References

[1] Abad, Z. S. H., Karras, O., Schneider, K., Barker, K., and Bauer, M. Task interruption in software development projects. In *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018* (June 2018), ACM.

[2] Abdalkareem, R., Nourry, O., Wehaibi, S., Mujahid, S., and Shihab, E. Why do developers use trivial packages? an empirical case study on npm. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (Aug. 2017), ACM.

[3] Aghajani, E., Nagy, C., Vega-Marquez, O. L., Linares-Vasquez, M., Moreno, L., Bavota, G., and Lanza, M. Software documentation issues unveiled. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)* (May 2019), IEEE.

[4] Ajami, S., Woodbridge, Y., and Feitelson, D. G. Syntax, predicates, idioms—what really affects code complexity? *Empirical Software Engineering 24*, 1 (June 2018), 287–328.

[5] Åkerblom, B., and Wrigstad, T. Measuring polymorphism in python programs. *ACM SIGPLAN Notices 51*, 2 (May 2016), 114–128.

[6] Al-Subaihin, A. A., Sarro, F., Black, S., Capra, L., and Harman, M. App store effects on software engineering practices. *IEEE Transactions on Software Engineering 47*, 2 (Feb. 2021), 300–319.

[7] Alencar da Costa, D., McIntosh, S., Treude, C., Kulesza, U., and Hassan, A. E. The impact of rapid release cycles on the integration delay of fixed issues. *Empirical Software Engineering 23*, 2 (Nov. 2017), 835–904.

[8] Ali, R. H., Parlett-Pelleriti, C., and Linstead, E. Cheating death: A statistical survival analysis of publicly available python projects. In *Proceedings of the 17th International Conference on Mining Software Repositories* (June 2020), ACM.

[9] Almeida, D. A., Murphy, G. C., Wilson, G., and Hoye, M. Do software developers understand open source licenses? In *2017 IEEE/ACM 25th International Conference on Program Comprehension (ICPC)* (May 2017), IEEE.

[10] Altadmri, A., and Brown, N. C. 37 million compilations: Investigating novice programming mistakes in large-scale student data. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (Feb. 2015), ACM.

[11] Ameller, D., Ayala, C., Cabot, J., and Franch, X. How do software architects consider non-functional requirements: An exploratory study. In *2012 20th IEEE International Requirements Engineering Conference (RE)* (Sept. 2012), IEEE.

[12] Ames, M. G. Hackers, computers, and cooperation: A critical history of logo and constructionist learning. *Proceedings of the ACM on Human-Computer Interaction 2*, CSCW (Nov. 2018), 1–19.

[13] Anda, B., Sjøberg, D., and Mockus, A. Variability and reproducibility in software engineering: A study of four companies that developed the same system. *IEEE Transactions on Software Engineering 35*, 3 (May 2009), 407–429.

[14] Apel, S., Liebig, J., Brandl, B., Lengauer, C., and Kästner, C. Semistructured merge. In *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering - SIGSOFT/FSE '11* (2011), ACM Press.

[15] Bafatakis, N., Boecker, N., Boon, W., Salazar, M. C., Krinke, J., Oznacar, G., and White, R. Python coding style compliance on stack overflow. In *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)* (May 2019), IEEE.

[16] Balachandran, V. Reducing human effort and improving quality in peer code reviews using automatic static analysis and reviewer recommendation. In *2013 35th International Conference on Software Engineering (ICSE)* (May 2013), IEEE.

[17] Balali, S., Steinmacher, I., Annamalai, U., Sarma, A., and Gerosa, M. A. Newcomers' barriers... is that all? an analysis of mentors' and newcomers' barriers in OSS projects. *Computer Supported Cooperative Work (CSCW) 27*, 3-6 (Apr. 2018), 679–714.

[18] Baltes, S., Park, G., and Serebrenik, A. Is 40 the new 60? how popular media portrays the employability of older software developers. *IEEE Software 37*, 6 (Nov. 2020), 26–31.

[19] Bao, L., Xia, X., Lo, D., and Murphy, G. C. A large scale study of long-time contributor prediction for GitHub projects. *IEEE Transactions on Software Engineering 47*, 6 (June 2021), 1277–1298.

[20] Barik, T., Smith, J., Lubick, K., Holmes, E., Feng, J., Murphy-Hill, E., and Parnin, C. Do developers read compiler error messages? In *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)* (May 2017), IEEE.

[21] Barke, H., and Prechelt, L. Role clarity deficiencies can wreck agile teams. *PeerJ Computer Science 5* (Dec. 2019), e241.

[22] Barnett, M., Fähndrich, M., Leino, K. R. M., Müller, P., Schulte, W., and Venter, H. Specification and verification: the spec# experience. *Communications of the ACM 54*, 6 (June 2011), 81–91.

[23] Barr, E. T., Bird, C., Rigby, P. C., Hindle, A., German, D. M., and Devanbu, P. Cohesive and isolated development with branches. In *Proceedings of the 15th international conference on Fundamental Approaches to Software Engineering* (2012), Springer Berlin Heidelberg, pp. 316–331.

[24] Barzilay, O. Example embedding. In *Proceedings of the 10th SIGPLAN symposium on New ideas, new paradigms, and reflections on programming and software - ONWARD '11* (2011), ACM Press.

[25] Beck, F., and Diehl, S. On the congruence of modularity and code coupling. In *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering - SIGSOFT/FSE '11* (2011), ACM Press.

[26] Becker, B. A., Denny, P., Pettit, R., Bouchard, D., Bouvier, D. J., Harrington, B., Kamil, A., Karkare, A., McDonald, C., Osera, P.-M., Pearce, J. L., and Prather, J. Compiler error messages considered unhelpful. In *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education* (Dec. 2019), ACM.

[27] Behroozi, M., Parnin, C., and Barik, T. Hiring is broken: What do developers say about technical interviews? In *2019 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (Oct. 2019), IEEE.

[28] Behroozi, M., Shirolkar, S., Barik, T., and Parnin, C. Debugging hiring: What went right and what went wrong in the technical interview process. In *International Conference on Software Engineering (ICSE 2020)* (2020), ACM.

[29] Beller, M., Gousios, G., Panichella, A., Proksch, S., Amann, S., and Zaidman, A. Developer testing in the IDE: Patterns, beliefs, and behavior. *IEEE Transactions on Software Engineering 45*, 3 (Mar. 2019), 261–284.

[30] Beller, M., Gousios, G., Panichella, A., and Zaidman, A. When, how, and why developers (do not) test in their IDEs. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering* (Aug. 2015), ACM.

[31] Ben-Ari, M., Bednarik, R., Levy, R. B.-B., Ebel, G., Moreno, A., Myller, N., and Sutinen, E. A decade of research and development on program animation: The jeliot experience. *Journal of Visual Languages & Computing 22*, 5 (Oct. 2011), 375–384.

