

A close-up, artistic photograph of a human eye, looking slightly to the right. The eye is the central focus, with its iris and eyelashes clearly visible. The background is a soft, out-of-focus light blue.

# Quality assurance for mobile applications

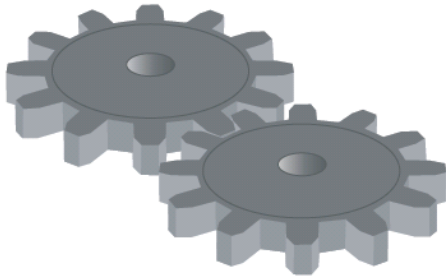
*Case studies for GUI test automation*

*Alexandra Schladebeck*

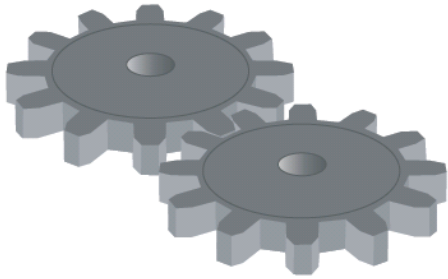
# Agenda

- ▶ **The history**
- ▶ **The new questions**
- ▶ **The candidates**
- ▶ **Our experiences**
- ▶ **Results and conclusion**

# The (hi)story



# The (hi)story



## The aims

- ▶ **Evaluate feasibility of mobile GUI test automation**  
Functional and technical perspective
- ▶ **Find out how continuous integration and testing can be implemented**
- ▶ **Analyse cross-platform testing in a mobile context**  
Write once, run anywhere
- ▶ **Make informed decisions on mobile testing strategy**

# Candidate 1: Customer application

- ▶ **Datawarehouse for competition analysis**

Enterprise desktop application

- ▶ **→ Go mobile**

Conference catalog that uses data

Extra comfort and features (bookmarking, quick navigation)

Target group = managers at automobile congresses

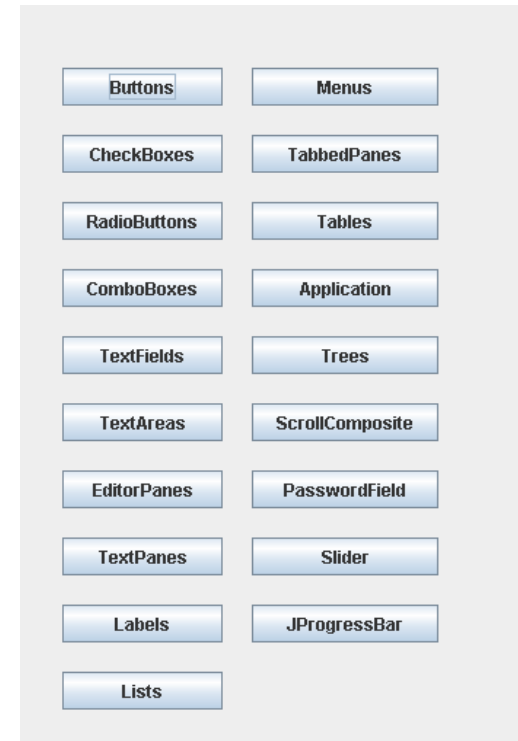
- ▶ **iOS only**

Version 5 → 6 → 7

- ▶ **Release date \*must\* be kept**

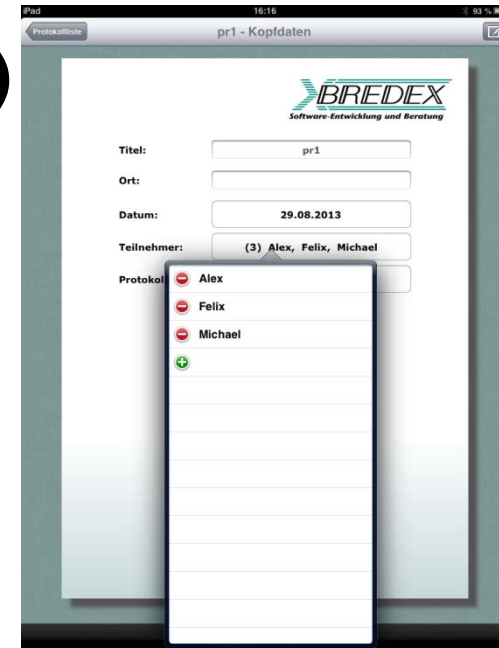
# Candidate 2: Continuous Integration

- ▶ **We test our test tool with our test tool...**
- ▶ **Testing support for widgets requires example applications**
  - Android, iOS, Windows 8 (modernUI)
  - Perform actions and checks
- ▶ **Functionally uninteresting**
- ▶ **Candidates for evaluating continuous integration**



# Candidate 3: In-house application

- ▶ **In-house application**  
Completely new development
- ▶ **For taking minutes in meetings**
- ▶ **Cross-platform development (Xamarin)**
- ▶ **Cross-platform automated testing**





# Experiences : Customer (iOS) project

## ▶ **Process**

- Manual test scripts specified as automated tests
- Automated test started manually
- Testable version of the app provided manually
- New productive data available very late

