Analysis of Rejected Defects in First and Second Half Builds of Software Testing Phase

Ravi Sankar Landu

Project Manager – Testing, Inatech Infosolutions Pvt. Ltd.

Bangalore, India -560052

Lravisankar@yahoo.com

Abstract— **The objective of this research paper is to analyze the behaviour of rejected defects due to "Implied Spec" and "Invalid" reasons. A comparative study is made on the behaviour of reasons for rejected defects in particular between the first 50% of the builds and the next 50% of the builds. The study includes statistical analysis of the rejected defects in the first 50% and the second 50% of builds. The aim is to identify the dominating factor for defects rejection in the initial stages (First 50% of Builds) of software development and in the final stages (Second 50% of Builds) of the software development. This research aids Quality Assurance and Quality Control teams to decide when and where to spend more effort in reducing the rejected defects.**

Keywords— **Implied Spec Defects, Invalid Defects, Rejected Defects, Software Builds, Software Testing**

I. Introduction

Software testing now-a-days is considered an important activity of Software Development, thanks to the money saved by the clients in business due to improved quality of the product. There are many challenges surrounding Software Testing. One of the main challenging issues is time constraint. As per [1], there is very limited testing time available due to squeezed timeframes for system testing. There are many ways to deal with this time constraint, and one of them is to reduce the effort spent on bugs. According to [2] the major activity performed during software testing is to identify bugs and as per [3] more than 40% of project time is consumed over this activity. So, time saved in the defects activity helps in facing the time constraint challenge. A considerable amount of time of defects activity goes in analyzing the defects. There is possibility that a defect is not a valid one and is rejected by Development team. Time taken for logging these defects and analyzing these invalid defects can be saved if we reduce such defects. Defects get rejected because of test error, which is described by [4] as "the test as designed was incorrect (deviating from stated requirements or design) or was executed incorrectly or the resultant output was incorrectly interpreted by the tester, resulting in a defect "logged in error"."

[5]Gives some reasons why defect reports become invalid. 27.73% of invalid defects are due to misunderstanding of functionality by the tester [5]. Sometimes environmental issues also cause the defects as invalid [6]. Duplicate bugs are also a source of reason for rejecting defects. A duplicate bug report is also a very common issue and in some projects one quarter of reported bugs is duplicate [7]. Identification of duplicate bugs from a list of already reported bugs is itself a very time consuming activity [7].

After spending considerable amount of time in analyzing the defect, and finally finding it as invalid means the effort spent on the defect has a negative impact on the productivity.

Before we proceed with this defects analysis, we need to understand the Software Build Release Process. As per Wikipedia [8], "the term software build refers either to the process of converting source code files into standalone software artifact(s) that can be run on a computer, or the result of doing so. One of the most important steps of a software build is the compilation process where source code files are converted into executable code." Software is released as Builds to the Test team for testing. A build is represented by a number. Sometimes it is also called as Version number of the software. Each build contains a set of functionalities and defect fixes for testing. Defects found in one build are fixed and released in the following builds for testing. Thus, by the time a software system is completely developed, a number of builds are released to testing team. The frequency of these builds might vary depending on the effort needed for development of a feature or as per a company's process. In our company, the builds are released weekly. A Release Management team takes care of the Build Release to Test Team. Up on receiving a build, test team does Smoke Testing, and once it passes, the build is accepted for thorough testing.

In this paper, we analyze the "rejected defects" identified in different builds for a number of projects. The two main rejection criteria we consider for analysis of rejected defects are 1. ImpliedSpec (Meaning the defect is invalid because the feature is as per the Requirement specification or as per the Design) and 2. Invalid (due to Duplicate or Data or Environment issue)

The above two are the terms we use in our organization for the issues mentioned in brackets beside them. Defects are not rejected for any other reasons in our organization. Hence all the Rejected Defects fall under one of the two categories above.

A statistical analysis is done to understand the behavior of the above two reasons of rejected defects.