[32] Bettenburg, N., Just, S., Schröter, A., Weiss, C., Premraj, R., and Zimmermann, T. What makes a good bug report? In *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering - SIGSOFT '08/FSE-16* (2008), ACM Press.

[33] Bird, C., Nagappan, N., Murphy, B., Gall, H., and Devanbu, P. Don't touch my code!: examining the effects of ownership on software quality. In *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering - SIGSOFT/FSE '11* (2011), ACM Press.

[34] Blackwell, A. F., Petre, M., and Church, L. Fifty years of the psychology of programming. *International Journal of Human-Computer Studies 131* (Nov. 2019), 52–63.

[35] Bluedorn, A. C., Turban, D. B., and Love, M. S. The effects of stand-up and sit-down meeting formats on meeting outcomes. *Journal of Applied Psychology 84*, 2 (1999), 277–285.

[36] Borle, N. C., Feghhi, M., Stroulia, E., Greiner, R., and Hindle, A. Analyzing the effects of test driven development in GitHub. *Empirical Software Engineering 23*, 4 (Nov. 2017), 1931–1958.

[37] Brown, C., and Parnin, C. Understanding the impact of GitHub suggested changes on recommendations between developers. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (Nov. 2020), ACM.

[38] Brown, N. C. C., Altadmri, A., Sentance, S., and Kölling, M. Blackbox, five years on: An evaluation of a large-scale programming data collection project. In *Proceedings of the 2018 ACM Conference on International Computing Education Research* (Aug. 2018), ACM.

[39] Brun, Y., Holmes, R., Ernst, M. D., and Notkin, D. Proactive detection of collaboration conflicts. In *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering - SIGSOFT/FSE '11* (2011), ACM Press.

[40] Butler, S., Gamalielsson, J., Lundell, B., Brax, C., Sjoberg, J., Mattsson, A., Gustavsson, T., Feist, J., and Lonroth, E. On company contributions to community open source software projects. *IEEE Transactions on Software Engineering* (2019), 1–1.

[41] Campos, E. C., and de Almeida Maia, M. Common bug-fix patterns: A large-scale observational study. In *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)* (Nov. 2017), IEEE.

[42] Catolino, G., Palomba, F., Tamburri, D. A., Serebrenik, A., and Ferrucci, F. Gender diversity and women in software teams: How do they affect community smells? In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)* (May 2019), IEEE.

[43] Chattopadhyay, S., Nelson, N., Au, A., Morales, N., Sanchez, C., Pandita, R., and Sarma, A. A tale from the trenches: cognitive biases and software development. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering* (June 2020), ACM.

[44] Chen, T.-H., Shang, W., Yang, J., Hassan, A. E., Godfrey, M. W., Nasser, M., and Flora, P. An empirical study on the practice of maintaining object-relational mapping code in java systems. In *Proceedings of the 13th International Conference on Mining Software Repositories* (May 2016), ACM.

[45] Cherubini, M., Venolia, G., DeLine, R., and Ko, A. J. Let's go to the whiteboard: how and why software developers use drawings. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Apr. 2007), ACM.

[46] Chong, J., and Hurlbutt, T. The social dynamics of pair programming. In *29th International Conference on Software Engineering (ICSE'07)* (May 2007), IEEE.

[47] Cinnéide, M. Ó., Tratt, L., Harman, M., Counsell, S., and Moghadam, I. H. Experimental assessment of software metrics using automated refactoring. In *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement - ESEM '12* (2012), ACM Press.

[48] Cogo, F. R., Oliva, G. A., Bezemer, C.-P., and Hassan, A. E. An empirical study of same-day releases of popular packages in the npm ecosystem. *Empirical Software Engineering 26*, 5 (July 2021).

[49] Costa, D. E. D., Bezemer, C.-P., Leitner, P., and Andrzejak, A. What's wrong with my benchmark results? studying bad practices in JMH benchmarks. *IEEE Transactions on Software Engineering* (2019), 1–1.

[50] Cruz-Lemus, J. A., Genero, M., Manso, M. E., Morasca, S., and Piattini, M. Assessing the understandability of UML statechart diagrams with composite states—a family of empirical studies. *Empirical Software Engineering 14*, 6 (Feb. 2009), 685–719.

[51] Dabbish, L., Stuart, C., Tsay, J., and Herbsleb, J. Social coding in GitHub: transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work - CSCW '12* (2012), ACM Press.

[52] Dagenais, B., and Robillard, M. P. Creating and evolving developer documentation. In *Proceedings of the eighteenth ACM SIGSOFT international symposium on Foundations of software engineering - FSE '10* (2010), ACM Press.

[53] Dang, Y., Wu, R., Zhang, H., Zhang, D., and Nobel, P. ReBucket: A method for clustering duplicate crash reports based on call stack similarity. In *2012 34th International Conference on Software Engineering (ICSE)* (June 2012), IEEE.

[54] de Oliveira Neto, F. G., Torkar, R., Feldt, R., Gren, L., Furia, C. A., and Huang, Z. Evolution of statistical analysis in empirical software engineering research: Current state and steps forward. *Journal of Systems and Software 156* (Oct. 2019), 246–267.

[55] de Paulo Sobrinho, E. V., Lucia, A. D., and de Almeida Maia, M. A systematic literature review on bad smells–5 w's: Which, when, what, who, where. *IEEE Transactions on Software Engineering 47*, 1 (Jan. 2021), 17–66.

[56] Decan, A., and Mens, T. What do package dependencies tell us about semantic versioning? *IEEE Transactions on Software Engineering 47*, 6 (June 2021), 1226–1240.

[57] Dias, E., Meirelles, P., Castor, F., Steinmacher, I., Wiese, I., and Pinto, G. What makes a great maintainer of open source projects? In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)* (May 2021), IEEE.

[58] Durieux, T., Goues, C. L., Hilton, M., and Abreu, R. Empirical study of restarted and flaky builds on travis CI. In *Proceedings of the 17th International Conference on Mining Software Repositories* (June 2020), ACM.

[59] Dzidek, W., Arisholm, E., and Briand, L. A realistic empirical evaluation of the costs and benefits of UML in software maintenance. *IEEE Transactions on Software Engineering 34*, 3 (May 2008), 407–432.

[60] Eichberg, M., Hermann, B., Mezini, M., and Glanz, L. Hidden truths in dead software paths. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering* (Aug. 2015), ACM.

[61] Emam, K. E., Benlarbi, S., Goel, N., and Rai, S. The confounding effect of class size on the validity of object-oriented metrics. *IEEE Transactions on Software Engineering 27*, 7 (July 2001), 630–650.

[62] Fagerholm, F., Kuhrmann, M., and Münch, J. Guidelines for using empirical studies in software engineering education. *PeerJ Computer Science 3* (Sept. 2017), e131.

[63] Ford, D., Behroozi, M., Serebrenik, A., and Parnin, C. Beyond the code itself: How programmers really look at pull requests. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)* (May 2019), IEEE.