## ▶ **Functional aspects**

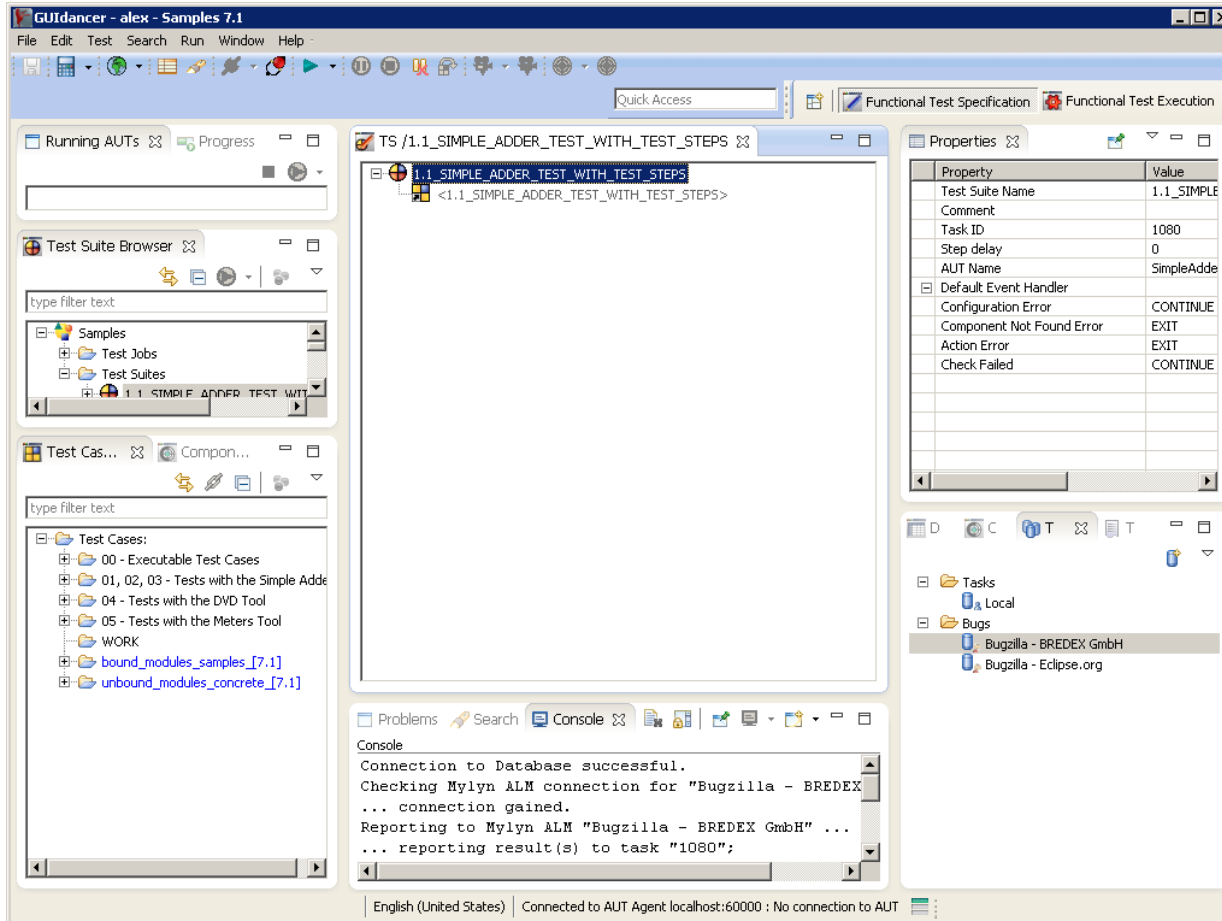
- Required features could be tested with automated test
- Regression testing for new versions (iOS 6)

## ▶ **Technical aspects**

- Some modifications necessary due to custom components



# Demo



The screenshot shows the Eclipse IDE with the GUI Tester plugin. The main window is titled "GUIDancer - alex - Samples 7.1". The interface includes several panels:

- Running AUTs:** Shows the progress of the test execution.
- Test Suite Browser:** Displays a tree view of test suites, with "1.1\_SIMPLE\_ADDER\_TEST\_WITH\_TEST\_STEPS" selected.
- Test Cases:** Shows a list of test cases, including "00 - Executable Test Cases", "01, 02, 03 - Tests with the Simple Adde", "04 - Tests with the DVD Tool", "05 - Tests with the Meters Tool", "WORK", "bound\_modules\_samples\_[7.1]", and "unbound\_modules\_concrete\_[7.1]".
- Properties:** A table showing the properties of the selected test suite.
- Console:** Displays the output of the test execution, including messages about database connection and reporting to Mylyn ALM.

The console output shows the following messages:

```

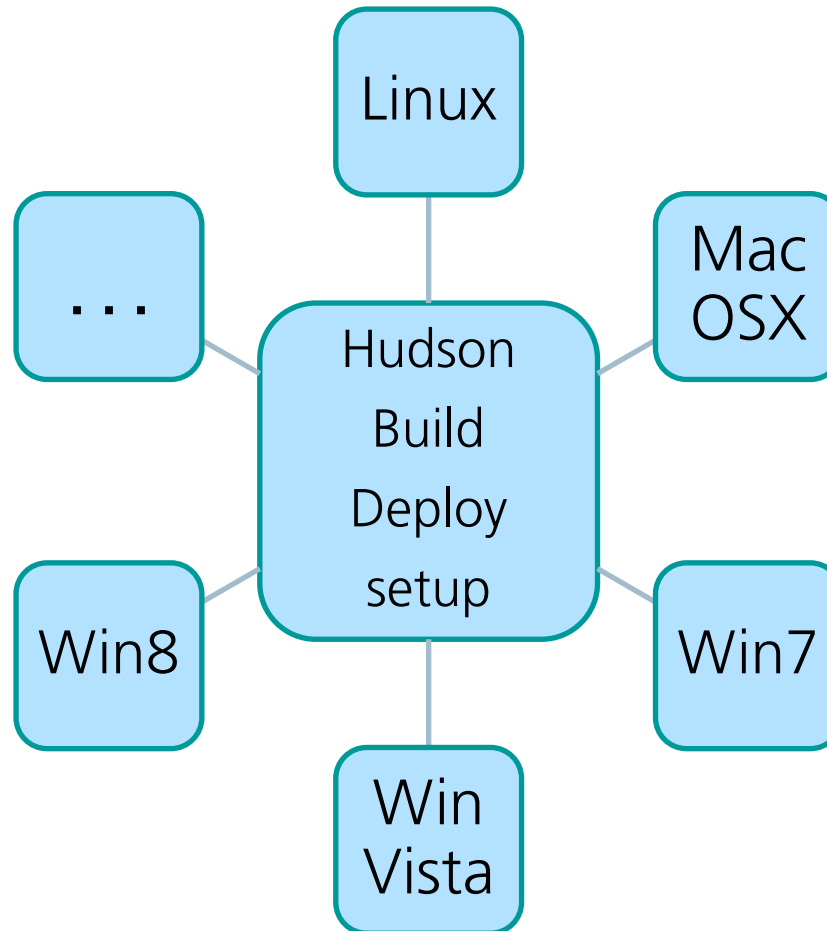
Connection to Database successful.
Checking Mylyn ALM connection for "Bugzilla - BREDEX
... connection gained.
Reporting to Mylyn ALM "Bugzilla - BREDEX GmbH" ...
... reporting result(s) to task "1080";
  
```

The Properties panel shows the following table:

