

**Macoun**

# Wir sind die Prototypen

Marco Köppel und Marius Bulla

# Ablauf

- Wer sind wir?
- Wieso Prototypen und welche Arten gibt es?
- Die 7 Phasen des Prototypen
- Do the Prototype

Wer sind wir?

# Marco Köppel

razorfish<sup>TM</sup>

- UI Developer bei Razorfish.
- Schraubt intensiv an immer neuen Erfahrungen für den Nutzer.  
Meist beschäftigt mit Audi City und disruptiven Projekten wie dem Audi TT Brochure Hack.

# Marius Bulla

razorfish<sup>TM</sup>

- Tech. Director bei Razorfish.
- Seit 16 Jahren bei Razorfish. Meine Prototypen sprachen schon JS, AS, Processing, OF, ObjC und Python, ich ganz gut Deutsch und ein wenig Englisch.

Wieso Prototypen und  
welche Arten gibt es?

# Wieso Prototypen?

- Hilfskonstruktion um eine Fragestellung zu beantworten.
- Machen aus abstrakten Ideen greifbare Dinge.
- Gutes Verständnis bei Kreativen, Entwicklern und Kunden.



# Wieso Prototypen?

- *„Prototyping ist überlebenswichtig. Nicht nur um die Umsetzbarkeit, sondern insbesondere das gesamte Konzept zu überprüfen.“*
- Prototypen, so früh und so oft wie möglich bzw. nötig.
- Sind Werkzeuge, nicht die Lösung.

# Welche Arten gibt es?

- Dummies
- PoC (Proof of Concept)

# Dummies

- Motion
- Content
- Experience
- Interaction

# PoC

- Software
- Hardware
- Integration
- Construction

Was verhindert die Umsetzung  
einer Idee und wie können  
Prototypen helfen?

# #1: Zweifel

- Ist meine Idee wirklich gut?
- Prototypen liefern schnelle, valide Rückmeldung.
- Sie helfen der Idee, sich zu verbessern.

# #2: Fehlendes Können

- Ich muss erst Gestalten oder Programmieren beherrschen.
- Etwas zu können hilft, es nicht zu können sollte es nicht verhindern.
- Prototypen sind abstrakt und rudimentär.

# #3: Fehlende Sicherheit

- Das Risiko bzw. der Einsatz ist zu hoch.
- Prototypen verringern das Risiko eines Fehlschlags.
- Sind oft schnell und kostengünstig umzusetzen.



# #4: Mangel an (weiteren) Ideen

- Was kommt davor oder danach?
- Die erste Idee ist oft nicht die, bei der man endet.
- Neue Ideen entstehen, während man an und mit einer arbeitet.

# Die 7 Phasen des Prototypen



User  
Stories

Tasks

Archi-  
techure

Flow  
Diagram

Sketches

Proto-  
type

User  
Testing

# #1: User Stories

- Welche Bedürfnisse ergeben sich aus dem zu überprüfenden Konzept?
- Grobe Beschreibung der Anforderungen.
- Beschreibt was gemacht werden soll, nicht wie.

#1

Tasks

Archi-  
techure

Flow  
Diagram

Sketches

Proto-  
type

User  
Testing

# #1: User Stories

As an app i want to generate and show Macoun presentations of a given topic.

#1

Tasks

Archi-  
techure

Flow  
Diagram

Sketches

Proto-  
type

User  
Testing

# #2: Tasks

- Welche Schritte sind nötig um das Ziel zu erreichen?
- Jedes Verb in der User Story stellt einen Task dar.

User  
Stories

#2

Archi-  
techure

Flow  
Diagram

Sketches

Proto-  
type

User  
Testing

# #2: Tasks

As an app i want to generate and show Macoun presentations of a given topic.

User  
Stories

#2

Archi-  
techure

Flow  
Diagram

Sketches

Proto-  
type

User  
Testing

# #2: Tasks

As an app i want to generate and show Macoun presentations of a given topic.

generate p.

show p.

give topic

User  
Stories

#2

Archi-  
techure

Flow  
Diagram

Sketches

Proto-  
type

User  
Testing



# #3: Architecture

- In welchen und wie vielen Schritten werden die Tasks ausgeführt?
- Nicht jeder Task ist ein Schritt. Sinnvoll kombinieren.
- Eine gute User Experience entwickelt sich iterativ und ist nicht beim ersten Versuch perfekt.

User  
Stories

Tasks

#3

Flow  
Diagram

Sketches

Proto-  
type

User  
Testing

# #3: Architecture

As an app i want to generate and show Macoun presentations of a given topic.

generate p.

show p.

give topic

User  
Stories

Tasks

#3

Flow  
Diagram

Sketches

Proto-  
type

User  
Testing

# #3: Architecture

generate p.

show p.