[64] Ford, D., Smith, J., Guo, P. J., and Parnin, C. Paradise unplugged: identifying barriers for female participation on stack overflow. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering* (Nov. 2016), ACM.

[65] Ford, D., Zimmermann, T., Bird, C., and Nagappan, N. Characterizing software engineering work with personas based on knowledge worker actions. In *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)* (Nov. 2017), IEEE.

[66] Foundjem, A., and Adams, B. Release synchronization in software ecosystems. *Empirical Software Engineering 26*, 3 (Mar. 2021).

[67] Fucci, D., Scanniello, G., Romano, S., and Juristo, N. Need for sleep: The impact of a night of sleep deprivation on novice developers' performance. *IEEE Transactions on Software Engineering 46*, 1 (Jan. 2020), 1–19.

[68] Fucci, D., Scanniello, G., Romano, S., Shepperd, M., Sigweni, B., Uyaguari, F., Turhan, B., Juristo, N., and Oivo, M. An external replication on the effects of test-driven development using a multi-site blind analysis approach. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (Sept. 2016), ACM.

[69] Furia, C. A., Feldt, R., and Torkar, R. Bayesian data analysis in empirical software engineering research. *IEEE Transactions on Software Engineering* (2019), 1–1.

[70] Gao, G., Voichick, F., Ichinco, M., and Kelleher, C. Exploring programmers' API learning processes: Collecting web resources as external memory. In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (Aug. 2020), IEEE.

[71] Gao, Z., Bird, C., and Barr, E. T. To type or not to type: Quantifying detectable bugs in JavaScript. In *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)* (May 2017), IEEE.

[72] García, B., Munoz-Organero, M., Alario-Hoyos, C., and Kloos, C. D. Automated driver management for selenium WebDriver. *Empirical Software Engineering 26*, 5 (July 2021).

[73] Gauthier, F., and Merlo, E. Semantic smells and errors in access control models: A case study in PHP. In *2013 35th International Conference on Software Engineering (ICSE)* (May 2013), IEEE.

[74] Gerosa, M., Wiese, I., Trinkenreich, B., Link, G., Robles, G., Treude, C., Steinmacher, I., and Sarma, A. The shifting sands of motivation: Revisiting what drives contributors in open source. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)* (May 2021), IEEE.

[75] Ghiotto, G., Murta, L., Barros, M., and van der Hoek, A. On the nature of merge conflicts: A study of 2,731 open source java projects hosted by GitHub. *IEEE Transactions on Software Engineering 46*, 8 (Aug. 2020), 892–915.

[76] Giger, E., Pinzger, M., and Gall, H. Using the gini coefficient for bug prediction in eclipse. In *Proceedings of the 12th international workshop and the 7th annual ERCIM workshop on Principles on software evolution and software evolution - IWPSE-EVOL '11* (2011), ACM Press.

[77] Gousios, G., Storey, M.-A., and Bacchelli, A. Work practices and challenges in pull-based development. In *Proceedings of the 38th International Conference on Software Engineering* (May 2016), ACM.

[78] Graziotin, D., Wang, X., and Abrahamsson, P. Happy software developers solve problems better: psychological measurements in empirical software engineering. *PeerJ 2* (Mar. 2014), e289.

[79] Green, T. R. G., and Petre, M. Usability analysis of visual programming environments: A 'cognitive dimensions' framework. *Journal of Visual Languages & Computing 7*, 2 (June 1996), 131–174.

[80] Gulzar, M. A., Interlandi, M., Yoo, S., Tetali, S. D., Condie, T., Millstein, T., and Kim, M. BigDebug: debugging primitives for interactive big data processing in spark. In *Proceedings of the 38th International Conference on Software Engineering* (May 2016), ACM.

[81] Hanenberg, S. An experiment about static and dynamic type systems. In *Proceedings of the ACM international conference on Object oriented programming systems languages and applications - OOPSLA '10* (2010), ACM Press.

[82] Hannay, J., Arisholm, E., Engvik, H., and Sjøberg, D. Effects of personality on pair programming. *IEEE Transactions on Software Engineering 36*, 1 (Jan. 2010), 61–80.

[83] Harms, K. J., Chen, J., and Kelleher, C. L. Distractors in parsons problems decrease learning efficiency for young novice programmers. In *Proceedings of the 2016 ACM Conference on International Computing Education Research* (Aug. 2016), ACM.

[84] Hata, H., Treude, C., Kula, R. G., and Ishio, T. 9.6 million links in source code comments: Purpose, evolution, and decay. In

2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE) (May 2019), IEEE.

[85] Hayashi, J., Higo, Y., Matsumoto, S., and Kusumoto, S. Impacts of daylight saving time on software development. In *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)* (May 2019), IEEE.

[86] Hemmati, H., Nadi, S., Baysal, O., Kononenko, O., Wang, W., Holmes, R., and Godfrey, M. W. The MSR cookbook: Mining a decade of research. In *2013 10th Working Conference on Mining Software Repositories (MSR)* (May 2013), IEEE.

[87] Hermans, F., and Aivaloglou, E. Do code smells hamper novice programming? a controlled experiment on scratch programs. In *2016 IEEE 24th International Conference on Program Comprehension (ICPC)* (May 2016), IEEE.

[88] Hermans, F., Pinzger, M., and van Deursen, A. Supporting professional spreadsheet users by generating leveled dataflow diagrams. In *Proceedings of the 33rd International Conference on Software Engineering* (May 2011), ACM.

[89] Herzig, K., Just, S., and Zeller, A. It's not a bug, it's a feature: How misclassification impacts bug prediction. In *2013 35th International Conference on Software Engineering (ICSE)* (May 2013), IEEE.

[90] Hindle, A., Barr, E. T., Gabel, M., Su, Z., and Devanbu, P. On the naturalness of software. *Communications of the ACM 59*, 5 (Apr. 2016), 122–131.

[91] Hindle, A., Bird, C., Zimmermann, T., and Nagappan, N. Relating requirements to implementation via topic analysis: Do topics extracted from requirements make sense to managers and developers? In *2012 28th IEEE International Conference on Software Maintenance (ICSM)* (Sept. 2012), IEEE.

[92] Hora, A. Googling for software development: What developers search for and what they find. In *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)* (May 2021), IEEE.

[93] Hora, A. What code is deliberately excluded from test coverage and why? In *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)* (May 2021), IEEE.

[94] Hoyos, J., Abdalkareem, R., Mujahid, S., Shihab, E., and Bedoya, A. E. On the removal of feature toggles: A study of python projects and practitioners motivations. *Empirical Software Engineering 26*, 2 (Feb. 2021).

[95] Hundhausen, C. D., Agarwal, P., and Trevisan, M. Online vs. face-to-face pedagogical code reviews. In *Proceedings of the 42nd ACM technical symposium on Computer science education - SIGCSE '11* (2011), ACM Press.

