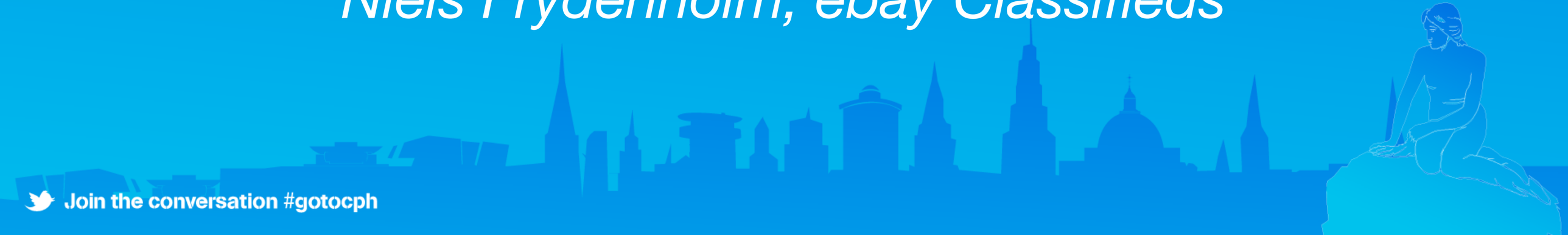
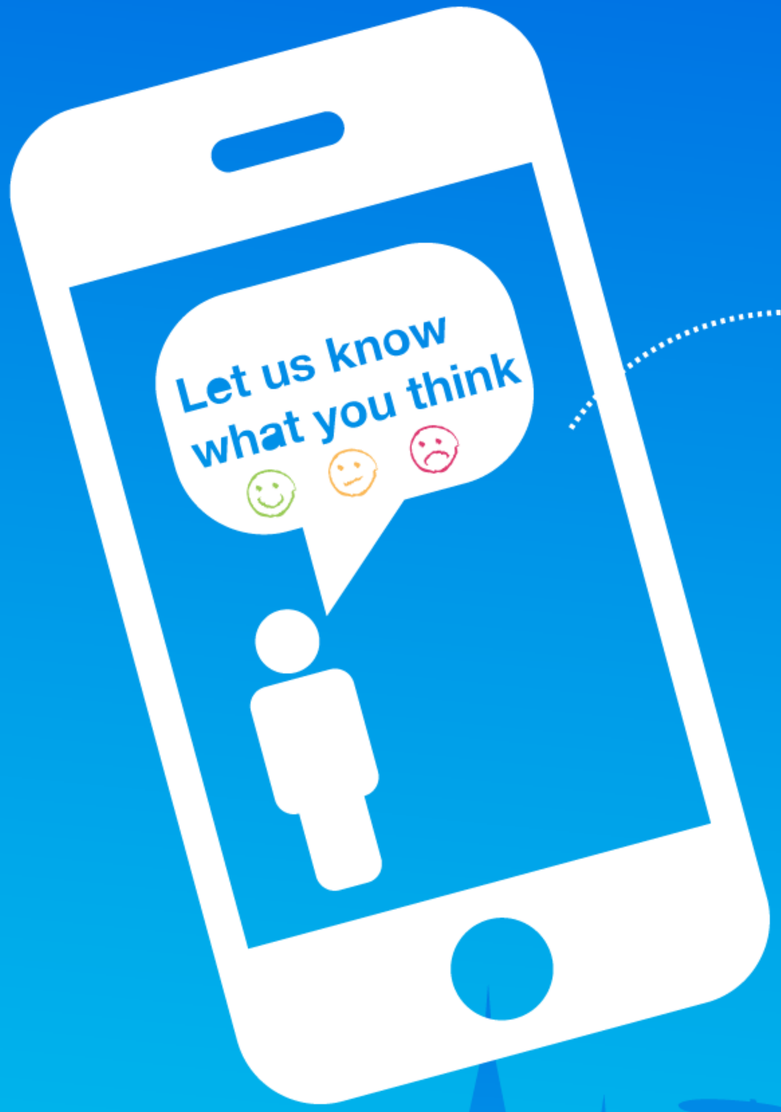
goto; conference