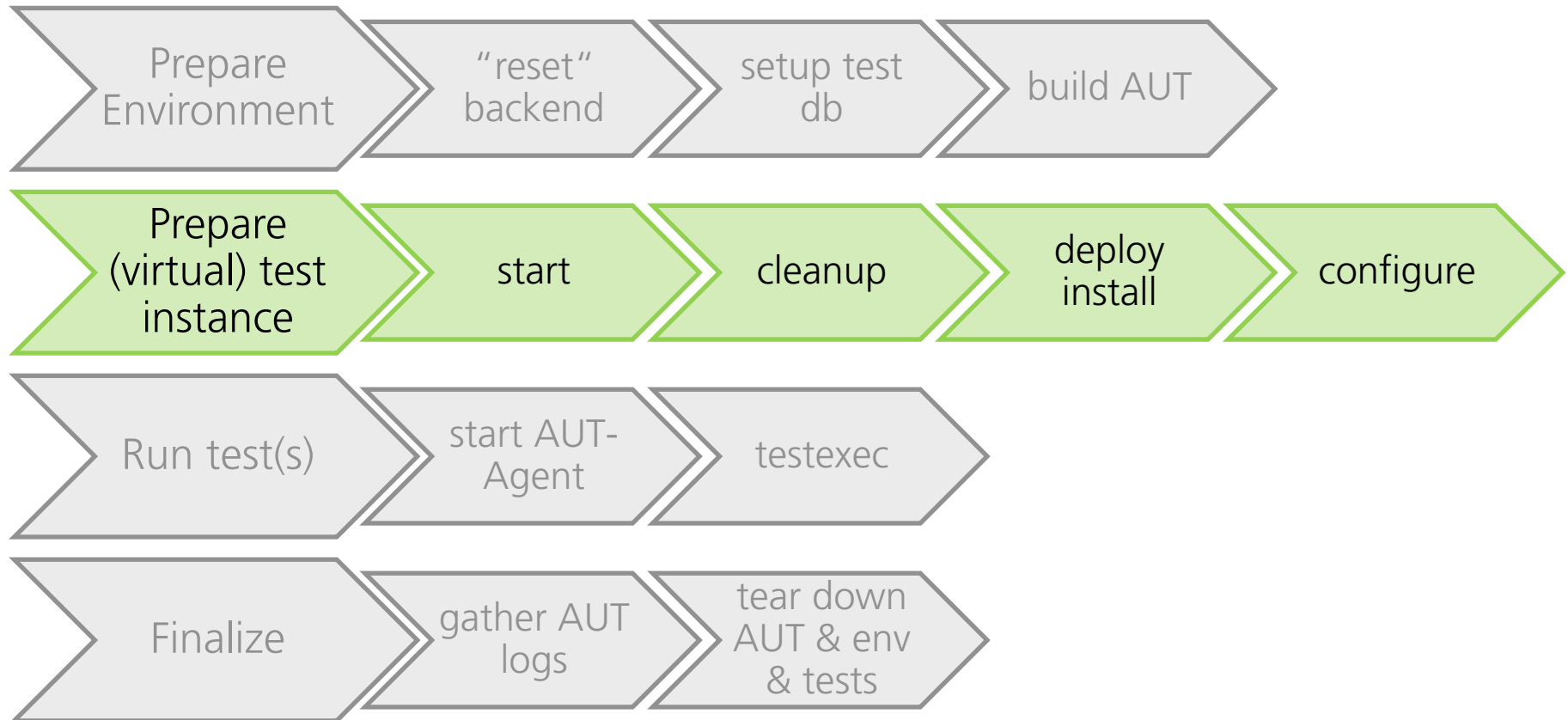
Property	Value
Test Suite Name	1.1_SIMPLE
Comment	
Task ID	1080
Step delay	0
AUT Name	SimpleAdde
Default Event Handler	
Configuration Error	CONTINUE
Component Not Found Error	EXIT
Action Error	EXIT
Check Failed	CONTINUE

The status bar at the bottom indicates: English (United States) | Connected to AUT Agent localhost:60000 : No connection to AUT

# Experiences: Continuous Integration



# Experiences: Continuous Integration



# Experiences : Continuous Integration

## ▶ iOS

Mac machine

XCode

Simulator / Device

Communication

Deploy application

(platform specific)

Start AUT



## ▶ Windows 8 (modernUI)

Windows 8 machine

Visual studio 2012 / 2013

Rights & signing

AUT automatically startable



# Experiences: Continuous Integration

- ▶ **Sandbox**
- ▶ **Platform-dependent scripts necessary**
- ▶ **Idiosyncrasies**
- ▶ **Simulators alone not enough**
- ▶ **Functional testing of**
  - Orientation changes
  - GPS
  - Battery
  - Internet connection
  - Access to external applications



# Test results

## Execution Stack

- **caa\_ios\_FULLTEST\_ipad\_portrait\_5.1 - TC (ok) - 0:13:16.360**
  - **caa\_ios\_FULLTEST\_ipad - TC (ok) - 0:13:16.360**
    - **FULLTEST - TC (ok) - 0:13:16.352**
      - **SPECIFIC - TC (ok) - 0:00:00.000**
      - **iOS.UIButton - TC (ok) - 0:01:43.906**
        - **iOS.AddContact - TC (ok) - 0:00:18.358**
          - **iOS - AddContact - Check Enablement - TC (ok) - 0:00:02.398**
          - **iOS - AddContact - Check Existence - TC (ok) - 0:00:02.487**
          - **iOS - AddContact - Check Selection - TC (ok) - 0:00:02.703**
          - **iOS - AddContact - Click - TC (ok) - 0:00:05.568**
          - **iOS - AddContact - Store Value - TC (ok) - 0:00:02.270**
          - **iOS - AddContact - Wait for Component - TC (ok) - 0:00:02.932**
        - **iOS.DetailDisclosure - TC (ok) - 0:00:17.951**
        - **iOS.InfoDark - TC (ok) - 0:00:17.943**
        - **iOS.InfoLight - TC (ok) - 0:00:17.657**
        - **iOS.RoundedRect - TC (ok) - 0:00:20.966**
        - **iOS.RoundedRectBackground - TC (ok) - 0:00:05.522**
        - **iOS.RoundedRectForeground - TC (ok) - 0:00:05.509**
      - **iOS.UISwitch - TC (ok) - 0:00:20.224**
      - **iOS.UIStepper - TC (ok) - 0:00:10.641**
      - **iOS.UISegmentedControl - TC (ok) - 0:00:43.425**
        - **iOS - UISegmentedControl - Check Enablement - TC (ok) - 0:00:01.922**
        - **iOS - UISegmentedControl - Check Existence - TC (ok) - 0:00:02.428**
        - **iOS - UISegmentedControl - Click - TC (ok) - 0:00:05.520**
        - **iOS - UISegmentedControl - Wait for Component - TC (ok) - 0:00:02.903**
        - **iOS - UISegmentedControl - Check Existence of Tab - TC (ok) - 0:00:02.764**
        - **iOS - UISegmentedControl - Check Enablement of Tab by Index - TC (ok) - 0:00:02.239**
        - **iOS - UISegmentedControl - Check Enablement of Tab by Value - TC (ok) - 0:00:02.307**
        - **iOS - UISegmentedControl - Check Existence of Tab - TC (ok) - 0:00:02.736**
        - **iOS - UISegmentedControl - Check Selection of Tab by Index - TC (ok) - 0:00:04.617**
        - **iOS - UISegmentedControl - Check Selection of Tab by Value - TC (ok) - 0:00:04.624**
        - **iOS - UISegmentedControl - Check Text of Tab by Index - TC (ok) - 0:00:02.204**
        - **iOS - UISegmentedControl - Select Tab by Index - TC (ok) - 0:00:04.513**
        - **iOS - UISegmentedControl - Select Tab by Value - TC (ok) - 0:00:04.648**
    - **iOS.UITextField - TC (ok) - 0:08:17.703**
      - **iOS - UITextField - Click - TC (ok) - 0:00:05.606**
      - **iOS - UITextField - Check Existence - TC (ok) - 0:00:01.912**

