References Cited

[1] J. F. Allen, "Mixed-initiative interaction," *IEEE Intelligent Systems and their Applications*, vol. 14, no. 5, pp. 14–16, 1999.

[2] G. Ammons, R. Bodík, and J. R. Larus, "Mining specifications," *Proceedings of SIGPLAN Symposium on Principles of Programming Languages*, pp. 4–16, 2002.

[3] T. Arts, "On shrinking randomly generated load tests," in *Proceedings of the Thirteenth ACM SIGPLAN workshop on Erlang*, ser. Erlang '14. New York, NY, USA: Association for Computing Machinery, Sep. 2014, pp. 25–31. [Online]. Available: http://doi.org/10.1145/2633448.2633452

[4] T. Arts, J. Hughes, J. Johansson, and U. Wiger, "Testing telecoms software with quviq quickcheck," in *Proceedings of the 2006 ACM SIGPLAN Workshop on Erlang*, 2006, pp. 2–10.

[5] T. Arts, J. Hughes, U. Norell, and H. Svensson, "Testing autosar software with quickcheck," in *2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. IEEE, 2015, pp. 1–4.

[6] E. T. Barr, M. Harman, P. McMinn, M. Shahbaz, and S. Yoo, "The Oracle Problem in Software Testing: A Survey," *IEEE Transactions on Software Engineering*, vol. 41, no. 5, pp. 507–525, May 2015, conference Name: IEEE Transactions on Software Engineering.

[7] B. Bizot, "Expanding the pipeline: Gender and ethnic differences in phd specialty areas," 2019, computing Research Association. https://cra.org/cra/expanding-the-pipeline-gender-and-ethnic-differences-in-phd-specialty-areas/.

[8] A. F. Blackwell, "First steps in programming: A rationale for attention investment models," in *Proceedings of the Symposia on Human Centric Computing Languages and Environments*. IEEE, 2002, pp. 2–10.

[9] A. Blandford, D. Furniss, and S. Makri, *Qualitative HCI Research: Going Behind the Scenes*, ser. Synthesis Lectures on Human-Centered Informatics. Morgan & Claypool Publishers, 2016.

[10] H. Borges and M. Tulio Valente, "What's in a GitHub Star? Understanding Repository Starring Practices in a Social Coding Platform," *Journal of Systems and Software*, vol. 146, pp. 112–129, Dec. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121218301961

[11] J. Bornholt, R. Joshi, V. Astrauskas, B. Cully, B. Kragl, S. Markle, K. Sauri, D. Schleit, G. Slatton, S. Tasiran, J. V. Geffen, and A. Warfield, "Using lightweight formal methods to validate a key-value storage node in Amazon S3," in *SOSP 2021*, 2021.

[12] J. I. Brachthäuser, A. S. Boruch-Gruszecki, and M. Odersky, Eds., *Representing Monads with Capabilities*, 2021, hOPE 2021 Workshop.

[13] J. Brandt, V. Pattamatta, W. Choi, B. Hsieh, and S. R. Klemmer, "Rehearse: Helping programmers adapt examples by visualizing execution and highlighting related code," Stanford University, Technical report, 2010.

[14] R. M. Braquehais, "Tools for discovery, refinement and generalization of functional properties by enumerative testing," October 2017. [Online]. Available: http://etheses.whiterose.ac.uk/19178/

[15] R. Brown, "Consideration of the origin of Herbert Simon's theory of "satisficing" (1933–1947)," *Management Decision*, vol. 42, no. 10, pp. 1240–1256, 2004.

[16] B. Burg, R. Bailey, A. J. Ko, and M. D. Ernst, "Interactive record/replay for web application debugging," in *Proceedings of the Symposium on User Interface Software and Technology*. ACM, 2013, pp. 473–483.

[17] S. E. Chasins, E. L. Glassman, and J. Sunshine, "PL and HCI: better together," *Communications of the ACM*, vol. 64, no. 8, pp. 98–106, Aug. 2021. [Online]. Available: https://dl.acm.org/doi/10.1145/3469279

[18] K. Claessen and J. Hughes, "QuickCheck: a lightweight tool for random testing of Haskell programs," in *5th ACM SIGPLAN International Conference on Functional Programming (ICFP)*, ser. ICFP. ACM, 2000, pp. 268–279. [Online]. Available: http://www.eecs.northwestern.edu/~robby/courses/395-495-2009-fall/quick.pdf

[19] K. Claessen, N. Smallbone, and J. Hughes, "Quickspec: Guessing formal specifications using testing," in *International Conference on Tests and Proofs*. Springer, 2010.

