



Figure 3: An example of how we identify source code properties using constructivist grounded theory.

and the description behind the identified properties. We use Figure 3 to explain how we use the three grounded theory elements to identify a property. We first start with defect-related XCMs and code changes from defect-related commits, to derive concepts. According to Figure 3, from the defect-related XCM 'fix file location for interfaces change-id i0b3c40157', we extract the concept 'fix file location'. Next, we generate categories from the concepts, for example, we use the concept 'fix file location' to determine the category which states an erroneous file location might need fixing. We use three concepts to derive category 'Path to external file or script needs fixing'. Finally, we use the categories 'File location needs fixing' and 'Path to external file or script needs fixing' to derive a proposition related to file location. This proposition gives us a property 'File' and the description behind that property is 'Scripts that set file paths can be defect-prone'.

Upon completion of constructivist grounded theory, we obtain a set of source

IEEE Transactions on Software Engineering 38 (6) (2012) 1276–1304.
doi:10.1109/TSE.2011.103.

- [49] F. Rahman, P. Devanbu, How, and why, process metrics are better, in: Proceedings of the 2013 International Conference on Software Engineering, ICSE '13, IEEE Press, Piscataway, NJ, USA, 2013, pp. 432–441.
URL <http://dl.acm.org/citation.cfm?id=2486788.2486846>
- [50] B. Ghotra, S. McIntosh, A. E. Hassan, Revisiting the impact of classification techniques on the performance of defect prediction models, in: Proceedings of the 37th International Conference on Software Engineering - Volume 1, ICSE '15, IEEE Press, Piscataway, NJ, USA, 2015, pp. 789–800.
URL <http://dl.acm.org/citation.cfm?id=2818754.2818850>
- [51] S. Lessmann, B. Baesens, C. Mues, S. Pietsch, Benchmarking classification models for software defect prediction: A proposed framework and novel findings, IEEE Trans. Softw. Eng. 34 (4) (2008) 485–496.
doi:10.1109/TSE.2008.35.
URL <http://dx.doi.org/10.1109/TSE.2008.35>
- [52] T. Menzies, A. Dekhtyar, J. Distefano, J. Greenwald, Problems with precision: A response to "comments on 'data mining static code attributes to learn defect predictors'", IEEE Trans. Softw. Eng. 33 (9) (2007) 637–640.
doi:10.1109/TSE.2007.70721.
URL <http://dx.doi.org/10.1109/TSE.2007.70721>
- [53] C. Tantithamthavorn, S. McIntosh, A. E. Hassan, K. Matsumoto, An empirical comparison of model validation techniques for defect

prediction models, *IEEE Trans. Softw. Eng.* 43 (1) (2017) 1–18.
doi:10.1109/TSE.2016.2584050.

URL <https://doi.org/10.1109/TSE.2016.2584050>

[54] J. Walden, J. Stuckman, R. Scandariato, Predicting vulnerable components: Software metrics vs text mining, in: 2014 IEEE 25th International Symposium on Software Reliability Engineering, 2014, pp. 23–33.
doi:10.1109/ISSRE.2014.32.

[55] M. F. Porter, *Readings in information retrieval*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997, Ch. An Algorithm for Suffix Striping, pp. 313–316.

URL <http://dl.acm.org/citation.cfm?id=275537.275705>

[56] A. Meneely, L. Williams, Secure open source collaboration: An empirical study of linus' law, in: *Proceedings of the 16th ACM Conference on Computer and Communications Security, CCS '09*, ACM, New York, NY, USA, 2009, pp. 453–462. doi:10.1145/1653662.1653717.

URL <http://doi.acm.org/10.1145/1653662.1653717>

[57] T. Fukushima, Y. Kamei, S. McIntosh, K. Yamashita, N. Ubayashi, An empirical study of just-in-time defect prediction using cross-project models, in: *Proceedings of the 11th Working Conference on Mining Software Repositories, MSR 2014*, ACM, New York, NY, USA, 2014, pp. 172–181.
doi:10.1145/2597073.2597075.

URL <http://doi.acm.org/10.1145/2597073.2597075>

[58] A. Tosun Misirli, B. Murphy, T. Zimmermann, A. Basar Bener, An ex-

planatory analysis on eclipse beta-release bugs through in-process metrics, in: Proceedings of the 8th International Workshop on Software Quality, WoSQ '11, ACM, New York, NY, USA, 2011, pp. 26–33. doi:10.1145/2024587.2024595.

URL <http://doi.acm.org/10.1145/2024587.2024595>

[59] R. Chillarege, I. S. Bhandari, J. K. Chaar, M. J. Halliday, D. S. Moebus, B. K. Ray, M. Y. Wong, Orthogonal defect classification—a concept for in-process measurements, *IEEE Transactions on Software Engineering* 18 (11) (1992) 943–956. doi:10.1109/32.177364.

[60] C. Lewis, Z. Lin, C. Sadowski, X. Zhu, R. Ou, E. J. Whitehead Jr., Does bug prediction support human developers? findings from a google case study, in: Proceedings of the 2013 International Conference on Software Engineering, ICSE '13, IEEE Press, Piscataway, NJ, USA, 2013, pp. 372–381.

URL <http://dl.acm.org/citation.cfm?id=2486788.2486838>