

# Rationalize Android development with StAnD

Clement Escoffier, akquinet A.G.



# What about me ?



- Head of the Innovation Competence Center at akquinet
- Apache Software Foundation
  - PMC Apache Felix, Apache Ace
  - Apache Felix iPOJO main developer
- OW2
  - Chameleon project leader
- A lot of others contributions
  - maven-android-plugin
  - H-UBU

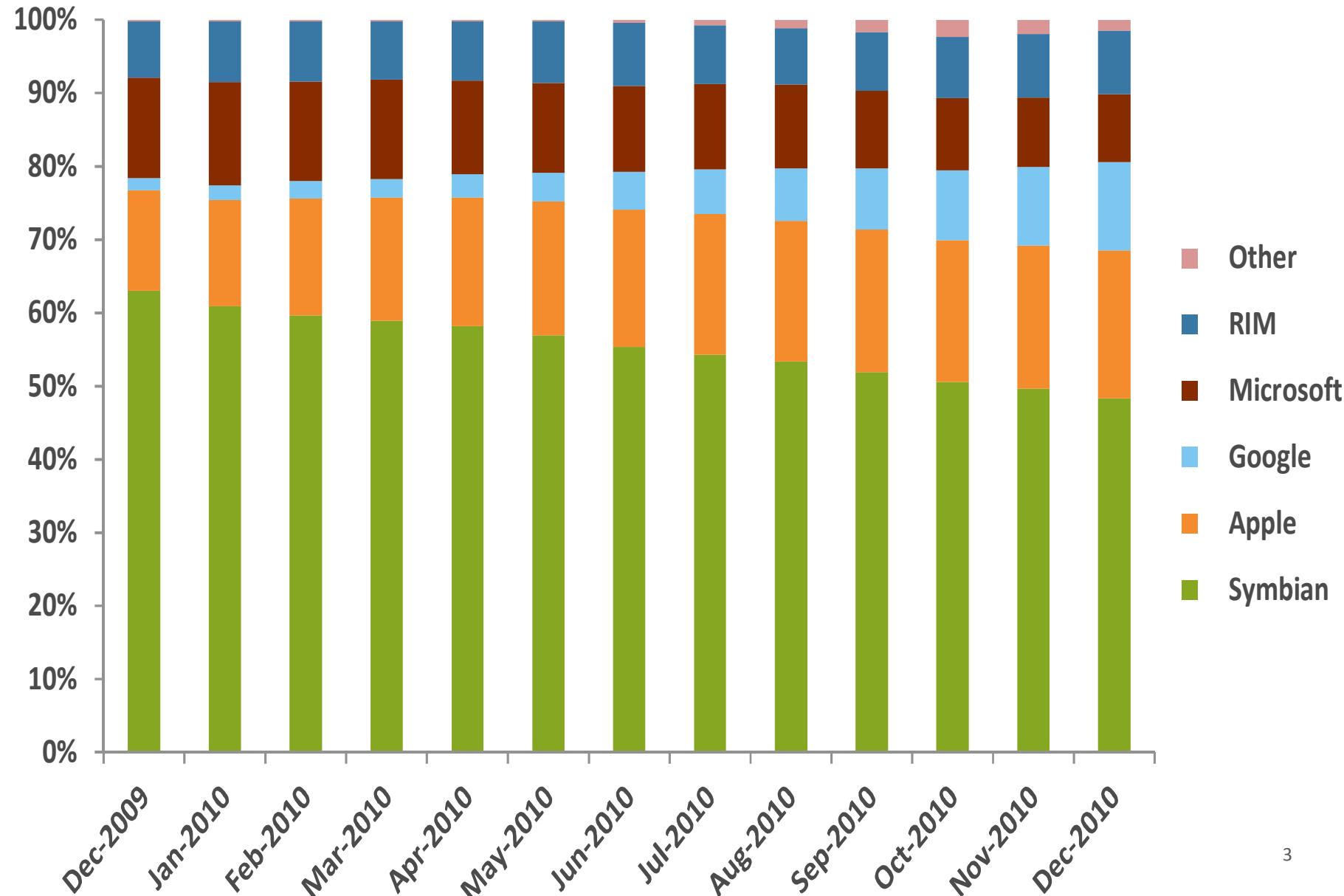


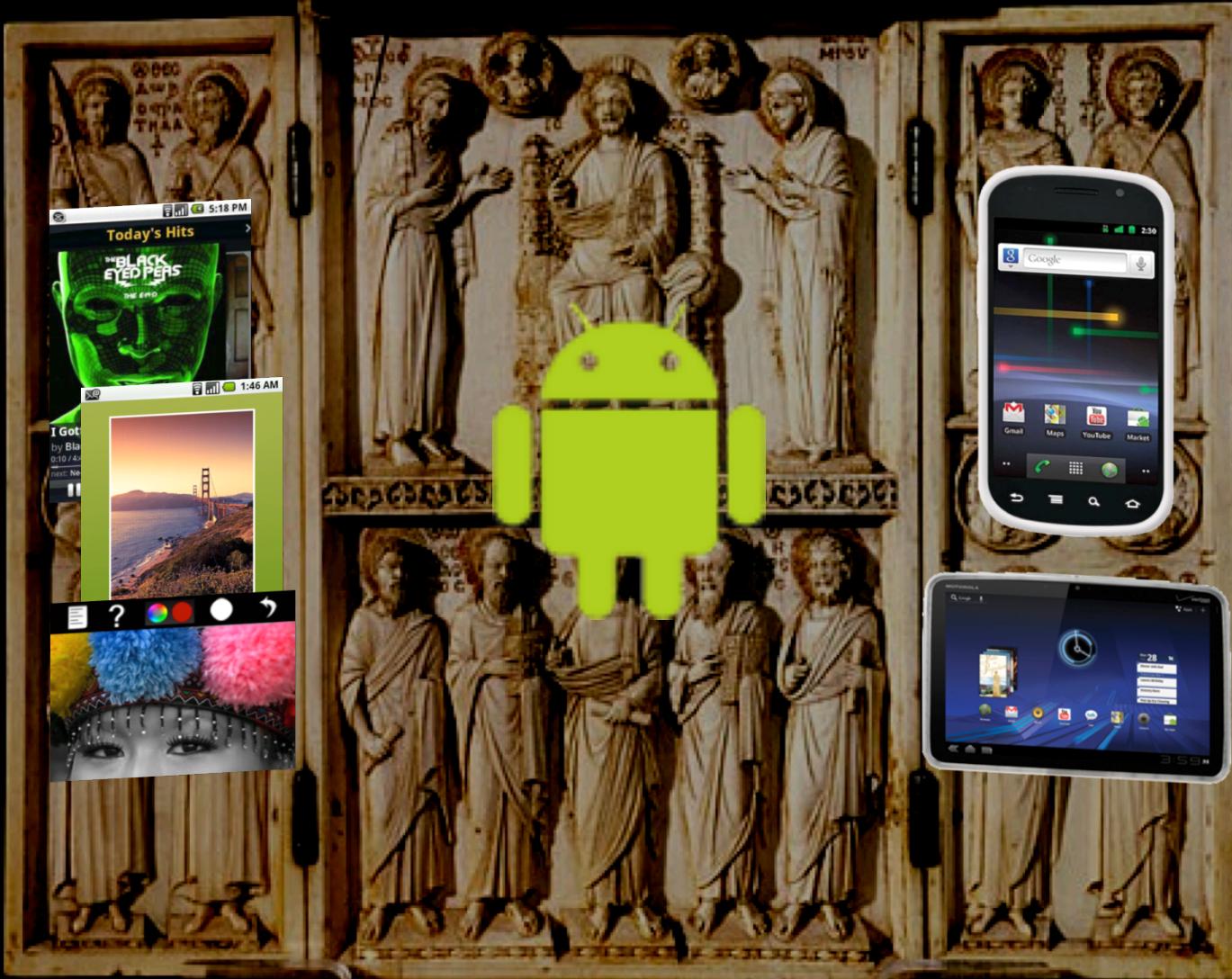
**The Apache Software Foundation**  
<http://www.apache.org/>



# % Share of Smartphone Market by Operating System in EU5

Source: comScore MobiLens, 3 mo. avg. ending Dec-2009 to Dec-2010







*I love driving  
developers  
crazy*



- **Eclipse Android Development**
  - Development, Tests, Debug
  - Export and signature of APKs
  - Limited modularity
  - Eclipse-based build process
  - No release process (export wizard)
- **Apache Ant**
  - Compilation, Test, Packaging
  - Release
  - Dependencies management
  - Hard to maintain – Scalability – Recursion
  - Can quickly differ from standard builds



