Volume 6, Issue 7, July 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Case Study-Factors Affecting Software Testing

Megha Chhabra*
Department of Computer Science and Engineering, Sharda University,
Greater Noida, Uttar Pradesh, India

Abstract— Automated processes of a business are substantially dependent on productivity and reliability of software program used. Software productivity is traceable to software testing. Factors affecting software testing can be broadly classified into three categories as time, cost and reliability. All the Direct or indirect factors are identified and placed in relation with all groups with centroids maintained as reliability, cost and time. The membership of allocation of factors to these three groups make between-class variance maximum and with-in class variance minimum. The purpose of entire classification is to maintain concrete identification of which factors can provide enhancement in a group of one kind while keeping others constant. The discretization of effort to enhance reliability maximizes the focus on factors belonging to group one whereas others are not gleaned upon hence saves time and budget and increases productivity in considerate area.

Keywords— Testing, Membership, Reliability, Classification.

I. INTRODUCTION

The optimal reliability is a term achieved to develop a software within the constraints of time and budget yet reliable enough to be driven by correctness of output. The driver entity behind a successful software is to expect the correct output everytime it runs [1]. In order to achieve the same, software needs a rigorous testing and verification of the output so far. Testing phase is a crucial step as it is dependent on various factors or objects which may be directly and/or indirectly affecting its pace or quality. These factors can be since beginning of idea or a client's budget allocation constraints or may be testing tools available for the problem in hand etc.

Reliability, time and cost are the three broad categories which are presently considered as constraints and mandates in every phase of software development life cycle (SDLC). Hence, these three mentioned factors are treated as the direct factors which are affecting testing of the software.

Any other aspect of SDLC can be seem to affect testing procedure indirectly. The effect of all other non-direct factors are in relation with reliability and/or time and/or cost involved in software production [2].

There exist many factors in literature which showcase a definite proportion of action resulting in change or improvement in testing procedure of software.

Xuemei Zhang et.al [3] clearly states the potential factors which may impact the software reliability by focusing on primary stakeholders. Statistical measures are used to analyse the variance among all and find the key features which may affect the reliability.

Nasib S. Gill [4] address the ground theory that software quality assurance (SQA) needs to be addressed keeping in view the new strategies, tool, methodologies and techniques applicable to software development life cycle. The author also states that ISO approach applicable to software quality management, testing improvement mechanisms are starters for the SQA maintainability.

O. Taipale et.al [5] presents another guide book for drifting testing schedule as a key ingredient in completion of software development and addressing the deficiency of standards for the time constraints in testing phase.

Paul Clarke et.al [6] clusters the factors of an optimal software reliability into situational factors like team size, nature of project, more or less, and presence of situational framework. The need of the reference framework is suggested as key constraint and way out to reduce the failure rate of software development.

Neil F et.al [7] projects the need of quality development rather than a timely delivery concern of software. His study showcases that generating revenue from IT, how a subset of traditional systems success factors can be improved. The estimate level has varied to a lot in last 30 years and now seems to be stabilised due to keen interest in factors to forecast the revenue generating factors for specific projects. In addition to it, Juha Itkonen et.al [8] presently seconds the notion by considering quality of testing as an integral part of developing relatively a better revenue generating system than an unpredictable system. Just mechanical exercise dependent on test cases is not justified. Some factor of testers experience and expertise adds up the reliability factor. Tanjila Kanij et.al [9] seconds the opinion by concluding that domain knowledge, experience and intelligence of a software tester are equally appreciated while considered testing tools and training. As such, while systematic methodologies are important, the individual most clearly does matter in software testing. Jiajun Xu et.al [10] furthers the research by comparing various types of software and their debugging model on the basis of environmental uncertainty and thereby predicting how the default rate.

© 2016, IJARCSSE All Rights Reserved

II. FACTORS AND THEIR DEPENDENCY

Starting from system development initiatives to quality management, present study classifies the factors into three categories. These clusters are centroid at reliability, time and cost involved in software. All the factors are allocated to each cluster accounting how much these are affected by their respective centroids or how much these factors affect their centroids. The reference goes both ways. Table 1 represents list of all factors and its direct or indirect effect on software testing. Compare and imagine the affect if these are left open and are not associated to any centroid. Since the actionable factors for keeping balance between time and budget yet maintaining reliability are interdependent, making change in a subset of factors will affect the rest of the factors as well. Whereas, if set of connected or closely dependent factors are bound to certain boundaries based on time constraints, cost effectiveness and reliability maintenance, the actionable factors to alter time constraints will not alter factors from other clusters or boundaries.

