

**Inhaltsverzeichnis**

[Abkürzungsverzeichnis 1](#_Toc210732025)

[Abbildungsverzeichnis 2](#_Toc210732026)

[1. Einleitung 3](#_Toc210732027)

[1.1. Ausgangssituation und Problemstellung 3](#_Toc210732028)

[1.2. Projektziele 3](#_Toc210732029)

[1.3. Rahmenbedingungen und Abgrenzung 3](#_Toc210732030)

[1.4. Anforderungen an die Anwendung 4](#_Toc210732031)

[1.4.1. Funktionale Anforderungen 4](#_Toc210732032)

[1.4.2. Nicht-funktionale Anforderungen 5](#_Toc210732033)

[2. Projektplanung 6](#_Toc210732034)

[2.1. Vorgehensmodell 6](#_Toc210732035)

[2.2. Ressourcen- und Ablaufplanung 6](#_Toc210732036)

[2.2.1. Zeitplanung 6](#_Toc210732037)

[2.2.2. Kostenplanung 6](#_Toc210732038)

[2.3. Risikoanalyse 6](#_Toc210732039)

[3. Technische Umsetzung 7](#_Toc210732040)

[3.1. Architektur und Design 7](#_Toc210732041)

[3.1.1. Architektur-Konzept (MVC) 7](#_Toc210732042)

[3.1.2. Datenfluss 7](#_Toc210732043)

[3.2. Auswahl der Technologien 7](#_Toc210732044)

[3.2.1. Software- und Hardware 7](#_Toc210732045)

[3.2.2. Programmiersprache und Frameworks 7](#_Toc210732046)

[3.3. Implementierung 7](#_Toc210732047)

[3.3.1. Kernkomponenten der Simulation 7](#_Toc210732048)

[3.3.2. Zufallselemente und Verteilung 7](#_Toc210732049)

[3.4. Benutzeroberfläche 7](#_Toc210732050)

[3.4.1. UI-Konzept und Usability 7](#_Toc210732051)

[3.4.2. Elemente und Animation 7](#_Toc210732052)

[4. Qualitätssicherung 8](#_Toc210732053)

[4.1. Testkonzept 8](#_Toc210732054)

[4.2. Testdurchführung 8](#_Toc210732055)

[4.3. Validierung der Zufallsmodelle 8](#_Toc210732056)

[4.4. Soll-Ist-Vergleich 8](#_Toc210732057)

[5. Zusammenfassung und Ausblick 9](#_Toc210732058)

[5.1. Fazit 9](#_Toc210732059)

[5.2. Ausblick 9](#_Toc210732060)

[Literaturverzeichnis 10](#_Toc210732061)

[Anhang I](#_Toc210732062)

[A1 UML-Diagramme I](#_Toc210732063)

[A2 Zeitplanung II](#_Toc210732064)

[A3 III](#_Toc210732065)

# Abkürzungsverzeichnis

**API** - Application Programming Interface

**GUI** - Graphical User Interface

**IHK** - Industrie- und Handelskammer

**ISO** - International Organization for Standardization

**JDK** - Java Development Kit

**LTS** - Long Term Support

**MVC** - Model-View-Controller (Software-Design-Pattern)

**SE** - Standard Edition

**UI** - User Interface

**UML** - Unified Modeling Language

**UX** - User Experience

# Abbildungsverzeichnis

# Einleitung

## Ausgangssituation und Problemstellung

Im Rahmen des dritten Lehrjahres wird das Simulationsprojekt „Das unberechenbare Volk“ entwickelt. Während das übergeordnete Schulprojekt der Wissensabfrage und Prüfungsvorbereitung dient, ist das Kernziel der Anwendung die Erstellung einer komplexen Wahlsimulation. Diese soll den Einfluss von externen Faktoren auf Wahlergebnisse simulieren, auswerten und visualisieren.

Das Projekt wird als Windows-Anwendung mit grafischer Benutzeroberfläche realisiert. Die Lauffähigkeit muss auf den von der Schule bereitgestellten PCs gewährleistet sein. Die zentrale Herausforderung besteht darin, die Unvorhersehbarkeit politischer Prozesse durch Zufallsereignisse zu modellieren und diese in einer performanten Echtzeit-Anwendung darzustellen.

## Projektziele

Die primären Projektziele umfassen die Entwicklung einer intuitiven, modernen und benutzerfreundlichen grafischen Benutzeroberfläche nach ISO 9241-110. Ebenso zentral ist die nachvollziehbare Darstellung der Interaktionen zwischen Wählern, Parteien und externen Einflüssen (wie medialen Faktoren) in festgelegten Zeitabschnitten.

