An ANOVA based Code Review Evaluation Approach

Syed Usman Ahmed
Department of Information Technology
Jodhpur Institute of Engineering and Technology
Jodhpur, India
syedusman.ahmed@jietjodhpur.com

Gulrez Jamal Ansari
Industry Software and Solution, SSE
Computer Science Corporation
Mumbai, India
gansari@csc.com

***Abstract*— The data should be characterized by some statistical measure for the purpose of estimation or comparison with similar data or making inference about the sample population to which the data belong. Statistical measures can be classified into measure of central tendency, measure of variation and measure of skewness. In this paper we will present a approach by which we can evaluate the static testing technique called code review by using a measure of variation known as Analysis of Variance (ANOVA). This evaluation is based on parametric constraints like day of review, order of code review, knowledge and experience of subject, complexity of programs under review.**

Keywords*— *Code review, ANOVA technique, effectiveness, efficiency, Static testing.

I. Introduction

The ANOVA (Analysis of Variance) test is applied when our data consist of a quantitative response variable and one or more categorical explanatory variables. The categorical explanatory variable is also known as factor. Based on the number of factor involved ANOVA techniques can be classified as one way ANOVA which uses one factor and one response variable or two ways ANOVA which uses two factors.

As software testing is gaining momentum many researchers are getting associated with this domain to explore the real potential of testing [15]. Quality of software work product depends on the amount and quality of testing being done software reliability, scalability and performance are some of the factors that are very much valued by the customer [16]. Code review is a type of static testing approach by which the source code of software is examined for the presence of defects or errors. This code review has two dependent variables: failure detection or observation and fault isolation. A fault is observed if the subject now can find out the difference between the legal specification and his own recorded specification. Fault isolation means that the subject can precisely describe the problem in the source code of the program and also suggest the cause of occurrence [1, 9].

These two dependent parameters is affected by four independent parameters: day of code review, complexity of source code, order of code review, and subject experience. This study evaluates the affect of the above mentioned independent parameter on code review approach.

II. LITERATURE SURVEY

Software development is aimed to provide either software or a service for its clients. In future of software engineering (FOSE) a road map for testing was presented [4]. This road map laid stress on some fundamental research work and one of the parameter of fundamental research was demonstrating effectiveness of testing techniques using empirical studies. In FOSE 2007 [3] it was mentioned that additional research was needed to provide three types of evidences: analytical, statistical or empirical of the effectiveness of test selection criteria in revealing faults in order to understand the classes of faults for which the criteria are useful. In FOSE 2007 empirical body of evidence was identified as one of the important challenges. It is mentioned in [3] that in every topic of software engineering research, empirical studies are essential to evaluate proposed techniques and practices, to know how and when they work and to improve on them. This research work got its motivation for developing an empirical body of knowledge which is at the basis for building and evolving the theory for testing.

Moreover in a official report “State of code review 2013” [5] released by SmartBear software revealed that over 70% of respondent said that they do collaborative review in some capacity and those who do review are twice as likely as highly satisfied with their overall software quality. Over 90% of respondent said that conducting code review is important.

As per the current industry standard the software testing techniques can be classified into two basic categories: *static testing* and *dynamic testing*. If in a testing technique we require to execute the actual code and find out the bug or defects or errors then it falls under dynamic testing technique, whereas those testing technique in which execution of final code is not required for locating defects or bugs or errors are called as static testing techniques [7]. ***Code review*** is a systematic examination of source code and it is intended to detect and isolate mistakes overlooked in the development phases. Code review improves both the quality of software and the developers’ skills. There are various forms of reviews: peer review (*informal*), walkthrough (*informal*), inspection (*formal*).

Dynamic Testing / Execution based techniques focus on the range of ways that are used to ascertain software quality and validate the software through actual executions of the software under test [9].

III. RELATED WORK

The research on the comparison of testing technique traces back to as early as 35 years ago with Hetzel making a start in 1976 by conducting a controlled experiment in order to analyze three defect detection methods [8]. The most commonly studied factors in the experiments evaluating testing techniques are their effectiveness (i.e., number of detected defects) and efficiency (i.e., effort required to apply the technique) in programs [9]. By tracing the major research results that have contributed to the growth of software testing techniques we can analyze the maturation of software testing techniques research. We can also assess the change of research paradigms over time by tracing the types of research questions and strategies used at various stages [10]. Three directions of research have been found related to evaluation of testing techniques [9]:

1) Actual evaluations and comparisons of testing techniques based either on analytical or empirical methods.
2) Evaluation frameworks or methodologies for comparing and/or selecting testing techniques.
3) Surveys of empirical studies on testing techniques which have summarized available work and have highlighted future trends.

