Automated GUI Testing of Android Apps: From Research to Practice

Kevin Moran
College of William & Mary
Williamsburg, VA, USA, 23185
kpmoran@cs.wm.edu

Mario Linares-Vásquez
Universidad de los Andes
Bogotá, Colombia, 111711
m.linaresv@uniandes.edu.co

Denys Poshyvanyk
College of William & Mary
Williamsburg, VA, USA, 23185
denys@cs.wm.edu

***Abstract*—The last decade has seen tremendous proliferation of mobile computing in our society. Billions of users have access to millions of mobile apps that can be installed directly on their mobile devices, electrical appliances, and watches. Factors such as new monetization/revenue models, programming models, and distribution infrastructures contribute to an "attractive" movement that captivates new and traditional developers, as well as a crowd of other professionals that explore, design, and implement mobile apps. Also, the need for "enterprise apps" that support start-ups or serve as a new front-end for traditional companies is pushing software-related professionals to embrace mobile technologies. However, the nature of the economy (devices, apps, markets) imposes new challenges on how mobile apps are envisioned, designed, implemented, tested, released, and maintained. This technology briefing aims to help address the challenges of testing and maintaining mobile apps by providing participants from both academic and industrial backgrounds with information on the state-of-art and state-of-practice mobile testing techniques. Specifically, we aim to (i) highlight new techniques and methodologies for making effective automated testing of mobile apps practical and accessible to developers; and (ii) discuss open academic research questions related to such technology transfer.**

***Index Terms*—Android, GUI, testing, automation, apps**

I. Overview and Objective

Industrial mobile developers and testers face the following critical challenges: (i) continuous pressure from the market for frequent releases, (ii) platform fragmentation at device and OS levels, and (iii) rapid platform/library evolution and API instability. Therefore, continuous testing of mobile apps on a large set of device configurations and under different contextual events is a "must-have" component of the development process to ensure the quality of mobile apps. However, this must be enabled within the constraints of tight release schedules and limited developer and hardware resources. Additionally, both practitioners and researchers must contend with mobile-specific challenges during the execution and testing of mobile apps including: their event-driven nature, gesture-based interactions, interfaces with sensors, and a multitude of possible contextual states.

This technology briefing aims to provide participants with up-to-date information on the state-of-the-art and state-of-the practice regarding mobile app testing. Specifically it will address, the challenges, approaches, tools, and best practices for GUI testing of Android apps, with a particular focus on balancing practicality and research innovation. The technology briefing content will help participants understand the main challenges behind mobile testing, and will provide them with useful and actionable information concerning the pros and cons of the approaches and tools available for mobile app GUI testing. Our focus on GUI testing is motivated by the fact that mobile apps are event-driven and the user interaction is GUI gesture-driven; also, the proposers of this briefing have extensive experience in both using dynamic analysis and building tools for GUI-based testing of Android apps. The content of this briefing is based on the knowledge and experience gathered during the last four years of academic research and industrial collaborations of the authors.

II. Importance to the Community

Taking into account the challenges in both communities mentioned above (industry and academia), there is a clear need for automated methods/tools that support dynamic analysis and testing of mobile apps. In particular, the pervasive hardware and platform fragmentation impose further complexity on automated testing approaches. However, while there are several tools available to support automated execution of Android apps for validation purposes, in practice, testing is still performed mostly manually as recent survey study results show [2], [3], [9]. Major reasons for this preference are the reduced time available for testing, limitations of available tools, lack of knowledge regarding state-of-the-art approaches, and lack of support for tailored large-scale testing. Thus, a corresponding major goal for this technology briefing is to expose researchers and practitioners to the challenges, approaches, tools, and best practices for testing of Android apps.

Developers and researchers currently lack specifications for low cost, practical and open source-based solutions that enable scalable parallel execution of Android apps. Specifically, recent research efforts forgo discussion of the challenges and issues of enabling such a framework, particularly in the context of mobile testing. This leaves both academic research teams and industrial partitioners without a clear set of guidelines, experiences or best practices. Therefore, we will briefly present and demonstrate our results and experiences in this subject, as part of our recent research efforts and collaboration with a major telecommunication company. In particular, we will present a taxonomy of issues experienced,

the solutions we designed for key issues, a set of guidelines, and an experimental framework which should prove useful for those interested in designing infrastructures for large-scale execution/testing of Android apps. In summary, we expect this briefing to be useful both to researchers concerned with state-of-the-art automated mobile testing, and those interested in large-scale dynamic analysis of Android apps.

III. Content of the Briefing

In the last four years, the presenters have designed and validated several solutions to support evolution and maintenance of Android apps [1], [4]–[13]. The content of this briefing will draw from the knowledge base gained throughout the course of these projects and includes (i) an overview of the current concepts and approaches from both industry and academia related to mobile application testing and maintenance with an emphasis on automated approaches, (ii) an overview of the major challenges in testing mobile apps, (iii) the author's vision of solutions to some of these challenges as well as open questions for future research, and (iv) guidelines for attendees to build their own infrastructure for large-scale testing. The slides covered in the briefing as well as additional materials will be made available online. In particular, the website will include (i) links to the tools useful for automated GUI testing, (ii) solutions we have prepared for some accidental/essential issues of automated GUI testing, and (iii) guidelines for a large-scale infrastructure enabling execution and testing of Android apps. The presenters will also leave sufficient time at the end of the presentation for questions and discussion. We hope that this presentation will provide a useful overview of the current landscape of both academic and commercial solutions for supporting the testing of mobile applications.