- **Let's rationalize Android development**
  - Support Android application production from **development to delivery**
- **Simplification of the development model**
  - Bootstrap a project in less than 1 minute
  - Improve the logging system (Androlog)
  - Injection framework (Roboject)
- **Reliable build and sophisticated branding**
  - Maven-based build system (maven-android-plugin)
  - Advanced modularity and composition (rindirect)
- **Improve quality and feedback**
  - Integration and UI tests (Marvin)
  - Continuous integration (Android plugin for Jenkins)
  - Complete crash report (Androlog)



*This is a work of fiction. Names, characters, places and projects are all products of the author's imagination. Any resemblance to actual events, locales or projects is entirely coincidental ;-)*

- An eBook Reader



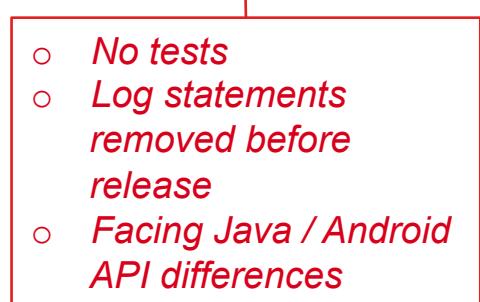
- DRM
  - Native library
  - Server integration
  - Tethering
- Library / Vault, Book Store
  - Server integration
- Social aspect
  - Friends, Sharing, Loaning
- Multi-platform
  - iOS, Android, Desktop (MacOS / Windows)
  - Synchronization
- > 2 years of active development



Alice opened the door and found that it led into a small passage, not much larger than a rat-hole: she knelt down and looked along the passage into the loveliest garden you ever saw. How she longed to get out of that dark hall, and wander about among those beds of bright flowers and those cool fountains, but she could not even get her head through the doorway; "and even if my head *would* go through," thought poor Alice, "it

## Version 1

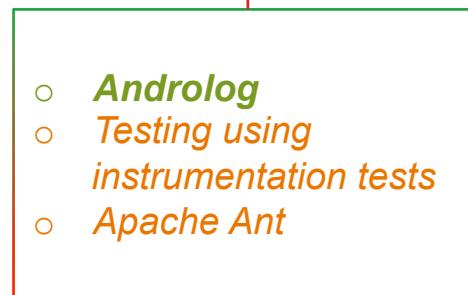
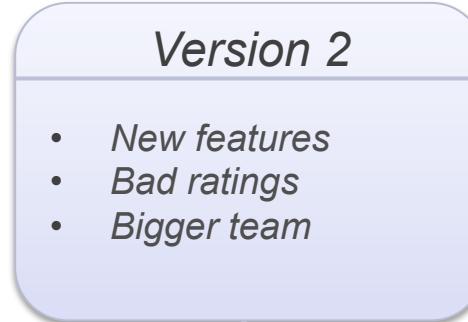
- *Eclipse*
- *2 developers*
- *Native lib integration*

- 
- 
- *No tests*
  - *Log statements removed before release*
  - *Facing Java / Android API differences*

## Version 1

- Eclipse
- 2 developers
- Native lib integration

- 
- A large red rectangular box contains a diagonal red stamp with the word "FAILED" in white capital letters. Below the stamp, there is a bulleted list of issues:
- No tests
  - Log statements removed before release
  - Native Java / Android API differences



## Androlog is a configurable log facade for Android

- **Per-logger level configuration**
  - Log disabled on end-user devices
  - Can be enabled by developers to reproduce issues
- **Reporting**
  - Crashes can be reported by mail or a portal
    - Device info
    - Log
    - Complete stack traces
- **Makes logging a lot easier**
- **Faster bug fixes**



