

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2015

## Identification of Critical Defect Types in Projects Development

Pandurangan.A, Standley Sagayaraj.S

Department of Mathematics, Bharath University, Chennai, Tamil Nadu, India

Junior Quality Consultant, Temenos India (P) Ltd., Chennai, Chennai, Tamil Nadu, India

**ABSTRACT:** In software development, defects are inevitable irrespective of domain, technology and the life cycle followed. There may be 40-50 defect types found during the complete life cycle. A methodology is proposed in this paper with a case study to identify the critical defect types to focus on the possible X factors which will impact the Y. Also to identify the causes for the critical defect types based on which the recommendations can be made to reduce the defect and thereby deliver the defect free product and service to the customer.

**KEYWORDS:** ANOVA, F-Test, IDD, CI, Critical Defect Type, ODA.

### 1. INTRODUCTION

Software development is so unique to consider that each development process as unique, with different people, with different skill sets and priorities. The production of a functional or technical design artifact is a unique event. Variation in size and complexity of the artifact can have non-linear effects on defect levels. There can be many similarities between the development processes and the produced artifacts, of course, but not in the same way[11,7,9].

Reviews are performed during the course of software development irrespective of life cycle. The defect data or count obtained during the review will help to assess the quality of the deliverables and delivering the quality to the customer at first. Software review is the source to capture the defect count and a major source of the most valuable software measure – defect counts.

The review process enables the collection of data about the artifact inspected - the defect data. Software review and other software quality controls are performed primarily to identify defects injected at which phase to enable them to be removed before delivering to the customer[12]. Even though various methods and measures are taken to analyse the defects and to prevent bugs[1, 2, 9], many other causes influence for the occurrence of defects.

Defect counts can indicate the amount of rework required to fix and make the artifact useable, which also will have an impact on the development schedule. “Defect Densities” - the number of defects divided by the size - may give indications about the quality of the artifact. When compared with defect data and schedules from similar, previously developed artifacts they can provide predictions using the regression models[13]. The distribution of defects within an artifact can aid in the identification of problem areas or risk’.

The number of defects in a work product is an important measure as it gives us an idea about how much the customer will be satisfied[3]. It also gives ideas about the amount of rework to be done, how efficient our review processes work, which processes need improvement, which components of the system are error prone etc. Therefore, defect counts provide evidence not only on the quality of the product, but also on the quality of the related processes and ensure the goodwill when defect free or service is delivered in the first time itself[6].

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2015

## II. DEFECTS CLASSIFICATION

Before starting to count defects in software products, however, it is important to establish necessary background in order to provide consistency among measurements. First of all, there is not an exact agreement on the meaning of defect in software industry. The terms defect, fault, failure and error are usually used interchangeably[4]. In order to avoid such misconceptions and to establish a universal understanding. Although there are several often cited classifications and even an IEEE standard [5], none of which have become a truly and broadly applied practice.

However, it has been found in different case studies [3, 4] that general defect type classifications are difficult to use in practice and need to be refined or adapted to the specific domain and project environment. In [4, 6] it is shown how specific adoptions and domain-specific defect types can be found and defined. Yet, we are not aware of a larger use of these approaches[7].

**Defect:** A product anomaly is a defect. It is due to (i) omissions and imperfections found during early life cycle phases, and (ii) faults contained in software sufficiently for test or operation.

**Error:** Human action or inaction that results in software containing a fault.

**Failure:** A failure may be produced when a fault is encountered as a result of (i) the termination of the ability of a functional unit to perform its required function. (ii) An event in which a system or system component does not perform a required function within specified limit[8]s..

**Fault:** A fault, if encountered, may cause failure. It may be (i) an accidental condition that causes a functional unit fails to perform its required function. (ii) A manifestation of an error in software.

From these definitions, we understand that defect is named as a fault when it is found in software, and a failure occurs when the software malfunctions as a result of a fault. Thus, not every fault results in a failure; and a number of faults together may cause a system or component to fail its operation[10]. Actually, it is common to come up against different definitions of these terms in different environments. However, for ease of understanding, it is appropriate to refer to any software or documentation anomaly as a defect in our analysis.

If we consider an automobile, we see that it may encounter many problems with different criticalities, impacts, repair times and costs. If the breaks do not work properly, the result may be very serious, and even fatal. Therefore, a mechanic makes different treatments according to the characteristics of the problems. Software is similar to an automobile in that, each defect has distinctive properties. Not all defects have the same impact, urgency, origination point and cause. For this reason, it is not a good practice to compare software products only by looking at the number of defects without additional information[11].

Another difficulty arises because of the necessity to distinguish between an enhancement and an error. Sometimes change procedures may not be appropriate to distinguish between an enhancement and a defect since both represent a change. Moreover, the change procedures should be defined well enough to distinguish between an enhancement and a defect. It can be made either by using different processes and forms to implement an enhancement and a defect, or by depicting the type of change explicitly.

