INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

Test Case Prioritization on Shopping Websites

V.Rama Krishna[1], S.Venkata Krishna[2], P.Sneha[3]
G.Pragna Chowadary[4], P.Sahithi[5]

[1]Associate Professor
K L University
Vaddeswaram,Guntur, AP, INDIA-522502
v.ramakrishna@gmail.com
[2] Assistant Professor
K L University
Vaddeswaram,Guntur, AP, INDIA-522502
s.venkatakrishna@kluniversity.in
[3,4,5]B.Tech
K L University
Vaddeswaram,Guntur, AP, INDIA-522502
[3] snehapasala@outlook.com, [4]iampragna@gmail.com, [5]sahithi.paprepally93@gmail.com

Abstract*: Shopping sites usage, which is increasing steadily, change the state based on incoming events. The various events pose a challenge to test the functioning of the Shopping websites because there are a large number of possible event sequences that users can invoke through a user interface which generally include adding an item to the cart, editing the contents etc... Many models have been developed for testing the Shopping Websites, but they are not satisfying the needs of the end-users. So, an attempt is made to develop a Single model that is generic enough to test the Shopping websites. Different prioritization criteria are given in the literature for testing the web application test cases of a test suite. In our study we propose a criterion based on the transactions carried out in the Shopping websites. The new criterion is applied on four Shopping Websites and its fault detection effectiveness is ascertained using APFD metric. Transaction based prioritization provides us with the Shopping Websites which detects faults in the test suite quickly and effectively.*

Keywords: *Shopping Website, Single Model, Prioritization criteria, Transactions, Faults, APFD.*

1. INTRODUCTION

In this paper, test cases within a test suite of Shopping Websites are prioritized based on APFD metric. Following section covers the components present in our proposed work.

1.1. Shopping Websites

Online shopping was started at 1994 by pizza Hut. In the same year a German company Inter-shop Communication introduced world's first online shopping system. After that Amazon (1995) and eBay (1996) was launched one by one. Now online shopping is so much popular that E-commerce B2C product sale in USA touched around $200 billion, it is almost 1/10 of total retail product sale in USA. Research says that online retailer in USA will worth around $300 billion by 2015. Not only in the USA, is online shopping usage increasing all around the world.

1.2. Prioritization Strategies

Now for these Shopping Websites test case prioritization will be done. In previous papers, [1] test case prioritization is done on relevant data basing on some strategies . These strategies prioritize test cases of a test suite. [3-4], [6], [8-10].These strategies are nothing but criteria's. Some of the criteria's are:

- interaction-based
- count-based
- frequency-based

Interaction –based criteria: Test case depends on parameters and values for execution. It has two types:

- 1-way
- 2-way

1-way and 2-way parameter -value interaction coverage techniques select tests to systematically cover PV-interactions between windows. 1 -way criterion s elects the test which does not appear in previous tests . It covers the new PV, but not the one which is already covered. In the second type, 2-way PV interactions are present. It chooses new PV which is uncovered.

Count-based criteria: It is based on count of number of windows, actions, parameter-values covered.[1][11]. Here there are 3 categories:

- Unique-window coverage
- Action-count based
- PZV-count based

- Unique window coverage gives preference to those test cases that cover the most unique windows which are not covered in previous tests. Here it is hypothesized that faults will be exposed when

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

windows are visited and that windows should be visited as -soon-as possible.

- Action-count based criteria prioritizes the number of actions in each test (duplicates are also included).Here preference is given to those that includes maximum number of actions. It can be divided into Action-StoL and Action-LtoS.

- Action-StoL (small to large) gives priority to test cases with smallest number of actions.
- Action-LtoS (large to small) gives priority to test cases with smallest number of actions

-In Parameter Value count based tests are prioritized by the number of parameters that are set to values in a test case. Duplicates are also included here.

Frequency-based criteria: [1][12]Preference is given to the windows. It observes how many times the window is being accessed in the test cases. The following three criteria's differ in how they view the frequency of presence of a window sequence in a test case and thus produce different prioritized orders.

- MFPS
- APS
- Weighted-Freq

- MFPS (Most Frequently Present Sequence of windows) determines the number of times each sequence appears in the test suite. MFPS gives importance to a particular window sequence.

