akquinet

# Rationalizing Android Development

Philipp Kumar

# Who am I?


**Philipp Kumar**

- **akquinet tech@spree** GmbH
- **Mobile** Solutions
- Focus: **Android** and its **Enterprise Integration**

# Who are we?

OSGi

UI Design

Trainings

Android

JBoss

Development

Java / JEE

Consulting


User Experience

Modular Systems

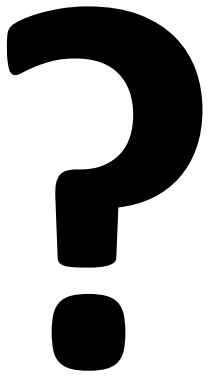
Web Portals

Open Source

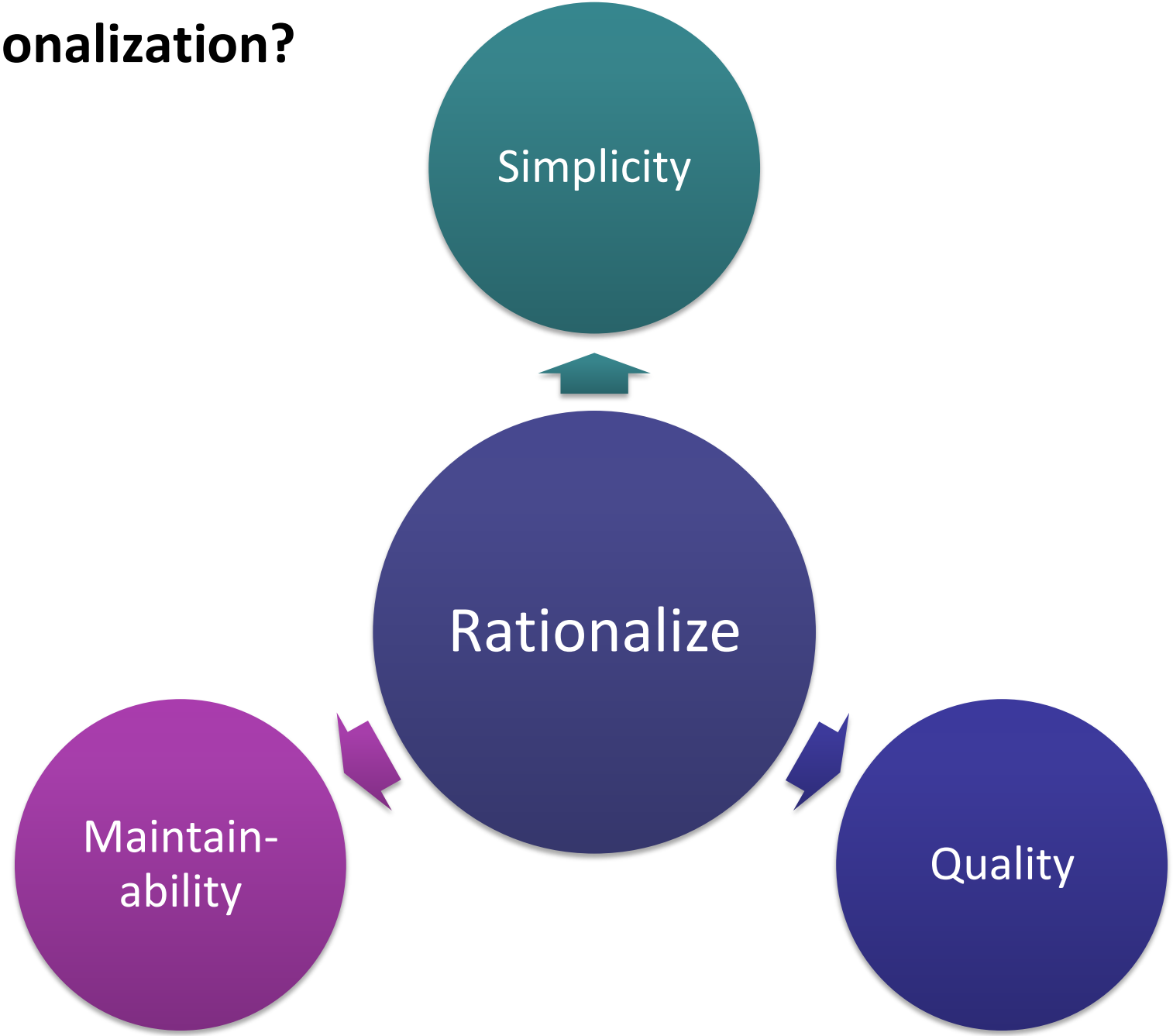
Integration

Mobile Systems

# Who are you?



# Rationalization?



# DEMO

## Android Development

# Android's default set of tools


## What we have…

- **Android SDK**
  - Compiler, Packager, Emulator, …
- **Eclipse Plugin**
  - Android IDE

# Now imagine…

akquinet

**… more complexity.**

- **One** screen -> **Multitude** of screens
- **Offline** app -> **Server** interaction
- **Java only** -> Java plus **native** C/C++ components