[20] "Clojure test.check," https://github.com/clojure/test.check.

[21] M. Coblenz, G. Kambhatla, P. Koronkevich, J. L. Wise, C. Barnaby, J. Sunshine, J. Aldrich, and B. A. Myers, "PLIERS: A Process that Integrates User-Centered Methods into Programming Language Design," *ACM Transactions on Computer-Human Interaction*, vol. 28, no. 4, pp. 28:1–28:53, Jul. 2021. [Online]. Available: https://doi.org/10.1145/3452379

[22] L. D'Antoni, D. Kini, R. Alur, S. Gulwani, M. Viswanathan, and B. Hartmann, "How can automatic feedback help students construct automata?" *ACM Transactions on Computer-Human Interaction*, vol. 22, no. 2, pp. 9:1–9:24, 2015.

[23] L. de Moura and N. Bjørner, "Z3: An Efficient SMT Solver," in *Tools and Algorithms for the Construction and Analysis of Systems*, ser. Lecture Notes in Computer Science, C. R. Ramakrishnan and J. Rehof, Eds. Berlin, Heidelberg: Springer, 2008, pp. 337–340.

[24] J. DeLaat, "PyMonad: Data structures and utilities for monadic style functional programming." [Online]. Available: https://github.com/jasondelaat/pymonad

[25] K. T. Dewey, "Automated black box generation of structured inputs for use in software testing," Ph.D. dissertation, 2017.

[26] Z. H. Dodds, current maintainer of Hypothesis (https://github.com/HypothesisWorks/hypothesis). Personal communication, 2022.

[27] "Shrinking · Issue #62 · stedolan/crowbar." [Online]. Available: https://github.com/stedolan/crowbar/issues/62

[28] S. Dolan and M. Preston, "Testing with crowbar," in *OCaml Workshop*, 2017.

[29] J. Ellson, E. Gansner, L. Koutsofios, S. C. North, and G. Woodhull, "Graphviz— Open Source Graph Drawing Tools," in *Graph Drawing*, ser. Lecture Notes in Computer Science, P. Mutzel, M. Jünger, and S. Leipert, Eds. Berlin, Heidelberg: Springer, 2002, pp. 483–484.

[30] Y. Feng, R. Martins, O. Bastani, and I. Dillig, "Program synthesis using conflict-driven learning," *ACM SIGPLAN Notices*, vol. 53, no. 4, pp. 420–435, Jun. 2018. [Online]. Available: https://doi.org/10.1145/3296979.3192382

[31] J. N. Foster, "Bidirectional programming languages," Ph.D. dissertation, University of Pennsylvania, 2009.

[32] A. Gerdes, J. Hughes, N. Smallbone, S. Hanenberg, S. Ivarsson, and M. Wang, "Understanding formal specifications through good examples," in *Proceedings of the International Workshop on Erlang*. ACM, 2018, pp. 13–24.

[33] P. Godefroid, A. Kiezun, and M. Y. Levin, "Grammar-based whitebox fuzzing," in *Proceedings of the 29th ACM SIGPLAN conference on programming language design and implementation*, 2008, pp. 206–215.

[34] P. Godefroid, H. Peleg, and R. Singh, “Learn&fuzz: Machine learning for input fuzzing,” in *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2017, pp. 50–59.

[35] H. Goldstein, J. W. Cutler, A. Stein, B. C. Pierce, and A. Head, “Some Problems with Properties,” vol. 1, Dec. 2022. [Online]. Available: https://harrisongoldste.in/papers/hatra2022.pdf

[36] H. Goldstein, J. Hughes, L. Lampropoulos, and B. C. Pierce, “Do judge a test by its cover: Combining combinatorial and property-based testing,” in *Programming Languages and Systems, 30th European Symposium on Programming, ESOP 2021, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2021, Luxembourg City, Luxembourg, March 27 - April 1, 2021, Proceedings*, ser. Lecture Notes in Computer Science, N. Yoshida, Ed., vol. 12648. Springer, 2021, pp. 264–291. [Online]. Available: https://doi.org/10.1007/978-3-030-72019-3_10

[37] H. Goldstein and B. C. Pierce, “Parsing Randomness,” *Proceedings of the ACM on Programming Languages*, vol. 6, no. OOPSLA2, pp. 128:89–128:113, Oct. 2022. [Online]. Available: https://doi.org/10.1145/3563291

