Parallel Automated Testing on Mobile Devices through BDD-Cucumber and Open Source Resources

Presented by Jorge Asensio
Summary

1. Why Mobile Testing, DevOps and BDD
   • Background
   • The Mobile DevOps challenge
   • BDD and Mobile Projects

2. A bit of history
   • Our needs and how we tackled them

3. The chain of command
   • Workflow
   • Tools, languages and frameworks
   • Test run

4. Demo of UAT Automation
   • Live demo with 4 different devices

5. Outlook in the future
   • App usage will increase
   • Pitfalls
   • What remains as is and prospective improvements
1- Why Mobile Testing, DevOps and BDD
Background

- Apps are leading the market.
- Many different devices with different OS type/Versions.
- Quick releases: automation solution.
- High demand for mobile testing in DevOps environments.

Source: ComScore Media
Background

- Apps are leading the market.
- Many different devices with different OS type/Versions.
- Quick releases: automation solution.
- High demand for mobile testing in DevOps environments.

Source: ComScore Media
The Mobile DevOps challenge

- Market fragmentation.
- Different tooling than Desktop DevOps.
- Speeding-up DevOps activities: cloud/local/3rd parties.
- No continuous deployments.
- No rollback in releases.
The Mobile DevOps challenge (2)

- Scalable automation.
- Continuous Everything against real environments.
- Poor app instrumentation.
- Back-end changes: services may not serve mobile.
BDD and mobile projects

- Mobile testing is particularly user XP-oriented.
- Gherkin standardizes documentation.
- BDD enhances interdepartmental collaboration.
2- A bit of history
## Our needs and how we tackled them

<table>
<thead>
<tr>
<th>Requirement\Issue</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real environments.</td>
<td><strong>Real devices, self-hosted</strong> solution.</td>
</tr>
<tr>
<td>Time, costs, customizable tools and frameworks.</td>
<td>Free <a href="https://github.com">open-source</a> tools, <strong>self-hosted CI</strong> solution.</td>
</tr>
<tr>
<td>Privacy and security of the data.</td>
<td><strong>Self-hosted</strong> solution.</td>
</tr>
<tr>
<td>Portability.</td>
<td>Wheeled structure.</td>
</tr>
<tr>
<td>Improve \ modify framework.</td>
<td><strong>Scalable code</strong> (e.g. Page Object Model techniques).</td>
</tr>
<tr>
<td>Current commercial tools not satisfying.</td>
<td>Customisable open-source tools.</td>
</tr>
<tr>
<td>Dead batteries.</td>
<td><a href="https://www.usb.org">USB</a> powered on hub machine.</td>
</tr>
</tbody>
</table>
3- The chain of command
Workflow

- Test project triggered from CI.
- Test automation run on devices connected to the slave machine.
- Unified report delivered.
Tools, languages and frameworks
Test run

1. Run test script
2. Run Appium server
3. Create app build (iOS)
4. Call Cucumber
5. Execute webdriver
6. Run test scenario
7. Test report

Cucumber
4- Demo of UAT Automation
Live demo with 4 different devices

- Samsung A5 under Android 7.0
- iPhone 7 under iOS 11.2.4
- Sony Xperia XZ1 under Android 8.0
- HTC U11 under Android 7.1.1

- Threads run in parallel.
- Unified reports.
5- Outlook in the future
App usage will increase

- App download estimation: over 79% increase in the next 4 years
- Market points towards mobile app direction

Source: TechCrunch
Pitfalls

- SDKs and tool versions not aligned with OS.
- Test environment restrictions.
- iOS device driving requirements.
- iOS parallel testing is tricky.
- App code is not test-friendly.
- Powerful hub machines.
- Appium API deprecations.
- Different Android implementations.
- Steep learning curve.
What remains as is and prospective improvements

- Solution remains self-hosted and using Gherkin.

- Performance tests: JMeter/Locust:
  - Measure global performance in terms of time.

- Security tests: OWASP Zed Attack Proxy/ LinkedIn QARK.

- Improved test reports.
Thank you for your attention!