However, the most significant study was conducted by [11]. This experiment studied the effectiveness and efficiency of different code evaluation techniques. The work of Basili and Selby was first replicated by [1]. This replication assumed the same working hypotheses as in initial experiment, but the experiment changed the programming used of the source code. A fault isolation phase was also added in the experiment [9]. Their work was replicated again by [12]. Their experiment followed exactly the same guidelines as the experiment run by Kamsties and Lott (who had built a laboratory package to ease external replication of the experiment), although new analyses were added [9]. Further the experiment was replicated by [13]. Their experiment stressed on the fault types and did not considered efficiency of testing techniques.

IV. ANOVA TEST

As mentioned in [14] decomposition of total variability into its component is called analysis of variance (ANOVA). The various terms used in the ANOVA test can be explained by using the following example.

Example: Suppose we want to measure the height of some plants under the effect of three fertilizers.

The tabular layout of the problem shown in table 1

TABLE 1: EXAMPLE OF ANOVA TABLE

Treatment	Measures			Mean	$\hat{A}_i$
X	1	2	2	...	...
Y	5	6	5	...	...
Z	2	1		...	...
Overall mean	// ...				

Now each value of measure $Y_{i,j}$ is affected by three factors; individual mean of treatment, variance of individual mean from overall mean and error in treatment. The mean of treatment can be calculated as shown below:

$$\begin{aligned} Mean_X &= \frac{1+2+2}{3} = 1.667 \\ Mean_Y &= \frac{5+6+5}{3} = 5.333 \\ Mean_Z &= \frac{1+2}{2} = 1.5 \end{aligned}$$

The estimated overall mean û is calculated as follows:

$$\hat{\mu} = \frac{1+2+2+5+6+5+2+1}{8} = 3$$

The estimated affect $\bar{A}_i$ is the difference between the “estimated treatment mean” and the “estimated overall mean”, i.e.

$$\bar{A}_i = Mean_i - \hat{u}$$

So,

$$\begin{aligned} \bar{A}_1 &= 1.667 - 3 = -1.333 \\ \bar{A}_2 &= 5.333 - 3 = 2.333 \\ \bar{A}_3 &= 1.5 - 3 = -1.5 \end{aligned}$$

If we now modify the above mentioned table 1 with the values of mean, overall mean and estimated affect we will get the following updated table 2:

TABLE 2: EXAMPLE OF ANOVA TABLE

Treatment	Measures			Mean	$\hat{A}_i$
X	1	2	2	1.667	-1.333
Y	5	6	5	5.333	2.333
Z	2	1		1.5	-1.5
Overall mean	// 3				

There are four important terms associated with ANOVA table:

1. Degree of freedom (*df*)
2. Sum of squares (*SS*)
3. Mean squares (*MS*)
4. F-value test (*F*)

Sum of square (*SS*) are supposed to measure different kinds of variability in the data (between the group / within the group), however they also directly or indirectly influenced by the number of groups (order) and number of observations (subjects) in the test. This influence is measured by quantities called degree of freedom (*df*) which is associated with each sum of squares. Degree of freedom can be calculated as:

$$df_{\mathrm{Tot}} = N - 1, \qquad df_G = g - 1, \qquad df_E = N - g.$$

For the example of fertilizer the value of

$df_{\mathrm{Tot}} = 8 - 1 = 7, \quad df_{\mathrm{treat}} = 3 - 1 = 2, \quad df_{\mathrm{res}} = 8 - 3 = 5$

The can verify the fact that $df_{\mathrm{Tot}} = df_{\mathrm{G}} + df_{\mathrm{E}}$.

Mean squares (*MS*) are just sum of squares divided by their degree of freedom. The focus of ANOVA is a hypothesis test for checking whether all the groups have the same population mean. This is same as testing whether the response variable depends on the factor. It is sometime called as F-test.

$MS_{treat} = SS_{treat} / df_{\mathrm{treat}} = 13.08$
$MS_{res} = SS_{res} / df_{\mathrm{res}} = 0.37$

The F value is calculated as

$$F = MS_{treat} / MS_{res} = 35.68$$

The F-value is used to study the variation of the data from the hypothesis. It can be used to either accept or reject a null hypothesis in case of ANOVA technique.

V. Proposed work and Experiment design

A. Proposed Work

In order to use ANOVA test for evaluating the code review technique we have to use Goal-Question-Metrics (GQM) approach. This approach is used to state the goals of the experiment and based on the GQM approach we frame some main hypothesis. The main hypothesis is then further divided into testable hypothesis. Testable hypotheses are set of statements that can be tested using experiments by comparing the actual results with expected results [9].

The goal of the experiment can be stated as follows:

- **Find out the effectiveness in revealing failures**
- **Find out the efficiency in revealing failures**
- **Find out the effectiveness in isolating faults**
- **Find out the effectiveness in isolating faults**

Based on the goal above some question can be framed that will help us to test the hypotheses like:

1. What influence does each independent variable have on effectiveness of failure observation and fault isolation?
2. What influence does each independent variable have on the time to observe failure, time to isolate failure and the total time?
3. What influence does each independent variable have on the efficiency of failure observation and fault isolation?