[38] B. Greenman, S. Saarinen, T. Nelson, and S. Krishnamurthi, “Little Tricky Logic: Misconceptions in the Understanding of LTL,” *The Art, Science, and Engineering of Programming*, vol. 7, no. 2, p. 7, Oct. 2022. [Online]. Available: https://programming-journal.org/2023/7/7

[39] A. Groce, C. Zhang, E. Eide, Y. Chen, and J. Regehr, “Swarm testing,” in *Proceedings of the 2012 International Symposium on Software Testing and Analysis*, 2012, pp. 78–88.

[40] R. Hamlet, “Random testing,” *Encyclopedia of software Engineering*, vol. 2, pp. 971–978, 1994.

[41] Z. Hatfield-Dodds, “HypoFuzz.” [Online]. Available: https://hypofuzz.com/

[42] A. Hazimeh, A. Herrera, and M. Payer, “Magma: A Ground-Truth Fuzzing Benchmark,” *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 4, no. 3, pp. 49:1–49:29, Jun. 2021. [Online]. Available: http://doi.org/10.1145/3428334

[43] A. Head, C. Appachu, M. A. Hearst, and B. Hartmann, “Tutorons: Generating context-relevant, on-demand explanations and demonstrations of online code,” in *Proceedings of the Symposium on Visual Languages and Human-Centric Computing*. IEEE, 2015, pp. 3–12.

[44] A. Head, E. Glassman, G. Soares, R. Suzuki, L. Figueredo, L. D’Antoni, and B. Hartmann, “Writing reusable code feedback at scale with mixed-initiative program synthesis,” in *Proceedings of the Conference on Learning at Scale*. ACM, 2017, pp. 89–98.

[45] A. Head, E. L. Glassman, B. Hartmann, and M. A. Hearst, “Interactive extraction of examples from existing code,” in *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 2018, Paper 85.

[46] A. Head, F. Hohman, T. Barik, S. M. Drucker, and R. DeLine, “Managing messes in computational notebooks,” in *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 2019, Paper 270.

[47] A. Head, J. Jiang, J. Smith, M. A. Hearst, and B. Hartmann, “Composing flexibly-organized step-by-step tutorials from linked source code, snippets, and outputs,” in *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 2020, To appear.

[48] A. Head, C. Sadowski, E. Murphy-Hill, and A. Knight, “When not to comment: Questions and tradeoffs with API documentation for C++ projects,” in *Proceedings of the International Conference on Software Engineering*. ACM, 2018, pp. 643–653.

[49] B. Hempel, J. Lubin, and R. Chugh, “Sketch-n-sketch: Output-directed programming for SVG,” in *Proceedings of the Symposium on User Interface Software and Technology*. ACM, 2019, pp. 281–292.

[50] A. Z. Henley and S. D. Fleming, "The Patchworks code editor: Toward faster navigation with less code arranging and fewer navigation mistakes," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 2014, pp. 2511–2520.

[51] J. Hibschman and H. Zhang, "Telescope: Fine-tuned discovery of interactive web ui feature implementation," in *Proceedings of the Symposium on User Interface Software and Technology*. ACM, 2016, pp. 233–245.

[52] F. Hohman, A. Head, R. Caruana, R. DeLine, and S. M. Drucker, "Gamut: A design probe to understand how data scientists understand machine learning models," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 2019, Paper 579.

[53] F. Hohman, K. Wongsuphasawat, M. B. Kery, and K. Patel, "Understanding and visualizing data iteration in machine learning," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 2020, Paper 50.

[54] C. Holler, K. Herzig, and A. Zeller, "Fuzzing with code fragments," in *21st USENIX Security Symposium (USENIX Security 12)*, 2012, pp. 445–458.

[55] R. Holmes and R. J. Walker, "Systematizing pragmatic software reuse," vol. 21, no. 4, 2012.

[56] K. Holtzblatt and H. Beyer, *Contextual Design: Defining Customer-Centered Systems*. Elsevier, 1997.

[57] K.-B. Hou (Favonia) and Z. Wang, "Logarithm and program testing," *Proceedings of the ACM on Programming Languages*, vol. 6, no. POPL, pp. 64:1–64:26, Jan. 2022. [Online]. Available: http://doi.org/10.1145/3498726

[58] J. Hughes, "QuickCheck Testing for Fun and Profit," in *Practical Aspects of Declarative Languages*, ser. Lecture Notes in Computer Science, M. Hanus, Ed. Berlin, Heidelberg: Springer, 2007, pp. 1–32.