[96] Jacobson, I., Ng, P.-W., McMahon, P. E., Spence, I., and Lidman, S. *The Essence of Software Engineering: Applying the SEMAT Kernel*. Addison-Wesley Professional, 2013.

[97] Jalote, P., and Kamma, D. Studying task processes for improving programmer productivity. *IEEE Transactions on Software Engineering 47*, 4 (Apr. 2021), 801–817.

[98] Johnson, B., Zimmermann, T., and Bird, C. The effect of work environments on productivity and satisfaction of software engineers. *IEEE Transactions on Software Engineering 47*, 4 (Apr. 2021), 736–757.

[99] Johnson, J., Lubo, S., Yedla, N., Aponte, J., and Sharif, B. An empirical study assessing source code readability in comprehension. In *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (Sept. 2019), IEEE.

[100] Jolak, R., Savary-Leblanc, M., Dalibor, M., Wortmann, A., Hebig, R., Vincur, J., Polasek, I., Pallec, X. L., Gérard, S., and Chaudron, M. R. V. Software engineering whispers: The effect of textual vs. graphical software design descriptions on software design communication. *Empirical Software Engineering 25*, 6 (Sept. 2020), 4427–4471.

[101] Jones, D. M. *Evidence-based Software Engineering: based on the publicly available data*. Knowledge Software, Ltd., Nov. 2020.

[102] Jørgensen, M., and Grimstad, S. The impact of irrelevant and misleading information on software development effort estimates: A randomized controlled field experiment. *IEEE Transactions on Software Engineering 37*, 5 (Sept. 2011), 695–707.

[103] Jørgensen, M., and Grimstad, S. Software development estimation biases: The role of interdependence. *IEEE Transactions on Software Engineering 38*, 3 (May 2012), 677–693.

[104] Kamienski, A. V., Palechor, L., Bezemer, C.-P., and Hindle, A. PySStuBs: Characterizing single-statement bugs in popular open-source python projects. In *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)* (May 2021), IEEE.

[105] Kanat-Alexander, M. *Code Simplicity: The Science of Software Development*. O'Reilly, 2012.

[106] Kapser, C. J., and Godfrey, M. W. "cloning considered harmful" considered harmful: patterns of cloning in software. *Empirical Software Engineering 13*, 6 (July 2008), 645–692.

[107] Kasi, B. K., and Sarma, A. Cassandra: Proactive conflict minimization through optimized task scheduling. In *2013 35th International Conference on Software Engineering (ICSE)* (May 2013), IEEE.

[108] Kavaler, D., Trockman, A., Vasilescu, B., and Filkov, V. Tool choice matters: JavaScript quality assurance tools and usage outcomes in GitHub projects. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)* (May 2019), IEEE.

[109] Khomh, F., Dhaliwal, T., Zou, Y., and Adams, B. Do faster releases improve software quality? an empirical case study of mozilla firefox. In *2012 9th IEEE Working Conference on Mining Software Repositories (MSR)* (June 2012), IEEE.

[110] Kiefer, M., Warzel, D., and Tichy, W. F. An empirical study on parallelism in modern open-source projects. In *Proceedings of the 2nd International Workshop on Software Engineering for Parallel Systems* (Oct. 2015), ACM.

[111] Kim, D., Kwon, Y., Liu, P., Kim, I. L., Perry, D. M., Zhang, X., and Rodriguez-Rivera, G. Apex: automatic programming assignment error explanation. In *Proceedings of the 2016 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications* (Oct. 2016), ACM.

[112] Kim, D., Nam, J., Song, J., and Kim, S. Automatic patch generation learned from human-written patches. In *2013 35th International Conference on Software Engineering (ICSE)* (May 2013), IEEE.

[113] Kim, D. J., Chen, T.-H., and Yang, J. The secret life of test smells—an empirical study on test smell evolution and maintenance. *Empirical Software Engineering 26*, 5 (July 2021).

[114] Kinshumann, K., Glerum, K., Greenberg, S., Aul, G., Orgovan, V., Nichols, G., Grant, D., Loihle, G., and Hunt, G. Debugging in the (very) large: ten years of implementation and experience. *Communications of the ACM 54*, 7 (July 2011), 111–116.

[115] Klotins, E., Unterkalmsteiner, M., Chatzipetrou, P., Gorschek, T., Prikladnicki, R., Tripathi, N., and Pompermaier, L. B. A progression model of software engineering goals, challenges, and practices in start-ups. *IEEE Transactions on Software Engineering 47*, 3 (Mar. 2021), 498–521.

[116] Kocaguneli, E., Menzies, T., and Keung, J. W. On the value of ensemble effort estimation. *IEEE Transactions on Software Engineering 38*, 6 (Nov. 2012), 1403–1416.

[117] Kochhar, P. S., Kalliamvakou, E., Nagappan, N., Zimmermann, T., and Bird, C. Moving from closed to open source: Observations from six transitioned projects to GitHub. *IEEE Transactions on Software Engineering* (2019), 1–1.

[118] Kosar, T., Gaberc, S., Carver, J. C., and Mernik, M. Program comprehension of domain-specific and general-purpose languages: replication of a family of experiments using integrated development environments. *Empirical Software Engineering 23*, 5 (Feb. 2018), 2734–2763.

[119] Krein, J. L., Prechelt, L., Juristo, N., Nanthaamornphong, A., Carver, J. C., Vegas, S., Knutson, C. D., Seppi, K. D., and Eggett, D. L. A multi-site joint replication of a design patterns experiment using moderator variables to generalize across contexts. *IEEE Transactions on Software Engineering 42*, 4 (Apr. 2016), 302–321.

[120] Krueger, R., Huang, Y., Liu, X., Santander, T., Weimer, W., and Leach, K. Neurological divide: an fMRI study of prose and code writing. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering* (June 2020), ACM.

[121] Latendresse, J., Abdalkareem, R., Costa, D. E., and Shihab, E. How effective is continuous integration in indicating single-statement bugs? In *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)* (May 2021), IEEE.

[122] Leitão, R. Technology-facilitated intimate partner abuse: a qualitative analysis of data from online domestic abuse forums. *Human–Computer Interaction 36*, 3 (Dec. 2019), 203–242.

[123] Levy, K., and Schneier, B. Privacy threats in intimate relationships. *Journal of Cybersecurity 6*, 1 (Jan. 2020).

[124] Lewis, C., Lin, Z., Sadowski, C., Zhu, X., Ou, R., and Whitehead, E. J. Does bug prediction support human developers? findings from a google case study. In *2013 35th International Conference on Software Engineering (ICSE)* (May 2013), IEEE.

[125] Li, S., Zhou, H., Lin, H., Xiao, T., Lin, H., Lin, W., and Xie, T. A characteristic study on failures of production distributed data-parallel programs. In *2013 35th International Conference on Software Engineering (ICSE)* (May 2013), IEEE.

