



**NOSERENGINEERING**

WE KNOW HOW

## **Android & open ADK** eine kleine Einführung und Betrachtung

Rainer Fritzsche



**WE KNOW HOW**



# Agenda

---

- Vorstellung
  - Wer?
- Android
  - Woher, Wohin?
  - Womit, Wie?
- ADK
  - Was, Woher?
  - Womit, Wie?
- ShowCase
  - Was?
  - Wie?

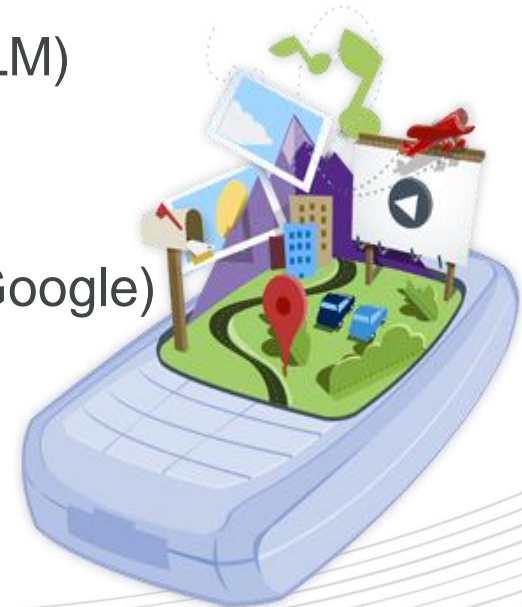


# Vorstellung



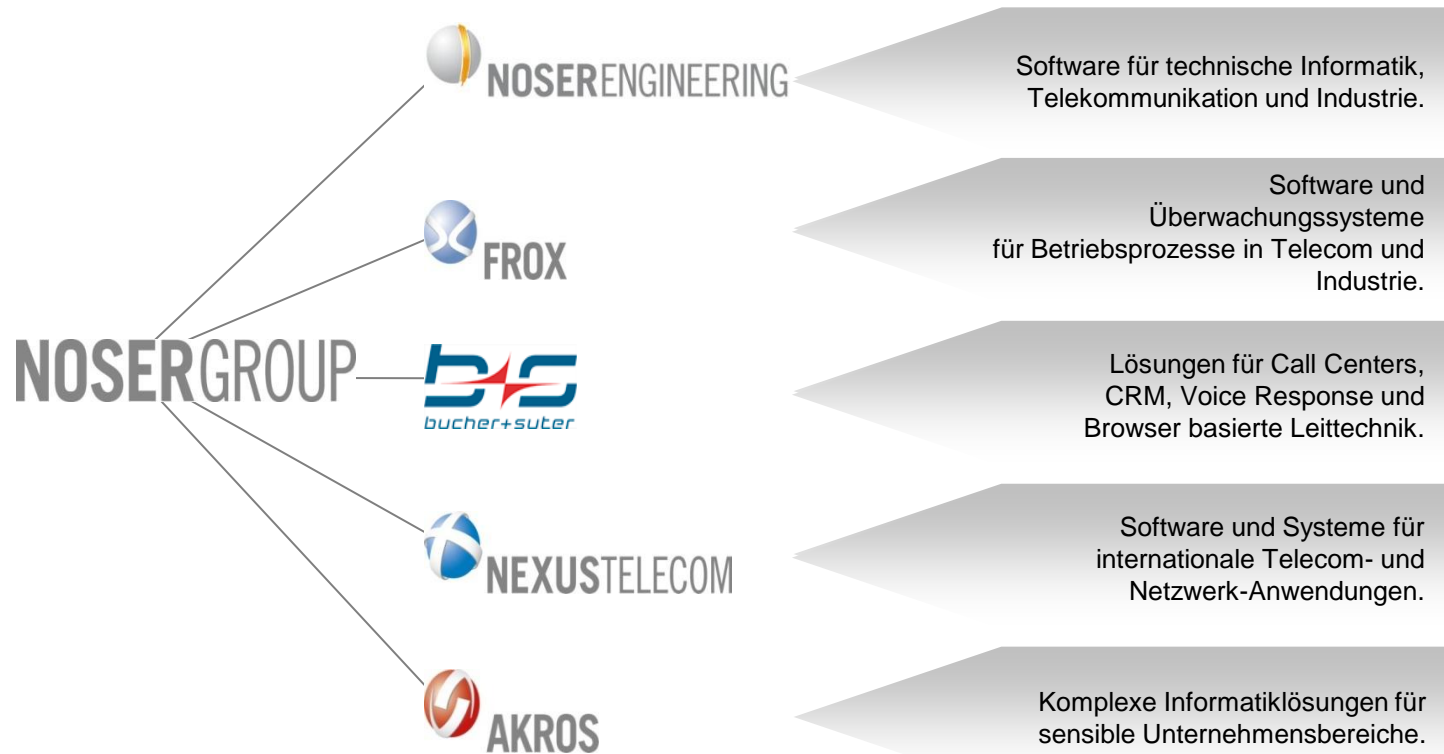
# Vorstellung: Noser Engineering AG

- Gegründet 1984
- 120 Ingenieurinnen und Ingenieure
- 400 Mitarbeiter/innen in der Noser-Gruppe
  