- APS (All Present Sequence of windows) accommodates all sequences during prioritization. MFPS gives importance only to the frequency of occurrence of a single most frequently present sequence .There is the possibility of losing important information about other frequently present sequences which are not included. So, APS checks frequency of occurrence of all sequences to order the test suite. APS selects test cases based on only one sequence.

- Weighted Sequence of windows (Weighted-freq) assigns each test case a weighted value based on all the window sequence it contains and the importance of the window sequence. Initially, frequency of appearance of each unique sequence of windows in the test suite is identified. And then weighted matrix is built for each unique window sequence. Prioritization criteria improve the rate of fault detection of the test cases over random orderings of tests. All these three criteria's are briefly mentioned here. Apart from these criteria's we have proposed a new criterion namely **"Transaction based criteria".**

2. WORK

A. *Transactions:* Individual operation is said to be transaction. A transaction comprise unit of work performed within a system. It is independent of other transactions. Each transaction succeeds or fails as a complete unit but cannot remain in an intermediate state. [2] All transaction processing which are interactive allows multiple individual operations to be linked together as a single indivisible unit. Trans actions checks that all operations are completed without errors. In this paper as we are concentrating on Shopping Websites each transaction is considered as a "test case".

Transactions can be like:

- Creating an account. Browsing various items in various categories.
- Searching for some particular item.
- Adding an item to the cart.
- Removing an item from the cart.
- Editing the quantity in the cart.
- Adding items to wish list.
- Proceeding for the checkout. Giving address for the delivery.
- Payment mode.
- Placing the final reviewed order.

Apart from these there are many such multiple transactions in Shopping Website. Here we have just given the sample transactions. But, in our work, we consider the first five Transactions as test cases.

They are:

- Account Creation
- Loign Authentication
- User Interface
- Showing Relevant Search Results
- Purchasing The Product

B. Faults: Mistakes in the code is said to be Fault. A fault is the cause of an error. Software fault lies in software, a hardware fault lies in hardware. In simple terms, error leads to fault. Each application has a default fault matrix which is the representation of a set of faults detected by each test case. Faults in the Shopping Websites are seeded manually. Additionally, some naturally occurring faults are also discovered during deployment are also seeded in the application. Faults help in extending functionality and its development. High value for faults means that each test case in the suite is detecting a large number of the faults [1],[3],[7],[13] Low value for faults indicates that each test case detects only a small number of the faults. Here, in our paper faults in each shopping websites are found out manually. We have found out faults up to the maximum extent. Faults disrupt the user from doing the desired work. It acts like an error. In our work, as we deal with email transactions, in those terms we can say a Fault as the one which interrupts users stepping into the next window. The test cases for our test suites have relatively different fault values for different email applications (some

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

high and some low). We observed that some faults are repeating and some are unique. Some duplicated faults occur with different fault time and some with same fault time. Faults observed for all Shopping Websites are given in the below tables.

***C. Time*:** Here faults are found out with respect to time. Time taken for a particular fault to occur is calculated in seconds.[3] Time taken for a specific fault to occur differs from one Shopping Website to other. Time is unique for all faults.Time changes for each and every fault in a test case of an Shopping website. As we have already told above that some duplicated faults occur with different fault time and some with same fault time. All these conditions may vary based on following factors:

- Fault Type
- Internet Speed
- Entry Of Authenticated Data

Following 1 to 5 shows the faults occurred and its relative time for four Shopping Websites.

Shopping Website1: ebay

Test Case1: Account Creation

S.NO	FAULTS	TIME(sec)
1	Invalid email id with correct format(having special characters)(F1)	3
2	Incorrect format (F2)	1
3	Non-existing email provider (F3)	1
4	Mismatch of provided email addresses (F4)	1
5	Non-existing email address(F5)	3
6	Email id is already registered(F6)	2
7	Password must be alphanumeric and of length 6-15 character (F7)	2
8	Week password with less characters(F8)	1
9	Password mismatch(F9)	1
10	Invalid phone number(F10)	1

Table 1: Faults occurring during "Account creation" in a Shopping Website

Test Case2: Login Authentication

S.NO	FAULTS	TIME(sec)
1	Invalid email id or email format (F1)	1
2	Mismatch of e-mail Address and password(F11)	1
3	Non-existing email address(F5)	1
4	Incorrect answer for the security question(Forgot password)(F12)	0
5	Lost email address (requires updation of new email address) (F13)	3
6	Time out (F14)	2