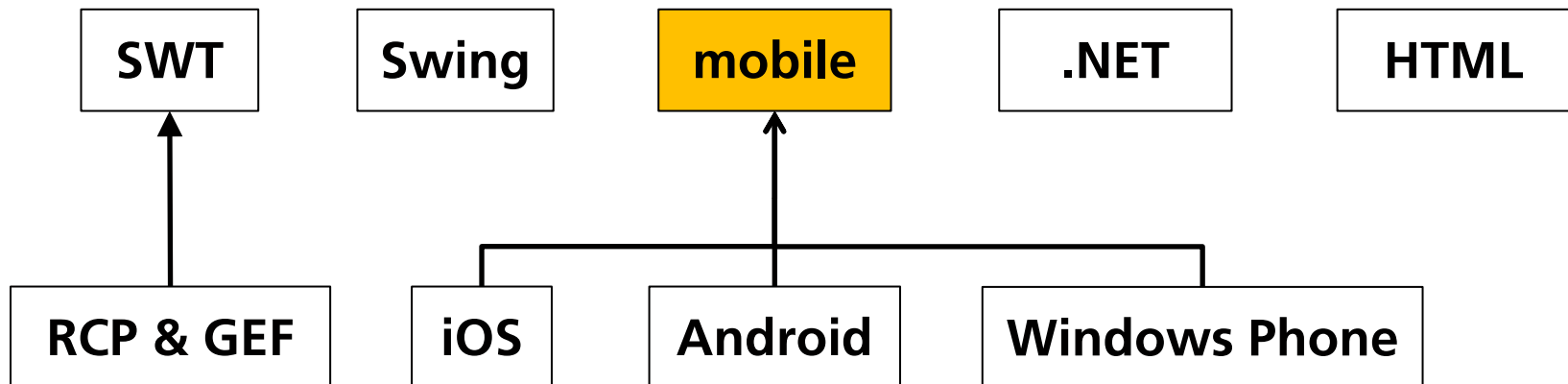
# Experiences: Cross-Platform Project

- ▶ **Cross-platform development**
- ▶ **Test : write once, run anywhere**

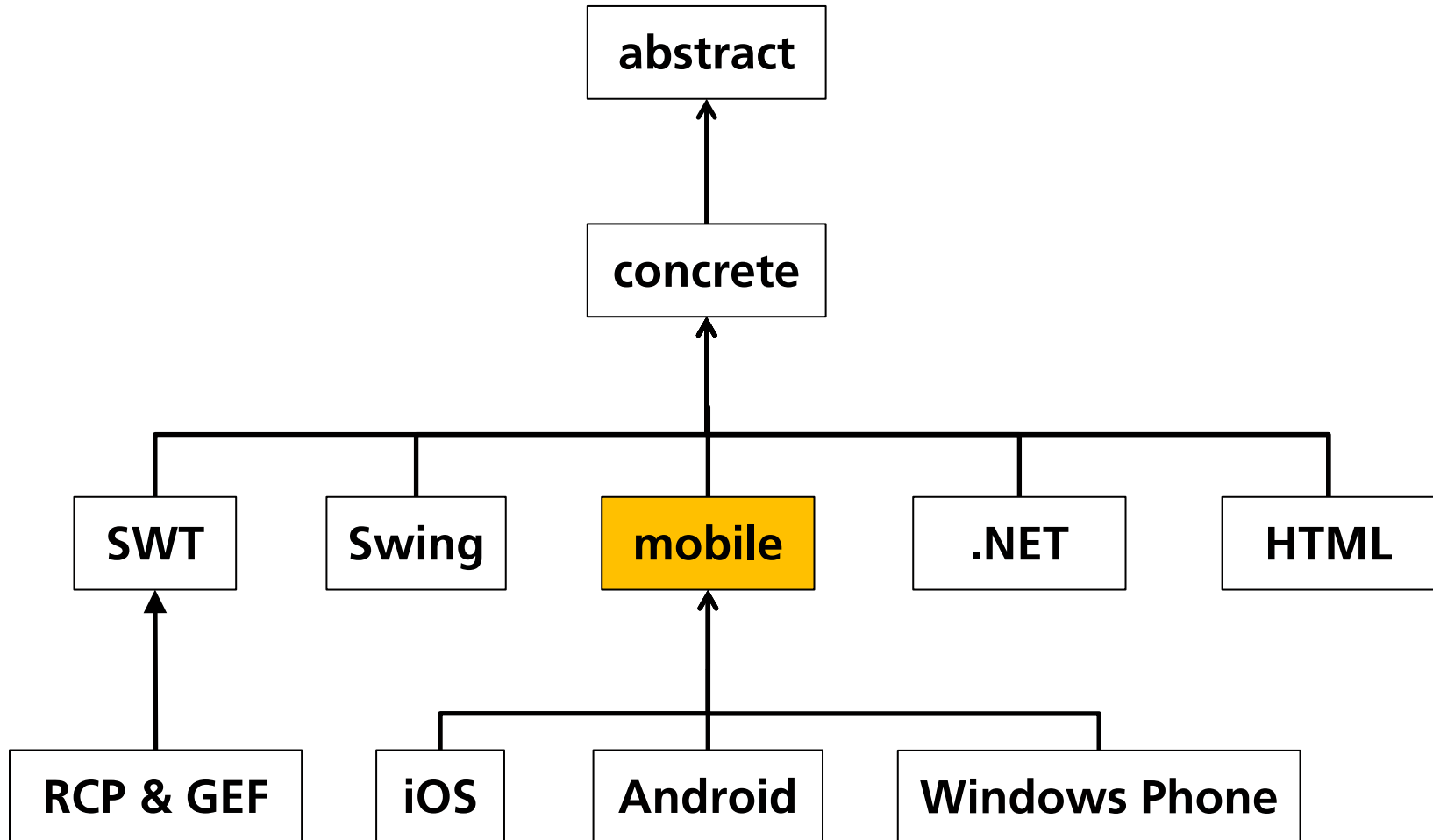