- Microsoft Gold Partner
- Microsoft Inner Circle Award Winner 2009 (ALM)
  
- ICT Award Gewinner 2011 (Champion Award)
- Gründungsmitglied Open Handset Alliance (Google)





# Vorstellung: Die Noser Gruppe



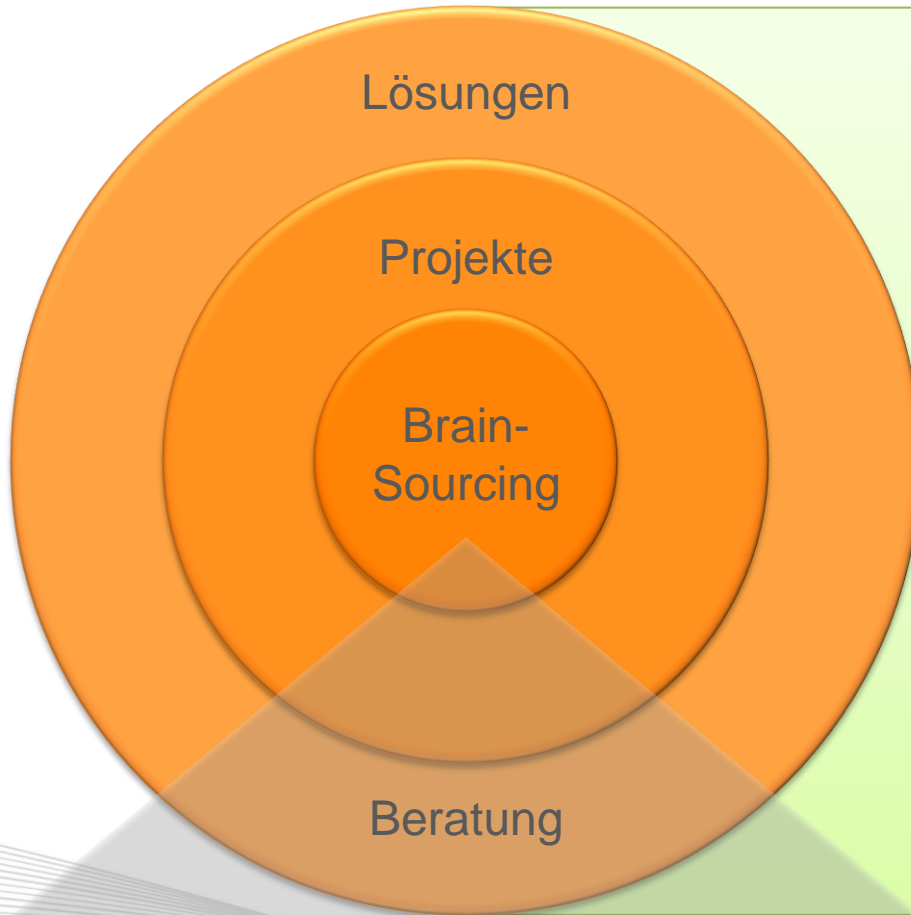
**NOSERENGINEERING**

Rainer Fritzsche, 19.01.2012

**WE KNOW HOW**



# Vorstellung: Noser Engineering AG



## Solutions

Wir realisieren und betreiben Ihre professionelle Lösung.  
Zusammen verbessern wir Ihre Wertschöpfung.

## Projects

Wir liefern Ihnen Ihr Produkt auf Termin.  
Sie kümmern sich um Ihre Kunden, wir uns um den Rest.

## Brain Sourcing

Wir haben die Ingenieure für Ihr Informatikprojekt.  
Gefragtes Wissen zur rechten Zeit.

## Consulting

Wir sind führend in Software Engineering.  
Mit unserer Erfahrung bringen wir Sie weiter.

**NOSERENGINEERING**

Rainer Fritzsche, 19.01.2012

**WE KNOW HOW**



# Vorstellung: KPZ Java



Automotive

Deutscher Automobilhersteller  
Android for Premium Class Cars (NDA)  
  
Amerikanischer Automobilkonzern  
Android Consulting (NDA)