## Received reports

Mon Sep 19 15:22:35 MESZ 2011  
on Nexus S using Android 10

Mon Sep 19 15:22:35 MESZ 2011  
on Nexus S using Android 10

Mon Sep 19 15:22:34 MESZ 2011  
on Nexus S using Android 10

Mon Sep 19 15:22:34 MESZ 2011  
on Nexus S using Android 10

Mon Sep 19 15:22:34 MESZ 2011  
on Nexus S using Android 10

Mon Sep 19 15:22:34 MESZ 2011  
on Nexus S using Android 10

Mon Sep 19 15:22:33 MESZ 2011  
on Nexus S using Android 10

Mon Sep 19 15:22:32 MESZ 2011  
on Nexus S using Android 10

Mon Sep 19 15:21:28 MESZ 2011  
on Nexus S using Android 10

Mon Sep 19 15:21:28 MESZ 2011  
on Nexus S using Android 10

Mon Sep 19 15:21:28 MESZ 2011  
on Nexus S using Android 10

Mon Sep 19 15:21:27 MESZ 2011  
on Nexus S using Android 10

Mon Sep 19 15:21:27 MESZ 2011

Mon Sep 19 15:22:35 MESZ 2011

## ▼ Messages

Report received on *Mon Sep 19 15:22:35 MESZ 2011*.

## Message

Uncaught Exception



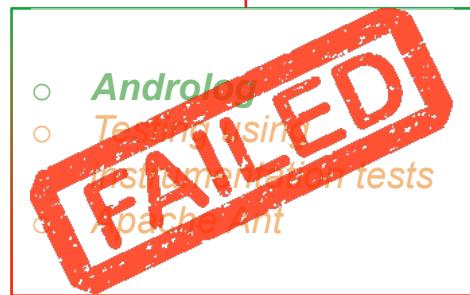
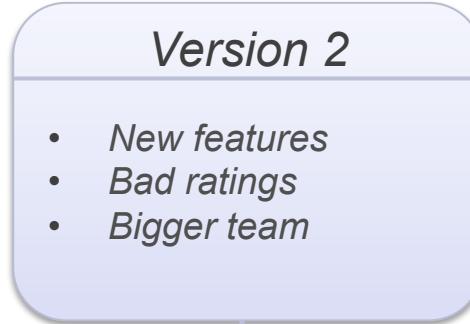
## Error : ...

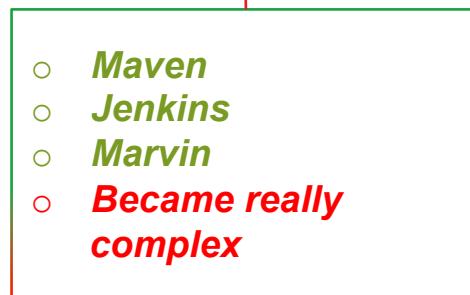
```
java.lang.UnsupportedOperationException: ...
at
de.akquinet.gomobile.androlog.test.PostReporterTest$1.run(PostRe
porterTest.java:132)
at java.lang.Thread.run(Thread.java:1019)
```



## Cause :

- ▶ Log
- ▶ Device
- ▶ Application Data





## Reliable build system

- **Same build on all machines**
  - Convention over configuration
  - Continuous integration support
  - Dependency management
    - Modularity / Branding
- **Release process**
  - Create *marketplace-ready* APK
- **Archetypes**
  - Quickstart, with-test, release, stand
- **More reliable**
- **More maintainable**
- **Release process**



Tests Android applications continuously on several platforms

- **1.6 to 3.1**
  - A lot of devices are available
  - Huge Fragmentation (1.6)
- **Tests executed on 25 devices**
  - Covering the most common devices
  - Emulator configurations
- **More reliable applications**
- **Less surprises especially when using internals**





# Build #58 (Sep 13, 2011 12:07:01 PM)

[Keep this build forever](#)[Delete this build](#) [add description](#)

Started 8 days 2 hr ago  
Took [1 hr 10 min](#)



## Changes

1. [maven-release-plugin] prepare release txtr-project-4.0.4 ([commit: ae1a903ae8fceba999fb37a49e4ce1284f32c44d](#)) ([detail](#))
2. [maven-release-plugin] prepare for next development iteration ([commit: d7f0529660262c25e0c264cc5fa9df9b43021c2e](#)) ([detail](#))

[Started by an SCM change](#)**Revision:** d7f0529660262c25e0c264cc5fa9df9b43021c2e

- origin/master

## Android Configuration

	Result
Google G1	
Acer Liquid A1	
HTC Magic	
LG GT540 Optimus	
LG LU2300 Optimus	
Motorola i7500 Galaxy	
Sony Xperia X10 (1.6)	
Archos 5	
Acer Liquid E	
HTC Desire (2.1)	
HTC Hero (2.1)	
HTC Legend (2.1)	
Google Nexus One	
Motorola Milestone (2.1)	
Motorola DroidX (2.1)	
Samsung Galaxy S	
Sony Xperia (2.1)	



Instrumentation tests are ... really annoying to write !