II. RELATED WORK

A lot of research studies are done on software defects pattern. For example [9] stressed on how analysis of defects found in first iteration can provide feedback for defect prevention in later iterations, leading to quality and productivity improvement. Most of the research focuses on classifying the defects into different severities, priorities, based on root causes, phase of occurrence etc. Study of defect patterns across different phases of SDLC (Software Development Life Cycle), and the analysis of this pattern is done by [10] and [11]. There are some studies, for example [2], addressing the rejected defects, which again concentrate on the root causes of rejections and measures to prevent them. White paper [12] talks about the defect reject reasons mentioned by Developers and Business Analysts, and how to deal with them. [2] Talks about root causes of defects software bug rejections.

The main gap in the previous studies is that they did not address the pattern and root causes of defects rejection in the initial builds and in the final builds of software within the Testing phase. A detailed study is needed to segregate the builds in testing phase of SDLC into first half and second half and study the defects rejection pattern in these two halves. This bisection of the testing phase and studying rejection patterns helps in identifying the specific root causes to a smaller data range instead of the whole defects data. Our paper bisects the rejected defects data and studies the dominant causes of rejection in the first half and second half of builds in Testing Phase. In this research paper we try to answer the question - when and where to spend more effort in reducing the rejected defects.

III. SCOPE AND LIMITATIONS

The selected projects are all completed and delivered projects, and are not considered for their type (Client-Server, Web, mobile, Cloud etc), Domain (Banking, Insurance, Embedded, Logistics etc) or Technology (Oracle Apps, SAP, DotNet, Java etc).

The scope of testing is limited to studying the behavior of the rejected defects in the First 50% of Builds and Second 50% of builds. Study of root causes for the behavior is beyond the scope of this paper. Only rejected defects with reasons ImpliedSpec and Invalid are considered. Rejected defects due to other reasons are out of scope of this research. Preventive and Corrective actions for reducing the defect rejections is not part of this research. Defect properties (like Severity, Priority etc) other than the reject reason are not considered for this research.

In case the number of builds is an even number, the first half and second half are divided equally as N/2 where N is number of builds in a project. In case the number of builds is an odd number, the first half is rounded to the lesser number and the second half is rounded to the greater number. For example, if there are 71 builds in a project, then 35 builds are considered as the First Half and 36 builds are considered as the second half.

IV. MATERIAL AND METHOD

Defects data from 8 completed projects is considered. Projects which are still in development or maintenance are not considered. Defects data is collected from defect trackers for all these projects. The defects are categorized based on build number. Rejected defects are identified as those whose status is marked as “ImpliedSpec” or “Invalid”. Defects which are not closed (instead postponed) are not considered in the defects count.

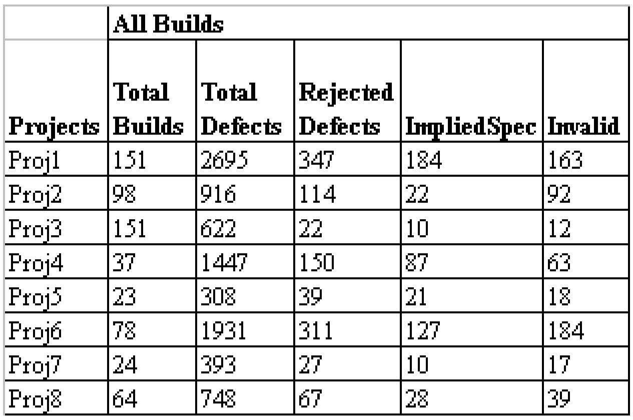
Defects are summed into three categories – 1) Sum of Defects of each project in All Builds 2) Sum of Defects in the First 50% of the builds 3) Sum of Defects in the Second 50% of the defects.

Defects are further divided as Total Defects, Total Rejected Defects, Total ImpliedSpec and Total Invalid under the above three categories.