Healthcare  
Medical Systems



Defense



Transportation







# Vorstellung: KPZ Java



Telecommunication  
ICT

ascom

Google

[search.ch]  
*Immer ein Volltreffer.*

sage

swisscom

hp

orange™



Logistics

PostLogistics  
*DIE POST+*

ALSTOM

SWISS  
WINDOWS  
*kompromisslos besser*



Manufacturing

wipf

komax

**Dätwyler Cables**



Machine Engineering

MICAMATION

**schlatter**

soutec+

+GF+  
AgieCharmilles

AGATHON  
SWITZERLAND

MÜLLER MARTINI

# NOSERENGINEERING

Rainer Fritzsche, 19.01.2012

# WE KNOW HOW





# Vorstellung: KPZ Java

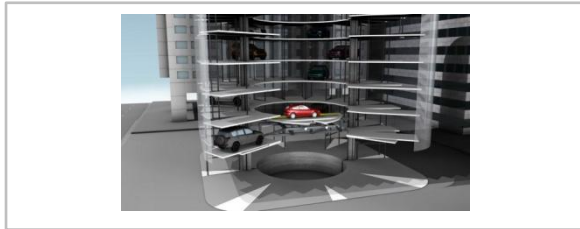


Energy

**ABB**

**BKW**<sup>®</sup>  
BKW FMB Energie AG

**swissgrid**



Plant Engineering

  
**Schindler**

**SULZER**

**SKYLINE  
PARKING**  
SMART CAR  
LOCKING SYSTEMS



Textile

**RIETER**

**Luwa**

**LOEPFE**  
MASTERS IN TEXTILE QUALITY CONTROL



Home Appliances  
Measuring Appliances

  
**TESA**  
TECHNOLOGY

**KISTLER**  
measure. analyze. innovate.

**V  
ZUG**

**schaerer**<sup>®</sup>  
coffee comes to life

**NOSERENGINEERING**  
Rainer Fritzsche, 19.01.2012

**WE KNOW HOW**



# Vorstellung: KPZ Java



Metrology

[**search.ch**]  
*Immer ein Volltreffer.*



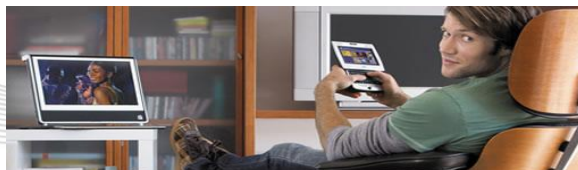
Semiconductors

**œerlikon**



Building Technology

**SIEMENS**



Digital Home

**SIEMENS**



# Vorstellung: Noser Engineering AG

SP  
N  
K  
P  
K

Microsoft

Java

Embedded

Test Engineering



# Vorstellung: KPZ Java

## – Know-how

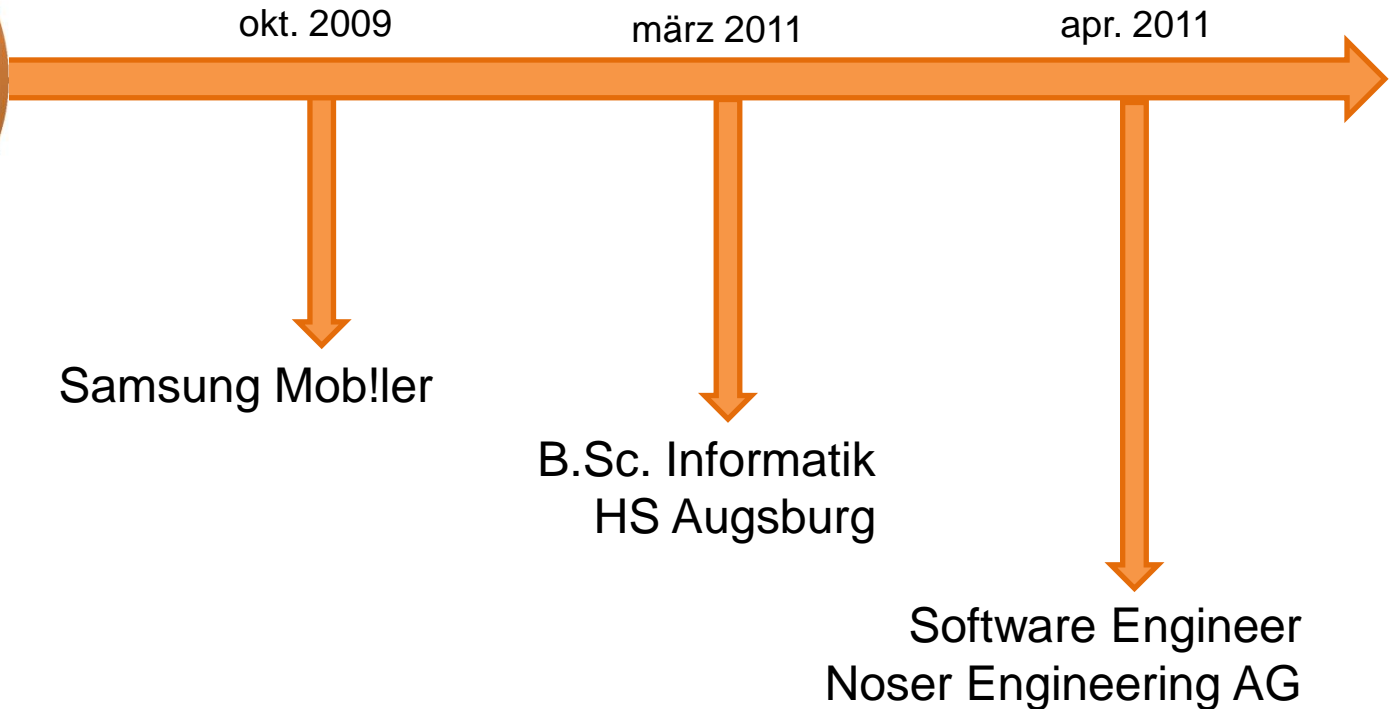
- Java / Linux / Android / DroidCloud / iOS / iPhone / iPad  
iPod touch / Eclipse

