

# Web-applikation für organisatorische Planung im Alltag auf Basis von C++



Alma Kosuta  
Bea Janott  
Jake Pettke  
Leo Hepting  
Mikel Thiele  
Philip Adam

# Agenda

- Appvorstellung (Problem=>Lösung)
- Ausgangssituation
- Nutzeranforderungen
- Initiierungsphase
- Technische Umsetzung
- Architektur
- Kommunikation
- Request Flow
- Datenspeicherung
- Planungsphase
- Konzeptionsphase (Architektur)
- Implementierungsphase
- Dokumentationsphase
- Abschlussphase

# Appvorstellung

+ Neue Liste erstellen

Meine Liste

Zweite Liste

Erste Aufgabe Hoch 15.02.2026

Zweite Aufgabe Mittel 17.02.2026

Dritte Aufgabe Niedrig 19.02.2026

Vierte Aufgabe Hoch 21.02.2026

Fünfte Aufgabe Mittel 23.02.2026

Sechste Aufgabe Niedrig 25.02.2026

Bald fällige Aufgaben

Liste abschließen

+ Neues Todo erstellen Filter

## Meine Liste

- Erste Aufgabe Hoch 15.02.2026
- Zweite Aufgabe Mittel 17.02.2026
- Dritte Aufgabe Niedrig 19.02.2026
- Vierte Aufgabe Hoch 21.02.2026
- Fünfte Aufgabe Mittel 23.02.2026
- Sechste Aufgabe Niedrig 25.02.2026

# Ausgangssituation (Problem)

- Fehlender zentraler Überblick über offene Aufgaben
- Potenzielle Doppelarbeit
- Aufgaben und Fristen werden übersehen



# Nutzeranforderungen (Lösung)

- Zentrale Verwaltung von Aufgaben
- Eingabe von: Titel, Beschreibung, Status, Priorität, Deadline
- Einfache Sortier- und Filterfunktionen
- Lokale Speicherung ohne Cloud
- Schnelle Bedienbarkeit und intuitive Oberfläche



A screenshot of a user interface titled "Meine Liste". On the left, there are buttons for "+ Neue Liste erstellen" and "Meine Liste" (selected). On the right, there are buttons for "+ Neues Todo erstellen" and "Filter". The main area displays two tasks. To the right, there are filtering options:

- Filtern nach
- Priorität: Alle
- Name: Nach Name filtern
- Fälligkeitsdatum: TT.MM.JJJJ
- Filter zurücksetzen



# Technische Umsetzung

- Browserbasierte Bedienung über lokalen HTTP-Server
- Speicherung lokal (Socket nimmt Json und speichert in eine lokale Json)
- Programmiersprachen; C++, React-TypeScript



# Architektur

Frontend (React / Vite)

↓ HTTP

Backend (C++ HTTP-Server)

↓

JsonService

↓

mainDATA.json

# API-Kommunikation Frontend ↔ Backend

## HTTP

	<u>Endpoint</u>
GET	/todos
POST	/todos
PUT	/todos/{id}
PATCH	/todos/{id}/checked
DELETE	/todos/{id}

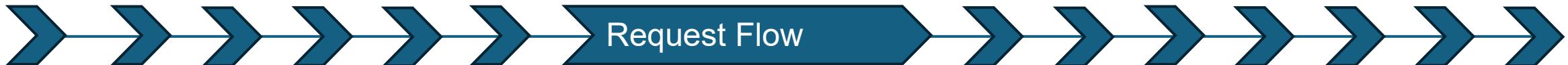
## Zweck

Todos abrufen
Neues Todo anlegen
Todo ändern
Status ändern
Todo löschen



# Request Flow

- User klickt „Add“
- Frontend → POST /todos
- Backend → JsonService::AddTodo()
- JSON-Datei wird gespeichert
- Response 201 + neues Todo
- React State Update



# Datenspeicherung

mainDATA.json

```
{  
  id: 1,  
  name: "Meine Liste",  
  todos: [  
    {  
      checked: false,  
      priority: "HIGH",  
      dueDate: "15.02.2026",  
      title: "Erste Aufgabe",  
      id: 1  
    },  
    {  
      checked: false,  
      priority: "MEDIUM",  
      dueDate: "17.02.2026",  
      title: "Zweite Aufgabe",  
      id: 10  
    }  
  ]  
}
```



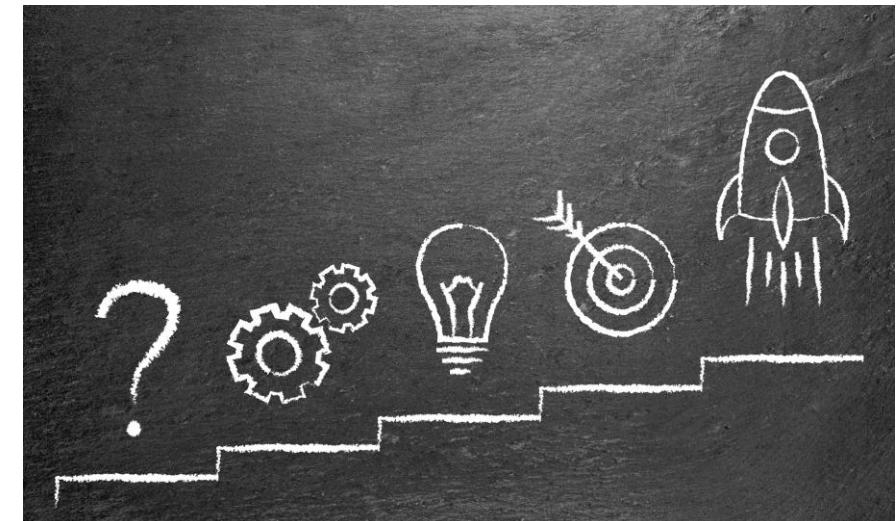
# Initiierungsphase

- Gemeinsame Besprechung der Projektidee:  
Was soll entwickelt werden?
- Erste Zieldefinition: Funktionsumfang,  
grober Zweck der Anwendung.



# Planungsphase

- Rollenklärung: Wer übernimmt welche Aufgabe?
- Kompetenzanalyse: Wer kann welche Programmiersprachen und Tools? Oder nutzt es als Gelegenheit es zu lernen?
- Entscheidung über eingesetzte Technologien (z. B. C++, JSON, Sockets, Frontend-Tools)
- Aufteilung in Arbeitsbereiche (Backend, Frontend, Dokumentation)



# Konzeptionsphase

- Erstellung eines Mockups für das Frontend  
(2 Personen)
- Erstellung einer Request Page als Grundlage  
für die Kommunikation Backend → Frontend
- Definition der Services im Backend  
(Socket-Kommunikation, JSON-Struktur)



# Implementierungsphase

- Backend-Team (2 Personen):
  - Person A: Implementierung der Socket-Kommunikation
  - Person B: Erstellung des JSON-Service zur Datenverarbeitung
- Frontend-Team (2 Personen):
  - Umsetzung des Mockups in funktionale UI
  - Implementierung der Request Page



# Dokumentationsphase

- Dokumentations-Team (2 Personen):
  - Konsolidierung aller Informationen
  - Sammlung von Screenshots aus Frontend/Backend
  - Erstellung der vollständigen Projektdokumentation: Aufbau, Funktionen, technische Erläuterungen
  - Runbook



# Abschlussphase

- Zusammenführen von Backend, Frontend und Dokumentation
- Finaler Projektstand präsentiert
- Ergebnisse gesichert und festgehalten

