Miguel Velez

Publications

Books

[65] C. Molnar. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. `https://christophm.github.io/interpretable-ml-book/`. 2019.

[64] J. Bloch. *Effective Java™, Third Edition*. Third. Addison-Wesley, 2018.

[63] E. M. Clarke Jr, O. Grumberg, D. Kroening, D. Peled, and H. Veith. *Model Checking*. MIT press, 2018.

[62] R. E. Bryant and D. R. O'Hallaron. *Computer Systems: A Programmer's Perspective*. 3rd. Pearson, 2015.

[61] T. Lindholm, F. Yellin, G. Bracha, and A. Buckley. *The Java Virtual Machine Specification, Java SE 8 Edition*. 1st. Addison-Wesley Professional, 2015.

[60] J. Saldaña. *The coding manual for qualitative researchers*. Sage, 2015.

[59] N. Viswanadham and Y. Narahari. *Performance modeling of automated systems*. PHI Learning Pvt. Ltd., 2015.

[58] E. W. Frees, R. A. Derrig, and G. Meyers. *Predictive modeling applications in actuarial science*. Vol. 1. Cambridge University Press, 2014.

[57] A. Shostack. *Threat Modeling: Designing for Security*. 1st. Wiley Publishing, 2014.

[56] S. Apel, D. Batory, C. Kästner, and G. Saake. *Feature-Oriented Software Product Lines: Concepts and Implementation*. Berlin/Heidelberg, Germany: Springer-Verlag, 2013.

[55] A. Downey. *Think Bayes*. " O'Reilly Media, Inc.", 2013.

[54] M. Harchol-Balter. *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*. 1st. New York, NY, USA: Cambridge University Press, 2013.

[53] D. R. Kuhn, R. N. Kacker, and Y. Lei. *Introduction to Combinatorial Testing*. 1st. Chapman & Hall/CRC, 2013.

[52] T. Lindholm, F. Yellin, G. Bracha, and A. Buckley. *The Java Virtual Machine Specification, Java SE 7 Edition*. 1st. Addison-Wesley Professional, 2013.

[51] T. Parr. *The Definitive ANTLR 4 Reference*. 2nd. Pragmatic Bookshelf, 2013.

[50] A. Brown and G. Wilson. *The Architecture of Open Source Applications, Volume II*. The Architecture of Open Source Applications v. 2. Kristian Hermansen, 2012.

[49] A. V. Levitin. *Introduction to the Design and Analysis of Algorithms (3rd Edition)*. Boston, MA, USA: Addison-Wesley, 2012.

[48] J. Mongan, N. Suojanen, and E. Giguere. *Programming Interviews Exposed*. Wiley Publishing, Inc., 2012.

[47] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[46] M. Schreier. *Qualitative content analysis in practice*. Sage publications, 2012.

[45] H. J. Seltman. *Experimental Design and Analysis*. Carnegie Mellon University, 2012.

[44] A. Brown and G. Wilson. *The Architecture of Open Source Applications, Volume I*. The Architecture of Open Source Applications v. 1. CreativeCommons, 2011.

[43] M. Odersky, L. Spoon, and B. Venners. *Programming in Scala: A Comprehensive Step-by-Step Guide, 2nd Edition*. 2nd. USA: Artima Incorporation, 2011.

[42] I. Sommerville. *Software Engineering*. Addison-Wesley, 2011.

[41] I. H. Witten, E. Frank, and M. A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. 3rd. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.

[40] F. P. Brooks. *The Design of Design: Essays from a Computer Scientist*. 1st. Addison-Wesley Professional, 2010.

[39] R. Jääskeläinen. *Think-aloud protocol*. John Benjamins Publishing Amsterdam/Philadelphia, 2010, pp. 371–374.

[38] F. Nielson, H. R. Nielson, and C. Hankin. *Principles of Program Analysis*. Springer Publishing Company, Incorporated, 2010.

[37] I. Sommerville. *Software Engineering*. 9th. USA: Addison-Wesley Publishing Company, 2010.

[36] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Third Edition*. 3rd. MIT Press, 2009.

[35] T. Parr. *Language Implementation Patterns: Create Your Own Domain-Specific and General Programming Languages*. 1st. Pragmatic Bookshelf, 2009.

[34] R. Yin. *Case Study Research: Design and Methods*. 4th. Applied Social Research Methods. SAGE Publications, 2009.

[33] A. Zeller. *Why programs fail: a guide to systematic debugging*. Elsevier, 2009.

[32] J. Bloch. *Effective Java™, Second Edition*. Second. Upper Saddle River, NJ, USA: Prentice Hall Press, 2008.

[31] M. Herlihy and N. Shavit. *The Art of Multiprocessor Programming*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2008.

[30] A. C. Rencher and G. B. Schaalje. *Linear models in statistics*. John Wiley & Sons, 2008.

[29] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola. *Global sensitivity analysis: the primer*. John Wiley & Sons, 2008.

[28] R. Williams. *The Non-designer's Design Book, Third Edition*. Third. Berkeley, CA, USA: Peachpit Press, 2008.

[27] T. Parr. *The Definitive ANTLR Reference: Building Domain-Specific Languages*. Pragmatic Bookshelf, 2007.

[26] F. Shull, J. Singer, and D. I. Sjøberg. *Guide to Advanced Empirical Software Engineering*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[25] A. V. Aho, M. S. Lam, R. Sethi, and J. D. Ullman. *Compilers: Principles, Techniques, and Tools (2nd Edition)*. Boston, MA, USA: Addison-Wesley, 2006.

[24] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[23] P. Feiler, K. Sullivan, K. Wallnau, R. Gabriel, J. Goodenough, R. Linger, T. Longstaff, R. Kazman, M. Klein, L. Northrop, and D. Schmidt. *Ultra-Large-Scale Systems: The Software Challenge of the Future*. Software Engineering Institute, Carnegie Mellon University, 2006.

[22] D. C. Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons, 2006.

[21] E. S. Allman, E. S. Allman, and J. A. Rhodes. *Mathematical models in biology: an introduction*. Cambridge University Press, 2004.

[20] S. Krishnamurthi. *Programming Languages: Application and Interpretation*. e-book, 2003.

[19] R. Laddad. *AspectJ in Action: Practical Aspect-Oriented Programming*. Greenwich, CT, USA: Manning Publications, 2003.

[18] P. Clements, B. Felix, L. Bass, D. Garlan, J. Ivers, R. Little, P. Merson, R. Nord, and J. Stafford. *Documenting Software Architectures: Views and Beyond*. Pearson Education, 2002.

[17] P. Clements and L. Northrop. *Software Product Lines*. Addison-Wesley, 2002.

[16] D. J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. New York, NY, USA: Cambridge University Press, 2002.

[15] A. Shalloway and J. R. Trott. *Design Patterns Explained: A New Perspective on Object-Oriented Design*. Boston, MA, USA: Addison-Wesley, 2002.

[14] C. Szyperski. *Component Software: Beyond Object-Oriented Programming*. 2nd. Boston, MA, USA: Addison-Wesley, 2002.

[13] K. Czarnecki and U. Eisenecker. *Generative Programming: Methods, Tools, and Applications*. New York, NY, USA: ACM Press/Addison-Wesley, 2000.

[12] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering: An Introduction*. Norwell, MA, USA: Kluwer Academic Publishers, 2000.

[11] D. E. Knuth. *The Art of Computer Programming, Volume 3: (2Nd Ed.) Sorting and Searching*. Redwood City, CA, USA: Addison-Wesley, 1998.

[10] D. E. Knuth. *The Art of Computer Programming, Volume 1 (3rd Ed.): Fundamental Algorithms*. Redwood City, CA, USA: Addison-Wesley, 1997.

[9] D. E. Knuth. *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms*. Boston, MA, USA: Addison-Wesley, 1997.

[8] T. M. Mitchell. *Machine Learning*. 1st. New York, NY, USA: McGraw-Hill, Inc., 1997.

[7] H. Abelson and G. J. Sussman. *Structure and Interpretation of Computer Programs*. 2nd. Cambridge, MA, USA: MIT Press, 1996.

[6] M. Shaw and D. Garlan. *Software Architecture: Perspectives on an Emerging Discipline*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.

[5] H. A. Simon. *The Sciences of the Artificial (3rd Ed.)* Cambridge, MA, USA: MIT Press, 1996.

[4] F. P. Brooks Jr. *The Mythical Man-Month*. anniversary. Boston, MA, USA: Addison-Wesley, 1995.

[3] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-oriented Software*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995.

[2] D. P. Siewiorek, C. G. Bell, and A. Newell. *Computer structures: principles and examples*. McGraw-Hill computer science series. McGraw-Hill, 1987. Chap. 2, Levels and Abstraction.

[1] P. Naur and B. Randell. *Software Engineering: Report of a Conference Sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 Oct. 1968, Brussels, Scientific Affairs Division, NATO*. NATO, 1969.

Refereed Journal Articles

[123] C. Kaltenecker, A. Grebhahn, N. Siegmund, and S. Apel. "The Interplay of Sampling and Machine Learning for Software Performance Prediction". In *IEEE Software* (2020).

[122] M. Velez, P. Jamshidi, F. Sattler, N. Siegmund, S. Apel, and C. Kästner. "ConfigCrusher: Towards White-Box Performance Analysis for Configurable Systems". In *Autom Softw Eng* (2020).

[121] M. Sayagh, N. Kerzazi, F. Petrillo, K. Bennani, and B. Adams. "What should your run-time configuration framework do to help developers?" In *Empirical Software Engineering* 25.2 (Jan. 2020).

[120] F. Psallidas, Y. Zhu, B. Karlas, M. Interlandi, A. Floratou, K. Karanasos, W. Wu, C. Zhang, S. Krishnan, C. Curino, and M. Weimer. "Data Science through the looking glass and what we found there". In *arXiv* 1912.09536 (2019).

[119] C. Rudin. "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead". In *Nature Machine Intelligence* 5 (2019), pp. 206–215.

[118] Z. Wan, X. Xia, D. Lo, and G. C. Murphy. "How does Machine Learning Change Software Development Practices?" In *IEEE Trans. Softw. Eng. (TSE)* (2019).

[117] T. Xu. "Configuration Testing: Testing Configuration Values Together with Code Logic". In *CoRR* abs/1905.12195 (2019).

[116] C. Rudin and J. Radin. "Why Are We Using Black Box Models in AI When We Don't Need To? A Lesson From An Explainable AI Competition". In *Harvard Data Science Review* 1.2 (Nov. 2019).

[115] R. Padhye, C. Lemieux, K. Sen, L. Simon, and H. Vijayakumar. "FuzzFactory: Domain-Specific Fuzzing with Waypoints". In *Proc. Int'l Conf. Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)* 3 (Oct. 2019).

[114] B. Shen, W. Zhang, H. Zhao, G. Liang, Z. Jin, and Q. Wang. "IntelliMerge: A Refactoring-Aware Software Merging Technique". In *Proc. Int'l Conf. Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)* (Oct. 2019).

[113] R. Padhye, C. Lemieux, K. Sen, M. Papadakis, and Y. Le Traon. "Semantic Fuzzing with Zest". In (June 2019), pp. 329–340.

[112] J. Aldrich, D. Garlan, C. Kaestner, C. Le Goues, A. Mohseni-Kabir, I. Ruchkin, S. Samuel, B. Schmerl, C. S. Timperley, M. Veloso, I. Voysey, J. Biswas, A. Guha, J. Holtz, J. Camara, and P. Jamshidi. "Model-Based Adaptation for Robotics Software". In *IEEE Software* 36.2 (Mar. 2019), pp. 83–90.

[111] M. Kim, T. Zimmermann, R. DeLine, and A. Begel. "Data Scientists in Software Teams: State of the Art and Challenges". In *IEEE Transactions on Software Engineering* 44.11 (2018), pp. 1024–1038.

[110] J. Meinicke, C.-P. Wong, C. Kästner, and G. Saake. "Understanding differences among executions with variational traces". In *arXiv preprint arXiv:1807.03837* (2018).

[109] P. S. Park, J. E. Blumenstock, and M. W. Macy. “The strength of long-range ties in population-scale social networks”. In *Science* 362.6421 (2018), pp. 1410–1413.

[108] J. Somers. “The scientific paper is obsolete”. In *The Atlantic* (2018).

[107] S. Souto and M. d’Amorim. “Time-space efficient regression testing for configurable systems”. In *Journal of Systems and Software* (2018).

[106] M. Lillack, C. Kästner, and E. Bodden. “Tracking Load-time Configuration Options”. In *IEEE Transactions on Software Engineering* 44.12 (Dec. 2018), pp. 1269–1291.

[105] A. Barcomb, A. Kaufmann, D. Riehle, K. Stol, and B. Fitzgerald. “Uncovering the Periphery: A Qualitative Survey of Episodic Volunteering in Free/Libre and Open Source Software Communities”. In *IEEE Transactions on Software Engineering* (Oct. 2018), pp. 1–1.

[104] J. Bornholt and E. Torlak. “Finding Code That Explodes Under Symbolic Evaluation”. In *Proc. Int’l Conf. Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)* 2 (Oct. 2018), 149:1–149:26.

[103] C.-P. Wong, J. Meinicke, L. Lazarek, and C. Kästner. “Faster Variational Execution with Transparent Bytecode Transformation”. In *Proc. Int’l Conf. Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)* 2 (Oct. 2018).

[102] T. Yu and M. Pradel. “Pinpointing and Repairing Performance Bottlenecks in Concurrent Programs”. In *Empirical Softw. Eng.* 23.5 (Oct. 2018), pp. 3034–3071.

[101] M. Allamanis, E. T. Barr, P. Devanbu, and C. Sutton. “A Survey of Machine Learning for Big Code and Naturalness”. In *ACM Computing Surveys (CSUR)* 51.4 (July 2018), 81:1–81:37.

[100] A. Halin, A. Nuttinck, M. Acher, X. Devroey, G. Perrouin, and B. Baudry. “Test them all, is it worth it? Assessing configuration sampling on the JHipster Web development stack”. In *Empirical Software Engineering* (July 2018).

[99] J. Ousterhout. “Always Measure One Level Deeper”. In *Commun. ACM* 61.7 (June 2018), pp. 74–83.

[98] S. Kolesnikov, N. Siegmund, C. Kästner, A. Grebhahn, and S. Apel. “Tradeoffs in modeling performance of highly configurable software systems”. In *Software and System Modeling (SoSyM)* (Feb. 2018).

[97] I. Abal, J. Melo, Ş. Stănciulescu, C. Brabrand, M. Ribeiro, and A. Wąsowski. “Variability Bugs in Highly Configurable Systems: A Qualitative Analysis”. In *ACM Trans. Softw. Eng. Methodol. (TOSEM)* 26.3 (Jan. 2018), 10:1–10:34.

[96] F. Doshi-Velez and B. Kim. “Towards a rigorous science of interpretable machine learning”. In *arXiv preprint arXiv:1702.08608* (2017).

[95] S. Gulwani, O. Polozov, and R. Singh. “Program synthesis”. In *Foundations and Trends® in Programming Languages* 4.1-2 (2017), pp. 1–119.

[94] J. Guo, D. Yang, N. Siegmund, S. Apel, A. Sarkar, P. Valov, K. Czarnecki, A. Wasowski, and H. Yu. "Data-efficient performance learning for configurable systems". In *Empirical Software Engineering* 23 (2017), pp. 1826–1867.

[93] A. Tomkins, M. Zhang, and W. D. Heavlin. "Single versus Double Blind Reviewing at WSDM 2017". In *CoRR* abs/1702.00502 (2017).

[92] G. Pinto and F. Castor. "Energy Efficiency: A New Concern for Application Software Developers". In *Commun. ACM* 60.12 (Nov. 2017), pp. 68–75.

[91] J. Späth, K. Ali, and E. Bodden. "IDEal: Efficient and Precise Alias-aware Dataflow Analysis". In *Proc. ACM Program. Lang.* 1.OOPSLA (Oct. 2017), 99:1–99:27.