## – Dienstleistungen

- Beratung / Projektleitung
- Prozessanalysen und Architektur-Design
- Enterprise Application Integration (EAI)
- Lösungen für die Industrie (Automotive, Entertainment, ICT)
- Lösungen für Linux/Android (Embedded, Portierungen etc.)
- Mobile Lösungen



# Vorstellung: Rainer Fritzsche





# Android

Betriebssystem / Plattform für mobile Geräte



# Android: Historie

## Open Handset Alliance

- Gründung: Nov 2007
- Google & 80+ Mitgliedsfirmen (30+ bei Gründung)
- Ziel: offene Standards für Mobilgeräte



## Android

- 2003 Andy Rubin gründet Android (Unternehmen)
- Okt. 2008 Marktstart des T-Mobile G1
  
- Apps-basierte Plattform mit Market
- Starke Einbindung der Google Dienste







# Android: Historie

2008: Android 1.0



2011: Android 4.0

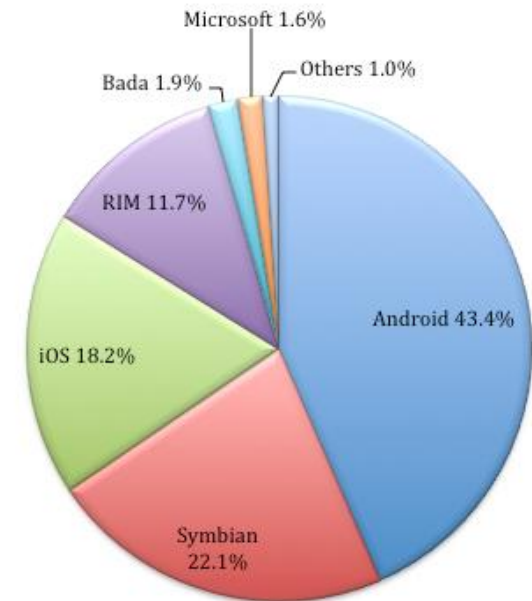
<http://goo.gl/1pSqW>



# Android: Situation

- 700.000 Aktivierung/Tag  
(google 21.12.2011)
- 620.000 Apps im Market  
(40.000 monatlich)

