

Mobile Automation Primer

Sher Singh Bhachu
SinfoniQA

Quick creds...

- Worked with mobile apps since 2005...
 - **Citrix Client** for Nokia
 - **Monitise 'Bank Anywhere'** Mobile Banking
 - **PowaTAG** Purchase goods using QR Codes
 - **mPOWA** Chip and Pin mobile app for retailers
 - **SkyGO** Watch Sky channels on the go
 - **Sky Store** Purchase movies to buy or rent on mobile devices
 - **BlinkBox Books** Buy and read eBooks
- Implementation of Continuous Integration
- Implementation of Cloud Based testing
- Implementation of performance and monitoring tracking
- More recently, working towards a Continuous Delivery model

Agenda

- Why bother automating mobile?
- What are the challenges?
- What tools are available?
- What are the 'wins'?
- Useful reading
- Next steps
- Q & A

Why bother?

- Apps change constantly, if yours don't, you're doing something wrong.
- Reduce the time to market.
- 10+ Android , 3+ iOS versions.
- Unlimited* devices.
- Confidence in releases.
- Stop cowboy development, mobile developers are notorious.
- Move towards continuous delivery*.
- ...

Challenges?

- Still 'niche'* ...think of web automation 3+ years ago.
- Cross platform, Native, Hybrid apps*.
- Devices of different screen sizes, etc.
- Constant changes in OS's make it difficult for tools and 3rd party tools to catch up.
- More technical knowledge of the underlying OS is required, more so than web.
 - From a testing perspective, this is the most important thing in my opinion.

Development differences*

- Mobile development is NOT 'just front end stuff'
 - Enforcing good API design
 - Put in place the ability to undertake A/B testing
- Many Android developers don't really know Java, they know 'Android'.
- Unit and integration testing and throw many iOS developers off, most don't even realise they need to do them or how.
 - There has been massive improvements with every new iOS release since iOS6, so use them.
- It's cheaper to start again, seriously.

Tools?

- A lot lot lot more than there is for web
- Apple's 'native' UIAutomator (JavaScript)
- Android 'native' UI Automation (Java)...also a Python one.
- Android has a very good Element Locator tool (ships with the SDK)
- Other tooling:
 - Appium, Calabash, Rubymotion*, Frank, CoffeeScript...
- Windows and Blackberry have proprietary ones.
- Many online 'record and playback' tools.

Tools?

- Cloud tooling
 - TestDroid, Xamarin, etc.
- Getting your app out to a select few or a mass audience
 - TestFlight (purchased by Apple)
 - HockeyApp (purchased by Microsoft)

Tools Comparison

Tool	Platform	Pro's	Con's	Other
Appium Calabash UIAutomation UIAutomator	Android + iOS	<ul style="list-style-type: none"> Basically Selenium WebDriver for mobile* Ruby or Java No modification required to the application Has a server based model 	<ul style="list-style-type: none"> Ruby library is still lagging Constant regressions 	<ul style="list-style-type: none"> Seems to be the defacto where environments have been using Java with WebDriver Its server based model allows you to get performance metrics
	Android + iOS	<ul style="list-style-type: none"> Ruby Comprehensive documentation and very stable Strong developer base 	<ul style="list-style-type: none"> Application requires modification 	<ul style="list-style-type: none"> Generally up to date Fairly active google groups
	Android	<ul style="list-style-type: none"> Well documented Test code can live with project code 	<ul style="list-style-type: none"> Java only No modification required to the app, ok, a few very small lines 	<ul style="list-style-type: none"> Always something new in releases
	iOS	<ul style="list-style-type: none"> Fast as hell Allows you to have very low level control Performance Metrics* No modification required to the app 	<ul style="list-style-type: none"> JavaScript only Constantly breaking Changes made with not enough documentation Documentation is 'meh' 	<ul style="list-style-type: none"> Learn by doing record and play and then view your script

How do the tools work?

- To the whiteboard!!!

What are the wins?

- Watching mobile automation ‘looks’ cool.
- Quick(er) feedback
- Personally, you will learn a lot of technologies and tools...
 - Python, Ruby, JavaScript, Java, Objective C...
- Ability to have a well structured and common release process that is NO different to releasing an API, Web App, etc.

Useful Reading

- Google is your friend
- Git repositories...plenty of examples out there
- Documentation for iOS and Android
 - [UI Testing on Android](#)
 - [UI Testing on Apple](#)
- [Calabash on GitHub](#)
- [Appium on GitHub](#)
- [TestDroid](#)
- [Xamarin Test Cloud](#)

Next steps

- Get reading.
 - Official docs, blogs, tutorials...
- Get coding (in what ever your preferred language is).
- Get talking and ask questions.
 - Google groups, Git repos, ...
 - MeetUps, if that's your 'thing'
- Consider in-house and/or external training.
 - There are people within the business who must have done this before!? Surely?
- Spend your individual training budgets.
 - You have one, trust me!

Q & A

- Any questions?