Die Abgabe der Anwendung erfolgt als einzelne, auf Windows lauffähige, ausführbare Datei.

## Rahmenbedingungen und Abgrenzung

Als Vorgehensmodell für dieses Projekt wird das Wasserfallmodell gewählt, um die festen Anforderungen strukturiert und phasenbasiert umzusetzen.

Die technische Umsetzung basiert auf Java SE (JDK 21) und dem Framework JavaFX für die grafische Benutzeroberfläche. Diese Wahl begründet sich in der Stabilität der Long-Term-Support (LTS) Version (JDK 21) und der guten Eignung von JavaFX für komplexe Desktop-Anwendungen. Zur Verwaltung der Projektabhängigkeiten wird Maven eingesetzt.

Eine zentrale Rahmenbedingung ist die Zielplattform: Das Projekt muss als einzelne, ausführbare Windows-Anwendung konzipiert, sein, um die Lauffähigkeit auf den Schulrechnern zu gewährleisten

Um den Projektumfang klar zu definieren (Abgrenzung) und eine Überschreitung des Zeitplans zu vermeiden, wird auf komplexe Zusatzfunktionen bewusst verzichtet. Dazu zählen insbesondere die Anbindung einer externen Datenbank zur Speicherung von Simulationsergebnissen sowie der Einsatz künstlicher Intelligenz zur Wählersteuerung.

## Anforderungen an die Anwendung

Um die Softwarequalität und den Betrieb sicherzustellen, muss das Projekt eine Reihe von funktionalen und nicht-funktionalen Anforderungen erfüllen, die im Folgenden detailliert aufgeführt werden.

### Funktionale Anforderungen

Funktionale Anforderungen beschreiben, welche konkreten Operationen das System für den Benutzer ausführen muss. Ein Anwendungsfalldiagramm (siehe Anhang A, Abbildung 1) visualisiert diese Interaktionen und veranschaulicht die Systemgrenzen.

Gemäß dem Projektauftrag wurden folgende funktionale Kernanforderungen definiert:

* **Simulation:** Entwicklung einer zeitabhängigen Simulation.
* **Zufallslogik:** Die Systemlogik muss mindestens drei Zufallswerte generieren, die mit drei verschiedenen Verteilungsformen (z. B. Normal-, Gleich- und Exponentialverteilung) implementiert werden.
* **Konfiguration:** Die Simulation muss über eine grafische Benutzeroberfläche (GUI) mit mindestens sieben Eingabeparametern konfigurierbar sein.
* **Steuerung:** Es muss eine Simulationsgeschwindigkeitseinstellung implementiert werden (mindestens drei Stufen oder frei variabel).
* **Visualisierung:** Die Simulationsergebnisse sind in Echtzeit visuell darzustellen und auszuwerten.
* **Animation:** Zur visuellen Unterstützung ist eine zur Thematik passende Animation zu implementieren.
* **Auslieferung:** Das Projekt muss als einzelne, ausführbare Windows-Datei konzipiert sein, die auf den Schulrechnern lauffähig ist.

### Nicht-funktionale Anforderungen

Nicht-funktionale Anforderungen definieren die Qualitätsmerkmale und Randbedingungen des Systems:

* **Codequalität:** Die Entwicklung erfolgt nach strengen Clean-Code-Kriterien.
* **Usability:** Die Konzeption der Benutzeroberfläche (UI) orientiert sich an der Norm ISO 9241-110 , um eine hohe Benutzerfreundlichkeit zu gewährleisten.
* **User Experience (UX):** Die integrierte Animation soll die User Experience aktiv verbessern.
* **Stabilität & Performance:** Das System muss stabil und performant laufen. Um dies zu validieren, werden Belastungstests mit bis zu 2.000.000 Wählern durchgeführt, um die Datenintegrität auch unter Last sicherzustellen.

# Projektplanung

## Vorgehensmodell

Für dieses Projekt wurde das lineare Wasserfall-Modell gewählt. Die Entscheidung für dieses Vorgehensmodell basiert auf den beiden zentralen Projektbedingungen: dem festen Abgabetermin (06.02.2026) und dem klar definierten, unveränderlichen Anforderungskatalog.

Das Modell sieht vor, dass jede Phase sequenziell und erst nach dem vollständigen Abschluss der vorhergehenden Phase begonnen wird. Dieser strukturierte Ansatz eignet sich ideal für Projekte mit stabilen Anforderungen, da er eine genaue Dokumentation jeder Entwicklungsphase sicherstellt.

Zudem minimiert das Wasserfall-Modell das Risiko einer schleichenden Umfangserweiterung (Scope Creep) und unterstützt die konsequente Einhaltung des Zeitplans.

