

Web Clients

Prof. D. König

*"You cannot think your way
to experience."*

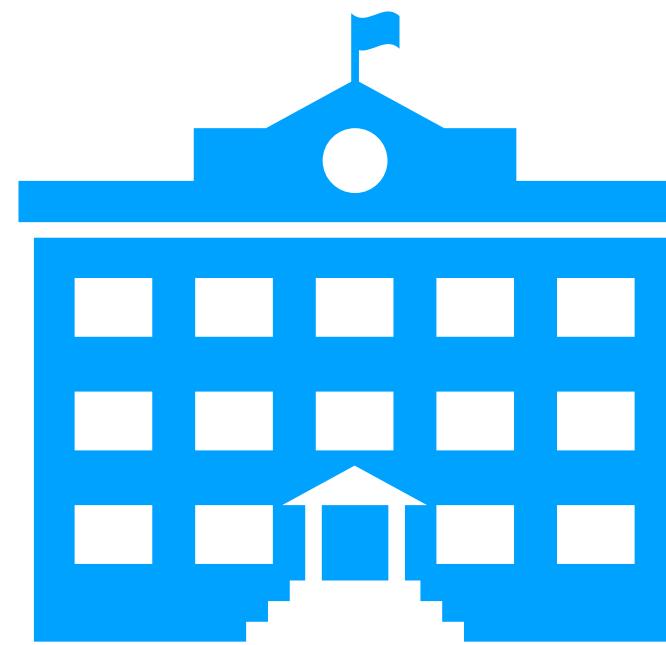
Kent Beck

Teams

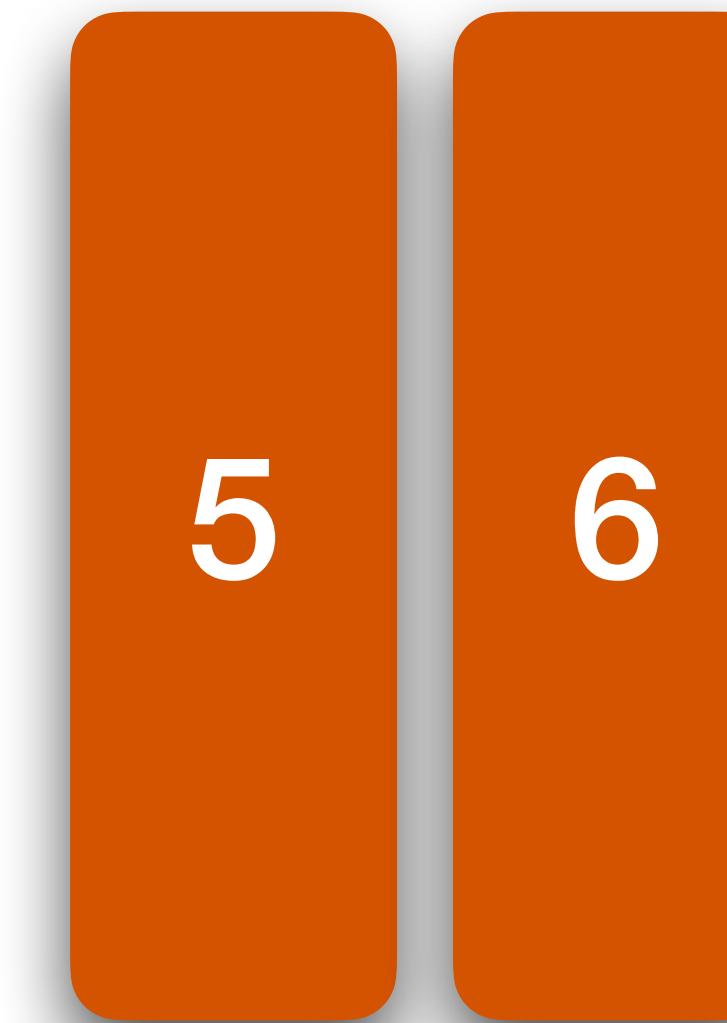