give topic

User  
Stories

Tasks

#3

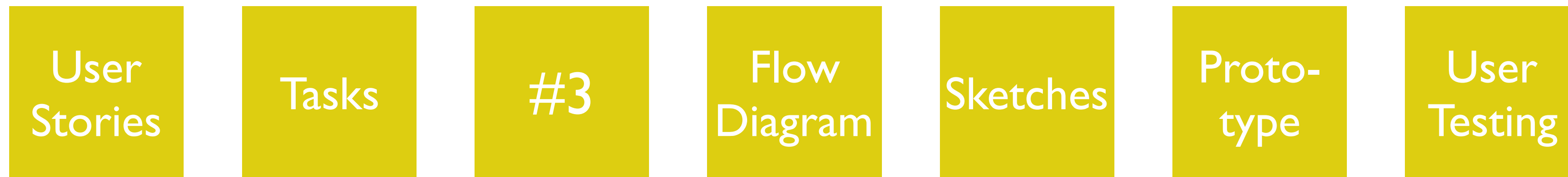
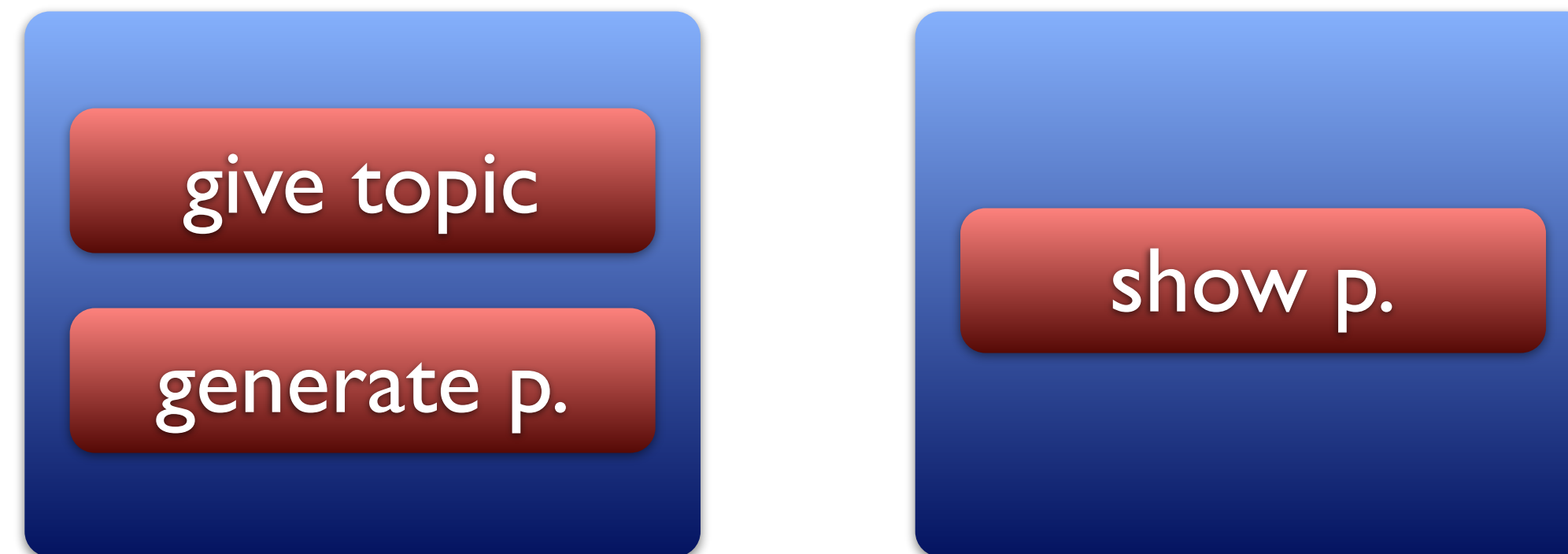
Flow  
Diagram

Sketches

Proto-  
type

User  
Testing

# #3: Architecture



# #4: Flow Diagram

- Wie sind die Schritte miteinander verbunden?
- Die Definition der User Journey.