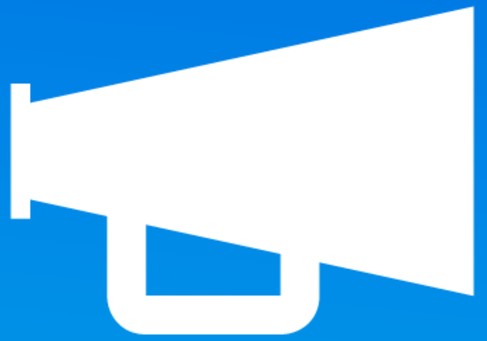
Continuous delivery for native apps

Niels Frydenholm, ebay Classifieds



goto; conference





Click ‘engage’
to rate sessions
and ask questions

Join the conversation #gotocph

Continuous delivery

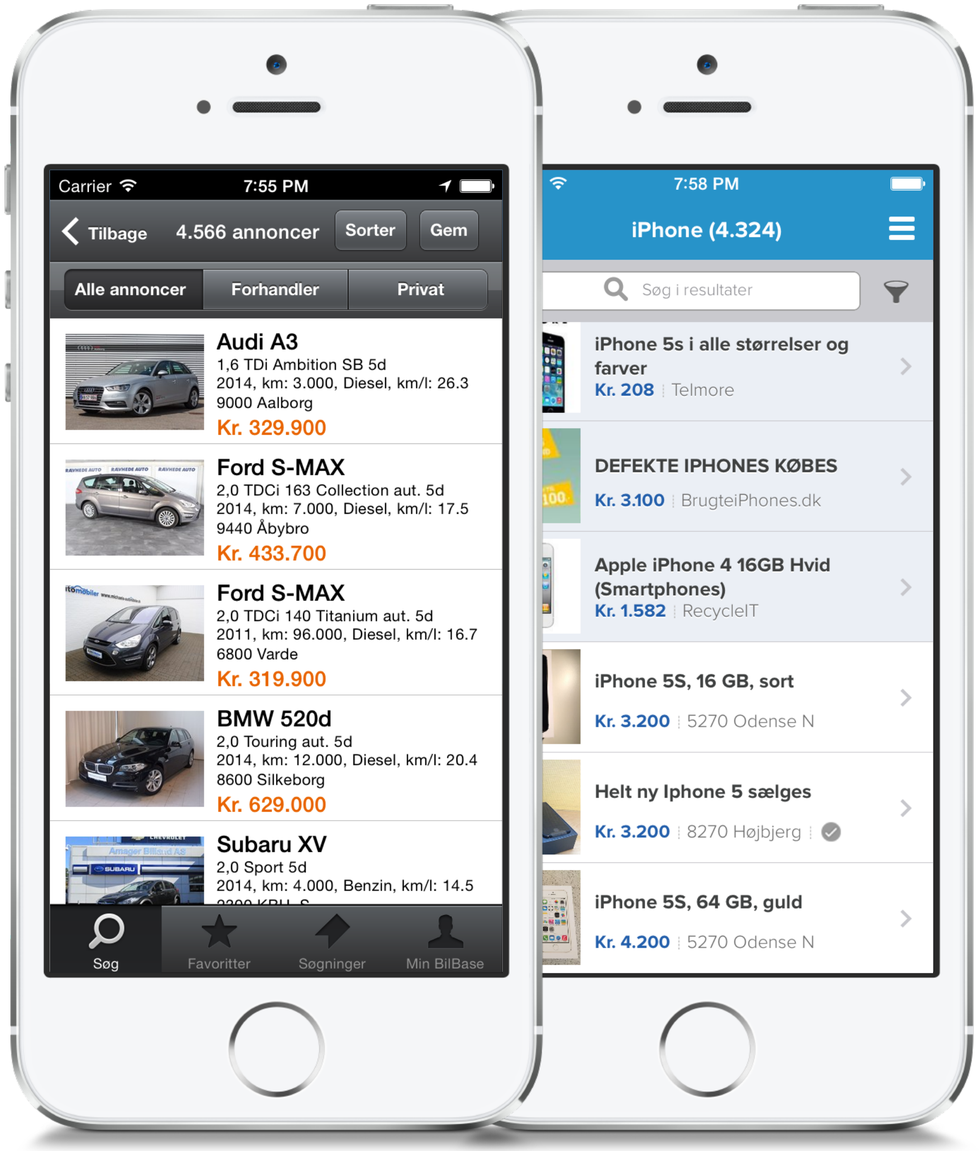
Continuous delivery

Being able to build a release-ready binary at any given time.