Thus, a detailed conceptual analysis will be needed to decide whether a change is a defect or not, and most probably such a practice will not be feasible. Even with relevant change procedures, it will still be difficult to determine the reason of a change because of marketing issues. From the customer's point of view, a functionality may be regarded as an inherent property, for which there is no need to write an explicit requirement. So when that functionality is not present, a customer may reluctant to accept the product. The developer would not think this way, and may argue against the customer's intention. Most probably, the change will be performed by the software company considering the market

# International Journal of Innovative Research in Science, Engineering and Technology

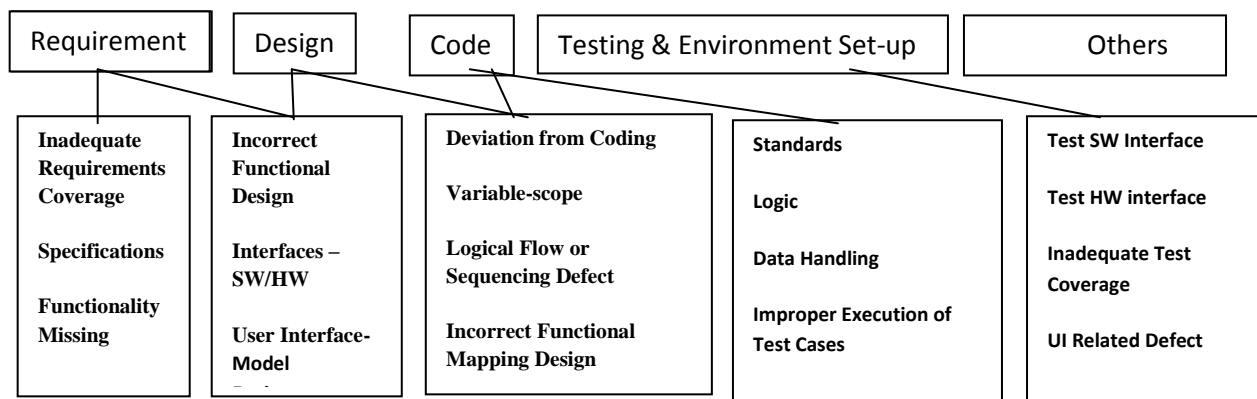
(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2015

conditions. Such cases are very frequently seen in software industry. And the natures of software development cause difficulties in determining whether such a change is an enhancement or a defect.

We see that, analysis and interpretation of defect requires some preliminary work and a precise understanding of some issues. First of all, a global perception of defect should be retained in order to avoid possible misconceptions regarding the definition of defect. Secondly, we should categorize defects and products with respect to specific characteristics up to a certain level for meaningful comparison among metric data. Neither all defects nor all the products are the same. The defects can be grouped by type, priority and impact, whereas the products can be classified within certain product groups such as requirements documents, design documents, code, and test documents. It is still possible to divide groups into subgroups as soon as the data is available and the detailed analysis is feasible. Each software organization should individually determine the level of detail and the categorization of the groups after working on the historical data, determining specific needs and analyzing related processes.

Having an understanding on defect origins beside the defective products would also be very beneficial for making a proper interpretation of defects. This information allows us to identify the cause of a defect and take corrective actions to improve the problematic process. A future defect may be a result of any previous process within the lifecycle. Thus, only the number of defects for a product may be misleading in terms of evaluating the quality of the related process. By recording the origin of each defect, we may extend our analysis to investigate the defect injection potentials of different processes.



**Fig-1: Classifications of Defects in Software Development Life Cycle**

### III. DATA COLLECTION AND METHODOLOGY

Various methods are available in the literature for the analysis of defects[7, 8]. The data is collected from a sample of 21 live projects which covered the entire life cycle of software envelopments projects. Waterfall model or the traditional life cycle with requirements, design, coding, testing, operation and maintenance is followed. Totally 45 possible defects were brainstormed from the projects teams and the possible and frequent occurring and most impacting defects listed below. From the available defect types the most critical and impacting defects to be identified using the appropriate statistical techniques. For the defects, injected defect density is calculated as total number defects (internal and external) divided by the total effort of developments in person hours. Table-1 shows the total defects of various types in each project.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2015

**Table-1**  
**Projects with the Defect Count of leading Software Company**

Project Name	Total Defects in Each Project
Proejct-1	23
Proejct-2	29
Proejct-3	25
Proejct-4	30
Proejct-5	33
Proejct-6	22
Proejct-7	40
Proejct-8	8
Proejct-9	17
Proejct-10	56
Proejct-11	36
Proejct-12	48
Proejct-13	39
Proejct-14	70
Proejct-15	58
Proejct-16	58
Proejct-17	19
Proejct-18	134
Proejct-19	15
Proejct-20	96
Proejct-21	55

It was decided to group those 45 critical defect types into 6 factors for all the projects. The total defects of 911 was categorized into 6 factors for the purpose of applying one-way classification. Table below shows the total defects for each factor.

