International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 4 Issue 6, June-2015, pp: (566-571), Impact Factor: 1.252, Available online at: www.erpublications.com

Research Paper on Analysis of Correlation between Prerelease and Post release Defects

Satyander Kumar[1], Kamna Solanki[2]

[1]M.Tech Student, Dept. of CSE, UIET, MDU, Rohtak
[2]Asst. Professor, Dept. of CSE, UIET, MDU, Rohtak

Abstract: This paper presents that if there exists some correlation between the software defects before the release of the software and after the release of the software. The data were collected for 30 modules. The total product defects before and after the release of software module to market was also counted using a simple counter. The collected data were then analyzed using two formulae for finding the coefficient of correlation between two components which are:

1. **Karl Pearson's coefficient of correlation**
2. **Spearman's Rank coefficient of correlation**

INTRODUCTION

Defects are really hard to handle and even more dangerous when they are not detected in the stage in which they are injected. As the defect keep moving from one stage to another it becomes harder to detect it and even harder to fix it. It takes a lot of efforts in terms of time and money. Software Testing is the process of executing a program or system with the intent of finding errors Or it involves any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. Pre-release testing is also known as beta testing. Beta testing is performed prior to the commercial release. Post release defects are the defects encountered during the actual use of the software by the customers. Research will help the industries to catch the defects before the release of the software product, it will be easier and cheap for them to recover from the errors before the the release of the software product.

Data Collection

The data were collected from XYZ Company located at Noida. To record the test data, the XYZ Company used MANTIS defects tracking system. MANTIS is written in PHP and works with MY SQL and MS SQL databases. The pre-release software testers manually recorded defects found before the release of software into defect repository system. The approximate time to conduct testing and collect data was approximately 60 days. Then the data were analyzed using Karl Pearson's coefficient of correlation and Spearman's coefficient of correlation.

Research Findings

The purpose of the study was to access the relationship between Pre-release and Post release software defects. The data regarding the pre-release and post release defects were collected from the XYZ company situated at Noida for a software X. The study involved observing and assessing the software testing activities of a specific application performed by the testing team of XYZ Software Company, a fictitious name for a commercial company located in Noida. The data were collected for 30 modules. The total product defects before and after the release of software module to market was also counted using a simple counter. Then the data were used to find out the relationship between pre-release and post release defects.

The collected data were then analyzed using two formulae for finding the coefficient of correlation between two components which are:

1. **Karl Pearson's coefficient of correlation**
2. **Spearman's Rank coefficient of correlation**

Karl Pearson coefficient of correlation

In statistics, the Pearson product-moment correlation coefficient (r) is a common measure of the correlation between two variables X and Y. When measured in a population the Pearson Product Moment correlation is designated by the Greek letter rho (?). When computed in a sample, it is designated by the letter "r" and is sometimes called "Pearson's r." Pearson's correlation reflects the degree of linear relationship between two variables. It ranges from +1 to -1. A correlation of +1 means that there is a perfect positive linear relationship between variables. A correlation of means that there is a perfect negative linear relationship between variables. A correlation of 0 means there is no linear relationship between the two variables. Correlations are rarely if ever 0, 1, or -1. If you get a certain outcome it could indicate whether correlations were negative or positive.

The quantity r, called the linear correlation coefficient, measures the strength and the direction of a linear relationship between two variables. The linear correlation coefficient is sometimes referred to as the Pearson product moment correlation coefficient in honor of its developer Karl Pearson.

The mathematical formula for computing r is:

$$r = N\sum xy - \left(\sum x\right)\left(\sum y\right) \div \sqrt{\left[N\sum x^2 - \left(\sum x\right)^2\right]}\sqrt{\left[N\sum y^2 - \left(\sum y\right)^2\right]}$$

Where:

N= number of pairs of scores
$\sum xy$= sum of the product of paired scores
$\sum x$=sum of x scores
$\sum y$=sum of y scores
$\sum x^2$=sum of squared x scores
$\sum y^2$=sum of squared y scores

Results after using Karl Pearson coefficient

The following data were applied to the Karl Pearson coefficient of correlation:

S. No.	No. of pre-release Defects	No. of Post Release Defects
1	10	3
2	30	4
3	21	2
4	51	9
5	20	19
6	41	4
7	2	10
8	50	14
9	33	4
10	7	5
11	16	0

12	55	9
13	9	0
14	110	3
15	25	0
16	8	3
17	118	15
18	78	4
19	7	3
20	98	10
21	67	0
22	7	5
23	97	12
24	12	6
25	102	27
26	4	8
27	59	10
28	26	3
29	39	0
30	18	4

These data were applied to the formula and the results were:

N= 30
$\sum x$=1220
$\sum y$=196
$\sum x^2$=86090
$\sum y^2$=2396
$\sum xy$= 10811

From the above results the value of r= 0.39. It means a weak but positive and significant relationship between the number of pre-release and post release defects. Hence the null hypothesis was rejected.