ebay Classifieds, Denmark

ebay Classifieds, Denmark

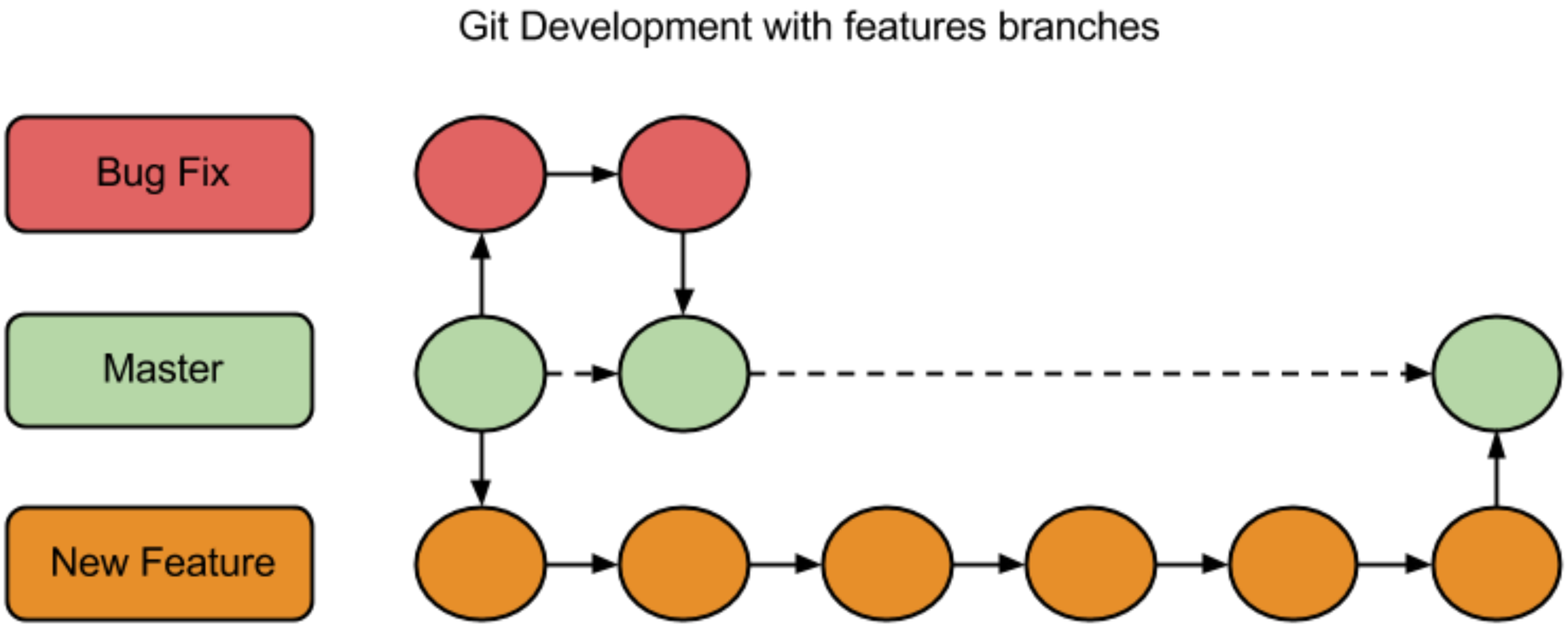
- Native apps
  - DBA
  - BilBasen



Tools that support multiple platforms

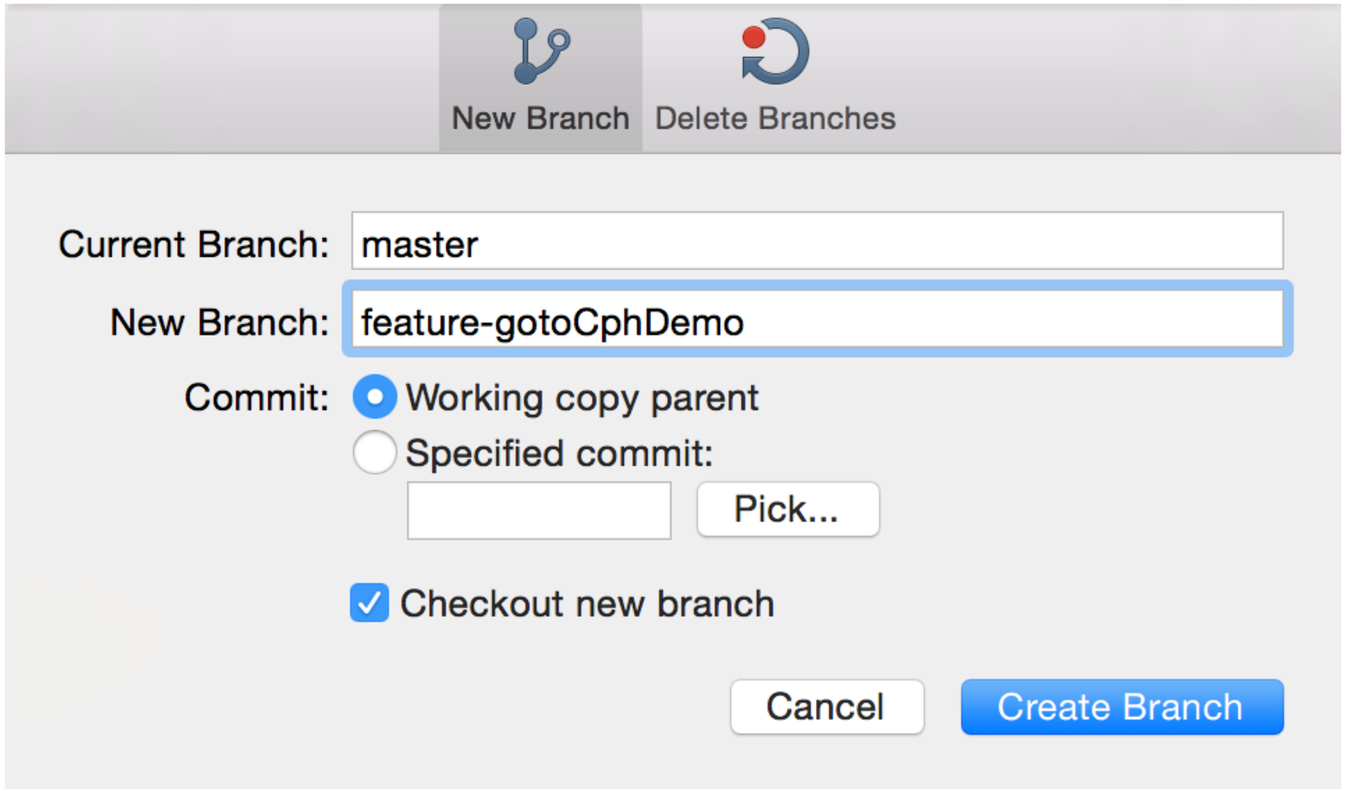
Branches, build and distribution

Feature branches



Jenkins jobs pr. branch

Jenkins jobs pr. branch



Jenkins jobs pr. branch

BB-feature-gotoCphDemo-HockeyApp	1 hr 5 min - #1	N/A
BB-feature-gotoCphDemo-Integration	N/A	N/A
BB-feature-gotoCphDemo-UI-iPad	N/A	N/A
BB-feature-gotoCphDemo-UI-iPhone	N/A	N/A
BB-feature-gotoCphDemo-Unit	N/A	N/A
BB-Master-HockeyApp-T	13 hr - #639	N/A
BB-Master-Infrastructure-Library-Unit	13 hr - #739	12 days - #730
BB-Master-Integration-T	12 hr - #1116	N/A
BB-Master-UI-Cloud-iPad	21 hr - #288	4 days 5 hr - #286
BB-Master-UI-Cloud-iPhone	20 hr - #390	4 days 5 hr - #388
BB-Master-UI-iPad-T	11 hr - #683	5 days 11 hr - #681
BB-Master-UI-iPhone-T	12 hr - #950	2 days 12 hr - #948
BB-Master-Unit-T	12 hr - #968	5 days 12 hr - #964

Jenkins jobs pr. branch - configuration

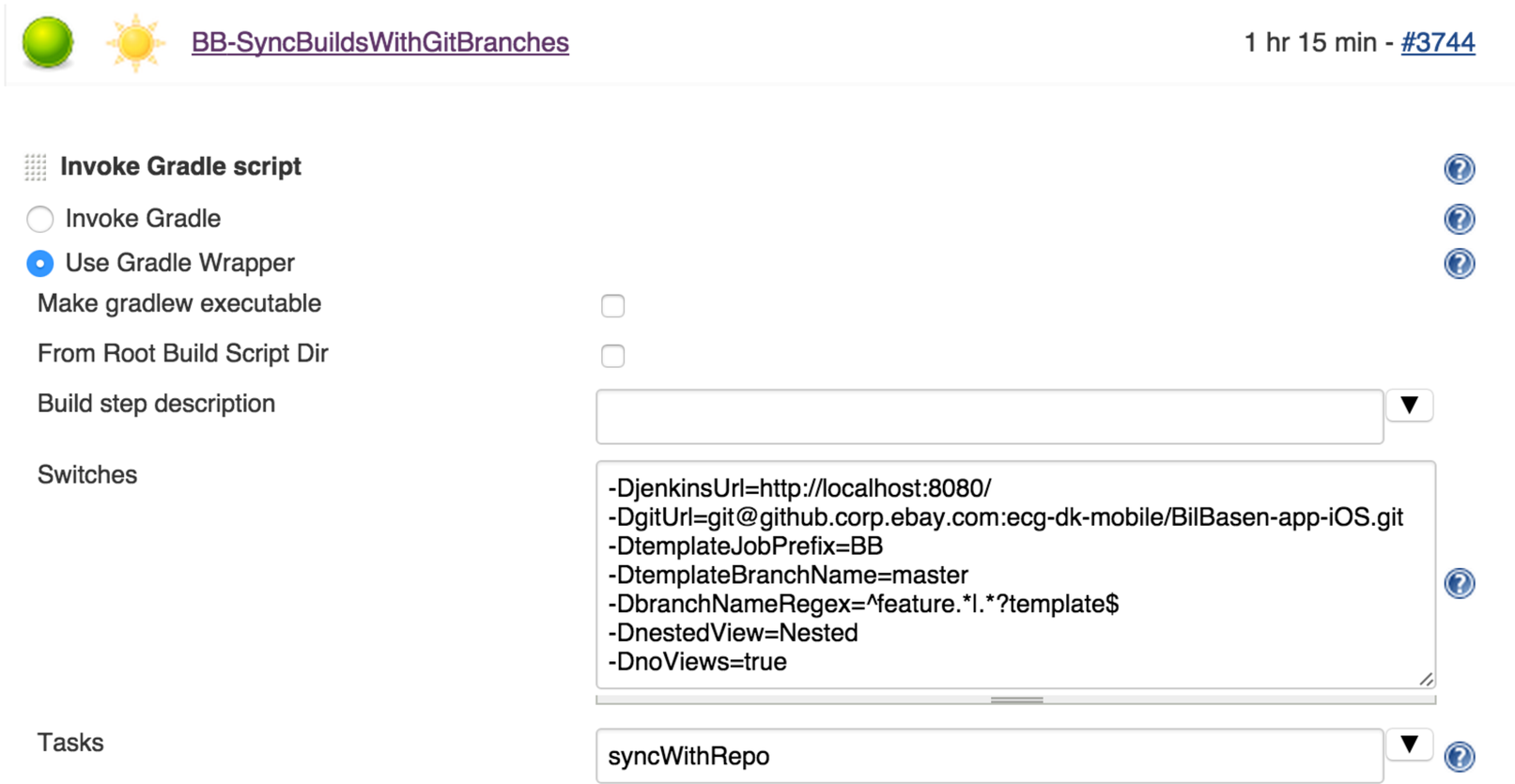
- Plugin: http://entagen.github.io/jenkins-build-per-branch/