[90] S. Martnez, V. Cosentino, and J. Cabot. "Model-based Analysis of Java EE Web Security Misconfigurations". In *Comput. Lang. Syst. Struct.* 49.C (Sept. 2017), pp. 36–61.

[89] S. Price and P. A. Flach. "Computational Support for Academic Peer Review: A Perspective from Artificial Intelligence". In *Commun. ACM* 60.3 (Feb. 2017), pp. 70–79.

[88] M. Balog, A. L. Gaunt, M. Brockschmidt, S. Nowozin, and D. Tarlow. "Deepcoder: Learning to write programs". In *arXiv preprint arXiv:1611.01989* (2016).

[87] J. L. Jenkins, B. B. Anderson, A. Vance, C. B. Kirwan, and D. Eargle. "More harm than good? How messages that interrupt can make us vulnerable". In *Information Systems Research* 27.4 (2016), pp. 880–896.

[86] N. Polikarpova, J. Yang, S. Itzhaky, and A. Solar-Lezama. "Type-Driven Repair for Information Flow Security". In *CoRR* abs/1607.03445 (2016).

[85] B. A. Myers, A. J. Ko, T. D. LaToza, and Y. Yoon. "Programmers Are Users Too: Human-Centered Methods for Improving Programming Tools". In *Computer* 49.7 (July 2016), pp. 44–52.

[84] B. Gregg. "The Flame Graph". In *Commun. ACM* 59.6 (May 2016), pp. 48–57.

[83] B. A. Myers and J. Stylos. "Improving API Usability". In *Commun. ACM* 59.6 (May 2016), pp. 62–69.

[82] A. Hervieu, D. Marijan, A. Gotlieb, and B. Baudry. "Optimal Minimisation of Pairwise-covering Test Configurations Using Constraint Programming". In *Information and Software Technology* 71 (Mar. 2016), pp. 129–146.

[81] E. Gelenbe and Y. Caseau. "The Impact of Information Technology on Energy Consumption and Carbon Emissions". In *Ubiquity* 2015.June (June 2015).

[80] P. D. O. Castro, C. Akel, E. Petit, M. Popov, and W. Jalby. "CERE: LLVM-Based Codelet Extractor and REplayer for Piecewise Benchmarking and Optimization". In *ACM Trans. Archit. Code Optim. (TACO)* 12.1 (Apr. 2015), 6:1–6:24.

[79] K.-J. Stol and B. Fitzgerald. "Theory-Oriented Software Engineering". In *Science of Computer Programming* 101 (Apr. 2015).

[78] M. Schweinberger, M. Petrescu-Prahova, and D. Q. Vu. "Disaster response on September 11, 2001 through the lens of statistical network analysis". In *Social Networks* 37 (2014), pp. 42–55.

[77] E. Štrumbelj and I. Kononenko. "Explaining prediction models and individual predictions with feature contributions". In *Knowledge and information systems* 41.3 (2014), pp. 647–665.

[76] C. Zhang, A. Hindle, and D. M. German. "The Impact of User Choice on Energy Consumption". In *IEEE Software* 31.3 (2014), pp. 69–75.

[75] J. Bell and G. Kaiser. "Phosphor: Illuminating Dynamic Data Flow in Commodity JVMs". In *SIGPLAN Notices* 49.10 (Oct. 2014), pp. 83–101.

[74] T. Thüm, S. Apel, C. Kästner, I. Schaefer, and G. Saake. "A Classification and Survey of Analysis Strategies for Software Product Lines". In *ACM Comput. Surv. (CSUR)* 47.1 (June 2014), 6:1–6:45.

[73] J. M. Barnes, D. Garlan, and B. Schmerl. "Evolution Styles: Foundations and Models for Software Architecture Evolution". In *Softw. Syst. Model. (SoSyM)* 13.2 (May 2014), pp. 649–678.

[72] A. Narayanan and S. Vallor. "Why Software Engineering Courses Should Include Ethics Coverage". In *Commun. ACM* 57.3 (Mar. 2014), pp. 23–25.

[71] C. Yilmaz, S. Fouché, M. B. Cohen, A. Porter, G. Demiroz, and U. Koc. "Moving Forward with Combinatorial Interaction Testing". In *Computer* 47.2 (Feb. 2014), pp. 37–45.

[70] F. Hutter, L. Xu, H. H. Hoos, and K. Leyton-Brown. "Algorithm Runtime Prediction: Methods & Evaluation". In *Artif. Intell.* 206 (Jan. 2014), pp. 79–111.

[69] N. Esfahani, A. Elkhodary, and S. Malek. "A Learning-Based Framework for Engineering Feature-Oriented Self-Adaptive Software Systems". In *IEEE Transactions on Software Engineering* 39.11 (Nov. 2013), pp. 1467–1493.

[68] C. Yilmaz. "Test Case-Aware Combinatorial Interaction Testing". In *IEEE Trans. Softw. Eng. (TSE)* 39.5 (May 2013), pp. 684–706.

[67] J. Lawrance, C. Bogart, M. Burnett, R. Bellamy, K. Rector, and S. D. Fleming. "How Programmers Debug, Revisited: An Information Foraging Theory Perspective". In *IEEE Trans. Softw. Eng. (TSE)* 39.2 (Feb. 2013), pp. 197–215.

[66] A. Bergel, F. Bañados, R. Robbes, and W. Binder. "Execution profiling blueprints". In *Software: Practice and Experience* 42.9 (2012), pp. 1165–1192.

[65] F. Konietschke, L. A. Hothorn, and E. Brunner. "Rank-based multiple test procedures and simultaneous confidence intervals". In *Electronic Journal of Statistics* 6 (2012), pp. 738–759.

[64] N. Siegmund, M. Rosenmüller, M. Kuhlemann, C. Kästner, S. Apel, and G. Saake. "SPLConqueror: Toward Optimization of Non-functional Properties in Software Product Lines". In *Software Quality Journal* 20.3-4 (Sept. 2012), pp. 487–517.

[63] H. H. Hoos. “Programming by Optimization”. In *Commun. ACM* 55.2 (Feb. 2012), pp. 70–80.

[62] D. Giffhorn. “Advanced chopping of sequential and concurrent programs”. In *Software Quality Journal* 19.2 (2011), pp. 239–294.

[61] T. Ball, V. Levin, and S. K. Rajamani. “A Decade of Software Model Checking with SLAM”. In *Commun. ACM* 54.7 (July 2011), pp. 68–76.

[60] C. Scaffidi, S. Fleming, D. Piorkowski, M. Burnett, R. Bellamy, and J. Lawrance. “Unifying Software Engineering Methods and Tools: Principles and Patterns from Information Foraging”. In (May 2011).

[59] G. Bell and D. P. Siewiorek. “The Book Computer Structures: Thoughts After 40 Years”. In *IEEE Ann. Hist. Comput.* 33.2 (Apr. 2011), pp. 89–95.

[58] B. J. Garvin, M. B. Cohen, and M. B. Dwyer. “Evaluating Improvements to a Meta-heuristic Search for Constrained Interaction Testing”. In *Empirical Softw. Engg.* 16.1 (Feb. 2011), pp. 61–102.

[57] C. Nie and H. Leung. “A Survey of Combinatorial Testing”. In *ACM Comput. Surv. (CSUR)* 43.2 (Feb. 2011), 11:1–11:29.

[56] A. Bessey, K. Block, B. Chelf, A. Chou, B. Fulton, S. Hallem, C. Henri-Gros, A. Kamsky, S. McPeak, and D. Engler. “A Few Billion Lines of Code Later: Using Static Analysis to Find Bugs in the Real World”. In *Commun. ACM* 53.2 (Feb. 2010), pp. 66–75.

[55] P. Zave. “Modularity in distributed feature composition”. In *Software Requirements and Design: The Work of Michael Jackson* (2009), pp. 267–290.

[54] S. Becker, H. Koziolek, and R. Reussner. “The Palladio Component Model for Model-driven Performance Prediction”. In *J. Syst. Softw.* 82.1 (Jan. 2009), pp. 3–22.

[53] N. Ayewah, D. Hovemeyer, J. D. Morgenthaler, J. Penix, and W. Pugh. “Using Static Analysis to Find Bugs”. In *IEEE Softw.* 25.5 (Sept. 2008), pp. 22–29.

[52] M. B. Cohen, M. B. Dwyer, and J. Shi. “Constructing Interaction Test Suites for Highly-Configurable Systems in the Presence of Constraints: A Greedy Approach”. In *IEEE Trans. Softw. Eng. (TSE)* 34.5 (Sept. 2008), pp. 633–650.

[51] Y. Lei, R. Kacker, D. R. Kuhn, V. Okun, and J. Lawrence. “IPOG-IPOG-D: Efficient Test Generation for Multi-way Combinatorial Testing”. In *Softw. Test. Verif. Reliab.* 18.3 (Sept. 2008), pp. 125–148.

[50] A. Nhlabatsi, R. Laney, and B. Nuseibeh. “Feature Interaction: The Security Threat from within Software Systems”. In *Progress in Informatics* 5 (Mar. 2008), pp. 75–89.

[49] J. Donath. “Signals in Social Supernets”. In *J. Comp.-Med. Commun.* 13.1 (Oct. 2007), pp. 231–251.

[48] A. Georges, D. Buytaert, and L. Eeckhout. “Statistically Rigorous Java Performance Evaluation”. In *SIGPLAN Notices* 42.10 (Oct. 2007), pp. 57–76.

[47] S. Kounev. “Performance modeling and evaluation of distributed component-based systems using queueing Petri Nets”. In *IEEE Transactions on Software Engineering* 32.7 (2006), pp. 486–502.

[46] A. J. Ko, B. A. Myers, M. J. Coblenz, and H. H. Aung. “An Exploratory Study of How Developers Seek, Relate, and Collect Relevant Information during Software Maintenance Tasks”. In *IEEE Trans. Softw. Eng.* 32.12 (Dec. 2006), pp. 971–987.

[45] W. Visser. “Designing as construction of representations: A dynamic viewpoint in cognitive design research”. In *Human–Computer Interaction* 21.1 (Dec. 2006), pp. 103–152.

[44] B. Schmerl, J. Aldrich, D. Garlan, R. Kazman, and H. Yan. “Discovering Architectures from Running Systems”. In *IEEE Trans. Softw. Eng. (TSE)* 32.7 (July 2006), pp. 454–466.

[43] D. P. Siewiorek and P. Narasimhan. “Fault-tolerant architectures for space and avionics applications”. In *NASA Ames Research* (2005).

[42] B. Xu, J. Qian, X. Zhang, Z. Wu, and L. Chen. “A brief survey of program slicing”. In *ACM SIGSOFT Software Engineering Notes* 30.2 (2005), pp. 1–36.

[41] N. Ducheneaut. “Socialization in an Open Source Software Community: A Socio-Technical Analysis”. In *Comput. Supported Coop. Work* 14.4 (Aug. 2005), pp. 323–368.

[40] J. F. Maranzano, S. A. Rozsypal, G. H. Zimmerman, G. W. Warnken, P. E. Wirth, and D. M. Weiss. “Architecture Reviews: Practice and Experience”. In *IEEE Softw.* 22.2 (Mar. 2005), pp. 34–43.

[39] D. Garlan, S.-W. Cheng, A.-C. Huang, B. Schmerl, and P. Steenkiste. “Rainbow: Architecture-Based Self-Adaptation with Reusable Infrastructure”. In *Computer* 37.10 (Oct. 2004), pp. 46–54.

[38] D. Batory, J. N. Sarvela, and A. Rauschmayer. “Scaling Step-wise Refinement”. In *IEEE Trans. Softw. Eng. (TSE)* 30.6 (June 2004), pp. 355–371.

[37] D. P. Siewiorek, R. Chillarege, and Z. T. Kalbarczyk. “Reflections on Industry Trends and Experimental Research in Dependability”. In *IEEE Trans. Dependable Secur. Comput.* 1.2 (Apr. 2004), pp. 109–127.

[36] E. Bruneton, R. Lenglet, and T. Coupaye. “ASM: a code manipulation tool to implement adaptable systems”. In *Adaptable and extensible component systems* 30.19 (2002).

[35] J. Lerner and J. Tirole. “Some Simple Economics of Open Source”. In *The Journal of Industrial Economics* 50.2 (2002), pp. 197–234.

[34] G. Tassey. “The economic impacts of inadequate infrastructure for software testing”. In *National Institute of Standards and Technology, RTI Project* 7007.011 (2002).

[33] A. Mockus, R. T. Fielding, and J. D. Herbsleb. “Two Case Studies of Open Source Software Development: Apache and Mozilla”. In *ACM Trans. Softw. Eng. Methodol. (TOSEM)* 11.3 (July 2002), pp. 309–346.

[32] A. Zeller. “Yesterday, My Program Worked. Today, It Does Not. Why?” In *SIGSOFT Softw. Eng. Notes* 24.6 (Oct. 1999), pp. 253–267.

[31] G. C. Murphy and D. Notkin. “Reengineering with Reflexion Models: A Case Study”. In *Computer* 30.8 (Aug. 1997), pp. 29–36.

[30] J. Wang and C. J. Wu. “A hidden projection property of Plackett-Burman and related designs”. In *Statistica Sinica* (1995), pp. 235–250.

[29] D. Garlan, R. Allen, and J. Ockerbloom. “Architectural Mismatch: Why Reuse Is So Hard”. In *IEEE Softw.* 12.6 (Nov. 1995), pp. 17–26.

[28] R. E. Kraut and L. A. Streeter. “Coordination in Software Development”. In *Commun. ACM* 38.3 (Mar. 1995), pp. 69–81.

[27] V. R. B.-G. Caldiera and H. D. Rombach. “Goal question metric paradigm”. In *Encyclopedia of Software Engineering* 1 (1994), pp. 528–532.

[26] V. Tiwari, S. Malik, and A. Wolfe. “Power analysis of embedded software: a first step towards software power minimization”. In *IEEE Trans. Very Large Scale Integration (VLSI) Systems* 2.4 (Dec. 1994), pp. 437–445.

[25] N. Dahlbäck, A. Jönsson, and L. Ahrenberg. “Wizard of Oz studies—why and how”. In *Knowledge-Based Systems* 6.4 (1993), pp. 258–266.

[24] H. Agrawal and J. R. Horgan. “Dynamic program slicing”. In *ACM SIGPlan Notices* 25.6 (1990), pp. 246–256.

[23] G. D. Gopen and J. A. Swan. “The science of scientific writing”. In *American Scientist* 78.6 (1990), pp. 550–558.

[22] A. Hall. “Seven Myths of Formal Methods”. In *IEEE Softw.* 7.5 (Sept. 1990), pp. 11–19.

[21] B. Korel and J. Laski. “Dynamic program slicing”. In *Information processing letters* 29.3 (1988), pp. 155–163.

[20] B. Curtis, H. Krasner, and N. Iscoe. “A Field Study of the Software Design Process for Large Systems”. In *Commun. ACM* 31.11 (Nov. 1988), pp. 1268–1287.

[19] R. Johnson and B. Foote. “Designing Reusable Classes”. In *Journal of Object-Oriented Programming SIGS* 1.5 (June 1988), pp. 22–35.

[18] B. W. Boehm. “A Spiral Model of Software Development and Enhancement”. In *Computer* 21.5 (May 1988), pp. 61–72.

[17] W. S. Humphrey. “Characterizing the Software Process: A Maturity Framework”. In *IEEE Softw.* 5.2 (Mar. 1988), pp. 73–79.

[16] D. Harel. “Statecharts: A Visual Formalism for Complex Systems”. In *Sci. Comput. Program.* 8.3 (June 1987), pp. 231–274.

[15] F. P. Brooks Jr. “No Silver Bullet Essence and Accidents of Software Engineering”. In *Computer* 20.4 (Apr. 1987), pp. 10–19.