[59] ——, "Experiences with QuickCheck: Testing the Hard Stuff and Staying Sane," in *A List of Successes That Can Change the World: Essays Dedicated to Philip Wadler on the Occasion of His 60th Birthday*, ser. Lecture Notes in Computer Science, S. Lindley, C. McBride, P. Trinder, and D. Sannella, Eds. Cham: Springer International Publishing, 2016, pp. 169–186. [Online]. Available: https://doi.org/10.1007/978-3-319-30936-1_9

[60] J. Hughes, B. C. Pierce, T. Arts, and U. Norell, "Mysteries of Dropbox: Property-based testing of a distributed synchronization service," in *International Conference on Software Testing, Verification and Validation (ICST)*, Apr. 2016.

[61] "Hypothesis," https://github.com/HypothesisWorks/hypothesis.

[62] Jetbrains. [Online]. Available: https://www.jetbrains.com/

[63] R. Jung, R. Krebbers, J.-H. Jourdan, A. Bizjak, L. Birkedal, and D. Dreyer, "Iris from the ground up: A modular foundation for higher-order concurrent separation logic," *Journal of Functional Programming*, vol. 28, p. e20, 2018, publisher: Cambridge University Press. [Online]. Available: https://www.cambridge.org/core/journals/journal-of-functional-programming/article/iris-from-the-ground-up-a-modular-foundation-for-higherorder-concurrent-separation-logic/26301B518CE2C52796BFA12B8BAB5B5F

[64] H. Kang and P. J. Guo, "Omnicode: A novice-oriented live programming environment with always-on run-time value visualizations," in *Proceedings of the Symposium on User Interface Software and Technology*. ACM, 2017, pp. 737–745.

[65] M. B. Kery, D. Ren, F. Hohman, D. Moritz, K. Wongsuphasawat, and K. Patel, "mage: Fluid moves between code and graphical work in computational notebooks," in *Proceedings of the Symposium on User Interface Software and Technology*. ACM, 2020, pp. 140–151.

[66] A. J. Ko, T. D. LaToza, and M. M. Burnett, "A practical guide to controlled experiments of software engineering tools with human participants," *Empirical Software Engineering*, vol. 20, pp. 110–141, 2013.

[67] A. J. Ko and B. A. Myers, "A framework and methodology for studying the causes of software errors in programming systems," *Journal of Visual Languages and Computing*, vol. 16, pp. 41–84, 2005.

[68] A. J. Ko, B. A. Myers, and H. H. Aung, "Six learning barriers in end-user programming systems," in *Proceedings of the Symposium on Visual Languages and Human-Centric Computing*. IEEE, 2004, pp. 199–206.

[69] H. Krasner, "The cost of poor software quality in the US: A 2022 report," Dec. 2022, consortium for Information and Software Quality (CISQ). www.it-cisq.org.

[70] K. Krippendorff, *Content Analysis: An Introduction to Its Methodology*, 3rd ed. SAGE Publications, Inc., 2013.

[71] L. Lampropoulos, D. Gallois-Wong, C. Hriţcu, J. Hughes, B. C. Pierce, and L. Xia, "Beginner's Luck: A language for random generators," in *Foundations of Programming and Software systems: Probabilistic Programming*, G. Barthe, J.-P. Katoen, and A. Silva, Eds., 2020, an earlier version appeared in ACM SIGPLAN Symposium on Principles of Programming Languages (POPL), Jan 2017. [Online]. Available: https://arxiv.org/abs/1607.05443

[72] L. Lampropoulos, M. Hicks, and B. C. Pierce, "Coverage guided, property based testing," *Proc. ACM Program. Lang.*, vol. 3, no. OOPSLA, pp. 181:1–181:29, 2019. [Online]. Available: https://doi.org/10.1145/3360607

[73] ——, "Coverage guided, property based testing," *PACMPL*, vol. 3, no. OOPSLA, pp. 181:1–181:29, 2019. [Online]. Available: https://doi.org/10.1145/3360607

[74] ——, "Coverage guided, property based testing," Apr. 2019, to appear in PACMPL / OOPSLA 2019.

