Use of robotic arms for automation of android-based mobile app testing through a truly black box perspective

Antonio Adrielson Rodrigues Pinheiro[1,2], **Breno Miranda**[1]

[1] Informatics Center – Federal University of Pernambuco (UFPE)
Recife – PE – Brazil

[2]Programa de Pós-graduação em Ciência da Computação
Nível do curso: Mestrado, Data de início: Mar-22, Data de término: Aug-24

{aarp,bafm}@cin.ufpe.br

Abstract. *As companies strive to reduce the time between software deliveries, manual testing remains a prevalent choice due to its perceived reliability and, in some cases, the lack of expertise in implementing automated test suites that can effectively verify application requirements and functionalities. This is evident in the realm of mobile device testing, where factors such as sensors and hardware can significantly influence the interpretation of test results. This research explores the utilization of robotic arms as a novel approach to enhance automated testing and investigates the impact of software testing on mobile devices with Android from a genuine black box perspective.*

Resumo. *À medida que as empresas se esforçam para reduzir o tempo entre as entregas de software, o teste manual continua sendo uma escolha predominante devido à sua confiabilidade percebida e, em alguns casos, à falta de experiência na implementação de conjuntos de testes automatizados que podem verificar com eficácia os requisitos e funcionalidades do aplicativo. Isso é evidente na área de teste de dispositivos móveis, onde fatores como sensores e hardware podem influenciar significativamente a interpretação dos resultados do teste. Esta pesquisa explora a utilização de braços robóticos como uma nova abordagem para melhorar o teste automatizado e investiga o impacto do teste de software em dispositivos móveis com Android a partir de uma perspectiva genuína de caixa preta.*

1. Introduction and Problem Characterization

With the increasing popularity of mobile devices and the greater accessibility of purchasing electronic products, there is a surging demand for the development of accelerated mobile applications. Manufacturers are in constant competition, introducing new features and innovative forms of interaction, such as camera gestures, motion sensor activation, and other captivating elements to delight the end-users. Consequently, the continuous technological advancements in hardware and software lead to applications with an array of new features, resulting in more exhaustive and complex testing requirements. [Joorabchi et al. 2013]

The diversity of devices poses challenges in test production, with manual testing still prevalent and prone to errors. However, the increasing presence of robots in daily life presents a promising solution. Robots, equipped with their ability to handle repetitive and interactive tasks, can replace manual testing efforts. Leveraging robotic arms can lead to a more efficient and accurate approach to testing.

In practice, many companies encounter challenges when initiating automation processes that require human intervention. Implementing such tests can be a daunting task, leading to uncertainties and obstacles.

The primary goal of this Master's thesis is to develop a prototype of an existing robotic arm in conjunction with test cases. This research will be supported by an Analysis of related works of relevant literature, aimed at creating a comprehensive guideline that serves as a tutorial. The guideline will not only catalogue test cases for specific projects but also provide valuable initial examples for creating customized tests. To achieve this, a test framework based on Pytest will be employed, utilizing the Python programming language.

The main focus of the testing approach will be a truly black box perspective, allowing for comprehensive testing of mobile applications without needing to delve into the internal codebase. By combining the power of robotic arms, automated test cases, and a black box perspective, this research aims to streamline and enhance the mobile app testing process, ultimately leading to more reliable and user-friendly applications.

The remainder of this master thesis proposal is as follows: Section 2 presents the objectives planned to be followed in order to achieve the intended results, Section 3 presents the background, and Section 4 outlines the Methodology. Some preliminary and Expected results are highlighted in Section 5 and Section 6, respectively.

2. Objectives

The general objective of this thesis is to investigate the impact of using robotic arms for test automation on software testing and to develop a comprehensive framework for creating and deploying automated test cases in an open-source manner. Specifically, the thesis aims to achieve the following objectives: 1) Conduct an Analysis of related works of the existing literature to analyze and synthesize the influence of robotic arms on software testing for test automation. 2) Design and configure a robotic arm model that can effectively and non-intrusively perform automated tests on mobile devices. 3) Devise a suite of automated test cases that are easily executable by individuals with basic programming knowledge. 4) Establish a comprehensive guideline for deploying tests in an open-source environment, facilitating the creation of new test cases based on the provided guide.

By accomplishing these objectives, this thesis will contribute to the advancement of software testing methodologies and promote the wider adoption of robotic arms for test automation in the software industry.

3. Background

In contrast to traditional web and desktop applications, mobile applications constantly interact with and respond to the user's context. Mobile apps are exposed to various inputs, including hardware keys and touch screen gestures, while also utilizing multiple sensors available on the device. [Maciel et al. 2022]

Although the most existing techniques adopt intrusive methods, either fully or partially white box, to execute generated test cases. Many of these techniques needs modifications to the app's code or the device platform. Even the so-called 'black box' approaches uses a test harness to establish communication with the app under test. Unfortunately, this approach is not entirely black box, as it involves a machine-to-machine interface between the test harness and the app, potentially compromising the purity of the testing process.