[14] E. J. Weyuker. “Axiomatizing Software Test Data Adequacy”. In *IEEE Trans. Softw. Eng. (TSE)* 12.12 (Dec. 1986), pp. 1128–1138.

[13] E. M. Clarke, E. A. Emerson, and A. P. Sistla. “Automatic Verification of Finite-state Concurrent Systems Using Temporal Logic Specifications”. In *ACM Trans. Program. Lang. Syst.* 8.2 (Apr. 1986), pp. 244–263.

[12] P. J. Fleming and J. J. Wallace. “How Not to Lie with Statistics: The Correct Way to Summarize Benchmark Results”. In *Commun. ACM* 29.3 (Mar. 1986), pp. 218–221.

[11] M. Shaw. “The impact of abstraction concerns on modern programming languages”. In *Proc. of the IEEE* 68.9 (Apr. 1980), pp. 1119–1130.

[10] V. Chvatal. “A Greedy Heuristic for the Set-Covering Problem”. In *Math. Oper. Res.* 4.3 (Aug. 1979), pp. 233–235.

[9] R. A. De Millo, R. J. Lipton, and A. J. Perlis. “Social Processes and Proofs of Theorems and Programs”. In *Commun. ACM* 22.5 (May 1979), pp. 271–280.

[8] J. C. King. “Symbolic Execution and Program Testing”. In *Commun. ACM* 19.7 (July 1976), pp. 385–394.

[7] W. A. Wulf, R. L. London, and M. Shaw. “An Introduction to the Construction and Verification of Alphard Programs”. In *IEEE Trans. Softw. Eng. (TSE)* 2.4 (July 1976), pp. 253–265.

[6] F. DeRemer and H. Kron. “Programming-in-the-Large Versus Programming-in-the-Small”. In *IEEE Trans. Softw. Eng. (TSE)* SE-2.2 (June 1976), pp. 80–86.

[5] H. W. J. Rittel and M. M. Webber. “Dilemmas in a general theory of planning”. In *Policy Sciences* 4.2 (June 1973), pp. 155–169.

[4] C. A. R. Hoare. “Proof of correctness of data representations”. In *Acta Informatica* 1.4 (Dec. 1972), pp. 271–281.

[3] D. L. Parnas. “On the Criteria to Be Used in Decomposing Systems into Modules”. In *Commun. ACM* 15.12 (Dec. 1972), pp. 1053–1058.

[2] C. A. R. Hoare. “An Axiomatic Basis for Computer Programming”. In *Commun. ACM* 12.10 (Oct. 1969), pp. 576–580.

[1] M. E. Conway. “How do committees invent?” In *Datamation* 14.4 (1968), pp. 28–31.

Refereed Conference Publications

[307] M. Kim, T. Zimmermann, R. DeLine, and A. Begel. "The Emerging Role of Data Scientists on Software Development Teams". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Austin, TX, USA: ACM, pp. 96–107.

[306] M. Velez, P. Jamshidi, N. Siegmund, S. Apel, and C. Kästner. "White-Box Analysis over Machine Learning: Modeling Performance of Configurable Systems". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Madrid, Spain: IEEE, May 2021. (23% acceptance rate).

[305] M. Weber, S. Apel, and N. Siegmund. "White-Box Performance-Influence Models: A Profiling and Learning Approach". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Madrid, Spain: IEEE, May 2021.

[304] T. Zhang, Z. Chen, Y. Zhu, P. Vaithilingam, X. Wang, and E. Glassman. "Interpretable Program Synthesis". In *Proc. Conf. Human Factors in Computing Systems (CHI)*. Apr. 2021.

[303] M. Copik, A. Calotoiu, T. Grosser, N. Wicki, F. Wolf, and T. Hoefler. "Extracting Clean Performance Models from Tainted Programs". In *Proc Symposium Principles and Practice of Parallel Programming (PPoPP)*. Seoul, South Korea: ACM, Feb. 2021.

[302] M. Velez, P. Jamshidi, N. Siegmund, S. Apel, and C. Kästner. "White-box Analysis over Machine Learning: Modeling Performance of Configurable Systems". In *Under review*. 2020.

[301] Z. Chen, Y. Cao, Y. Liu, H. Wang, T. Xie, and X. Liu. "Understanding Challenges in Deploying Deep Learning Based Software: An Empirical Study". In *Proc. Int'l Symp. Foundations of Software Engineering (FSE)*. Sacramento, CA, USA, Nov. 2020.

[300] H. He, Z. Jia, S. Li, E. Xu, T. Yu, Y. Yu, J. Wang, and X. Liao. "CP-Detector: Using Configuration-related PerformanceProperties to Expose Performance Bugs". In *Proc. Int'l Conf. Automated Software Engineering (ASE)*. Nov. 2020.

[299] Y. Huang, K. Leach, Z. Sharafi, N. McKay, T. Santander, and W. Weimer. "Biases and Differences in Code Review Using Medical Imaging and Eye-Tracking: Genders, Humans, and Machines". In *Proc. Int'l Symp. Foundations of Software Engineering (FSE)*. New York, NY, USA: ACM, Nov. 2020, pp. 456–468.

[298] H. Huijgens, A. Rastogi, E. Mulders, G. Gousios, and A. Deursen. "Questions for Data Scientists in Software Engineering: A Replication". In *Proc. Europ. Software Engineering Conf. Foundations of Software Engineering (ESEC/FSE)*. New York, NY, USA: ACM, Nov. 2020.

[297] I. Pashchenko, D.-L. Vu, and F. Massacci. "A Qualitative Study of Dependency Management and Its Security Implications (To be appear in ACM CCS 2020)". In *Conf. Cmputer and Communications Security (CCS)*. New York, NY, USA: ACM, Nov. 2020.

[296] A. P. Koenzen, N. A. Ernst, and M. D. Storey. "Code Duplication and Reuse in Jupyter Notebooks". In *Symposium Visual Languages and Human-Centric Computing (VL/HCC)*. Aug. 2020, pp. 1–9.

[295] F. Brown, J. Renner, A. Nötzli, S. Lerner, H. Shacham, and D. Stefan. "Towards a verified range analysis for JavaScript JITs". In *Proc. Conf. Programming Language Design and Implementation (PLDI)*. London, UK: ACM, June 2020.

[294] A. M. Brittany Johnson Yuriy Brun. "Causal Testing: Understanding Defects' Root Causes". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Seoul, South Korea: ACM, May 2020.

[293] J. Hu, J. Joung, M. Jacobs, K. Gajos, and M. Seltzer. "Revealing Injection Vulnerabilities by Leveraging Existing Tests". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Seoul, South Korea: ACM, May 2020.

[292] C. H. Katherine Hough Gebrehiwet Welearegai and J. Bell. "Revealing Injection Vulnerabilities by Leveraging Existing Tests". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Seoul, South Korea: ACM, May 2020.

[291] D. van der Linden, P. Anthonysamy, B. Nuseibeh, T. Tun, M. Petre, M. Levine, J. Towse, and A. Rashid. "Schrödinger's Security: Opening the Box on App Developers'Security Rationale". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Seoul, South Korea: ACM, May 2020.

[290] C.-A. Staicu, M. T. Torp, M. Schäfer, A. Møller, and M. Pradel. "Extracting Taint Specifications for JavaScript Libraries". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Seoul, South Korea: ACM, May 2020.

[289] S. Chattopadhyay, I. Prasad, A. Z. Henley, A. Sarma, and T. Barik. "What's Wrong with Computational Notebooks? Pain Points, Needs, and Design Opportunities". In *Proc. Conf. Human Factors in Computing Systems (CHI)*. Apr. 2020.

[288] C. Li, S. Wang, H. Hoffmann, and S. Lu. "Statically Inferring Performance Properties of Software Configurations". In *Proc. European Conf. Computer Systems (EuroSys)*. Heraklion, Greece: ACM, Apr. 2020.

[287] A. Head, F. Hohman, T. Barik, S. M. Drucker, and R. DeLine. "Managing Messes in Computational Notebooks". In *Proc. Conf. Human Factors in Computing Systems (CHI)*. Glasgow, Scotland Uk: ACM, 2019, 270:1–270:12.

[286] O. Keyes, J. Hutson, and M. Durbin. "A Mulching Proposal: Analysing and Improving an Algorithmic System for Turning the Elderly into High-Nutrient Slurry". In Glasgow, Scotland: ACM, 2019, pp. 1–11.

[285] D. He, H. Li, L. Wang, H. Meng, H. Zheng, J. Liu, S. Hu, L. Li, and J. Xue. "Performance-Boosting Sparsification of the IFDS Algorithm with Applications to Taint Analysis". In *Proc. Int'l Conf. Automated Software Engineering (ASE)*. Nov. 2019, pp. 267–279.

[284] H. Ha and H. Zhang. "Performance-Influence Model for Highly Configurable Software with Fourier Learning and Lasso Regression". In *Proc. Int'l Conf. Software Maintance and Evolution (ICSME)*. Sept. 2019, pp. 470–480.

[283] J. P. Sandoval Alcocer, F. Beck, and A. Bergel. "Performance Evolution Matrix: Visualizing Performance Variations Along Software Versions". In *Conf. Software Visualization (VISSOFT)*. IEEE, Sept. 2019, pp. 1–11.

[282] T. Durieux, F. Madeiral, M. Martinez, and R. Abreu. “Empirical Review of Java Program Repair Tools: A Large-Scale Experiment on 2,141 Bugs and 23,551 Repair Attempts”. In *Proc. Europ. Software Engineering Conf. Foundations of Software Engineering (ESEC/FSE)*. Tallinn, Estonia: ACM, Aug. 2019, pp. 302–313.

[281] P. H. Nguyen, H. Song, F. Chauvel, R. Muller, S. Boyar, and E. Levin. “Using Microservices for Non-intrusive Customization of Multi-tenant SaaS”. In *Proc. Europ. Software Engineering Conf. Foundations of Software Engineering (ESEC/FSE)*. Tallinn, Estonia: ACM, Aug. 2019, pp. 905–915.

[280] S. Amershi, A. Begel, C. Bird, R. DeLine, H. Gall, E. Kamar, N. Nagappan, B. Nushi, and T. Zimmermann. “Software Engineering for Machine Learning: A Case Study”. In *Proc. Int’l Conf. Software Engineering: Software Engineering in Practice*. Montreal, Quebec, Canada: IEEE, May 2019, pp. 291–300.

[279] H. Ha and H. Zhang. “DeepPerf: Performance Prediction for Configurable Software with Deep Sparse Neural Network”. In *Proc. Int’l Conf. Software Engineering (ICSE)*. Montreal, Quebec, Canada: IEEE, May 2019, pp. 1095–1106.

[278] V. J. Hellendoorn, S. Proksch, H. C. Gall, and A. Bacchelli. “When Code Completion Fails: A Case Study on Real-World Completions”. In *Proc. Int’l Conf. Software Engineering (ICSE)*. Montreal, Quebec, Canada: IEEE, May 2019, pp. 960–970.

[277] K. Holstein, J. Wortman Vaughan, H. Daumé, M. Dudik, and H. Wallach. “Improving Fairness in Machine Learning Systems: What Do Industry Practitioners Need?” In *Proc. Conference Human Factors in Computing Systems (CHI)*. Glasgow, Scotland, UK: ACM, May 2019, pp. 1–16.

[276] C. Kaltenecker, A. Grebhahn, N. Siegmund, J. Guo, and S. Apel. “Distance-Based Sampling of Software Configuration Spaces”. In *Proc. Int’l Conf. Software Engineering (ICSE)*. Montreal, Quebec, Canada: IEEE, May 2019.

[275] T. Lopez, H. Sharp, T. Tun, A. K. Bandara, M. Levine, and B. Nuseibeh. “"Hopefully We Are Mostly Secure": Views on Secure Code in Professional Practice”. In *Proc. Int’l Workshop Cooperative and Human Aspects of Software Engineering (CHASE)*. Montreal, Quebec, Canada: IEEE, May 2019, pp. 61–68.

[274] B. J. Muscedere, R. Hackman, D. Anbarnam, J. M. Atlee, I. J. Davis, and M. W. Godfrey. “Detecting Feature-Interaction Symptoms in Automotive Software using Lightweight Analysis”. In *Int’l Conf. Software Analysis, Evolution and Reengineering (SANER)*. Feb. 2019, pp. 175–185.

[273] E. Bodden. “Self-adaptive Static Analysis”. In *Proc. Int’l Conf. Software Engineering (ICSE): New Ideas and Emerging Results*. Gothenburg, Sweden: ACM, 2018, pp. 45–48.

[272] J. Cito, P. Leitner, C. Bosshard, M. Knecht, G. Mazlami, and H. C. Gall. “PerformanceHat: Augmenting Source Code with Runtime Performance Traces in the IDE”. In *Proc. Int’l Conf. Software Engineering: Companion Proceeedings*. Gothenburg, Sweden: ACM, 2018, pp. 41–44.

[271] C. Lemieux, R. Padhye, K. Sen, and D. Song. "Perffuzz: Automatically generating pathological inputs". In *Proc. Int'l Symp. Software Testing and Analysis (ISSTA)*. Amsterdam, Netherlands: ACM, 2018, pp. 254–265.

[270] L. Qiu, Y. Wang, and J. Rubin. "Analyzing the Analyzers: FlowDroid/IccTA, AmanDroid, and DroidSafe". In *Proc. Int'l Symp. Software Testing and Analysis (ISSTA)*. Amsterdam, Netherlands: ACM, 2018, pp. 176–186.

[269] X. Zhang, A. Solar-Lezama, and R. Singh. "Interpreting Neural Network Judgments via Minimal, Stable, and Symbolic Corrections". In *Proc. Int'l Conf. Neural Information Processing Systems (NeurIPS)*. Montréal, Canada: Curran Associates Inc., Dec. 2018, pp. 4879–4890.

[268] P. Jamshidi, M. Velez, C. Kästner, and N. Siegmund. "Learning to Sample: Exploiting Similarities Across Environments to Learn Performance Models for Configurable Systems". In *Proc. Int'l Symp. Foundations of Software Engineering (FSE)*. Lake Buena Vista, FL, USA: ACM, Nov. 2018, pp. 71–82. (21% acceptance rate).

[267] S. Ma, Y. Liu, W.-C. Lee, X. Zhang, and A. Grama. "MODE: Automated Neural Network Model Debugging via State Differential Analysis and Input Selection". In *Proc. Europ. Software Engineering Conf. Foundations of Software Engineering (ESEC/FSE)*. Lake Buena Vista, FL, USA: ACM, Nov. 2018, pp. 175–186.

[266] F. Pauck, E. Bodden, and H. Wehrheim. "Do Android Taint Analysis Tools Keep Their Promises?" In *Proc. Int'l Symp. Foundations of Software Engineering (FSE)*. Lake Buena Vista, FL, USA: ACM, Nov. 2018, pp. 331–341.

[265] S. E. Chasins, M. Mueller, and R. Bodik. "Rousillon: Scraping Distributed Hierarchical Web Data". In *Proc. Symposium User Interface Software and Technology (UIST)*. Berlin, Germany: ACM, Oct. 2018, pp. 963–975.

[264] J. Businge, O. Moses, S. Nadi, E. Bainomugisha, and T. Berger. "Clone-Based Variability Managementin the Android Ecosystem". In *Proc. Int'l Conf. Software Maintance and Evolution (ICSME)*. Madrid, Spain, Sept. 2018.

[263] M. Cashman, M. B. Cohen, P. Ranjan, and R. W. Cottingham. "Navigating the Maze: The Impact of Configurability in Bioinformatics Software". In *Proc. Int'l Conf. Automated Software Engineering (ASE)*. Montpellier, France: ACM, Sept. 2018, pp. 757–767.

[262] X. Han, T. Yu, and D. Lo. "PerfLearner: Learning from Bug Reports to Understand and Generate Performance Test Frames". In *Proc. Int'l Conf. Automated Software Engineering (ASE)*. Montpellier, France: ACM, Sept. 2018, pp. 17–28.