Table I list of factors affecting software testing and quality enhancement

Sno	Factors	Direct/ Indirect Affect	Sno	Factors	Direct/ Indirect Affect
1	Accuracy of cost estimation	Indirect	20	Overall testing strategy	Indirect
2	Accuracy of time estimation	Indirect	21	Place/Environment	Indirect
3	Architecture design	Indirect	22	Product oriented purpose	Indirect
4	Availability of resources	Indirect	23	Project complexity	Indirect
5	Budget revisions	Indirect	24	proportion of reused code	Indirect
6	Cost of pre-study	Indirect	25	Quality of delivery	Indirect
7	Deadline revision	Indirect	26	Reliability	Direct(Centroid1)
8	Development platform	Indirect	27	Requirement analysis	Indirect
9	Development team size	Indirect	28	Risk classification	Indirect
10	Documentation	Indirect	29	Skills of others	Indirect
11	Expenditure on external units	Indirect	30	Skills of Quality assurance team	Indirect
12	Frequency of specification changes	Indirect	31	Skills of technical team	Indirect
13	Implementation efficiency	Indirect	32	Spec justification	Indirect
14	Integration testing effort	Indirect	33	Sponsor/ Client	Indirect
15	Length of pre-study	Indirect	34	Testing Environment	Indirect
16	Liable for delivery	Indirect	35	Testing strategy	Indirect
17	Overall Cost forecast	Direct(Centroid3)	36	Time	Direct(Centroid2)
18	Overall schedule	Indirect	37	Training cost	Indirect
19	Overall testing plan	Indirect	38	Training time	Indirect

The factors have their existence since beginning. The membership function of each is defined and updated by the closeness level of the factors to their centroid. Lesser the variation in the cluster, more stable or discrete the membership of the factor to their respective cluster. If variation tend to increase with-in cluster, the boundaries tend to weaken and a factor then can belong to more centroids. The belongingness to multiple clusters tends to create an unstable membership value which needs to be updated for every change in interlinked factors.

III. FACTOR ALLOCATION

A factor can be ideally responsible to enhance or control one centroid i.e either reliability, time or cost, effectively. But the realization of membership or level of participation of any factor to any cluster of any representative centroid is achieved by minimizing the with-in class variance of factors of one cluster and maximizing the between class variance so that the overall variation in values of participation of factors should be stabilized. Table2 represents the respective allocation of factors to clusters. There are, in general, a few of these factors are members of multiple clusters.

Table 2 factors allocated to their respective clusters

Cluster	Single Cluster_membership			Multiple Cluster_membership		
	Cluster1_Reliability	Cluster2_Time	Cluster3_Cost	Cost/ Reliability	Cost/ Time	Cost/ Time/ Reliability
Factors	Architecture design	Accuracy of time estimation	Accuracy of cost estimation	Availability of resources	Length of pre-study	Documentation
	Development	Deadline	Budget	Frequency of	Requiremen	proportion of

© 2016, IJARCSSE All Rights Reserved

	platform	revision	revisions	specification changes	t analysis	reused code
	Liable for delivery	Overall schedule	Cost of pre-study	Place/environment		Risk classification
	Overall testing plan	Testing strategy	Development team size	Project complexity		Product oriented purpose
	Overall testing strategy	Training time	Expenditure on external units	Skills of Quality assurance team		
	Quality of delivery		Implementation efficiency	Skills of technical team		
	Specification justification		Integration testing effort	Skills of others		
	Sponsor/ client		Overall Cost forecast			
	Testing Environment		Training cost			

The interrelation between factors is ideally assumed to be zero, the moment membership function achieves stability. The general scenario observed never finds these factors to be independent of each other. The correlation between factors belonging to cluster 1 i.e. reliability and factors of cluster2 i.e. Time is show in Table3 and similar correlation between factors belonging to cluster 1 i.e. reliability and factors of cluster3 i.e. Cost is show in Table4. A cumulative representation can be seen in fig1.