- **Foreground** only -> **Background** tasks/services
- **10** Lines of Code -> >**10000** Lines of Code
- …

# DEMO

## A real-world app

# Default Set of Tools



**Again, what we have…**

- **Android SDK**
  - Compiler, Packager, Emulator, …
- **Eclipse Plugin**
  - Android IDE

# Default Set of Tools


**Again, what we have…**

- **Android SDK**
  - Compiler, Packager, Emulator, …
- **Eclipse Plugin**
  - Android IDE

**insufficient…**

# Default Set of Tools



**Again, what we have…**

- **Android SDK**
  - Compiler, Packager, Emulator, …
- **Eclipse Plugin**
  - Android IDE

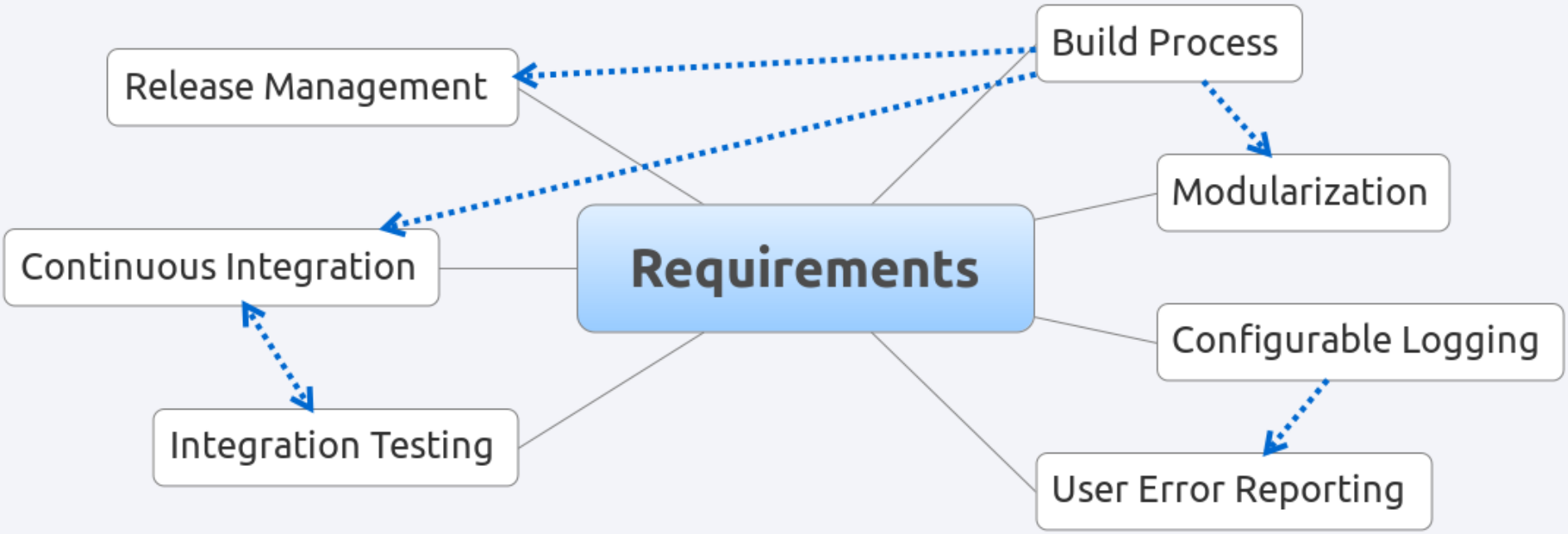
**insufficient…**

**why?**

# Requirements

- **Correctness**
  - testable, consistent, market-ready
- **Portability**
  - address Android fragmentation
- **Maintainability**
  - **simple ↔ comprehensible → changeable/extensible**