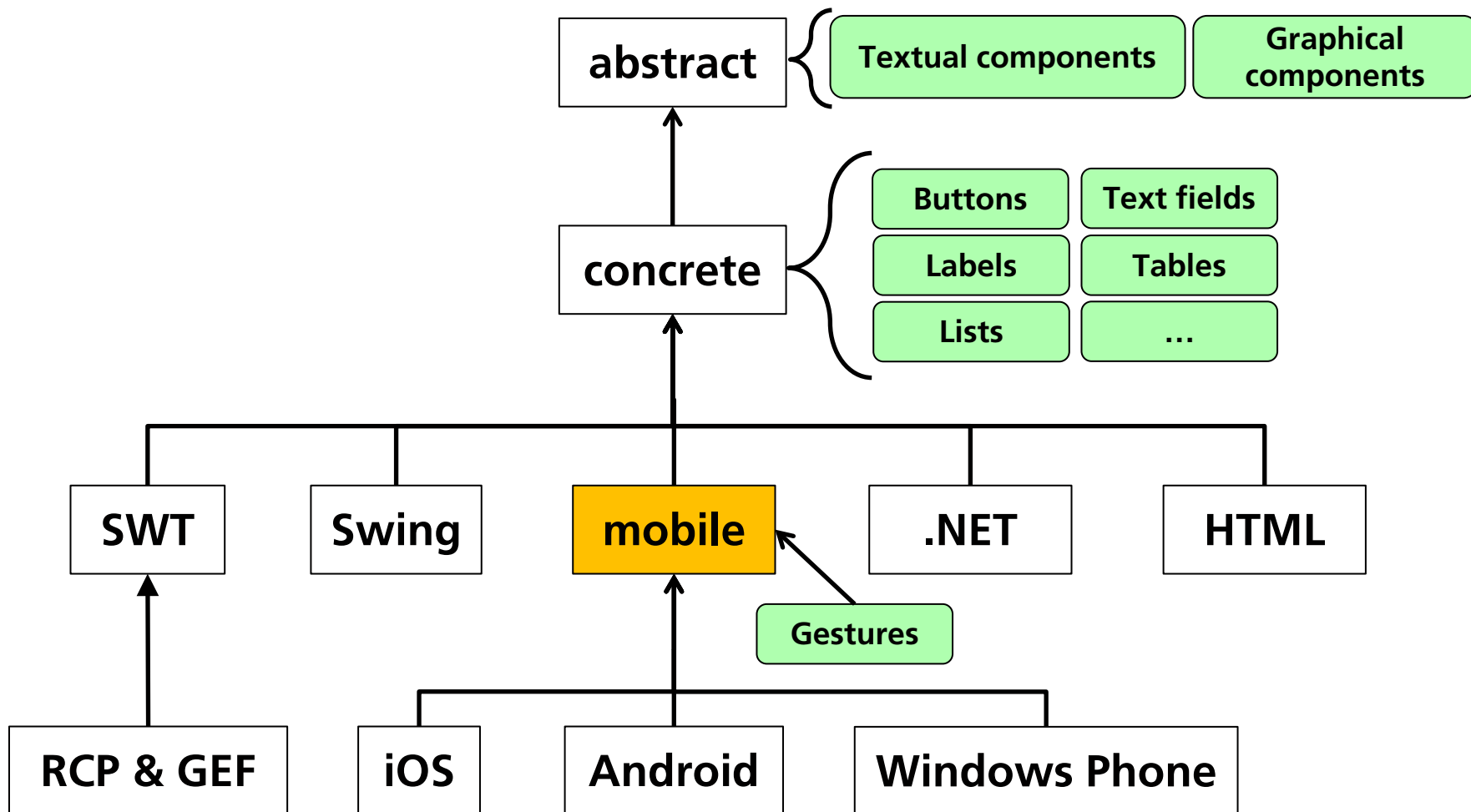
# Cross-Plattform



# Cross-Platform



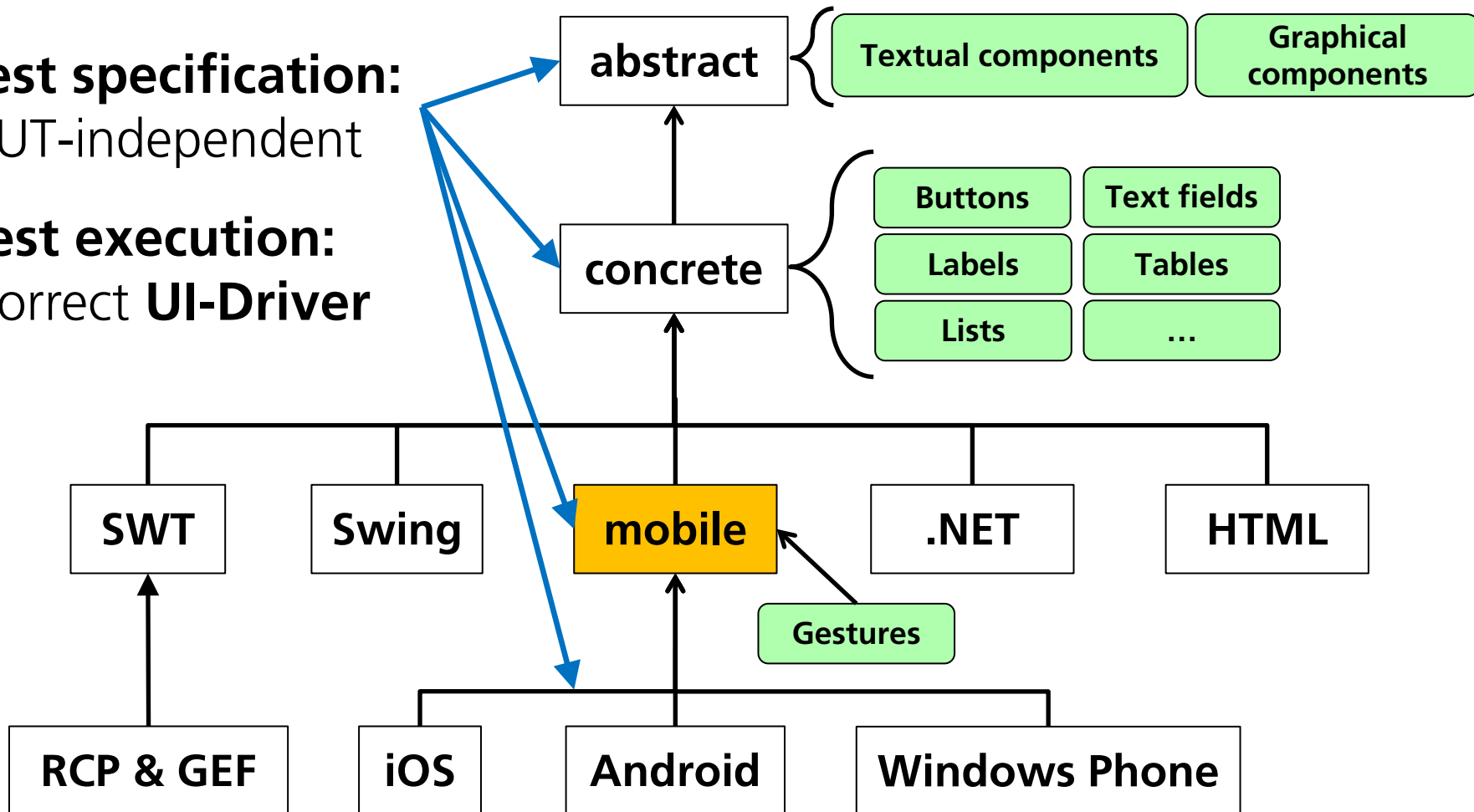
# Cross-Platform



# Cross-Platform

**Test