# Web Programming

## Week 13

"Frameworks and APIs change fast. Software design principles are evergreen. Learn principles that translate across language barriers."

Eric Elliot

# Storybook (initial)

Drehbuch, Intro, Functions
Scientific foundations
Algebraic Data Types, Snake
Applied Science

Scripting, PWA, Plotter, Excel
Objects
Classes
JS Types, JsDoc

Async Programming
Modules
Data Flow, Excel improved
Iterator Protocol, Sequences

Programming Moves
User Interfaces
Professional JS with Kolibri
Crazy JS

# Agenda

Observable and MVC in preparation of Web Clients

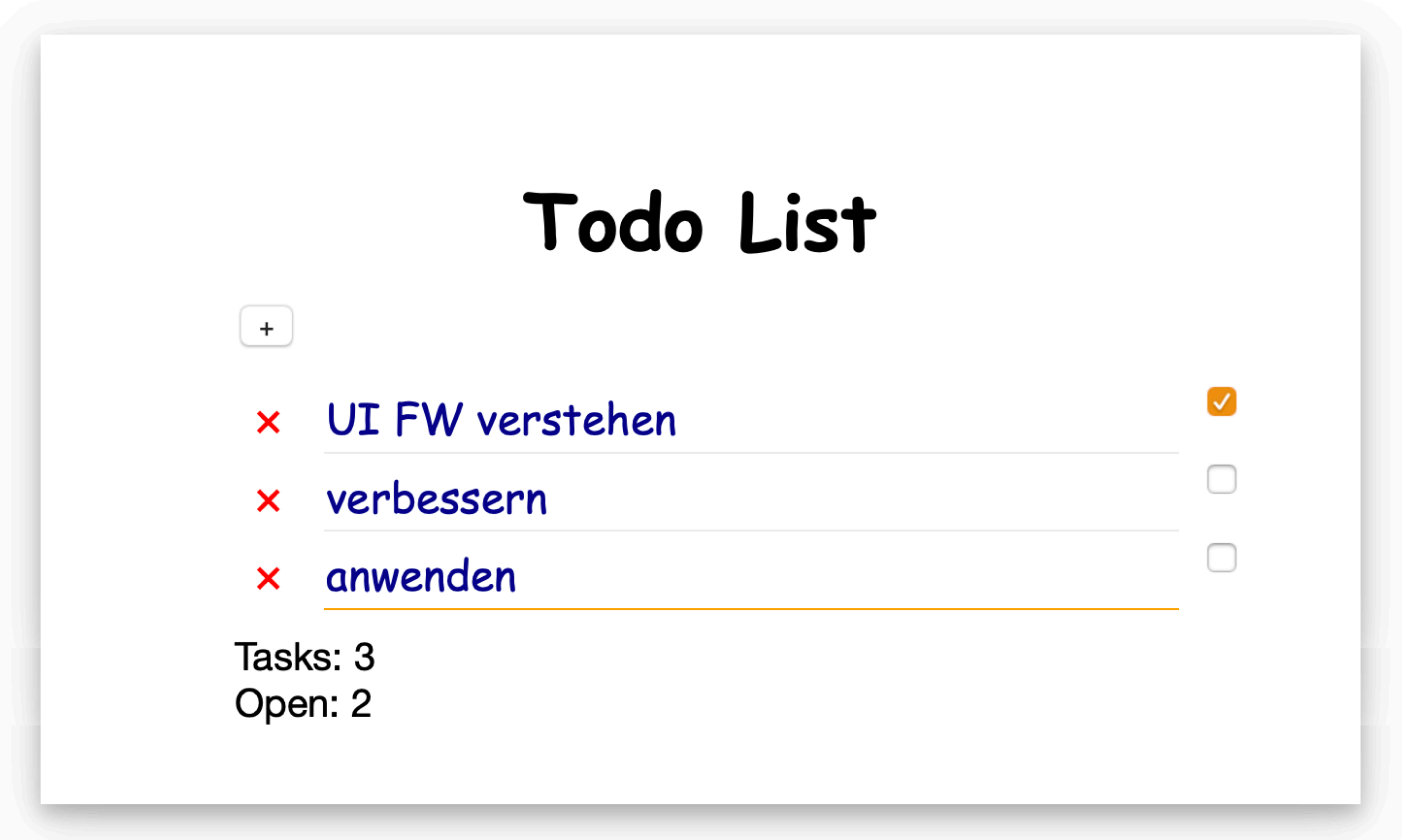
Quiz

# UI "Framework"

FW usage requires FW insight:
  what it does,
  how it does it,
  why it does it this way.

" I could build this myself! "

# "Hello, World!" of UI FWs



## Todo List

- ✗ UI FW verstehen ☑
- ✗ verbessern □
- ✗ anwenden □

Tasks: 3
Open: 2

# Start "explore"

Move 1: start at the end

Tests are red

The simplest solution that could possibly work™

# Milestone 0

It runs

Reasonable amount of tests

All tests ok

# Red-Green-Refactor

Move: Reorganisation

Separate the un-essentials

Improve clarity (= abstraction)

# Improve clarity

Responsibilities: "What" vs "How"

Dependencies: Who knows whom?