Considerations	Framework Automation	Robotic Automation
Precision and Consistency	Precision and Consistency	Exceptional precision and consistency perform tasks identically each time
Execution Speed	Slower, when dealing with complex interactions or extensive test scenarios	Perform tests at a constant and fast speed
Controlled Environment	Subject to variable conditions such as network and lighting	Take place in a controlled environment minimizing external influence
Detection of Physical Failures	Focused on user interface interactions and is not designed to detect physical issues in the device such as faulty sensors or hardware failures	Programmed to identify physical issues, such as incorrect screen, responsiveness, sensor problems and faulty buttons

Tabela 1. Considerations of an Automated Framework and Robotic Automation

To overcome these limitations and achieve a more realistic testing environment, a non-intrusive or truly black box approach is proposed. A truly black box method refrains from making any assumptions and solely relies on the device-level cyber-physical interface, which emulates human interactions with the app. [Joorabchi et al. 2013]

Since most Android devices come with built-in sensors capable of accurately measuring movement, orientation, and environmental factors they can generate precise raw data and serve multiple purposes, including monitoring device movement, positioning, and environmental changes. These sensors are significantly important as an oracle of testing. For example, gaming applications can harness gravity sensor readings to interpret intricate user gestures like tilt, vibration, rotation, or wobble. Likewise, weather apps make use of temperature and humidity sensors to calculate and report the dew point. In the same vein, travel applications utilize the geomagnetic field sensor and accelerometer to provide cardinal point information. These sensor-based functionalities enhance the user experience and extend the range of applications available on Android devices. [Sikder et al. 2019]

Non-intrusive testing is advantageous for mobile apps and sensitive systems, offering realistic scenarios and preserving system integrity without code changes. It maintains user privacy, doesn't demand elevated permissions, and reduces testing overhead while achieving broad coverage. This method is portable, adapts to dynamic environments, and seamlessly integrates with existing frameworks. It simulates user interactions for a user-centric, comprehensive assessment of software behavior in real-world conditions. [Nistor et al. 2015]

In spite of Appium and similar automation frameworks are powerful tools for mobile device testing, they come with inherent limitations. [Samuel and Pfahl 2016] Mobile device testing tools are limited in their ability to replicate physical and environmental conditions, such as phantom touches and ambient changes. They cannot perform durability tests or assess responses to external interactions like call interruptions. In contrast, a robotic arm excels in these areas, offering comprehensive testing for greater realism and precision in mobile device testing. It can simulate various environmental factors and perform physical durability tests, providing valuable insights. This makes it a vital asset for testing scenarios requiring a high degree of accuracy and realism (see Table 1).

Among various categories of robotic arms, those suitable for supporting tests include collaborative robots, designed to interact with humans in shared spaces, and consumer robots,

Initial Setup	Test Step	Expected Results
1. Gesture App is installed	Enable Gestures on the App	Verify the device is in idle screen
-	Check fast flashlight is enabled	-
-	Press power key	-
-	Performs two chops of the phone to turn ”ON”the feature	Verify Flash light is turned ”ON” and that there is a vibration sound
-	Performs two chops again	Verify Flash light is turned ”OFF” and that there is a vibration sound

Tabela 2. TEST CASE: Chop Twice: Perform Chop Twice on device’s idle screen

available for purchase or DIY - *Do It Yourself*, use to perform specific tasks. [Guizzo 2018] The implementation for this thesis will be entirely based on a robotic arm kit, complemented by affordable, readily available, and interchangeable commodity hardware components.

4. Methodology

The proposed steps for the execution of this master thesis are displayed in Figure 1. The activities that will be develop in each step are detailed as follows.

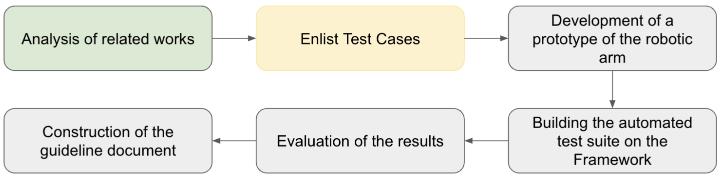

Figura 1. Proposed Master Project steps.

4.1. Analysis of related works

The first step, highlighted in green in Figure 1, is to perform an Analysis of related works of the existing literature to analyze and synthesize the influence of robotic arms on software testing for test automation, which is the ongoing phase phase of this work. To achieve such goals it was conducted a comprehensive literature review to analyze existing research and studies related to automated testing, robotic arms, and mobile app testing, and currently identify the strengths, limitations, and gaps in the current state of the art, and by the end, using the analysis to move forward to the development of the proposed methodology.

4.2. Enlist Test Cases

Directly following the Analysis of related works, collecting scenarios and specific applications along the literature, it will be suitable to: 1) Define the objectives and scope of the test cases to be automated using the robotic arm. 2) Identify critical scenarios and functionalities of Android-based mobile apps that need to be tested. 3) Create a list of test cases that will be automated using the robotic arm to cover various aspects of mobile app functionality, as per example on Table 2.