Spearman's Rank Coefficient of Correlation

Spearman's Rank correlation coefficient is used to identify and test the strength of a relationship between two sets of data. It is often used as a statistical method to aid with either proving or disproving a hypothesis e.g. the depth of a river does not progressively increase the further from the river bank. The formula used to calculate Spearman's Rank is shown below:

$$r = 1 - 6\sum d^2 \div \left(n^3 - n\right)$$

.

Where, r = relationship between Two components
d = difference between ranks
n = number of samples/observations

Results after Using Spearman's Rank coefficient of correlation

After using the Spearman's formula the results were:

S. No.	No. of Pre-release defects	No. of Post release defects	Rank 1	Rank2	d^2
1	10	3	8	9	1
2	30	4	16	14	4
3	21	2	13	6	49
4	51	9	21	21.5	0.25
5	20	19	12	29	289
6	41	4	19	14	25
7	2	10	1	24	529
8	50	14	20	27	49
9	33	4	17	14	9
10	7	5	4	1705	182.25
11	16	0	10	3	49
12	55	9	22	21.5	0.25
13	9	0	7	3	16
14	110	3	29	9	400
15	25	0	14	3	121
16	8	3	6	9	9
17	118	15	30	28	4
18	78	4	25	14	121
19	7	3	4	9	25
20	98	10	27	24	9
21	67	0	24	3	441
22	7	5	4	17.5	182.25
23	97	12	26	26	0
24	12	6	9	19	100
25	102	27	28	30	4
26	4	8	2	20	324
27	59	10	23	24	1
28	26	3	15	9	36
29	39	0	18	3	225
30	18	4	11	14	9

The value of $\sum d^2 = 3214$

The value of r = 0.28.

Again it means that number of pre-release and post release defects are related to each other. There is a positive and significant relationship between them.

Hence, from both the results the null hypothesis H_0 is rejected. The result shows that the cumulative sum of number of discovered post release defects slowly increases according to the increase of the pre-release defects.

Conclusion

An investment in the detection of pre-release defects is known to be less expensive than the cost of fixing defects found post release [29]. Therefore, during the pre-release testing process, software managers spend time and efforts detecting and fixing pre-release product defects [30]. Nevertheless, despite time and efforts spent, the use of a measurable pre-release testing process is still lacking in business practice [31]. The effect of pre-release software testing on the quality of the product is unpredictable, causing software managers to describe software product quality without a real degree of certainty [32].

