IMPACT OF CODING PHASE ON OBJECT ORIENTED SOFTWARE TESTING

Sanjeev Patwa[1]

&

Anil Kumar Malviya[2]

[1]FASC, Mody University of Science and Technology, Lakshmangarh, Sikar, Rajasthan,

[2]KNIT, Govt. Engg. College, Sultanpur, U.P.

[1]sanjeevpatwa@yahoo.com, [2]anilkmalviya@yahoo.com

Abstract: The paper demonstrate the findings of empirical research from 23 software development companies to identify the factors of coding phase which effects the testing of Object Oriented (OO) software. Six potential factors of coding phase have been identified. The study uses a relative weight method and ANOVA test to analyse these factors and identify the place of these factors according to their impact on Object Oriented software testing. The survey focuses on the crucial participants like programmers and testers who highly involve in coding phase.

Keywords: ANOVA, Factors of Testing, Object Oriented Software, Relative weight method.

I. Introduction

The software development process consists of five phases: analysis, design, coding, testing, and operation (Lee et.al., 1999).According to (Beizer, 1990), testing is considered as one of the most costly development processes, sometimes exceeding fifty per cent of total development costs. In Object-Oriented systems the majority of development is concerned with analysis, design, and coding phases. Each phase of development has its own importance in testing. Testing is a continuous process involves in each phase of software development but software testing has not kept pace with the advances in OO system programming. Some OO systems do not conform to traditional testing definitions or techniques. There are several factors in each phase which affect the software testing. In this paper we are considering only coding phase and related factors.

Although the question of what are the factors which affect the testing techniques is a question which testers face every time they have to test a system. In general there are several testing techniques are exists, but it is unfortunate that some are never considered for use, and others are used over again in different projects without even observing, whether or not they were really useful. There are basically two reasons why developers do not make good choices:

- The information about testing techniques are distributed across academic tools like journals, books, articles etc. and technical people like developers and testers rarely read or study the literature.
- They have fewer tendencies to share the technical experience with others of using testing techniques in different projects. This means that they miss out the chance of learning about the experiences of others.

The problem we emphasis here is to identify relevant factors and their places with coding phase, which affect the testing techniques in OO software and to find the opinion of programmer and tester on significance of selected factors for, object oriented software testing. The aim of solving this problem is to help testers to find factors which affect the testing of project in coding phase and select the suitable testing technique

II. Related Work

The problem of identifying and analyzing the factors which affect the testing in coding phase has not yet been specifically studied in the testing area, there have been some attempts which focuses on comparing the testing techniques. There are not solutions for the problem on which we are emphasizing here, although the knowledge they input can be used in analyzing the problem that we propose.

A main problem with testing object-oriented systems is that standard testing methodologies may not be useful. Smith and Robson (Smith & Robson, 1994) say that current IEEE testing definitions and guidelines cannot be applied blindly to OO testing. Harold (Harold et al., 1992) present a technique that takes advantage of the hierarchical nature of classes, utilizing information from the super classes to test related groups of classes. Parrish (Parrish et al., 1993) present a technique for testing OO systems that is based entirely on generating test cases from the class implementation. McGregor (McGregor and Korson, 1994) discussed a high-level view of testing OO systems within the entire software development cycle. Research on software process innovation found that factors, such as organization size, technological diversity, professionalism, and IS slack are related to the adoption process (Grover et.al, 1997). Characteristics of a software organization, such as IT infrastructure, staff, managers, and control systems, have an effect on identifying, adopting, testing and implementing an innovation (Tornatzky & Fleischer, 1990).These attributes and organizational characteristics comprise the basis for our research model and hypotheses.

In previous some years, many software testing techniques has been proposed for achieving software reliability. However in early stages of software development mostly all the factors which affect the testing are ignored for simplicity reasons. But in order to improve the understanding and simulation of a complex process such as the software coding process, factors like Programmer/Tester skill, Programmer/Tester organization (the percentage of high-quality programmers), Development team size, Program workload (stress), Domain knowledge, Human nature (mistake and work omission) need to be considered and incorporated into selecting the software testing techniques.

To our knowledge very less research has been conducted in the area of finding the factors and analyse their impact on testing of coding phase, and how people consider these factors in enhancing the software testing process.

III. Research Goals

In this study, we performed a survey and obtained opinion from programmers and testers who participated in the software development. In this research we identified six factors which involve in the coding phase of the software development process.