[261] M. Mukelabai, D. Nešić, S. Maro, T. Berger, and J.-P. Steghöfer. "Tackling Combinatorial Explosion: A Study of Industrial Needs and Practices for Analyzing Highly Configurable Systems". In *Proc. Int'l Conf. Automated Software Engineering (ASE)*. Montpellier, France: ACM, Sept. 2018, pp. 155–166.

[260] R. Just, C. Parnin, I. Drosos, and M. D. Ernst. "Comparing Developer-provided to User-provided Tests for Fault Localization and Automated Program Repair". In *Proc. Int'l Symp. Software Testing and Analysis (ISSTA)*. Amsterdam, Netherlands: ACM, July 2018, pp. 287–297.

[259] M. T. Ribeiro, S. Singh, and C. Guestrin. “Semantically Equivalent Adversarial Rules for Debugging NLP models”. In *Proc. Meet. Association for Computational Linguistics*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 856–865.

[258] E. Pariwono, D. Chiba, M. Akiyama, and T. Mori. “Don’T Throw Me Away: Threats Caused by the Abandoned Internet Resources Used by Android Apps”. In *Proc. Asia Conf. Computer and Communications Security (ASIACCS)*. Incheon, Republic of Korea: ACM, June 2018, pp. 147–158.

[257] M. Raghothaman, S. Kulkarni, K. Heo, and M. Naik. “Interactive program reasoning using Bayesian inference”. In *Proc. Conf. Programming Language Design and Implementation (PLDI)*. Philadelphia, PA, USA: ACM, June 2018.

[256] A. Alami, Y. Dittrich, and A. Wąsowski. “Influencers of Quality Assurance in an Open Source Community”. In *Proc. Int’l Workshop Cooperative and Human Aspects of Software Engineering (CHASE)*. Gothenburg, Sweden: ACM, May 2018, pp. 61–68.

[255] V. Borsotti. “Barriers to Gender Diversity in Software Development Education: Actionable Insights from a Danish Case Study”. In *Proc. Int’l Conf. Software Engineering: Software Engineering Education and Training (ICSE-SEET)*. Gothenburg, Sweden: ACM, May 2018, pp. 146–152.

[254] C. Jaspan, M. Jorde, A. Knight, C. Sadowski, E. K. Smith, C. Winter, and E. Murphy-Hill. “Advantages and Disadvantages of a Monolithic Repository: A Case Study at Google”. In *Proceedings Int’l Conf. Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. Gothenburg, Sweden: ACM, May 2018, pp. 225–234.

[253] J. Middleton, E. Murphy-Hill, D. Green, A. Meade, R. Mayer, D. White, and S. McDonald. “Which Contributions Predict Whether Developers Are Accepted into Github Teams”. In *Proc. Int’l Conf. Mining Software Repositories*. Gothenburg, Sweden: ACM, May 2018, pp. 403–413.

[252] L. Sousa, A. Oliveira, W. Oizumi, S. Barbosa, A. Garcia, J. Lee, M. Kalinowski, R. de Mello, B. Fonseca, R. Oliveira, C. Lucena, and R. Paes. “Identifying Design Problems in the Source Code: A Grounded Theory”. In *Proc. Int’l Conf. Software Engineering (ICSE)*. Gothenburg, Sweden: ACM, May 2018, pp. 921–931.

[251] I. Steinmacher, G. Pinto, I. S. Wiese, and M. A. Gerosa. “Almost There: A Study on Quasi-contributors in Open Source Software Projects”. In *Proc. Int’l Conf. Software Engineering (ICSE)*. Gothenburg, Sweden: ACM, May 2018, pp. 256–266.

[250] M. Wen, J. Chen, R. Wu, D. Hao, and S. Cheung. “Context-Aware Patch Generation for Better Automated Program Repair”. In *Proc. Int’l Conf. Software Engineering (ICSE)*. May 2018, pp. 1–11.

[249] D. Ford, K. Lustig, J. Banks, and C. Parnin. “"We Don’T Do That Here": How Collaborative Editing with Mentors Improves Engagement in Social Q&A Communities”. In *Proc. Conf. Human Factors in Computing Systems (CHI)*. Montreal, QC, Canada: ACM, Apr. 2018, 608:1–608:12.

[248] E. Glassman, T. Zhangk, B. Hartmann, and M. Kim. "Visualizing API Usage Examples at Scale". In *Proc. Conf. Human Factors in Computing Systems (CHI)*. Montreal, Canada: ACM, Apr. 2018.

[247] M. Marathe and K. Toyama. "Semi-Automated Coding for Qualitative Research: A User-Centered Inquiry and Initial Prototypes". In *Proc. Conference Human Factors in Computing Systems (CHI)*. New York, NY, USA: ACM, Apr. 2018, pp. 1–12.

[246] S. Wang, C. Li, H. Hoffmann, S. Lu, W. Sentosa, and A. I. Kistijantoro. "Understanding and Auto-Adjusting Performance-Sensitive Configurations". In *Proc. Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS)*. Williamsburg, VA, USA: ACM, Mar. 2018, pp. 154–168.

[245] L. N. Q. Do, K. Ali, B. Livshits, E. Bodden, J. Smith, and E. Murphy-Hill. "Just-in-time Static Analysis". In *Proc. Int'l Symp. Software Testing and Analysis (ISSTA)*. Santa Barbara, CA, USA: ACM, 2017, pp. 307–317.

[244] D.-J. Munoz. "Achieving energy efficiency using a software product line approach". In *Proc. Int'l Conf. Systems and Software Product Line*. 2017, pp. 131–138.

[243] M. Sayagh, N. Kerzazi, and B. Adams. "On Cross-stack Configuration Errors". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Buenos Aires, Argentina: IEEE, 2017, pp. 255–265.

[242] E. Derr, S. Bugiel, S. Fahl, Y. Acar, and M. Backes. "Keep Me Updated: An Empirical Study of Third-Party Library Updatability on Android". In *Proc. Conf. Computer and Communications Security (CCS)*. Dallas, TX, USA: ACM, Oct. 2017, pp. 2187–2200.

[241] P. Jamshidi, N. Siegmund, M. Velez, C. Kästner, A. Patel, and Y. Agarwal. "Transfer Learning for Performance Modeling of Configurable Systems: An Exploratory Analysis". In *Proc. Int'l Conf. Automated Software Engineering (ASE)*. Urbana-Champaign, IL, USA: ACM, Oct. 2017. (21% acceptance rate).

[240] S. Scalabrino, G. Bavota, C. Vendome, M. Linares-Vásquez, D. Poshyvanyk, and R. Oliveto. "Automatically Assessing Code Understandability: How Far Are We?" In *Proc. Int'l Conf. Automated Software Engineering (ASE)*. Singapore, Singapore: ACM, Oct. 2017.

[239] R. Abdalkareem, O. Nourry, S. Wehaibi, S. Mujahid, and E. Shihab. "Why Do Developers Use Trivial Packages? An Empirical Case Study on npm". In *Proc. Europ. Software Engineering Conf. Foundations of Software Engineering (ESEC/FSE)*. Paderborn, Germany: ACM, Sept. 2017.

[238] O. Chaparro, J. Lu, F. Zampetti, L. Moreno, M. Di Penta, A. Marcus, G. Bavota, and V. Ng. "Detecting Missing Information in Bug Descriptions". In *Proc. Europ. Software Engineering Conf. Foundations of Software Engineering (ESEC/FSE)*. Paderborn, Germany: ACM, Sept. 2017, pp. 396–407.

[237] J. Coelho and M. T. Valente. "Why Modern Open Source Projects Fail". In *Proc. Europ. Software Engineering Conf. Foundations of Software Engineering (ESEC/FSE)*. Paderborn, Germany: ACM, Sept. 2017, pp. 186–196.

[236] D. Garbervetsky, E. Zoppi, and B. Livshits. “Toward Full Elasticity in Distributed Static Analysis: The Case of Callgraph Analysis”. In *Proc. Europ. Software Engineering Conf. Foundations of Software Engineering (ESEC/FSE)*. Paderborn, Germany: ACM, Sept. 2017, pp. 442–453.

[235] D. Gopstein, J. Iannacone, Y. Yan, L. DeLong, Y. Zhuang, M. K.-C. Yeh, and J. Cappos. “Understanding Misunderstandings in Source Code”. In *Proc. Europ. Software Engineering Conf. Foundations of Software Engineering (ESEC/FSE)*. Paderborn, Germany: ACM, Sept. 2017, pp. 129–139.

[234] J. Oh, D. Batory, M. Myers, and N. Siegmund. “Finding Near-optimal Configurations in Product Lines by Random Sampling”. In *Proc. Europ. Software Engineering Conf. Foundations of Software Engineering (ESEC/FSE)*. Paderborn, Germany: ACM, Sept. 2017, pp. 61–71.

[233] Y. Zhu, J. Liu, M. Guo, Y. Bao, W. Ma, Z. Liu, K. Song, and Y. Yang. “BestConfig: Tapping the Performance Potential of Systems via Automatic Configuration Tuning”. In *Proc. Symposium Cloud Computing (SoCC)*. Santa Clara, CA, USA: ACM, Sept. 2017, pp. 338–350.

[232] X.-B. D. Le, D.-H. Chu, D. Lo, C. Le Goues, and W. Visser. “S3: Syntax- and Semantic-Guided Repair Synthesis via Programming by Examples”. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*. Paderborn, Germany: ACM, Aug. 2017, pp. 593–604.

[231] V. Nair, T. Menzies, N. Siegmund, and S. Apel. “Using Bad Learners to Find Good Configurations”. In *Proc. Europ. Software Engineering Conf. Foundations of Software Engineering (ESEC/FSE)*. ESEC/FSE 2017. Paderborn, Germany: ACM, Aug. 2017, pp. 257–267.

[230] R. Just, B. Kurtz, and P. Ammann. “Inferring Mutant Utility from Program Context”. In *Proc. Int’l Symp. Software Testing and Analysis (ISSTA)*. Santa Barbara, CA, USA: ACM, July 2017, pp. 284–294.

[229] S. Mostafa, X. Wang, and T. Xie. “PerfRanker: Prioritization of Performance Regression Tests for Collection-intensive Software”. In *Proc. Int’l Symp. Software Testing and Analysis (ISSTA)*. Santa Barbara, CA, USA: ACM, July 2017, pp. 23–34.

[228] B. Wang, Y. Xiong, Y. Shi, L. Zhang, and D. Hao. “Faster Mutation Analysis via Equivalence Modulo States”. In *Proc. Int’l Symp. Software Testing and Analysis (ISSTA)*. Santa Barbara, CA, USA: ACM, July 2017, pp. 295–306.

[227] E. S. Andreasen, A. Møller, and B. B. Nielsen. “Systematic Approaches for Increasing Soundness and Precision of Static Analyzers”. In *Proc. Int’l Workshop State Of the Art in Program Analysis (SOAP)*. Barcelona, Spain: ACM, June 2017, pp. 31–36.

[226] P. Jamshidi, M. Velez, C. Kästner, N. Siegmund, and P. Kawthekar. “Transfer Learning for Improving Model Predictions in Highly Configurable Software”. In *Proc. Int’l Symp. Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. Buenos Aires, Argentina: IEEE, May 2017, pp. 31–41. (23% acceptance rate).

[225] J. Melo, F. B. Narcizo, D. W. Hansen, C. Brabrand, and A. Wasowski. "Variability through the Eyes of the Programmer". In *Proc. Int'l Conference Program Comprehension (ICPC)*. Buenos Aires, Argentina: IEEE, May 2017, pp. 34–44.

[224] T. Sedano, P. Ralph, and C. Péraire. "Software Development Waste". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Buenos Aires, Argentina: IEEE, May 2017, pp. 130–140.

[223] L. Song and S. Lu. "Performance Diagnosis for Inefficient Loops". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Buenos Aires, Argentina: IEEE, May 2017, pp. 370–380.

[222] S. Souto, M. d'Amorim, and R. Gheyi. "Balancing Soundness and Efficiency for Practical Testing of Configurable Systems". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Buenos Aires, Argentina: IEEE, May 2017, pp. 632–642.

[221] P. Weisenburger, M. Luthra, B. Koldehofe, and G. Salvaneschi. "Quality-aware Runtime Adaptation in Complex Event Processing". In *Proc. Int'l Symp. Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. Buenos Aires, Argentina: IEEE, May 2017, pp. 140–151.

[220] M. M. u. Alam, T. Liu, G. Zeng, and A. Muzahid. "SyncPerf: Categorizing, Detecting, and Diagnosing Synchronization Performance Bugs". In *Proc. European Conference on Computer Systems (EuroSys)*. Belgrade, Serbia: ACM, Apr. 2017, pp. 298–313.

[219] P. Valov, J.-C. Petkovich, J. Guo, S. Fischmeister, and K. Czarnecki. "Transferring Performance Prediction Models Across Different Hardware Platforms". In *Proc. Int'l Conf. on Performance Engineering (ICPE)*. L'Aquila, Italy: ACM, Apr. 2017, pp. 39–50.

[218] M. Arciniegas-Mendez, A. Zagalsky, M.-A. Storey, and A. F. Hadwin. "Using the Model of Regulation to Understand Software Development Collaboration Practices and Tool Support". In *Proc. Conf. Computer Supported Cooperative Work (CSCW)*. Portland, OR, USA: ACM, Feb. 2017, pp. 1049–1065.

[217] Y. Tang and H. Leung. "StiCProb: A novel feature mining approach using conditional probability". In *Int'l Conf. Software Analysis, Evolution and Reengineering (SANER)*. Klagenfurt, Austria: IEEE, Feb. 2017, pp. 45–55.

[216] C. Luckeneder, M. Rathmair, and H. Kaindl. "Investigating and Coordinating Safety-critical Feature Interactions in Automotive Systems Using Simulation". In *Hawaii Int'l Conf. System Sciences (HICSS)*. Hilton Waikoloa Village, HI, USA, Jan. 2017, p. 10.

[215] Q. Zhang and Z. Su. "Context-sensitive Data-dependence Analysis via Linear Conjunctive Language Reachability". In *Proc. Symp. Principles of Programming Languages (POPL)*. Paris, France: ACM, Jan. 2017, pp. 344–358.

[214] M. Christakis and C. Bird. "What Developers Want and Need from Program Analysis: An Empirical Study". In *Proc. Int'l Conf. Automated Software Engineering (ASE)*. Singapore, Singapore: ACM, 2016, pp. 332–343.

[213] K.-J. Stol, P. Ralph, and B. Fitzgerald. "Grounded Theory in Software Engineering Research: A Critical Review and Guidelines". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Austin, TX, USA: ACM, 2016, pp. 120–131.

[212] Y. Wang, H. Zhang, and A. Rountev. "On the Unsoundness of Static Analysis for Android GUIs". In *Proc. Int'l Workshop State Of the Art in Program Analysis (SOAP)*. Santa Barbara, CA, USA: ACM, 2016, pp. 18–23.

[211] P. Tsankov, M. Pistoia, O. Tripp, M. Vechev, and P. Ferrara. "FASE: Functionality-aware Security Enforcement". In *Proc. Conf. Computer Security Applications (ACSAC)*. Los Angeles, CA, USA: ACM, Dec. 2016, pp. 471–483.

[210] C. Bogart, C. Kästner, J. Herbsleb, and F. Thung. "How to Break an API: Cost Negotiation and Community Values in Three Software Ecosystems". In *Proc. Int'l Symp. Foundations of Software Engineering (FSE)*. Seattle, WA, USA: ACM, Nov. 2016, pp. 109–120.

[209] E. Kang, A. Milicevic, and D. Jackson. "Multi-representational Security Analysis". In *Proc. Int'l Symp. Foundations of Software Engineering (FSE)*. Seattle, WA, USA: ACM, Nov. 2016, pp. 181–192.

