[15] N. Mirzaei, J. Garcia, H. Bagheri, A. Sadeghi, and S. Malek, "Reducing combinatorics in GUI testing of android applications," in Proceedings of the 38th International Conference on Software Engineering, 2016, pp. 559-570.

[16] K. Song, A.-R. Han, S. Jeong, and S. D. Cha, "Generating various contexts from permissions for testing Android applications," in SEKE, 2015, pp. 87-92.

[17] A. Méndez-Porras, C. Quesada-López, and M. Jenkins, "Automated testing of mobile applications: a systematic map and review," in XVIII Ibero-American Conference on Software Engineering, Lima-Peru, 2015, pp. 195-208.

[18] D. Amalfitano, A. R. Fasolino, P. Tramontana, and N. Amatucci, "Considering context events in event-based testing of mobile applications," in IEEE Sixth International Conference on Software Testing, Verification and Validation Workshops (ICSTW), 2013, pp. 126-133.

[19] W. Song, X. Qian, and J. Huang, "Ehbdroid: beyond GUI testing for android applications," in Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering, 2017, pp. 27-37.

[20] A. Machiry, R. Tahiliani, and M. Naik, "Dynodroid: An input generation system for android apps," in Proceedings of the 9th Joint Meeting on Foundations of Software Engineering, 2013, pp. 224-234.

[21] T. Y. Chen, F.-C. Kuo, R. G. Merkel, and T. Tse, "Adaptive random testing: The art of test case diversity," Journal of Systems and Software, vol. 83, pp. 60-66, 2010.

[22] T. Griebe and V. Gruhn, "A model-based approach to test automation for context-aware mobile applications," in Proceedings of the 29th Annual ACM Symposium on Applied Computing, 2014, pp. 420-427.

[23] T. A. Majchrzak and M. Schulte, "Context-dependent testing of applications for mobile devices," Open Journal of Web Technologies (OJWT), vol. 2, pp. 27-39, 2015.

[24] S. Yu and S. Takada, "Mobile application test case generation focusing on external events," in Proceedings of the 1st International Workshop on Mobile Development, 2016, pp. 41-42.

[25] A. Bartel, J. Klein, Y. Le Traon, and M. Monperrus, "Dexpler: converting android dalvik bytecode to jimple for static analysis with soot," in Proceedings of the ACM SIGPLAN International Workshop on State of the Art in Java Program analysis, 2012, pp. 27-38.

[26] B. Wichmann, A. Canning, D. Clutterbuck, L. Winsborrow, N. Ward, and D. Marsh, "Industrial perspective on static analysis," Software Engineering Journal, vol. 10, pp. 69-75, 1995.

[27] S. Yang, H. Wu, H. Zhang, Y. Wang, C. Swaminathan, D. Yan, et al., "Static window transition graphs for Android," Automated Software Engineering, vol. 25, pp. 833-873, 2018.

[28] W. Yang, M. R. Prasad, and T. Xie, "A grey-box approach for automated GUI-model generation of mobile applications," in International Conference on Fundamental Approaches to Software Engineering, 2013, pp. 250-265.

[29] I. A. Salihu, R. Ibrahim, and A. Mustapha, "A Hybrid Approach for Reverse Engineering GUI Model from Android Apps for Automated Testing," Journal of Telecommunication, Electronic and Computer Engineering (JTEC), vol. 9, pp. 45-49, 2017.

[30] "GATOR: Program Analysis Toolkit For Android."

[31] S. Mujahid, R. Abdalkareem, and E. Shihab, "Studying permission related issues in android wearable apps," in 2018 IEEE International Conference on Software Maintenance and Evolution (ICSME), 2018, pp. 345-356.

[32] A. Usman, N. Ibrahim, and I. A. Salihu, "Test Case Generation from Android Mobile Applications Focusing on Context Events," in Proceedings of the 2018 7th International Conference on Software and Computer Applications, 2018, pp. 25-30.

[33] J. Levinson, Software Testing with Visual Studio 2010: Pearson Education, 2011.

[34] F. Horváth, T. Gergely, Á. Beszédes, D. Tengeri, G. Balogh, and T. Gyimóthy, "Code coverage differences of Java bytecode and source code instrumentation tools," Software Quality Journal, vol. 27, pp. 79-123, 2019.

[35] 2018 Emma, An open source Java code coverage tool [Online]. Available: http://emma.sourceforge.net/.

[36] G. Gay, M. Staats, M. Whalen, and M. P. Heimdahl, "The risks of coverage-directed test case generation," IEEE Transactions on Software Engineering, vol. 41, pp. 803-819, 2015.

[37] L. Inozemtseva and R. Holmes, "Coverage is not strongly correlated with test suite effectiveness," in Proceedings of the 36th International Conference on Software Engineering, 2014, pp. 435-445.

[38] R. Gopinath, C. Jensen, and A. Groce, "Code coverage for suite evaluation by developers," in Proceedings of the 36th International Conference on Software Engineering, 2014, pp. 72-82.

[39] M. Papadakis, M. Kintis, J. Zhang, Y. Jia, Y. Le Traon, and M. Harman, "Mutation testing advances: an analysis and survey," in Advances in Computers. vol. 112, ed: Elsevier, 2019, pp. 275-378.

[40] Y. Wei, "MuDroid: Mutation Testing for Android Apps," 2016.

[41] Y. Jia and M. Harman, "An analysis and survey of the development of mutation testing," IEEE transactions on software engineering, vol. 37, pp. 649-678, 2011.

[42] C. Iida and S. Takada, "Reducing mutants with mutant killable precondition," in 2017 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), 2017, pp. 128-133.