The paper emphasis on the hypothesis that:

H0: All the selected factors have significant impact on testing of object oriented software in coding phase.

In hypothesis test the confidence-level is consider at 95% (at 5% level of significance) that means if the value of p is less than 0.05 ($p < 0.05$) we reject the null hypothesis.

Software development is a very complex process which involves several factors. These factors vary from project to project. In our modern society, object oriented software has become a very critical component in all kinds of systems and software.

The factors considered in this study are Programmer/Tester Skill, Programmer/Tester Organization (the percentage of high-quality programmers), Development Team Size, Program Workload (stress), Domain Knowledge and Human Nature (mistake and work omission).Some of these factors also considered by Zhang (Zhang & Pham, 2000) for measurement of software reliability. The description of factors is as follows:

1. Programmer/Tester Skill: Skill can be defined as the average number of years of programming experience of programmers. This can be calculated as the ratio between total year of experience of all programmers/tester and total no of programmers/testers in organization.
2. Programmer/Tester Organization: Programmer

/Tester organization (PO) is defined as the percentage of high-quality programmers. PO is computed as follows: PO = n_h/n ; where n_h is the number of programmers, whose programming experience is more than six years, and n be the total number of programmers in organization. As PO is high, we can select better testing technique [ix].

3. Development Team Size: This factor identifies that quality of project would improve if the size of team will be large or the quality will improve with the less but experienced development team.
4. Program Workload: During the software development, stress factors in terms of "work contents" such as schedule pressure and too much work are the major factors. This factor may affects in selecting the Testing technique.
5. Domain Knowledge: Domain knowledge refers to the programmer's and tester's knowledge of the input space and output result. Insufficient knowledge may cause problems in coding and testing procedures.
6. Human Nature: This refers to the tester and developers characteristics, including the ability to avoid the making of working mistakes, careless work omission and selecting the testing technique.

The results of this study may be utilized in selecting the testing techniques in coding phase by incorporating significant factors. This study aims to present the rank of factors according to their significance in OO software testing.

IV. Statistical Methodology

Software testing in coding phase, in the context of analysing factors is a new area of research; this study is exploratory in nature yet specific in view of the conceptual models. Field examination through questionnaire and study was chosen as the overall design approach. The factors are described in table I.

Table I. Factors of Coding Phase

Coding
Programmer/Tester Skill
Programmer/Tester Organization
Development Team Size
Program Workload (stress)
Domain Knowledge
Human Nature (mistake and work omission)

We utilized Relative weight method to analyze the ranking of factors and for hypothesis testing parametric test ANOVA (Analysis of Variance) has been used.
Places and determining the relative weights of factors based solely on the participants opinions as reflected in the questionnaire. Under this methodology, we treat every single participant equally without considering his/her background information.

V. Data Collection

The questionnaire focused on factors which affect the testing in coding phase and try to examine the view of participants.

Data were collected using a formal survey questionnaire given directly to the software developers/programmers and testers in 23 Indian software development and testing organizations including focusqa.com, pure testing software Pvt. Ltd., TCS, Metacube Systems, etc. Demographic data on the participants are summarized in Table II.
Questionnaire used a 5-point scale to identify the degree to which each factor (the independent variables) has significant influence on software testing in coding phase. In the questionnaire, "1" indicates "not significant" and "5" stands for "most significant".

Table II. Demographic data of survey participants

Personal/Demographic factor	Mean score	Sample size
1. Current job Position (distribution of the survey participants)	Programmer: 72.61% (122) Tester: 27.38% (46)	168
2. Number of experience (years)	4.67=5 Years	156
3. % of people agreed on significance of factors	82.85%	168

VI. Hypothesis Test and Analysis

A. Relative weight method

First, the relative weight method was used to obtain the final places for the factors. Let rij be the original ranking of the i th factor on the j th survey. We first normalize these rij's such that

$$wij = \frac{rij}{\sum_{i=1}^{n} rij} \qquad (1)$$

Where n is the number of factors on the jth survey. Therefore $\sum_{i=1}^{n} wij$ =1 for all j.
Different people may give different original ranking and some of them may give higher scores for all factors. By normalizing the original

ranking scores using Eq. (1), one can get rid of this bias. We then average these wij's to obtain the final weight for the ith factor such that

$$wi* = \frac{\sum_{j=1}^{l} wij}{l} \qquad (2)$$

Where l is the number of surveys used in this method. Based on these relative weights, we could obtain the final weight for each factor.

