9th UCAAT User Conference on Advanced Automated Testing

Monkey Test Automation on Android & iOS Native Apps

Lakshminarayanan Krishnamurthy

<epam>

15/09/2022

The Context

Application Background:

- A server driven mobile application development platform
  - Based on Swift and Kotlin
  - 40+ Brands

Testing Objectives:

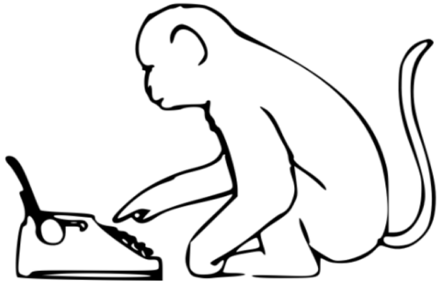
1) Need open-source based framework
   a) Generic, configuration driven test suite
   b) No hard dependency on programming language
   c) Should support both iOS and Android native apps
   d) CI/ CD Integration
2) Uncover application crashes early in the development life cycle

Problem Statement

1. Frequent app crashes on end user devices
2. Application workflow-based test automation across 40+ brands - Not scalable & maintenance intensive.
3. Commercial/ Open-Source Monkey test framework(s) are focused on either Android or iOS
4. Appium based Monkey test solution not available.

What is Monkey Testing?

Monkey testing is a type of testing where we send random inputs to application while checking behavior or seeing whether the application or system will crash. Monkey test is a fast and effective method for testing the stability and robustness of software



Any sequence of events which has a non-zero probability of happening, at least as long as it hasn't occurred, will almost certainly eventually occur.

Key Drivers

Common Reasons for App Crash

- Poor Memory Management
- Inadequate Testing
- Unhandled Errors & Exceptions
- Excessive Code
- Device Incompatibility
- Network Issues
- Poor front-end optimization
- Bandwidth Constraints

Impact

- Bad User Reviews and Ratings
- Mobile App Uninstalls
- Interrupted or Lost Transactions

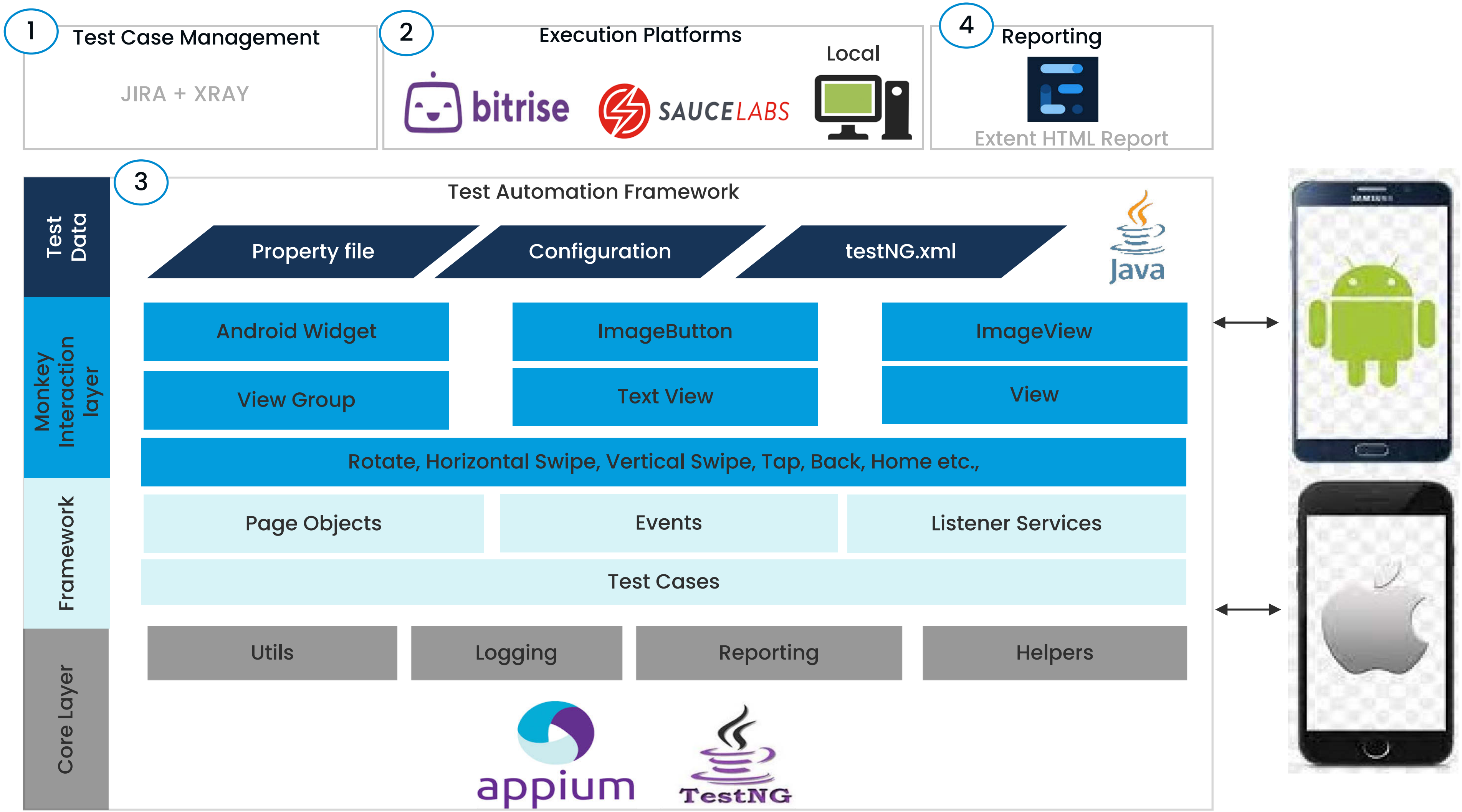
Assessment Focus Areas

1. Android native app support
2. iOS native app support
3. Record and playback option
4. Option to write custom scripts
5. Commercial/ Open Source
6. Integration with CI/CD process
7. Framework development effort
8. Technical Skills required
9. Cloud Farm Availability

Tool Options

1. Firebase Robotest
2. UI/Application Exerciser (ADB)
3. UI Auto Monkey
4. SWIFT Monkey
5. Perfectomator
6. APPIUM ?

Custom Solution Blueprint



Data Collected from Google Firebase

Total number of unique users: 17,71,708

Top 10 Devices where a greater number of events noticed

iOS Devices
iPhone XR
iPhone 11
iPhone 8
iPad
iPhone 8 Plus
iPhone X
iPhone 7
iPhone 12
iPhone 13 Pro Max
iPhone 12 Pro

Android Devices
Samsung Galaxy S21 5G
Samsung Galaxy S9
Samsung Galaxy S21 Ultra 5G
Samsung Galaxy S10
Samsung Galaxy S10e
Samsung Galaxy S10+
Galaxy S8
Samsung Galaxy Note9
Samsung Galaxy S9 Plus
Samsung Galaxy Note 20 Ultra 5G

Observations & Outcome

1. Stress Testing and quick pseudo-random events (Appium vs ADB & Swift Monkey)
2. Handle Video, Images, Advertisement screen etc.,
3. Crashes uncovered during trial runs in CI/CD cycle: (Real Devices vs Emulator)
   - ANR – Application Not Responding
   - Native Crashes
4. Integration of Monkey Tests and Functional Tests
5. Common Framework and Programming language for iOS and Android

Any further questions?

Contact me:
Lakshminarayanan.Krishnamurthy@
Epam.com