

Risk Scenarios on Web Applications

Leticia Davila-Nicanor*

Universidad Autónoma del Estado de México, Centro Universitario UAEM Valle de México, Atizapán de Zaragoza, Estado de México, México.

* Corresponding author. Email: ldavilan@uaemex.mx

Manuscript submitted January 15, 2018; accepted March 1, 2018.

doi: 10.17706/jcp.13.8.979-987

Abstract: In this proposal a process to estimate test cases prioritization on Web application have been developed. The Software Reliability Engineering approach have been related to statistical simulation in order to develop a model. The model is the base to obtain the navigation sequences and risk probability on components. The results become a guide to establish the test coverage through the knowledge of the most critical paths and components on the system.

Key words: Regression functional test, test case prioritization, web application.

1. Introduction

Nowadays the Web applications are a part of daily life on humanity. On these systems operate economics process as banking services, business sites, social networks and e-commerce. Many companies have established Intranets in order to linking data sharing in their productive process. In the scientific activity, big knowledge databases have been implemented, and the important events are broadcast by these applications.

A software system is subjected to continuous evaluation due to the system requirements evolution. The perspectives to Web applications are big, today millions of users are supported by the Web applications infrastructures and thousands of requests are served simultaneously. Because increasing organizational demands toward Web Applications and continuous changes in the requirements, increases the importance of improving the testing approach. Regression testing is the facto activity to address the testing problems caused by software evolution [1].

After changes have been made to a Web application, regression testing can be performed to detect new faults. On regression testing three major branches of the most widely used technique have been identified: test suite minimized (reduction), test case selection and test case prioritization. Test suite minimization aims at obtaining a minimal subset of a test suite that preserves a specified adequacy criterion [2]. Test case selection concerns reusing test cases from an existing test suite to test the modified part of the program [3]. Test cases prioritization schedules test cases for execution in an order that attempts to increase their effectiveness at meeting some desirable properties, such as the rate of fault detection [4].

When software tests are performed, if the test cases are inadequate, the testing process is adversely affected and most of potential failures are not located [5]. The test cases prioritization to set by operational sequences in the model navigation is the main goal, be-cause is expensive to evaluate all combinations between the components [6]. In order to select and to prioritization the test cases, some proposals have explored the architectural approach [7] and [8]. Another interesting arise is the relation between Web applications functionality and user demand to do the navigation model [9] and [10], these usage techniques are log files [9], web crawlers [10] and statistical simulation to represent user activity on a navigation

model.

In this proposal we have analyzed functional and user demand attributes on Web applications to make a test cases prioritization model. The Software Reliability Engineering [11] approach have been related statistical simulation in order to develop the model. In the analysis of this model the navigation sequences and components with high probability of risk have been obtained. The results become a guide to establish the test coverage through the knowledge of the most critical paths and components on the system.

This paper is organized as follows, in section 2 the related work is presented. In section 3, the Operational Profile Dependency Graph (OPDG) is described and the process to be obtained. In section 4 the OPDG is applied to practical case study. Finally, in section 5 and 6 future work and conclusions are presented.

2. Related Work

In the Ricca and Tonella proposal [9], the Web application evaluation is from architectural point of view. In this work from UML model a graph is proposed, the graph is the base of ReWeb tool. A model of web application is developed using reverse engineering and TestWeb is the artefact to executed test cases prioritized from the modeled computed. The model was implemented on a semiautomatic context. In this case the architectonical design is based on frames and static HTML. Qi, Kung and Wong [8] proposed a model from events of Web site evaluated. The authors have generated an Event Dependencies Graph (EDG) that subsequently it is transformed in a tree of events to estimate functional regression testing. The authors reduced redundancy in the test cases and thus limiting the universe of path navigations involving the assessment of a Web site. Both proposal have demonstrated the efficiency in tests cases determination, but taking account the requirements importance, these do not have a close relationship from a functional approach, in these cases the accurately on the results is limited. Kong [12] proposed a testing model, it has two aspects: cost and demand, the perspective is projected in terms of structure, environment and location. In this case traceability is modeled between abstractions, the system structure, environment and location from the point of view of user profiles. The traceability is a requirements property, the main goal is mapping these with the software architectural. Bansal and Sabharwal [13] have proposed a model based on user requirements and system navigation events taking the client server architecture of Web applications to develop test sequences. Liping and Zhongsheng have developed a proposal to determine the test coverage on Web applications [7]. In this proposal authors have used UML diagrams to describe the requirements specification, these are formally transformed, UML Diagrams are converted to XMI output on the fly through the Finite State Machine algorithm to determine test cases. Broadly these proposals have been developed taking account functional approach. The results obtained are relevant, though the perspectives are good also it is possible improve them, due to kind of application is necessary to take account the prospect on use on the system. The Web applications on a real context must to attend thousands or millions of requests, in this case the number of request also impact to the Web applications operation. Sprenkle, Pollocky and Simko [14] have presented an empirical study to support specific design on constructing usage based navigation models to determine test cases. The study is very interesting due to it emphasizes the importance that the use on Web application is not determinist, it has a random perspective, due to many conditions on real context are affecting. This approach also allows us to see that architecture sometimes not respond the needs of users.