From the results by the relative weight method places of factors is given in Table III. The column named Normalized Priorities gives the contribution of each factor. For example, Programmer/Tester Skill contributes approximately 4.4% (its relative weight. 0.0446987). Higher priority value indicates a higher place. Since lower class rank implies decrease in magnitude of relative importance, software programmer and tester should then pay more attention to the factors with high places. The final priority information can then be used to guide the Object Oriented software testing process in coding phase of different applications.

Table III. Final ranking based on relative weight method

Rank	Rank factor	Factor Name	Normalized Priorities
1	F1	Programmer/Tester Skill	0.0446987
2	F5	Domain Knowledge	0.0430985
3	F6	Human Nature (mistake and work omission)	0.0369283
4	F2	Programmer/Tester Organization	0.0348077
5	F4	Program Workload (stress)	0.0339011
6	F3	Development Team Size	0.0226598

From the demographic data in table II it is observed that 83% participants agreed on the significance of impact of selected factors on testing techniques. Table III indicates the top two factors of coding phase are Programmer/Tester Skill and Domain Knowledge.

B. ANOVA test

We conduct ANOVA test for each factor and analyse the statistics as, the testing hypothesis is accepted if the p-value is more than 0.05 at 95% level of confidence, otherwise the testing hypothesis is not accepted (rejected). Resultant tables and descriptions are as follows:

Programmer/Tester skill (F1):

Table IV ANOVA test for Programmer/Tester Skill

		Sum of Squares	df	Mean Square	F	Sig.
F1	Between Groups	7.723	2	3.861	2.675	0.072
	Within Groups	238.182	165	1.444		
	Total	245.905	167			

Table IV evident that the computed value of F-statistics is 2.675 which is less than tabular value of F statistics (3.00) therefore the Null hypothesis is accepted at 5% level of significance for this factor. Subsequently same result can be established with respect to p-value which is 0.075. Therefore, it is concluded that the Programmer/Tester Skill significantly affect the testing in coding phase.

Programmer/Tester organization (F2):

Table V –ANOVA test for Programmer/Tester Organization

		Sum of Squares	df	Mean Square	F	Sig.
F2	Between Groups	15.911	2	7.955	5.064	0.007
	Within Groups	259.208	165	1.571		
	Total	275.119	167			

It is evident from table V that the computed value of F-statistics is 5.064 which is more than tabular value of F statistics (3.00) therefore the hypothesis is rejected at 5% level of significance. Subsequently same result can be established with respect to p-value which is 0.007. Therefore, it is concluded that the Programmer/Tester Organization significantly affect the testing in coding phase.

Development Team Size (F3)

Table VI ANOVA test for Development Team Size

		Sum of Squares	df	Mean Square	F	Sig.
F3	Between	24.407	2	12.204	5.723	0.004

	Groups					
	Within Groups	351.872	165	2.133		
	Total	376.28	167			

Table VI evident that the computed value of F-statistics is 5.723 which is more than tabular value of F statistics (3.00) therefore the hypothesis is rejected at 5% level of significance. Subsequently same result can be established with respect to p-value which is 0.004.

Therefore, it is concluded that the Development Team Size less significantly affect the testing in coding phase.

Program workload (stress) (F4)

Table VII ANOVA test for Program Workload (Stress)

		Sum of Squares	df	Mean Square	F	Sig.
F4	Between Groups	22.628	2	11.314	10.182	0.00
	Within Groups	183.348	165	1.111		
	Total	205.976	167			

It is evident from table VII that computed value of F-statistics is 10.182 which is more than tabular value of F statistics (3.00) therefore the hypothesis is rejected at 5% level of significance. Subsequently same result can be established with respect to p-value which is 0.00.

Therefore, it is concluded that the Program Workload less significantly affect the testing in coding phase.

Domain Knowledge (F5)

Table VIII –ANOVA test for Domain Knowledge

		Sum of Squares	df	Mean Square	F	Sig.
F5	Between Groups	7.215	2	3.608	1.841	0.162
	Within Groups	323.261	165	1.959		
	Total	330.476	167			

Table VIII represent that the computed value of F-statistics is 1.841 which is less than tabular value of F statistics (3.00) therefore the hypothesis is accepted at 5% level of significance. The same result can be established with respect to p-value which is 0.162.