specification:**  
AUT-independent

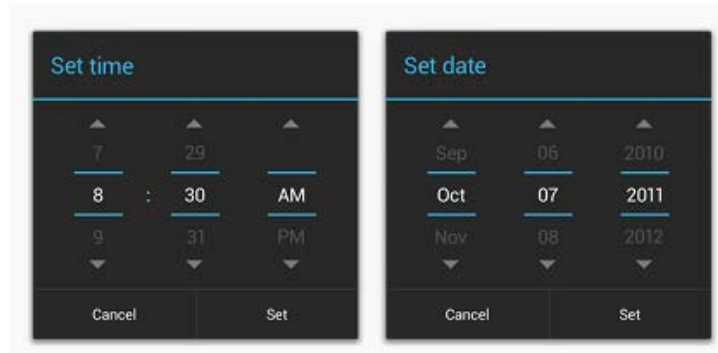
**Test execution:**  
Correct **UI-Driver**



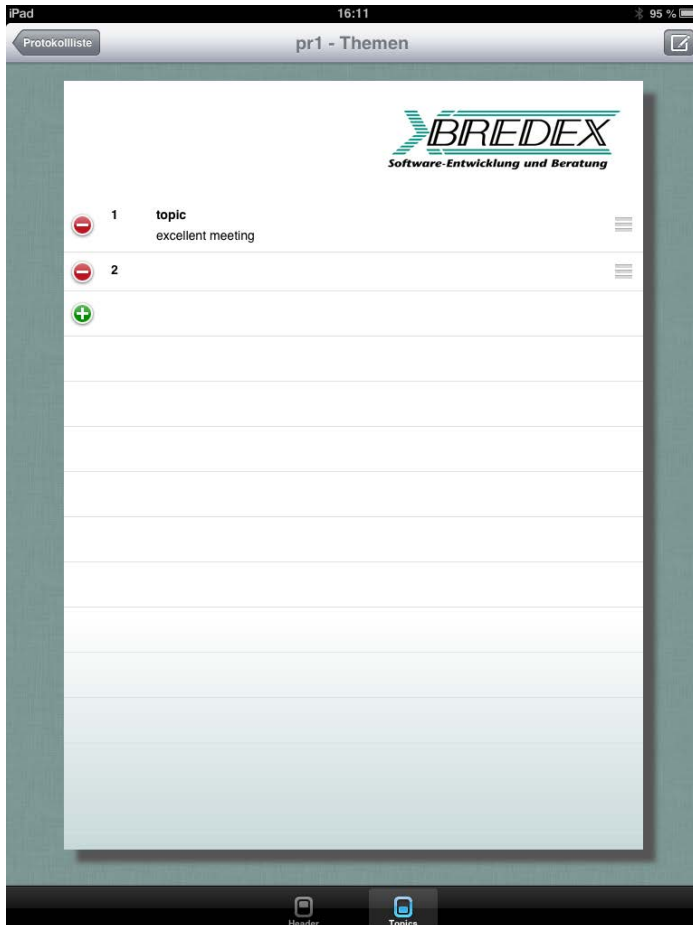
# The good news...