3. Process to Define Risk Scenarios on Web Applications

To accord the Software Reliability Engineering (SRE) [11] approach, an operational profile is a set of operations (major system logical task) of a system with a prioritization by their occurrence probability, this

probability is related to user demand in the system. In this proposal the OPG is a deterministic model, where in the Web application functionality is represented through a directed graph. In the graph developed, relations and transitions (vertex and edges) have been defined from SRE approach, while weight is related to components complexity that solve functional operations.

3.1. Process to Define the Operational Perfil Graph

Definition 1. The directed graph *OPG* is defined as a non-empty set of vertex $V = \{v_1, v_2, \dots, v_n\}$ and a set of edges $A = \{a_1, a_2, \dots, a_n\}$. The relationships are associated with the functionality of Web application evaluated.

1. To define business goals in the Web application
2. To set operational profiles op_j from the requirements specification, the op_j is related to actors from UML use case diagram
 - a. The vertices derived from system access (v_s) represent operational profiles $\{v_{op1}, v_{op2}, \dots, v_{opj}\}$ on the system.
3. To set relations operations o_i associated with operational profiles op_i . These operations are related to the actor roles by the UML use case diagrams
 - a. The vertices derived from each profile $\{v_{op1}, v_{op2}, \dots, v_{opj}\}$ represent the operations associated with profiles $\{v_{o11}, v_{o12}, \dots, v_{oji}\}$
 - b. The vertices that relate to operations deriving design system are represented with the nomenclature $\{v_{d1}, v_{d2}, \dots, v_{di}\}$
 - c. The edges that connect vertices indicate the information flow of operational profiles with the operations
4. To establish the set of components $C_1..C_i$ (or modules) in the software architecture that solve the operations located in point 2 and 3
5. To evaluate each component C_i with a complexity metric. In this case, if the components have been developed in different programming languages is necessary to find a complexity metric common to the programming languages
6. To set the weight to arcs according to the complexity metrics of components set that solve the transition relationship between the nodes of the g *OPG*)
7. To run an algorithm to estimate the shortest route in the *OPG*. It could be the Floyd-Washall algorithm:
 -Given a directed graph $G = (V, E)$, a weight $d: E \rightarrow R +$ assigned and well established vertices in Vs , find the shortest path from s to each vertex $v \in V$.

where:

S_k = The navigation paths obtained by the Floyd-Washall algorithm

n = Navigation paths number

8. To sort the paths in order descending

The algorithm discovers the navigation paths denominated Scenarios (S_k), they are sequences that represent test cases in the Web application, the order is ascending, in the first instance are found the less complex paths, finally the most complex.

3.2. Process to Estimate the Occurrence Probability Rate on Navigation Paths.

The influence of user requests on the behavior of Web application is so important. In this proposal those events are related to occurrence probability on the model in order to establish test cases prioritization. The occurrence probability rate is determinate in each operation of Web application by mean statistical simulation oriented to discrete events, the logical model on the simulation is according to the OPG's functional operating. The occurrence probability rate is determine by the functions 1 and 2

$$OrateVoij = Operationj_Ocurrence / Total_simulation_time \tag{1}$$

$$PoSk = OrateVoij / \sum_i^m \sum_j^n OrateVoi \tag{2}$$

where:

$OrateVoij$ = Ocurrence rate on vertex ij

PoS_k = occurrence probability of vertex operation

rateVo_{ij} = operation occurrence vertex ij

j = operation profile number, i = operation number associated to operational profile

3.3. Risk Probability Process

The Risk Scenarios determination is based on Risk Probability. This probability is the relationship between Occurrence Probability rate and scenarios weight according to following definition:

Definition 2. Being risk probability of a navigation path RP_{Sk} equal to the relationship between complexity level and occurrence probability in navigation path.

$$aSk = Wsk / \sum_1^i Wsj \tag{3}$$

$$RPSk = aSk * PoSk \tag{4}$$

where:

a_{Sk} = Coefficient of Complexity Variability on S_k

W_{Sk} = Weight of the navigation path

m = Total navigation paths

RP_{Sk} = Risk probability of S_k .

Po_{Sk} = Occurrence probability of the navigation path

Definition 3. The Occurrence probability OP_{Ci} of each component is calculated by the linear equations system solution of the RP_{Sj} navigation paths in relationship with the C_i component implied in the path and the Coefficient of Complexity Variability.

$$\begin{cases} a11OPC1 + \dots + a1iOPCi = RPS1 \\ a21OPC1 + \dots + a2iOPCi = RPS2 \\ \dots + \dots + \dots = \dots \\ aj1OPC1 + \dots + ajiOPCi = RPSj \end{cases} \tag{5}$$

where:

$OPCi$ = Occurrence Probability of the Component i .

a_{ji} = Coefficient of Complexity Variability at the Component i on Scenario (S_j).

When the risk probability to components and navigation paths have been obtained, it is possible to determinate the prioritization test cases on the coverage, in order to maximize opportunities to locate faults with less resources on time and effort.

4. Study Case

For implementing the model in a real case, an evaluation of a Web-based Software for an On-line application to graduate courses inscriptions in a University (SOGU) was used. SOGU's functionality is according to following, the system have four main operational profiles: (a). Professors, (b). Students and (c) General Public Users and (d) Academic Coordinator. To access is necessary have account and password. Each type of user is allowed to view only a part of the system related their responsibilities. In general this system is capable of allowing students to register for courses in a given University, and also allows Professors to register their courses, verify the data provided by the students and provides grades for their courses. Students and Professors can Insert, Delete, Update and Read information from their own databases. General public users can only read courses information. This system is composed of 3 different databases: the courses database, the student's database and the Professor's database. The system controls the access of different users and manages the information flow provided to the users. Changes to databases are allowed only during specific time periods (e.g., start of the terms), this period is established by the Academic Coordinator. The coordinator can register new users and make changes at the level of overall system administrator. The UML use case diagram is presented in Fig. 1.

4.1. Process to Estimate the Occurrence Probability Rate on SOGU System

The SOGU’s UML diagram was the entry to make the OPG. The model resulting from applying this approach in the SOGU system, it can be seen in Fig. 2. Vertex v_{s1} and v_{s2} represent the system access, v_{op2} , v_{op3} , v_{op4} , and v_{op5} represent the general public, students, professors and academic coordinator views. The vertex v_{o7} , v_{o9} , v_{o11} , v_{o12} , v_{o14} , v_{o18} , v_{o19} , v_{o21} , v_{o23} and v_{o25} represent the operations derived from operational profiles. The weight on each edge is assigned by estimating of Cyclomatic Complexity of the components implicit the functional operation, if there is more than one embedded component then the sum of weights is done. The results are shown in Table 1, in the first column the identifier (ID) of Scenarios are reported, the next column shows the functional relationships, in the third column the path weight is reported, the fourth column shows sequence of related components. In this case the navigation paths are ordered according to their weight, i.e. its complexity, thus S_1 is the most complex sequence and S_{11} is the least complex.