B. Experimental desing

A procedure that is used to execute an experiment serves as a baseline to guarantee the accuracy of the experiment in the given environment. The procedure may involve training activities, execution of experiment, collecting data, providing feedback etc. A limited number of subjects should be taken so that the variation can be studied properly and conclusion can be drawn accurately. This selection should be based on certain criteria like Experience of subject, highest level of education, Knowledge of domain and subject area etc.

To use ANOVA of data set the data should be randomized and we should be using parametric technique for data analysis.

The subjects apply code review technique to say n number of different **programs** (first independent variable) in different **orders/groups** (second independent variable). The subject are required to complete this code review in some x number days and all subject work on same or different defect detection technique on same day. Finally the **subject** can be considered as the forth independent variable, which is however an uncontrolled independent variable.

Based on these parameters we can have the following types of hypothesis:

H_1: The independent variables do not have any effect on the failure observation or failure observation time or rate.

H_2: The independent variables have effect on the failure observation or failure observation time or rate.

H_3: The independent variables do not have any effect on the fault isolation or fault isolation time or rate.

H_4: The independent variables have effect on the fault isolation or fault isolation time or rate.

By studying the four parameters we can generate seven metrics which are:

1. Percentage of faults detected
2. Percentage of faults isolated
3. Time to detect faults
4. Time to isolate faults
5. Total time to detect and isolate faults
6. No. faults found / time
7. No. of faults isolated / time

If we study the affect of four parameters on these seven metrics we can find F-value for each metrics. Corresponding to each F-value we will get a p-value (significance value) if this significance value is equal to or less than 0.05 then we can reject the hypothesis otherwise the hypothesis is accepted.

VI. CONCLUSION

The ultimate goal of this evaluation approach is to generate metrics for assessing code review technique using ANOVA technique. There are seven metrics that can be generated from the raw data collected in the experiment. The metrics are:

1. Percentage of faults detected
2. Percentage of faults isolated
3. Time to detect faults
4. Time to isolate faults
5. Total time to detect and isolate faults
6. No. faults found / time
7. No. of faults isolated / time

This evaluation can be used to study the effectiveness and efficiency of code review technique with respect to varying lines of code or same line of code

REFERENCES

[1] Erik Kamsties and Christopher M. Lott, "An Empirical Evaluation of Three Defect-Detection Techniques", Experimental study, 1995.

[2] Malik, Qaisar Ahmad, "Combining Model-Based Testing and Stepwise Formal Development", Turku Centre for Computer Science. 2010.

[3] A. Bertolino, "Software testing research: Achievements, challenges, dreams", In future of software engineering 2007, FOSE'07, page 85-103 IEEE 2007.

[4] Mary Jean Harrold, "Testing: A Roadmap", In Future of Software Engineering, 22nd International Conference on Software Engineering, June 2000

[5] “State of code review 2013”, SmartBear Software Inc., SB-C-041713-WEB.

[6] Sheikh Umar Farooq, S.M.K. Quadri, "Evaluating Effectiveness of Software Testing Techniques With Emphasis on Enhancing Software Reliability", Journal of Emerging Trends in Computing and Information Sciences, ISSN 209-8407, Vol 2, No.12, 2011.

[7] M. Roper, Software testing, McGraw-Hill, Inc. , 1995.

[8] Hetzel, W. An experimental analysis of program verification methods. 1976.

[9] Juristo, N., Moreno, A., and Vegas, S. Reviewing 25 years of testing technique experiments. Empirical Software Engineering, 9(1):7-44, 2004.

[10] Luo, L. Software testing techniques. Institute for software research international Carnegie mellon university, Pittsburgh, PA, 2001.

[11] Basili, V. and Selby, R. Comparing the effectiveness of software testing strategies. Software Engineering, IEEE Transactions on, (12):1278-1296, 1987.

[12] Roper, M., Wood, M., and Miller, J. An empirical evaluation of defect detection techniques. Information and Software Technology, 39(11):763 - 775, 1997.

[13] Juristo, N., Moreno, A., and Vegas, S. Limitations of empirical testing technique knowledge. SERIES ON SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING, 12:1-38, 2003.

[14] R. Panneerselvam, Research Methodology, ISBN 81-203-2458-8, pp:71-81.

[15] Ahmed, Syed Usman and Azmi, Muhammad Asim, “A Novel Model Based Testing (MBT) approach for Automatic Test Case Generation”, International Journal of Advanced Research in Computer Science, 4(11), pp 81-83, 2013.

[16] Ahmed, Syed Usman, Sahare, Sneha Anil and Ahmed, Alfia, “Automatic test case generation using collaboration UML diagrams”, World Journal of Science and Technology, Vol-2, pp4-6, 2012.