**Table-2**  
**Sample Defect Type obtained from a Project**

Defect Type	Defect Count
Deviation From Coding Standards	257
Improper Execution Of Test Cases	52
Inadequate Requirement Coverage	196
Inadequate Test Coverage	57
Logical Flow Or Sequencing Defect	197
UI Related Defect	152

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2015

The contribution of different defect types to the projects is depicted in the figure as

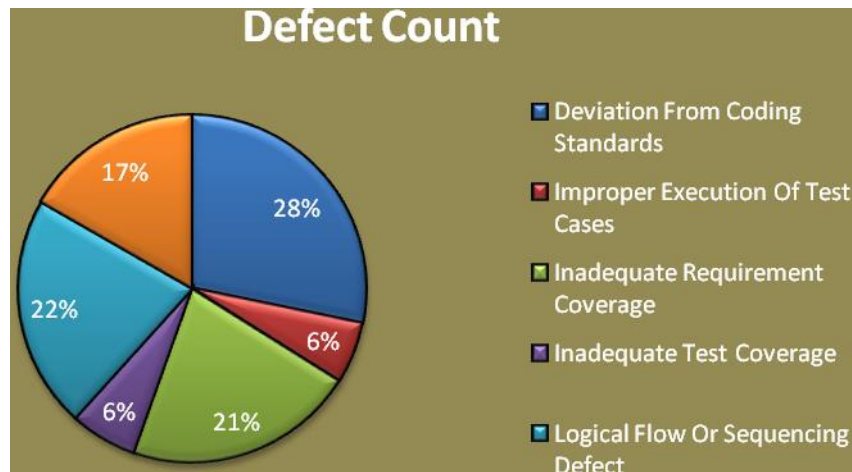


Fig-2: Distribution of the Defect Types

## IV. ANALYSIS

In order to apply ANOVA technique some important assumptions are made. The various assumptions are:

- Response variables are normally distributed
- Samples are independent.
- Variances of populations are unknown but equal.

If the defect density fails for the normality test, it better to make the appropriate transformation and ensure that defect density follows normal distribution. The transformation technique such as Box-Cox or Johnson or any other transformation can be used. Here, the Johnson transformations is applied to transform the defect density to follow a normal distribution. The ANOVA is performed for the transformed data.

### 4.1 One-Way Analysis of Variance (ANOVA)

One-way analysis of variance (ANOVA) is a technique used to compare means of three or more samples using the F-Test. For ANOVA, the null hypothesis is taken as samples in three or more groups are drawn from populations with the same mean values. To do this, three estimates are made of the population variance. In ANOVA the F-statistic is the ratio of the variance calculated among the means to the variance within the samples.

### 4.2 One Way ANOVA for Defect Types

**Objective:** Objective is to determine if there is a statistical significant difference between the Defect densities of various defect types.

We use One Way ANOVA to assess if there are true differences to act upon or not! Defect density of 45 defect types collated are subjected to ANOVA to determine the statistical difference.

**Null Hypothesis ( $H_0$ ):** There is no statistical difference among the Defect densities of defect type Means.

i.e.,  $H_0$ : Defect Type-1 = Defect Type-2 = ... = Defect Type -6.

**Alternative Hypothesis  $H_1$ :** There is a statistical difference among the defect type Means.

i.e.,  $H_1$ : Defect Type-1  $\neq$  Defect Type-2  $\neq$  ...  $\neq$  Defect Type -6.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2015

**Table-3**  
**One-way ANOVA for the six critical defect types**

Source of variation	D.F	SS	MSS	F-Ratio	P-Value
<b>Among Factors</b>	5	18.084	3.617	4.65	0.001
<b>Error</b>	120	93.392	0.778		
<b>Total</b>	125	111.476			

The P-value is 0.001, is less than 0.05, and so the Null Hypothesis  $H_0$  is rejected. As per  $H_1$ , there is statistical difference of the mean defect density between different defect types. It leads to the conclusion that all the 6 critical defect types vary among them in the 21 projects of our study.

### 4.3 Fisher's Test

To identify the statistically significant defect type(s) among the 6 defect types with respect to mean defect density, the Fisher's test is used. This test is a multiple comparison test applied when  $H_0$  is rejected in the one way ANOVA classification. The aim of Fisher's test here is to see whether the mean defect density of defect type, significantly differ between the different defect types. The output then consists of the mean difference (Center) and the 95% confidence interval (CI) for that mean differences.

