

ANDROID PRORAMMIERUNG

Max Wielsch

25.03.2015



Saxonia Systems

So geht Software.

Programm ...

1. UMSTÄNDE UND ANFORDERUNGEN AN APPS
2. KONZEPTE
3. VORSTELLUNG DES PRAXISBEISPIELS
4. CODING
5. FAZIT





UMSTÄNDE UND ANFORDERUNGEN AN APPS

Alltag, Interaktion, Hektik, Wandel, ...



Saxonia Systems
So geht Software.

Alltag, Interaktion, Hektik

Der Alltag ist sehr **hektisch**, es gibt viele **Unterbrechungen**.

Man widmet sich kaum noch speziell einer Aktivität.

- **Massiv parallele** Kommunikation
- Möglichst **sofort** reagieren, **schnelle Wechsel** zwischen Aktivitäten
- **Kurze Interaktionssequenzen**, dafür **häufigere Wechsel**
- **Apps** werden **kombiniert genutzt**, um ein Ziel zu erreichen / Problem zu lösen

Diese Eigenschaften des Alltags treffen insbesondere in der Geschäftswelt zu.

Private besteht bei vielen ein **soziales Bedürfnis**, **Informationen** zu **teilen** - sich mitzuteilen.

Technischer Wandel, Veraltung von Hardware, Vielfalt

Die **Weiterentwicklung** von Hard- und Software geht gerade im mobilen Bereich **rasend schnell**.

- **APIs veralten**
- **Neue Features** nutzen und **abwärtskompatibel** sein **Plattform-konform** und zeitgetreu bleiben

Es existiert eine Vielzahl an Geräten ...

- und eine **Vielzahl** an **Display-Formaten** und **-Auflösungen**
- mit **unterschiedlicher Leistung**

Worauf muss ich achten?

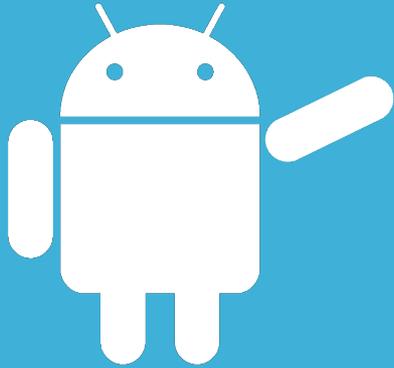
- Meine App sollte eine Fokussierung auf eine **Kernfunktion** besitzen, die sie besonders gut kann.
- Meine **Funktionalitäten** können **Bestandteil eines größeren Workflows** sein. Daher sollten Funktionalitäten und Daten **dem System bereitgestellt** werden.
- **Unterbrechung** ist inherent, meine App muss die **schnelle** und **unkomplizierte Abarbeitung** erlauben. Meine App muss den Nutzer **effizient Probleme lösen** lassen.



Worauf muss ich achten?

- Meine Moral: Die ***Daten gehören dem Nutzer!*** Er will sie teilen, und wiederverwendenden können.
- Mein Design muss für den ***Anwendungsfall optimiert*** sein und auf vielen ***unterschiedlichen Auflösungen*** und ***Formaten*** funktionieren.
- ***Plattform-konformität erleichtert*** dem Nutzer die ***Bedienung*** und erlaubt die ***Verschmelzung*** von ***Betriebssystem und App***. Stichwort UX.





Konzepte

*Design, Abwärtskompatibilität, Ressourcen-
Management, Activities, Fragmente*



Saxonia Systems
So geht Software.