Table 2: Faults occurring during "Login Authentication" in a Shopping Website

Test Case3: User Interface

S.NO	FAULTS	TIME(sec)
1	Not Understanding complex input/s from user (F15)	2
2	Invalid or broken URL (F16)	2
3	Un ordered search results (F17)	1

Table 3: Faults occurring during "User Interface" in a Shopping Website

Test Case4: Showing Relevant Search Results

S.NO	FAULTS	TIME(sec)
1	Exceeding cart limit(F18)	5
2	Unknown product name(F19)	2
3	Inconsistent data (F20)	5
4	Out of stock(F21)	2
5	Internal errors(F22)	1
6	Adding items that are limited in number. (F23)	1

Table 4: Faults occurring during "Showing Relevant Search Results" in a Shopping Website

Test Case5: Purchasing The Product

S.NO	FAULTS	TIME(sec)
1	Invalid address(unknown city name) (F24)	1
2	Invalid pincode (F25)	1
3	No name specified for delivery (F26)	1
4	No address is specified(F27)	1
5	No city is specified(F28)	1
6	Address not recognized(F29)	2
7	Invalid phone number(ex:123#)(F10)	2
8	Invalid credit card number(F30)	0.4
9	Expiration date error(F31)	0.3

Table 5: Faults occurring during "Purchasing the Product" in a Shopping Website

Thus these tables show the faults occurred during those particular transactions in ebay. We can see that there are totally 31 faults. The numbering in the brackets (F1,F2,F3,….) are given for the convenient purpose of our understanding, which represents that it is first fault, second fault and so on till thirty first fault. In this way, faults are found out with time for all the test cases of the remaining Shopping Websites(Amazon, Flipkart, Jabong).

D. Fault Matrix: From these faults and time obtained previously from the above tables "Fault Matrix" is constructed.

	T1	T2	T3	T4	T5
F1	*	*			
F2	*				
F3	*				
F4	*				

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

F5	*	*			
F6	*				
F7	*				
F8	*				
F9	*				
F10	*				*
F11		*			
F12		*			
F13		*			
F14		*			
F15			*		
F16			*		
F17			*		
F18				*	
F19				*	
F20				*	
F21				*	
F22				*	
F23				*	
F24					*
F25					*
F26					*
F27					*
F28					*
F29					*
F30					*
F31					*
Number of Faults	10	6	3	6	9
Time(sec)	16	8	5	16	9.7

Table 6: Fault Matrix

F1, F2, F3… F31 : Total number of faults
T1, T2, T3, T4, T5 : Test cases
*: Fault occurring in a specific test case.

E. Rate of Fault Detection (VTi): Faults are detected for each test case in the test suite. [3][7][13] Total time taken for each test case is presumed. VTi is the rate of fault detection. It is calculated using following formula:

VTi = fault / t ime Eq.1

As we have found out fault and time init ially, now we mus t calculate VTi.

VT1=10/16=0.62
VT2=6/8=0.75
VT3=3/5=0.60
VT4=6/16=0.37
VT5=9/9.7=0.92

Based upon these VTi values, test cases will be prioritized in decreasing order:

T5, T2, T1, T3, T4

F. APFD: A PFD is defined by Rotherme l et.al. Rate of fault detection of the prioritization criteria is measured using "Average Percentage of Faults Detected" (APFD) metric [3][4]. APFD evaluates effectiveness of prioritized test suite order. It is calculated by taking the weighted average of the number of faults detected during the run of the test suite. APFD is only possible when faults are available. They are used for evaluation.

$$APFD = 1 - \frac{TF1 + TF2 + TF3 + \ldots + TFM}{m*n} + \frac{1}{2*n} \quad \text{Eq.2}$$

TFI : position of the firs t test case "t" in T
m : nu mber of fau lts
n : nu mber of tes t cas es
F : fau lt
T : tes t suite

Basing on this formula APFD is calculated. But prior to that "TF" should be found out. Now let us find out TF1 value. In the above fault matrix, we can see that fault F1 is firs t given by test case T5 (according to priority order T5, T2, T1, T3, T4). Now check the position of T5 in prioritized sequence, which is firs t. Thus, value of TF1 is 1.Similarly from T F1 to TF11 value is 1. Now, fault F12 is firs t given by T3 according to prioritized order. So TF12=2. Similar pattern goes from TF13 to TF19. Fault 20 is given by test case T5 s o thus TF20=1