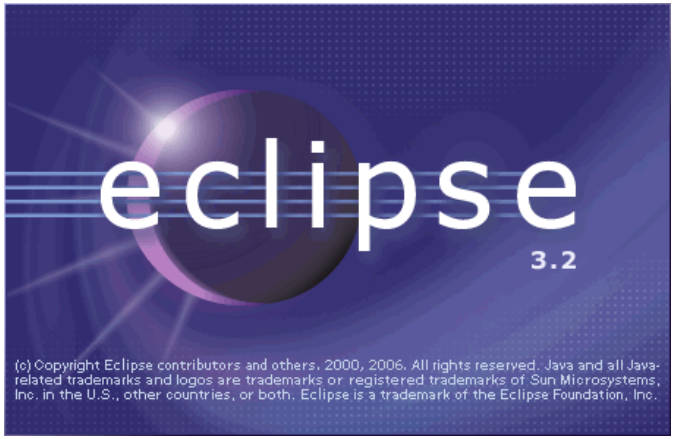
# Requirements



akquinet

# Build Process

# Build from IDE




- Easy

**but …**

- **Platform-dependent**
  - Non-reproducible
  - Error-prone
- **Cannot be automated**

# Build using Ant


- Can be automated
- Reproducible

**but …**

- **No standardized build process**
- **No dependency management**
- **Maintenance nightmare**

# Solution: Use Maven!

Descriptive

Modular

Standardized


Release Process

Dependency Management

# Modularize with Maven

maven-android-plugin

Easy to use Maven plugin for Android™!