Jenkins jobs pr. branch - configuration

- Plugin: http://entagen.github.io/jenkins-build-per-branch/

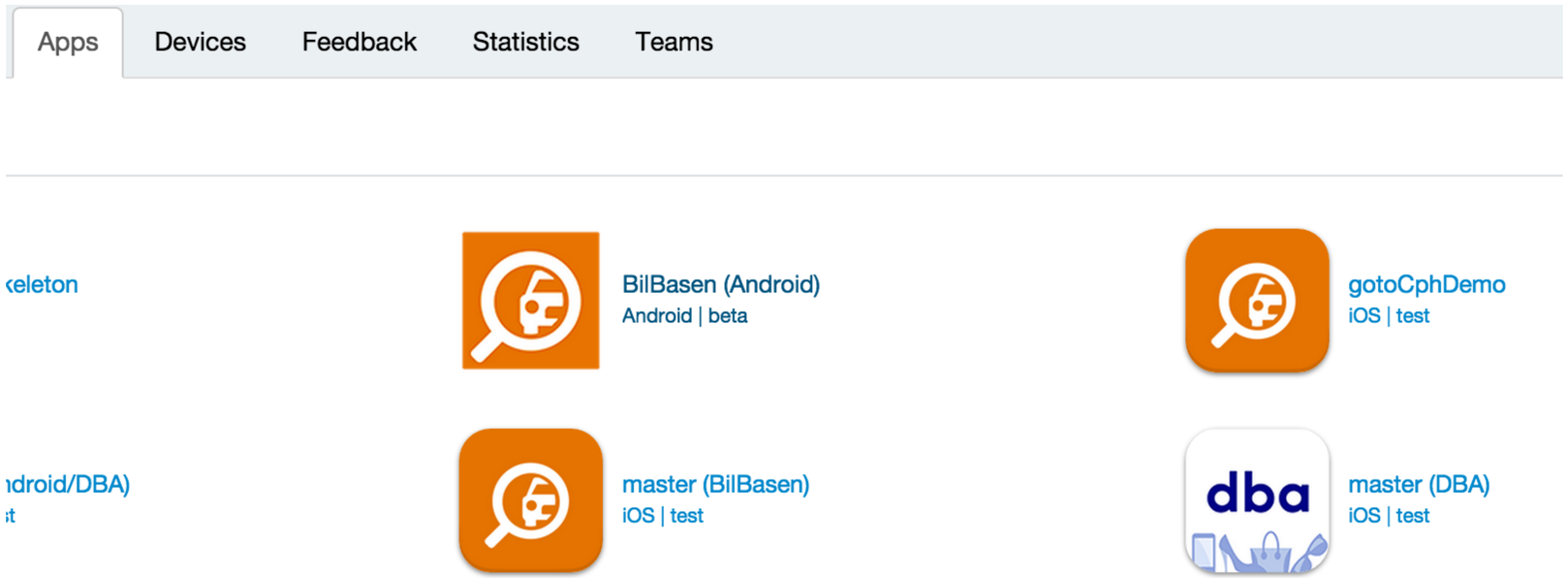
Jenkins jobs pr. branch - configuration

- Plugin: http://entagen.github.io/jenkins-build-per-branch/



Hockey app distribution pr. branch

- A new app is created for each feature branch
  - Easy manuel testing on devices
  - UX verification
- Deleted again together with the branch
  - Hockey app does not get cluttered



Tests

Prerequisites

- Required
  - Stable test environment(s)
  - Good test data

Prerequisites

- Required
  - Stable test environment(s)
  - Good test data

- Optional (but a very good idea)
  - Ways to setup data to a given state
    - E.g trigger something normally done in a batch job

Different kind of tests

Different kind of tests

Unit tests

Like all other platforms - super fast, and an easy way to tests all objects in isolation

Different kind of tests

Unit tests

Like all other platforms - super fast, and an easy way to tests all objects in isolation

Integration tests

Test of backend api´s from the app code - verifying json, mapping to the model, error codes etc.

Different kind of tests

Unit tests

Like all other platforms - super fast, and an easy way to tests all objects in isolation