- Android instrumentation tests
  - A lot of asynchronicity
  - Tests don't easily navigate across activities
  - UI Tests are big burdens
- Marvin simplifies test writing
  - A layer on the top of instrumentation tests
  - A set of *hamcrest* matchers
  - Integrated into the build process
- **Reduce the amount of code required to write tests**
- **Support complex workflows**
- **More reliable applications**

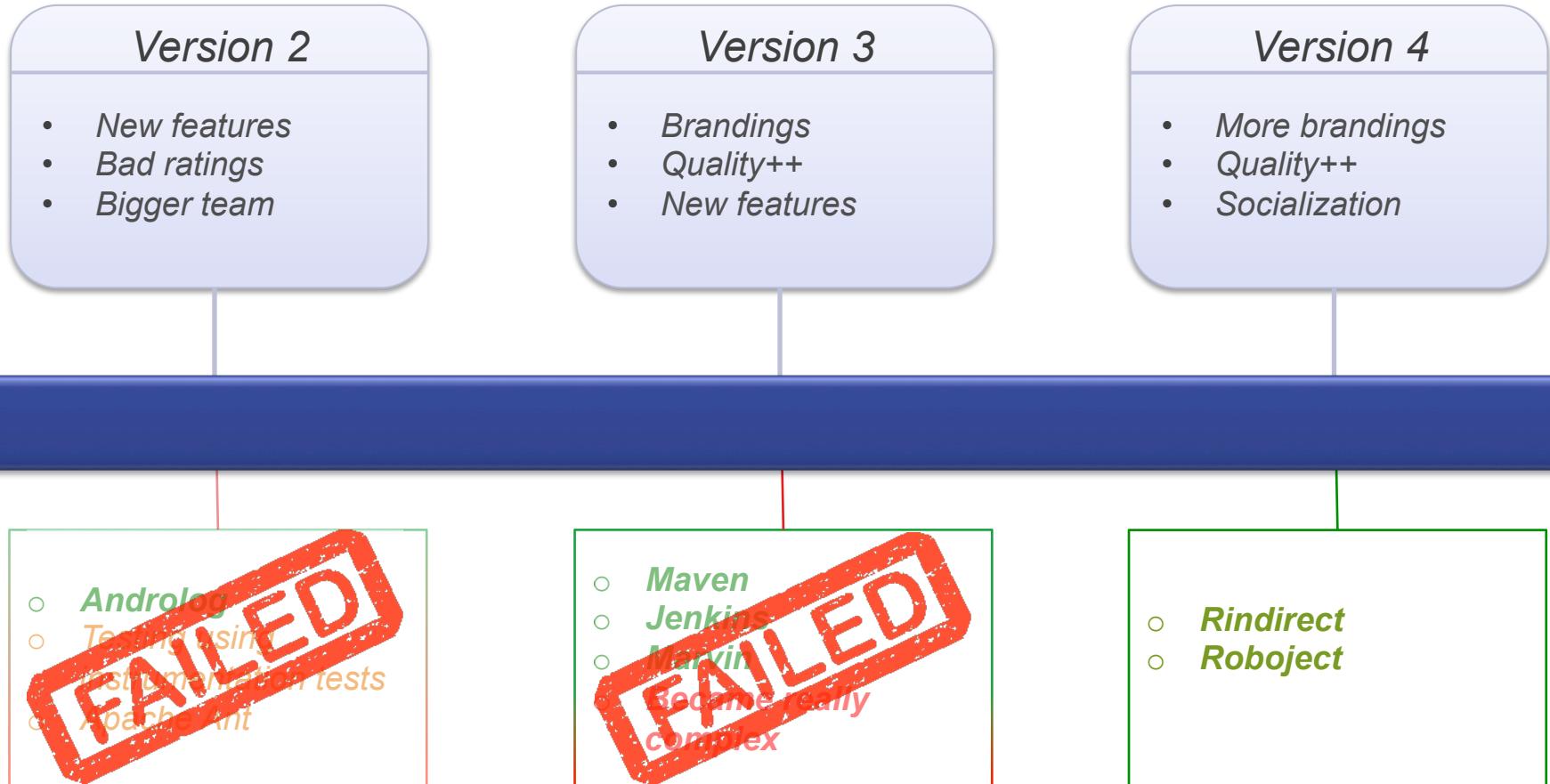


```
public void testLogin() throws Exception {
    Instrumentation instr = getInstrumentation();

    ActivityMonitor monitor =
        instr.addMonitor(MainActivity.class.getName(), null,
        false);
    Intent intent = new Intent(Intent.ACTION_MAIN);
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    intent.setClassName(instr.getTargetContext(),
        LoginActivity.class.getName());
    instr.startActivitySync(intent);
    instr.waitForIdleSync();
    instr.sendStringSync("clement");
    instr.sendKeyDownUpSync(KeyEvent.KEYCODE_DPAD_DOWN);
    instr.sendStringSync("magic");
    instr.sendKeyDownUpSync(KeyEvent.KEYCODE_DPAD_DOWN);
    instr.sendKeyDownUpSync(KeyEvent.KEYCODE_DPAD_CENTER);

    Activity mainActivity =
        instr.waitForMonitorWithTimeout(monitor, 30000);
    assertNotNull(mainActivity);
}
```

```
public void testLogin() throws Exception {
    LoginActivity loginActivity =
        startActivity(LoginActivity.class);
    assertThat(loginActivity)
        .sendString("clement")
        .keyDownUp(KeyEvent.KEYCODE_DPAD_DOWN)
        .sendString("magic")
        .keyDownUp(KeyEvent.KEYCODE_DPAD_DOWN)
        .keyDownUp(KeyEvent.KEYCODE_DPAD_CENTER)
        .startActivity(MainActivity.class);
}
```



