
Significance of Chi-square Test for Software Fault Prediction

S. Warambhe*, Ritika Pande

*Department of Computer Engineering, Department of Information Technology, Xavier
Institute of Engineering, Mumbai, Maharashtra, India
Email: warambhesujata78@gmail.com*

ABSTRACT

Software Defect Prediction (SDP) is an important activity in testing Phase of Software Development Life Cycle. It identifies the modules that are defect prone and requires extensive testing. This way, the testing resources can be used efficiently without violating the constraints. Though Software Defect Prediction (SDP) is very helpful in testing, it's not always easy to predict the defective modules. There are various issues that hinder the smooth performance as well as use of the defect prediction models. In my research, I would like to carry out a survey on different software firms to identify and analyze their software prediction models, carry out a SWOT (Strength Weakness Opportunities Threats) analysis of each model, identify non value added activities in these models using VSM (Value Stream Mapping) and FMEA (Failure Model Effective Analysis) as a tool & finally come out with an adoptive model which will have benefits of all models & can be widely used.

Keywords: *Value Stream Mapping, Failure Model Effective Analysis, SWOT*

INTRODUCTION

The procedure is a discourse in which the learning that must turn into the product is united and encapsulated in the product. The procedure gives connection amongst clients and originators, amongst clients and developing devices, and amongst architects and advancing apparatuses [technology]. It is an iterative procedure in which the developing device itself fills in as the medium for correspondence, with each new round of the discourse evoking more helpful learning from the general population included.

Hazard Analysis and Management are a progression of steps that assistance a product group to comprehend and oversee vulnerability. Numerous issues can torment a product venture. A hazard is a potential issue—it may happen, it may not. In any case, paying little heed to the result, it's a great plan to distinguish it, evaluate its likelihood of event, assess its effect, and build up an emergency course of action should the issue really happen. Everybody engaged with the product procedure—supervisors, programming specialists, and clients—take part in chance examination and administration. To defeat chance, Software Defect Prediction can recognize surrenders and distinguish potential deformities utilizing relapse.

Existing programming deformity expectation models that are improved to anticipate expressly the quantity of imperfections in a product module may neglect to give a precise request since it is extremely hard to foresee the correct number of imperfections in a product module because of boisterous information. There are diverse blame expectation approaches accessible in the Software Engineering discipline. For example, blame expectation, security forecast, exertion forecast, reusability forecast, test exertion forecast and quality expectation. These methodologies assist us with minimizing the cost of testing which limit the cost of the undertaking.

Many research thinks about in 10 years have concentrated on proposing new measurements to fabricate expectation models. Generally considered measurements are Source Code and Process Metrics. Source Code measurements measure how source code is perplexing and the principle reason of the source code measurements is that source code with higher multifaceted nature can be more bug-inclined. Process Metrics are separated from programming files, for example, variant control frameworks and issue following frameworks that deal with all advancement chronicles. Process Metrics evaluate numerous parts of programming improvement process, for example, changes of source code, responsibility for code documents, designer collaborations, and so on. Value of process measurements for imperfection expectation has been demonstrated in numerous examinations.

Most deformity forecast ponders are led in view of factual approach, i.e. machine learning. Expectation models learned by machine learning calculations can foresee either bug-inclination of source code (order) or the quantity of deformities in source code (relapse). Some examination contemplates received ongoing machine learning procedures, for example, dynamic/semi-directed figuring out how to enhance forecast execution. BugCache calculation, which uses territory data of past imperfections and keeps a rundown of most bug-inclined source code records or strategies. BugCache calculation is a non-factual model and not quite the same as the current imperfection forecast approaches utilizing machine learning strategies. Analysts additionally centered around better expectation granularity.

LITERATURE SURVEY

Er.Rohit Mahajan et.al. [1] Author saw that the majority of the methods of programming issue identification depend on the machine learning methodologies and utilizing the NASA's open datasets to anticipate the product shortcomings. Open Datasets are for the most part situated in PROMISE and NASA MDP (Metrics Data Program) stores and they are disseminated uninhibitedly. Technique Level measurements and Class Level measurements are vitally utilized. Machine learning models have preferable highlights over Statistical strategies or master conclusion. In this way, it is discovered that machine learning models are for the most part utilized and these models are utilized to build the use of open datasets for blame expectation in future [1]

Pradeep Kumar Singh et.al. [2] According to Authors study, it was obvious that item metric, process measurements and protest arranged measurements are broadly utilized in blame expectation strategies. Be that as it may, blame expectation result is additionally reliant on human mastery separated from these measurements. So estimating human skill in programming shortcoming forecast procedures is normal for future work. It is clear that blame expectation is subject to skewed information. In any case, there is no proof of Fault forecast methods for huge information with constant and intelligent informational indexes in this SR audit and is normal for future work [2]