Design

Das Rad neu erfinden - heute mal nicht :-)

- Designressourcen nutzen
- Design Guide und UI-Patterns nutzen

<http://developer.android.com/index.html>

<https://github.com/google/material-design-icons/releases/tag/1.0.0>

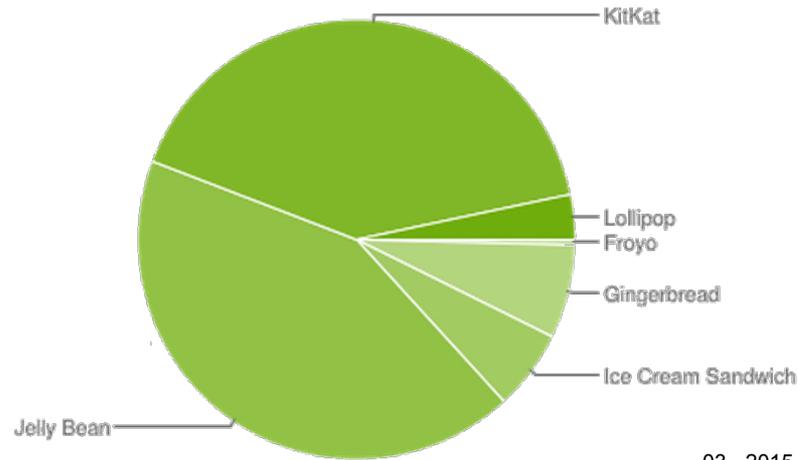
<http://app.colors.co/3f7cac-95afba-bdc4a7-d5e1a3-e2f89c>



Abwärtskompatibilität

Problem: Zwar ist die Aktualisierungsrate der APIs sehr hoch, aber nicht die der Geräte!

Version	Codename	API	Distribution
2.2	Froyo	8	0.4%
2.3.3 - 2.3.7	Gingerbread	10	6.9%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	5.9%
4.1.x	Jelly Bean	16	17.3%
4.2.x		17	19.4%
4.3		18	5.9%
4.4	KitKat	19	40.9%
5.0	Lollipop	21	3.3%



03 - 2015

Abwärtskompatibilität

Was tun?

- Google verteilt die Plattformupdates zeitiger an die Hersteller.
- Einführung der Play Services
- Verwendung der Support-Library

Support Library:

- View-Komponenten
- Systemkomponenten / -APIs



Ressourcen Management

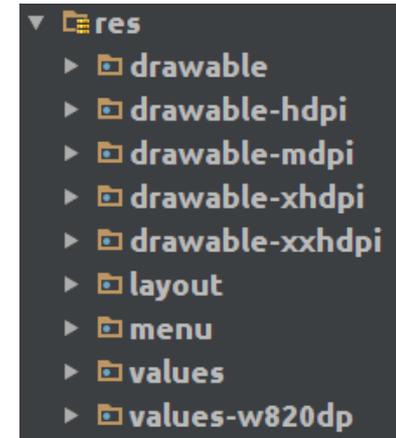
Unterschiedliche Auflösungen und Display-Formate müssen beim Design berücksichtigt werden.

Ressourcen, die in speziellen Verzeichnissen bereitgestellt werden können:

- Layouts / Views
- Grafiken
- Texte
- Größen
- Styles, ...

Qualifier für:

- Lokalisierung
- Auflösung
- Orientierung



App =

Android
Manifest.xml
(Metadaten)

+

Activities
(ViewController)

+

Views
(XML-Layouts)

+

Business-
Klassen
(Model, ...)



