American Journal of Software Engineering, 2013, Vol. 1, No. 1, 1-4
Available online at http://pubs.sciepub.com/ajse/1/1/1
© Science and Education Publishing
DOI:10.12691/ajse-1-1-1

Testing of Object Oriented Software: A Study to Identify the Factors

Sanjeev Patwa[1,*], Anil Kumar Malviya[2]

[1]Faculty of Arts, Science and Commerce, Mody Institute of Technology & Science,(Deemed University), Lakshmangarh, Sikar Rajasthan, India
[2]Department of Computer Science & Engineering, KNIT, Sultanpur, U.P., India
*Corresponding author: sanjeevpatwa@yahoo.com

Received August 16, 2013; Revised September 09, 2013; Accepted September 12, 2013

Abstract In recent years, there has been a surge of interest in Objects Oriented (OO) methodology in the development of software. However there is a dilemma as to how best fit the OO culture with existing structured approach in testing of software. The present research study is a survey on the factors which affect the testing of Object Oriented systems. These factors were identified based on existing literature. A survey was conducted among professionals, from several cities across the India. We analyze all the factors to identify their significance on testing techniques in OO software.

Keywords: *object oriented software, factors, testing*

Cite This Article: Sanjeev Patwa, and Anil Kumar Malviya, "Testing of Object Oriented Software: A Study to Identify the Factors." *American Journal of Software Engineering* 1, no. 1 (2013): 1-4. doi: 10.12691/ajse-1-1-1.

1. Introduction

When tester adopts testing approaches for object-oriented software, they cannot ignore several factors which may affect testing techniques. An object oriented design approach mainly affects detailed design and code, but planning, requirements analysis, architectural design, deployment and maintenance are largely independent in the use of a specific design approach [1]. An Object oriented approach curtails development time, also has greater clarity of expression [2], makes software "resist both accidental and malicious corruption attempts", more understandable and is more maintainable [3]. The testing of object oriented software depends on many factors: Complexity in Logic, Program Categories, Difficulty of Programming Language, Amount of Programming Effort, Level of Programming Technologies and Percentage of Reused Modules. These factors have been identified based on existing literature and by the opinion of respondents who are involved in Questionnaire. Object Oriented development will experience 'different types and proportions of errors that require a different approach to testing' compared to conventional development methodologies and languages [4]. Ilene Burnstein described in his book [5] that testing principles are important to test specialists/engineers because they provide the foundation for developing testing knowledge and acquiring testing skills. Testing as a component of the software engineering discipline also has a specific set of principles that serve as guidelines for the tester.

Software testers are often specially trained, published information at conferences, as well as forums and blogs, and a number of research papers on the Internet. This paper makes an attempt to realize that to what extent software professionals* believe that the factors are significant for OO software testing.

The ultimate goal of software testing is to help designers, developers, and managers to construct systems with high quality. Thus, research and development on testing purports for efficiently and effectively perform the testing in order to find more errors in various phases as: requirement, design and implementation of the software.

The paper is organized as follows: Section 2 presents Literature Review; Section 3 discusses about Questionnaire as Methodology, categorization of respondents and selected factors. Section 4 describes the Hypothesis and presents the analysis of survey data with statistical tool and finally, Section 5 concludes our work with its significance.

2. Literature Review

The first publication in object-oriented software testing with formal analysis was presented by Perry and Kaiser [6] with adequacy criteria for object-oriented software testing. Smith and Robson [7] discussed about issues in object-oriented software testing in several points of view. Although the issues are similar to ones in [8], they are discussed in different ways. Japanese researchers Furuyama et al. [9,10] studied factors such as working stress, development methodologies, etc. using design of experiment methods. They found that different settings of these factors have statistically significant impact on the quality of final software products. Previous research on new developments has identified a large number of factors which may have an impact on testing and software reliability [11]. Patwa and Malviya [15] proposed a metrics, Reusability of a class in a System (RCS) which identifies that Testing Effort of class Ci ∝ 1/number of successors of class Ci. McGregor and Korson [12] discuss a high-level view of testing OO systems within the entire software development cycle.

3. The Methodology

This study was exploratory in nature yet specific in view of the conceptual models. Field examination through Questionnaire and study was chosen as the ideal overall design approach.

Questionnaire is divided in two parts. Part A comprises factors on which we want to examine the perceptions of respondents with regard to involving of these factors in Object Oriented software testing. The survey used a 5 point Likert scale to identify the degree to which each factor (the independent variables) has a significant influence on testing. In the survey form, "1" indicates "not significant" and "5" stands for "most significant". If these factors are irrelevant, score of "1" would be expected; if they have significant impacts on selecting software testing techniques, the average score would be close to "5".

The second part (Part B) of the questionnaire relates to the personal and organizational professional data to explore some background data of the survey participants and the categories of the software application they are using.