[75] L. Lampropoulos, Z. Paraskevopoulou, and B. C. Pierce, "Generating good generators for inductive relations," *PACMPL*, vol. 2, no. POPL, pp. 45:1–45:30, 2018. [Online]. Available: http://doi.acm.org/10.1145/3158133

[76] J. Lawrance, C. Bogart, M. Burnett, R. Bellamy, K. Rector, and S. D. Fleming, "How programmers debug, revisited: An information foraging theory perspective," *IEEE Transactions on Software Engineering*, vol. 39, no. 2, pp. 197–215, 2010.

[77] T.-D. B. Le and D. Lo, "Deep specification mining," in *Proceedings of the International Symposium on Software Testing and Analysis*. ACM, 2018, pp. 106–117.

[78] D. J.-L. Lee, D. Tang, K. Agarwal, T. Boonmark, C. Chen, J. Kang, U. Mukhopadhyay, J. Song, M. Yong, M. A. Hearst, and A. G. Parameswaran, "Lux: Always-on visualization recommendations for exploratory dataframe workflows," in *Proceedings of the International Conference on Very Large Data Bases*, 2022, pp. 727–738.

[79] K. R. M. Leino, "Dafny: An Automatic Program Verifier for Functional Correctness," in *Logic for Programming, Artificial Intelligence, and Reasoning*, ser. Lecture Notes in Computer Science, E. M. Clarke and A. Voronkov, Eds. Berlin, Heidelberg: Springer, 2010, pp. 348–370.

[80] Y. Li, B. C. Pierce, and S. Zdancewic, "Model-based testing of networked applications," in *ISSTA '21: 30th ACM SIGSOFT International Symposium on Software Testing and Analysis, Virtual Event, Denmark, July 11-17, 2021*, C. Cadar and X. Zhang, Eds. ACM, 2021, pp. 529–539. [Online]. Available: https://doi.org/10.1145/3460319.3464798

[81] Light table. [Online]. Available: http://lighttable.com/

[82] S. Lim, J. Hibschman, H. Zhang, and E. O'Rourke, "Ply: A visual web inspector for learning from professional webpages," in *Proceedings of the Symposium on User Interface Software and Technology*. ACM, 2018, pp. 991–1002.

[83] A. Löscher and K. Sagonas, "Targeted property-based testing," in *Proceedings of the 26th ACM SIGSOFT International Symposium on Software Testing and Analysis*, ser. ISSTA 2017. New York, NY, USA: Association for Computing Machinery, 2017, p. 46–56. [Online]. Available: https://doi.org/10.1145/3092703.3092711

[84] D. R. MacIver and A. F. Donaldson, "Test-Case Reduction via Test-Case Generation: Insights from the Hypothesis Reducer (Tool Insights Paper)," in *34th European Conference on Object-Oriented Programming (ECOOP 2020)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), R. Hirschfeld and T. Pape, Eds., vol. 166. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020, pp. 13:1–13:27, iSSN: 1868-8969. [Online]. Available: https://drops.dagstuhl.de/opus/volltexte/2020/13170

[85] D. R. MacIver, Z. Hatfield-Dodds *et al.*, "Hypothesis: A new approach to property-based testing," *Journal of Open Source Software*, vol. 4, no. 43, p. 1891, 2019.

[86] D. MacQueen, "Modules for Standard ML," in *Proceedings of the 1984 ACM Symposium on LISP and functional programming*, ser. LFP '84. New York, NY, USA: Association for Computing Machinery, Aug. 1984, pp. 198–207. [Online]. Available: http://doi.org/10.1145/800055.802036

[87] A. Mista and A. Russo, "Deriving compositional random generators," in *Proceedings of the 31st Symposium on Implementation and Application of Functional Languages*, 2019, pp. 1–12.

[88] E. Moggi, "Notions of computation and monads," *Information and computation*, vol. 93, no. 1, pp. 55–92, 1991.

[89] E. Murphy-Hill, C. Jaspan, C. Sadowski, D. Shepherd, M. Phillips, C. Winter, A. Knight, E. Smith, and M. Jorde, "What predicts software developers' productivity?" *IEEE Transactions on Software Engineering*, vol. 47, no. 3, pp. 582–594, 2019.

[90] T. Nelson, E. Rivera, S. Soucie, T. Del Vecchio, J. Wrenn, and S. Krishnamurthi, "Automated, targeted testing of property-based testing predicates," *arXiv preprint arXiv:2111.10414*, 2021.