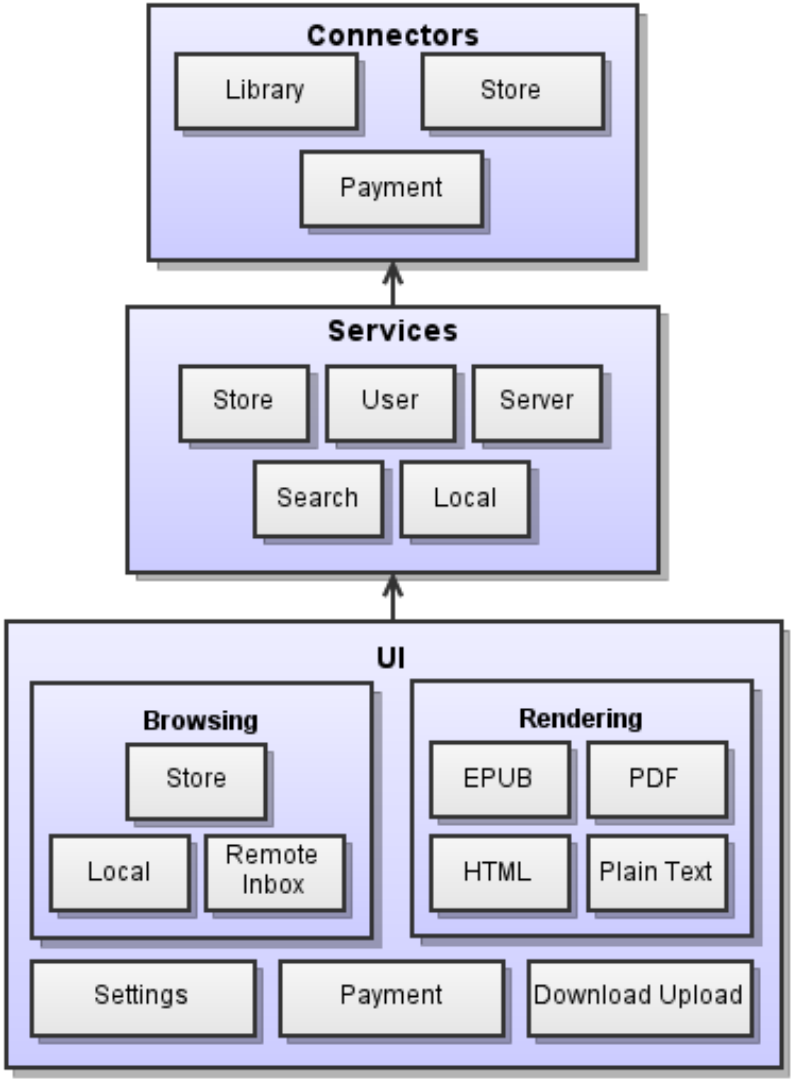
\+ maven-rindirect-plugin

- **Divide** your Android projects into **smaller** projects
- Subprojects can depend on
  - **Ordinary Java** projects
  - **Android** projects

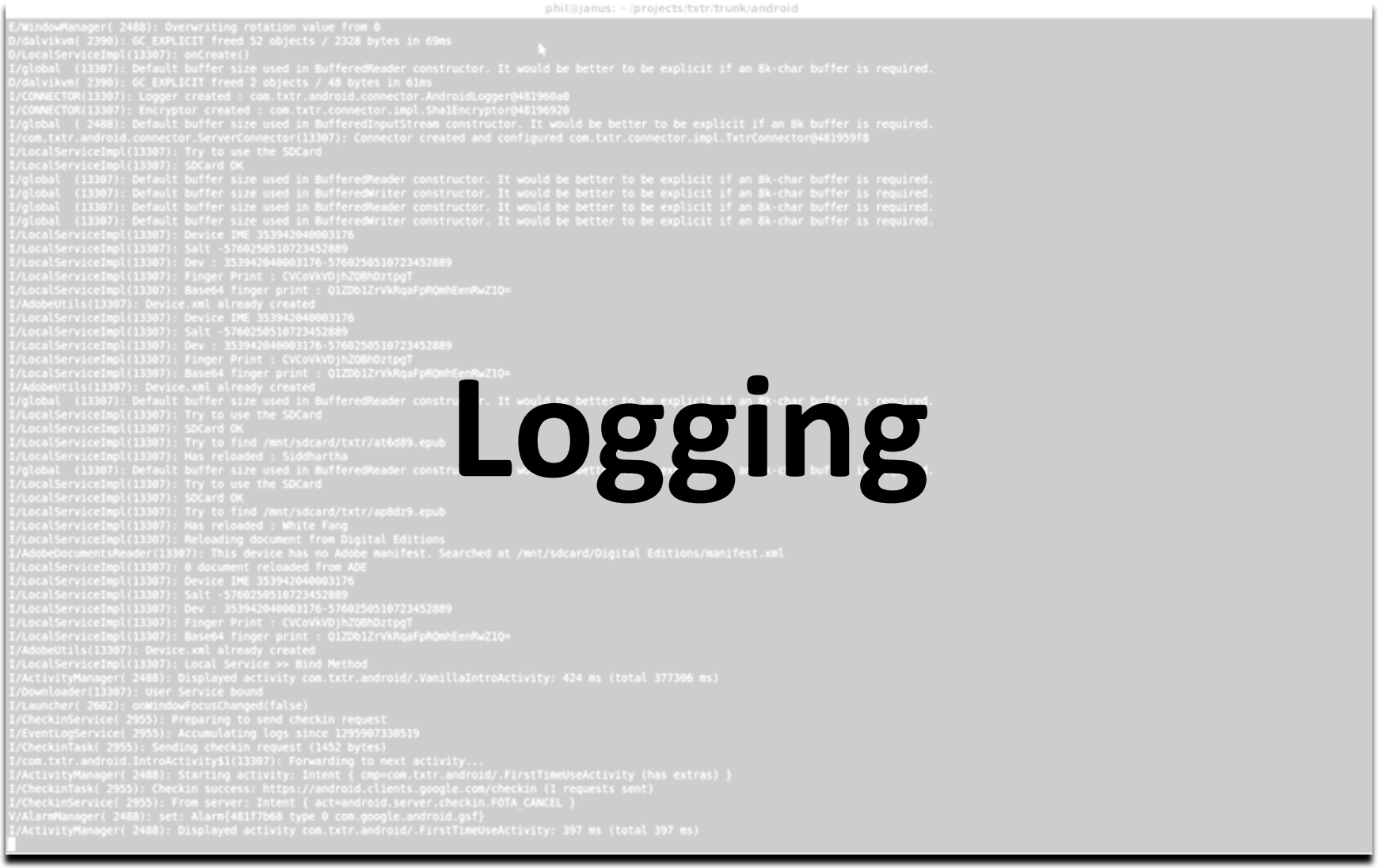
**http://code.google.com/p/maven-android-plugin**