3.1. Categorization of Respondents

One of the key tasks for designing the realistic representation was the selection of the respondents. The characteristics of the people involved in the construction of the realistic representation can have a significant influence on the resulting model. The people involved were as heterogeneous as possible to assure that the representation does not reflect a unilateral viewpoint. The type of respondents for this study is very busy people (programmers, testers, and managers) who have hardly any time for this sort of activities. Even so, 168 respondents from a range of fields, with varying experience and of different places of the country have been explored for the research. Testing is the activity which is involved in each phase of software development. There are various roles for testing team members. Information derived from software testing may be used to correct the process by which software is developed [13].

The set of participant's profile was as heterogeneous as possible, we looked for people who engage in different phases of software development and perform the role as tester. As manager examines the overall design and customer satisfaction, programmer examines the code by own and the tester tests the whole system and its output with different test cases.

Input data is collected using a formal survey questionnaire given directly to the programmer, tester or manager in various organizations. They have a relatively good mixture of software development and testing experience, on different program categories. Demographic data on the participants are summarized in Table 1 which is based on part B of questionnaire.

Table 1. Demographic Data of Survey Participants

Personal/Demographic factor	Mean Score	Sample Size
1. Current job position (distribution of the survey participants)	Project Manager: 13.00%	168
	Programmer: 63.00%	
	Tester: 22.00%	
	Others: 2.00%	
2. Number of experience (years)	4.67=5 Years	156
3. % of participants who agreed that factors impact on testing	82.73%	168

It is evident from Table 1 that in software industry number of programmer is higher than the tester or manager (Programmer 63% and tester 22%, manager 13%), and approximate 83% (139 out of 168) of respondents agreed that selected factors affect the Object Oriented Software testing.

Section A of questionnaire contains some factors which are chosen on the base of literature of software testing. The descriptions of these factors are as follows:

F1. Complexity in logic: Program size (Kiloline of code: KLOC) is used as a measure of program complexity. As in OO software, complexity is depend on number of classes, their inheritance level, coupling of classes, number of loops and decision statements. The "High" level of these feature means the program size is greater and program is complex.

F2. Program Categories: There may be five program categories, which may impact in selecting the testing techniques: operating system, communication control program, database management system, web application, and languages processor.

F3. Difficulty of Programming Language: As difficulty level of OO programming languages and conventional programming languages are differ, we need to have a solid understanding of Memory management and OOP to use them effectively, and thus this may be a factor, which affects the testing.

F4. Amount of Programming Effort: The deliberate programming effort may be regarded as effective for reducing the number of errors made. This is calculated in man-years.

F5. Level of Programming Technologies: The programming technologies are classified into four categories: design and documentation techniques (DFD, UML, STD, Flow Charts, and Algorithms etc.), programming techniques (including programming languages) and development of computer access environment.

F6. Percentage of Reused modules: When people develop some new software products or when they update the old version of their software products, they usually keep some of the module of the code which can be reused, and added in some new ones.

4. Hypothesis and Analysis

Field examination through survey and study was chosen as the optimum overall approach. Zhang and Pham [11] used statistical test for analyzing the survey data on factors affecting software reliability, earlier Kanij et.al [14] used HSD test for analysis of survey data. In this paper we wish to determine that selected factors significantly affect the testing or not. We have the following hypothesis:

H_{01}: Complexity in logic, Program Categories, Difficulty of Programming Language, Amount of Programming Effort, Level of Programming Technologies and Percentage of Reused modules significantly affects the testing techniques.

We thus performed a one-way analysis of variance (ANOVA) with all the factors to test the Null Hypothesis. In ANOVA test if p-value is < 0.05, the chance that this is the case is under 5%. To permit a decision between the null hypothesis and the alternative hypothesis, significance limits are often specified in advance. The level of significance of 0.05 (or 5%) is often chosen. If the p-value is less than this limit, the result is significant and it is agreed that the null hypothesis should be rejected and the alternative hypothesis accepted.

Table 2. ANOVA for significance of factor F1

		Sum of Squares	df	Mean Square	F	Sig.
F1	Between Groups	12.589	2	6.295	3.17	0.045
	Within Groups	327.69	165	1.986		
	Total	340.28	167			

Table 2 shows that p-value for Complexity in logic is 0.045 which is less than 0.05; similarly F- value at 5% level of significance comes out to be 3.17 which are more than the tabulated value. So it can be expressed that the testing hypothesis has been rejected for this factor.

Table 3. ANOVA for significance of factor F2

		Sum of Squares	df	Mean Square	F	Sig.
F2	Between Groups	20.448	2	10.224	7.604	0.001
	Within Groups	221.838	165	1.344		
	Total	242.286	167			

Table 3 shows that p-value for Program Categories is 0.001 which is less than 0.05, Similarly F- value at 5% level of significance comes out to be 7.604 which is more than the tabulated value. So it can be expressed that the testing hypothesis has been rejected for this factor.