If the CI (Confidence Interval) does not span zero (i.e. the signs do not differ), then the difference is significant between the chosen defect type and the other one under review. This is a statistically significant (different) defect type that is prioritized. Prioritize those defect types that have a +ve +ve sign for their confidence interval (-ve -ve symbolizes low defect density). If the Confidence Interval spans zero (i.e. the signs differ), then the difference in the means between the defect type under review and the chosen defect types are not significant.

#### Individual 95% CIs for Mean Based on Pooled Standard Deviations

Level	N	Mean	St. Dev	
-----+-----+-----+-----+				
Deviation from Coding	21	1.0409	0.6011	(-----*-----)
Inadequate Requirement	21	0.0478	0.9831	(-----*-----)
Logical Flow Or Sequencing	21	-0.0159	0.8958	(-----*-----)
UI Related Defect	21	-0.0099	1.0291	(-----*-----)
Inadequate Test	21	0.1365	0.9008	(-----*-----)
Improper Execution	21	0.0098	0.8177	(-----*-----)
-----+-----+-----+-----+				

## V. CONCLUSION

The identification of the critical defect type is so important for the project irrespective of the life cycle followed. In order to choose the critical type, the traditional way is to plot the Pareto chart and to say the defect with larger count will have the impacted, but the critical defect type can be statistically identified by Fisher F-Test. The defect type identified using Fisher test which will help in reducing the overall defect in the project at the injected phase. In this study it is found that Deviation from Coding Standards is more significant as compared with the other defect types such as, Inadequate Requirement Coverage, Logical Flow or Sequencing Defect UI Related Defect, Inadequate Test Coverage and Improper Execution of Test Case. This work can be further extended using design of experiments and orthogonal defect analysis with root cause analysis.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2015

## REFERENCES

- [1].Arnold Jesse C., and Reynolds Marion R. Jr., "CUSUM Control with Sample Size and Sampling Intervals", *Journal of Quality Technology*, **33**(1), pp. 66-81.(2001)
- [2].Jayaraman B., Valiathan G.M., Jayakumar K., Palaniyandi A., Thenumgal S.J., Ramanathan A., "Lack of mutation in p53 and H-ras genes in phenytoin induced gingival overgrowth suggests its non cancerous nature", *Asian Pacific journal of cancer prevention : APJCP*, ISSN : 13(11) (2012) PP. 5535-5538.
- [3].Box, and M. E. Mullerm "A Note on the Generation of Random Normal Deviates", *Ann. Math. Statistics*, **29**, pp. 610-611. (1958)
- [4].Kalaiselvi V.S., Saikumar P., Prabhu K., Prashanth Krishna G., "The anti Mullerian hormone-a novel marker for assessing the ovarian reserve in women with regular menstrual cycles", *Journal of Clinical and Diagnostic Research*, ISSN : 0973 - 709X, **6**(10) (2012) PP.1636-1639.
- [5].Costa A. F. B., "Chart with Variable Sample Size", *Journal of Quality Technology*, **26**, pp.155-163. (1994)
- [6].Subhashree A.R., Shanthy B., Parameaswari P.J., "The Red Cell Distribution Width as a sensitive biomarker for assessing the pulmonary function in automobile welders- a cross sectional study", *Journal of Clinical and Diagnostic Research*, ISSN : 0973 - 709X, **7**(1) (2013) PP. 89-92.
- [7].Lucas J. M., and Crosier R. B., "Fast Initial Response for CUSUM Quality Control Schemes", *Technometrics*, **24**, (1982).
- [8].Gopalakrishnan K., Prem Jeya Kumar M., Sundeep Aanand J., Udayakumar R., "Analysis of static and dynamic load on hydrostatic bearing with variable viscosity and pressure", *Indian Journal of Science and Technology*, ISSN : 0974-6846, **6**(S6) (2013) PP.4783-4788.
- [9]. Montgomery D.C. Introduction to Statistical Quality Control, 5<sup>th</sup> Edn. John Wiley and Sons, New York. 2005.
- [10].Srinivasan V., "Analysis of static and dynamic load on hydrostatic bearing with variable viscosity and pressure", *Indian Journal of Science and Technology*, ISSN : 0974-6846, **6**(S6) (2013) PP.4777-4782.
- [11]. Page. E.S "Cumulative Sum Control Charts," *Technometrics*, Vol 3, pp. 1-9. (1961)
- [12]. Pandurangan A., "Some Applications of Markov Dependent Sampling Scheme in SQC", *Unpublished Ph.D Thesis*, Bharathidasan University, Tiruchirappalli, (2002).
- [13]. Pandurangan A and R.Varadharajan, "Standardized CUSUM Control Chart". *International Journal of Mathematics and Applied Statistics(IJMAS)*, Vol 1(2). pp. 99-110.( 2010)