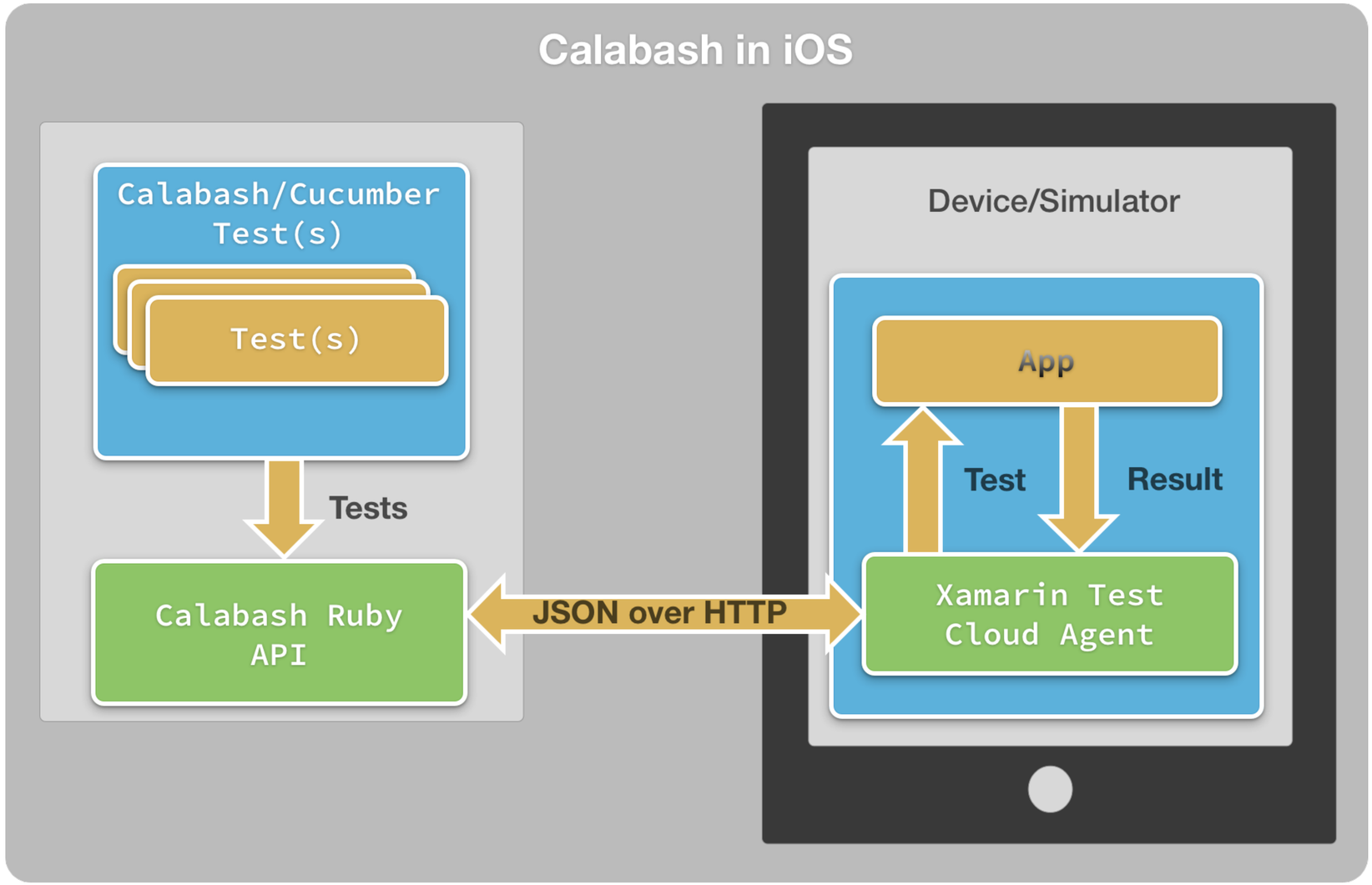
Integration tests

Test of backend api´s from the app code - verifying json, mapping to the model, error codes etc.

UI tests

Cucumber / Calabash - user scenarios to verify app from end-to-end

Cucumber / Calabash

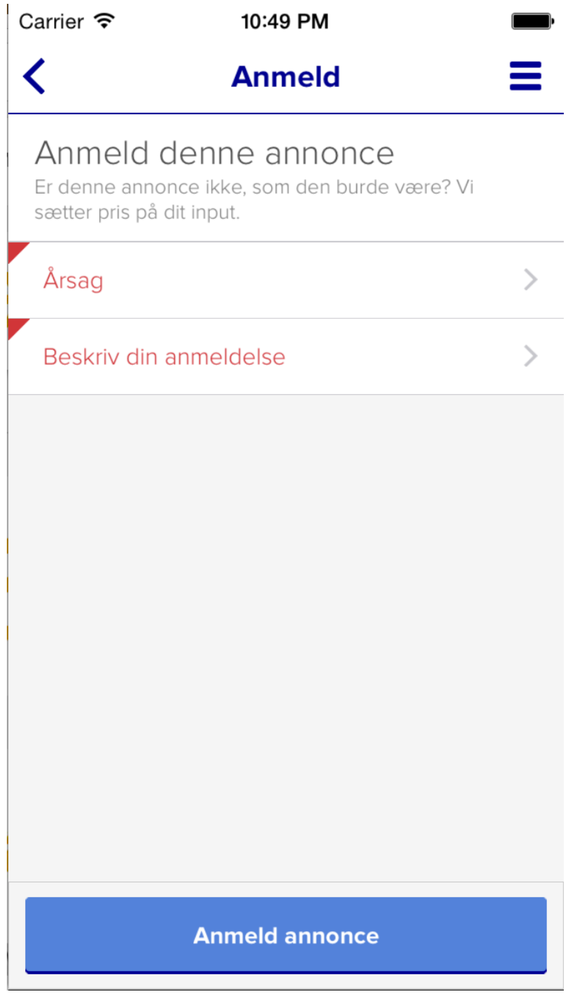


Example

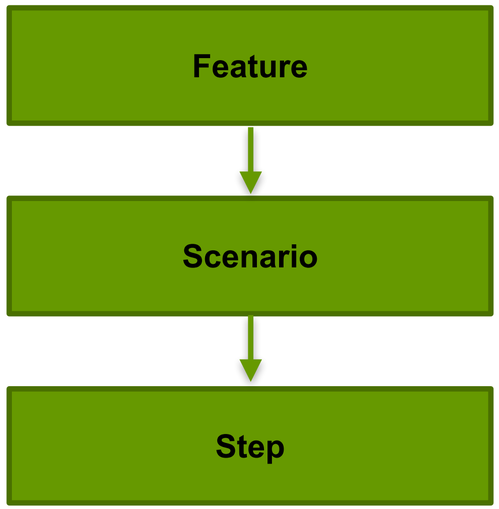
Scenario: I can only send a valid report of a listing once
Given I am logged in as "UniqueSeller" using quick login
And I am on the VIP for "iPhone"
When I go to report listing
And I try to send the report
Then I see the validation error for "Årsag, Beskriv din anmeldelse"

When I close the system message view
And I select report listing cause "Annoncen er ulovlig"
And I set report description text to "Den er billigere end min!"
And I send the report
Then I see the report listing VIP

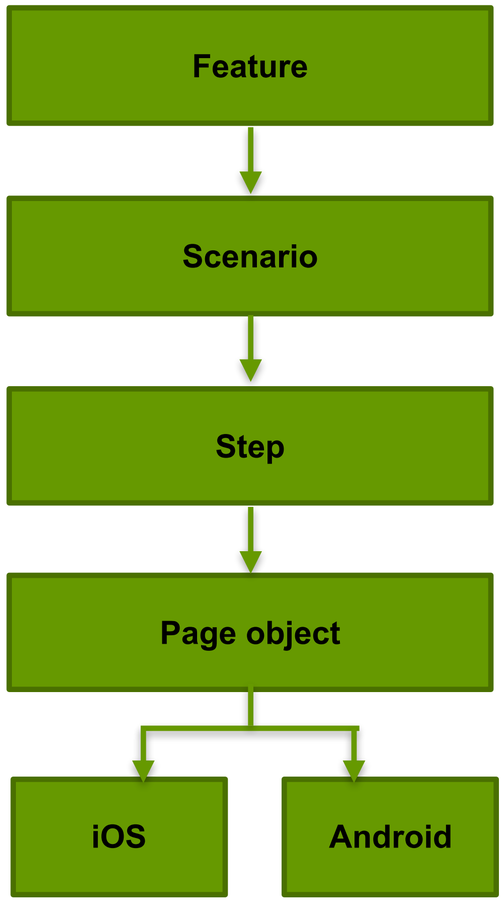
When I close the system message view
And I touch the report listing button
Then I am informed that I already has reported the listing