[91] L. O'Connor and O. Wickström, "Quickstrom: property-based acceptance testing with LTL specifications," in *Proceedings of the 43rd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, ser. PLDI 2022. New York, NY, USA: Association for Computing Machinery, Jun. 2022, pp. 1025–1038. [Online]. Available: https://doi.org/10.1145/3519939.3523728

[92] C. Omar, D. Moon, A. Blinn, I. Voysey, N. Collins, and R. Chugh, "Filling typed holes with live GUIs," in *Proceedings of the Conference on Programming Language Design and Implementation*. ACM, 2021, pp. 511–525.

[93] C. Omar, I. Voysey, R. Chugh, and M. A. Hammer, "Live functional programming with typed holes," in *Proceedings of SIGPLAN Symposium on Principles of Programming Languages*. ACM, 2019, pp. 14:1–14:32, Article 14.

[94] C. Omar, Y. Yoon, T. D. LaToza, and B. A. Myers, "Active code completion," in *Proceedings of the International Conference on Software Engineering*, 2012, pp. 859–869.

[95] S. Oney and B. Myers, "Firecrystal: Understanding interactive behaviors in dynamic web pages," in *Proceedings of the Symposium on Visual Languages and Human-Centric Computing*. IEEE, 2009, pp. 105–108.

[96] T. Paguatte, H. Goldstein, and B. C. Pierce, "PYTEST-MUTAGEN: An open-source tool for mutation testing in python," 2020, https://pypi.org/project/pytest-mutagen/.

[97] M. Papadakis, M. Kintis, J. Zhang, Y. Jia, Y. L. Traon, and M. Harman, "Mutation Testing Advances: An Analysis and Survey," *Advances in Computers*, Jan. 2018. [Online]. Available: http://dx.doi.org/10.1016/bs.adcom.2018.03.015

[98] Z. Paraskevopoulou, C. Hriţcu, M. Dénès, L. Lampropoulos, and B. C. Pierce, "Foundational Property-Based Testing," in *Interactive Theorem Proving*, ser. Lecture Notes in Computer Science, C. Urban and X. Zhang, Eds. Cham: Springer International Publishing, 2015, pp. 325–343.

[99] M. H. Pałka, K. Claessen, A. Russo, and J. Hughes, "Testing an optimising compiler by generating random lambda terms," in *Proceedings of the 6th International Workshop on Automation of Software Test*, ser. AST '11. New York, NY, USA: Association for Computing Machinery, May 2011, pp. 91–97. [Online]. Available: http://doi.org/10.1145/1982595.1982615

[100] B. C. Pierce, "(when) will property-based testing rule the world?" May 2022, keynote at *Yow! Lambda Jam* Conference.

[101] P. Pirolli, "Exploring and finding information," in *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science*. Morgan Kaufmann, 2003, pp. 157–191.

[102] "pytest," https://github.com/pytest-dev/pytest.

[103] "Python Developers Survey 2021 Results." [Online]. Available: https://lp.jetbrains.com/python-developers-survey-2021/

[104] "Wikipedia entry on QuickCheck," https://en.wikipedia.org/wiki/QuickCheck.

[105] S. Reddy, C. Lemieux, R. Padhye, and K. Sen, "Quickly generating diverse valid test inputs with reinforcement learning," in *ICSE '20: 42nd International Conference on Software Engineering, Seoul, South Korea, 27 June - 19 July, 2020*, G. Rothermel and D. Bae, Eds. ACM, 2020, pp. 1410–1421. [Online]. Available: https://doi.org/10.1145/3377811.3380399

[106] M. P. Robillard, "What Makes APIs Hard to Learn? Answers from Developers," vol. 26, no. 6, pp. 27–34, 2009.

[107] RTI, "The economic impacts of inadequate infrastructure for software testing," *National Institute of Standards and Technology*, vol. 1, 2002, prepared by RTI Health, Social, and Economics Research for Gregory Tassey, Ph.D. RTI Project Number 7007.011. https://lara.epfl.ch/w/_media/misc/rti02economicimpactsinadequateinfrastructuresoftwaretesting.pdf.

[108] C. Runciman, M. Naylor, and F. Lindblad, "Smallcheck and Lazy Smallcheck: Automatic exhaustive testing for small values," in *Proceedings of the 1st ACM SIGPLAN Symposium on Haskell, Haskell 2008, Victoria, BC, Canada, 25 September 2008*, A. Gill, Ed. ACM, 2008, pp. 37–48. [Online]. Available: https://doi.org/10.1145/1411286.1411292

[109] "Rust QuickCheck," https://github.com/BurntSushi/quickcheck.

