



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

Study on random order testing

M. Arjun

sudhaarjun@gmail.com

Sathyabama University, Chennai, Tamil Nadu

John Bruce

johnbruce@sathyabamauniversity.ac.in

Sathyabama University, Chennai, Tamil Nadu

ABSTRACT

Software testing is very highly used and very expensive. The main function of the software testing is to find the defects. Not all software defects are caused by coding errors. One common source of expensive defects is requirement gaps, e.g. unrecognized requirements that result in errors of omission by the program designer. Requirement gaps can often be no-functional requirements such as testability, scalability, maintainability, usability, performance, and security. A programmer makes an error (mistake), which results in a defect (fault, bug) in the software source code. If this defect is executed, in certain situations the system will produce wrong results, causing a failure. So there are different test cycles in which a project can be tested. In this project I have taken a shopping cart website, the testing has been done to each test case multiple times to check the amount of errors get in the testing. The duration of the testing is also noted and total randomizations of testing in known possible ways also have been tested and see the successful testing output without any errors and normal testing for the same shopping kart site is done simultaneously and comparing the difference and time consuming and a maximum number of errors obtained.

Keywords: Software Testing, Testing.

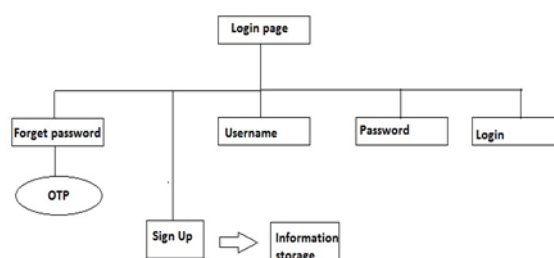
1. INTRODUCTION

Software Testing is the process of identifying the correctness and quality of the software program. The purpose is to check whether the software satisfies the specific requirements, needs, and expectations of the customer [1]. Software testing consumes lots of time and man power. A lot of attention is often paid to the cost of development in software, while testing and aspects related to testing tend to get ignored. The cost and time associated with software testing are often comparable and sometimes even more valuable to the time and costs that are associated with the development of software.[5]

As the Digital Economy fuels the desire to outdo the competition, several IT organizations are charting out methods to optimize software testing, enhance testing productivity, curb costs, and reduce work hours; all of which can produce a major positive impact on the business.[9]

There are multiple ways of achieving testing optimization and cost reduction.

- Most software defects detected during testing originate in the early analysis phase. Hence defects that are detected early are less expensive to fix
- Early phase testing can provide early indicators of the software fulfillment and quality and enhance the familiarity and knowledge that testers have regarding the software [3].



2. PROPOSED WORK

Here are the fundamental things for doing randomization of testing:

2.1 Testing a shopping kart site:

A shopping cart website is taken and tested each part of the page with the help of different testing strategies. The lists of testing strategies used are given bellow:

1. Smoke testing
2. Sanity testing
3. Assertion Testing
4. All-pairs Testing
5. Boundary Value Testing

2.1.1 Testing Work

2.1.1.1 Login page testing

The testing for the login page is to be performed in which the values are not of the same and hence the randomization testing is used in order to verify the conditions

There are 5 main conditions has to be verified. Those are:

- Username
- Password
- Login
- Signup
- Forgot password

For the first two conditions, the randomization testing data is used and also for the signup function after the sign in page is navigated to sign up page it is used. The boundary value testing is done to verify the functions. It is also tested that the forgot password page has sent the one-time password (OTP) to the concerned email or mobile number. The diagram in fig 1.1 shows the testing ways in which the test is done.

2.1.1.2 Functions testing

In this functions testing, we use the smoke testing strategies to check the main module and the critical test functionalities. The functions are being tested one by one and checking the navigation of search and also the mouse motion event. For example, if the picture is clicked it should be navigated to the next page where the related content should be present.

2.1.1.3 Buy now and cancellation

When the picture is clicked it is navigating to the page where is content for the website is present, on that page the pin code is tested for several times. There are two functionalities Add to cart and Buy now. Add to cart is tested in a way that the item is added to the cart and also the cost which is changed during the offer period is also tested. In the buy now it is tested that the person who is buying the product is logged in or not and after that, in the address section the changes can be made and also there is "use current location" function which is also tested. To check the purchase of the product the product summary is shown and tested if that is the correct product that we have selected. There are payment options which the use can select any of the 4 options. After the order has been placed the order id will be generated and the date of delivery with the option of where the product is also been tested completely. The cancellation is done with fewer steps and the reason for cancelation is also taken into the storage area. The amount of product that we are going to buy is also tested.

2.1.1.4 Logout

The logout is not available as in the smoke testing; it is available in my account section inside the smoke testing module. It is tested that the after clicking the logout it is safely been logged out.