[126] Liao, S. N., Zingaro, D., Laurenzano, M. A., Griswold, W. G., and Porter, L. Lightweight, early identification of at-risk CS1 students.

In *Proceedings of the 2016 ACM Conference on International Computing Education Research* (Aug. 2016), ACM.

[127] Lima, L. P., Rocha, L. S., Bezerra, C. I. M., and Paixao, M. Assessing exception handling testing practices in open-source libraries. *Empirical Software Engineering 26*, 5 (June 2021).

[128] Liu, K., Kim, D., Bissyande, T. F., Yoo, S., and Traon, Y. L. Mining fix patterns for FindBugs violations. *IEEE Transactions on Software Engineering 47*, 1 (Jan. 2021), 165–188.

[129] Lo, D., Nagappan, N., and Zimmermann, T. How practitioners perceive the relevance of software engineering research. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering* (Aug. 2015), ACM.

[130] Louis, A., Dash, S. K., Barr, E. T., Ernst, M. D., and Sutton, C. Where should i comment my code?: a dataset and model for predicting locations that need comments. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: New Ideas and Emerging Results* (June 2020), ACM.

[131] Lucia, A. D., Gravino, C., Oliveto, R., and Tortora, G. An experimental comparison of ER and UML class diagrams for data modelling. *Empirical Software Engineering 15*, 5 (Dec. 2009), 455–492.

[132] Ma, Y., Dey, T., Bogart, C., Amreen, S., Valiev, M., Tutko, A., Kennard, D., Zaretzki, R., and Mockus, A. World of code: enabling a research workflow for mining and analyzing the universe of open source VCS data. *Empirical Software Engineering 26*, 2 (Feb. 2021).

[133] Maalej, W., Tiarks, R., Roehm, T., and Koschke, R. On the comprehension of program comprehension. *ACM Transactions on Software Engineering and Methodology 23*, 4 (Sept. 2014), 1–37.

[134] Macho, C., Beyer, S., McIntosh, S., and Pinzger, M. The nature of build changes: An empirical study of maven-based build systems. *Empirical Software Engineering 26*, 3 (Mar. 2021).

[135] Mäenpää, H., Mäkinen, S., Kilamo, T., Mikkonen, T., Männistö, T., and Ritala, P. Organizing for openness: six models for developer involvement in hybrid OSS projects. *Journal of Internet Services and Applications 9*, 1 (Aug. 2018).

[136] Majumder, S., Chakraborty, J., Agrawal, A., and Menzies, T. Why software projects need heroes: Lessons learned from 1100+ projects. *arxiv.org abs/1904.09954* (2019).

[137] Malik, M., Schimel, A. C. G., Masetti, G., Roche, M., Deunf, J. L., Dolan, M. F., Beaudoin, J., Augustin, J.-M., Hamilton, T., and Parnum, I. Results from the first phase of the seafloor backscatter processing software inter-comparison project. *Geosciences 9*, 12 (Dec. 2019), 516.

[138] Malloy, B. A., and Power, J. F. An empirical analysis of the transition from python 2 to python 3. *Empirical Software Engineering 24*, 2 (July 2018), 751–778.

[139] Mangano, N., LaToza, T. D., Petre, M., and van der Hoek, A. How software designers interact with sketches at the whiteboard. *IEEE Transactions on Software Engineering 41*, 2 (Feb. 2015), 135–156.

[140] Marinescu, C. Are the classes that use exceptions defect prone? In *Proceedings of the 12th international workshop and the 7th annual ERCIM workshop on Principles on software evolution and software evolution - IWPSE-EVOL '11* (2011), ACM Press.

[141] Masood, Z., Hoda, R., and Blincoe, K. How agile teams make self-assignment work: a grounded theory study. *Empirical Software Engineering 25*, 6 (Sept. 2020), 4962–5005.

[142] Mattmann, C. A., Garcia, J., Krka, I., Popescu, D., and Medvidović, N. Revisiting the anatomy and physiology of the grid. *Journal of Grid Computing 13*, 1 (Jan. 2015), 19–34.

[143] May, A., Wachs, J., and Hannák, A. Gender differences in participation and reward on stack overflow. *Empirical Software Engineering 24*, 4 (Feb. 2019), 1997–2019.

[144] McGee, S., and Greer, D. Software requirements change taxonomy: Evaluation by case study. In *2011 IEEE 19th International Requirements Engineering Conference* (Aug. 2011), IEEE.

[145] McIntosh, S., Adams, B., Nguyen, T. H., Kamei, Y., and Hassan, A. E. An empirical study of build maintenance effort. In *Proceedings of the 33rd International Conference on Software Engineering* (May 2011), ACM.

[146] McLeod, L., and MacDonell, S. G. Factors that affect software systems development project outcomes. *ACM Computing Surveys 43*, 4 (Oct. 2011), 1–56.

[147] Meneely, A., Rotella, P., and Williams, L. Does adding manpower also affect quality?: an empirical, longitudinal analysis. In *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering - SIGSOFT/FSE '11* (2011), ACM Press.

[148] Meng, N., Kim, M., and McKinley, K. S. Lase: Locating and applying systematic edits by learning from examples. In *2013 35th International Conference on Software Engineering (ICSE)* (May 2013), IEEE.

[149] Meyer, A. N., Barr, E. T., Bird, C., and Zimmermann, T. Today was a good day: The daily life of software developers. *IEEE Transactions on Software Engineering 47*, 5 (May 2021), 863–880.

[150] Meyer, A. N., Fritz, T., Murphy, G. C., and Zimmermann, T. Software developers' perceptions of productivity. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering* (Nov. 2014), ACM.

[151] Miller, B., Zhang, M., and Heymann, E. The relevance of classic fuzz testing: Have we solved this one? *IEEE Transactions on Software Engineering* (2020), 1–1.

[152] Miller, C. S., and Settle, A. Some trouble with transparency: An analysis of student errors with object-oriented python. In *Proceedings of the 2016 ACM Conference on International Computing Education Research* (Aug. 2016), ACM.

[153] Mitropoulos, D., Louridas, P., Salis, V., and Spinellis, D. Time present and time past: Analyzing the evolution of JavaScript code in the wild. In *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)* (May 2019), IEEE.

[154] Mo, R., Cai, Y., Kazman, R., Xiao, L., and Feng, Q. Architecture anti-patterns: Automatically detectable violations of design principles. *IEEE Transactions on Software Engineering 47*, 5 (May 2021), 1008–1028.

[155] Mockus, A. Organizational volatility and its effects on software defects. In *Proceedings of the eighteenth ACM SIGSOFT international symposium on Foundations of software engineering - FSE'10* (2010), ACM Press.

[156] Moe, N. B., Dingsøyr, T., and Dybå, T. A teamwork model for understanding an agile team: A case study of a scrum project. *Information and Software Technology 52*, 5 (May 2010), 480–491.