A. Data

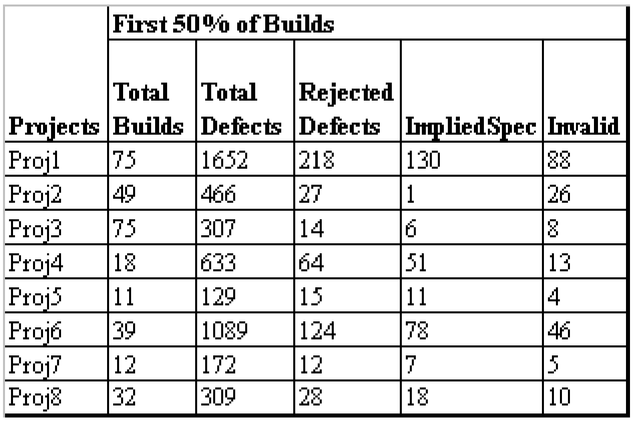
The Tables I, II, and III below show the defects data obtained for the 8 projects for All Builds, First 50% Builds and Second 50% Builds respectively.

TABLE I



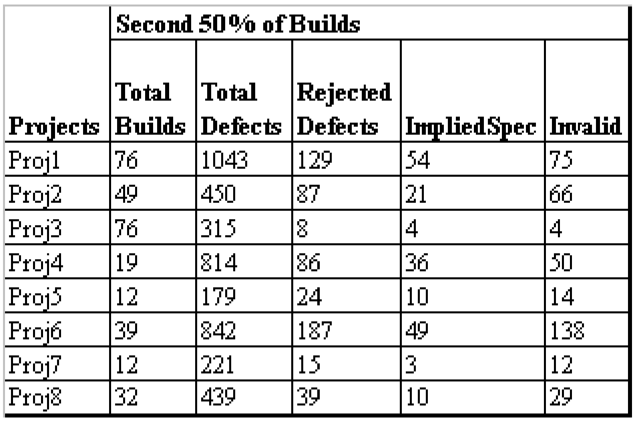
	All Builds				
Projects	Total Builds	Total Defects	Rejected Defects	ImpliedSpec	Invalid
Proj1	151	2695	347	184	163
Proj2	98	916	114	22	92
Proj3	151	622	22	10	12
Proj4	37	1447	150	87	63
Proj5	23	308	39	21	18
Proj6	78	1931	311	127	184
Proj7	24	393	27	10	17
Proj8	64	748	67	28	39

TABLE II



Projects	First 50% of Builds				
	Total Builds	Total Defects	Rejected Defects	ImpliedSpec	Invalid
Proj1	75	1652	218	130	88
Proj2	49	466	27	1	26
Proj3	75	307	14	6	8
Proj4	18	633	64	51	13
Proj5	11	129	15	11	4
Proj6	39	1089	124	78	46
Proj7	12	172	12	7	5
Proj8	32	309	28	18	10

TABLE III



Projects	Second 50% of Builds				
	Total Builds	Total Defects	Rejected Defects	ImpliedSpec	Invalid
Proj1	76	1043	129	54	75
Proj2	49	450	87	21	66
Proj3	76	315	8	4	4
Proj4	19	814	86	36	50
Proj5	12	179	24	10	14
Proj6	39	842	187	49	138
Proj7	12	221	15	3	12
Proj8	32	439	39	10	29

The analysis of ImpliedSpec and Invalid defects can be done globally with respect to the Total Defects in All Builds or with respect to the Rejected Defects in All Builds. The analysis can also be done with respect to the Total Defects in each set (First 50% Builds and Second 50% Builds) or with respect to the Rejected Defects in each set.

For our study, we have taken the comparisons of ImpliedSpec and Invalid Defects with respect to Rejected defects in the first and Second 50% Builds.

Based on this approach, the following sets of comparisons can be made:

Rejected Defects in First 50% Builds Vs ImpliedSpec in First 50% of Builds

Rejected Defects in First 50% Builds Vs Invalid Defects in First 50% of Builds

Rejected Defects in Second 50% Builds Vs ImpliedSpec in Second 50% of Builds

Rejected Defects in Second 50% Builds Vs Invalid Defects in Second 50% of Builds

The research also involves calculating the percentages of ImpliedSpec and Invalid defects with respect to Rejected defects for the respective All, First 50% and Second 50% Builds.

Also, Regression analysis is done for the defects. MS Excel is used for plotting graphs and for regression analysis.