Smartphone market share 2Q11



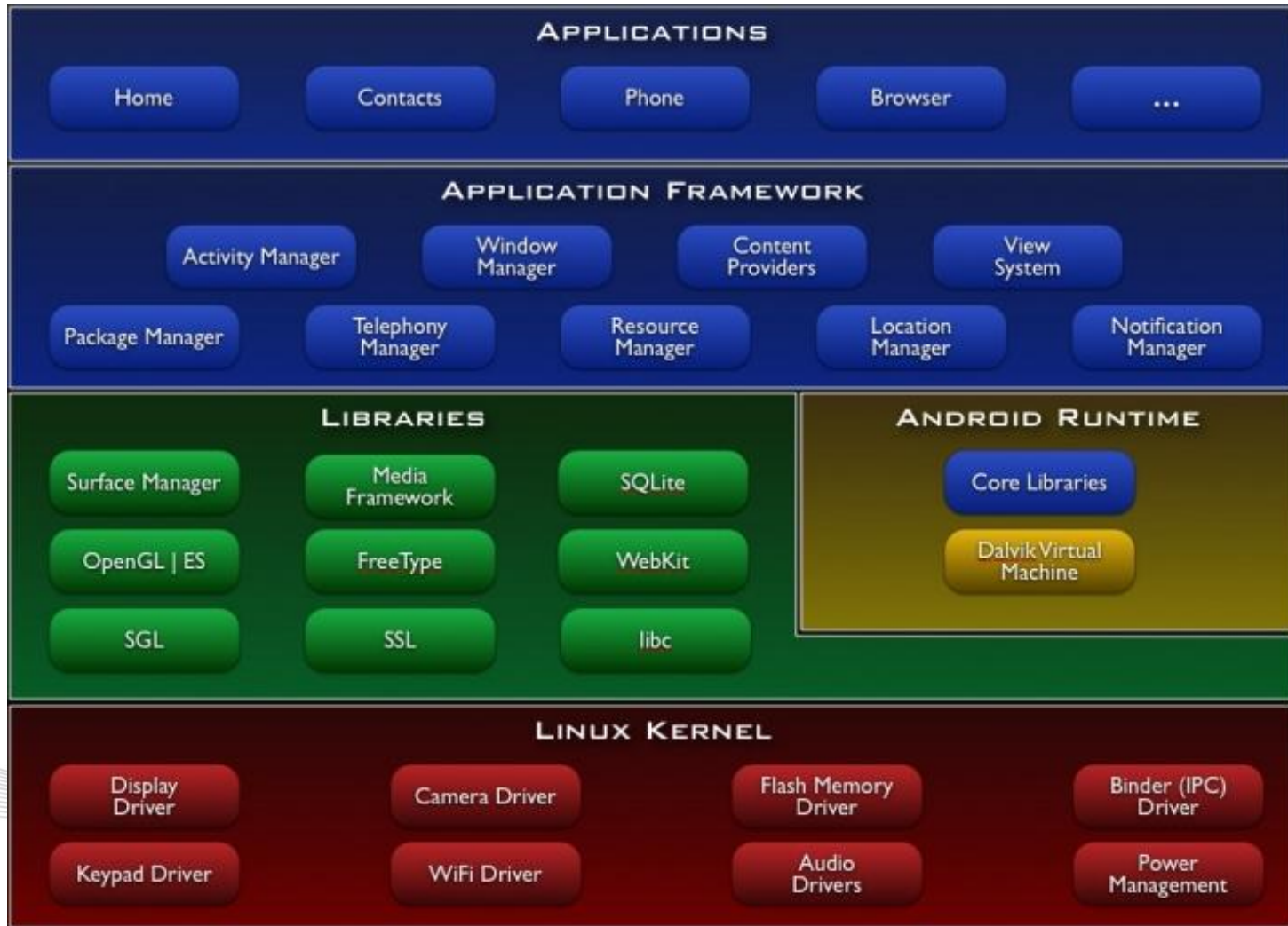


# Android: Einsatzgebiete





# Android: Konzepte





# Android: Konzepte

- **OpenSource**
  - Basiert auf Linux-Kernel 2.6
  - Google apps nicht Open Source
  - Android 3.0-Gate
  - Cyanogenmod
  
- **Activities & Services**  
Lifecycle
- **Intents**  
implicit, explicit
  
- XML-Definition für GUI
- Seit 4.0 Design Guidelines



# Android: Konzepte

- Security-Prinzip: Sandbox
- Apps benötigen Permissions
  
- Resources
  - Manifest.XML
  - Multi-Density, Multi-Language, etc...
  
- Alle Geräte Out-Of-The-Box-Debuggable
  
- Google zertifiziert Geräte
  - Market-Zugang unter bestimmten Voraussetzungen
  - GPS von Anfang an dabei
  
- Kein Zulassungsverfahren für Apps





# Android: Tooling

- Eclipse + ADB-Plugin (windows, linux, macOS)  
beliebige JAVA IDE möglich
- SDK inkl. Emulator
- JUnit
- Jedes Android-Device  
(LogCat)







# Tooling: Eclipse

- Eclipse  
<http://www.eclipse.org/downloads/>
- Android SDK  
<http://developer.android.com/sdk/index.html>
- ADB PlugIn  
<https://dl-ssl.google.com/android/eclipse/>





# Tooling: Eclipse



<live show>



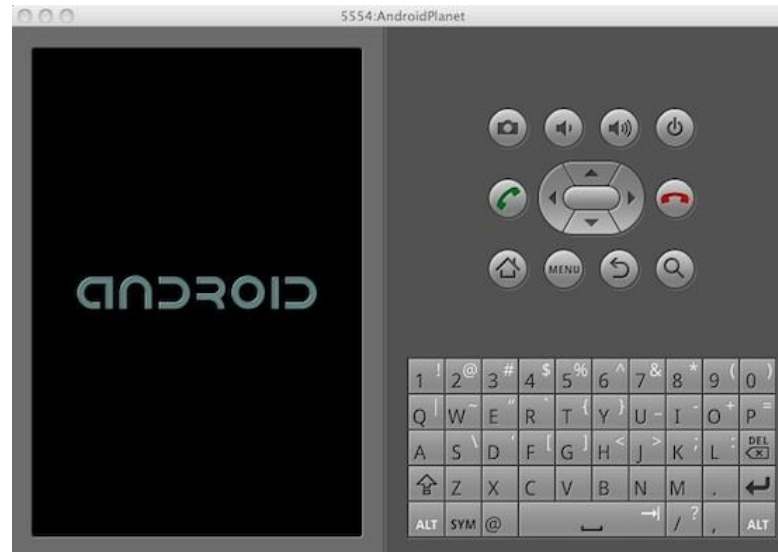
# Tooling: Emulator

- **API-Level**  
auf dem Emulator läuft ein vollwertiges Android OS in einer definierten Version
- **Hardwareprofile**  
Kamera, QWERTZ keyboard, RAM, etc...
- **Diverses**  
Emulatorskin, Displaygröße, emulierter SD-Kartenspeicher
- **Persistent**  
Einstellungen und Daten werden gespeichert