[157] Mokhov, A., Mitchell, N., and Jones, S. P. Build systems à la carte. *Proceedings of the ACM on Programming Languages 2*, ICFP (July 2018), 1–29.

[158] Moldon, L., Strohmaier, M., and Wachs, J. How gamification affects software developers: Cautionary evidence from a natural experiment on GitHub. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)* (May 2021), IEEE.

[159] Moraes, J. P., Polato, I., Wiese, I., Saraiva, F., and Pinto, G. From one to hundreds: multi-licensing in the JavaScript ecosystem. *Empirical Software Engineering 26*, 3 (Mar. 2021).

[160] Moreira Soares, D., Júnior, M. L. L., Murta, L., and Plastino, A. What factors influence the lifetime of pull requests? *Software: Practice and Experience 51*, 6 (Dec. 2020), 1173–1193.

[161] Murphy-Hill, E., Jaspan, C., Sadowski, C., Shepherd, D., Phillips, M., Winter, C., Knight, A., Smith, E., and Jorde, M. What predicts software developers' productivity? *IEEE Transactions on Software Engineering 47*, 3 (Mar. 2021), 582–594.

[162] Nagappan, M., Robbes, R., Kamei, Y., Tanter, É., McIntosh, S., Mockus, A., and Hassan, A. E. An empirical study of goto in c code from GitHub repositories. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering* (Aug. 2015), ACM.

[163] Nagappan, N., Maximilien, E. M., Bhat, T., and Williams, L. Realizing quality improvement through test driven development: results and experiences of four industrial teams. *Empirical Software Engineering 13*, 3 (Feb. 2008), 289–302.

[164] Nakshatri, S., Hegde, M., and Thandra, S. Analysis of exception handling patterns in java projects. In *Proceedings of the 13th International Conference on Mining Software Repositories* (May 2016), ACM.

[165] Near, J. P., and Jackson, D. Finding security bugs in web applications using a catalog of access control patterns. In *Proceedings of the 38th International Conference on Software Engineering* (May 2016), ACM.

[166] Nguyen-Duc, A., Kemell, K.-K., and Abrahamsson, P. The entrepreneurial logic of startup software development: A study of 40 software startups. *Empirical Software Engineering 26*, 5 (July 2021).

[167] Nielebock, S., Krolikowski, D., Krüger, J., Leich, T., and Ortmeier, F. Commenting source code: is it worth it for small programming tasks? *Empirical Software Engineering 24*, 3 (Nov. 2018), 1418–1457.

[168] Nüssli, M.-A., and Jermann, P. Effects of sharing text selections on gaze cross-recurrence and interaction quality in a pair programming task. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work - CSCW '12* (2012), ACM Press.

[169] Oliveira, E., Fernandes, E., Steinmacher, I., Cristo, M., Conte, T., and Garcia, A. Code and commit metrics of developer productivity: a study on team leaders perceptions. *Empirical Software Engineering 25*, 4 (Apr. 2020), 2519–2549.

[170] Olsson, J., Risfelt, E., Besker, T., Martini, A., and Torkar, R. Measuring affective states from technical debt. *Empirical Software Engineering 26*, 5 (July 2021).

[171] Overney, C., Meinicke, J., Kästner, C., and Vasilescu, B. How to not get rich: an empirical study of donations in open source. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering* (June 2020), ACM.

[172] Palomba, F., Tamburri, D. A., Fontana, F. A., Oliveto, R., Zaidman, A., and Serebrenik, A. Beyond technical aspects: How do community smells influence the intensity of code smells? *IEEE Transactions on Software Engineering 47*, 1 (Jan. 2021), 108–129.

[173] Pankratius, V., Schmidt, F., and Garreton, G. Combining functional and imperative programming for multicore software: An empirical study evaluating scala and java. In *2012 34th International Conference on Software Engineering (ICSE)* (June 2012), IEEE.

[174] Parnin, C., and Rugaber, S. Programmer information needs after memory failure. In *2012 20th IEEE International Conference on Program Comprehension (ICPC)* (June 2012), IEEE.

[175] Passos, L., Queiroz, R., Mukelabai, M., Berger, T., Apel, S., Czarnecki, K., and Padilla, J. A. A study of feature scattering in the linux kernel. *IEEE Transactions on Software Engineering 47*, 1 (Jan. 2021), 146–164.

[176] Patitsas, E., Berlin, J., Craig, M., and Easterbrook, S. Evidence that computer science grades are not bimodal. In *Proceedings of the 2016 ACM Conference on International Computing Education Research* (Aug. 2016), ACM.

[177] Peitek, N., Apel, S., Parnin, C., Brechmann, A., and Siegmund, J. Program comprehension and code complexity metrics: An fMRI study. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)* (May 2021), IEEE.

[178] Perez De Rosso, S., and Jackson, D. Purposes, concepts, misfits, and a redesign of git. In *Proceedings of the 2016 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications* (Oct. 2016), ACM.

[179] Petre, M. UML in practice. In *2013 35th International Conference on Software Engineering (ICSE)* (May 2013), IEEE.

[180] Philip, K., Umarji, M., Agarwala, M., Sim, S. E., Gallardo-Valencia, R., Lopes, C. V., and Ratanotayanon, S. Software reuse through methodical component reuse and amethodical snippet remixing.

In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work - CSCW '12* (2012), ACM Press.

[181] Pizard, S., Acerenza, F., Otegui, X., Moreno, S., Vallespir, D., and Kitchenham, B. Training students in evidence-based software engineering and systematic reviews: a systematic review and empirical study. *Empirical Software Engineering 26*, 3 (Mar. 2021).

[182] Porter, L., Lee, C. B., and Simon, B. Halving fail rates using peer instruction. In *Proceeding of the 44th ACM technical symposium on Computer science education - SIGCSE '13* (2013), ACM Press.

[183] Posnett, D., Hindle, A., and Devanbu, P. Got issues? do new features and code improvements affect defects? In *2011 18th Working Conference on Reverse Engineering* (Oct. 2011), IEEE.

[184] Prabhu, P., Zhang, Y., Ghosh, S., August, D. I., Huang, J., Beard, S., Kim, H., Oh, T., Jablin, T. B., Johnson, N. P., Zoufaly, M., Raman, A., Liu, F., and Walker, D. A survey of the practice of computational science. In *State of the Practice Reports on - SC '11* (2011), ACM Press.

[185] Prana, G. A. A., Treude, C., Thung, F., Atapattu, T., and Lo, D. Categorizing the content of GitHub README files. *Empirical Software Engineering 24*, 3 (Oct. 2018), 1296–1327.

[186] Pritchard, D. Frequency distribution of error messages. In *Proceedings of the 6th Workshop on Evaluation and Usability of Programming Languages and Tools* (Oct. 2015), ACM.