Sequence: What happens when?

Clear and simple rules.

# 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));
        }
    }
};
```

many

protection

ordering

# Observable Topics++

Observer modifies Observable

"Bindstorm"

Less obvious: A -> B -> C -> A

One or more Observers?

Callback at registration? Remove?

# Milestone 2

Separate Todo operations
from Todo display

All tests ok

# Improve Clarity

A) single Todo changes

B) the List of Todos changes

=> observable list

# Milestone 3

View, Controller, and Model are separated

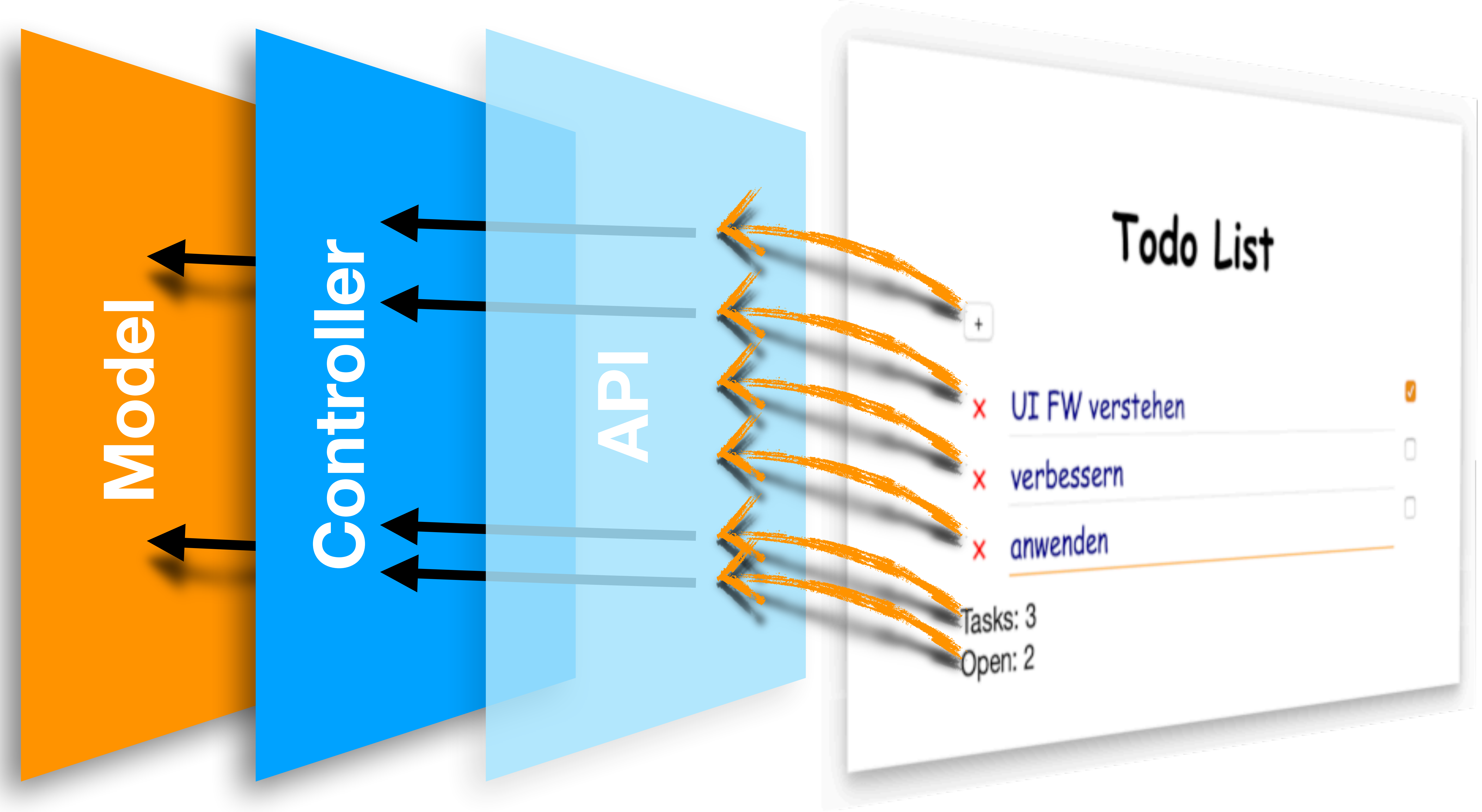
All tests ok

# Milestone 4

All Views separated

All tests ok

# MVC, classic version



# Possible Extensions

- 👍 Observer Pattern: Observable Values
- 👍 Observable Collections (List, Map)
- Binding, Validation, Conversion
- Asynchronous Data Flow (remote)

# Expert Consideration

Can you spot the memory-leaks in the Observable List?

Can you test it?

Can you resolve it?