[208] T. Nguyen, T. Koc, J. Cheng, J. S. Foster, and A. A. Porter. "iGen Dynamic Interaction Inference for Configurable Software". In *Proc. Int'l Symp. Foundations of Software Engineering (FSE)*. Seattle, WA, USA: IEEE, Nov. 2016.

[207] T. Xu, X. Jin, P. Huang, Y. Zhou, S. Lu, L. Jin, and S. Pasupathy. "Early Detection of Configuration Errors to Reduce Failure Damage". In *Proc. Conf.Operating Systems Design and Implementation (OSDI)*. Savannah, GA, USA: USENIX Association, Nov. 2016, pp. 619–634.

[206] L. Braz, R. Gheyi, M. Mongiovi, M. Ribeiro, F. Medeiros, and L. Teixeira. "A Change-centric Approach to Compile Configurable Systems with #Ifdefs". In *Proc. Int'l Conf. Generative Programming and Component Engineering (GPCE)*. Amsterdam, Netherlands: ACM, Oct. 2016, pp. 109–119.

[205] S. P. De Rosso and D. Jackson. "Purposes, Concepts, Misfits, and a Redesign of Git". In *Proc. Int'l Conf. Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)*. Amsterdam, Netherlands: ACM, Oct. 2016, pp. 292–310.

[204] Z. Dong, A. Andrzejak, D. Lo, and D. Costa. "ORPLocator: Identifying Read Points of Configuration Options via Static Analysis". In *Proc. Int'l Symposium Software Reliability Engineering (ISSRE)*. IEEE, Oct. 2016, pp. 185–195.

[203] M. Al-Hajjaji, S. Krieter, T. Thüm, M. Lochau, and G. Saake. "IncLing: Efficient Product-line Testing Using Incremental Pairwise Sampling". In *Proc. Int'l Conf. Generative Programming and Component Engineering (GPCE)*. Amsterdam, Netherlands: ACM, Oct. 2016, pp. 144–155.

[202] V. Rothberg, C. Dietrich, A. Ziegler, and D. Lohmann. "Towards Scalable Configuration Testing in Variable Software". In *Proc. Int'l Conf. Generative Programming and Component Engineering (GPCE)*. Amsterdam, Netherlands: ACM, Oct. 2016, pp. 156–167.

[201] X. Han and T. Yu. "An Empirical Study on Performance Bugs for Highly Configurable Software Systems". In *Proc. Int'l Symposium Empirical Software Engineering and Measurement (ESEM)*. Ciudad Real, Spain: ACM, Sept. 2016, 23:1–23:10.

[200] P. Jamshidi and G. Casale. “An Uncertainty-Aware Approach to Optimal Configuration of Stream Processing Systems”. In *Int’l Symp. Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*. London, UK, Sept. 2016, pp. 39–48.

[199] J. Meinicke, C.-P. Wong, C. Kästner, T. Thüm, and G. Saake. “On Essential Configuration Complexity: Measuring Interactions in Highly-configurable Systems”. In *Proc. Int’l Conf. Automated Software Engineering (ASE)*. Singapore, Singapore: ACM, Sept. 2016, pp. 483–494.

[198] L. Wei, Y. Liu, and S.-C. Cheung. “Taming Android Fragmentation: Characterizing and Detecting Compatibility Issues for Android Apps”. In *Proc. Int’l Conf. Automated Software Engineering (ASE)*. Singapore, Singapore: ACM, Sept. 2016, pp. 226–237.

[197] M. T. Ribeiro, S. Singh, and C. Guestrin. “"Why Should I Trust You?": Explaining the Predictions of Any Classifier”. In *Proc. Int’l Conf. Knowledge Discovery and Data Mining (KDD)*. San Francisco, CA, USA: ACM, Aug. 2016, pp. 1135–1144.

[196] M. Velez, J. Sawin, A. Ingerson, and D. Chiu. “Improving Bitmap Execution Performance Using Column-Based Metadata”. In *Int’l Conf. Future Internet of Things and Cloud (FiCloud)*. Vienna, Austria: IEEE, Aug. 2016, pp. 371–378. (30% acceptance rate).

[195] R. Jabbarvand, A. Sadeghi, H. Bagheri, and S. Malek. “Energy-aware Test-suite Minimization for Android Apps”. In *Proc. Int’l Symp. Software Testing and Analysis (ISSTA)*. Saarbrücken, Germany: ACM, July 2016, pp. 425–436.

[194] P. Kawthekar and C. Kästner. “Sensitivity Analysis For Building Evolving & Adaptive Robotic Software”. In *Proc. Workshop Autonomous Mobile Service Robots (WSR)*. New York, NY, USA, July 2016.

[193] J. Toman and D. Grossman. “Legato: An At-Most-Once Analysis with Applications to Dynamic Configuration Updates”. In *European Conf. Object-Oriented Programming (ECOOP)*. Amsterdam, Netherlands, July 2016.

[192] J. Toman and D. Grossman. “Staccato: A Bug Finder for Dynamic Configuration Updates”. In *Proc. European Conf. Object-Oriented Programming (ECOOP)*. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, July 2016.

[191] T. Yu and M. Pradel. “SyncProf: Detecting, Localizing, and Optimizing Synchronization Bottlenecks”. In *Proc. Int’l Symp. Software Testing and Analysis (ISSTA)*. Saarbrücken, Germany: ACM, July 2016, pp. 389–400.

[190] C. Curtsinger and E. D. Berger. “COZ: Finding Code that Counts with Causal Profiling”. In *USENIX Annual Technical Conference (ATC)*. Denver, CO, USA: USENIX Association, June 2016.

[189] V. Garousi, M. Felderer, and M. V. Mäntylä. “The Need for Multivocal Literature Reviews in Software Engineering: Complementing Systematic Literature Reviews with Grey Literature”. In *Proc. Int’l Conf. Evaluation and Assessment in Software Engineering (EASE)*. Limerick, Ireland: ACM, June 2016, 26:1–26:6.

[188] J. Yang, T. Hance, T. H. Austin, A. Solar-Lezama, C. Flanagan, and S. Chong. "Precise, Dynamic Information Flow for Database-backed Applications". In *Proc. Conf. Programming Language Design and Implementation (PLDI)*. Santa Barbara, CA, USA: ACM, June 2016, pp. 631–647.

[187] S. A. Chowdhury and A. Hindle. "GreenOracle: Estimating Software Energy Consumption with Energy Measurement Corpora". In *Proc. Int'l Conf. Mining Software Repositories*. Austin, TX, USA: ACM, May 2016, pp. 49–60.

[186] M. Dhok, M. K. Ramanathan, and N. Sinha. "Type-aware Concolic Testing of JavaScript Programs". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Austin, TX, USA: ACM, May 2016, pp. 168–179.

[185] J. Gui, D. Li, M. Wan, and W. G. J. Halfond. "Lightweight Measurement and Estimation of Mobile Ad Energy Consumption". In *Proc. Int'l Workshop Green and Sustainable Software (GREENS)*. Austin, TX, USA: ACM, May 2016, pp. 1–7.

[184] S. Hasan, Z. King, M. Hafiz, M. Sayagh, B. Adams, and A. Hindle. "Energy Profiles of Java Collections Classes". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Austin, TX, USA: ACM, May 2016.

[183] D. Li, Y. Lyu, J. Gui, and W. G. Halfond. "Automated Energy Optimization of HTTP Requests for Mobile Applications". In Austin, TX, USA: ACM, May 2016.

[182] F. Long and M. Rinard. "An Analysis of the Search Spaces for Generate and Validate Patch Generation Systems". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Austin, TX, USA: ACM, May 2016, pp. 702–713.

[181] I. Manotas, C. Bird, R. Zhang, D. Shepherd, C. Jaspan, C. Sadowski, L. Pollock, and J. Clause. "An Empirical Study of Practitioners' Perspectives on Green Software Engineering". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Austin, TX, USA: ACM, May 2016, pp. 237–248.

[180] F. Medeiros, C. Kästner, M. Ribeiro, R. Gheyi, and S. Apel. "A Comparison of 10 Sampling Algorithms for Configurable Systems". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Austin, TX, USA: ACM, May 2016, pp. 643–654.

[179] J. Melo, C. Brabrand, and A. Wąsowski. "How Does the Degree of Variability Affect Bug Finding?" In *Proc. Int'l Conf. Software Engineering (ICSE)*. Austin, TX, USA: ACM, May 2016, pp. 679–690.

[178] R. Pereira, M. Couto, J. Saraiva, J. Cunha, and J. P. Fernandes. "The Influence of the Java Collection Framework on Overall Energy Consumption". In *Proc. Int'l Workshop Green and Sustainable Software (GREENS)*. Austin, TX, USA: ACM, May 2016, pp. 15–21.

[177] M. T. Rahman, L.-P. Querel, P. C. Rigby, and B. Adams. "Feature Toggles: Practitioner Practices and a Case Study". In *Proc. Int'l Conf. Mining Software Repositories*. Austin, TX, USA: ACM, May 2016, pp. 201–211.

[176] B. Ray, V. Hellendoorn, S. Godhane, Z. Tu, A. Bacchelli, and P. Devanbu. "On the "Naturalness" of Buggy Code". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Austin, TX, USA: ACM, May 2016, pp. 428–439.

[175] P. Thongtanunam, S. McIntosh, A. E. Hassan, and H. Iida. "Revisiting Code Ownership and Its Relationship with Software Quality in the Scope of Modern Code Review". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Austin, TX, USA: ACM, May 2016, pp. 1039–1050.

[174] K. Vaniea and Y. Rashidi. "Tales of Software Updates: The Process of Updating Software". In *Proc. Conf. Human Factors in Computing Systems (CHI)*. San Jose, CA, USA: ACM, May 2016, pp. 3215–3226.

[173] B. Vasilescu, K. Blincoe, Q. Xuan, C. Casalnuovo, D. Damian, P. Devanbu, and V. Filkov. "The Sky is Not the Limit: Multitasking Across GitHub Projects". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Austin, TX, USA: ACM, May 2016, pp. 994–1005.

[172] Y. Kwon, D. Kim, W. N. Sumner, K. Kim, B. Saltaformaggio, X. Zhang, and D. Xu. "LDX: Causality Inference by Lightweight Dual Execution". In *Proc. Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS)*. Atlanta, GA, USA: ACM, Apr. 2016, pp. 503–515.

[171] A. S. Buyukkayhan, K. Onarlioglu, W. Robertson, and E. Kirda. "CrossFire: An Analysis of Firefox Extension-Reuse Vulnerabilities". In *Network and Distributed System Security Symposium (NDSS)*. San Diego, CA, USA: Internet Society, Feb. 2016.

[170] S. Savage, A. Monroy-Hernández, and T. Hollerer. "Botivist: Calling Volunteers to Action using Online Bots". In *Conf. Computer-Supported Cooperative Work and Social Computing (CSWC)*. ACM, Feb. 2016.

[169] F. Angerer, A. Grimmer, H. Prähofer, and P. Grünbacher. "Configuration-Aware Change Impact Analysis". In *Proc. Int'l Conf. Automated Software Engineering (ASE)*. Washington, DC, USA: IEEE, 2015, pp. 385–395.

[168] P. Barros, R. Just, S. Millstein, P. Vines, W. Dietl, M. dAmorim, and M. D. Ernst. "Static Analysis of Implicit Control Flow: Resolving Java Reflection and Android Intents (T)". In *Proc. Int'l Conf. Automated Software Engineering (ASE)*. Washington, DC, USA: IEEE, 2015, pp. 669–679.

[167] M. Rapoport, O. Lhoták, and F. Tip. "Precise Data Flow Analysis in the Presence of Correlated Method Calls". In *SAS*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 54–71.

[166] A. C. Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten. "Detecting Credit Card Fraud Using Periodic Features". In *Int'l Conf. Machine Learning and Applications (ICMLA)*. Miami, FL, USA: IEEE, Dec. 2015, pp. 208–213.

[165] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison. "Hidden Technical Debt in Machine Learning Systems". In *Proc. Int'l Conf. Neural Information Processing Systems (NIPS)*. Montreal, Canada: MIT Press, Dec. 2015, pp. 2503–2511.

[164] J. Lerch, J. Späth, E. Bodden, and M. Mezini. "Access-Path Abstraction: Scaling Field-Sensitive Data-Flow Analysis with Unbounded Access Paths (T)". In *Proc. Int'l Conf. Automated Software Engineering (ASE)*. Washington, DC, USA: IEEE, Nov. 2015, pp. 619–629.

[163] F. Lv, H. Zhang, J.-g. Lou, S. Wang, D. Zhang, and J. Zhao. "CodeHow: Effective Code Search Based on API Understanding and Extended Boolean Model". In *Proc. Int'l Conf. Automated Software Engineering (ASE)*. Washington, DC, USA: IEEE, Nov. 2015, pp. 260–270.

[162] A. Sarkar, J. Guo, N. Siegmund, S. Apel, and K. Czarnecki. "Cost-Efficient Sampling for Performance Prediction of Configurable Systems". In *Proc. Int'l Conf. Automated Software Engineering (ASE)*. Washington, DC, USA: IEEE, Nov. 2015, pp. 342–352.

[161] A. Hora, R. Robbes, N. Anquetil, A. Etien, S. Ducasse, and M. T. Valente. "How do developers react to API evolution? The Pharo ecosystem case". In *Proc. Int'l Conf. Software Maintance and Evolution (ICSME)*. Bremen, Germany: IEEE, Sept. 2015, pp. 251–260.

[160] M. Sayagh and B. Adams. "Multi-layer software configuration: Empirical study on wordpress". In *Proc. Int'l Conf. Source Code Analysis and Manipulation (SCAM)*. Sept. 2015, pp. 31–40.

[159] F. Behrang, M. B. Cohen, and A. Orso. "Users Beware: Preference Inconsistencies Ahead". In *Proc. Europ. Software Engineering Conf. Foundations of Software Engineering (ESEC/FSE)*. Bergamo, Italy: ACM, Aug. 2015, pp. 295–306.

[158] M. Eichberg, B. Hermann, M. Mezini, and L. Glanz. "Hidden Truths in Dead Software Paths". In *Proc. Europ. Software Engineering Conf. Foundations of Software Engineering (ESEC/FSE)*. Bergamo, Italy: ACM, Aug. 2015, pp. 474–484.

[157] N. Siegmund, A. Grebhahn, S. Apel, and C. Kästner. "Performance-influence Models for Highly Configurable Systems". In *Proc. Europ. Software Engineering Conf. Foundations of Software Engineering (ESEC/FSE)*. Bergamo, Italy: ACM, Aug. 2015, pp. 284–294.

[156] E. K. Smith, E. T. Barr, C. Le Goues, and Y. Brun. "Is the Cure Worse than the Disease? Overfitting in Automated Program Repair". In *Proc. Europ. Software Engineering Conf. Foundations of Software Engineering (ESEC/FSE)*. Bergamo, Italy: ACM, Aug. 2015, pp. 532–543.

[155] T. Xu, L. Jin, X. Fan, Y. Zhou, S. Pasupathy, and R. Talwadker. "Hey, You Have Given Me Too Many Knobs!: Understanding and Dealing with Over-designed Configuration in System Software". In *Proc. Europ. Software Engineering Conf. Foundations of Software Engineering (ESEC/FSE)*. Bergamo, Italy: ACM, Aug. 2015, pp. 307–319.

[154] J. Bell and G. Kaiser. "Dynamic Taint Tracking for Java with Phosphor (Demo)". In *Proc. Int'l Symp. Software Testing and Analysis (ISSTA)*. Baltimore, MD, USA: ACM, July 2015, pp. 409–413.

[153] S. Zhang and M. D. Ernst. "Proactive Detection of Inadequate Diagnostic Messages for Software Configuration Errors". In *Proc. Int'l Symp. Software Testing and Analysis (ISSTA)*. Baltimore, MD, USA: ACM, July 2015, pp. 12–23.