User  
Stories

Tasks

Archi-  
techure

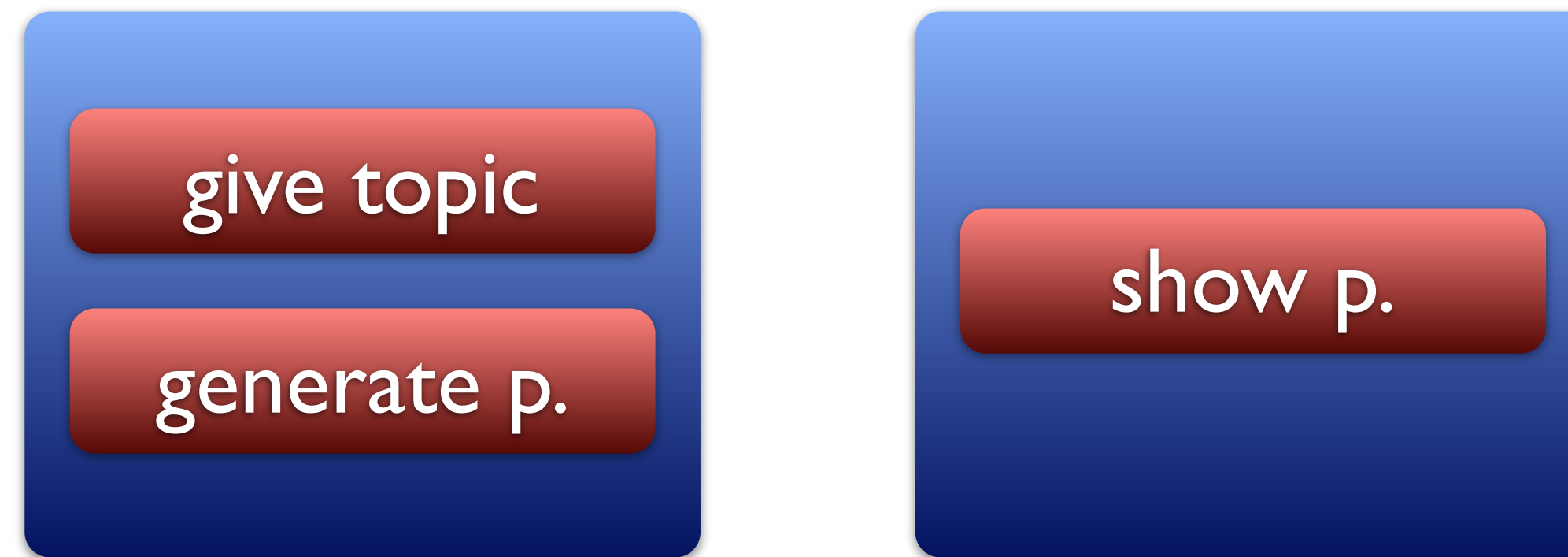
#4

Sketches

Proto-  
type

User  
Testing

# #4: Flow Diagram



User  
Stories

Tasks

Archi-  
techure

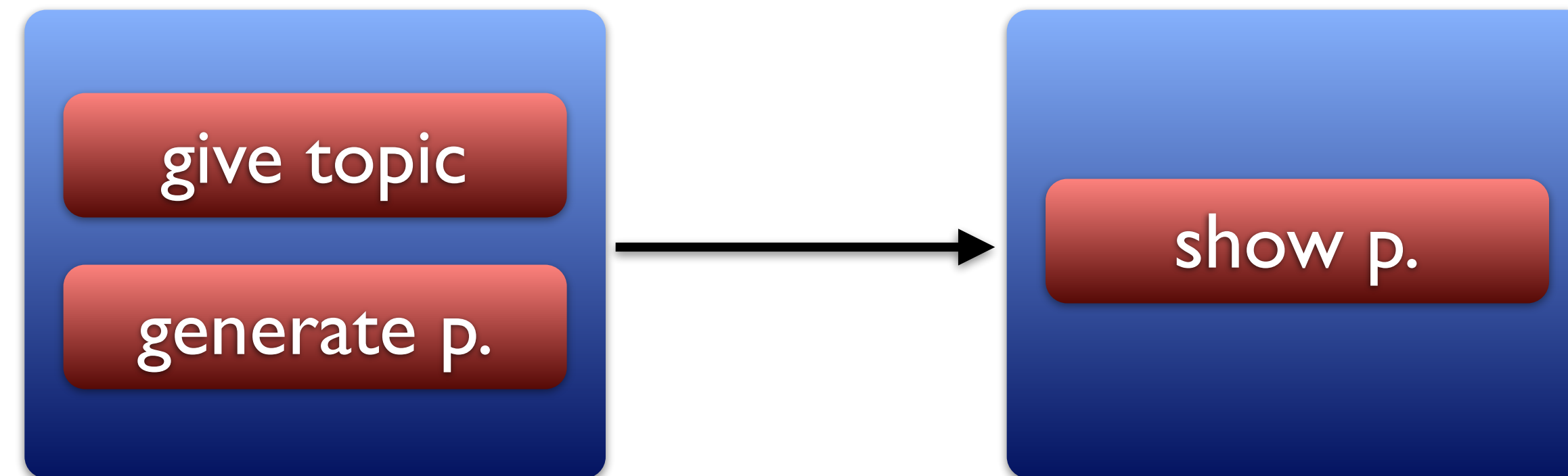
#4

Sketches

Proto-  
type

User  
Testing

# #4: Flow Diagram



User  
Stories

Tasks

Archi-  
techure

#4

Sketches

Proto-  
type

User  
Testing

# #5: Sketches

- Grobe Gestaltung der einzelnen Screen-Inhalte.
- Rudimentäres Blockdesign.
- Je detaillierter, desto weniger offen ist man für Anpassungen und Experimente.