**https://github.com/akquinet/android-rindirect**

# Real-world modularization



# Logging



# Logging

akquinet

## What Android provides

- Logging mechanism
- Different log levels
- Log level can be set at runtime (per-device)
  - Requires appropriate rights!

## What we needed

- Per default, **do not log anything!**
- Enable/configure logging at runtime **if needed**
- Configure logging **per app**

# Use Androlog for logging

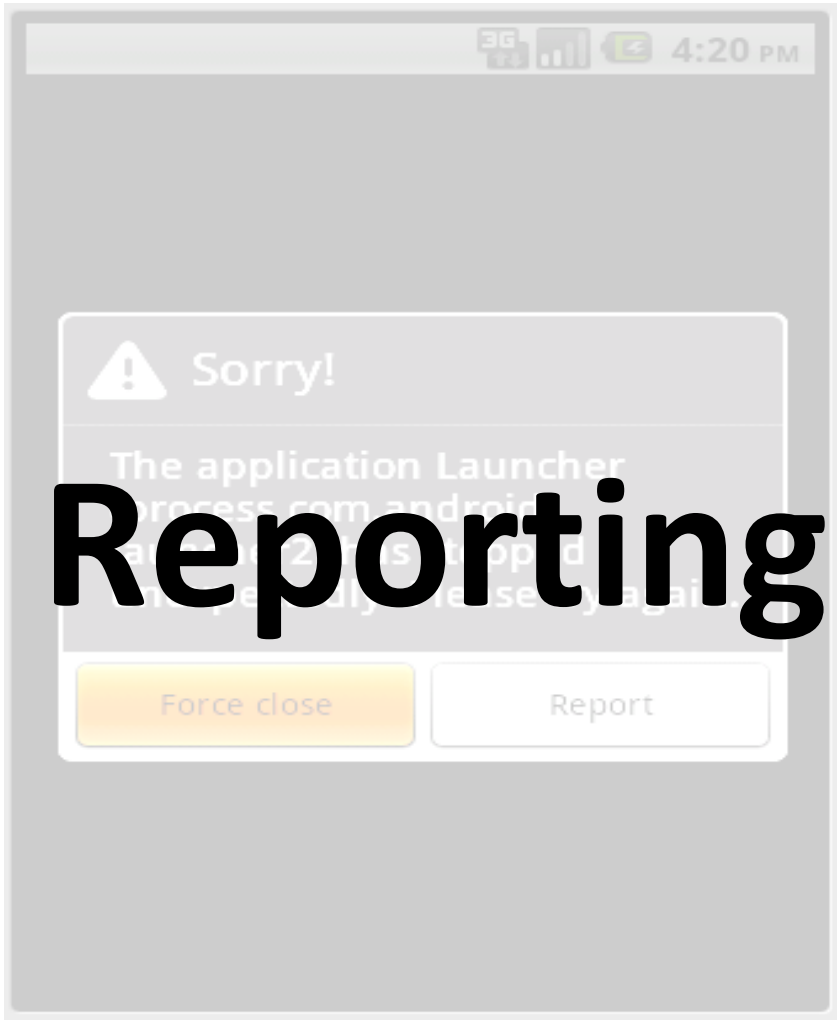
akquinet

## androlog framework

- Wrapper on top of Android‘s Log system
- Enable/disable logging
- Configure loglevel
- Open-source