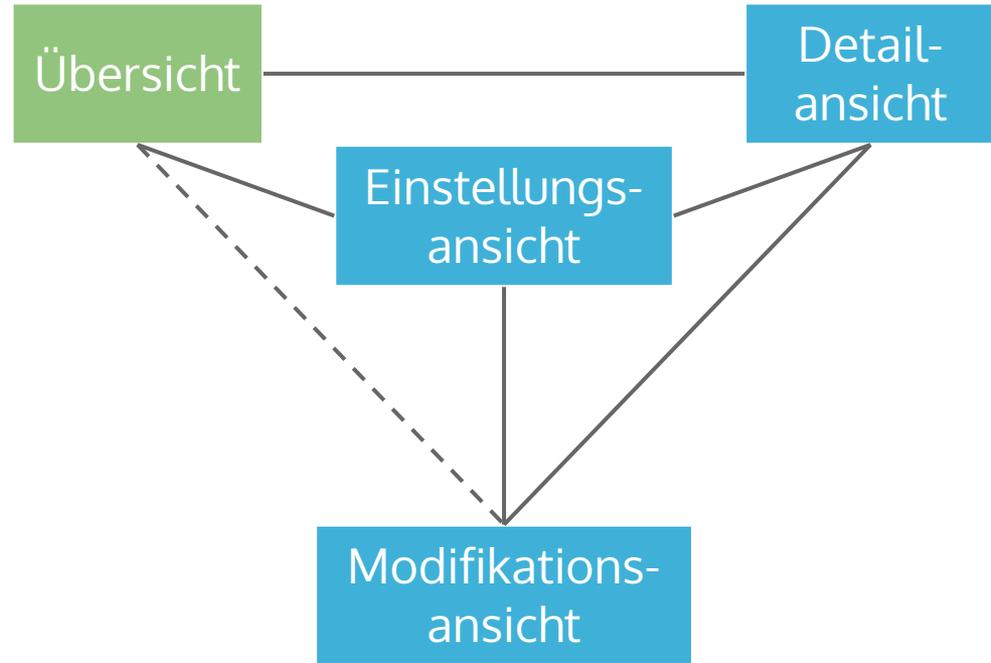
Activitys

Activitys sind ViewController, d.h. halb View, halb Cotroller

Ein fachlich abgeschlossener **Use-Case pro Activity**

Mehrere Activitys werden zu einem Workflow **orchestriert**.

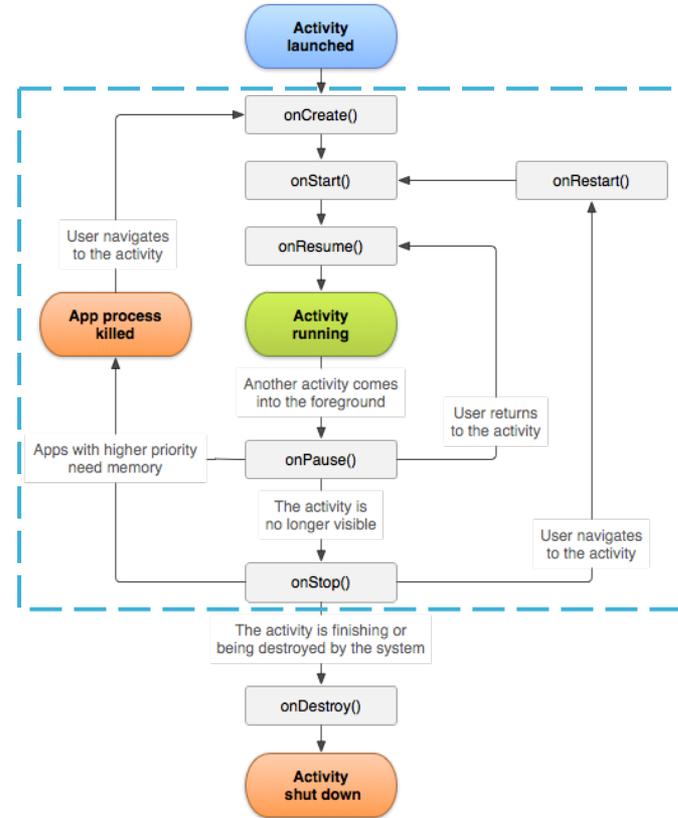
Activitys sollten **lose gekoppelt** sein und kommunizieren über **Intents**



