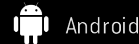




KUPFERWERK®

Best Apps.



Why you want to automate your mobile testing process

Maik Krüger (GDG Regensburg)

05.11.2015

*„Test your software -
or your user will“*

*Andrew Hunt, David Thomas: The Pragmatic Programmer. From Journeyman to
Master (1999)*

„88% of Americans Have Negative Feelings About Brands with Poorly Performing Websites and Mobile Apps“

SOASTA's 2013 Website and Mobile App Survey

- ▶ Majority of users will never return if your app doesn't launch fast enough
- ▶ Users have high expectations when it comes to app quality

Testing

- ▶ In-the-wild testing
- ▶ Automated testing
- ▶ Post-launch analytics

Relying on one QA method is an outdated - and lacking - approach to application quality. In today's world, quality demands a more comprehensive, far-reaching approach.

Applause.com

Automated UI Testing

- ▶ Is performed on the same user interface your daily user is confronted with
- ▶ More reasonable test results
- ▶ Essential part of software development and is getting increasingly popular in mobile development
- ▶ In agile development: automated tests scripts run quickly and effortlessly whenever a new build is added
- ▶ The team can ensure that they are moving in the right direction and not damaging previous work

How it works

```
@smoke
```

```
Scenario: Does the App Start?
```

```
  Given I have started the app
```

```
  Then I see the login page
```

```
@ios @login @login_success
```

```
Scenario: I enter the right password
```

```
  Given I have started the app
```

```
  When I enter the right username
```

```
  And I enter the right password
```

```
  Then I see the main page
```


@login @login_failure

Scenario: I enter the wrong password

Given I have started the app

When I enter the right username

When I enter a wrong password

Then I see a failure message

When I click "OK"

Then I see the login page

Step definition

```
Given(/^I have started the app$/) do
  @main_page = MyPage.new
  ignore_popup
end

And(/^I see "([^"]*)"$/) do |arg|
  @main_page.check_selectItem_byName(arg)
end

When(/^I enter a wrong username$/) do
  @main_page.enter_wrong_username
end

Then(/^I see a failure message$/) do
  $driver.set_wait 0
  $driver.wait(5) { @main_page.see_error_message }
end
```

```
def see_main_page
  sleep(7)
  $driver.find_element(:id, "Image")
  $driver.find_element(:id, "Menu")
end
```

Results

```
# todo: check for allowing location services
@ios @login @login_success
Scenario: I enter the right password # features/AUMSI.feature:32
  Given I have started the app      # features/step_definitions/AUMSI.rb:24
    ERROR -- TAKING SCREENSHOT
    Ignore popup
    Ignore popup
    Skip Onboarding
  When I enter the right username    # features/step_definitions/AUMSI.rb:87
  And I enter the right password     # features/step_definitions/AUMSI.rb:91
  Then I see the main page           # features/step_definitions/AUMSI.rb:62
```

```
@ios @login @login_failure
Scenario: I enter the wrong password # features/AUMSI.feature:21
  Given I have started the app      # features/step_definitions/AUMSI.rb:24
    Ignore popup
    Ignore popup
    Skip Onboarding
  When I enter the right username    # features/step_definitions/AUMSI.rb:87
  When I enter a wrong password     # features/step_definitions/AUMSI.rb:50
  Then I see a failure message      # features/step_definitions/AUMSI.rb:37
    timed out after 3 seconds () (Selenium::WebDriver::Error::TimeoutError)
    ./features/step_definitions/AUMSI.rb:39:in `/^I see a failure message$/'
    features/AUMSI.feature:25:in `Then I see a failure message'
  When I click "OK"                 # features/step_definitions/AUMSI.rb:42
  Then I see the login page         # features/step_definitions/AUMSI.rb:46
```

Failing Scenarios:

```
cucumber -p appium_ios8.4 features/AUMSI.feature:21 # Scenario: I enter the wrong password
```

5 scenarios (1 failed, 4 passed)

20 steps (1 failed, 2 skipped, 17 passed)

5m31.885s

Where to run your automated tests

- ▶ Emulator/Simulator
- ▶ Google Cloud Test Lab
- ▶ Real devices



Limitations

- ▶ Development time
- ▶ Maintenance
- ▶ Real environment (bad connectivity, GPS, ...)
- ▶ Layout issues
- ▶ Confined, lab-based environment far removed from real life situations
- ▶ Human ingenuity, creativity and even stupidity

Benefits

- ▶ Integration into daily business
- ▶ Permanent control over an ongoing development process
- ▶ Swifter detection of regressions in development
- ▶ Less risk when making backend and infrastructure changes
- ▶ shorter time to market for new features
- ▶ Faster iterations (more confident developer, manual testers have more time for real world tests)

How to automate your process



1. Plan – do your work up front

- ▶ User stories
- ▶ Basis: Manual testcases
- ▶ First concentrate on breadth, then depth
- ▶ Do not underestimate the effort and complexity involved

2. Understand the limitations of automation

- ▶ Reality check – in a perfect world, everything could be automated, unfortunately, it is not
- ▶ Automation can not – and never will – replace manual testers

3. Use the right talent and tool

- ▶ Wrong way: non-technical or development personnel
- ▶ Tool: platforms, maintenance

4. Build a process that's easy to maintain

- ▶ Sustainability
- ▶ Think ahead
- ▶ Regular script executions

5. Report

- ▶ Grant visibility into the process/results
- ▶ Stakeholders need informations to understand the value automation is (or is not) providing

Conclusion

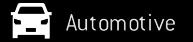
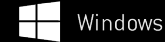
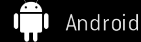
- ▶ All departments from design to development to QA have responsibility that the product works
- ▶ Test early and test often
- ▶ Combine automated and manual testing to cover your app from every angle.

Best practices

- ▶ Integrate your framework with your systems early in the process, including your CI, test management and defect tracking systems as possible
- ▶ Begin with your most critical cases
- ▶ Involve development to make sure all product objects are uniquely identified
- ▶ Focus the automation team on creating, executing and maintaining scripts. Failures should be investigated to ensure that they are related to a bug, and then passed to the test or development team
- ▶ Utilize real devices and real networks
- ▶ Time – bound tests (test execution should take part in an appropriate time)

More information

- ▶ Applause.com
 - ▶ The Key Steps of Comprehensive App Testing
 - ▶ 5 Ways to Win With Mobile Automation
 - ▶ The essential guide to mobile app testing
- ▶ Oracle
 - ▶ When to automate your test (and when not to)



Thanks!

Maïke Krüger

krueger@kupferwerk.com