[187] Qiu, H. S., Nolte, A., Brown, A., Serebrenik, A., and Vasilescu, B. Going farther together: The impact of social capital on sustained participation in open source. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)* (May 2019), IEEE.

[188] Racheva, Z., Daneva, M., Sikkel, K., Herrmann, A., and Wieringa, R. Do we know enough about requirements prioritization in agile projects: Insights from a case study. In *2010 18th IEEE International Requirements Engineering Conference* (Sept. 2010), IEEE.

[189] Ragkhitwetsagul, C., Krinke, J., Paixao, M., Bianco, G., and Oliveto, R. Toxic code snippets on stack overflow. *IEEE Transactions on Software Engineering 47*, 3 (Mar. 2021), 560–581.

[190] Rahman, A., Farhana, E., Parnin, C., and Williams, L. Gang of eight: a defect taxonomy for infrastructure as code scripts. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering* (June 2020), ACM.

[191] Rahman, F., and Devanbu, P. Ownership, experience and defects: a fine-grained study of authorship. In *Proceedings of the 33rd International Conference on Software Engineering* (May 2011), ACM.

[192] Rahman, F., and Devanbu, P. How, and why, process metrics are better. In *2013 35th International Conference on Software Engineering (ICSE)* (May 2013), IEEE.

[193] Rahman, M. M., Khomh, F., and Castelluccio, M. Why are some bugs non-reproducible? an empirical investigation using data fusion. In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (Sept. 2020), IEEE.

[194] Reyes, R. P., Dieste, O., Fonseca, E. R., and Juristo, N. Statistical errors in software engineering experiments. In *Proceedings of the 40th International Conference on Software Engineering* (May 2018), ACM.

[195] Rigby, P. C., and Storey, M.-A. Understanding broadcast based peer review on open source software projects. In *Proceedings of the 33rd International Conference on Software Engineering* (May 2011), ACM.

[196] Rigger, M., and Su, Z. Finding bugs in database systems via query partitioning. *Proceedings of the ACM on Programming Languages 4*, OOPSLA (Nov. 2020), 1–30.

[197] Rivers, K., Harpstead, E., and Koedinger, K. Learning curve analysis for programming. In *Proceedings of the 2016 ACM Conference on International Computing Education Research* (Aug. 2016), ACM.

[198] Robillard, M. P., and DeLine, R. A field study of API learning obstacles. *Empirical Software Engineering 16*, 6 (Dec. 2010), 703–732.

[199] Rodríguez-Pérez, G., Robles, G., Serebrenik, A., Zaidman, A., Germán, D. M., and Gonzalez-Barahona, J. M. How bugs are born: a model to identify how bugs are introduced in software components. *Empirical Software Engineering 25*, 2 (Feb. 2020), 1294–1340.

[200] Rossbach, C. J., Hofmann, O. S., and Witchel, E. Is transactional programming actually easier? *ACM SIGPLAN Notices 45*, 5 (May 2010), 47–56.

[201] Sadowski, C., and Zimmermann, T., Eds. *Rethinking Productivity in Software Engineering*. Apress, 2019.

[202] Sarker, F., Vasilescu, B., Blincoe, K., and Filkov, V. Socio-technical work-rate increase associates with changes in work patterns in online projects. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)* (May 2019), IEEE.

[203] Scalabrino, S., Bavota, G., Vendome, C., Linares-Vasquez, M., Poshyvanyk, D., and Oliveto, R. Automatically assessing code understandability. *IEEE Transactions on Software Engineering 47*, 3 (Mar. 2021), 595–613.

[204] Scalabrino, S., Linares-Vásquez, M., Oliveto, R., and Poshyvanyk, D. A comprehensive model for code readability. *Journal of Software: Evolution and Process 30*, 6 (June 2018), e1958.

[205] Scanniello, G., Risi, M., Tramontana, P., and Romano, S. Fixing faults in c and java source code. *ACM Transactions on Software Engineering and Methodology 26*, 2 (Oct. 2017), 1–43.

[206] Sedano, T., Ralph, P., and Peraire, C. Software development waste. In *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)* (May 2017), IEEE.

[207] Shao, S., Qiu, Z., Yu, X., Yang, W., Jin, G., Xie, T., and Wu, X. Database-access performance antipatterns in database-backed web applications. In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (Sept. 2020), IEEE.

[208] Sharma, P. N., Savarimuthu, B. T. R., and Stanger, N. Extracting rationale for open source software development decisions—a study of python email archives. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)* (May 2021), IEEE.

[209] Sharp, H., Dittrich, Y., and de Souza, C. R. B. The role of ethnographic studies in empirical software engineering. *IEEE Transactions on Software Engineering 42*, 8 (Aug. 2016), 786–804.

[210] Shrestha, N., Botta, C., Barik, T., and Parnin, C. Here we go again: why is it difficult for developers to learn another programming language? In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering* (June 2020), ACM.

[211] Soremekun, E., Kirschner, L., Böhme, M., and Zeller, A. Locating faults with program slicing: an empirical analysis. *Empirical Software Engineering 26*, 3 (Apr. 2021).

[212] Soto-Valero, C., Harrand, N., Monperrus, M., and Baudry, B. A comprehensive study of bloated dependencies in the maven ecosystem. *Empirical Software Engineering 26*, 3 (Mar. 2021).

[213] Spadini, D., Çalikli, G., and Bacchelli, A. Primers or reminders?: the effects of existing review comments on code review. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering* (June 2020), ACM.

[214] Spadini, D., Palomba, F., Baum, T., Hanenberg, S., Bruntink, M., and Bacchelli, A. Test-driven code review: An empirical study. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)* (May 2019), IEEE.

[215] Spiegler, S. V., Heinecke, C., and Wagner, S. An empirical study on changing leadership in agile teams. *Empirical Software Engineering 26*, 3 (Mar. 2021).

[216] Spinellis, D., and Avgeriou, P. Evolution of the unix system architecture: An exploratory case study. *IEEE Transactions on Software Engineering 47*, 6 (June 2021), 1134–1163.

[217] Staples, M., Kolanski, R., Klein, G., Lewis, C., Andronick, J., Murray, T., Jeffery, R., and Bass, L. Formal specifications better than function points for code sizing. In *2013 35th International Conference on Software Engineering (ICSE)* (May 2013), IEEE.

[218] Stefik, A., and Siebert, S. An empirical investigation into programming language syntax. *ACM Transactions on Computing Education 13*, 4 (Nov. 2013), 1–40.

[219] Stefik, A., Siebert, S., Stefik, M., and Slattery, K. An empirical comparison of the accuracy rates of novices using the quorum, perl, and randomo programming languages. In *Proceedings of the 3rd ACM SIGPLAN workshop on Evaluation and usability of programming languages and tools - PLATEAU '11* (2011), ACM Press.

[220] Stol, K.-J., and Fitzgerald, B. The ABC of software engineering research. *ACM Transactions on Software Engineering and Methodology 27*, 3 (Oct. 2018), 1–51.