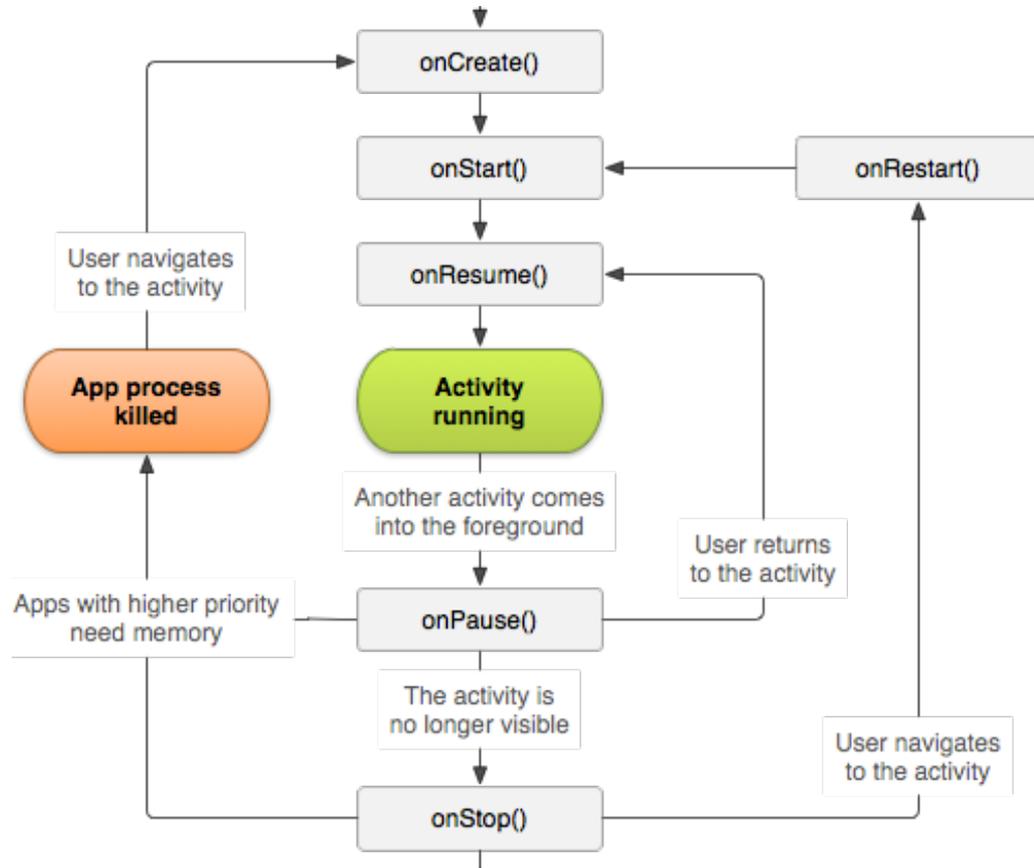
Activitys

... sind autark. D. h. sie verwalten ihren Zustand selbst

... haben einen Lifecycle, den man als Entwickler kennen lernen und beachten muss.