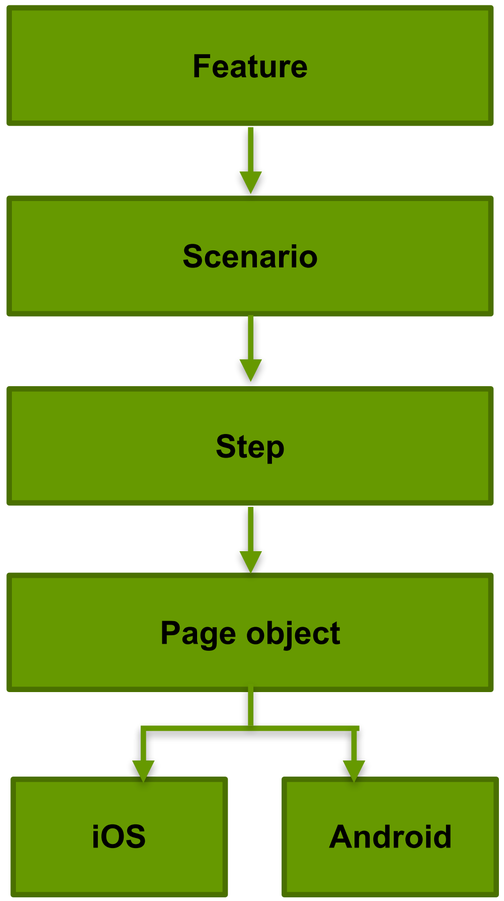
X-platform UI tests



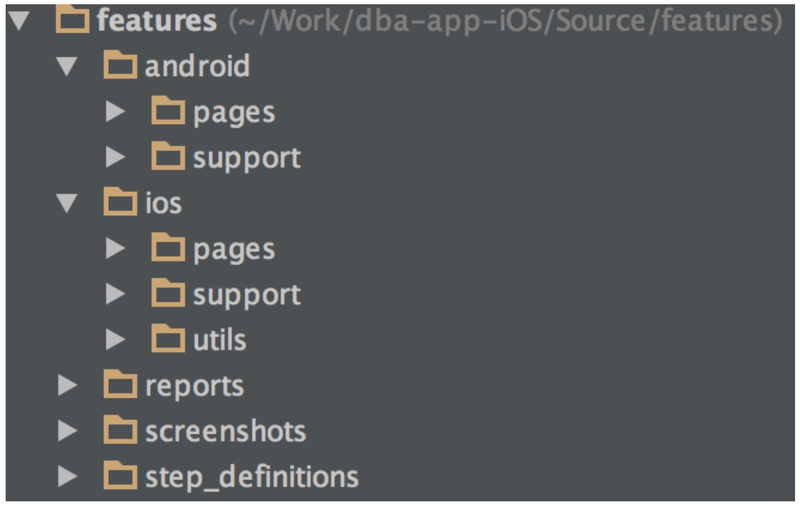
X-platform UI tests



X-platform UI tests



- Same scenarios / Steps
- Page objects are specific to the platform
  - Use -r in Cucumber to load for each platform
- features-folder is a Git SubModule



Structure your test code with Page objects

Step definition

```
When(/^I send the report$/) do
  @page = @page.send_the_report
end
```

Structure your test code with Page objects

Step definition

```
When(/^I send the report$/) do
  @page = @page.send_the_report
end
```

ReportListingpage - Page object (iOS)

```
def send_the_report
  transition(:tap => "view marked:'Anmeld annonce'", :page => self.previous_page)
end
```

Avoid timing-issues

Avoid timing-issues

Don’t

```
And /^I set price to "(.*?)"$/ do |price|
  macro 'I swipe up'
  sleep(1)
  touch("view marked:'Price'")
  sleep(0.5)
  set_text("view marked:'Price'", price)
  sleep(0.5)
  touch("view marked:'OK'")
  sleep(1)
end
```

Avoid timing-issues

Don’t

```
And /^I set price to "(.*?)"$/ do |price|
  macro 'I swipe up'
  sleep(1)
  touch("view marked:'Price'")
  sleep(0.5)
  set_text("view marked:'Price'", price)
  sleep(0.5)
  touch("view marked:'OK'")
  sleep(1)
end
```

Do

```
And /^I set price to "(.*?)"$/ do |price|
  @page.write_price(price)
end
```

page object

```
def write_price(price)
  scroll_and_wait_for_row_with_mark("priceCell")
  touch("view marked:'Price'")
  keyboard_enter_text price
  close_keyboard
end
```

Re-run failed tests

- Underlying bits and pieces can be unstable
  - UIAutomation
  - Simulator
- Cucumber rerun formatter
  - Give failed test one more try

Re-run failed tests

- Underlying bits and pieces can be unstable
  - UIAutomation
  - Simulator
- Cucumber rerun formatter
  - Give failed test one more try

```
cucumber -f rerun --out rerun.txt

cucumber @rerun.txt
```

Fast feedback

- Only run part of UI test suite on each commit
  - use @tags
- Use “backdoor” to setup/teardown
  - E.g. login or create a new user

Fast feedback

Fast feedback

@commit
Scenario: I can manage my listing and see the changes
  Given I am logged in as "UniqueSeller" using quick login
  And I have created a listing for "Hovedtelefoner" and is on the SYI VIP
  Then I see the VIP for "Her kommer en rimelig lang tekst"

  When I go back to my listings page
  And the created listing is in the list
      …more steps omitted…

Fast feedback

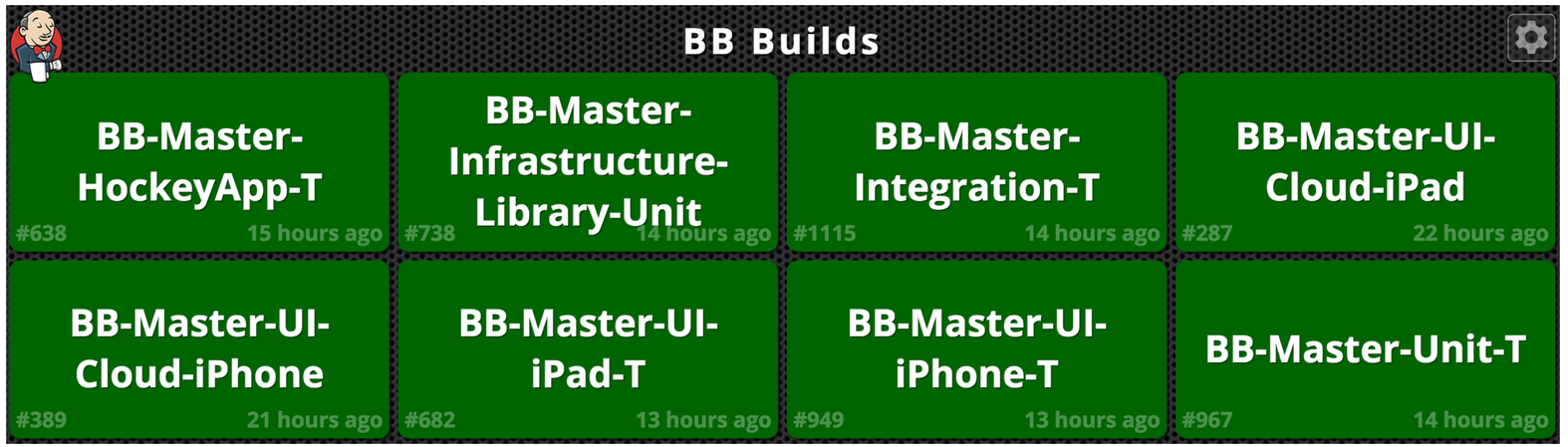
@commit
Scenario: I can manage my listing and see the changes
  Given I am logged in as "UniqueSeller" using quick login
  And I have created a listing for "Hovedtelefoner" and is on the SYI VIP
  Then I see the VIP for "Her kommer en rimelig lang tekst"