User  
Stories

Tasks

Archi-  
techure

Flow  
Diagram

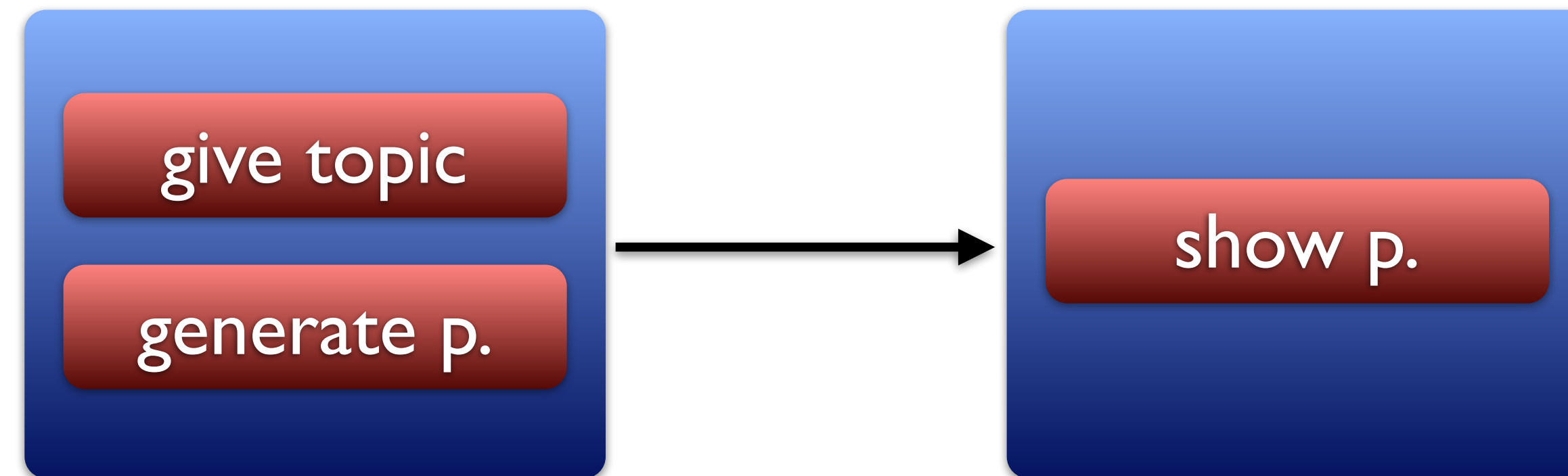
#5

Proto-  
type

User  
Testing



# #5: Sketches



User  
Stories

Tasks

Archi-  
techure

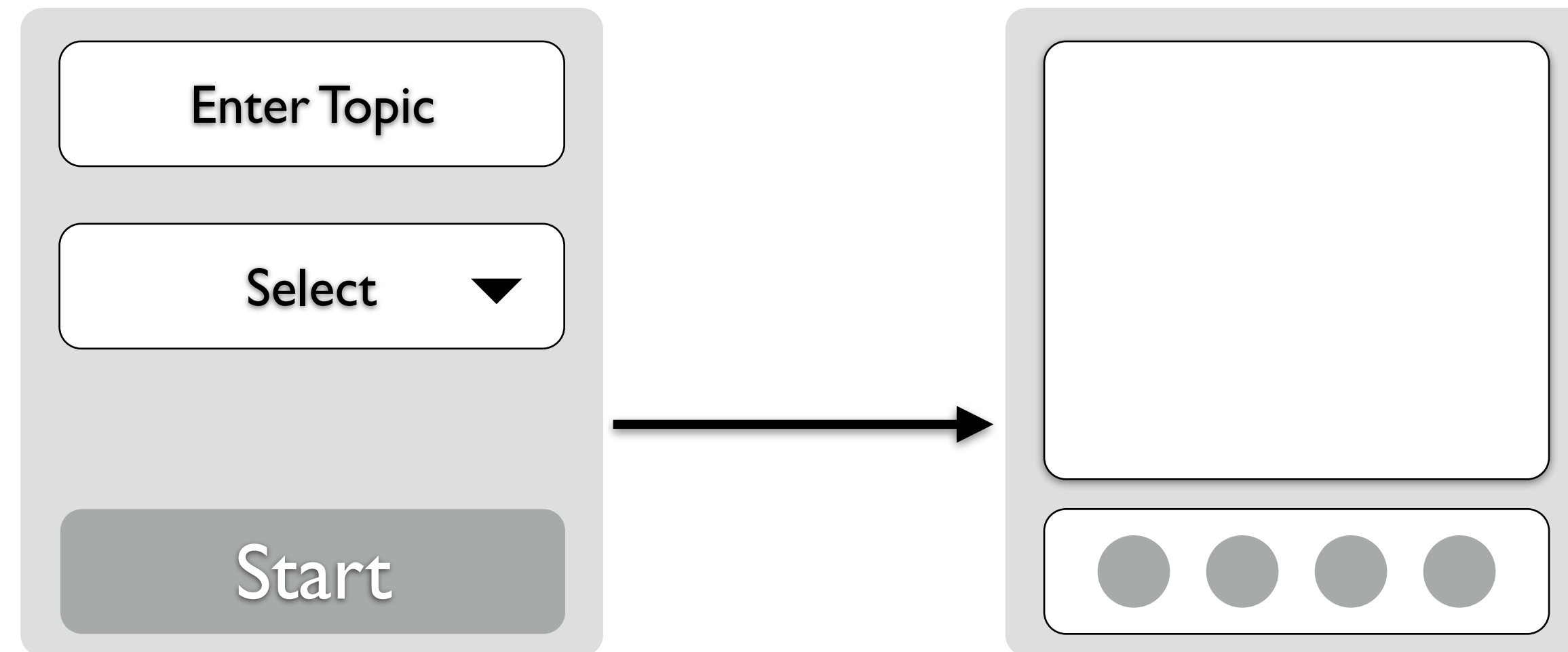
Flow  
Diagram

#5

Proto-  
type

User  
Testing

# #5: Sketches



User  
Stories

Tasks

Archi-  
techure

Flow  
Diagram

#5

Proto-  
type

User  
Testing

# #6: Prototype

- Basis des Prototypen aufbauen.
- Marcos Tipps befolgen.

User  
Stories

Tasks

Archi-  
techure

Flow  
Diagram

Sketches

#6

User  
Testing

# #7: User Testing

- Nutzerverhalten beobachten und Rückmeldungen einholen.
- Im Zweifel Varianten anbieten, aber den Testumfang begrenzen.
- Tester bitten, laut zu denken.

User  
Stories

Tasks

Archi-  
techure

Flow  
Diagram

Sketches

Proto-  
type

#7

# #7: User Testing

- Keine Freunde oder Verwandten, man benötigt ehrliche Rückmeldung.
- Der erste Prototyp wird nicht der Beste sein.
- Mit den Ergebnissen iterativ den Prototypen anpassen.