[110] "ScalaCheck," https://github.com/typelevel/scalacheck.

[111] J. Shi, B. Pierce, and A. Head, "Towards a Science of Interactive Proof Reading," in *PLATEAU 2023*, ser. Proceedings of the 2023 PLATEAU Workshop, 2023. [Online]. Available: https://jwshi21.github.io/Plateau.pdf

[112] C. Smith, G. Ferns, and A. Albarghouthi, "Discovering relational specifications," in *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*. Paderborn Germany: ACM, Aug. 2017, pp. 616–626. [Online]. Available: https://dl.acm.org/doi/10.1145/3106237.3106279

[113] E. Soremekun, E. Pavese, N. Havrikov, L. Grunske, and A. Zeller, "Inputs from hell: Learning input distributions for grammar-based test generation," *IEEE Transactions on Software Engineering*, 2020.

[114] P. Srivastava and M. Payer, "Gramatron: Effective grammar-aware fuzzing," in *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2021, pp. 244–256.

[115] "Stack Overflow Developer Survey 2022." [Online]. Available: https://survey.stackoverflow.co/2022/?utm_source=social-share&utm_medium=social&utm_campaign=dev-survey-2022

[116] D. Steinhöfel and A. Zeller, "Input invariants," in *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2022. New York, NY, USA: Association for Computing Machinery, Nov. 2022, pp. 583–594. [Online]. Available: https://doi.org/10.1145/3540250.3549139

[117] R. Suzuki, G. Soares, A. Head, E. Glassman, R. Reis, M. Mongiovi, L. D'Antoni, and B. Hartmann, "Tracediff: Debugging unexpected code behavior using trace divergences," in *Proceedings of the Symposium on Visual Languages and Human-Centric Computing*. IEEE, 2017, pp. 107–115.

[118] S. L. Tanimoto, "Viva: A visual language for image processing," vol. 1, no. 2, pp. 127–139, 1990.

[119] "Testing Extensions for Visual Studio Code in Marketplace." [Online]. Available: https://marketplace.visualstudio.com/search?sortBy=Installs&category=Testing&target=VSCode

[120] E. Torlak and R. Bodik, "Growing solver-aided languages with rosette," in *Proceedings of the 2013 ACM international symposium on New ideas, new paradigms, and reflections on programming & software*, ser. Onward! 2013. New York, NY, USA: Association for Computing Machinery, Oct. 2013, pp. 135–152. [Online]. Available: https://doi.org/10.1145/2509578.2509586

[121] G. Uddin and M. P. Robillard, "How API documentation fails," vol. 32, no. 4, pp. 68–75, 2015.

[122] S. Veggalam, S. Rawat, I. Haller, and H. Bos, "Ifuzzer: An evolutionary interpreter fuzzer using genetic programming," in *European Symposium on Research in Computer Security*. Springer, 2016, pp. 581–601.

[123] J. Wang, B. Chen, L. Wei, and Y. Liu, "Superion: Grammar-aware greybox fuzzing," in *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*. IEEE, 2019, pp. 724–735.

[124] J. O. Wobbrock and J. A. Kientz, "Research contributions in human-computer interaction," vol. 23, no. 3, pp. 38–44, 2016.

[125] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer, "Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 649–658, Jan. 2016, conference Name: IEEE Transactions on Visualization and Computer Graphics.

[126] K. Wongsuphasawat, Z. Qu, D. Moritz, R. Chang, F. Ouk, A. Anand, J. Mackinlay, B. Howe, and J. Heer, "Voyager 2: Augmenting Visual Analysis with Partial View Specifications," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. Denver Colorado USA: ACM, May 2017, pp. 2648–2659. [Online]. Available: https://dl.acm.org/doi/10.1145/3025453.3025768

[127] J. Wrenn, T. Nelson, and S. Krishnamurthi, "Using relational problems to teach property-based testing," *The art science and engineering of programming*, vol. 5, no. 2, 2021.

[128] L.-y. Xia, D. Orchard, and M. Wang, "Composing bidirectional programs monadically," in *European Symposium on Programming*. Springer, 2019, pp. 147–175.

[129] M. Zalewski, "AFL quick start guide," http://lcamtuf.coredump.cx/afl/QuickStartGuide.txt, Apr. 2018.