  When I go back to my listings page
  And the created listing is in the list
      …more steps omitted…

```
1 scenario (1 passed)
19 steps (19 passed)
0m34.054s
```

```
1 scenario (1 passed)
19 steps (19 passed)
1m16.610s
```

Visible feedback



Visible feedback

jenkins BOT 11:13 AM

DBA-Master-UI-iPhone-T - #1024 Back to normal after 30 min (Open)
Test Status:
Passed: 21, Failed: 0, Skipped: 0

jenkins BOT 11:29 AM

DBA-Master-UI-iPad-T - #726 Back to normal after 46 min (Open)
Test Status:
Passed: 21, Failed: 0, Skipped: 0

jenkins BOT 12:13 PM

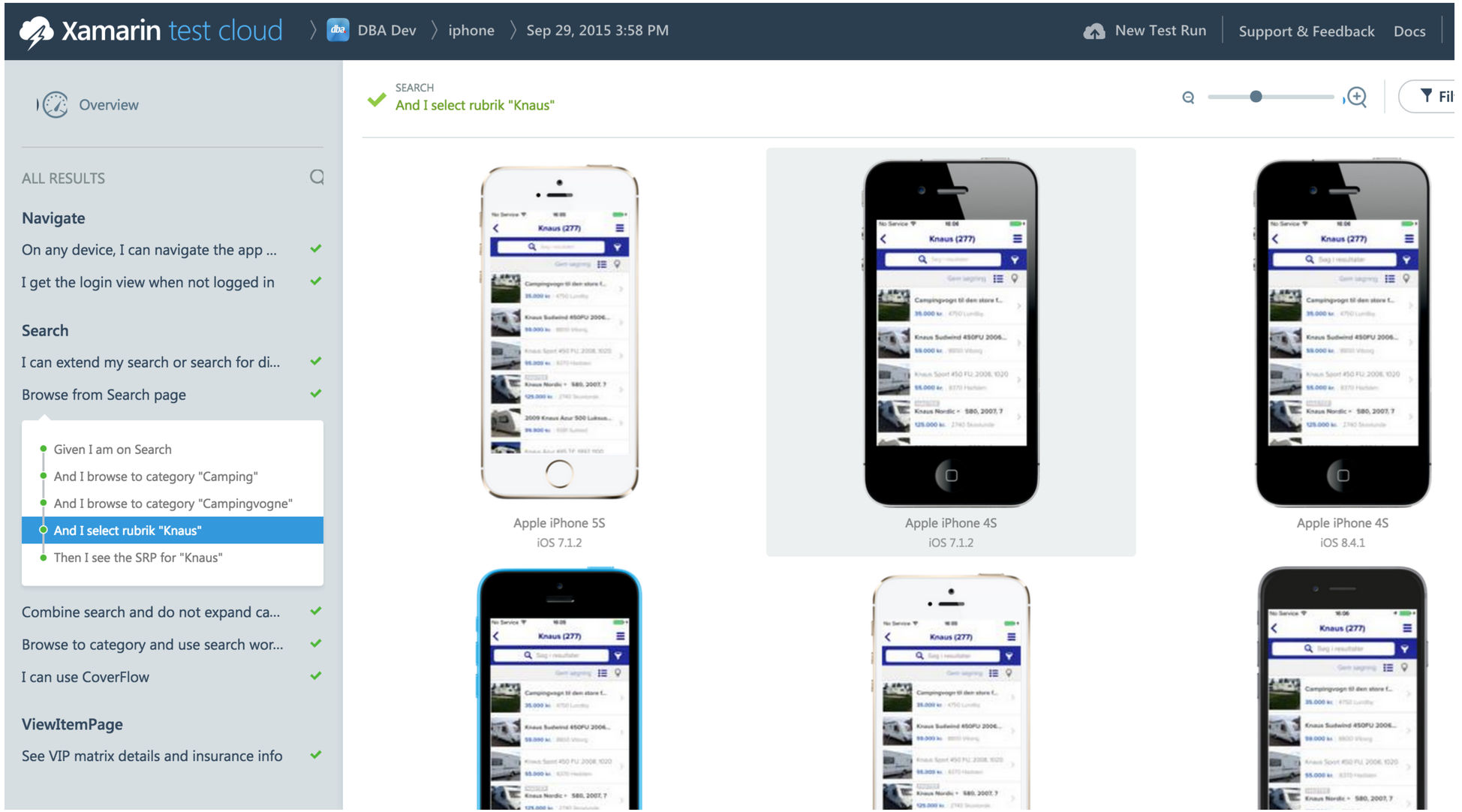
DBA-Master-UI-All-iPad-T - #484 Back to normal after 1 hr 29 min (Open)
Test Status:
Passed: 78, Failed: 0, Skipped: 0

jenkins BOT 12:50 PM

DBA-Master-UI-Cloud-iPad - #263 Failure after 0.14 sec (Open)
No Tests found.

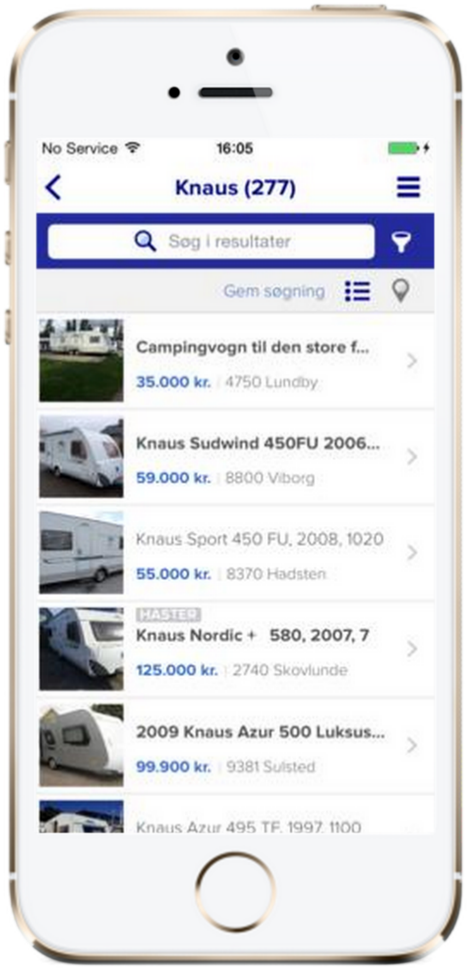
Test on real devices

Test on real devices



Test on real devices

Console Log



OS
iOS 8.4.1

Screen size
4 in (10.2 cm)

Model
iPhone6,2

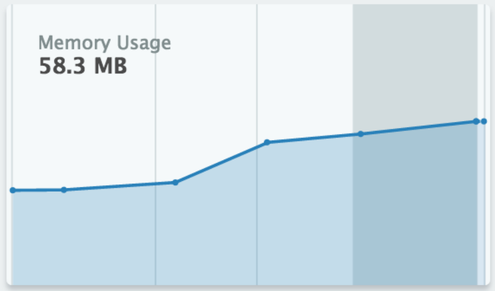
Resolution
640 x 1136 (326 ppi)

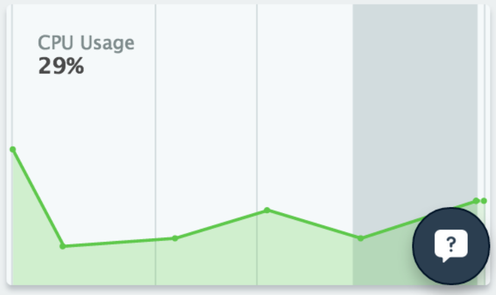
Release date
September 2013

Full Size | Device Log | Test Log

Test duration:
275.22 sec

Step duration:
6.58 sec





Is test automation all you need?