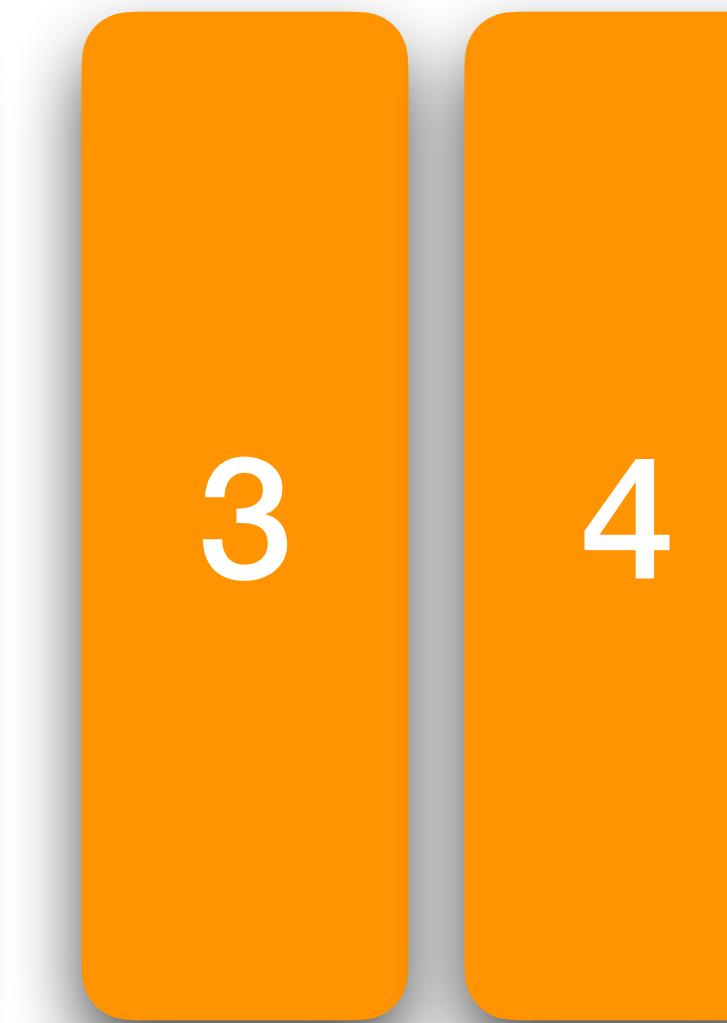
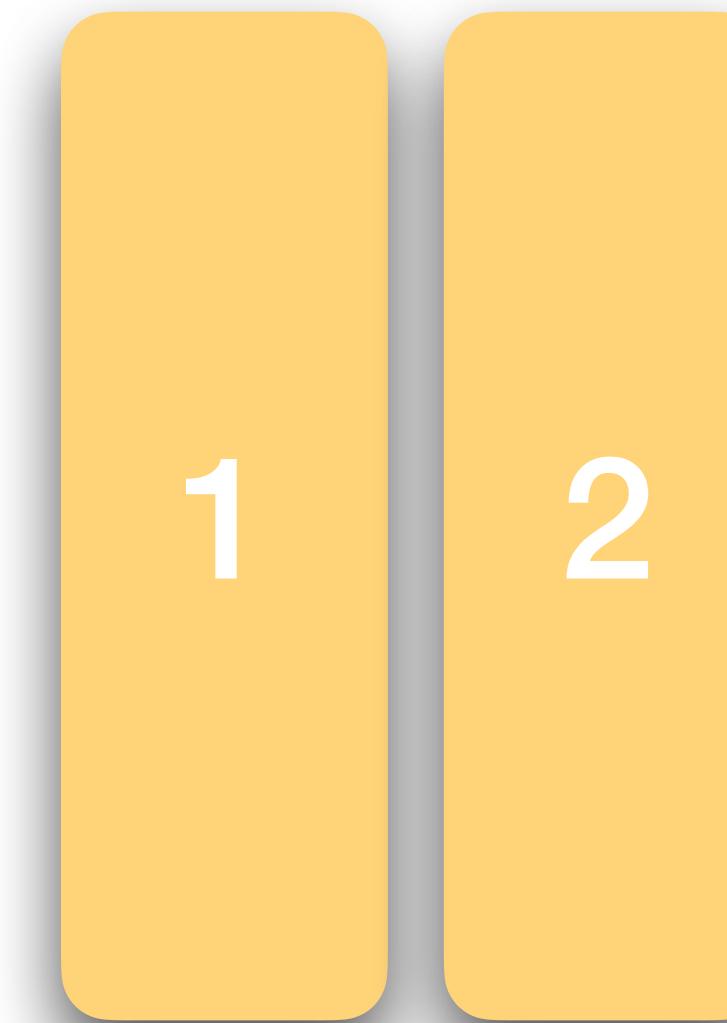
Wofür wir MS Teams nutzen

Sollen wir auch in Zukunft
aufzeichnen?