Therefore, it is concluded that the Domain Knowledge significantly affect the testing in coding phase

Human Nature (mistake and work omission) (F6)

Table IX ANOVA test for Human Nature

		Sum of Squares	df	Mean Square	F	Sig.
F6	Between Groups	22.203	2	11.101	6.929	0.001
	Within Groups	264.369	165	1.602		
	Total	286.571	167			

It is evident from table IX that the computed value of F-statistics is 6.929 which is more than tabular value of F statistics (3.00) therefore the hypothesis is rejected at 5% level of significance. Subsequently same result can be established with respect to p-value which is 0.001.

Therefore, it is concluded that the Human Nature less significantly affect the testing in coding phase.

From above discussion it is evident that hypothesis has been accepted for two factors out of six. The two factors Programmer/Tester Skill and Domain Knowledge also has highest ranking according to relative weight method in table III. Other four factors do not accept the null hypothesis means they are less significant for testing in coding phase this result also verified the ranking of factors indicated in table III.

VII. Conclusion and Future Work

This paper shares out with the factors involved in the testing of coding phase in Object Oriented Software development process. A study was performed to collect the data. The relative places of the factors have been provided in terms of the significance of their impact on software testing. Developers and testers can check the list and find out the most significant ones for their projects. From the opinion of our study participants, we can see that most of the people (82.85%) of the respondents agreed on the fact that selected factors influence on software testing.

However survey result shows that the factors, Programmer/Tester Skill (F1) and Domain Knowledge (F5) significantly affect the testing of Object Oriented software in coding phase while other factors which less significantly affect the testing are as follows: Programmer/Tester organization (F2),

Development Team Size (F3), Program workload (stress) (F4) and Human Nature (mistake and work omission) (F6).
The findings, however, are based on the group of people who participated in the questionnaire. Caution need to be taken when applying these results in other applications. This study provides a basis for many different directions for further research one of them is to introduce more factors in the questionnaire.

References

Beizer, B. (1990). Software Testing Techniques, Second Edition, Van Nostrand Reinhold Company Limited.

Grover, V., Fiedler, K. & Teng, J. (1997). Empirical evidence on Swanson's tri-core model of information systems innovation. Information Systems Research, 89(3), 273-287.

Harold, M., McGregor, J., & Fitzpatrick, K. (1992). Incremental Testing of Object-Oriented Class Structures. Proceeding of the International Conference on Software Engineering, 68 – 79.

Lee, M., Pham, H., & Zhang, S. (1999).A methodology of priority settings its application on software development. European Journal of Operational Research.

McGregor, J., & Korson, T. (1994). Integrated Object-Oriented Testing an Development Processes. Communications of the ACM, 59-77.

Parrish, A., Borie, R., & Cordes, D. (1993). Automated Flow Graph-Based Testing of Object-Oriented Software Modules. The Journal of Systems and Software, 95–109.

Smith, M., & Robson, D. (1994). A Framework for Testing Object- Oriented programs. Journal of Object-Oriented Programming, 45 – 53.

Tornatzky, L.G. & Fleischer, M.(1990). The Processes of Technological Innovation, Lexington, MA: Lexington Books.

Zhang, X., & Pham, H.(2000). An analysis of factors affecting software reliability. The Journal of Systems and Software, Elsevier, 50, .43-56.

Authors Biography:

 Mr. Sanjeev Patwa is Lecturer in Computer Science at FASC, Mody University of Science and Technology, (MUST), Lakshmangarh (Sikar) .He has long teaching experience. He received his B.Sc. (PCM) from M.L.Sukhadia University Udaipur , MSc (C.S.) & M.C.A. from JRN Deemed University, Udaipur. He also received "B" Level certificate from DOEACC Society, New Delhi. He has also done MTech (IT) from KSOU, Maysore. He has published 12 papers in National & International Journals and 10 papers are in International/National conferences and seminars. He is also author of 4 computer science books. His research interest is Software Engineering. He is Life Member of Rajasthan Science Congress.

Dr. Anil Kumar Malviya is an Associate Professor in Computer Science & Engineering Department at Kamla Nehru Institute of Technology, (KNIT), Sultanpur. He received his B.Sc. & M.Sc. both in Computer Science from Banaras Hindu University, Varanasi respectively in 1991 and 1993 and Ph.D. degree in Computer Science from Dr. B.R. Ambedkar University, Agra in 2006. He is Life Member of CSI, India. He has published over 40 papers in National & International Journals and 5 papers are in International/National conferences and seminars. His research interests are Data mining and Software engineering, Cryptography & Network Security.