## Sophisticated modularity for Android

- Android does not support reusing components from another application
  - Package conflict
    - Resources not found in R
  - Libraries do share the same package
- Rindirect removes those limitations by generating the missing/required R files
  - Automated during your build process
- **Allow advanced modularity**
- **Improve component sharing**



## A CDI for Android

- Android code can be really tiresome
  - A lot of asynchronicity
    - Service bindings, intents...
    - Predominance of the main thread
  - UI code can be smaller
    - Retrieving a View, Setting a Layout
- Roboject drastically simplifies Android code
  - `@InjectX` where X is Service, Intent, Extra, View, Layout...
  - Additional lifecycle method : `onReady`
- **Reduce the amount of code required to develop Android applications**
- **Minimum runtime overhead**
- **Extensible**



```
@InjectLayout(R.layout.searchinputactivity)
public class TweetSearchActivity extends RobojectActivity {
    @InjectView
    private EditText editTextSearch;

    @InjectView
    private Button buttonSearch;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        buttonSearch.setOnClickListener(new OnClickListener() {
            public void onClick(View arg0) {
                Intent intent = new Intent(TweetSearchActivity.this,
                    TweetListActivity.class);
                intent.putExtra("searchTerm",
                    editTextSearch.getText().toString());
                startActivity(intent);
            }
        });
    }
}
```

## Version 2

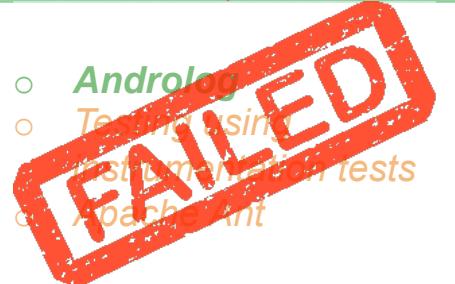
- New features
- Bad ratings
- Bigger team

## Version 3

- Brandings
- Quality++
- New features

## Version 4

- More brandings
- Quality++
- Socialization



*An example  
would be useful  
right about now*

*If we have time...*



- ✓ *Androlog*
- ✓ *Maven-Android-Plugin*
- ✓ *Jenkins Plugin*
- ✓ *Marvin*
- ✓ *Rindirect*
- ✓ *Roboject*
- ✓ *Enterprise Extensions*



*<http://stand.spree.de>*