[152] V. Avdiienko, K. Kuznetsov, A. Gorla, A. Zeller, S. Arzt, S. Rasthofer, and E. Bodden. "Mining Apps for Abnormal Usage of Sensitive Data". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Florence, Italy: IEEE, May 2015, pp. 426–436.

[151] C. Henard, M. Papadakis, M. Harman, and Y. Le Traon. "Combining Multi-objective Search and Constraint Solving for Configuring Large Software Product Lines". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Florence, Italy: IEEE, May 2015, pp. 517–528.

[150] R. Jabbarvand, A. Sadeghi, J. Garcia, S. Malek, and P. Ammann. "EcoDroid: An Approach for Energy-based Ranking of Android Apps". In *Proc. Int'l Workshop Green and Sustainable Software (GREENS)*. Florence, Italy: IEEE, May 2015, pp. 8–14.

[149] H. Malik, P. Zhao, and M. Godfrey. "Going Green: An Exploratory Analysis of Energy-Related Questions". In *Proc. Int'l Conf. Mining Software Repositories*. Florence, Italy: IEEE Press, May 2015, pp. 418–421.

[148] A. Nistor, P.-C. Chang, C. Radoi, and S. Lu. "Caramel: Detecting and Fixing Performance Problems That Have Non-intrusive Fixes". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Florence, Italy: IEEE, May 2015, pp. 902–912.

[147] J. Siegmund, N. Siegmund, and S. Apel. "Views on Internal and External Validity in Empirical Software Engineering". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Florence, Italy: IEEE, May 2015, pp. 9–19.

[146] C.-P. Bezemer, J. Pouwelse, and B. Gregg. "Understanding software performance regressions using differential flame graphs". In *Int'l Conf. Software Analysis, Evolution, and Reengineering (SANER)*. IEEE, Mar. 2015, pp. 535–539.

[145] K. Aggarwal, C. Zhang, J. C. Campbell, A. Hindle, and E. Stroulia. "The Power of System Call Traces: Predicting the Software Energy Consumption Impact of Changes". In *Proc. Int'l Conf. Computer Science and Software Engineering (CASCON)*. Markham, Ontario, Canada: IBM Corp., 2014, pp. 219–233.

[144] F. Angerer, H. Prähofer, D. Lettner, A. Grimmer, and P. Grünbacher. "Identifying Inactive Code in Product Lines with Configuration-aware System Dependence Graphs". In *Proc. Int'l Software Product Line Conference (SPLC)*. Florence, Italy: ACM, 2014, pp. 52–61.

[143] S. Elbaum, G. Rothermel, and J. Penix. "Techniques for Improving Regression Testing in Continuous Integration Development Environments". In *Proc. Int'l Symp. Foundations of Software Engineering (FSE)*. Hong Kong, China: ACM, Nov. 2014, pp. 235–245.

[142] R. Just, D. Jalali, L. Inozemtseva, M. D. Ernst, R. Holmes, and G. Fraser. "Are Mutants a Valid Substitute for Real Faults in Software Testing?" In *Proc. Int'l Symp. Foundations of Software Engineering (FSE)*. Hong Kong, China: ACM, Nov. 2014, pp. 654–665.

[141] L. Song and S. Lu. "Statistical Debugging for Real-World Performance Problems". In *Proc. Int'l Conf. Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)*. Portland, OR, USA: ACM, Oct. 2014, pp. 561–578.

[140] F. Anon, V. Navarathinarasah, M. Hoang, and C.-H. Lung. "Building a Framework for Internet of Things and Cloud Computing". In *Proc. Int'l Conf. Internet of Things (iThings)*. Taipei, Taiwan: IEEE, Sept. 2014, pp. 132–139.

[139] A. Gupta, T. Zimmermann, C. Bird, N. Nagappan, T. Bhat, and S. Emran. "Mining Energy Traces to Aid in Software Development: An Empirical Case Study". In *Proc. Int'l Symposium Empirical Software Engineering and Measurement (ESEM)*. Torino, Italy: ACM, Sept. 2014, 40:1–40:8.

[138] D. Jin, M. B. Cohen, X. Qu, and B. Robinson. "PrefFinder: Getting the Right Preference in Configurable Software Systems". In *Proc. Int'l Conf. Automated Software Engineering (ASE)*. Vasteras, Sweden: ACM, Sept. 2014, pp. 151–162.

[137] M. Lillack, C. Kästner, and E. Bodden. "Tracking Load-time Configuration Options". In *Proc. Int'l Conf. Automated Software Engineering (ASE)*. Vasteras, Sweden: ACM, Sept. 2014, pp. 445–456.

[136] R. Olaechea, D. Rayside, J. Guo, and K. Czarnecki. "Comparison of Exact and Approximate Multi-objective Optimization for Software Product Lines". In *Proc. Int'l Software Product Line Conference (SPLC)*. Florence, Italy: ACM, Sept. 2014, pp. 92–101.

[135] S. Raemaekers, A. van Deursen, and J. Visser. "Semantic Versioning Versus Breaking Changes: A Study of the Maven Repository". In *Proc. Int'l Conference Source Code Analysis and Manipulation (SCAM)*. Victoria, BC, Canada: IEEE, Sept. 2014, pp. 215–224.

[134] A. Tarvo and S. P. Reiss. "Automated Analysis of Multithreaded Programs for Performance Modeling". In *Proc. Int'l Conf. Automated Software Engineering (ASE)*. Vasteras, Sweden: ACM, Sept. 2014, pp. 7–18.

[133] M. Sun and G. Tan. "NativeGuard: Protecting Android Applications from Third-party Native Libraries". In *Proc. Conf. Security and Privacy in Wireless & Mobile Networks (WiSec)*. Oxford, United Kingdom: ACM, July 2014, pp. 165–176.

[132] S. Arzt, S. Rasthofer, C. Fritz, E. Bodden, A. Bartel, J. Klein, Y. Le Traon, D. Octeau, and P. McDaniel. "FlowDroid: Precise Context, Flow, Field, Object-sensitive and Lifecycle-aware Taint Analysis for Android Apps". In *Proc. Conf. Programming Language Design and Implementation (PLDI)*. Edinburgh, UK: ACM, June 2014, pp. 259–269.

[131] D. Garlan. "Software Architecture: A Travelogue". In *Proc. Future of Software Engineering (FOSE)*. Hyderabad, India: ACM, June 2014, pp. 29–39.

[130] D. Jin, X. Qu, M. B. Cohen, and B. Robinson. "Configurations Everywhere: Implications for Testing and Debugging in Practice". In *Companion Proc. Int'l Conf. Software Engineering*. Hyderabad, India: ACM, May 2014, pp. 215–224.

[129] M. Linares-Vásquez, G. Bavota, C. Bernal-Cárdenas, R. Oliveto, M. Di Penta, and D. Poshyvanyk. "Mining Energy-greedy API Usage Patterns in Android Apps: An Empirical Study". In *Proc. Conf. Mining Software Repositories (MSR)*. Hyderabad, India: ACM, May 2014, pp. 2–11.

[128] Y. Liu, C. Xu, and S.-C. Cheung. "Characterizing and Detecting Performance Bugs for Smartphone Applications". In *Proc. Int'l Conf. Software Engineering (ICSE)*. ICSE 2014. Hyderabad, India: ACM, May 2014, pp. 1013–1024.

[127] S. Zhang and M. D. Ernst. “Which Configuration Option Should I Change?” In *Proc. Int'l Conf. Software Engineering (ICSE)*. Hyderabad, India: ACM, May 2014, pp. 152–163.

[126] J. Zhang, L. Renganarayana, X. Zhang, N. Ge, V. Bala, T. Xu, and Y. Zhou. “EnCore: Exploiting System Environment and Correlation Information for Misconfiguration Detection”. In *Proc. Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS)*. Salt Lake City, UT, USA: ACM, Mar. 2014, pp. 687–700.

[125] B. Vasilescu, A. Serebrenik, P. Devanbu, and V. Filkov. “How Social Q&A Sites Are Changing Knowledge Sharing in Open Source Software Communities”. In *Proc. Conf. Computer Supported Cooperative Work (CSCW)*. Baltimore, MA, USA: ACM, Feb. 2014, pp. 342–354.

[124] L. Li and C. Wang. “Dynamic Analysis and Debugging of Binary Code for Security Applications”. In *RV*. 2013.

[123] C. Wilke, S. Richly, S. Götz, C. Piechnick, and U. Amann. “Energy Consumption and Efficiency in Mobile Applications: A User Feedback Study”. In *Proc. Int'l Conf. Green Computing and Communications*. IEEE, 2013, pp. 134–141.

[122] J. Guo, K. Czarnecki, S. Apel, N. Siegmund, and A. Wąsowski. “Variability-aware performance prediction: A statistical learning approach”. In *Proc. Int'l Conf. Automated Software Engineering (ASE)*. Silicon Valley, CA, USA: ACM, Nov. 2013, pp. 301–311.

[121] P. Ohmann and B. Liblit. “Lightweight control-flow instrumentation and postmortem analysis in support of debugging”. In *Proc. Int'l Conf. Automated Software Engineering (ASE)*. IEEE. Silicon Valley, CA, USA: ACM, Nov. 2013.

[120] T. Xu, J. Zhang, P. Huang, J. Zheng, T. Sheng, D. Yuan, Y. Zhou, and S. Pasupathy. “Do Not Blame Users for Misconfigurations”. In *Proc. Symp. Operating Systems Principles*. Farminton, PA, USA: ACM, Nov. 2013, pp. 244–259.

[119] B. Burg, R. Bailey, A. J. Ko, and M. D. Ernst. “Interactive Record/Replay for Web Application Debugging”. In *Proc. Symposium User Interface Software and Technology (UIST)*. St. Andrews, Scotland, United Kingdom: ACM, Oct. 2013, pp. 473–484.

[118] H. Miller, P. Haller, E. Burmako, and M. Odersky. “Instant Pickles: Generating Object-oriented Pickler Combinators for Fast and Extensible Serialization”. In *Proc. Int'l Conf. Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)*. Indianapolis, IN, USA: ACM, Oct. 2013, pp. 183–202.

[117] N. Siegmund, A. von Rhein, and S. Apel. “Family-Based Performance Measurement”. In *Proc. Int'l Conf. Generative Programming and Component Engineering (GPCE)*. Indianapolis, IN, USA: ACM, Oct. 2013, pp. 95–104.

[116] J. P. Sandoval Alcocer, A. Bergel, S. Ducasse, and M. Denker. “Performance evolution blueprint: Understanding the impact of software evolution on performance”. In *Conf. Software Visualization (VISSOFT)*. IEEE, Sept. 2013, pp. 1–9.

[115] M. Böhme, B. C. d. S. Oliveira, and A. Roychoudhury. "Regression Tests to Expose Change Interaction Errors". In *Proc. Europ. Software Engineering Conf. Foundations of Software Engineering (ESEC/FSE)*. Saint Petersburg, Russia: ACM, Aug. 2013, pp. 334–344.

[114] C. H. P. Kim, D. Marinov, S. Khurshid, D. Batory, S. Souto, P. Barros, and M. d'Amorim. "SPLat: Lightweight Dynamic Analysis for Reducing Combinatorics in Testing Configurable Systems". In *Proc. Europ. Software Engineering Conf. Foundations of Software Engineering (ESEC/FSE)*. Saint Petersburg, Russia: ACM, Aug. 2013, pp. 257–267.

[113] J. Liebig, A. von Rhein, C. Kästner, S. Apel, J. Dörre, and C. Lengauer. "Scalable Analysis of Variable Software". In *Proc. Europ. Software Engineering Conf. Foundations of Software Engineering (ESEC/FSE)*. New York, NY, USA: ACM Press, Aug. 2013, pp. 81–91.

[112] B. Wang, L. Passos, Y. Xiong, K. Czarnecki, H. Zhao, and W. Zhang. "SmartFixer: Fixing Software Configurations Based on Dynamic Priorities". In *Proc. Int'l Software Product Line Conference (SPLC)*. Tokyo, Japan: ACM, Aug. 2013, pp. 82–90.

[111] Y. Kwon, S. Lee, H. Yi, D. Kwon, S. Yang, B.-G. Chun, L. Huang, P. Maniatis, M. Naik, and Y. Paek. "Mantis: Automatic Performance Prediction for Smartphone Applications". In *Proc. Conf. Annual Technical Conference (USENIX ATC)*. San Jose, CA, USA: USENIX Association, June 2013, pp. 297–308.

[110] S. Hao, D. Li, W. G. J. Halfond, and R. Govindan. "Estimating Mobile Application Energy Consumption Using Program Analysis". In *Proc. Int'l Conf. Software Engineering (ICSE)*. San Francisco, CA, USA: IEEE, May 2013, pp. 92–101.

[109] B. Johnson, Y. Song, E. Murphy-Hill, and R. Bowdidge. "Why Don't Software Developers Use Static Analysis Tools to Find Bugs?" In *Proc. Int'l Conf. Software Engineering (ICSE)*. San Francisco, CA, USA: IEEE, May 2013, pp. 672–681.

[108] A. Nistor, T. Jiang, and L. Tan. "Discovering, Reporting, and Fixing Performance Bugs". In *Proc. Int'l Conf. Mining Software Repositories*. San Francisco, CA, USA: IEEE, May 2013, pp. 237–246.

[107] A. Nistor, L. Song, D. Marinov, and S. Lu. "Toddler: Detecting Performance Problems via Similar Memory-access Patterns". In *Proc. Int'l Conf. Software Engineering (ICSE)*. San Francisco, CA, USA: IEEE, May 2013, pp. 562–571.

[106] W. N. Sumner and X. Zhang. "Comparative Causality: Explaining the Differences Between Executions". In *Proc. Int'l Conf. Software Engineering (ICSE)*. San Francisco, CA, USA: IEEE, May 2013, pp. 272–281.

[105] S. Zhang and M. D. Ernst. "Automated Diagnosis of Software Configuration Errors". In *Proc. Int'l Conf. Software Engineering (ICSE)*. San Francisco, CA, USA: IEEE, May 2013, pp. 312–321.

[104] J. Marlow and L. Dabbish. "Activity Traces and Signals in Software Developer Recruitment and Hiring". In *Proc. Conf. Computer Supported Cooperative Work (CSCW)*. San Antonio, TX, USA: ACM, Feb. 2013, pp. 145–156.

[103] T. H. Austin, C. Flanagan, and M. Abadi. "A Functional View of Imperative Information Flow". In *Asian Symp. Programming Languages and Systems (APLAS)*. Ed. by R. Jhala and A. Igarashi. Kyoto, Japan: Springer Berlin Heidelberg, Dec. 2012, pp. 34–49.

[102] M. Attariyan, M. Chow, and J. Flinn. "X-Ray: Automating Root-Cause Diagnosis of Performance Anomalies in Production Software". In *Proc. Conf. Operating Systems Design and Implementation (OSDI)*. Hollywood, CA, USA: USENIX Association, Oct. 2012, pp. 307–320.

[101] M. F. Johansen, O. Haugen, and F. Fleurey. "An Algorithm for Generating T-wise Covering Arrays from Large Feature Models". In *Proc. Int'l Software Product Line Conference (SPLC)*. Salvador, Brazil: ACM, Sept. 2012, pp. 46–55.

[100] A. Niknafs and D. M. Berry. "The impact of domain knowledge on the effectiveness of requirements idea generation during requirements elicitation". In *Proc. Int'l Requirements Engineering Conf. (RE)*. Chicago, IL, USA: IEEE, Sept. 2012, pp. 181–190.

[99] D. Alrajeh, J. Kramer, A. v. Lamsweerde, A. Russo, and S. Uchitel. "Generating Obstacle Conditions for Requirements Completeness". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Zurich, Switzerland: IEEE, June 2012, pp. 705–715.