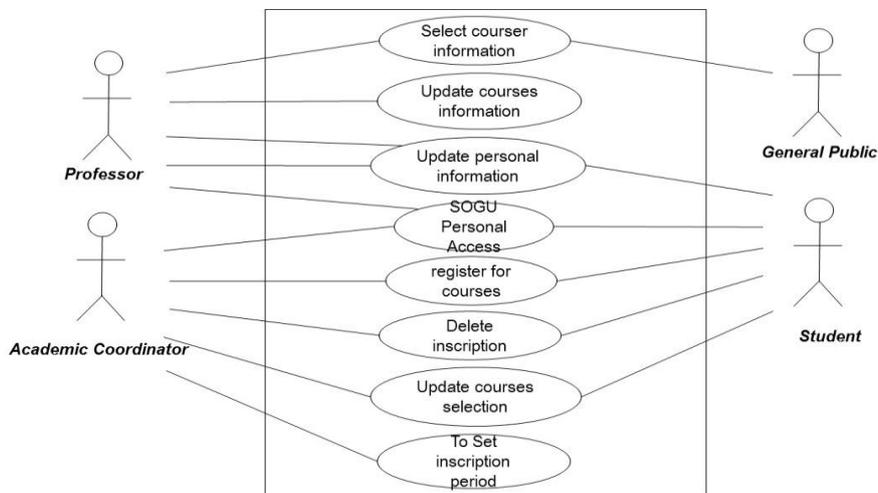


Fig. 1. SOGU’s functionality.

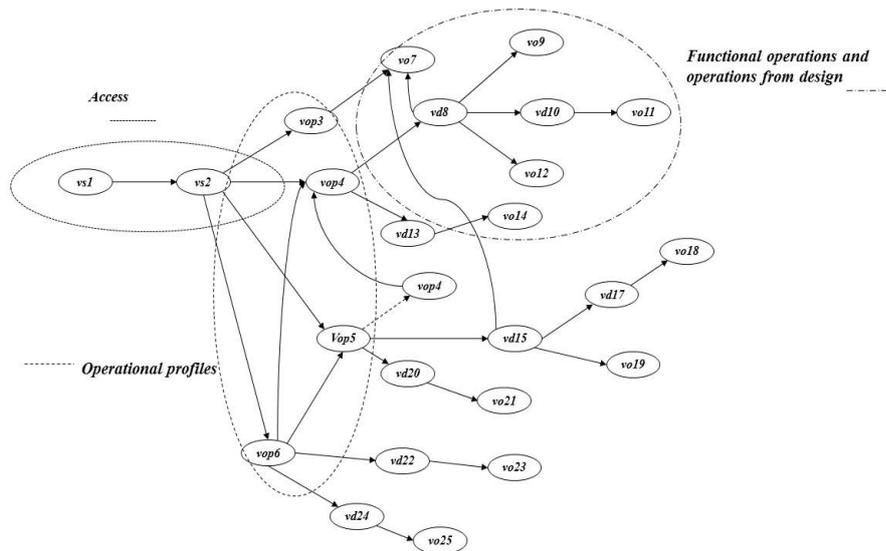


Fig. 2. OPG’s SOGU.

The logical model on the simulation is according to the OPG’s functional operating. To estimate the use occurrence probability, 5000 simulations were executed, with which the average frequency of visits for

each operational profile and operation estimates were made. The execution time for each simulation was 1000 units of time. To analyze the results obtained in the logs records reported, Shell scripts was implemented.

The simulation conditions are as follows:

1. The time of arrival for users to the system is determined by an exponential distribution $\mu = 5$.
2. Server queues M/M/1 type in the pool of application servers, this kind of queue server was used because one server attends the service in the real context.
3. To determine whether the user expects the server's attention in the queue, a probability of $1/(n+1)$, where n represents the current queue size was used.
4. The number limit of users is to accord the total users in the system with k users (where $k = 3400$).
5. Depending on the coverage of the evaluation, users can perform only x types of operations (transactions). The operating time depends on the user related transactions allocated to each operational profile coverage.

In the simulation the k parameter is related to the system capacity and it is according to rank handled in other investigations [9]. An important benefit with this technique is that the simulations were carried out in a short time, on an environment with a RISC processor and Solaris 2.7 platform, this process took only 4 hours to run 5000 simulations

In the Table 1, the fifth and sixth columns show the value of the probability of occurrence and the value of the Risk Probability for the each navigation path, test case or scenario S_j . It is important to observe how the values obtained in the probability of occurrence estimated differently priorities, so the S_1 scenario goes to the seventh position of priority according to the likelihood of occurrence, although it have higher complexity. In the results S_4 scenario has a greater probability of risk, in this case the use of this navigation path affects its risk probability.