Overview



Schule



Beruf

Eigenverantwortung

Web Modules

Workshop Web *

Web Clients

Web Programming

Web Engineering

Code Kitchen

Web Frameworks

Ambitionierte Web UIs

Effizient

Reichhaltig

Nachhaltig

file:///projects/git/fhnw/fs19-wodss-client/ProjectAllocation

DuckDuckGo Startpage.com...ne der Welt. Beliebt Chatter Hoogle Category: Th...amming Cafe Hulbee Froogle Home | grails...munity Slack > +

Project Allocation, drag-and-drop view

Begin 21 5 2019

End 21 5 2019

+ +

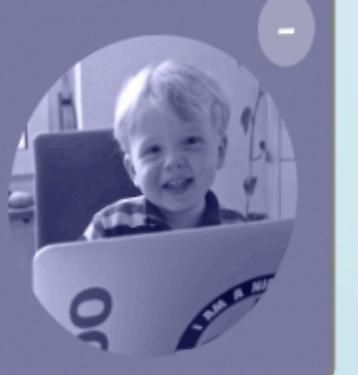
First Name: **Monika**
Last Name: **Mustermann**
Works %: **80**
Load:  30 %



First Name: **Christian**
Last Name: **Ribeaud**
Works %: **100**
Load:  100 %



First Name: **Florian**
Last Name: **König**
Works %: **20**
Load:  20 %



WebClients needs **1** FTE, has 1 FTE
assigned. Open: 0
X 100 Christian Ribeaud

WebProg needs **0.5** FTE, has 0.5 FTE
assigned. Open: 0
X 30 Monika Mustermann **X 20** Florian König

