POLITECNICO DI TORINO

Dipartimento di Automatica e Informatica

PhD in Computer and Control Engineering

XXXI cycle

Supervisor

Prof. Maurizio Morisio

Software Testing: definition, assessment and evolution of test suites

PhD Candidate:

Riccardo Coppola

1. Context

Market demands for faster delivery and higher quality software, especially in the ever-growing Android context, are progressively becoming more stringent. However, evidence from literature suggests that Android apps, are way less tested than desktop ones. Several techniques exist for testing Android apps and their GUIs, based on the recognition of properties of the elements of the screen (Layout-based tools) or on image-recognition of their visual characteristics. Each of them exposes its benefits and drawbacks, especially in terms of fragility of developed test cases, i.e. the possibility that a test case breaks during the normal evolution of the application. Thus, a key hindrance for software companies becomes how to test the software, due to the intrinsic costs of development, maintenance and evolution of testware.

2. Objectives

The aims of this PhD research are:

- Give insights about the testing habits of Android developers from both the open-source community and the industry, and quantify the use of testing tools among Android open-source applications;
- Provide instruments to track the evolution of test code between different releases of Android applications, and monitor the occurrence of fragilities and other issues;
- Provide tools for Android developers to better adapt their tests to the production code changes, in order to avoid fragility issues and perform better testing.

3. Methodology

The following research activities were conducted to comply with the objectives of the PhD:

- Empirical evaluation of the usability of testing tools and the fragility issue with students and practitioners;
- Study on Diffusion, Evolution an Fragility of open-source Android applications hosted on GitHub;
- Application of Grounded Theory to understand the principal reasons for modifications of GUI test cases [1];
- Analysis of the benefits of a translation-based combined approach for GUI testing of Android applications.

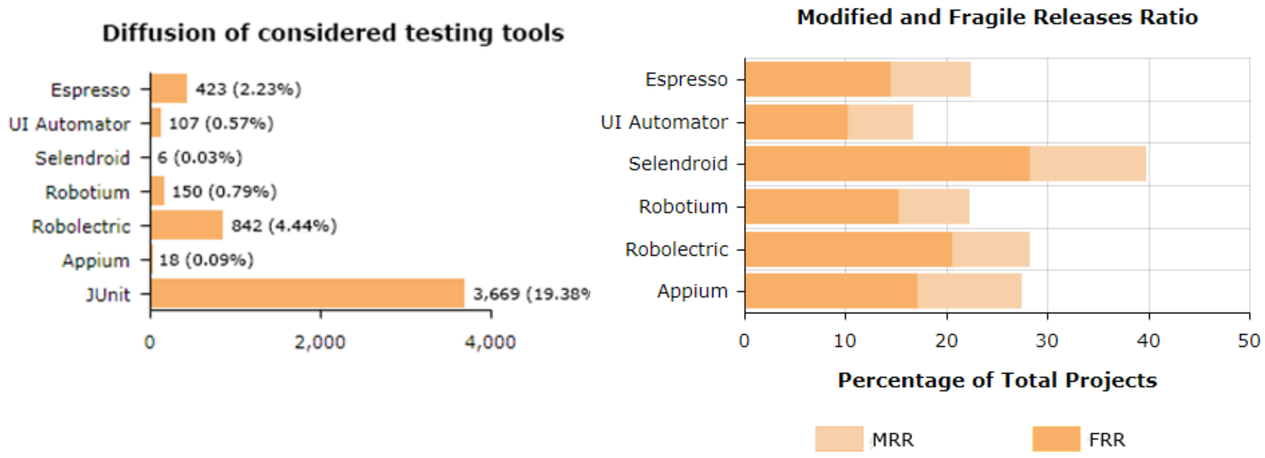

Fig. 1: Statistics about GUI Automation Frameworks usage and test class fragility

