1. REFERENCES

p[1] Moser, R. et al. 2008. A comparative analysis of the efficiency of change metrics and static code attributes for defect prediction. *Software Engineering, 2008. ICSE '08. ACM/IEEE 30th International Conference on* (May. 2008), 181–190

p[2] Shihab, E. et al. 2012. An Industrial Study on the Risk of Software Changes. *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering* (Cary, North Carolina, 2012), 62:1–62:11.

p[3] Matsumoto, S. et al. 2010. An Analysis of Developer Metrics for Fault Prediction. Proceedings of the 6th International Conference on Predictive Models in Software Engineering (Timisoara, Romania, 2010), 18:1–18:9.

p[5] Nagappan, N. et al. 2010. Change Bursts as Defect Predictors. *Software Reliability Engineering (ISSRE), 2010 IEEE 21st International Symposium on* (Nov. 2010), 309–318.

p[6] Zimmermann, T. et al. 2012. Characterizing and predicting which bugs get reopened. Software Engineering (ICSE), 2012 34th International Conference on (Jun. 2012), 1074–1083.

p[7] Biçer, S. et al. 2011. Defect Prediction Using Social Network Analysis on Issue Repositories. Proceedings of the 2011 International Conference on Software and Systems Process (Waikiki, Honolulu, HI, USA, 2011), 63–71.

p[8] Kwan, I. et al. 2011. Does Socio-Technical Congruence Have an Effect on Software Build Success? A Study of Coordination in a Software Project. IEEE Trans. Softw. Eng. 37, (May. 2011), 307–324

p[9] Pinzger, M. et al. 2008. Can Developer-module Networks Predict Failures? Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering (Atlanta, Georgia, 2008), 2–12.

p[10] Posnett, D. et al. 2011. Ecological inference in empirical software engineering. Automated Software Engineering (ASE), 2011 26th IEEE/ACM International Conference on (Nov. 2011), 362–371.

p[12] Meneely, A. and Williams, O. 2012. Interactive Churn Metrics: Socio-technical Variants of Code Churn. SIGSOFT Softw. Eng. Notes. 37, (Nov. 2012), 1–6.

p[13] Wolf, T. et al. 2009. Predicting build failures using social network analysis on developer communication. Software Engineering, 2009. ICSE 2009. IEEE 31st International Conference on (May. 2009), 1–11.

p[15] Meneely, A. et al. 2008. Predicting Failures with Developer Networks and Social Network Analysis. Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering (Atlanta, Georgia, 2008), 13–23.

p[16] Bird, C. et al. 2009. Putting It All Together: Using Socio-technical Networks to Predict Failures. Proceedings of the 2009 20th International Symposium on Software Reliability Engineering (Washington, DC, USA, 2009), 109–119.

p[17] Bettenburg, N. and Hassan, A. 2013. Studying the impact of social interactions on software quality. Empirical Software Engineering (2013), 375–431.

p[18] Mockus, A. and Weiss, D.M. 2000. Predicting risk of software changes. *Bell Labs Technical Journal* (2000), 169–180.

p[19] Meneely, A. and Williams, L. 2009. Secure Open Source Collaboration: An Empirical Study of Linus' Law. Proceedings of the 16th ACM Conference on Computer and Communications Security (Chicago, Illinois, USA, 2009), 453–462.

p[20] Rahman, F. and Devanbu, P. 2013. How, and Why, Process Metrics Are Better. Proceedings of the 2013 International Conference on Software Engineering (San Francisco, CA, USA, 2013), 432–441.

p[21] Ibrahim, W.M. et al. 2010. Should I contribute to this discussion? Mining Software Repositories (MSR), 2010 7th IEEE Working Conference on (May. 2010), 181–190.

p[22] Nagappan, N. et al. 2008. The Influence of Organizational Structure on Software Quality: An Empirical Case Study. Proceedings of the 30th International Conference on Software Engineering (Leipzig, Germany, 2008), 521–530.

p[24] Weyuker, E.J. et al. 2007. Using Developer Information as a Factor for Fault Prediction. Predictor Models in Software Engineering, 2007. PROMISE'07: ICSE Workshops 2007. International Workshop on (May. 2007), 8–8.

p[26] D'Ambros, M. et al. 2012. Evaluating Defect Prediction Approaches: A Benchmark and an Extensive Comparison. Empirical Softw. Engg. 17, (Aug. 2012), 531–577.

p[27] Bird, C. et al. 2011. Don'T Touch My Code!: Examining the Effects of Ownership on Software Quality. Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering (Szeged, Hungary, 2011), 4–14.

p[28] Posnett, D. et al. 2013. Dual Ecological Measures of Focus in Software Development. Proceedings of the 2013 International Conference on Software Engineering (San Francisco, CA, USA, 2013), 452–461.

p[29] Graves, T.L. et al. 2000. Predicting fault incidence using software change history. Software Engineering, IEEE Transactions on. 26, (Jul. 2000), 653–661.

p[30] Ratzinger, J. et al. 2007. EQ-mine: Predicting Short-term Defects for Software Evolution. Proceedings of the 10th International Conference on Fundamental Approaches to *Software Engineering* (Braga, Portugal, 2007), 12–26.