Wanjiang. Han et.al. [3] Author investigated the related innovations about classifiers and appropriation demonstrate. From the delegate gathered programming deserts information of GUI extends, the paper utilized a few classifier calculations to get deformity arrangement table, at that point connected numerical techniques to demonstrate that the circulation of this sort of programming venture absconds is reliable with the lognormal dissemination better. On the off chance that the dissemination of the product absconds obeyed as per the imperfections order is discovered then the utilization of blame infusion strategy to reenact programming

deficiency, and concentrate the quickened test technique under specific deformities appropriation, which can successfully enhance the product test scope, diminish test time, and decrease cost of test [3]

CHI-SQUARE TEST FOR DATA COLLECTED

Chi-Square Test to check Frequencies

Please Indicate the Type of Company

	Observed N	Expected N	Residual
Micro Enterprises (Investment does not Exceed 10 Lakhs)	139	123.3	15.7
Small Enterprises (Investment Between 10 Lakhs and 2 Crores)	60	123.3	-63.3
Medium Enterprises (Investment Between 2 Crores and 5 Crores)	171	123.3	47.7
Total	370		

Please Indicate Approximate Number of Employees

	Observed N	Expected N	Residual
1 - 10	39	92.5	-53.5
11 - 50	201	92.5	108.5
51 - 100	76	92.5	-16.5
101 - 500	54	92.5	-38.5
Total	370		

Please indicate the number of working shift per day

	Observed N	Expected N	Residual
1	66	123.3	-57.3
2	202	123.3	78.7
3	102	123.3	-21.3
Total	370		

Please Indicate Approximate Annual Turnover of Your Company

	Observed N	Expected N	Residual
Less than 5 Lakhs	98	74.0	24.0
5-10 Lakhs	81	74.0	7.0
11-20 Lakhs	54	74.0	-20.0
21 – 50 Lakhs	82	74.0	8.0
More than 50 Lakhs	55	74.0	-19.0
Total	370		

Please Indicate Average Annual Growth in turnover

	Observed N	Expected N	Residual
5 - 10%	56	92.5	-36.5
10 - 15%	138	92.5	45.5
15 - 20%	137	92.5	44.5
MORE THAN 20%	39	92.5	-53.5
Total	370		

Please Indicate Average Percentage of Your Turnover to Export Market

	Observed N	Expected N	Residual
0 - 5%	173	74.0	99.0
5 - 10%	96	74.0	22.0
10 - 15%	54	74.0	-20.0
15 - 20%	24	74.0	-50.0
More than 20%	23	74.0	-51.0
Total	370		

Please Indicate Average Annual Growth in Market Share

	Observed N	Expected N	Residual
0 - 5%	133	92.5	40.5
5 - 10%	135	92.5	42.5
10 - 15%	79	92.5	-13.5
More than 20%	23	92.5	-69.5
Total	370		

**Please Indicate Average Number of New Products Developed in Every Two Years
by the Company**

	Observed N	Expected N	Residual
1	75	74.0	1.0
2	30	74.0	-44.0
3	94	74.0	20.0
4	23	74.0	-51.0
More than 4	148	74.0	74.0
Total	370		

RESEARCH DESIGN

The focus of study is to understand why and what makes top performing companies different and what are the innovative and unique strategies in software defect prediction in creating a global standard organization. It was therefore, decided to use explorative and descriptive design, which befits into the pattern of investigation. I will focus on the current practices of developing and managing software defect prediction, explore, describe, analyze through appropriate questionnaire and schedule, including verbatim recording of the responses, per se.

DATA COLLECTION

An exhaustive empowerment questionnaire will be put to test. Software Defect prediction angles will be posed, valid and relevant on five point scale, viz: Strongly Agree; Agree; Can't Say; Disagree; and Strongly Disagree. Such areas that are put to test includes understanding the system of accountability within the organization, communication process adopted, decision making process, delegation and shared responsibility, feed-back system adopted, methods for information sharing, leadership development at all spheres, organizational transparency processes followed in SDLC and the like will be put to test.

- a) Pre-pilot observation
- b) Pre-pilot study and
- c) Final observation.

CONCLUSION

From the result it shows significance of CHI-SQUARE test being carried out for various aspects of software defect prediction being carried out which shows Observed Value, Expected Value and Residual.

REFERENCES

- 1) Er.Rohit Mahajan, Dr. Sunil Kumar Gupta, Rajeev Kumar Bedi, COMPARISON OF VARIOUS APPROACHES OF SOFTWARE FAULT PREDICTION: A REVIEW International Journal of Advanced Technology & Engineering Research (IJATER) www.ijater.com ISSN No: 2250-3536 Volume 4, Issue 4, July 2014
- 2) Pradeep Kumar Singh, Ranjan Kumar Panda and Om Prakash Sangwan , A Critical Analysis on Software Fault Prediction Techniques World Applied Sciences Journal 33 (3): 371-379, 2015 ISSN 1818-4952
- 3) Wanjiang. Han , Lixin. Jiang Tianbo. Lu, Xiaoyan &. Zhang, Sun Yi , Study on Residual Defect Prediction using Multiple Technologies , JOURNAL OF ADVANCES IN INFORMATION TECHNOLOGY, VOL. 5, NO. 3, AUGUST 2014