V. Analysis

Line Fit Plots are drawn below for the four cases specified in the Material and Method

1) Rejected Defects in First 50% of Builds Vs ImpliedSpec in First 50% of Builds

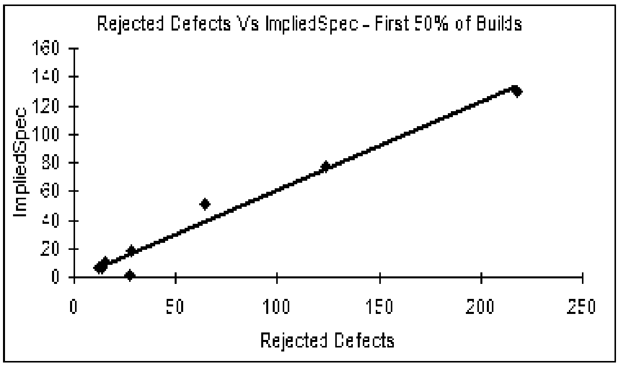


Fig. 1. Rejected Defects in First 50% of Builds Vs ImpliedSpec in First 50% of Builds.

Dots represent the data points while the solid line is best linear fit to the lines

From the Fig.1, we see that there is a linear relationship between the total number of Rejected Defects and the defects rejected due to ImpliedSpec reason in first 50% builds.

From Table IV below, we can find that the correlation (Adjusted R Square) is 96% between the Rejected Defects and ImpliedSpec Defects.

TABLE IV

Regression statistics for data in Fig. 1

Regression Analysis Coefficients Table for data in Fig. 1	
Multiple R	0.98619557
R Square	0.972581702
Adjusted R Square	0.968011986
Standard Error	8.200345159
Observations	8

R – Correlation Coefficient
R Square – Coefficient of Determination

TABLE V

REGRESSION ANALYSIS COEFFICIENTS FOR DATA IN FIG. 1

	Coefficients
Intercept	-0.977856734
Rejected Defects	0.617177

A regression equation framed from the above Table V, which is helpful in predicting the ImpliedSpec Defects based on Rejected Defects in the First 50% of builds is given below:
Regression Equation - In First 50% Builds:
ImpliedSpec Defects = -0.978 + 0.617(Rejected Defects)

Similar regression analysis is done below for the other 3 cases discussed in Material and Method

2) Rejected Defects in First 50% Builds Vs Invalid Defects in First 50% of Builds

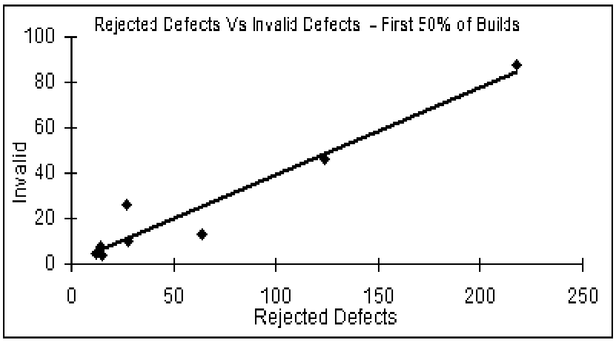


Fig. 2. Rejected Defects in First 50% Builds Vs Invalid Defects in First 50% of Builds

TABLE VI

REGRESSION STATISTICS FOR DATA IN FIG. 2

Regression Statistics for data in Fig. 2	
Multiple R	0.965261775
R Square	0.931730294
Adjusted R Square	0.920352009
Standard Error	8.200345159
Observations	8

TABLE VII

REGRESSION ANALYSIS COEFFICIENTS TABLE FOR DATA IN FIG. 2

	Coefficients
Intercept	0.977856734
Rejected Defects	0.382823

Regression Equation - In First 50% Builds:
Invalid Defects = 0.978 + 0.383(Rejected Defects)