TF1 = 2	TF9 = 3	TF17 = 4	TF25 = 1
TF2 = 3	TF10 = 1	TF18 = 5	TF26 = 1
TF3 = 3	TF11= 2	TF19 = 5	TF27= 1
TF4 = 3	TF12= 2	TF20 = 5	TF28 = 1
TF5 = 2	TF13 = 2	TF21 = 5	TF29 = 1
TF6 = 3	TF14= 2	TF22 = 5	TF30 = 1
TF7 = 3	TF15 = 4	TF23 = 5	TF31 = 1
TF8 = 3	TF16 = 4	TF24 = 1	

Upon substituting all these TF values in APFD:

APFD=1-((2+3+3+3+2+3+3+3+3+1+2+2+2+2+4+4 +4 +5+5+5+5+5+5+5+1+1+1+1+1+1+1+1+1)/(5*31))+(1/5*2) =0.47

0.47 is the obtained APFD value for prioritized test order. Now, APFD for non-prioritized test-order is calculated in the same manner. Non-prioritized order: T1, T2, T3, T4, T5

APFD=1-((1+1+1+1+1+1+1+1+1+1+2+2+2+2+3+3+3 +4+4+4+4+4+4+5+5+5+5+5+5+5+5)/(5*31)) +(1/5*2) = 0.42

In this case, after calculations we observe that APFD value is high for prioritized sequence (0.47>0.42). Thus in first Shopping Website i.e., ebay has higher value for prioritized test suite order. In this way, APFD calculations are done for all the Shopping Websites. They are summarized in table 7.

3. EXPERIMENTATION AND ANALYSIS

We have tested four Shopping Websites by taking five transactions for each application. Total number of faults occurring within those transactions /test cases is discovered by manual seeding. Simultaneously, time is overviewed and in some cases it will be presumed. [3] Time is measured in seconds. After finding out the main components of all transactions i.e., faults and time 'fault matrix' is constructed. From this matrix, fault rates are

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

Table 7: APFD CALCULATIONS FOR SHOPPING WEBSITE

	EBAY.COM					AMAZON.COM					FLIPKART.COM					JABONG.COM				
	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5
FAULT S	10	6	3	6	9	10	6	3	6	9	10	6	3	6	9	10	6	3	6	9
TIME	16	8	5	16	9.7	11.8	7.2	3.9	81.4	16	17.7	10	24	38	21	25	22	5	41	14.5
RATE OF FAULT DETECTION (VTi)	0.625	0.75	0.6	0.375	0.92	0.84	0.83	0.76	0.07	0.56	0.56	0.6	0.125	0.15	0.42	0.4	0.27	0.6	0.14	0.62
PRIORITIZED TEST ORDER	T_5,T_2,T_1,T_3,T_4					T_1,T_2,T_3,T_5,T_4					T_2,T_1,T_5,T_4,T_3					T_5,T_3,T_1,T_2,T_4				
APFD	0.47					0.43					0.46					0.44				
NON-PRIORITIZED TEST ORDER	T_1,T_2,T_3,T_4,T_5					T_1,T_2,T_3,T_4,T_5					T_1,T_2,T_3,T_4,T_5					T_1,T_2,T_3,T_4,T_5				
APFD	0.42					0.42					0.40					0.41				

detected i.e., VTi using the above s aid formula. No w, here test case prioritizing will be done. [1] As we have already discussed previously in section 1.2 about the three criteria's mentioned above (interaction -based, count-based and frequency-based). In this work, prioritization criteria will be transactional fault rate detection using APFD metric. 'Average Percentage of Faults Detected' determines the effectiveness of test suite orders either it may be prioritized/non -prioritized. Thus for measuring this certainty, APFD is chosen due to its effective results. So, in our work we have done this calculation for both the prioritized and non - prioritized test orders.