Is test automation all you need?

No, but they are

- Fast(er than humans)
- They never forget details (that you taught it)
- Great way to avoid “old” bugs (regression)

Is test automation all you need?

No, but they are

- Fast(er than humans)
- They never forget details (that you taught it)
- Great way to avoid “old” bugs (regression)

iOS 9 hack lets anyone see all photos and contacts even if ...
www.independent.co.uk/.../gad**gets**.../**ios**-9-hack-lets-... ▾ Oversæt denne side
for 6 dage siden - An **exploit** using Siri allows people to **get** around the iPhone's security ... on the passcode **screen** until the iPhone threatens to **lock** the user out.

iOS 7 Lock Screen Vulnerability Discovered Which Gives ...
techcrunch.com/.../**ios**-7-**lock-screen-vulnerability**-dis... ▾ Oversæt denne side
19. sep. 2013 - **Vulnerabilities** in Apple's **iOS lock screens** have become a fixture of ... (ahem) photos **get** stolen or shared out through your social accounts.

Manuel tests

- Still very important
- Part of Definition of Done for each story
- QA has more time to do
  - Exploratory tests
  - Focus on highest risk

Trust

- All team members write/maintaining tests
- QA trusts the automated tests
- Developers trusts QA to find the “tricky” bugs
- Everyone trusts that a red test means something is wrong
  - No broken windows
- Code coverage

Trust

- All team members write/maintaining tests
- QA trusts the automated tests
- Developers trusts QA to find the “tricky” bugs
- Everyone trusts that a red test means something is wrong
  - No broken windows
- Code coverage

Name	Packages	Files	Classes	Lines	Conditionals
Cobertura Coverage Report	97% 34/35	74% 182/245	74% 182/245	61% 7172/11825	67% 1119/1677

Coverage Breakdown by Package

Name	Files	Classes	Lines	Conditionals
[redacted]	0% 0/1	0% 0/1	0% 0/12	N/A
[redacted]	71% 5/7	71% 5/7	64% 197/306	59% 51/86
[redacted]	93% 14/15	93% 14/15	66% 414/631	63% 87/138
[redacted]	100% 2/2	100% 2/2	78% 36/46	N/A
[redacted]	100% 27/27	100% 27/27	73% 877/1203	74% 238/322
[redacted]	57% 8/14	57% 8/14	70% 364/520	69% 82/119
[redacted]	73% 8/11	73% 8/11	65% 187/286	50% 5/10
[redacted]	80% 4/5	80% 4/5	39% 52/135	100% 2/2
[redacted]	100% 2/2	100% 2/2	100% 22/22	75% 6/8
[redacted]	75% 3/4	75% 3/4	59% 126/214	N/A
[redacted]	57% 8/14	57% 8/14	75% 132/177	N/A
[redacted]	60% 6/10	60% 6/10	33% 94/282	50% 1/2
[redacted]	33% 4/12	33% 4/12	50% 336/672	55% 32/58
[redacted]	100% 1/1	100% 1/1	13% 9/70	N/A
[redacted]	50% 1/2	50% 1/2	44% 24/55	N/A
[redacted]	55% 6/11	55% 6/11	52% 203/389	50% 18/36
[redacted]	100% 11/11	100% 11/11	76% 789/1040	65% 141/218
[redacted]	33% 2/6	33% 2/6	34% 141/420	66% 27/41
[redacted]	100% 1/1	100% 1/1	84% 74/88	60% 15/25

Plan smaller and frequent releases

- Convince Product Owner of the agility
  - Automated app updates = fast adoption rate
- Define release scope with Product Owner

Plan smaller and frequent releases

- Convince Product Owner of the agility
  - Automated app updates = fast adoption rate
- Define release scope with Product Owner

Plan smaller and frequent releases

- Convince Product Owner of the agility
  – Automated app updates = fast adoption rate
- Define release scope with Product Owner

Prepare your app for it

Plan smaller and frequent releases

- Convince Product Owner of the agility
  - Automated app updates = fast adoption rate
- Define release scope with Product Owner

Prepare your app for it

- “Kill switch” to force users to update eventually
  - Work as an emergency brake as well
- Welcome screens
  - Highlight whats new
  - Teach the users how to use new features
  - Do not expect users to read “Whats new” in the AppStore

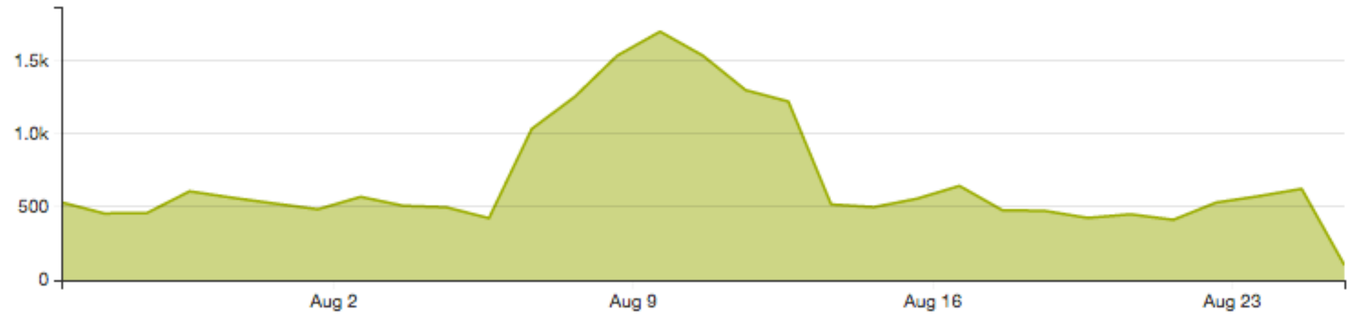
Relax - no worries!

Embrace failures

- Bug free software is an illusion
- Learn/improve from bugs/crashes found by users
  – Add new tests to prevent it from happening again
- Celebrate (potential) bugs found by tests before production

Know your app quality

- Visual information about app health
- Be proactive and fix bugs before too many users notice them



goto; conference

Please

Remember to rate this session

Thank you!

Join the conversation #gotocph