IV. Presenters

Kevin Moran is currently a Ph.D student in the Computer Science Department at the College of William and Mary (W&M). He is a member of the SEMERU research group and advised by Dr. Denys Poshyvanyk. His main research interests include software engineering, maintenance, and evolution with a focus on mobile platforms. Additionally, he explores applications of data mining and machine learning to software engineering problems. He graduated with a M.S. degree from William & Mary in August of 2015 and his thesis focused on improving bug reporting for mobile apps. He has published in several top peer-reviewed software engineering venues including: ICSE, ESEC/FSE, ICST, and MSR. He was recently the second-overall graduate student winner in the ACM Student Research Competition at ESEC/FSE'15. Moran has served as an external reviewer for ICSE, ICSME, FSE, and SCAM. More information available at http://www.kpmoran.com.

Mario Linares-Vásquez has recently received his Ph.D. from W&M, advised by Dr. Poshyvanyk, and he has joined Universidad de los Andes (Colombia), as a tenure-track Assistant Professor. His research interests include evolution and maintenance of mobile apps, automated GUI testing, and applications of data mining and machine learning techniques to support software engineering tasks. The topic of his dissertation focused on supporting evolution and maintenance of Android apps. His papers have been published in top peer-reviewed software engineering venues such as TSE, EMSE, FSE, ICSE, ISSTA, ASE, ICSME, ICST, and MSR. He was awarded with a Best Paper Award at ICSME'13 and an ACM SIGSOFT Distinguished Paper Award at ESEC/FSE'15. Linares-Vásquez has served as a program committee member for MSR, SANER, and ICPC. He has also reviewed for notable journals, including EMSE, JSS, IST, and IEEE Software. More information available at http://www.cs.wm.edu/~mlinarev.

Denys Poshyvanyk is an Associate Professor in the Computer Science Department at W&M where he leads SEMERU research group. He received his Ph.D. from Wayne State University, where he was advised by Dr. Andrian Marcus. His current research is in the area of software engineering, evolution and maintenance, program comprehension, reverse engineering, software privacy, repository mining, traceability, performance testing, mobile app (Android) development and testing, energy consumption, and reuse. He has received several Best Paper Awards at ICPC'06, ICPC'07, ICSM'10, SCAM'10, ICSM'13 and ACM SIGSOFT Distinguished Paper Awards at ASE'13, ICSE'15, ESEC/FSE'15 and ICPC'16. He is also a recipient of the NSF CAREER award (2013) and the Plumeri Award for Faculty Excellence (2016). Dr. Poshyvanyk has previously presented a technical briefing at ICSE'12 on "Software Engineering in the Age of Data Privacy". More information available at http://www.cs.wm.edu/~denys.

References

[1] G. Bavota, M. Linares-Vásquez, C. E. Bernal-Cárdenas, M. D. Penta, R. Oliveto, and D. Poshyvanyk. The impact of api change- and fault-proneness on the user ratings of android apps. *IEEE TSE*, 41(4):384–407.

[2] M. E. Joorabchi, A. Mesbah, and P. Kruchten. Real challenges in mobile apps. In *ESEM'13*, 2013.

[3] P. S. Kochhar, F. Thung, N. Nagappan, T. Zimmermann, and D. Lo. Understanding the test automation culture of app developers. In *ICST'15*.

[4] M. Linares-Vásquez. Enabling testing of android apps. In *ICSE'15 - SRC*, pages 763–765, 2015.

[5] M. Linares-Vásquez, G. Bavota, C. Bernal-Cárdenas, M. Di Penta, R. Oliveto, and D. Poshyvanyk. Api change and fault proneness: A threat to the success of android apps. In *ESEC/FSE'13*.

[6] 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 *MSR'14*, pages 2–11, 2014.

[7] M. Linares-Vásquez, G. Bavota, C. E. Bernal-Cárdenas, R. Oliveto, M. Di Penta, and D. Poshyvanyk. Optimizing energy consumption of GUIs in Android apps: a multi-objective approach. In *ESEC/FSE'15*.

[8] M. Linares-Vásquez, B. Dit, and D. Poshyvanyk. An exploratory analysis of mobile development issues using stack overflow. In *MSR'13*.

[9] M. Linares-Vásquez, C. Vendome, Q. Luo, and D. Poshyvanyk. How developers detect and fix performance bottlenecks in android apps. In *ICSME'15*, pages 352–361, 2015.

[10] M. Linares-Vásquez, M. White, C. Bernal-Cárdenas, K. Moran, and D. Poshyvanyk. Mining android app usages for generating actionable gui-based execution scenarios. In *MSR 15*, pages 111–122, 2015.

[11] K. Moran, M. Linares-Vásquez, C. Bernal-Cárdenas, and D. Poshyvanyk. Auto-completing bug reports for android applications. In *ESEC/FSE'15*, pages 673–686, 2015.

[12] K. Moran, M. Linares-Vásquez, C. Bernal-Cárdenas, and D. Poshyvanyk. Fusion: A tool for facilitating and augmenting android bug reporting. In *ICSE'16*, page to appear, 2016.

[13] K. Moran, M. Linares-Vásquez, C. Bernal-Cárdenas, C. Vendome, and D. Poshyvanyk. Automatically discovering, reporting and reproducing android application crashes. In *ICST'16*, pages 33–44, 2016.