# Tooling: Emulator



<live show>



## Tooling: GUI Editor

---

- WYSIWYG-Editor
- XML-Ansicht
  
- UI-Elemente aller Android Versionen nutzbar
- Verschiedene Auflösungen, Orientierungen
  
- Presets für spezielle Geräte (Samsung Galaxy Tab)



# Tooling: GUI Editor



<live show>



## Tooling: Debug

- Android Device & AVD (Emulator)  
(Funktionstest, DDMS, LogCat)
- Seit Android 4.0 mehr Möglichkeiten am Gerät





# Tooling: GUI Editor



<live show>



# Development: Beispiel

- Was?  
Ein Taschenrechner
- Warum?  
Erste Schritte und UI Konzepte lernen
- Wie genau?
  - Anlegen eines Projektes
  - Nutzen des Ui-Editors
  - Logging



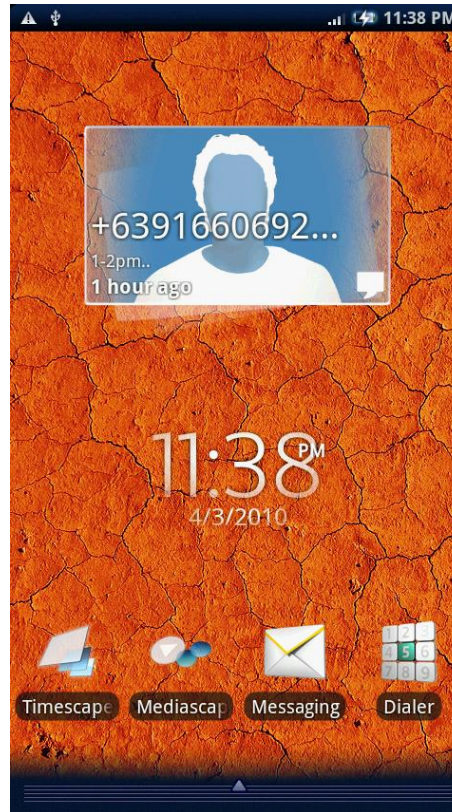
# Development: Beispiel



<live show>

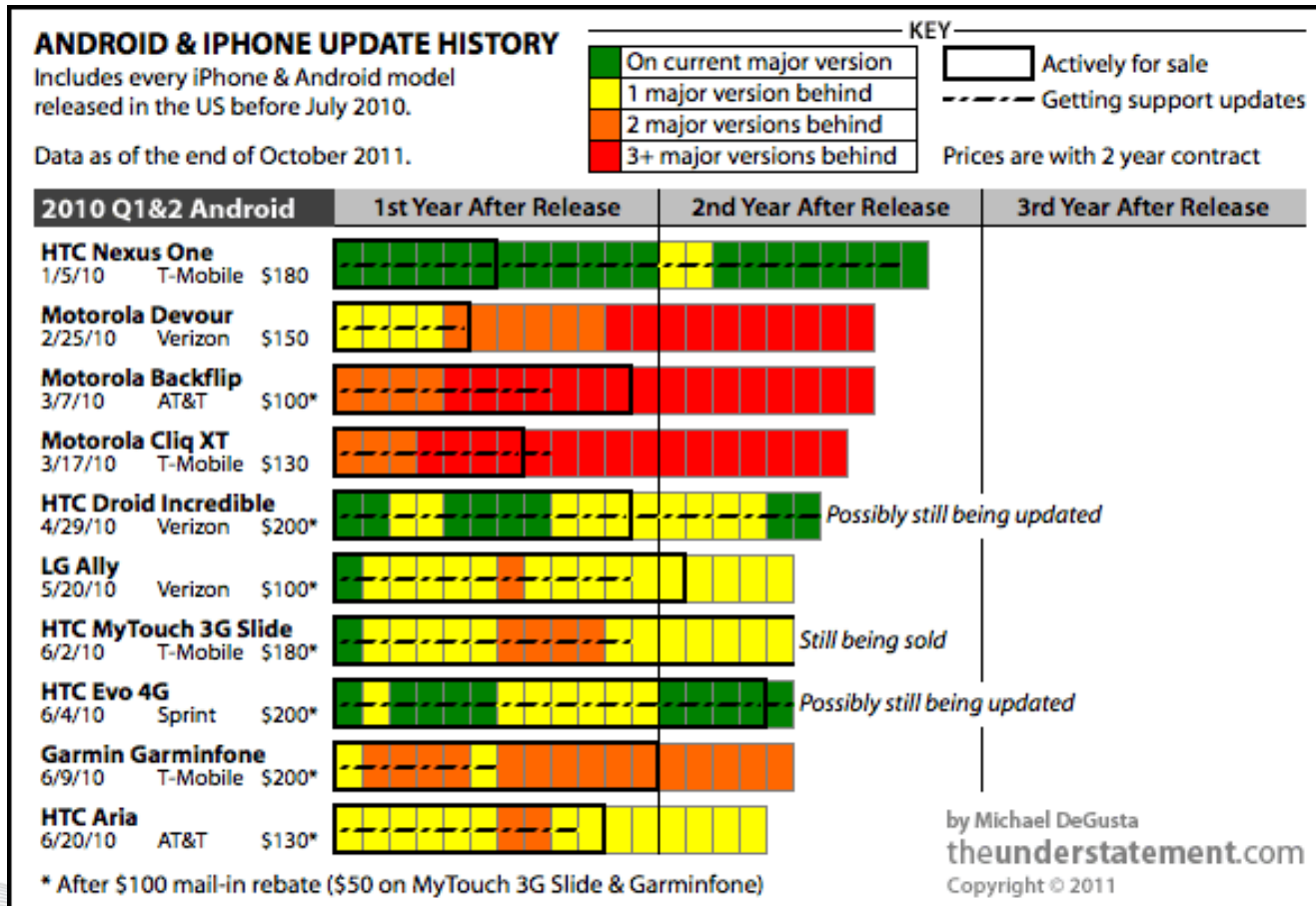


# Problem: Fragmentierung



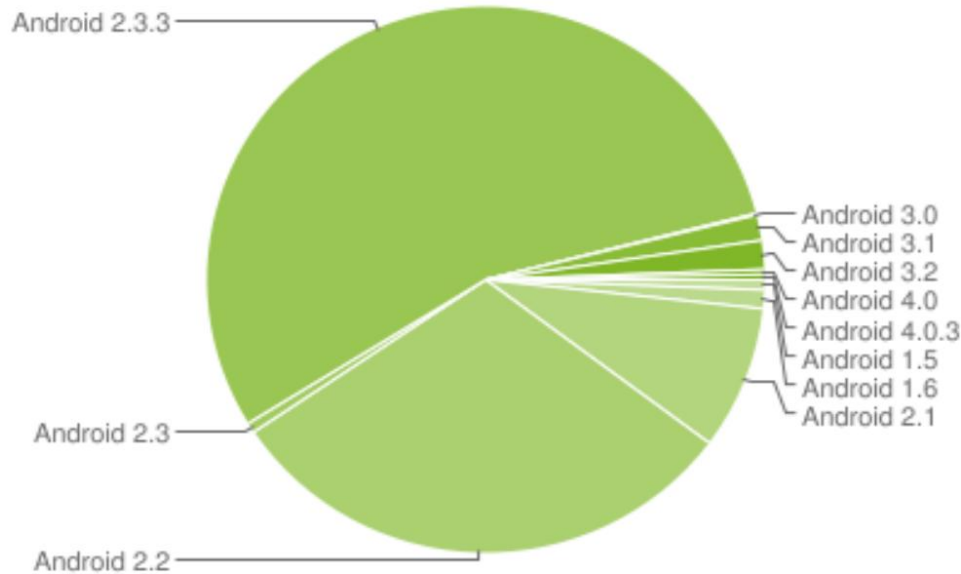


# Problem: Fragmentierung





# Problem: Fragmentierung



Platform	API Level	Distribution
<a href="#">Android 1.5</a>	3	0.6%
<a href="#">Android 1.6</a>	4	1.1%
<a href="#">Android 2.1</a>	7	8.5%
<a href="#">Android 2.2</a>	8	30.4%
<a href="#">Android 2.3 - Android 2.3.2</a>	9	0.6%
<a href="#">Android 2.3.3 - Android 2.3.7</a>	10	54.9%