3) Rejected Defects in Second 50% Builds Vs ImpliedSpec in Second 50% of Builds

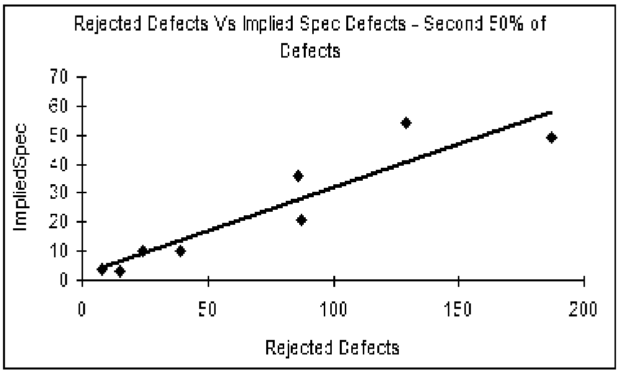


Fig. 3. Rejected Defects in Second 50% Builds Vs ImpliedSpec in Second 50% of Builds

TABLE VII

REGRESSION STATISTICS FOR DATA IN FIG. 3

Regression Statistics for data in Fig. 3	
Multiple R	0.927977811
R Square	0.861142817
Adjusted R Square	0.837999954
Standard Error	8.20344775
Observations	8

TABLE IX

REGRESSION ANALYSIS COEFFICIENTS TABLE FOR DATA IN FIG. 3

	Coefficients
Intercept	1.738337035
Rejected Defects	0.301031833

Regression Equation - In Second 50% Builds:
ImpliedSpec Defects = 1.738 + 0.301(Rejected Defects)

4) Rejected Defects in Second 50% Builds Vs Invalid Defects in Second 50% of Builds

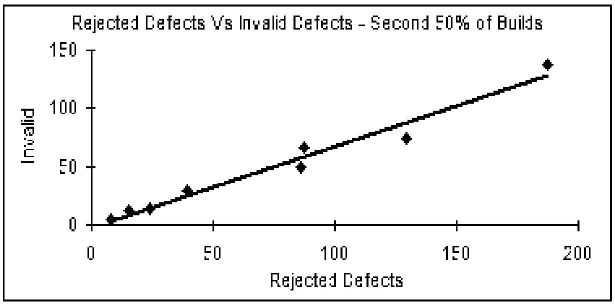


Fig. 4. Rejected Defects in Second 50% Builds Vs Invalid Defects in Second 50% of Builds

TABLE X

REGRESSION STATISTICS FOR DATA IN FIG. 4

Regression Statistics	
Multiple R	0.985372764
R Square	0.970959484
Adjusted R Square	0.966119398
Standard Error	8.20344775
Observations	8

TABLE XI

REGRESSION ANALYSIS COEFFICIENTS TABLE FOR DATA IN FIG. 4

	Coefficients
Intercept	-1.738337035
Rejected Defects	0.698968167

Regression Equation - In First 50% Builds:
Invalid Defects = -1.738 + 0.699(Rejected Defects

A. Summary

The coefficients of Rejected Defects for the ImpliedSpec and Invalid defects in First and Second 50% of Builds is summarized in the Table XII below:

TABLE XII

SUMMARY OF REGRESSION ANALYSIS COEFFICIENTS FROM TABLES V, VII, IX & XI

	First 50% of Builds	Second 50% of Builds
ImpliedSpec Defects	0.617177	0.301031833
Invalid Defects	0.382823	0.698968167

From the Table XII above, it is evident that in the First 50% of Builds, 62% of rejected defects are due to the reason "ImpliedSpec Defects" and in the latter half of the Builds, 70% of rejected defects are due to the reason "Invalid Defects".

The Adjusted R Square values of all the regressions are over 80% which means that there is a strong correlation between the variables – Rejected Defects and the corresponding ImpliedSpec Defects and Invalid Defects.

B. Representing in terms of Percentages

Percentage of Rejected Defects with respect to ImpliedSpec Defects and Invalid Defects is calculated in Table XIII and Table XIV below.

Graphs representing the trend lines of ImpliedSpec defects and Invalid defects in the First 50% and Second 50% of Builds are given below in Fig. 5 and Fig. 6.