WebEngineering needs **1** FTE, has 0 FTE
assigned. Open: 1

Laufende Themen

Sicht der Informatik

Technologie-übergreifend

Web for fun and profit

Didaktischer Ablauf

Rückschau, Fragen, Thema

Live-Coding

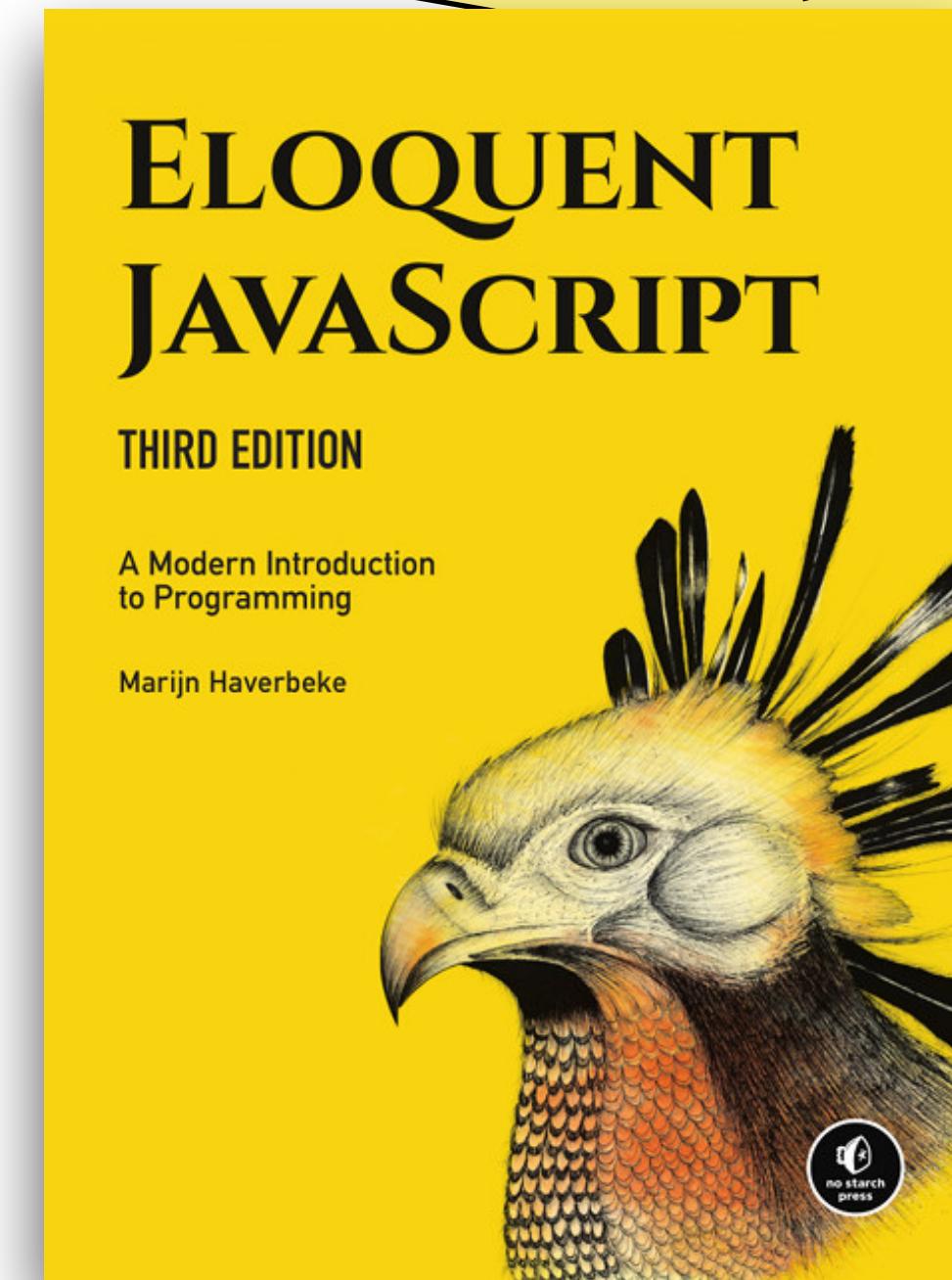
Übungen

Spezialaufgaben

Recommended Reading

*We will use
JavaScript2015, ES6*

Script/Lehrbuch
Eloquent JavaScript
You don't know JS



Bewertung

Erfahrungsnote
bewertet die Erfahrung &
kontinuierliche Arbeitsleistung

Punkte sammeln

Während jeder Vorlesung

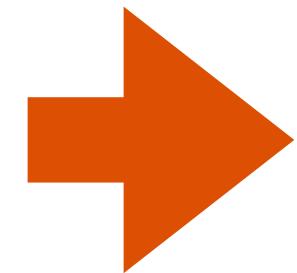
Beiträge, Aufgaben,
Präsentation

voraussichtlich 50 Punkte
60% notwendig zum bestehen

Punkte sammeln

Selbst die Punkte in Teams
"Notizen" eintragen

Drehbuch (initial)



1 Drehbuch, Intro, Classic MVC

2 Validation, Conversion, Attribute

3 Attribute Projector

4 Master-Detail

5 Projector Pattern

6 CSS++

7 SVG and Canvas

8 Animation

9 Custom Elements, PEP Aufbau

10 Big_UI_FW(TM) Integration

11 Connecting to REST

12 Service Abstraction

13 Reactive, Stable Binding

14 Applied Concepts

15 Team App: Putting it all together

UI Architecture

Custom UI
Elements

Remoting

Ansatz

Man versteht nur, was man im Prinzip selber bauen kann.

=> keine Abhängigkeiten

Anschluss an WebPr

Classic MVC

Testing, Modules, Varia, ...

<https://github.com/WebEngineering-FHNW/webcl-fs23.git>

"Hello, World!" der UI FWs



Klarheit schaffen

Klare Zuständigkeiten: "Was" vs "Wie"

Klare Abhängigkeiten: Wer kennt wen?

Klare Abfolgen: Was passiert wann?

Klare Regeln und Konzepte.