p[32] Mockus, A. 2010. Organizational Volatility and Its Effects on Software Defects. Proceedings of the Eighteenth ACM SIGSOFT International Symposium on Foundations of Software Engineering (Santa Fe, New Mexico, USA, 2010), 117–126.

p[33] Weyuker, E.J. et al. 2008. Do Too Many Cooks Spoil the Broth? Using the Number of Developers to Enhance Defect Prediction Models. Empirical Softw. Engg. 13, (Oct. 2008), 539–559.

p[34] Caglayan, B. et al. 2009. Merits of using repository metrics in defect prediction for open source projects. Emerging Trends in Free/Libre/Open Source Software Research and Development, 2009. FLOSS '09. ICSE Workshop on (May. 2009), 31–36.

p[35] Bernstein, A. et al. 2007. Improving Defect Prediction Using Temporal Features and Non Linear Models. Ninth International Workshop on Principles of Software Evolution: In Conjunction with the 6th ESEC/FSE Joint Meeting (Dubrovnik, Croatia, 2007), 11–18.

p[36] Shin, Y. et al. 2011. Evaluating Complexity, Code Churn, and Developer Activity Metrics as Indicators of Software Vulnerabilities. Software Engineering, IEEE Transactions on. 37, (Nov. 2011), 772–787.

p[37] Ekanayake, J. et al. 2012. Time Variance and Defect Prediction in Software Projects. Empirical Softw. Engg. 17, (Aug. 2012), 348–389.

p[38] Nagappan, N. et al. 2006. Using Historical In-Process and Product Metrics for Early Estimation of Software Failures. Software Reliability Engineering, 2006. ISSRE '06. 17th International Symposium on (Nov. 2006), 62–74.

p[39] Bacchelli, A. et al. 2010. Are Popular Classes More Defect Prone? Proceedings of the 13th International Conference on Fundamental Approaches to Software Engineering (Paphos, Cyprus, 2010), 59–73.

p[40] Alhassan, S. et al. 2010. Do More People Make the Code More Defect Prone?: Social Network Analysis in OSS Projects. Proceedings of the 22nd International Conference on Software Engineering & Knowledge Engineering (SEKE 2010), Redwood City, San Francisco Bay, CA, USA, July 1 - July 3, 2010 (2010), 93–98.

p[41] Bell, R.M. et al. 2011. Does Measuring Code Change Improve Fault Prediction? Proceedings of the 7th International Conference on Predictive Models in Software Engineering (Banff, Alberta, Canada, 2011), 2:1–2:8.

p[42] Bell, R. et al. 2013. The limited impact of individual developer data on software defect prediction. Empirical Software Engineering. 18, (2013), 478–505.

p[43] Cataldo, M. et al. 2009. Software Dependencies, Work Dependencies, and Their Impact on Failures. Software Engineering, IEEE Transactions on. 35, (Nov. 2009), 864–878.

p[44] Lee, T. et al. 2011. Micro Interaction Metrics for Defect Prediction. Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering (Szeged, Hungary, 2011), 311–321.

p[45] Zanetti, M.S. et al. 2013. Categorizing Bugs with Social Networks: A Case Study on Four Open Source Software Communities. Proceedings of the 2013 International Conference on Software Engineering (San Francisco, CA, USA, 2013), 1032–1041.

p[46] Zhang, H. et al. 2013. Predicting bug-fixing time: An empirical study of commercial software projects. Software Engineering (ICSE), 2013 35th International Conference on (May. 2013), 1042–1051.

p[47] Caglayan, B. et al. 2012. Factors Characterizing Reopened Issues: A Case Study. Proceedings of the 8th International Conference on Predictive Models in Software Engineering (Lund, Sweden, 2012), 1–10.

p[48] Duc Anh, N. et al. 2011. Empirical Validation of Human Factors in Predicting Issue Lead Time in Open Source Projects. Proceedings of the 7th International Conference on Predictive Models in Software Engineering (Banff, Alberta, Canada, 2011), 13:1–13:10.

p[49] Xuan, J. et al. 2012. Developer prioritization in bug repositories. Software Engineering (ICSE), 2012 34th International Conference on (Jun. 2012), 25–35.

p[50] Datta, S. et al. 2012. Talk Versus Work: Characteristics of Developer Collaboration on the Jazz Platform. Proceedings of the ACM International Conference on Object Oriented Programming Systems Languages and Applications (Tucson, Arizona, USA, 2012), 655–668.

p[51] Bhattacharya, P. et al. 2012. Graph-based analysis and prediction for software evolution. Software Engineering (ICSE), 2012 34th International Conference on (Jun. 2012), 419–429.

p[52] Karus, S. and Dumas, M. 2012. Code Churn Estimation Using Organisational and Code Metrics: An Experimental Comparison. Inf. Softw. Technol. 54, (Feb. 2012), 203–211.

p[53] Tosun Misirli, A. et al. 2011. An Explanatory Analysis on Eclipse Beta-release Bugs Through In-process Metrics. Proceedings of the 8th International Workshop on Software Quality (Szeged, Hungary, 2011), 26–33.

p[54]Illes-Seifert, T. and Paech, B. 2010. Exploring the relationship of a file´s history and its fault-proneness: An empirical method and its application to open source programs. Information and Software Technology. 52, (2010), 539–558.