TABLE XIII

IMPLIED SPEC & INVALID DEFECTS GIVEN AS PERCENT OF REJECTED DEFECTS – FIRST 50% BUILDS

First 50% Builds					
Project	Rejected Defects	Implied Spec Defects	Invalid Defects	% of Implied Spec Defects	% of Invalid Defects
Proj1	218	130	88	59.63	40.37
Proj2	27	1	26	3.70	96.29
Proj3	14	6	8	42.84	57.14
Proj4	64	51	13	79.69	20.31
Proj5	15	11	4	73.33	26.67
Proj6	124	78	46	62.90	37.09
Proj7	12	7	5	58.33	41.67
Proj8	28	18	10	64.29	35.71

% of Implied Spec Defects = (ImpliedSpec/Rejected)*100
% of Invalid Defects = (Invalid/Rejected)*100

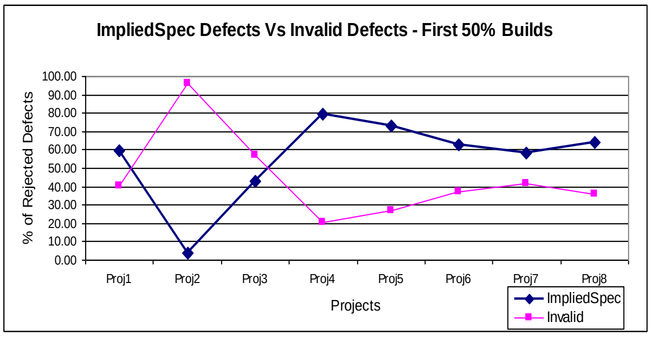
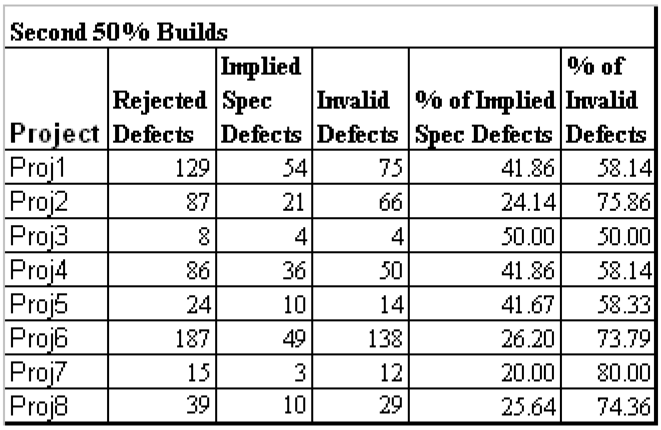


Fig. 5. ImpliedSpec and Invalid Defects as % of Rejected defects in different projects

From Fig.5, it is clear that major percent of Rejected Defects are due to ImpliedSpec Defects (except for Project2 and Project3) in the First 50% of Builds.

TABLE XIV

IMPLIED SPEC AND INVALID DEFECTS GIVEN AS PERCENT OF REJECTED DEFECTS – SECOND 50% BUILDS



Second 50% Builds					
Project	Rejected Defects	Implied Spec Defects	Invalid Defects	% of Implied Spec Defects	% of Invalid Defects
Proj1	129	54	75	41.86	58.14
Proj2	87	21	66	24.14	75.86
Proj3	8	4	4	50.00	50.00
Proj4	86	36	50	41.86	58.14
Proj5	24	10	14	41.67	58.33
Proj6	187	49	138	26.20	73.79
Proj7	15	3	12	20.00	80.00
Proj8	39	10	29	25.64	74.36

% of Implied Spec Defects = (ImpliedSpec/Rejected)*100
% of Invalid Defects = (Invalid/Rejected)*100

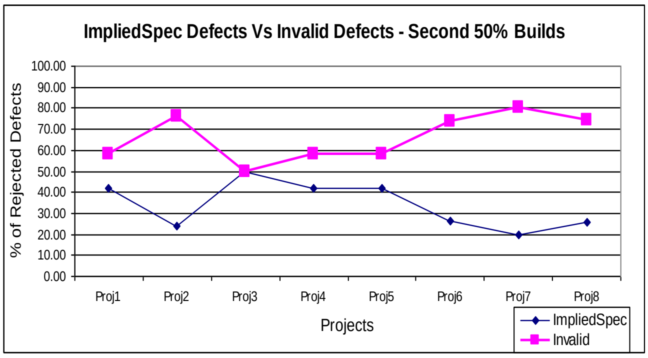


Fig. 6. ImpliedSpec and Invalid Defects as % of Rejected defects in different projects

From Fig.6, we see that in the Second 50% of Builds, major percent of Rejected Defects are due to Invalid Defects.