Fig. 1 Interdependent factors for three clusters

IV. CONCLUSIONS

Three clustered formed are used to create boundaries around all the actionable factors which may directly or indirectly affect the quality of testing and thereby enhancing the software development process. Factors identified and allocated to single or multiple clusters decide its potential to variation when any of the representative centroid of cluster is tend to change for betterment of the testing process. The membership of allocation of factors to these three groups make between-class variance maximum and with-in class variance minimum. The advantage of the clustering is to find out which factor(s) demands relatively more attention while keeping other factor(s) at a stable condition. This not only saves time and budget but also gives a focused attention to precise level of reliability where exactly required. In addition to it, the same can benefit in realization of real time presence or absence or shortage of any factor listed as a guide or reference for software testing.

REFERENCES

[1] Beizer, Boris, Software testing techniques (2nd ed.) New York, NY, USA: Van Nostrand Reinhold Co., 1990.

[2] M. Lyu, Handbook of Software Reliability Engineering, McGraw-Hill, 1996.

[3] X. Zhang, H. Pham, *An analysis of factors affecting software reliability*, Journal of Systems and Software, Volume 50, Issue 1, January 2000, Pages 43–56.

[4] N. Gill , *Factors affecting effective software quality management revisited*, ACM SIGSOFT Software Engineering Notes, Volume 30 Issue 2, March 2005, Pages 1-4.

[5] O. Taipale, K. Smolander ; H. Kalviainen, *Factors affecting software testing time schedule*, Australian Software Engineering Conference (ASWEC'06), 18-21 April 2006, ISSN: 1530-0803.

[6] P. Clarkea, R. V. O'Connor, *The situational factors that affect the software development process: Towards a comprehensive reference framework*, International journal of Information and Software Technology, Volume 54, Issue 5, May 2012, Pages 433–447.

[7] Neil F Doherty , C. Ashurst, J. Peppard, *Factors affecting the successful realisation of benefits from systems development projects: findings from three case studies*, Journal of Information Technology, March 2012, Volume 27, Issue 1, pp 1-16.

[8] J. Itkonen, M. Mäntylä, C. Lassenius, *The Role of the Tester's Knowledge in Exploratory Software Testing*, IEEE Transactions on Software Engineering (Volume:39 , Issue: 5), 707 – 724, ISSN: 0098-5589.

[9] T. Kanij, R. Merkel, J. Grundy, *A Preliminary Survey of Factors Affecting Software Testers*, Australasian conference on software engineering (ASWEC2014), Sydney, Australia, April 2014.

[10] J. Xu, S. Yao, *Software Reliability Growth Model with Partial Differential Equation for Various Debugging Processes*, Mathematical Problems in Engineering, Volume 2016 (2016), Article ID 2476584, 13 pages.

Table 3 Correlation Between Reliability and Time Factors

Correlation between Reliability and Time factor		**Time**				
		Accuracy of time estimation	**Deadline revision**	**Overall schedule**	**Testing strategy**	**Training time**
Reliability	**Architecture design**	+	-	-	+	-
	Development platform	+	-	-	+	+
	Liable for delivery	+	+	-	+	+
	Overall testing plan	+	-	-	+	-
	Overall testing strategy	+	-	-	-	-
	Quality of delivery	+	-	+	+	+
	Specification justification	+	-	+	+	
	Sponsor/ client	+	-	-	+	
	Testing Environment	+	-	-	+	+

Table 4 Correlation between reliability and cost factors

Correlation between Reliability and cost factor		**Cost**								
		Accuracy of cost estimation	**Budget revisions**	**Cost of pre-study**	**Development team size**	**Expenditure on external units**	**Implementation efficiency**	**Integration testing effort**	**Overall Cost forecast**	**Training cost**
Reliability	**Architecture design**	+	-	-		+	+	-	-	
	Development platform	+	-	-	+	+	+	+	+	+
	Liable for delivery	+	+	+	+	+	-	+	+	+

	Overall testing plan	+	-	-	+	+	+	+	+	-
	Overall testing	+	-	-	+	+	+	+	+	-
	Quality of delivery	+	-	+	+	+	+	+	+	+
	Specification justification		-	-	+	+	+	+	+	+
	Sponsor/ client	+	-	-			+		-	
	Testing Environment	+	-	+		+	+	+	-	-

© 2016, IJARCSSE All Rights Reserved