<a href="#">Android 3.0</a>	11	0.1%
<a href="#">Android 3.1</a>	12	1.5%
<a href="#">Android 3.2</a>	13	1.7%
<a href="#">Android 4.0 - Android 4.0.2</a>	14	0.3%
<a href="#">Android 4.0.3</a>	15	0.3%





# Fragmentierung: Das Problem

- Hersteller genießen viele Freiheiten, das führt zu:
  - Unterschiedlichen Auflösungen
  - Unterschiedlichen Seitenverhältnissen
  - Unterschiedlichen Ui-Elementen
  - Unterschiedlichen Treibern
  - Unterschiedlichen Markets

„Man sollte nicht von Android-Geräten reden,  
sondern von android-kompatiblen Geräten“

([www.readwriteweb.com](http://www.readwriteweb.com))





# Fragmentierung: Lösungsansatz

---

- Android Update Alliance (Mai 2011)
- Android 4.0 UI GuideLines  
(<http://goo.gl/mHIF0>)
- Android 4.0 Holo-Theme als Standard
  - Entwickler können immer darauf zurückgreifen
  - Trennung von custom Ui-Elementen und System  
=> schnellere Updates Möglich



# Arduino & Open ADK



# Arduino: Historie

- Seit 2005
- Soft- und Hardware sind Open Source
- Ziel: leichter Einstieg für disziplinerne Nutzer/Bastler
- Programmiersprache: Wiring-basiert  
vereinfachtes C/C++
- einfaches I/O-Board  
mit Mikrocontroller, analogen & digitalen Ein- und Ausgängen.