**https://github.com/akquinet/androlog**

# Behind the scenes



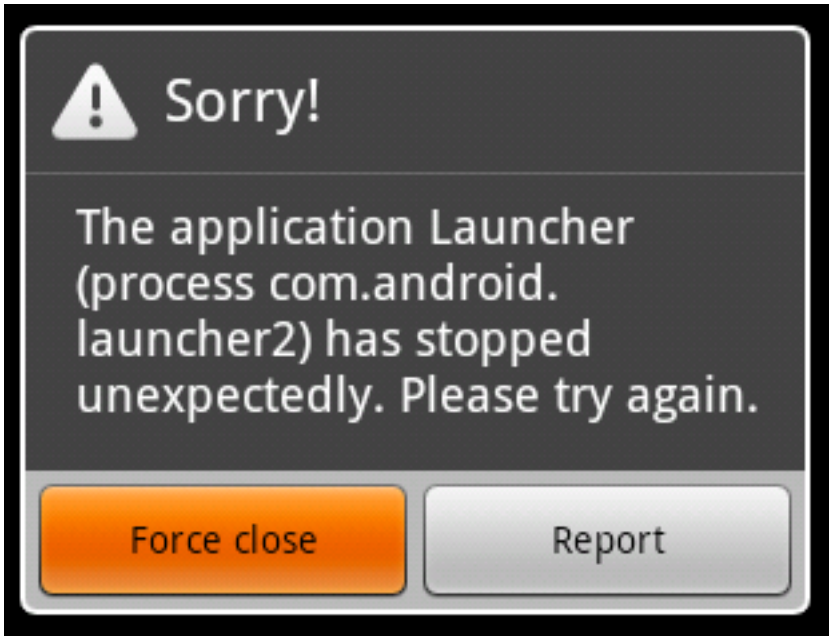

Reporting

# Reporting

## What Android 2.2 provides

- „Send report“ prompt on error
- Report contains
  - **Exception** stack trace
  - Custom **user message**

**=> Insufficient**



## What we needed

- **Pre**-Android-2.2 reporting!
- Report **context** information
- Report **device** information

# Reporting

## Androlog… again ☺

- Catch exceptions and send a **more detailed report**
  - containing log messages
  - containing device info
- Send reports via
  - Email
  - HTTP Post
  - …
    → extendable

**https://github.com/akquinet/androlog**

# Integration Testing

# Integration Testing

akquinet

## Automatically test the whole app on a device or emulator

- Supported in Android via **Instrumentation**
  - Start application under test…
  - … deliver events…
  - … and define assertions on resulting behavior.

# DEMO

## Integration Testing

# Integration Testing

## Marvin: Android Test Framework

- **Control** Activities and Services
- **Define** Assertions
- **Inject** events

  - **Ease writing and maintenance** of tests

**https://github.com/akquinet/android-marvin**

# Continuous Integration

## Continuous Integration

- **Build** the app on a **dedicated** server
- **Build automatically** after every commit
- **Test** by automatically **starting emulators**
  - … and run tests **covering all Android versions**
- **Send mails** immediately on build/test failure
- **Build nightly** beta versions

## We use

- Hudson/Jenkins + Android plugin

# Release Process

# Release Process

## TODOs on Release

- Run **tests**
- **Enforce** rules
  - e.g. app is not „debuggable“
- **Zip-align** the APK
- **Sign** APK with market key

## Can be automated…

- … using Maven ☺
- Releasing takes 2 minutes
- repeatable!

# Conclusion

# Release Process

## Developing apps for Android…

- … is non-trivial.

- **Maintainability**
  - **Use Maven** to build your app
  - **Modularize** your app
- **Correctness, Portability**
  - Use androlog to get **error reports** from users
  - Write **automated tests**, e.g. with Marvin
  - Set up Hudson to **build/test** your app **continuously**

# Prospects

## Current work in progress

- Easy **injection** of services and resources
- Automatically compute **quality metrics**
- Improve **IDE support**
- …

**http://blog.akquinet.de**

# DEMO

## Current Work in Progress

# Questions & Answers