[98] E. Bodden. "Inter-procedural Data-flow Analysis with IFDS/IDE and Soot". In *Proc. Int'l Workshop State of the Art in Java Program Analysis (SOAP)*. Beijing, China: ACM, June 2012, pp. 3–8.

[97] E. Bodden. "Position Paper: Static Flow-sensitive, Context-sensitive Information-flow Analysis for Software Product Lines". In *Proc. Workshop Programming Languages and Analysis for Security (PLAS)*. Beijing, China: ACM, June 2012, 6:1–6:6.

[96] J. Downs, B. Plimmer, and J. G. Hosking. "Ambient Awareness of Build Status in Collocated Software Teams". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Zurich, Switzerland: IEEE, June 2012, pp. 507–517.

[95] M. Grechanik, C. Fu, and Q. Xie. "Automatically Finding Performance Problems with Feedback-Directed Learning Software Testing". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Zurich, Switzerland: IEEE, June 2012, pp. 156–166.

[94] M. Greiler, A. v. Deursen, and M.-A. Storey. "Test Confessions: A Study of Testing Practices for Plug-in Systems". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Zurich, Switzerland: IEEE, June 2012, pp. 244–254.

[93] S. Han, Y. Dang, S. Ge, D. Zhang, and T. Xie. "Performance Debugging in the Large via Mining Millions of Stack Traces". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Zurich, Switzerland: IEEE, June 2012, pp. 145–155.

[92] A. Hindle, E. T. Barr, Z. Su, M. Gabel, and P. Devanbu. "On the Naturalness of Software". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Zurich, Switzerland: IEEE, June 2012, pp. 837–847.

[91] G. Jin, L. Song, X. Shi, J. Scherpelz, and S. Lu. "Understanding and Detecting Real-world Performance Bugs". In *Proc. Conf. Programming Language Design and Implementation (PLDI)*. Beijing, China: ACM, June 2012, pp. 77–88.

[90] C. Le Goues, M. Dewey-Vogt, S. Forrest, and W. Weimer. "A Systematic Study of Automated Program Repair: Fixing 55 out of 105 Bugs for $8 Each". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Zurich, Switzerland: IEEE, June 2012, pp. 3–13.

[89] J. Park, M. Kim, B. Ray, and D. Bae. "An empirical study of supplementary bug fixes". In *Proc. Int'l Conf. Mining Software Repositories*. Zurich, Switzerland, June 2012, pp. 40–49.

[88] N. Siegmund, S. S. Kolesnikov, C. Kästner, S. Apel, D. Batory, M. Rosenmüller, and G. Saake. "Predicting Performance via Automated Feature-interaction Detection". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Zurich, Switzerland: IEEE, June 2012, pp. 167–177.

[87] C. Song, A. Porter, and J. S. Foster. "iTree: Efficiently Discovering High-coverage Configurations Using Interaction Trees". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Zurich, Switzerland: IEEE, June 2012, pp. 903–913.

[86] A. Pathak, Y. C. Hu, and M. Zhang. "Where is the Energy Spent Inside My App?: Fine Grained Energy Accounting on Smartphones with Eprof". In *Pro. Conf. Computer Systems (EuroSys*. Bern, Switzerland: ACM, Apr. 2012, pp. 29–42.

[85] D. Garlan, V. Dwivedi, I. Ruchkin, and B. Schmerl. "Foundations and Tools for End-user Architecting". In *Proc. Conf. Large-Scale Complex IT Systems: Development, Operation and Management*. Oxford, UK: Springer-Verlag, Mar. 2012, pp. 157–182.

[84] L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb. "Social Coding in GitHub: Transparency and Collaboration in an Open Software Repository". In *Proc. Conf. Computer Supported Cooperative Work (CSCW)*. Seattle, WA, USA: ACM, Feb. 2012, pp. 1277–1286.

[83] T. H. Austin and C. Flanagan. "Multiple Facets for Dynamic Information Flow". In *Proc. Symp. Principles of Programming Languages (POPL)*. Philadelphia, PA, USA: ACM, Jan. 2012, pp. 165–178.

[82] A. Hubaux, Y. Xiong, and K. Czarnecki. "A User Survey of Configuration Challenges in Linux and eCos". In *Proc. Workshop Variability Modeling of Software-Intensive Systems (VAMOS)*. Leipzig, Germany: ACM, Jan. 2012, pp. 149–155.

[81] A. Hervieu, B. Baudry, and A. Gotlieb. "PACOGEN: Automatic Generation of Pairwise Test Configurations from Feature Models". In *Proc. Int'l Symposium Software Reliability Engineering*. Nov. 2011, pp. 120–129.

[80] T. D. LaToza and B. A. Myers. "Visualizing call graphs". In *Symposium Visual Languages and Human-Centric Computing (VL/HCC)*. Nov. 2011, pp. 117–124.

[79] M. Jovic, A. Adamoli, and M. Hauswirth. "Catch Me If You Can: Performance Bug Detection in the Wild". In *Proc. Int'l Conf. Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)*. Portland, Oregon, USA: ACM, Oct. 2011, pp. 155–170.

[78] J. Sunshine, K. Naden, S. Stork, J. Aldrich, and É. Tanter. "First-class State Change in Plaid". In *Proc. Int'l Conf. Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)*. Portland, OR, USA: ACM, Oct. 2011, pp. 713–732.

[77] Z. Yin, D. Yuan, Y. Zhou, S. Pasupathy, and L. Bairavasundaram. "How Do Fixes Become Bugs?" In *Proc. Europ. Software Engineering Conf. Foundations of Software Engineering (ESEC/FSE)*. Szeged, Hungary: ACM, Sept. 2011, pp. 26–36.

[76] A. Zeller, T. Zimmermann, and C. Bird. "Failure is a Four-letter Word". In *Proc. Int'l Conf. Predictive Models in Software Engineering (PROMISE)*. Banff, Alberta, Canada: ACM, Sept. 2011, 5:1–5:7.

[75] A. Zeller, T. Zimmermann, and C. Bird. "Failure is a Four-letter Word: A Parody in Empirical Research". In *Proc. Int'l Conf. Predictive Models in Software Engineering (PROMISE)*. Banff, Alberta, Canada: ACM, Sept. 2011, 5:1–5:7.

[74] C. Parnin and A. Orso. "Are Automated Debugging Techniques Actually Helping Programmers?" In *Proc. Int'l Symp. Software Testing and Analysis (ISSTA)*. Toronto, Canada: ACM, July 2011, pp. 199–209.

[73] A. Sampson, W. Dietl, E. Fortuna, D. Gnanapragasam, L. Ceze, and D. Grossman. "EnerJ: Approximate Data Types for Safe and General Low-power Computation". In *Proc. Conf. Programming Language Design and Implementation (PLDI)*. San Jose, CA, USA: ACM, June 2011, pp. 164–174.

[72] A. Rabkin and R. Katz. "Static Extraction of Program Configuration Options". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Waikiki, Honolulu, HI, USA: ACM, May 2011, pp. 131–140.

[71] A. Bhave, B. Krogh, D. Garlan, and B. Schmerl. "View Consistency in Architectures for Cyber-Physical Systems". In *Proc. Int'l Conf. Cyber-Physical Systems (ICCPS)*. Chicago, IL, USA: IEEE Computer Society Press, Apr. 2011, pp. 151–160.

[70] A. Pathak, Y. C. Hu, M. Zhang, P. Bahl, and Y.-M. Wang. "Fine-grained Power Modeling for Smartphones Using System Call Tracing". In *Proc. Conf. Computer Systems (EuroSys)*. Salzburg, Austria: ACM, Apr. 2011, pp. 153–168.

[69] H. Hoffmann, S. Sidiroglou, M. Carbin, S. Misailovic, A. Agarwal, and M. Rinard. "Dynamic Knobs for Responsive Power-aware Computing". In *Proc. Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS)*. Newport Beach, CA, USA: ACM, Mar. 2011, pp. 199–212.

[68] C. H. P. Kim, D. S. Batory, and S. Khurshid. "Reducing Combinatorics in Testing Product Lines". In *Proc. Int'l Conf. Aspect-Oriented Software Development (AOSD)*. Porto de Galinhas, Brazil: ACM, Mar. 2011, pp. 57–68.

[67] F. Hutter, H. H. Hoos, and K. Leyton-Brown. "Sequential Model-based Optimization for General Algorithm Configuration". In *Proc. Int'l Conf. Learning and Intelligent Optimization*. Rome, Italy: Springer-Verlag, Jan. 2011, pp. 507–523.

[66] S. Oster, I. Zorcic, F. Markert, and M. Lochau. "MoSo-PoLiTe: Tool Support for Pairwise and Model-based Software Product Line Testing". In *Proc. Workshop Variability Modeling of Software-Intensive Systems (VAMOS)*. Namur, Belgium: ACM, Jan. 2011, pp. 79–82.

[65] M. Abi-Antoun and J. M. Barnes. "Analyzing Security Architectures". In *Proc. Int'l Conf. Automated Software Engineering (ASE)*. Antwerp, Belgium: ACM, 2010, pp. 3–12.

[64] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth. "TaintDroid: An Information-flow Tracking System for Realtime Privacy Monitoring on Smartphones". In *Proc. Conf. Operating Systems Design and Implementation (OSDI)*. Vancouver, BC, Canada: USENIX Association, 2010, pp. 393–407.

[63] M. Attariyan and J. Flinn. "Automating Configuration Troubleshooting with Dynamic Information Flow Analysis". In *Proc. Conf. Operating Systems Design and Implementation (OSDI)*. Vancouver, BC, Canada: USENIX Association, Oct. 2010, pp. 237–250.

[62] T. D. LaToza and B. A. Myers. "Hard-to-Answer Questions about Code". In *Evaluation and Usability of Programming Languages and Tools*. Reno, NV, USA: ACM, Oct. 2010.

[61] N. Maiden, S. Jones, K. Karlsen, R. Neill, K. Zachos, and A. Milne. "Requirements Engineering As Creative Problem Solving: A Research Agenda for Idea Finding". In *Proc. Int'l Requirements Engineering Conf. (RE)*. Sydney, Australia: IEEE, Sept. 2010, pp. 57–66.

[60] P. Sawyer, N. Bencomo, J. Whittle, E. Letier, and A. Finkelstein. "Requirements-Aware Systems: A Research Agenda for RE for Self-adaptive Systems". In *Proc. Int'l Requirements Engineering Conf. (RE)*. Sydney, Australia: IEEE, Sept. 2010, pp. 95–103.

[59] D. Tang, A. Agarwal, D. O'Brien, and M. Meyer. "Overlapping Experiment Infrastructure: More, Better, Faster Experimentation". In *Proc. Int'l Conf. Knowledge Discovery and Data Mining (KDD)*. Washington, DC, USA: ACM, July 2010, pp. 17–26.

[58] T. H. Austin and C. Flanagan. "Permissive Dynamic Information Flow Analysis". In *Proc. Workshop Programming Languages and Analysis for Security (PLAS)*. Toronto, Canada: ACM, June 2010, 3:1–3:12.

[57] E. J. Schwartz, T. Avgerinos, and D. Brumley. "All You Ever Wanted to Know About Dynamic Taint Analysis and Forward Symbolic Execution (but Might Have Been Afraid to Ask)". In *Proc. Symp. Security and Privacy (SP)*. Oakland, CA, USA: IEEE, May 2010, pp. 317–331.

[56] J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. Chi. "Short and Tweet: Experiments on Recommending Content from Information Streams". In *Proc. Conf. Human Factors in Computing Systems (CHI)*. Atlanta, GA, USA: ACM, Apr. 2010, pp. 1185–1194.

[55] S. Breu, R. Premraj, J. Sillito, and T. Zimmermann. "Information Needs in Bug Reports: Improving Cooperation between Developers and Users". In *Proc. Conf. Computer Supported Cooperative Work (CSCW)*. Savannah, GA, USA: ACM, Feb. 2010, pp. 301–310.

[54] D. F. Sutherland and W. L. Scherlis. "Composable Thread Coloring". In *Proc. Symp. Principles and Practice of Parallel Programming*. Bangalore, India: ACM, Jan. 2010, pp. 233–244.

[53] T. H. Austin and C. Flanagan. "Efficient Purely-dynamic Information Flow Analysis". In *Proc. Workshop Programming Languages and Analysis for Security (PLAS)*. Dublin, Ireland: ACM, June 2009, pp. 113–124.

[52] J. Aranda and G. Venolia. "The Secret Life of Bugs: Going Past the Errors and Omissions in Software Repositories". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Vancouver, Canada: IEEE, May 2009, pp. 298–308.

[51] U. Dekel and J. D. Herbsleb. "Improving API Documentation Usability with Knowledge Pushing". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Vancouver, Canada: IEEE Computer Society Press, May 2009, pp. 320–330.

[50] D. King, B. Hicks, M. Hicks, and T. Jaeger. "Implicit Flows: Can'T Live with 'Em, Can'T Live Without 'Em". In *Proc. Int'l Conf. Information Systems Security (ICISS)*. Hyderabad, India: Springer-Verlag, Dec. 2008, pp. 56–70.

[49] A. J. Ko and B. A. Myers. "Debugging Reinvented: Asking and Answering Why and Why Not Questions About Program Behavior". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Leipzig, Germany: ACM, Oct. 2008, pp. 301–310.

[48] C. Seo, S. Malek, and N. Medvidovic. "Component-Level Energy Consumption Estimation for Distributed Java-Based Software Systems". In *Pro. Int'l Symposium Component-Based Software Engineering (CBSE)*. Karlsruhe, Germany: Springer-Verlag, Oct. 2008, pp. 97–113.

[47] H. Müller, M. Pezzè, and M. Shaw. "Visibility of Control in Adaptive Systems". In *Proc. Int'l Workshop Ultra-Large-Scale Software-Intensive Systems (ULSSIS)*. Leipzig, Germany: ACM, May 2008, pp. 23–26.

[46] N. Nagappan, B. Murphy, and V. Basili. "The Influence of Organizational Structure on Software Quality: An Empirical Case Study". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Leipzig, Germany: ACM, May 2008, pp. 521–530.

[45] C. Seo, S. Malek, and N. Medvidovic. "Estimating the Energy Consumption in Pervasive Java-Based Systems". In *Proc. Int'l Conf. Pervasive Computing and Communications (PerCom)*. Mar. 2008, pp. 243–247.

[44] J. Clause, W. Li, and A. Orso. "Dytan: A Generic Dynamic Taint Analysis Framework". In *Proc. Int'l Symp. Software Testing and Analysis (ISSTA)*. London, United Kingdom: ACM, 2007, pp. 196–206.

[43] K. Sen. "Concolic testing". In *Proc. Int'l Conf. Automated Software Engineering (ASE)*. Nov. 2007, pp. 571–572.

[42] J. Aranda, S. Easterbrook, and G. Wilson. "Requirements in the wild: How small companies do it". In *Proc. Int'l Requirements Engineering Conf. (RE)*. New Delhi, India: IEEE, Oct. 2007, pp. 39–48.

[41] G. Pothier, É. Tanter, and J. Piquer. "Scalable Omniscient Debugging". In *Proc. Int'l Conf. Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)*. Montreal, Quebec, Canada: ACM, Oct. 2007, pp. 535–552.

[40] D. Andrzejewski, A. Mulhern, B. Liblit, and X. Zhu. "Statistical Debugging Using Latent Topic Models". In *Proc. European Conf. Machine Learning*. Warsaw, Poland: Springer-Verlag, Sept. 2007, pp. 6–17.

[39] T. D. LaToza, D. Garlan, J. D. Herbsleb, and B. A. Myers. "Program Comprehension as Fact Finding". In *Proc. Europ. Software Engineering Conf. Foundations of Software Engineering (ESEC/FSE)*. Dubrovnik, Croatia: ACM, Sept. 2007, pp. 361–370.

[38] N. Nethercote and J. Seward. "Valgrind: A Framework for Heavyweight Dynamic Binary Instrumentation". In *Proc. Conf. Programming Language Design and Implementation (PLDI)*. San Diego, CA, USA: ACM, June 2007, pp. 89–100.