4.2. Risk Probabilty Process on SOGU System

The process evaluation was conducted by testing tool Functional Tester Runner [15]. The Coverage was established on the basis of OPG the results obtained are in the Table 2. In this table are reported general and particular data metrics, the Density Defects is general and particular are the defect density of components and their Risk Probability, test cycles is 11865 and virtual tester number is 2523, in the tests results, the component C_3 has the biggest value on its defect density, the model has predicted this situation through the risk probability on the component, because it has the biggest value on this likelihood. The set of the components C_4 , C_2 , C_6 , C_5 and C_8 have values of density of defects in an acceptable range according to expected with the probability of risk, but its accuracy still needs to improve. The remaining components have low values in the defect density, which coincides with the expected values according to the predictive model, in this case there are also variants on the accuracy. An element that does not match the expected values of the model is the C_{10} component, it has a defect density that does not match the expected value, and its value is according to the second group. In the results we can observe how the risk probabilities established by the model give us a framework of uncertainty very closely to the values obtained during tests SOGU system on a real operational context. Broadly the values match with the results obtained in the tests. Although there are variations in the accuracy of the probability of risk for the components in relation to the distribution of risk.

The OPG generation can be performed by anyone who understands software engineering concepts. The algorithm used to analyze the OPG is simple and efficient on generating navigation paths. The simulation process is more specialized but offers important advantages, like statistical parameters near to real context, another benefit is the reduced time to execute the simulations. In the third phase, determining the probabilities of the components was more expensive process, when growing the space variables is more

difficult to find the solutions for the algorithms we are using.

Table 1. Parameters for Scenarios on the SOGU System

Position	ID	PATH	Weight	Component Sequence	Ocu_Prob	RP _{Si}
1	S ₄	vs1, vs2, vop5, vd15, vo16	220	C1,C2,C3,C4,C5,C6	0.658	0.383
2	S ₉	vs1, vs2, vop3, vo7	130	C1,C2,C3,C4,C5,C6	0.658	0.361
3	S ₈	vs1, vs2, vop4, vd13, vo14k	140	C1,C2,C3,C4,C8,C9	0.255	0.169
4	S ₂	vs1, vs2, vop4, vd8, vd10, vo11	250	C1,C2,C3,C4,C5,C8,C10	0.189	0.156
5	S ₃	vs1, vs2, vop4, vd8, vo9	220	C1,C2,C3,C4,C5,C8,C10	0.189	0.148
6	S ₆	vs1, vs2, vop4, vd8, vo12	210	C1,C2,C3,C4,C5,C8,C11	0.124	0.114
7	S ₁	vs1, vs2, vop5, vd15, vd17, vo18	260	C1,C2,C3,C4,C5,C15,C16	0.035	0.082
8	S ₅	vs1,vs2,vop5, v15, v19	220	C1,C2,C3,C4,C5,C14	0.042	0.075
9	S ₇	vs1,vs2,vop5, vd20, vo21	150	C1,C2,C3,C4,C12,C3	0.05	0.062
k10	S ₁₁	vs1,vs2,vop6, vd24, vo25	110	C1,C2,C3,C4,C19,C20	0.066	0.06
11	S ₁₀	vs1,vs2,vop6, vd22, vo23	120	C1,C2,C3,C4,C17,C18	0.029	0.044

Table 2. Results of SOGU Evaluation on a Real Context

Test Number: 11865			Virtual testers:2523			General Density Defects:8635		
Component	Defect density	Risk probability RPCi	Component	Defect density	Risk probability RPCi	Component	Defect density	Risk probability RPCi
C ₃	2272	0.301	C ₄	1631	0.032	C ₂	1603	0.022
C ₆	1133	0.042	C ₅	455	0.077	C ₈	453	0.077
C ₇	226	0.002	C ₁₄	225	0.005	C ₁₃	224	0.006
C ₂₀	224	0.005	C ₁₁	223	0.008	C ₁₀	124	0.022

5. Future Work

Taking into account the results in the third phase of the process, it is pertinent to evaluate other mathematical techniques to solve the risk probability on components to improve the level of accuracy in the results of the probability of component risk. To solve this aspect is very important because it is the limiting factor for the proposal, with the adequate technique this process can be applied to large-scale systems. Another important point is to make the study on a bigger system and to do a comparative analysis with other model with similar features.