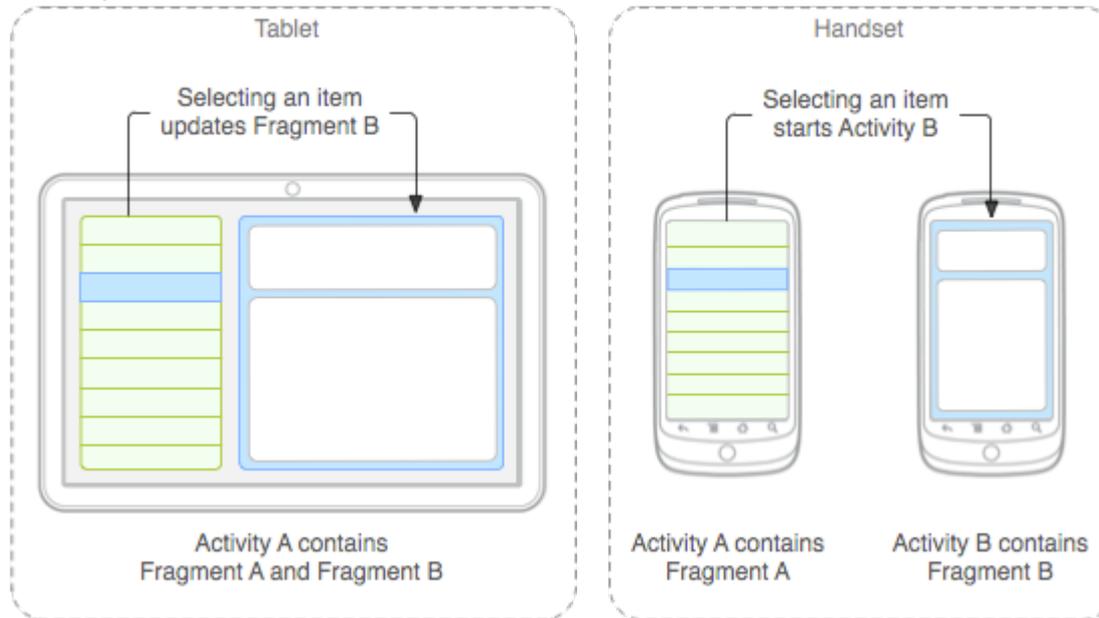


Activitys



Fragments

Für Fragments gilt dasselbe, wie für Activities - nur haben sie noch einen anderen und komplizierteren Lifecycle :-)





VORSTELLUNG DES PRAXISBEISPIELS



Saxonia Systems
So geht Software.

Mal wieder eine Notiz-App

Schritte

1. Vision
 - a. Nutzen, Mehrwert, Unterscheidungsmerkmale
 - b. Fokuspunkt
2. Sketching / Paper Prototyping / Mock-Ups erstellen
3. Implementierung und User-Tests
 - a. Top-Level-Ansichten
 - b. Navigation (Story Boards nutzen)
 - c. Layouting der Ansichten
 - d. Design, leichtes Branding
 - e. Eye-Candy: Animationen, Transitions, starkes Branding



Top-Level

Cat Ipsum
Scratch leg;
meow for ...



An der Strippe



Cat Ipsum
Stand in front
of the
computer
screen chase
mice, and
spread kitty ...

Cat Ipsum
Stand in front
of the
computer
screen chase
mice, and
spread kitty ...



Detail 1

← Cat Ipsum  

Scratch leg; meow for can opener to feed me jump launch to pounce upon little yarn mouse, bare fangs at toy run hide litter box until treats are fed, and bathe private parts with tongue then lick owner's face or if it fits, i sits. Stand in front of the computer screen chase mice, and spread kitty litter all over house. Rub face on everything jump launch to pounce upon little yarn mouse, bare fangs at toy run hide litter box until treats are fed jump off balcony, onto stranger's head. Kick up litter scamper.

Detail 2

← An der Stri ...  





CODING

...



Saxonia Systems
So geht Software.

Tooling und Bibliotheken

Android SDK***

Android Studio + Gradle / Eclipse + ADT Plugin + Maven***

Android Studio Plugin: [Android Drawable Import](#)

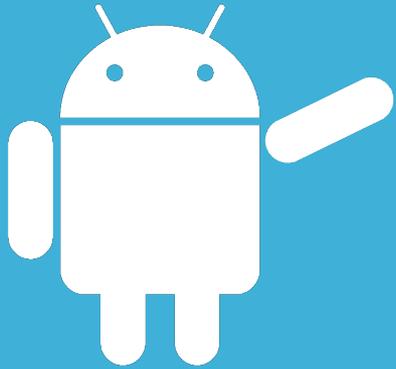
[Picasso](#), Lib für [Floating Action Buttons](#), [StaggeredGridView](#)



Android Konzept

- Lifecycle: besonders relevant z. B. bei Rotation (Recreation) und Unterbrechungen
- Intents zur Kommunikation zwischen Activities
- Custom Views
- Benennung von Activities / Fragments und ihren Layouts.





FAZIT

...



Saxonia Systems
So geht Software.

Referenzen

1. <http://developer.android.com/index.html>
2. <http://realm.io/news/joaquim-verges-making-falcon-pro-3/>
3. <http://romannurik.github.io/AndroidAssetStudio/>
4. <http://square.github.io/picasso/>
5. <http://android-arsenal.com/>
6. <http://colors.co/>
7. <https://github.com/winterDroid/android-drawable-importer-intellij-plugin>
8. <https://github.com/google/material-design-icons/releases/tag/1.0.0>



Kontakt

Max Wielsch

max.wielsch@saxsys.de

Der Code

<https://github.com/muxmax/JugGrNotesApp>