[221] Stolee, K. T., and Elbaum, S. Refactoring pipe-like mashups for end-user programmers. In *Proceedings of the 33rd International Conference on Software Engineering* (May 2011), ACM.

[222] Stylos, J., and Clarke, S. Usability implications of requiring parameters in objects' constructors. In *29th International Conference on Software Engineering (ICSE'07)* (May 2007), IEEE.

[223] Taipalus, T., Siponen, M., and Vartiainen, T. Errors and complications in SQL query formulation. *ACM Transactions on Computing Education 18*, 3 (Sept. 2018), 1–29.

[224] Tamburri, D. A., Blincoe, K., Palomba, F., and Kazman, R. "the canary in the coal mine..." a cautionary tale from the decline of SourceForge. *Software: Practice and Experience 50*, 10 (July 2020), 1930–1951.

[225] Tan, X., Zhou, M., and Sun, Z. A first look at good first issues on GitHub. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (Nov. 2020), ACM.

[226] Tew, A. E., and Guzdial, M. The FCS1: a language independent assessment of cs1 knowledge. In *Proceedings of the 42nd ACM technical symposium on Computer science education - SIGCSE '11* (2011), ACM Press.

[227] Thongtanunam, P., McIntosh, S., Hassan, A. E., and Iida, H. Revisiting code ownership and its relationship with software quality in the scope of modern code review. In *Proceedings of the 38th International Conference on Software Engineering* (May 2016), ACM.

[228] Tómasdóttir, K. F., Aniche, M., and van Deursen, A. The adoption of JavaScript linters in practice: A case study on ESLint. *IEEE Transactions on Software Engineering 46*, 8 (Aug. 2020), 863–891.

[229] Tomassi, D. A., Dmeiri, N., Wang, Y., Bhowmick, A., Liu, Y.-C., Devanbu, P. T., Vasilescu, B., and Rubio-Gonzalez, C. BugSwarm: Mining and continuously growing a dataset of reproducible failures and fixes. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)* (May 2019), IEEE.

[230] Tourani, P., Adams, B., and Serebrenik, A. Code of conduct in open source projects. In *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)* (Feb. 2017), IEEE.

[231] Tregubov, A., Boehm, B., Rodchenko, N., and Lane, J. A. Impact of task switching and work interruptions on software development processes. In *Proceedings of the 2017 International Conference on Software and System Process* (July 2017), ACM.

[232] Vanhanen, J., and Korpi, H. Experiences of using pair programming in an agile project. In *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)* (2007), IEEE.

[233] Venigalla, A. S. M., and Chimalakonda, S. On the comprehension of application programming interface usability in game engines. *Software: Practice and Experience 51*, 8 (May 2021), 1728–1744.

[234] Wang, P., Brown, C., Jennings, J. A., and Stolee, K. T. An empirical study on regular expression bugs. In *Proceedings of the 17th International Conference on Mining Software Repositories* (June 2020), ACM.

[235] Wang, X., Gulwani, S., and Singh, R. FIDEX: filtering spreadsheet data using examples. In *Proceedings of the 2016 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications* (Oct. 2016), ACM.

[236] Washburn, M., Sathiyanarayanan, P., Nagappan, M., Zimmermann, T., and Bird, C. What went right and what went wrong: an analysis of 155 postmortems from game development. In *Proceedings of the 38th International Conference on Software Engineering Companion* (May 2016), ACM.

[237] Weintrop, D., and Wilensky, U. Comparing block-based and text-based programming in high school computer science classrooms. *ACM Transactions on Computing Education 18*, 1 (Dec. 2017), 1–25.

[238] Weir, C., Becker, I., and Blair, L. A passion for security: Intervening to help software developers. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)* (May 2021), IEEE.

[239] Wessel, M., Serebrenik, A., Wiese, I., Steinmacher, I., and Gerosa, M. A. Effects of adopting code review bots on pull requests to OSS projects. In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (Sept. 2020), IEEE.

[240] Wicherts, J. M., Bakker, M., and Molenaar, D. Willingness to share research data is related to the strength of the evidence and the quality of reporting of statistical results. *PLoS ONE 6*, 11 (Nov. 2011), e26828.

[241] Wilkerson, J. W., Nunamaker, J. F., and Mercer, R. Comparing the defect reduction benefits of code inspection and test-driven development. *IEEE Transactions on Software Engineering 38*, 3 (May 2012), 547–560.

[242] Xu, T., Jin, L., Fan, X., Zhou, Y., Pasupathy, S., and Talwadker, R. Hey, you have given me too many knobs!: understanding and dealing with over-designed configuration in system software. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering* (Aug. 2015), ACM.

[243] Yasmin, J., Tian, Y., and Yang, J. A first look at the deprecation of RESTful APIs: An empirical study. In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (Sept. 2020), IEEE.

[244] Yin, Z., Yuan, D., Zhou, Y., Pasupathy, S., and Bairavasundaram, L. How do fixes become bugs? In *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering - SIGSOFT/FSE '11* (2011), ACM Press.

[245] Yu, Z., Bai, C., Seinturier, L., and Monperrus, M. Characterizing the usage, evolution and impact of java annotations in practice. *IEEE Transactions on Software Engineering 47*, 5 (May 2021), 969–986.

[246] Yuan, D., Luo, Y., Zhuang, X., Renna Rodrigues, G., Zhao, X., Jain, P. U., and Stumm, M. Simple testing can prevent most critical failures—an analysis of production failures in distributed data-intensive systems. In *11th USENIX Symposium on Operating System Design and Implementation (OSDI'14)* (2014).

[247] Zampetti, F., Vassallo, C., Panichella, S., Canfora, G., Gall, H., and Penta, M. D. An empirical characterization of bad practices in continuous integration. *Empirical Software Engineering 25*, 2 (Jan. 2020), 1095–1135.

[248] Zhang, H., Wang, S., Chen, T.-H., and Hassan, A. E. Reading answers on stack overflow: Not enough! *IEEE Transactions on Software Engineering* (2020), 1–1.

[249] Zhang, H., Wang, S., Chen, T.-H., Zou, Y., and Hassan, A. E. An empirical study of obsolete answers on stack overflow. *IEEE Transactions on Software Engineering 47*, 4 (Apr. 2021), 850–862.

[250] Zhang, J., Jiang, H., Ren, Z., Zhang, T., and Huang, Z. Enriching API documentation with code samples and usage scenarios from crowd knowledge. *IEEE Transactions on Software Engineering 47*, 6 (June 2021), 1299–1314.

[251] Zhu, W., and Godfrey, M. W. Mea culpa: How developers fix their own simple bugs differently from other developers. In *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)* (May 2021), IEEE.

[252] Zieris, F., and Prechelt, L. Explaining pair programming session dynamics from knowledge gaps. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering* (June 2020), ACM.