2.2 Comparison of two testing strategy

In the shopping cart website, I have divided it into 8 major critical test cases and neglected the other less priority test case. The way in which the critical functions are being decided is by the major use in those functions. The list below is the 8 major test cases:

No	Name
1	login
2	search
3	Click object
4	Product options
5	Details for order

6	checkout and cancelation
7	logout

I have tested all these functions in two ways they are

- Flow order testing
- Random order testing

In both these testing, the testing of all the functionalities is done and let's see the comparison.

1.2.1 Flow testing

Flow testing is similar to that of the waterfall model. In this flow testing, the test cases from 1 to 7 are executed one by one for several times and the time is taken for each execution is noted separately.

Test series	Time
t1	02:30
t2	02:29
t3	02:34
t4	02:35
t5	02:28

In this table, we can see how long it takes to test the functionality of the test suits. Each test suits take different time for execution, it depends on how to speed the internet connection is.

1.2.2 Random order testing

Random order testing is similar to that of the random testing. In random testing, random values are given to the testing data but in the Random order testing test cases are randomly allocated in order to reduce time and effort for the future use.

I tested the different ways in which the test cases can be allocated and tested each testing cases according to the pattern.

	Ways of tested	Time (hrs)	Error
t1	1,2,3,4,5,6,7	2:30	no
t2	1,3,2,4,5,6,7	2:56	no
t3	1,4,2,3,5,6,7	3:16	no
t4	1,5,2,3,4,6,7	4:55	no
t5	1,6,2,3,4,5,7	2:24	no
t6	1,7,2,3,4,5,6	0:40	error
t7	1,2,4,3,5,6,7	2:47	no
t8	1,2,5,3,4,6,7	2:36	no
t9	1,2,6,3,4,5,7	2:58	no
t10	1,2,7,3,4,5,6	0:45	error
t11	1,2,3,5,4,6,7	2:55	no
t12	1,2,3,6,4,5,7	2:35	no
t13	1,2,3,7,4,5,6	0:47	error
t14	1,2,3,4,6,5,7	2:47	no
t15	1,2,3,4,7,5,6	0:38	error
t16	1,2,3,4,5,7,6	0:45	error
t17	2,1,3,4,5,6,7	2:43	no
t18	3,1,2,4,5,6,7	2:25	no
t19	4,1,2,3,5,6,7	2:47	no
t20	5,1,2,3,4,6,7	2:30	no
t21	6,1,2,3,4,5,7	2:58	no
t22	7,1,2,3,4,5,6	2:30	no
t23	2,3,4,5,6,7,1	0:30	error
t24	2,4,5,6,7,1,3	1:00	error
t25	2,5,6,7,1,3,4	1:10	error
t26	2,6,7,1,3,4,5	0:45	error

	Ways of tested	Time (hrs)	Error
t27	2,7,1,3,4,5,6	2:10	no
t28	3,2,4,5,6,7,1	1:50	no
t29	3,4,5,6,7,1,2	2:23	no
t30	3,5,6,7,1,2,4	2:45	no
t31	3,6,7,1,2,4,5	2:33	no
t32	3,7,1,2,4,5,6	2:43	no
t33	4,2,3,5,6,7,1	2:58	no
t34	4,3,5,6,7,1,2	2:37	no
t35	4,5,6,7,1,2,3	2:30	no
t36	4,6,7,1,2,3,5	2:20	no
t37	4,7,1,2,3,5,6	1:00	error
t38	5,2,3,4,6,7,1	0:30	error
t39	5,3,4,6,7,1,2	0:56	error
t40	5,4,6,7,1,2,3	0:48	error
t41	5,6,7,1,2,3,4	2:25	no
t42	5,7,1,2,3,4,6	2:14	no
t43	6,2,3,4,5,7,1	2:47	no
t44	6,3,4,5,7,1,2	2:50	no
t45	6,4,5,7,1,2,3	0:26	error
t46	6,5,7,1,2,3,4	0:56	error
t47	6,7,1,2,3,4,5	0:45	error
t48	7,2,3,4,5,6,1	2:30	no
t49	7,3,4,5,6,1,2	2:53	no
t50	7,4,5,6,1,2,3	0:30	error
t51	7,5,6,1,2,3,4	0:30	error
t52	7,6,1,2,3,4,5	0:10	error

In the above table, the time taken to complete the test suits are listed and the least time taken for testing t27 is less compared to other test suits. The errors obtained for the other test suits are nothing but the interruption of other test cases.