We can summarize that in the First 50% of Builds, ImpliedSpec Defects are dominating the Invalid Defects, and in the Second 50% of Builds, Invalid Defects are dominating the ImpliedSpec Defects.

VI. Conclusion and Future Work

From our study of defects from 8 projects, we find that, in the first half builds of testing phase, more defects are rejected due to the reason ImpliedSpec, meaning the defects are rejected because they are as per requirements specification and design. In the second half of testing phase, more defects are rejected due to the reason Invalid defects, meaning either environment issue or data issue or duplicate defect.

So, in the initial builds (first half) of testing phase, the Quality team has to act more on improving the process to reduce the defects rejected due to reason “ImpliedSpec”. In the later builds (second half) of testing phase, the Quality team has to act more on reducing the defects rejected due to duplicity or environment issues.

Further studies can be made to identify the root causes for this behavior of rejected defects.

There is scope for further research in this direction. The builds can be split into more frequencies, like 10% intervals instead of 50%, and analyzed to identify the reasons at each stage of testing. More reasons can also be compared to find out which reason is causing more defects in which stage of testing.

References

[1] Nageswara Rao Pusuluri, Software Testing Concepts and Tools (Reprint Edition: 2008)

[2] Ghazia Zaineb & Dr. Irfan Anjum Manarvi, Identification And Analysis Of Causes For Software Bug Rejection With Their Impact Over Testing Efficiency, International Journal of Software Engineering & Applications (IJSEA), Vol.2, No.4, October 2011

[3] Trivedi, P., & Pachori, S., Modeling and Analysis of Software Defects Prevention Using ODC, International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 1, No. 3, pp75-77, 2010

[4] Root Cause Analysis at http://sqa.fyicenter.com/Introduction_to_Software_Testing/35_Root_Cause_Analysis.html

[5] Sun, J., (2011) “Why are Bug Reports Invalid?”, IEEE 4th International Conference on Software Testing Verification and Validation(ICST), pp407-410.

[6] Qin, F., Tucek, J., & Zhou, Y., (2005) “Treating Bugs As Allergies: A Safe Method for Surviving Software Failures”, Proceedings of the 10th conference on Hot Topics in Operating Systems(HOTOS), Vol. 10, pp.19-19.

[7] Jalbert, N., & Weimer, W., (2008) “Automated Duplicate Detection for Bug Tracking Systems”, IEEE international Conference on Dependable Systems and Networks with FTCS and DCC(DSN), pp52-61.

[8]Software Build, http://en.wikipedia.org/wiki/Software_build

[9]Pankaj Jalote, Naresh Agarwal, “Using Defect Analysis Feedback for Improving Quality and Productivity in Iterative Software Development” In proc- ITI 3rd International Conference on Information and Communications Technology, pp. 703-713, 2007

[10]Sakthi Kumaresh & R Baskaran, Defect Analysis and Prevention for Software Process Quality Improvement, International Journal of Computer Applications (0975 – 8887) Volume 8– No.7, October 2010

[11] Suma V and T R Gopalakrishnan Nair , “ Effective Defect Prevention Approach in Software Process for Achieving Better Quality Levels” Proceedings of World Academy of Science, Engineering and Technology Volume 32 August 2008

[12]Shaham Yusuf and Vivekanand Suman, Revitalizing Rejected Defects: A value addition to projects, White paper authored for: QAI STC 2011

AUTHOR'S BIOGRAPHY

Ravi has done Master of Technology from Sri Venkateswara University, Tirupati, India, in Instrumentation and Control Systems. Ravi has 10years of experience in Quality and Software Testing. Ravi has experience in testing both application and system softwares. Ravi has over 5years of experience in testing Network Management System applications. Ravi currently works as a Software Test Manager in IT field. He holds ISEB Intermediate Certification in Software Testing. His research areas of interest include improving quality through Software Testing, improving Software Engineering processes, software defects analysis, and usability testing.