User  
Stories

Tasks

Archi-  
techure

Flow  
Diagram

Sketches

Proto-  
type

#7

# #7: User Testing

User  
Stories

Tasks

Archi-  
techure

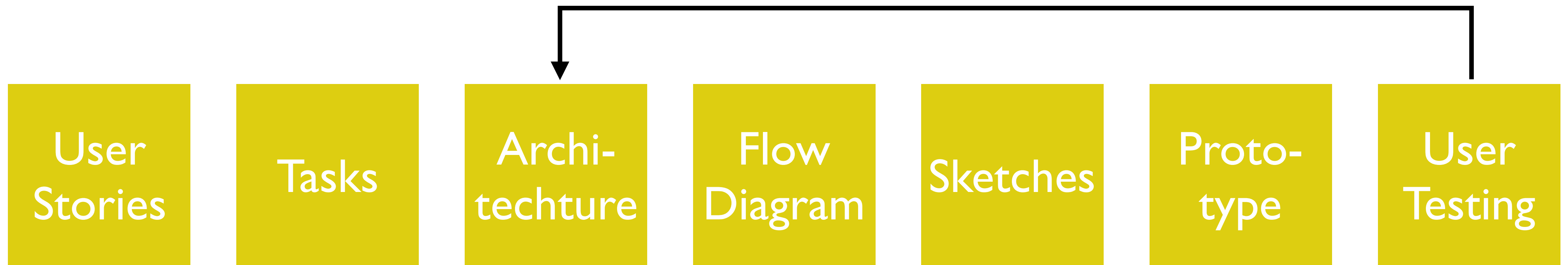
Flow  
Diagram

Sketches

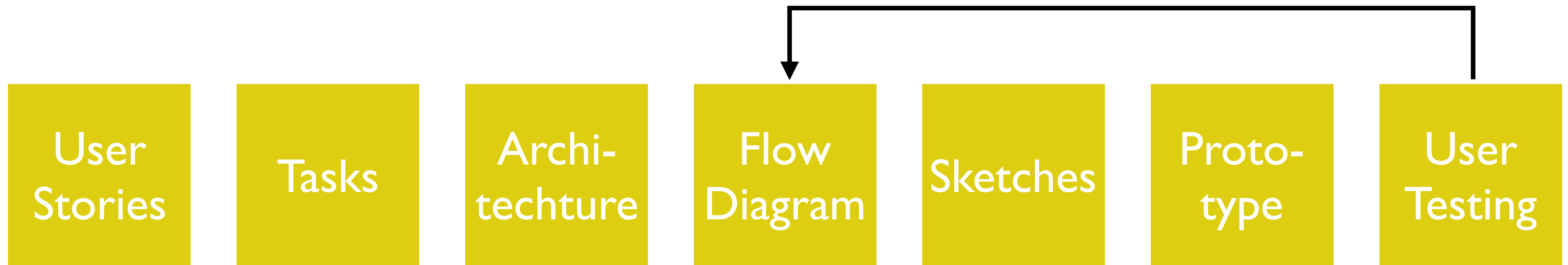
Proto-  
type

User  
Testing

# #7: User Testing

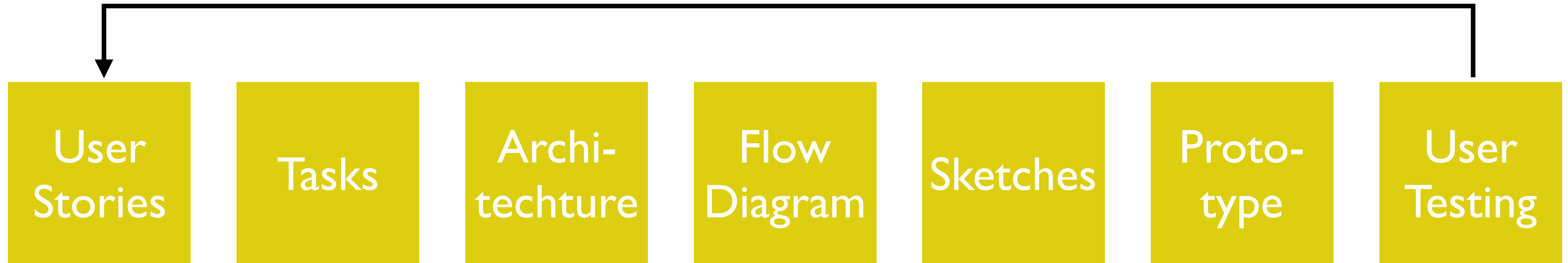


# #7: User Testing

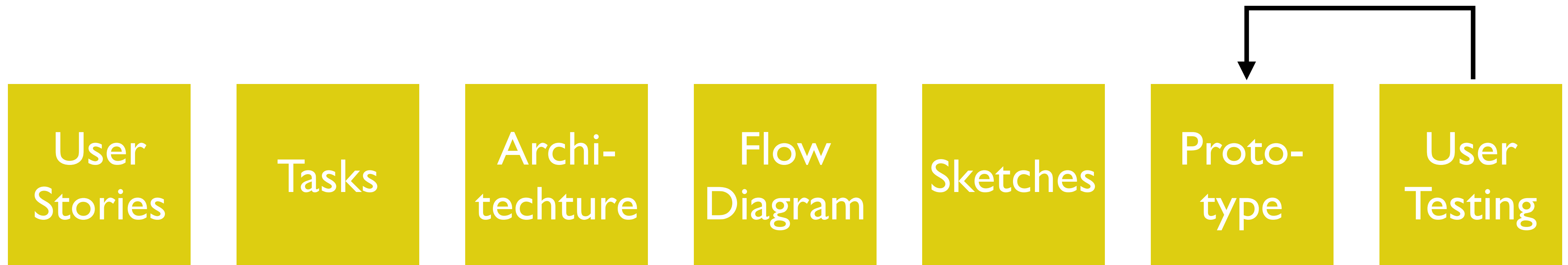




# #7: User Testing



# #7: User Testing

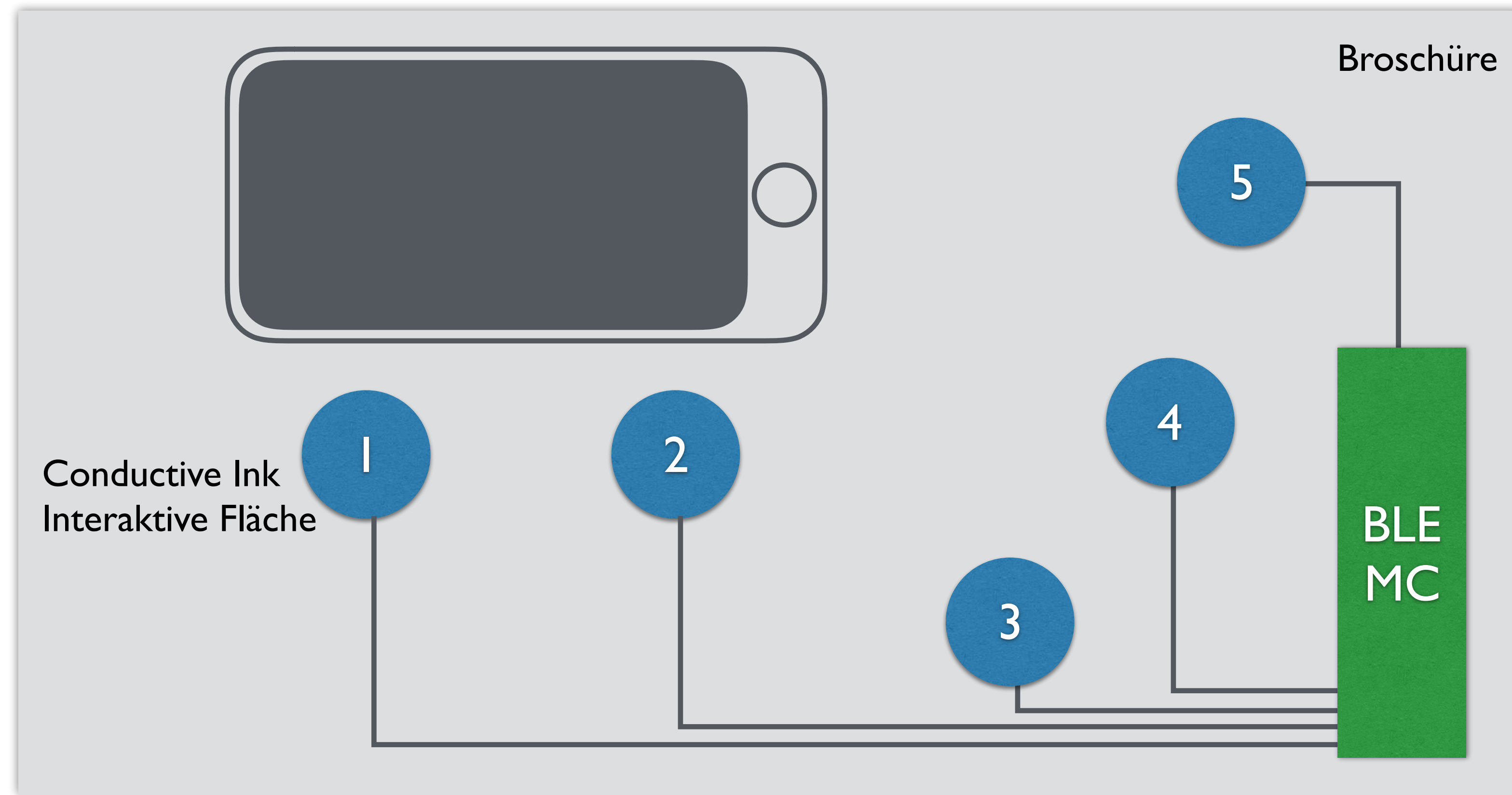


# Bau den Prototypen

# Das Beispiel

- Interaktive Werbebroschüre
- Conductive Ink - Novalia (UK)
- In voller medialer Herrlichkeit: <http://bit.ly/1LO4H6E>