1.2.3 The advantage of proposed system:

- I present the idea of “Random order testing” which is useful for a small project with flexible time.
- Large initial work with less future work
- Identifying the concept of other projects similar to the initial work and execution of the Patten used in the initial work and completing the similar project in less amount of time.
- Man power can be reduced after the initial work.
-

3. GRAPH

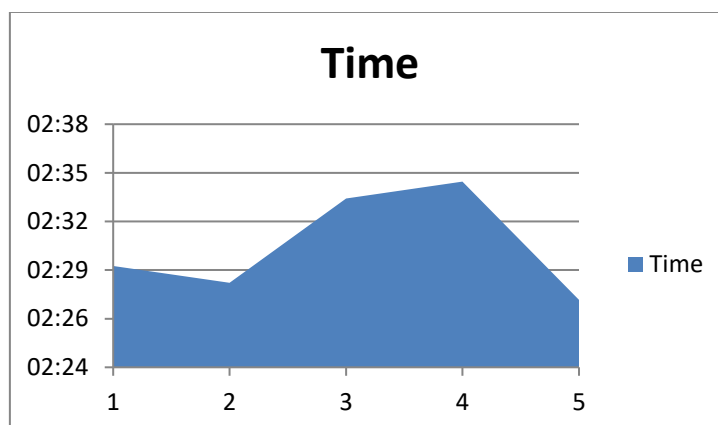


Figure 1

X -axis=time
Y -axis=test suits

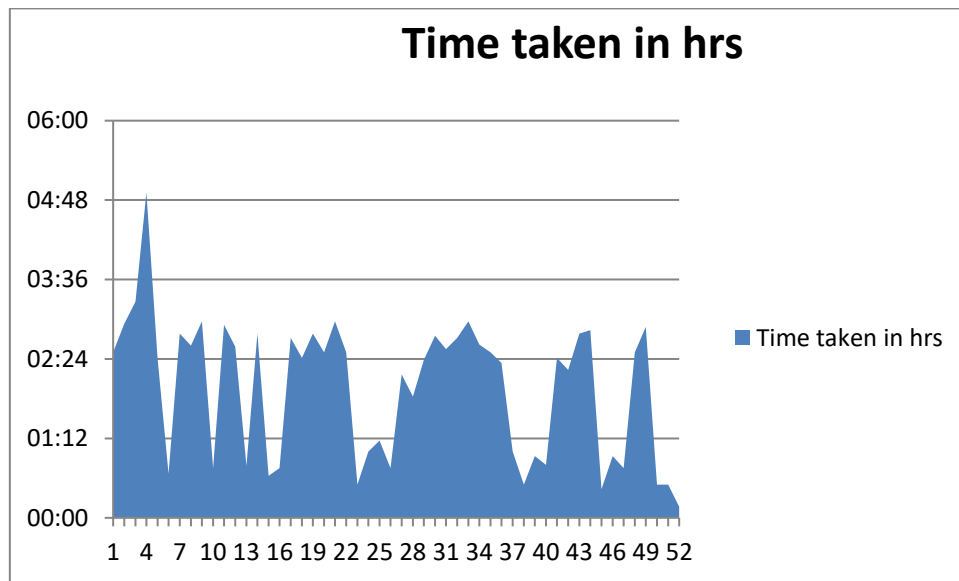


Figure 2

X -Axis = time
Y -axis= test suits

4. CONCLUSION

The comparison between the flow testing and the random order testing shows that the random order testing gives the idea of execution in the limited time but the execution of each test suits takes a long time. The similar documents can be taken in such a way and tested to get the minimum time and clarity on the project.

5. REFERENCE

- [1] Software testing practices and test management by NandreasSpillner, Thomas Rosnner. Page no: 23-26.
- [2] Software testing, principles, techniques and tools by M.G.Limaye, Tata mc graw hill education private limited. Page no:102-112.
- [3] Software testing 2nd edition by Ron patter, jams publication. Page no: 13-20,53-62,30-337.
- [4] Fundamentals of software testing by Aditya p. mathur .pageno: 585-588.
- [5] Components based software testing by hans-gerhand Gross, page no: 73-81.
- [6] Essential software testing by grea fournnier page no:9-18.
- [7] Testing IT an off-the shelf software testing process. By john-watt kins-cambridge university press. Page no: 90.
- [8] Testing object oriented software lifecycle solution by imrambashir ,amrit L. goel. Pg:36.
- [9] Quick test professional byvinnakottaravisakar .page no:29-40
- [10] Manage software testing by peter Farrell, auerbach publications.
- [11] Wile testing computer software by cemkaner and jack falk. Page no: 17-24, 65-81.
- [12] Software testing techniques by borisbeizerpublishef by dreamtech pageno:59-91,173.
- [13] Software testing fundamentals by MarnieL.Hutcheson age no:112, 205-239.