References

[1]. Marc Eaddy et al. "On the Relationship between Crosscutting Concerns and Defects: an Empirical Investigation" IEEE transaction on Software Engineering Vol. 34 No. 4 pp. 497-515. July 2008.
[2]. Humphrey, W. "The software quality challenge". Crosstalk: The Journal of defense Software Engineering, 21, 4-9. (2008).
[3]. LiGuo Huang. "Software Quality Analysis: A Value Based Approach".
[4]. Foutse Khomh et al. "Predicting Post-release defects Using Pre-release Field Testing Results" ICSM 2011.
[5]. Ganssle, J. "Lean Coding". Embedded System Design, 21, 36. (2008).
[6]. Gonzalez, L.A. "Approach to measuring software quality". Retrieved from http://www.sqamethods.com.
[7]. Gupta, A., & Bhatia. (2010). "Testing functional requirements using B model specifications". ACM SIGSOFT Software Engineering Notes, 35, 1-7.
[8]. Walsh,J.(2010). "Combining the power of DMAIC with testing processes". Retrieved from http://www.isixsigma.com .
[9]. Bertolino, A(2007). "Software Testing Research: Achievements, challenges, dreams". In FOSE'07: 2007 Future of Software Engineering. IEEE Computer Society.
[10]. Burgin, M. et al. "Measuring testing as a distributed component of the software life cycle". Journal of computational methods in sciences and engineering, 9, 211-223. New York, NY: The association for computing machinery press.
[11]. Fenton, N. E. "Quantitative analysis of faults and failures in a complex software system". IEEE transactions software engineering, 26, 797-814. (2000).
[12]. J. Musa, A. Iannino, and K. Okumoto, "Software Reliability. Measurement, Prediction, Application" McGraw-Hill, 1987.
[13]. A. Mockus and D. Weiss, "Interval quality: relating customer perceived quality to process quality," in ICSE 2008: Proceedings of the 30th international conference on Software engineering. NY, USA: ACM, pp. 723–732.2008.
[14]. Phil Bantz et. Al. "Management and software quality assurance: A White Paper" Softaware Quality Assurance Subcommittee of the Nuclear Weapons Quality Managers. Quality Reports SQAS92-002. June 1993.
[15]. Regnell, B. et al. "Supporting roadmapping of quality requirements". IEEE software, 25, 42-47. 2008.
[16]. Harrington, H. "Poor Quality Cost". New York, NY: AQQC Quality press. 1987.
[17]. Yin, R. K. "Case study research: design and methods". Thousand Oaks, CA: Sage. 2009.
[18]. Jones, C. "Measuring defects potential and defect removal efficiency." Crosstalk: The journal of defecnse Software engineering, 21, 11-13. 2008.
[19]. Wu, S., Wang, Q.(2008). "Quantitative analysis of faults and failures with multiple releases of software". In ESEM '08: Proceedings of the Second ACM-IEEE, New-York.
[20]. Trochim, W. & Donnelly, J. "The research methods knowledge base". Mason, OH:Thomson learning. 2008.
[21]. Zikmund, W.G. "Business research methods". Mason OH: Thomson Learning.2003.
[22]. Sakthi Kumaresh, R. Bhaskaran." Defect Analysis and prevention for software quality improvement". International journal of computer applications (0975-8887) volume 8 no. 7. 2010.
[23]. Gittens, J. (2008). "Lean Coding". Embedded System Design, 21, 36.
[24]. Myers, Glenford J., "The art of software testing". Publication info: New York : Wiley. ISBN: 0471043281 Physical description: xi, 177 p. : ill. ; 24 cm.1979.
[25]. http://www.rstcorp.com/definitions/software_testing.html
[26]. Hetzel, William C.,"The Complete Guide to Software Testing", 2nd ed. Publication info: Wellesley, Mass. : QED Information Sciences. ISBN: 0894352423.1988.

[27]. Cem Kaner, "Testing Computer Software". 1993.
[28]. Capers Jones, Olivier Bonsignour. "The Economics of Software Quality, Video Enhanced Edition". 2011.
[29]. Silva, L.S., & Someren, M.V. "Evolutionary testing of object-oriented software". In Proceedings of ACM Aymposium on applied computing. New York.2010.
[30]. Pai, G.J., & Dugan J.B. "Empirical Analysis of Software fault content and fault proneness using Bayesian methods". IEEE transaction on software engineering, 33, 675.2007
[31]. Glass, R. L., Collard, R., Bertolino, A., & Kaner, C. "Software Testing and industry needs". IEEE Software, 23, 55-57. 2006.
[32]. Solanki K., "Effective Defect Detection and Reporting" published in the International Journal ACST (Advances in Computational Sciences and Technology). (ISSN 0973-6107) Vol.1, No.3, pp 165-174. 2008.
[33]. Solanki K.,Dalal S.,Bharti V., "Software Engineering Education and Research in India: A Survey" published in the International Journal IJES (International Journal Of Engineering Studies) ISSN 0975- 6469, Vol.1, No.3,. pp. 181–192. 2009.
[34]. Bendifallah, S., and W. Scacchi, "Understanding Software Maintenance Work", IEEE Trans. Software Engineering, 13,3, 311-323, 1987.
[35]. Bendifallah, S. and W. Scacchi, "Work Structures and Shifts: An Empirical Analysis of Software Specification Teamwork", Proc. 11th. Intern. Conf. Software Engineering, IEEE Computer Society, 260-270, 1989.
[36]. Paulk, M.C., C.V. Weber, B. Curtis, "The Capability Maturity Model: Guidelines for Improving the Software Process", Addison-Wesley, New York, 1995.
[37]. Truex, D., R. Baskerville, and H. Klein," Growing Systems in an Emergent Organization", Communications ACM, 42(8), 117-123, 1999.
[38]. Winograd, T. and F. Flores, "Understanding Computers and Cognition: A New Foundation for Design", Ablex Publishers, Lexington, MA, 1986.
[39]. Williams, L., "Software Process Modeling: A Behavioral Approach", Proc. 10th. Intern. Conf. Software Engineering , IEEE Computer Society, 174-200, 1988.
[40]. Steve Easterbrook, "Software Lifecycles", University of Toronto Department of Computer Science, 2001.
[41]. A. Pnueli and M. Shalev." What is in a step: on the semantics of statecharts". Proceedings of the International Conference on Theoretical Aspects of Computer Software, Japan, September 1991.
[42]. Elaine J. Weyuker." Testing component based software: a cautionary tale". IEEE Software, 15(5), October 1998.