# Das Beispiel



# Die Ziele / Die Fragen

1. Können wir die BLE-Signale nutzen und Videos abspielen?
2. Macht das Spaß? Sieht das dann auch gut aus?
3. Können wir das in ein abgerundetes Szenario verpacken?

# Die Alternative

## The Deskboard

- Conductive Ink auf einem bedruckten Karton zur Steuerung einer App

Demo



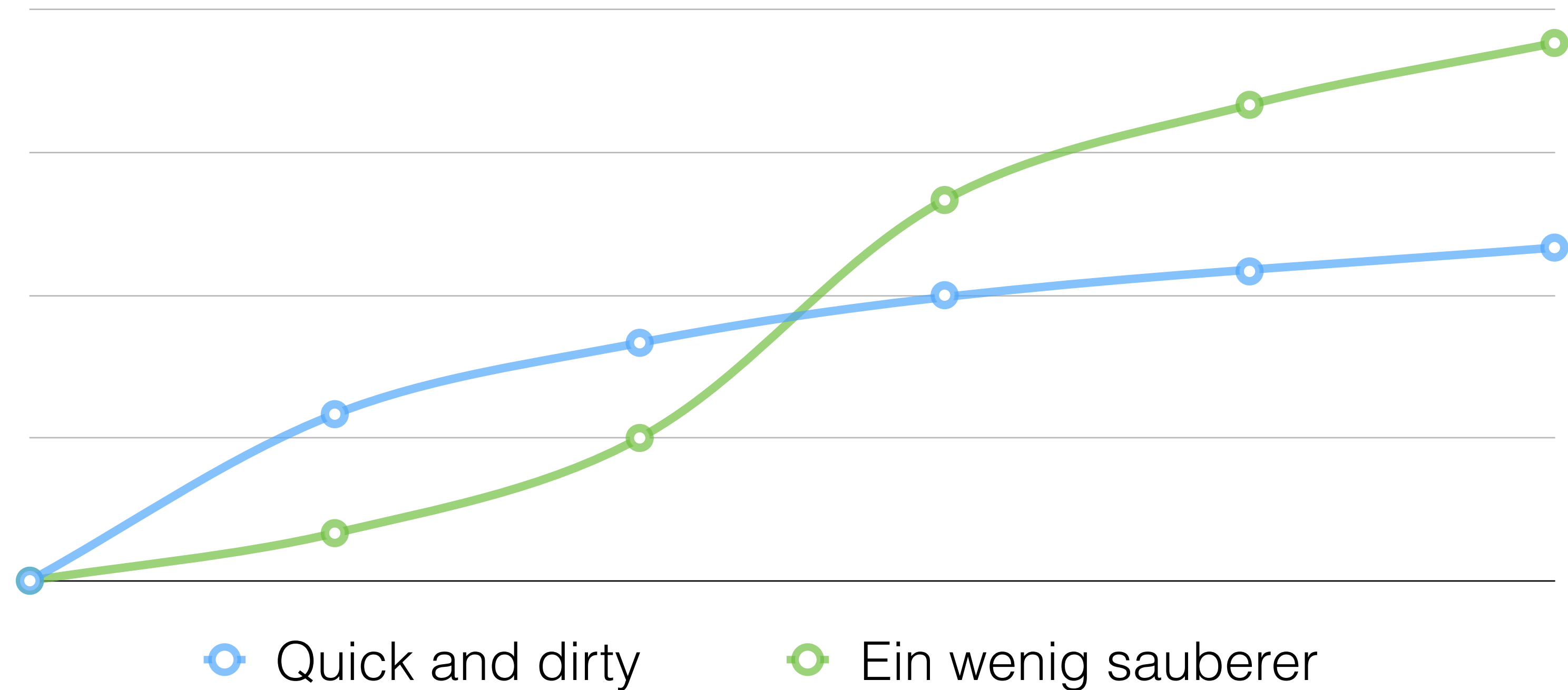
# Die 7 großen Regeln

- Regeln sind auf Interaction-Prototypen zugeschnitten
- Eher als Richtlinien zu verstehen