▶ **Different mobile UI toolkits use same/similar functional components**

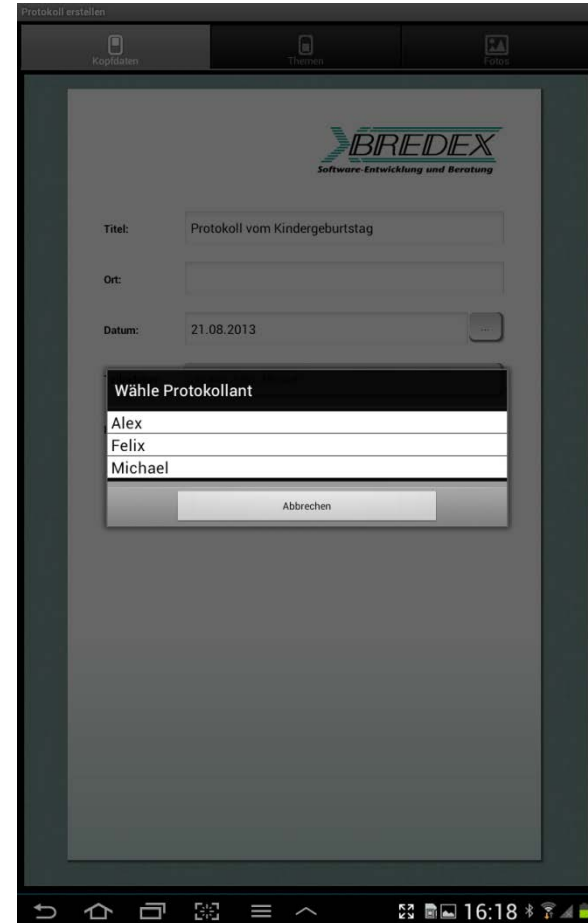
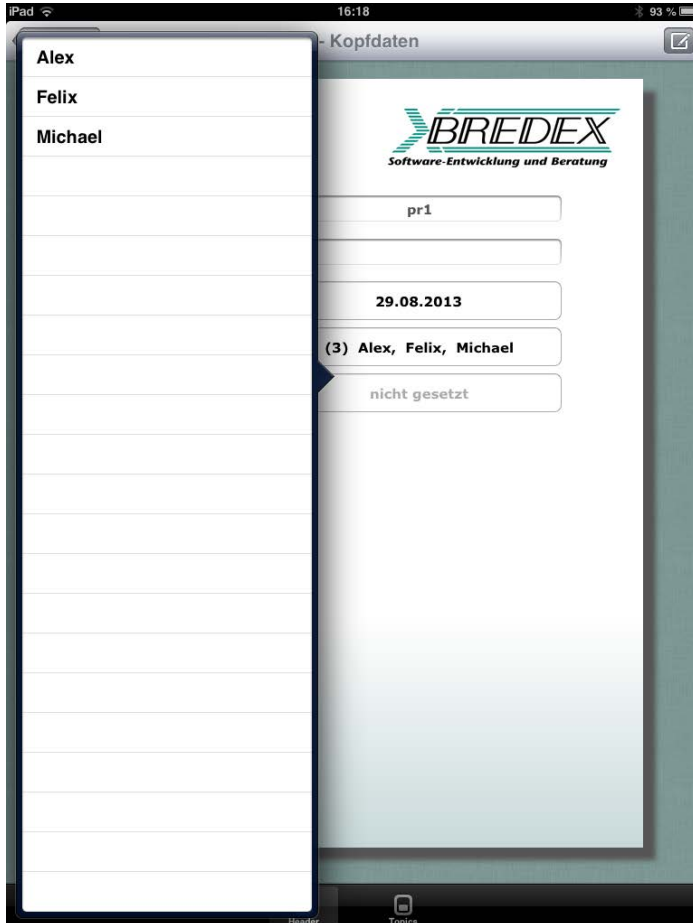
Buttons, textfields, lists, tabbed components, combo components