[37] M. Sridharan, S. J. Fink, and R. Bodik. "Thin Slicing". In *Proc. Conf. Programming Language Design and Implementation (PLDI)*. San Diego, CA, USA: ACM, June 2007, pp. 112–122.

[36] Y. Lei, R. Kacker, D. R. Kuhn, V. Okun, and J. Lawrence. "IPOG: A General Strategy for T-Way Software Testing". In *Proc. Int'l Conf. and Workshops Engineering of Computer-Based Systems (ECBS)*. Tucson, AZ, USA: IEEE, Mar. 2007, pp. 549–556.

[35] F. Steimann. "The Paradoxical Success of Aspect-oriented Programming". In *Proc. Int'l Conf. Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)*. Portland, OR, USA: ACM, Oct. 2006, pp. 481–497.

[34] F. Mancinelli, J. Boender, R. di Cosmo, J. Vouillon, B. Durak, X. Leroy, and R. Treinen. "Managing the Complexity of Large Free and Open Source Package-Based Software Distributions". In *Proc. Int'l Conf. Automated Software Engineering (ASE)*. Washington, DC, USA: IEEE, Sept. 2006, pp. 199–208.

[33] G. Serazzri, G. Casale, M. Bertoli, G. Serazzri, G. Casale, and M. Bertoli. "Java Modelling Tools: An Open Source Suite for Queueing Network Modelling andWorkload Analysis". In *Proc. Int'l Conf. the Quantitative Evaluation of Systems - (QEST'06)*. Sept. 2006, pp. 119–120.

[32] D. Garlan and B. Schmerl. "Architecture-driven Modelling and Analysis". In *Proc. Safety Critical Systems and Software (SCS)*. Melbourne, Australia: Australian Computer Society, Inc., Aug. 2006, pp. 3–17.

[31] D. Batory. "Feature Models, Grammars, and Propositional Formulas". In *Proc. Int'l Software Product Line Conference (SPLC)*. Rennes, France: Springer-Verlag, Sept. 2005, pp. 7–20.

[30] P. Godefroid, N. Klarlund, and K. Sen. "DART: Directed Automated Random Testing". In *Proc. Conf. Programming Language Design and Implementation (PLDI)*. Chicago, IL, USA: ACM, June 2005, pp. 213–223.

[29] A. J. Ko and B. A. Myers. "Designing the Whyline: A Debugging Interface for Asking Questions about Program Behavior". In *Proc. Conf Human Factors in Computing Systems (CHI)*. Vienna, Austria: ACM, Apr. 2004.

[28] J. Krinke. "Barrier Slicing and Chopping". In *Int'l Workshop Source Code Analysis and Manipulation (SCAM)*. Amsterdam, Netherlands: IEEE, Sept. 2003.

[27] D. Batory, J. N. Sarvela, and A. Rauschmayer. "Scaling Step-wise Refinement". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Portland, OR, USA: IEEE, May 2003, pp. 187–197.

[26] M. Shaw. "Writing Good Software Engineering Research Papers: Minitutorial". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Portland, Oregon: IEEE, May 2003, pp. 726–736.

[25] R. E. Grinter and L. Palen. "Instant Messaging in Teen Life". In *Proc. Conf. Computer Supported Cooperative Work (CSCW)*. New Orleans, LA, USA: ACM, Nov. 2002, pp. 21–30.

[24] J. Krinke. "Evaluating Context-Sensitive Slicing and Chopping". In *Int'l Conf. Software Maintenance (ICSM)*. Montréal, Canada: IEEE, Oct. 2002, pp. 22–31.

[23] J. I. Maletic, A. Marcus, and M. L. Collard. "A task oriented view of software visualization". In *Proc. Int'l Workshop Visualizing Software for Understanding and Analysis*. June 2002, pp. 32–40.

[22] T. A. Henzinger, R. Jhala, R. Majumdar, and G. Sutre. "Lazy Abstraction". In *Proc. Symp. Principles of Programming Languages (POPL)*. Portland, OR, USA: ACM, Jan. 2002, pp. 58–70.

[21] D. Fox. "KLD-Sampling: Adaptive Particle Filters". In *Advances in Neural Information Processing Systems 14*. MIT Press, 2001.

[20] A. Van Lamsweerde. "Goal-Oriented Requirements Engineering: A Guided Tour". In *Proc. Int'l Requirements Engineering Conf. (RE)*. Toronto, Canada: IEEE, Aug. 2001, pp. 249–262.

[19] T. Ball and S. K. Rajamani. "Automatically Validating Temporal Safety Properties of Interfaces". In *Proc. Int'l Workshop Model Checking of Software (SPIN)*. Toronto, Canada: Springer-Verlag New York, Inc., May 2001, pp. 103–122.

[18] B. R. Murphy and M. S. Lam. "Program Analysis with Partial Transfer Functions". In *Proc. Workshop Partial Evaluation and Semantics-based Program Manipulation*. Boston, MA, USA: ACM, Jan. 2000, pp. 94–103.

[17] R. Vallée-Rai, P. Co, E. Gagnon, L. Hendren, P. Lam, and V. Sundaresan. "Soot - a Java Bytecode Optimization Framework". In *Proc. Conf. Centre for Advanced Studies on Collaborative Research (CASCON)*. Mississauga, Ontario, Canada: IBM Press, Nov. 1999, pp. 13–.

[16] M. D. Ernst, J. Cockrell, W. G. Griswold, and D. Notkin. "Dynamically Discovering Likely Program Invariants to Support Program Evolution". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Los Angeles, CA, USA: ACM, May 1999, pp. 213–224.

[15] P. Tarr, H. Ossher, W. Harrison, and S. M. Sutton Jr. "N Degrees of Separation: Multi-dimensional Separation of Concerns". In *Proc. Int'l Conf. Software Engineering (ICSE)*. Los Angeles, CA, USA: ACM, May 1999, pp. 107–119.

[14] L. Cardelli. "Program Fragments, Linking, and Modularization". In *Proc. Symp. Principles of Programming Languages (POPL)*. Paris, France: ACM, Jan. 1997, pp. 266–277.

[13] C. Potts and W. C. Newstetter. "Naturalistic Inquiry and Requirements Engineering: Reconciling Their Theoretical Foundations". In *Proc. Int'l Requirements Engineering Conf. (RE)*. Annapolis, MD, USA: IEEE, Jan. 1997, pp. 118–127.

[12] M. Jackson. “The World and the Machine”. In *Proc. Int’l Conf. Software Engineering (ICSE)*. Seattle, WA, USA: ACM, Apr. 1995, pp. 283–292.

[11] O. Gotel and A. Finkelstein. “Contribution Structures”. In *Proc. Int’l Requirements Engineering Conf. (RE)*. York, U.K.: IEEE, Mar. 1995, pp. 100–107.

[10] T. Reps, S. Horwitz, and M. Sagiv. “Precise Interprocedural Dataflow Analysis via Graph Reachability”. In *Proc. Symp. Principles of Programming Languages (POPL)*. San Francisco, CA, USA: ACM, Jan. 1995, pp. 49–61.

[9] T. Biggerstaff. “The Library Scaling Problem and the Limits of Concrete Component Reuse”. In *Proc. Int’l Conf. Software Reuse (ICSR)*. Rio de Janeiro, Brazil: IEEE, 1994, pp. 102–109.

[8] R. Allen and D. Garlan. “Formalizing Architectural Connection”. In *Proc. Int’l Conf. Software Engineering (ICSE)*. Sorrento, Italy: IEEE Computer Society Press, May 1994, pp. 71–80.

[7] L. Osterweil. “Software Processes Are Software Too”. In *Proc. Int’l Conf. Software Engineering (ICSE)*. Monterey, CA, USA: IEEE Computer Society Press, Mar. 1987, pp. 2–13.

[6] W. W. Royce. “Managing the Development of Large Software Systems: Concepts and Techniques”. In *Proc. Int’l Conf. Software Engineering (ICSE)*. Monterey, CA, USA: IEEE Computer Society Press, Mar. 1987, pp. 328–338.

[5] S. T. Redwine Jr. and W. E. Riddle. “Software Technology Maturation”. In *Proc. Int’l Conf. Software Engineering (ICSE)*. London, England: IEEE Computer Society Press, Aug. 1985, pp. 189–200.

[4] D. L. Parnas, P. C. Clements, and D. M. Weiss. “The Modular Structure of Complex Systems”. In *Proc. Int’l Conf. Software Engineering (ICSE)*. Orlando, FL, USA: IEEE, Mar. 1984, pp. 408–417.

[3] M. Weiser. “Program Slicing”. In *Proc. Int’l Conf. Software Engineering (ICSE)*. San Diego, CA, USA: IEEE, Mar. 1981, pp. 439–449.

[2] D. L. Parnas. “Designing Software for Ease of Extension and Contraction”. In *Proc. Int’l Conf. Software Engineering (ICSE)*. Atlanta, GA, USA: IEEE, May 1978, pp. 264–277.

[1] G. A. Kildall. “A Unified Approach to Global Program Optimization”. In *Proc. Symp. Principles of Programming Languages (POPL)*. Boston, MA, USA: ACM, Oct. 1973, pp. 194–206.

Technical Reports

[8] M. Velez, P. Jamshidi, F. Sattler, N. Siegmund, S. Apel, and C. Kästner. *ConfigCrusher: Towards White-Box Performance Analysis for Configurable Systems*. Tech. rep. 1905.02066v2. arXiv, July 2020.

[7] D. Alvarez-Melis and T. S. Jaakkola. *On the robustness of interpretability methods*. Tech. rep. arXiv:1806.08049. arXiv, 2018.

[6] C. Fritz, S. Arzt, S. Rasthofer, E. Bodden, A. Bartel, J. Klein, Y. le Traon, D. Octeau, and P. McDaniel. *Highly Precise Taint Analysis for Android Applications*. Tech. rep. TUD-CS-2013-0113. EC SPRIDE, May 2013.

[5] T. H. Austin, T. Disney, A. Jeffrey, and C. Flanagan. *Dynamic Information Flow Analysis for Featherweight JavaScript*. Tech. rep. University of California Santa Cruz, 2011.

[4] J. Newsome and D. Song. *Dynamic Taint Analysis for Automatic Detection, Analysis, and Signature Generation of Exploits on Commodity Software*. CMU-CS-04-140. Carnegie Mellon University, 2005.

[3] R. E. Filman and D. P. Friedman. *Aspect-Oriented Programming is Quantification and Obliviousness*. Technical Report. NASA, 2000.

[2] R. Kazman, M. Klein, and P. Clements. *ATAM: Method for Architecture Evaluation*. Technical Report CMU/SEI-2000-TR-004. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000.

[1] D. Garlan and M. Shaw. *An Introduction to Software Architecture*. Technical Report CMU-CS-94-166. Carnegie Mellon University, Jan. 1994.

Part of Books

[2] M. Shaw. “The Role of Design Spaces”. In M. Petre and A. van der Hoek. *Software Designers in Action: A Human-Centric Look at Design Work*. CRC Press, 2013. Chap. 3.

[1] E. Kern, M. Dick, T. Johann, and S. Naumann. “Green Software and Green IT: An End Users Perspective”. In *Information Technologies in Environmental Engineering: New Trends and Challenges*. Ed. by P. Golinska, M. Fertsch, and J. Marx-Gómez. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 199–211.

Thesis

[4] X. Han. “CONFPROFITT: A CONFIGURATIONAWAREPERFORMANCE PROFILING,TESTING, AND TUNING FRAMEWORK”. PhD thesis. University of Kentucky, 2019.

[3] C. Kapfhammer. “Adjustable Family-based Performance Measurement”. MA thesis. University of Passau, 2017.

[2] N. Siegmund. “Measuring and Predicting Non-Functional Properties of Customizable Programs”. PhD thesis. University of Magdeburg, 2012.

[1] C. Kästner. “Virtual Separation of Concerns: Toward Preprocessors 2.0”. Logos Verlag Berlin. PhD thesis. Magdeburg, Germany: University of Magdeburg, May 2010.

Miscellaneous

[19] TurtleBot. *What is TurtleBot?*

[18] M. Velez, P. Jamshidi, N. Siegmund, S. Apel, and C. Kästner. *White-Box Analysis over Machine Learning: Modeling Performance of Configurable Systems - Supplementary Material - https://bit.ly/3bbbgG8*. 2021.

[17] R. Krishna, M. S. Iqbal, M. A. Javidian, B. Ray, and P. Jamshidi. *CADET: A Systematic Method For Debugging Misconfigurations using Counterfactual Reasoning*. 2020. arXiv: 2010.06061 [cs.SE].

[16] A. X. Zhang, M. Muller, and D. Wang. *How do Data Science Workers Collaborate? Roles, Workflows, and Tools*. 2020. arXiv: 2001.06684.

[15] A. Grebhahn, N. Siegmund, and S. Apel. *Predicting Performance of Software Configurations: There is no Silver Bullet*. 2019. arXiv: 1911.12643 [cs.SE].

[14] S. Saha, R. K. Saha, and M. R. Prasad. *Harnessing Evolution for Multi-Hunk Program Repair*. 2019. arXiv: 1906.08903 [cs.SE].

[13] M. Velez, P. Jamshidi, F. Sattler, N. Siegmund, S. Apel, and C. Kästner. *ConfigCrusher: Towards White-Box Performance Analysis for Configurable Systems - Supplementary Material - https://bit.ly/3diKZmK*. 2019.

[12] A. Vogelsang and M. Borg. *Requirements Engineering for Machine Learning: Perspectives from Data Scientists*. 2019. arXiv: 1908.04674.

[11] M. Velez, P. Jamshidi, C. Kästner, N. Siegmund, F. Sattler, and S. Apel. *White-Box Performance Discovery*. Poster. Google PhD Intern Research Conference. Sunnyvale, CA, USA, July 2019.

[10] M. Anders and M. I. Schwartzbach. *Static Program Analysis*. 2017.

[9] M. Velez, P. Jamshidi, C. Kästner, N. Siegmund, F. Sattler, and S. Apel. *White-Box Performance Discovery*. Poster. BRASS PI Meeting. Seattle, WA, USA, Nov. 2017.

[8] B. Nushi, E. Kamar, E. Horvitz, and D. Kossmann. *On Human Intellect and Machine Failures: Troubleshooting Integrative Machine Learning Systems*. 2016. arXiv: 1611.08309 [cs.LG].

[7] ROS.org. *amcl*. Ed. by FlorianSteinhardt. Aug. 18, 2016.

[6] M. Velez and J. Sawin. *Improving the Efficiency of CHA through Parallelization*. Poster. Inquiry at St. Thomas. St. Paul, MN, USA, May 2016.

[5] M. Velez and J. Sawin. *Faster WAH Compression Querying through the Use of Metadata*. Poster. Consortium for Computing Sciences in Colleges Midwest Region. 1^{st} place Discovery Track. Evansville, IN, USA, Oct. 2015.

[4] M. Velez and A. Solar-Lezama. *Simpler Implementation of Sketches through Enhanced Expressiveness*. Poster. MIT Summer Research Poster Session. Cambridge, MA, USA, Aug. 2015.

[3] M. Velez. *Current and Future Relationships Between Robots and Humans*. Summa Cum Laude Paper. Apr. 2015.

[2] M. Velez, P. Gittins, and J. Sawin. *Extending SMILES to Encode Reaction Mechanisms*. Poster. Inquiry at St. Thomas. St. Paul, MN, USA, May 2014.

[1] M. I. Schwartzbach. *Lecture Notes on Static Analysis*. 2008.

Manual

[2] M. Velez. *White-box Analysis for Modeling and Debugging the Performance of Configurable Systems*. Dec. 2020.

[1] E. Bruneton. *ASM 4.0 A Java bytecode engineering library*. 2nd. ASM. Sept. 2011.