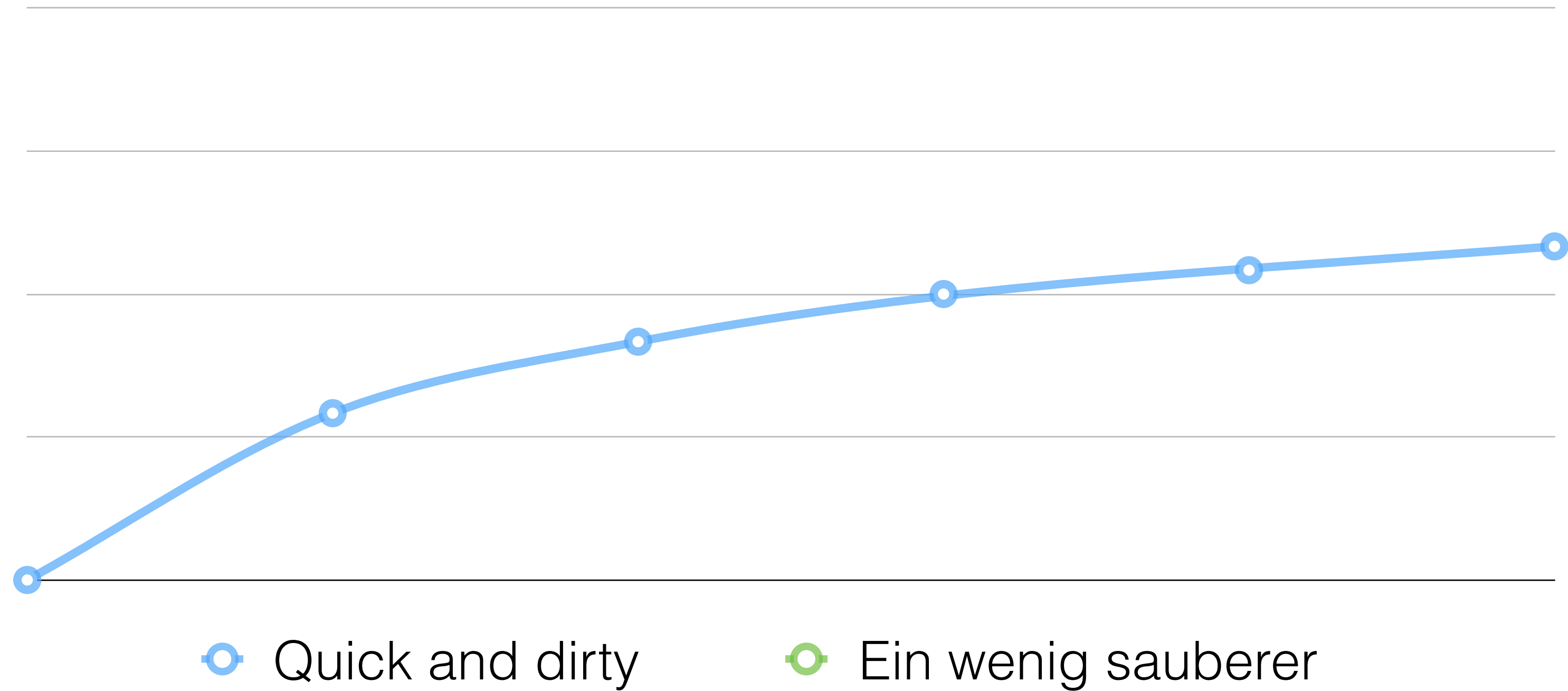
# #I: Aufwärmen

- Einfache Architektur
- Gedanken machen über mögliche Iterationen / Variationen
- Dennoch: Übertreibe es nicht

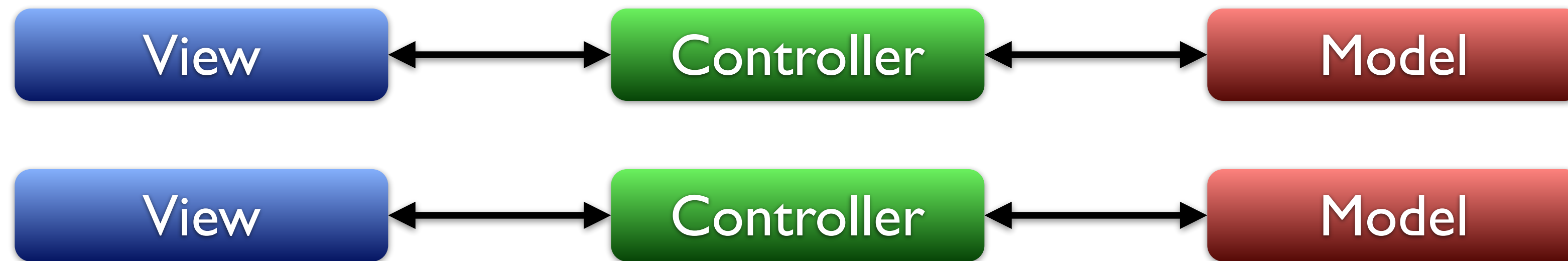
# #1: Aufwärmen



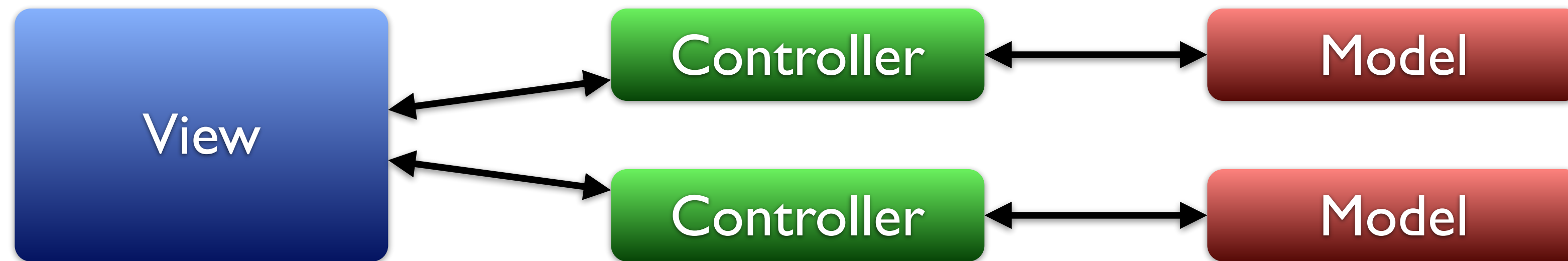
# #1: Aufwärmen



# #1: Aufwärmen



# #1: Aufwärmen



# #2: Reise leicht

- Mock it 'til you make it
- Mocking Frameworks tendentiell übertrieben
- Hardcoded ist vollkommen ok
- Best Case Szenario

# #2: Reise leicht

```
func updateWeather(updated: (weather : Weather) -> ()) {  
    //TODO: Fetch Weather from Weather Service.  
    updated(weather: Weather(c: "Sonnig", t: 17, i: UIImage(named: "sonne.png")!));  
}
```