6. Conclusions

In this proposal a process is performed to make Operational Profile Graph OPG from traceability approach. Through graph is applied a deterministic analysis based on software metrics. The analysis is also base to make a statistic analysis over use expected on Web application. By means of the relationship on both perspectives the risk probability of navigation paths is estimated and another important parameter which is the risk probability of components to be determined. These parameter are too important to set critical paths and to establish test coverage on testing phase. Risk Probability is a relevant aspect in the reuse approach, due to the fact that to have the probability of component risk is a support to determine

whether it is appropriate to reuse a specific component on a future version of the system or this in a new system.

By mean proposal an uncomplicated and efficient model is obtained, whose entries do not have a significant complication and the results are easy to interpret. The experimental results show that the model is efficient to find critical sequences and set the risk probability on the components in systems of medium scale.

References

- [1] Onoma A. K., Tsai W., Poonawala M., & Suganuma H. (1998). Regression testing in and industrial environment. *Commun. ACM*, 41(5), 81-86.
- [2] Rothermel, G., Harrold, M. J., Ronne J., & Hong, C. (2002). Empirical studies of test-suite reduction. *Softw. Test. Verif. Reliable*, 12(4), 219-249.
- [3] Rothermel, G., & Harrold, M. J. (1996). Analyzing regression test selection techniques. *IEEE Trans. Softw. Eng.*, 22(8), 529-551.
- [4] Rothermel, G., Untch, R. J., & Chu, C. (2001). Prioritizing test cases for regression testing. *IEEE Transactions on Software Engineering*, 27(10), 929-948.
- [5] Fenton, N. (2000). Predicting software defects in varying development lifecycles using Bayesian nets. *Information and Software Technology, El Servier*, 49(1), 32-43.
- [6] Kumar, D., & Mishra, K. (2016). The impacts of test automation on software's cost, quality and time to marke. *Proceedings of International Conference on Communication, Computing and Virtualization (ICCCV) 2016*.
- [7] Liping, L. I., Zhongsheng, O., & He, T. (2009). Test purpose-based test generation for web applications. *Proceedings of 1st International Conference on Networked Digital Technologies* (pp. 238-243).
- [8] Qi, Y., Kung, D., & Wong, E. (2005). An agent-based testing approach for web applications. *Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC'05)* (pp. 45-50).
- [9] Ricca, F., & Tonella, P. (2001). Analysis and testing of web applications. *Proceedings of the 23rd International Conference on Software Engineering* (pp. 25-34).
- [10] Sant, J., Souter, A., & Greenwald, L. (2005). *An exploration of statistical models of automated test case generation. International Workshop on Dynamic Analysis*.
- [11] Musa, J. (2004). *Software Reliability Engineering: More Reliable software Faster and Cheaper* (2nd ed.).
- [12] Kong, X., Liu, L., & Lowe, D. (2005). Web system trace model using a web application architecture framework, e-technology, e-commerce and e-service. *Proceedings of the 2005 IEEE International Conference* (pp. 508-513).
- [13] Bansal, P., & Sabharwal, S. (2013). A model based approach to test case generation for testing the navigation behavior of dynamic web applications. *Proceedings 6th International Conference on Contemporary Computing, IEEE 2013* (pp. 213-218).
- [14] Sprenkle, S., Pollocky, L., & Simko, L. (2011). A study of usage-based navigation models and generated abstract test cases for web applications. *Proceedings of the 4th IEEE International Conference on Software Testing, Verification and Validation* (pp. 230-239).
- [15] Davila-Nicanor, L., Benhumea-Peña, A., González-Morán, C. O., & Mejía-Álvarez, P. (2016). Functional tester runner tool. *Research in Computing Science*, 109, 69-79.



Leticia Dávila-Nicanor was born in Mexico city, Mexico, in 1973. She received the M.S and the Ph.D. in electrical engineering at Computer Science Department in 2008 by the

CINVESTAV-IPN. Currently, she works at Center University Mexico UAEM Valley as a full professor in the line of software engineering. She has conducted research and software development in the financial sector and telecommunications. Her work has focused on the quality of software from a formal point of view.