

A Critical Analysis of Software Defects Ability to Understand using of Software Testing

Samundra Singh¹ and Dr. Shivani²

Research Scholar, Department of CS¹

Assistant Professor, Department of CS²

Bhagwant University, Ajmer, India

Abstract: *Remaining defects are the cause of significant issues in the software development industry and not fixing them sooner increases the risk of negative effects such as system crashes, customer dissatisfaction and higher costs. Despite these observations, companies do not always remove newly found defects due to varying factors. These factors, which lead to longer term defects, are not well studied and therefore require more research.*

Keywords: Cause, Defect, Software, Testing, and Effect etc

BIBLIOGRAPHY

- [1]. Pressman, R. S. (n.d.). Software engineering Pressman, R. S. (n.d.). Software engineering (2nd ed.). New York: McGraw-Hill Book Company.
- [2]. Chauhan, R. K. & Singh, I. (2014). Latest Research and Development on Software Testing Techniques and Tools, 4(4), 2368–2372.
- [3]. Wala, T. & Sharma, A. K. (2014). Improvised Software Testing Tool, 3(9), 573–581.
- [4]. Merina, C. (2019). Tool Usability Parameter in Determining the Performance of Software Testing Tool, IJTB (International J. Technol. Business), 3(1), 8–18.
- [5]. Tassej, G. (2002). “The Economic Impacts of Inadequate infrastructure for software testing”. Natl. Inst. Stand. Technol.
- [6]. Raulamo-Jurvanen, P., Mäntylä, M. & Garousi, V. (2017). Choosing the Right Test Automation Tool, in 21st International Conference on Evaluation and Assessment in Software Engineering (EASE 2017), 21–30.
- [7]. Jain, V. & Rajnish, K. (2018). Comparative Study of Software Automation Testing Tools: OpenScript and Selenium, Int. J. Eng. Res. Appl., 8(2), 29–33.
- [8]. E. Arisholm, L. C. Briand, E. B. Johannessen, A systematic and comprehensive investigation of methods to build and evaluate fault prediction models, Journal of Systems and Software 83 (1) (2010) 2–17.
- [9]. R. Malhotra, A systematic review of machine learning techniques for software fault prediction, Applied Soft Computing 27 (C) (2015) 504–518.
- [10]. G. Denaro, L. Lavazza, M. Pez, An empirical evaluation of object oriented metrics in industrial setting, Journal of Object Technology Vol.
- [11]. M. A. D. Almeida, S. Matwin, Machine learning method for software quality model building, in: International Symposium on Foundations of Intelligent Systems, 1999, pp. 565–573.
- [12]. Z. Sun, Q. Song, X. Zhu, Using coding-based ensemble learning to improve software defect prediction, IEEE Transactions on Systems Man and Cybernetics Part C 42 (6) (2012) 1806–1817.
- [13]. T. Menzies, J. Distefano, A. Orrego, M. Chapman, Assessing predictors of software defects, Proceedings Workshop on Predictive Software Models.
- [14]. S. Lessmann, B. Baesens, C. Mues, S. Pietsch, Benchmarking classification models for software defect prediction: A proposed framework and novel findings, IEEE Transactions on Software Engineering 34 (4) (2008) 485–496.
- [15]. T. Zimmermann, R. Premraj, A. Zeller, Predicting defects for eclipse, in: International Workshop on Predictor MODELS in Software Engineering, 2007. Promise’07: ICSE Workshops, 2007, pp. 9–9.

- [16]. T. J. Ostrand, E. J. Weyuker, R. M. Bell, Predicting the location and number of faults in large software systems, *IEEE Transactions on Software Engineering* 31 (4) (2005) 340–355.
- [17]. X. Yang, K. Tang, X. Yao, A learning-to-rank approach to software defect prediction, *IEEE Transactions on Reliability* 64 (1) (2015) 234–246.
- [18]. B. A. Kitchenham, E. Mendes, G. H. Travassos, Cross versus withincompany cost estimation studies: A systematic review, *IEEE Transactions on Software Engineering* 33 (5) (2007) 316–329.
- [19]. Z. He, F. Shu, Y. Yang, M. Li, Q. Wang, An investigation on the feasibility of cross-project defect prediction, *Automated Software Engineering* 19 (2) (2012) 167–199.
- [20]. S. Herbold, Training data selection for cross-project defect prediction, in: *International Conference on Predictive MODELS in Software Engineering*, 2013, pp. 1–10.
- [21]. Y. Ma, G. Luo, X. Zeng, A. Chen, Transfer learning for cross-company software defect prediction, *Information and Software Technology* 54 (3) (2012) 248–256.
- [22]. B. Turhan, T. Menzies, A. B. Bener, J. D. Stefano, On the relative value of cross-company and within-company data for defect prediction, *Empirical Software Engineering* 14 (5) (2009) 540–578.
- [23]. T. Zimmermann, N. Nagappan, H. Gall, E. Giger, B. Murphy, Crossproject defect prediction: a large scale experiment on data vs. domain vs. process, in: *Joint Meeting of the European Software Engineering Conference and the ACM Sigsoft International Symposium on Foundations of Software Engineering*, 2009, pp. 91–100.
- [24]. E. Kocaguneli, G. Gay, T. Menzies, Y. Yang, J. W. Keung, When to use data from other projects for effort estimation, in: *International Conference on Automated Software Engineering*, 2010, pp. 321–324