error message for format, size and resolution mismatch should be displayed

4) *State Transition Testing*

It is a testing technique in which changes in input conditions cause's state changes in application under test. In this technique tester analyses the behavior of an application under test for different input condition in a sequence. In this technique tester provides both positive and negative input test values and records the system behavior.

Four parts of State Transition are:

- *State: (1st ry)*
- *Transition (From one state to another)*
- *Events (Original Transition)*
- *Action (The result from transition)*

Scenario: Consider a login screen of a banking application, entering valid username and password it takes you the the home page. Entering 3 times invalid username and password , 4th time when you tried to login your account in blocked.

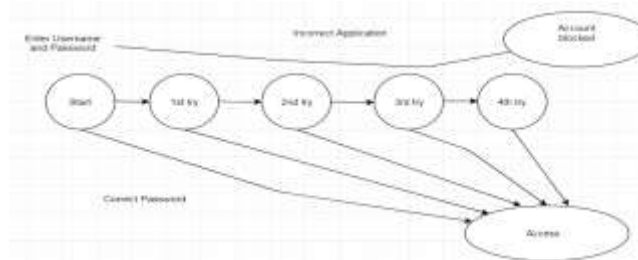


Figure 5: State Transition Example

It gives you the access to the application with correct password and login name, but what if you entered the wrong password.

The application allows three attempts, and if users enter the wrong password at 4th attempt, the system blocks the user. In this case, testing with the correct password and with an incorrect password is compulsory. For the test scenarios, log-in on 2nd, 3rd and 4th attempt anyone could be tested.

You can use State Table to determine invalid system transitions.

Table 2: State Transition Table

Column1	Column2	Column3
	Correct Password	Incorrect Password
S1)Start	S6	S2
S2) 1st Try	S6	S3
S3) 2nd Try	S6	S4
S4) 3rd Try	S6	S5
S5) 4th Try	S6	S7
S6) Access	S6	
S7) Account blocked		

You can use State Table to determine invalid system transitions.

In a state Table, all the valid states are listed on the left side of the table, and the events that cause them on the top.

Each cell represents the state system will move to when the corresponding event occurs.

For example, while in S1 state you enter a correct password you are taken to state S6 (Access Granted). Suppose if you have entered the wrong password at first attempt you will be taken to state S3 or 2nd Try.

Likewise, you can determine all other states.

Two invalid states are highlighted using this method. Suppose you are in state S6 that is you are already logged into the application, and you open another instance of bank application and enter valid or invalid passwords System response for such a scenario needs to be tested.

ii. Pair Wise Testing

It is a test case optimization technique used when the system needs to be tested with many combinations/inputs.

Scenario:

Consider a student registration form, where there are many combinations to test.

The image shows a web form for student registration. It includes the following elements:

- Full Name ***: A text input field.
- Extra Curricular Participation ***: A section with two checkboxes: Student Council and Class Officer.
- Do you have any scholarship? ***: A section with two radio buttons: Yes and No.
- Network Provider ***: A dropdown menu with options: Sun Cellular, Talk N' Text, and Touch Mobile (TM).
- Submit**: A button at the bottom of the form.

Figure 6: Pair Wise Testing Example

- Text box -> any alphabets of 100 characters
- checkbox -> checked or unchecked
- Radiobutton -> On and Off
- Listbox -> 0,1,2,3 values

Exhaustive combinations will be

Listbox =3
 Radiobutton =2
 Checkbox=2
 Textbox =100
 Total number of test cases = 100x2x2x3 = 1200
 Including negative test cases >1200

While using pairwise technique we can optimize and have only 4 testcases.

Table 3: Pair wise Test cases

Column1	Column2	Column3	Column4
Textbox	Checkbox	Radiobutton	Listbox
Valid characters	checked	On	Any value
Valid characters	Unchecked	off	Novalue
Invalid characters	checked	off	Anyvalue
Invalid characters	Unchekeed	ON	Novalue

4. CONCLUSION

Software Testing is the main phase in any software development life cycle. Without testing, the product/software cannot go live. Testing needs to be completed in the given period of time and testers should ensure they have tested all major scenarios in the given period of time and software/product should work as expected. This paper describes in detail about black box testing techniques, types of testing, strategies for writing test cases using these techniques. By using these techniques, test cases can be optimized and testers can ensure 100% coverage from the requirement document and deliver good quality software. Important stages in the process of testing are on the methods of designing test cases. It is impossible to find all the bugs from software, so that we have designed number of testing techniques that can be taken to analyze.

5. REFERENCES

[1] Bansal, A.; (2014) "A Comparative Study of Software Testing Techniques", <http://www.ijaprr.com/download/1440480921.pdf>.
 [2] Irena, J.; (2008) "Software Testing Methods and Techniques", <http://tir.ipsitransactions.org/2009/January/Paper%2006.pdf>.
 [3] Sharma, C; Sibal, R; (2013) "A Survey on Software Testing Techniques using Genetic Algorithm", <https://pdfs.semanticscholar.org/3c1d/f844a948f1401a253e1aeaa453edefc60c96.pdf>
 [4] Tarika, B; (2014) "Review on Software Testing Techniques", <http://www.ijritcc.org/download/Review%20on%20Software%20Testing%20Techniques.pdf>.
 [5] R.S. Pressman & Associates, Inc. (2005). Software Engineering: A Practitioner’s Approach, 6/e; Chapter 14: Software Testing Techniques,
 [6] Yadav, P; Kumari, P; (2015) "REVIEW PAPER ON SOFTWARE TESTING", http://www.ijirt.org/vol1/paperpublished/IJIRT102003_PAPER.pdf.
 [7] Sahil Batra and Dr. Rahul Rishi, "IMPROVING QUALITY USING TESTING STRATEGIES," Journal of Gobal Research in Computer Science, Volume 2, No.6, June 2011.
 [8] B. Beizer. Software Testing Techniques (2nd ed.). Van Nostrand Reinhold Co., New York, NY, USA, 1990. B. Beizer. Software Testing Techniques (2nd ed.). Van Nostrand Reinhold Co., New York, NY, USA, 1990.
 [9] S.M.K Quadri and Sheikh Umar Farooq, "Software Testing-Goals, Principles and Limitations," International Journal of Computer Applications, Volume 6-No.9, September 2010.
 [10] G. Bernot, M. C. Gaudel, and B. Marre. Software testing based on formal specifications: a theory and a tool. Softw. Eng. J., 6(6):387–405, 1991.
 [11] Myers, Glenford J. (1979), IBM Systems Research Institute, Lecturer in Computer Science, Polytechnic Institute of New York, The Art of Software Testing, by John Wiley & Sons, Inc.
 [12] ger S. Pressman, "Software engineering: A practitioner’s Approach," fifth edition, 2001.