4.3. Development of a prototype of the robotic arm

Also with the results from the Analysis of related works, it is expect to develop a prototype of the robotic arm, made with materials from a robot kit in addition to affordable, readily available, and interchangeable commodity hardware components. This will be done by selecting a suitable robotic arm model, like Axiz [Mao et al. 2017] and building a custom prototype based on project requirements, then implementing the necessary hardware and software components to enable seamless integration with Android devices for automated testing.

4.4. Building the automated test suite on the Framework

It will be used the Pytest framework, a popular and easy-to-use testing framework for Python. It simplifies the process of writing, organizing, and executing test cases for Python code. Pytest allows developers to write tests in a concise and readable format. Pytest automatically discovers and runs test cases, providing informative and detailed test reports. [krekel 2015] In addition to the developed robotic arm prototype to build an automated test suite, integrating the robotic arm with the testing framework to execute the predefined test cases on Android-based mobile apps, then ensuring the automated test suite is executable.

4.5. Evaluation of the results

In a timely manner following the execution of the automated test cases using the robotic arm on the Android-based mobile device, It is planned to collect and analyze the test results, including pass rates, coverage, and identified defects, and evaluate the effectiveness and efficiency of the robotic arm-based testing approach compared to traditional methods.

4.6. Construction of the guideline document

After performing the suite of tests on the framework, It is expected to collect all the artifacts from this work, as the developing kit for the robotic arm and the source code in an open source repository as well, and by all means, create a comprehensive guideline document that outlines the procedures, best practices, and setup instructions for using the robotic arm for automated testing, Including detailed steps for setting up the robotic arm, preparing Android devices, and executing the predefined test cases, providing guidance to apply non-intrusive tests.

5. Preliminary Results

At the time of this submission we are in the process of concluding the planned Analysis of related works (step in green from Figure 1).

To obtain the articles, we conducted a thorough search on the ScienceDirect digital library and also performed an extensive search on Google Scholar. From the selected primary studies, we are currently extracting data to develop a well-established model for the test scenarios, the robot arm, and the testing framework.

6. Related work

Two main works found in the Analysis literature review are mainly related to this work and their findings contributed to consolidating the ideas and project conception:

The work of [Mao et al. 2017] presents a comparison between simulated-based mobile test automation and their proposed robotic mobile testing. They introduce a robotic mobile device test generator called Axiz and demonstrate its application on the popular Google Calculator app. Axiz generates realistic tests that are executed using the robotic manipulator. A comparison is made with a traditional Nexus 7 tablet that allows intrusive testing. The process involves translating event instructions into motion specifications for Axiz's robotic arm controller. The robotic arm accurately executes the test events and passes all required oracle checkpoints, showcasing the effectiveness of robotic testing for truly black box automation.

In [Maciel et al. 2022] the authors investigate the use of robots for supporting mobile testing. The study conducts a systematic mapping of the literature to explore motivations, types of tests and reported effectiveness/efficiency. Out of 1353 papers, 20 primary studies were selected for analysis. The main reasons for adopting robotic mobile testing include simulating complex physical interactions, achieving more realistic interactions, avoiding manual testing limitations, and performing remote testing. Most selected studies focused on testing non-functional properties at the system level. While many studies reported positive outcomes, more empirical evidence is needed for generalization.

7. Expected Results

Spreading effective orientation for automating tests in a truly black box perspective is crucial for providing better quality assurance understanding and consequently, a pleasanter experience for the users when using the software mobile devices applications. After concluding the prototype, building the automated test suite on the Framework, and evaluating the results, it is supposed to deliver a guideline to be used by the mobile testing community, which will contain the robotic arm kit instructions in addition to the source code of the test suite in an open source repository.

Referências

Guizzo, E. (2018). Types of robots. `https://robotsguide.com/learn/types-of-robots/`. [accessed 01-aug-2023].

Joorabchi, M. E., Mesbah, A., and Kruchten, P. (2013). Real challenges in mobile app development. In *2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 15–24. IEEE.

krekel, h. (2015). pytest: helps you write better programs. `https://docs.pytest.org/en/7.4.x/`. [accessed 02-aug-2023].

Maciel, L., Oliveira, A., Rodrigues, R., Santiago, W., Silva, A., Carvalho, G., and Miranda, B. (2022). A systematic mapping study on robotic testing of mobile devices. In *2022 48th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 475–482. IEEE.

Mao, K., Harman, M., and Jia, Y. (2017). Robotic testing of mobile apps for truly black-box automation. *Ieee Software*, 34(2):11–16.

Nistor, A., Chang, P.-C., Radoi, C., and Lu, S. (2015). Caramel: Detecting and fixing performance problems that have non-intrusive fixes. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, volume 1, pages 902–912. IEEE.

Samuel, T. and Pfahl, D. (2016). Problems and solutions in mobile application testing. In *Product-Focused Software Process Improvement: 17th International Conference, PROFES 2016, Trondheim, Norway, November 22-24, 2016, Proceedings 17*, pages 249–267. Springer.

Sikder, A. K., Aksu, H., and Uluagac, A. S. (2019). A context-aware framework for detecting sensor-based threats on smart devices. *IEEE Transactions on Mobile Computing*, 19(2):245–261.