# Arduino: Tooling

- Arduino 1.0 IDE  
(windows, linux, macOS)
- Fehlersuche/Debugging  
Texthighlighting, Compiler, Terminalausgabe
- Richtiges Debugging nur mit „großen“ IDEs  
(Bsp.: AVR Studio von Atmel)





# Arduino



<live show>



# Open ADK

- Ab Android 3.1 bzw. 2.3.4 (backport)



- Accessory Mode
- Faked ID

- Unterstützt Open-Accessory Protokoll
- Stellt USB-Host dar
- Versetzt Android-Device in „Accessory Mode“



# Showcase „R2D2“





## ShowCase: Arduino ADK Mega

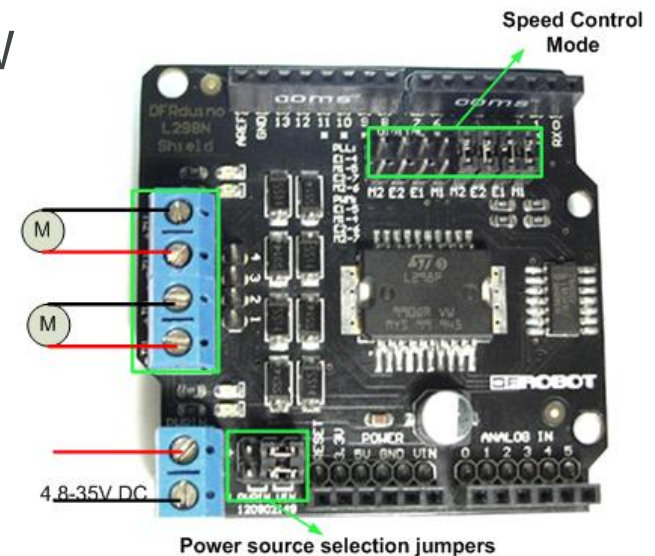
- ATmega2560 Microcontroller
- Operating Voltage 5V
- Input Voltage (recommended) 7-12V
- Input Voltage (limits) 6-20V
- Digital I/O Pins 54 (of which 14 provide PWM output)
- Analog Input Pins 16
- DC Current per I/O Pin 40 mA
- DC Current for 3.3V Pin 50 mA
- Flash Memory 256 KB  
(8 KB used by bootloader)
- SRAM 8 KB
- EEPROM 4 KB
- Clock Speed 16 MHz





## ShowCase: Arduino Motor Shield (L298N)

- Logic Control Voltage : 5V (From Arduino)
- Motor Driven Voltage : 4.8~35V (From Arduino or External Power Source)
- Logic supply current  $I_{ss}$  :  $\leq 36\text{mA}$
- Motor Driven current  $I_o$  :  $\leq 2\text{A}$
- Maximum power consumption : 25W
- PWM、PLL Speed control mode
- Control signal level:
  - High :  $2.3\text{V} \leq V_{in} \leq 5\text{V}$
  - Low :  $-0.3\text{V} \leq V_{in} \leq 1.5\text{V}$





## ShowCase: LEGO Mindstorms R2D2

- Mikrocontroller mit festen Programmen
- Eingebauter Motor (vor und zurück)
- STAR WARS Bonus





# ShowCase



<live show>



**Danke für die Aufmerksamkeit.**



## Quellen:

---

<http://www.android.com>

<http://developer.android.com>

<http://derstandard.at>

<http://www.theverge.com>

<http://www.creapptive.de>

<http://www.androidapptests.com>

<http://www.nodch.de>

<http://fotos.autozeitung.de>

<http://photos.techfieber.de>

<http://cdn-android.iculture.nl>

<http://www.kinder-reisen-sicher.de>

<http://www.browse-tools.com>

<http://www.openhandsetalliance.com/>

<http://www.effiziente-webprogrammierung.info>

<http://fortunebrainstormtech.files.wordpress.com>

<http://media02.hongkiat.com>

<http://www.xda-developers.com>

<http://www.arduino.cc/>

<http://www.makershed.com>

<http://www.productappeal.com>

<http://3.bp.blogspot.com>

<http://www.mobile-zeitgeist.com>



# NOSERENGINEERING

WE KNOW HOW

Home | Über uns | Aktivitäten | Industrien | Tech Centers | Kunden | Academy | Newsroom | Showcase | Jobs & Karrieren | Kontakt



BERATUNG | LÖSUNGEN | PROJEKTE | BRAIN SOURCING

NOSER ENGINEERING AG  
Konrad-Zuse-Platz 1  
D-81829

+49 89 99 014 88 13 direct  
+49 89 99 014 88 0 phone

rainer.fritzsche@noser.com  
www.noser.com

WE KNOW HOW