Observable (v.1.0)

```
const Observable = value => {
  const listeners = [];
  return {
    onChange: callback => listeners.push(callback),
    getValue: () => value,
    setValue: val => {
      if (value === val) return;
      value = val;
      listeners.forEach(notify => notify(val));
    }
  };
}
```

ordering ↗ ↘

many

protection

Klarheit schaffen

- A) einzelnes Todo ändert sich
- B) die Liste der Todos ändert sich

=> observierbare Liste

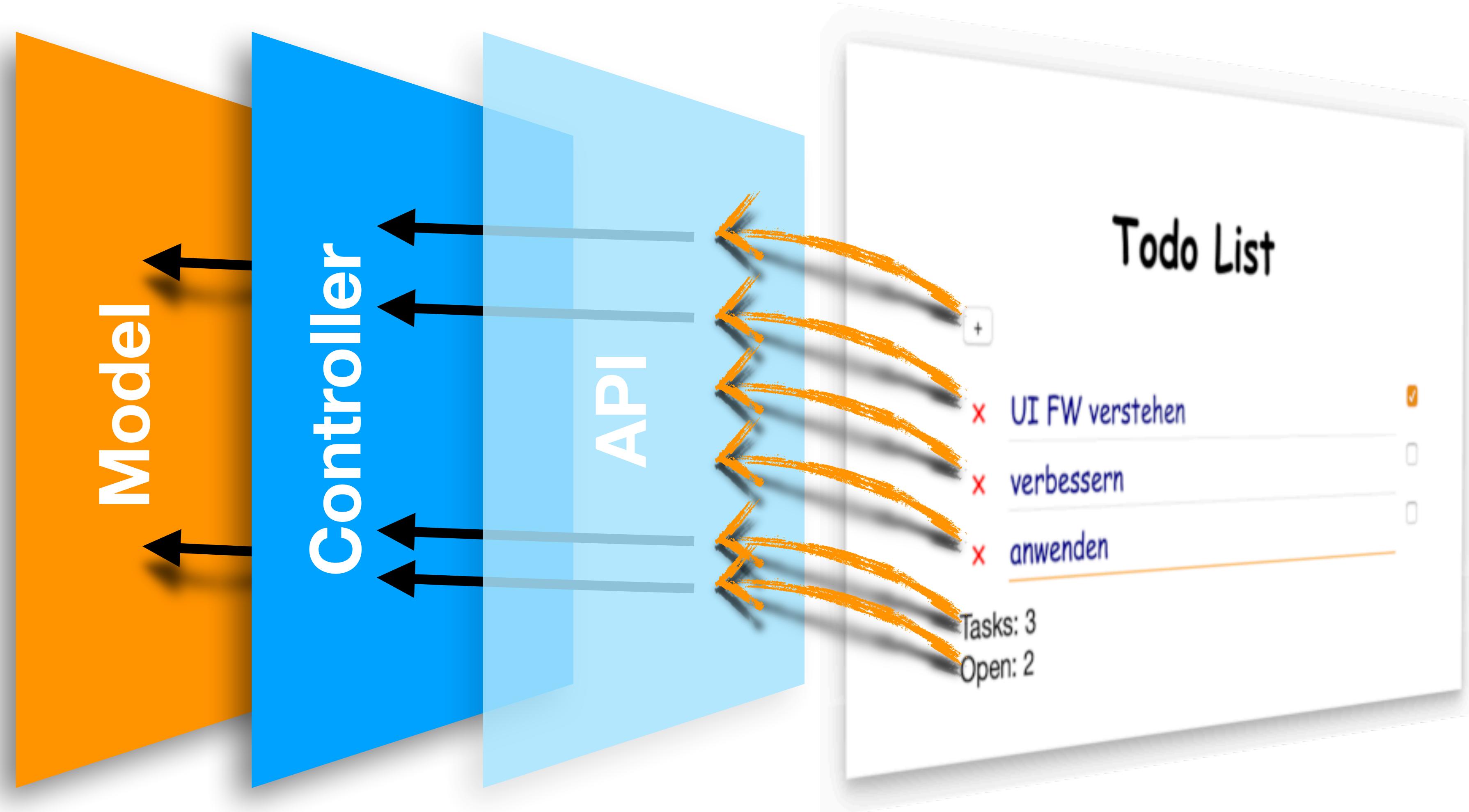
Ziel 1

View, Controller und Modell
vollständig voneinander getrennt

Ziel 2

Alle Views
vollständig voneinander getrennt

MVC, classic version



Erweiterungsrichtungen

- 👍 Observer Pattern: Observable Values
 - 👍 Observable Collections (Array, List, Set)
- Binding, Validation, Conversion
- Asynchronous Data Flow (e.g. REST)

Aufgabe

Wie kann man Validation und Konvertierungen einbauen?