In the table.7, we can observe that there are four Shopping Websites namely ebay, amazon, flipkart, jabong. Each one of them has different faults with different timing. Some Shopping websites may have same faults. And some may have different faults. For each e mail application five test cases or transactions are taken. Total time taken significantly varies. For some faults it is 1second and for some it's around 6 seconds (which means that it's taking mo re time for processing /completing the task or in simple terms we can s ay that it's consuming user's time making him sit in idle state). The fault which takes less time is relatively much better than Shopping Websites . So, from our calculations we found out that Amazon shopping website has least APFD value i.e., 0.43 which depicts that it detects faults quickly without any time delay. Other Shopping Websites also detect faults but when compared to Amazon there is a difference of fraction of seconds. Test suites are incremented 10% in each step and they are executed effectively.

4. CONCLUSION

In previous papers many criteria's are used for prioritizing the test cases. Interaction -based criteria used unique parameter-values. Count-based criteria considered maximum number of actions and unique windows. And frequency -based criteria considered the most frequently present window, weighted values, etc. These criteria's produced test suite orders for Web Applications. So, as a part of this , we proposed a new criterion 'Trans action' for test case prioritization in order to improve efficiency of testing. Transaction based criteria takes Shopping Websites, prioritizes test cases and tests the data for faults. These faults percentage is ascertained by APFD metric. Our testing on Shopping Websites produced results which showed that prioritization is important for executing test suites and detects faults effectively. Prioritization improves the efficiency of testing with respect to the four Shopping Websites and prioritized test cases are more effective. As a future part we would try to impart work on Commercial Applications like Database Management software like oracle, sqldbx by taking multiple trans actions together.

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

REFERENCES

[1] Renee. C. Bryce, Sreedevi Sampath and Atif. M. Memon, "Developing a Single Model and Test Prioritization Strategies for Event-Driven Software," in IEEE Trans. On Software Engineering, vol.X, no. X, J an. 2011

[2] http://en.wikipe di a.org/ wiki/Databasetrans ac t ion

[3] Praveen Ranjan Srivastava, "Test Case Prioritization," in Journal of Theoretical and Applied Information Technology, 2005-2008 JATIT

[4] Alexey G. Malishevky, Joseph R. Ruthruff, Gregg Rothermel, Sebastian Elbaum, "Cost-Cognizant Test Case Prioritization," 2006

[5] http:// www.softwaretestinggenius.com/ar ticalDetails ?qry=715

[6] S. Elbaum, A. Malishevsky and G. Rothermel "Test Case Prioritization: A family of empirical studies ," in IEEE Trans. on Software Engineering , Fe b 2002.

[7] J. C. Munson and T. M. Khoshgoftaar, " The Detection of Fault -Prone Programs ," IEEE Trans. On Software Engineering, vol.18, no.5,pp. 423-433,1992.

[8] R.C.Bryce and A. M. Memon, "Test suite prioritization by interaction coverage," in Proceedings of The Workshop on (DoSTA2007); ACM SIGSOFT Symposium on the Foundations of Software Engineering. Dubrovnik, Croatia: ACM, Sep. 2007, pp. 1 –7.

[9] J. A. Jones and M. J. Harrold, "Test-suite reduction and prioritization for modified condition / decision coverage," Trans. On Software Engineering, vol. 29, no. 3, pp. 195 –209, Mar. 2003.

[10] D. Jeffrey and N. Gupta, "Test case prioritization using relevant slices," in the International Computer Software and Applications Conference. IEEE Computer Society, Se p. 2006, pp. 411 –418

[11] J. Lee and X. He, "A methodology for tests election," Journal of Systems and Software, vol. 13, no. 3, pp. 177–185, Nov. 1990.

[12] M. Grindal, J. Offutt, and S. Andler, "Combination testing strategies: a survey," Software Testing, Verification, and Reliability, vol. 15, pp. 167–199, Mar. 2005.

[13] D.R.Kuhn, D.R.Wallace, and A. M. Gallo, "Software fault interactions and implications for software testing," IEEE Trans. on Software Engineering, vol. 30, no. 6, pp. 418 –421, Oct. 2004.

[14] Transaction Based Prioritization Strategies for Email Applications, V.Rama Krishna, M.R.Narasinga Rao, Sk.Sunaina Sulthana, T.Geethika.

[15] "Prioritizing Test Cases For Regression Testing" ,Gregg Rothermel, Member, IEEE Computer Society, Roland H. Untch, Member, IEEE Computer Society, Chengyun Chu, and Mary Jean Harrold, Member, IEEE Computer Society.