Table 4. ANOVA for significance of factor F3

		Sum of Squares	df	Mean Square	F	Sig.
F3	Between Groups	18.482	2	9.241	4.785	0.01
	Within Groups	318.637	165	1.931		
	Total	337.119	167			

Table 4 shows that p-value for Difficulty of Programming Language, is 0.01 which is less than 0.05, Similarly F- value at 5% level of significance comes out to be 4.78 which is more than the tabulated value. So it can be expressed that the testing hypothesis has been rejected for this factor.

Table 5.ANOVA for significance of factor F4

		Sum of Squares	df	Mean Square	F	Sig.
F4	Between Groups	16.086	2	8.043	3.628	0.029
	Within Groups	365.819	165	2.217		
	Total	381.905	167			

Table 5 shows that p-value for Amount of Programming Effort, is 0.029 which is less than 0.05,Similarly F- value at 5% level of significance comes out to be 3.628 which is more than the tabulated value. So it can be expressed that the testing hypothesis has been rejected for this factor.

Table 6.ANOVA for significance of factor F5

		Sum of Squares	df	Mean Square	F	Sig.
F5	Between Groups	7.058	2	3.529	1.885	0.155
	Within Groups	308.889	165	1.872		
	Total	315.946	167			

Table 6 shows that p-value for Level of Programming Technologies, is 0.155 which is more than 0.05, similarly the F-value (1.885) was much lower than the tabulated value at same degree of freedom. So the testing hypothesis has not been rejected for this factor.

Table 7. ANOVA for significance of factor F6

		Sum of Squares	df	Mean Square	F	Sig.
F6	Between Groups	0.669	2	0.335	0.213	0.809
	Within Groups	259.706	165	1.574		
	Total	260.375	167			

Table 7 shows that p-value for Percentage of Reused modules, is 0.809 which is more than 0.05, similarly the F-value (0.213) was much lower than the tabulated value at same degree of freedom. So the testing hypothesis has not been rejected for this factor.

5. Conclusion

Thus it has been concluded from above tables and descriptions that Level of Programming Technologies and Percentage of Reused Modules are two factors which significantly affect the testing in Object Oriented Software. These results are uniform to the metrics, Reusability of a class in a System (RCS) which identifies that Testing Effort of class Ci ∝ 1/number of successors of class Ci, proposed by Patwa and Malviya [15]. While Complexity in Logic, Program Categories, Difficulty of Programming Language and Amount of Programming Effort has rarely effect on the testing of OO software. The results are based on the group of people hosted in this survey. Care should to be taken when applying these results in other applications. The result may vary for different projects or applications.

References

[1] M. Pezz`e, M. Young.In Proceedings of the 26th International Conference on Software Engineering (ICSE'04), IEEE Computer Society, 2004.

[2] F.Brooks. No silver bullet: Essence and Accidents of software Engineering. Information Proceeding of Elsevier Science Publishers, 1986.

[3] G.Booch. Object Oriented Development. IEEE, 1986.

[4] D.G.Firesmith. Testing object-oriented software. Technical Report, U.S.A., Advanced Technology Specialists, 1992.

[5] I. Burnstein. Practical software testing: a process-oriented approach. Springer-Verlag New York, Inc, 2003.

[6] D. E. Perry, G. E. Kaiser. Adequacy Testing and Object-Oriented Programming. Journal of Object Oriented Programming, January/February 1990.

[7] M. D. Smith and D. J. Robson. Object-Oriented Programming the Problems of Validation. In Proceedings of Conference on Software Maintenance, San Diego, CA USA, pp. 272-281, November 1990.

[8] W. E. Howden. Reliability of the Path Analysis Testing Strategy. IEEE Transactions on Software Testing, pp. 208-215, September 1976.

[9] T. Furuyama. Fault generation model and mental stress effect analysis. Journal of Systems and Software, 26, 31-42, 1994.

[10] T. Furuyama, Y.Arai, K. Lio. Analysis of fault generation caused by stress during software development. Journal of Systems and Software, 38, 13-25, 1997.

[11] X. Zhang, H. Pham. An analysis of factors affecting software Reliability. Journal of Systems and Software, Elsevier, 50:43-56, 2000.

[12] J. McGregor and T. Korson. Integrated Object-Oriented Testing and Development Processes. Communications of the ACM, pp. 59-77, September 1994,

[13] A.Kolawa,D.Huizinga. Automated Defect Prevention: Best Practices in Software Management. Wiley-IEEE Computer Society Press. pp. 41-43, 2007.

[14] T. Kanij, R. Merkel, and J. Grundy. A preliminary study on factors affecting software testing team performance. In International Symposium on Empirical Software Engineering and Measurement,pp 350-362, IEEE Computer Society, 2011.

[15] S.Patwa, A.K. Malviya. Reusability Metrics and Effect of Reusability on Testing of Object Oriented Systems. ACM SIGSOFT Software Engineering Notes, 37(5), September 2012.