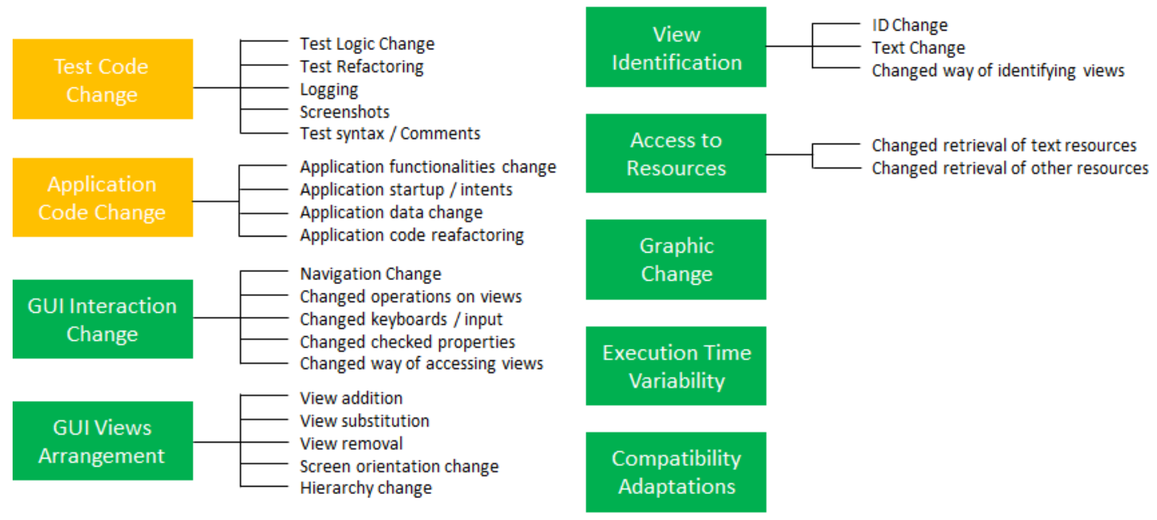

Fig. 2: Taxonomy of change reasons for Android GUI tests

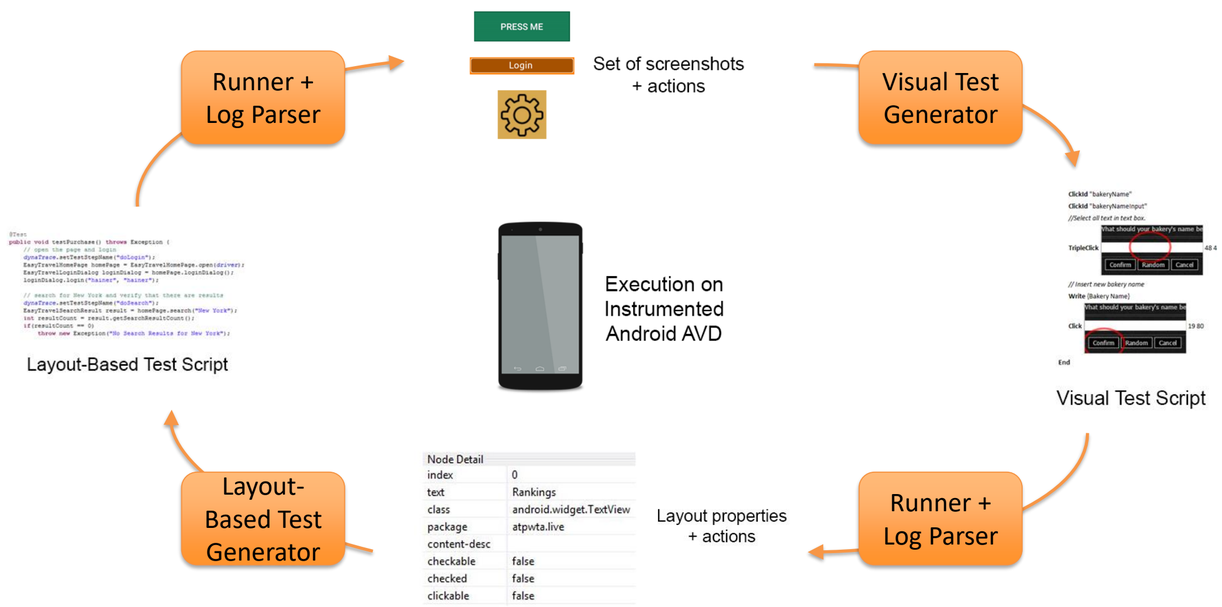

Fig. 3: Architecture of Layout to Visual test translator

4. Results

A set of metrics has been defined to evaluate the diffusion of testing tools, and the evolution and fragility of developed test suites [3]. The most cited GUI Automation Frameworks in literature have been examined on a context of Android apps mined from GitHub (the biggest to date for empirical studies) finding that while the diffusion of Automated testing is very scarce, it requires relevant effort from developers to co-evolve with the app.

A proof-of-concept of a translation-based approach (TOGGLE) has been described. The development is currently in progress [2].

5. Conclusion

State of the art Android GUI testing tools may benefit of additional features reducing the amount of effort needed by testers to keep their scripts up and running. The proposed translation-based testing tool would allow automated porting of existing scripts, reduced maintenance of test cases, reduced impact of fragilities, and automated generation of layout-based (visual) test cases by reusing existing visual (layout-based) test cases.

6. References

[1] Maintenance of Android Widget-based GUI Testing: A Taxonomy of test case modification reasons; Coppola, Riccardo; Morisio, Maurizio; Torchiano, Marco; NEXTA 2018 - 1st IEEE Workshop on NEXt level of Test Automation 2018

[2] Towards Automated Translation between Generations of GUI-based Tests for Mobile Devices ; Coppola, Riccardo; Torchiano, Marco; Ardito, Luca; Alegroth, Emil; INTUITESTBEDS 2018

[3] Mobile GUI Testing Fragility: A Study on Open-Source Android Applications; IEEE Transactions on Reliability; Coppola, Riccardo; Morisio, Maurizio; Torchiano, Marco