# #3: Hol dir Hilfe

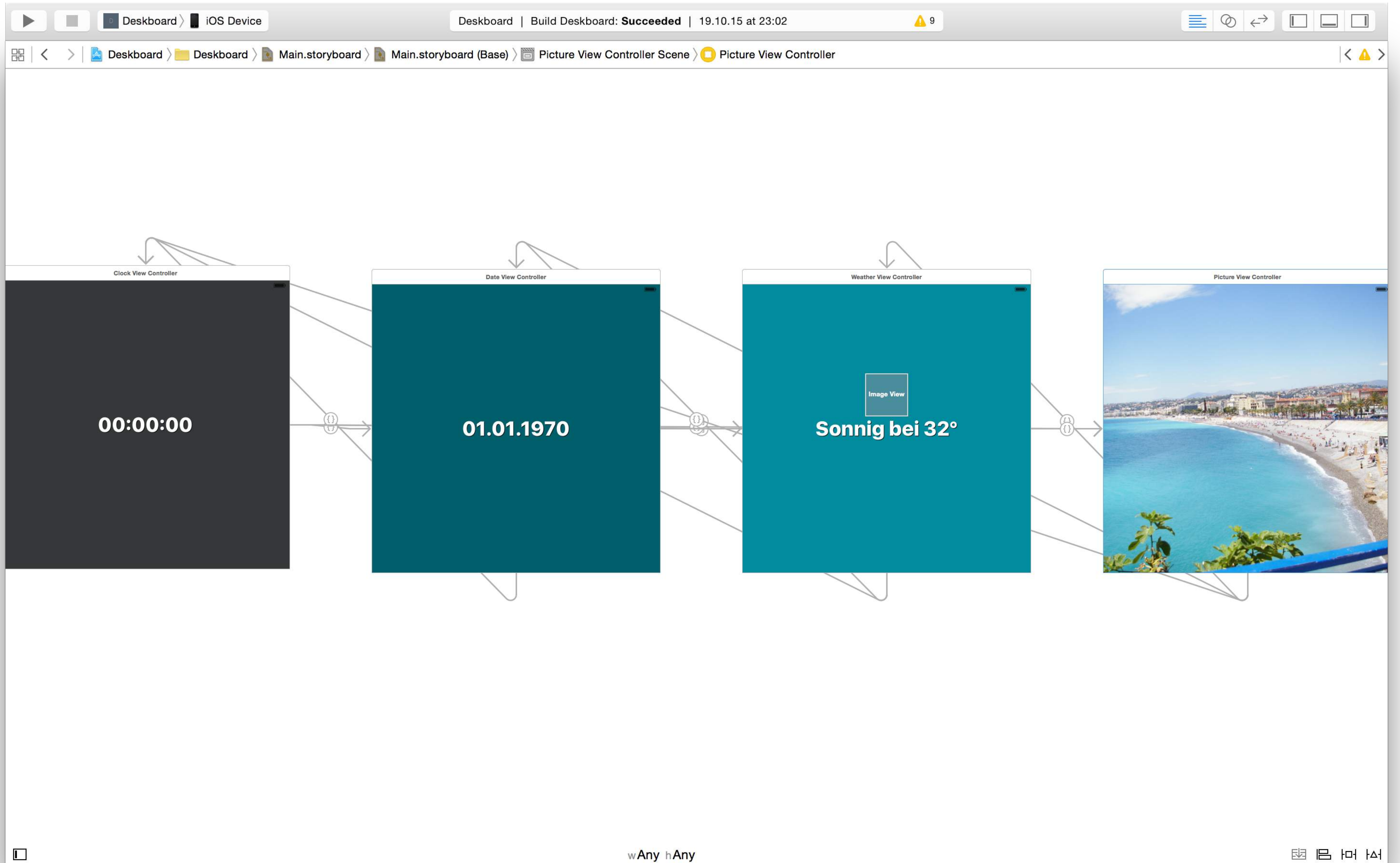
- Finde Partner
- Nutze Libraries

# #3: Hol dir Hilfe

- Müssen nicht perfekt sein
- Bevorzuge einfache, austauschbare Libraries
- Bevorzuge Libraries, die du schon kennst oder kennen lernen möchtest

# #3: Hol dir Hilfe

- No-brainer: Benutze Cocoapods
- No-brainer: Benutze UIKit



# #4: Konzentration

- Das Ziel des Prototypen hat immer Vorrang
- Funktionalitäten, die nur zur Vervollständigung dienen, aufs Nötigste beschränken (z.B. Fotogalerie, Custom Controls)
- Schiebe aufwendige Animationen auf

Demo

# #5: Versionieren

- Versionierung ist immer wichtig
- Häufige Commits
- Vorsicht mit „Magic Numbers“ und unausgereiften Strukturen

# #6: Lerne

- Evaluiere und verstehe die Ergebnisse des Prototypen
- Baue neue Iterationen, um den Prototypen weiterzuentwickeln



# #6: Lerne

- Evaluiere Designpatterns und technische Konzepte
- Evaluiere Libraries

# #6: Lerne

- Wichtig für die Implementierung des eigentlichen Produkts
- Wichtig für den nächsten Prototypen

# #7: Do not merge!

- Prototyp != Produkt
- Völlig unterschiedliche Zielsetzungen
- Unterschiedliche Erwartungen an Qualität und Stabilität

# #7: Do not merge!

- Lieber neu implementieren und alles, was man gelernt hat, einfließen lassen
- Code kann trotzdem wiederverwendet werden

# #7: Do not merge!

- „Aber der Prototyp ist doch ...“ >> NEIN!
- „Aber der Kunde will ...“ >> NEIN

# #7: Do not merge!

Spoiler:

- Aber am Ende ist der Kunde der König

# Links

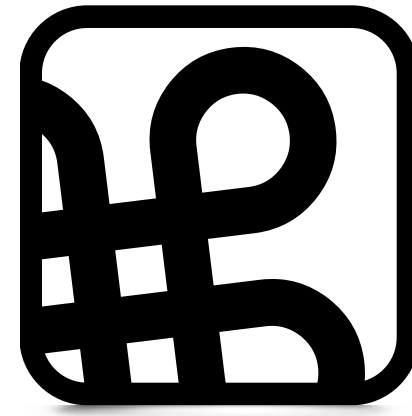
# Links

- ‚Fake It Make It‘ von Amir Khella
- ‚Prototyping‘ von Todd Zaki Warfel
- ‚Audi TT Brochure Hack‘ - [bit.ly/1LO4H6E](https://bit.ly/1LO4H6E)



Fragen?

**Vielen Dank**



**Macoun**