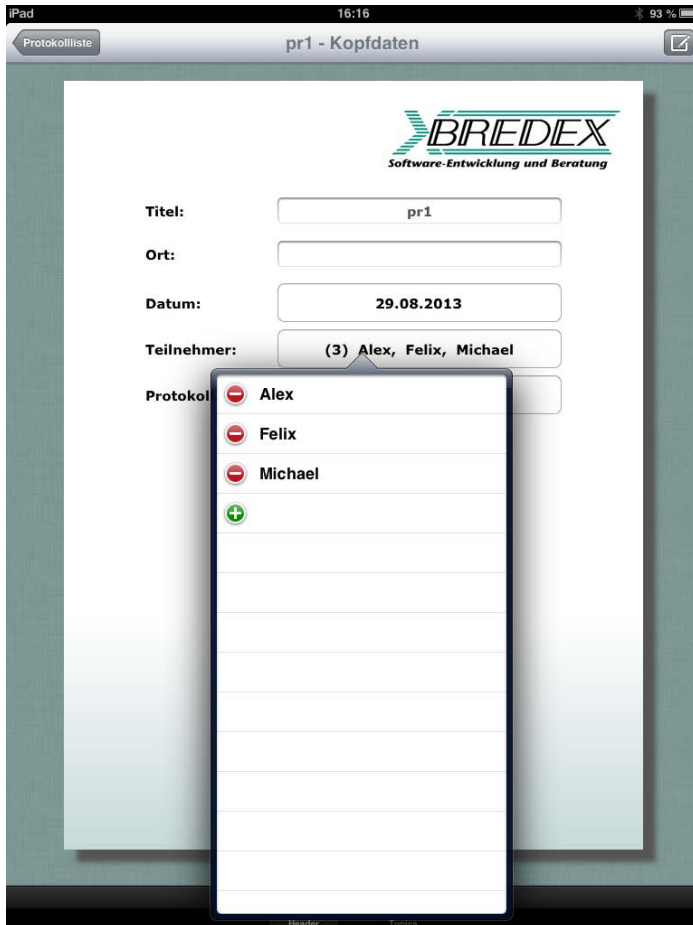
# Example 1: Edit list



# Example 2: Choose minute taker




# Example 3: Enter participants



iPad 16:16 93 %

Protokollliste pr1 - Kopfdaten



**Titel:**

**Ort:**

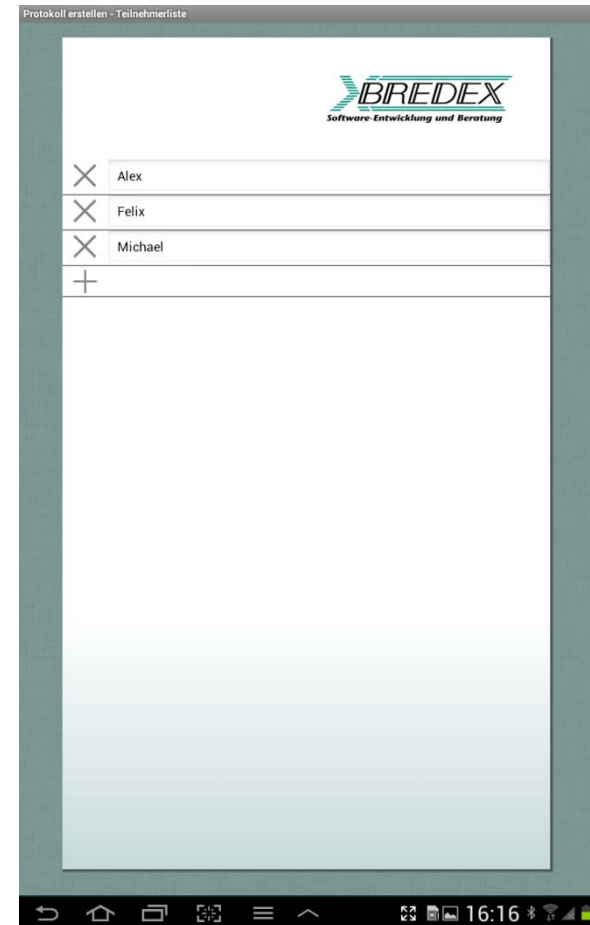
**Datum:**

**Teilnehmer:**


**Protokol**

- Alex
- Felix
- Michael
- 
- 
- 
- 
- 
- 
- 
- 

Header Topica



Protokoll erstellen - Teilnehmerliste



- Alex
- Felix
- Michael
- 

16:16



# Cross-platform differences are desired

- ▶ **Differences between desktop versions = minimal**
- ▶ **Differences between mobile clients = huge**
  - Features
  - Workflows
  - Use of hardware / software features
  - Users are used to / expect these differences.
- ▶ **Differences between versions often large**
  - iOS 6/7
  - Android 2/4

# Experiences: Cross-Platform Project

- ▶ **Cross-platform development**
- ▶ **Test : write once, run anywhere**



- ▶ **Technically possible**
- ▶ **Unlikely to be used...**

## Results: GUI test automation y/n?

### ▶ **Should we automate mobile tests?**

Multiplication factor over devices

New OS versions appear regularly

No choice or control over runtime environment

Quick feedback whether new version / feature is total fail

Short release cycles for own software

Changes happen that developers have no control over

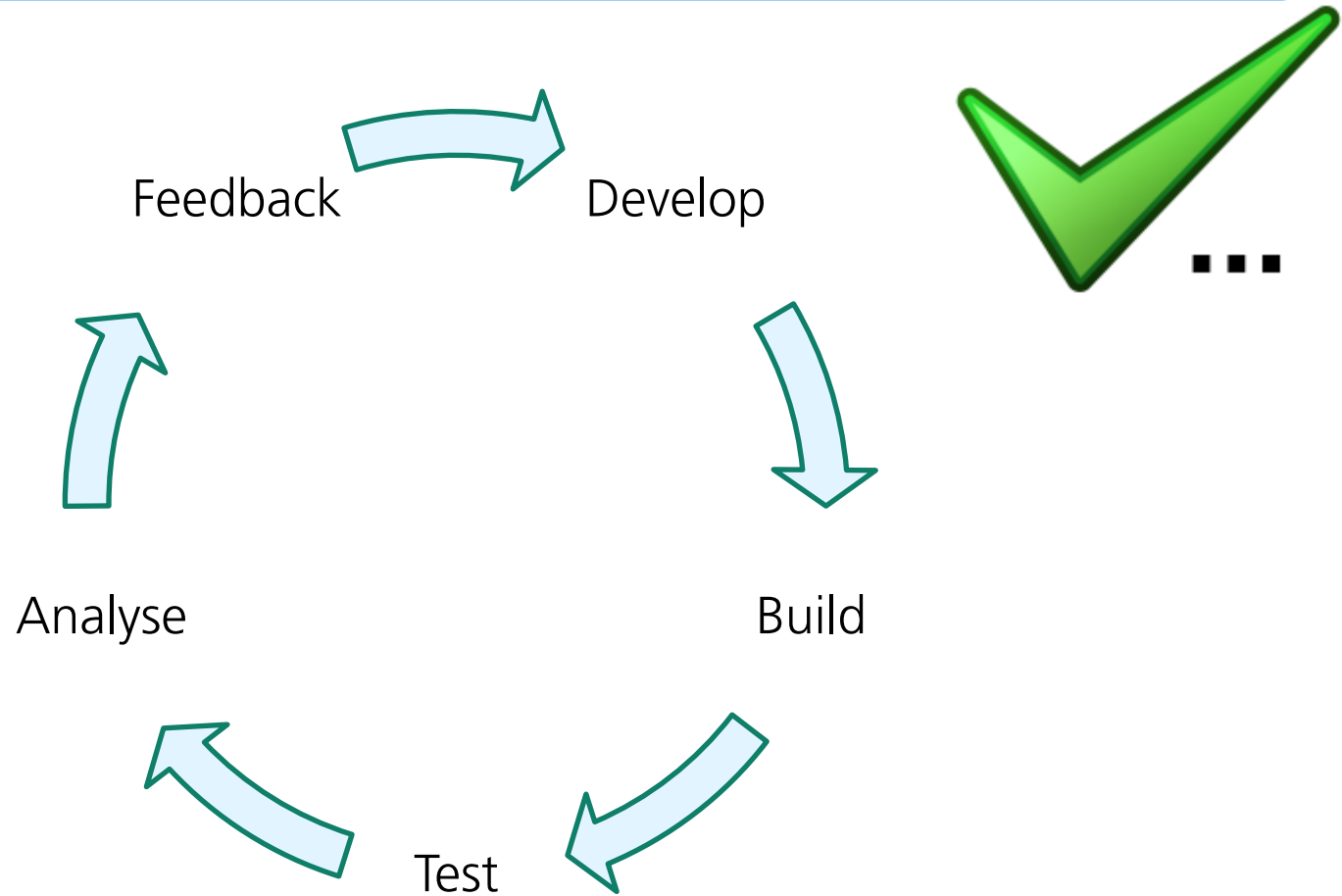


# Results: Writing automated tests for mobile

- ▶ **Apps are testable!**
- ▶ **Similar concepts, fewer components**
- ▶ **Extra hurdles**
  - More input from development team
  - Additional tests necessary (GPS, Battery, Internet)
  - Environment more complex



# Results: Continuous integration for mobile



## Results: Cross platform

- ▶ **Write once, run anywhere is possible**
- ▶ **But seems unlikely**
- ▶ **How much should / can development be influenced by test factors?**



# Results: Mobile test strategy



**Thank you!**

Alexandra Schladebeck

[alexandra.schladebeck@bredex.de](mailto:alexandra.schladebeck@bredex.de)

[@alex\\_schl](#)

<http://testing.bredex.de>

[blog.bredex.de](http://blog.bredex.de)

[www.bredex.de](http://www.bredex.de)

[www.guidancer.de](http://www.guidancer.de)

[www.eclipse.org/jubula](http://www.eclipse.org/jubula)