## Ressourcen- und Ablaufplanung

### Zeitplanung

### Kostenplanung

## Risikoanalyse

# Technische Umsetzung

## Architektur und Design

### Architektur-Konzept (MVC)

### Datenfluss

## Auswahl der Technologien

### Software- und Hardware

### Programmiersprache und Frameworks

## Implementierung

### Kernkomponenten der Simulation

### Zufallselemente und Verteilung

## Benutzeroberfläche

### UI-Konzept und Usability

### Elemente und Animation

# Qualitätssicherung

## Testkonzept

## Testdurchführung

## Validierung der Zufallsmodelle

## Soll-Ist-Vergleich

# Zusammenfassung und Ausblick

## Fazit

## Ausblick

# Literaturverzeichnis

# Anhang

## A1 UML-Diagramme

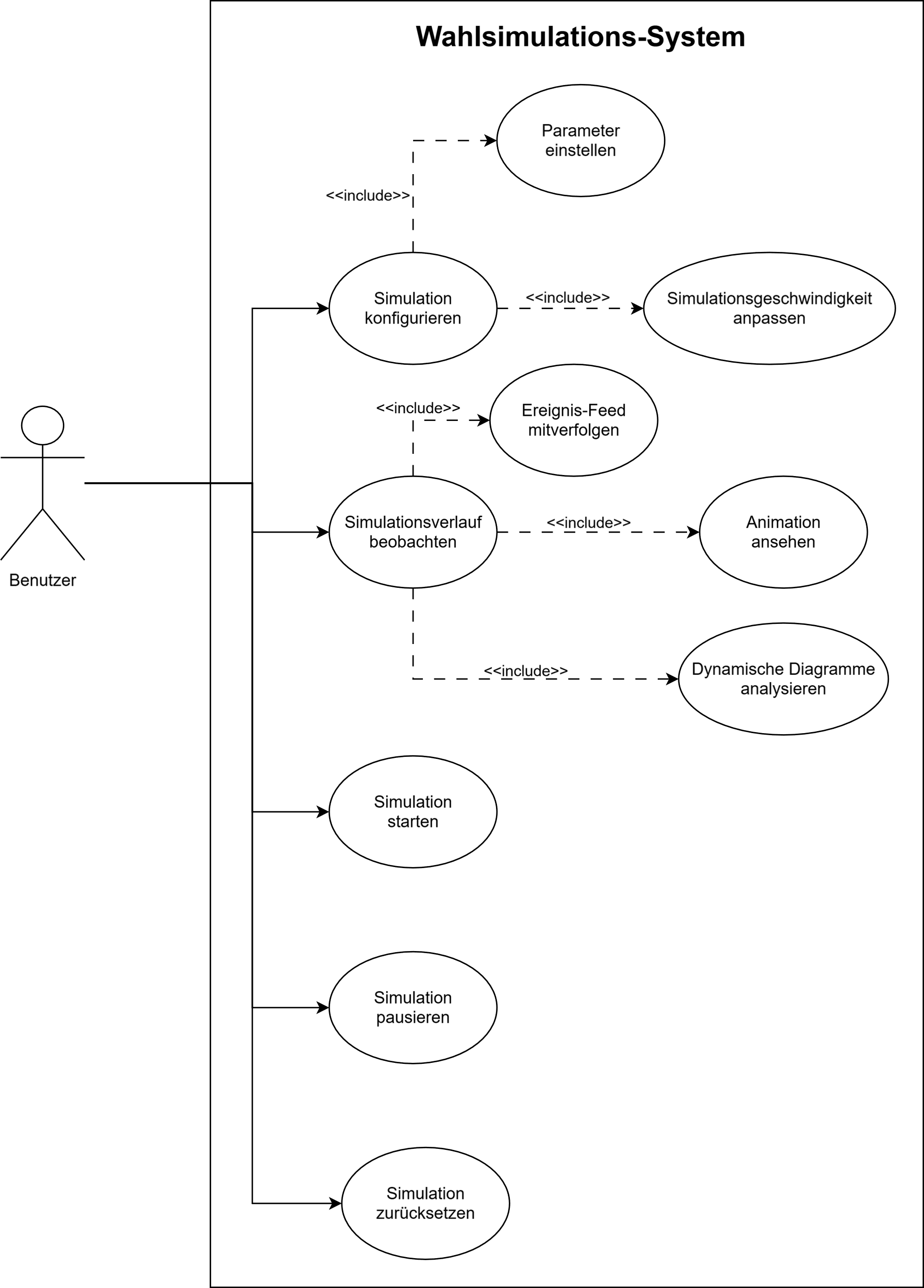


Abbildung 1: Anwendungsfalldiagramm des Wahlsystems

## A2 Zeitplanung

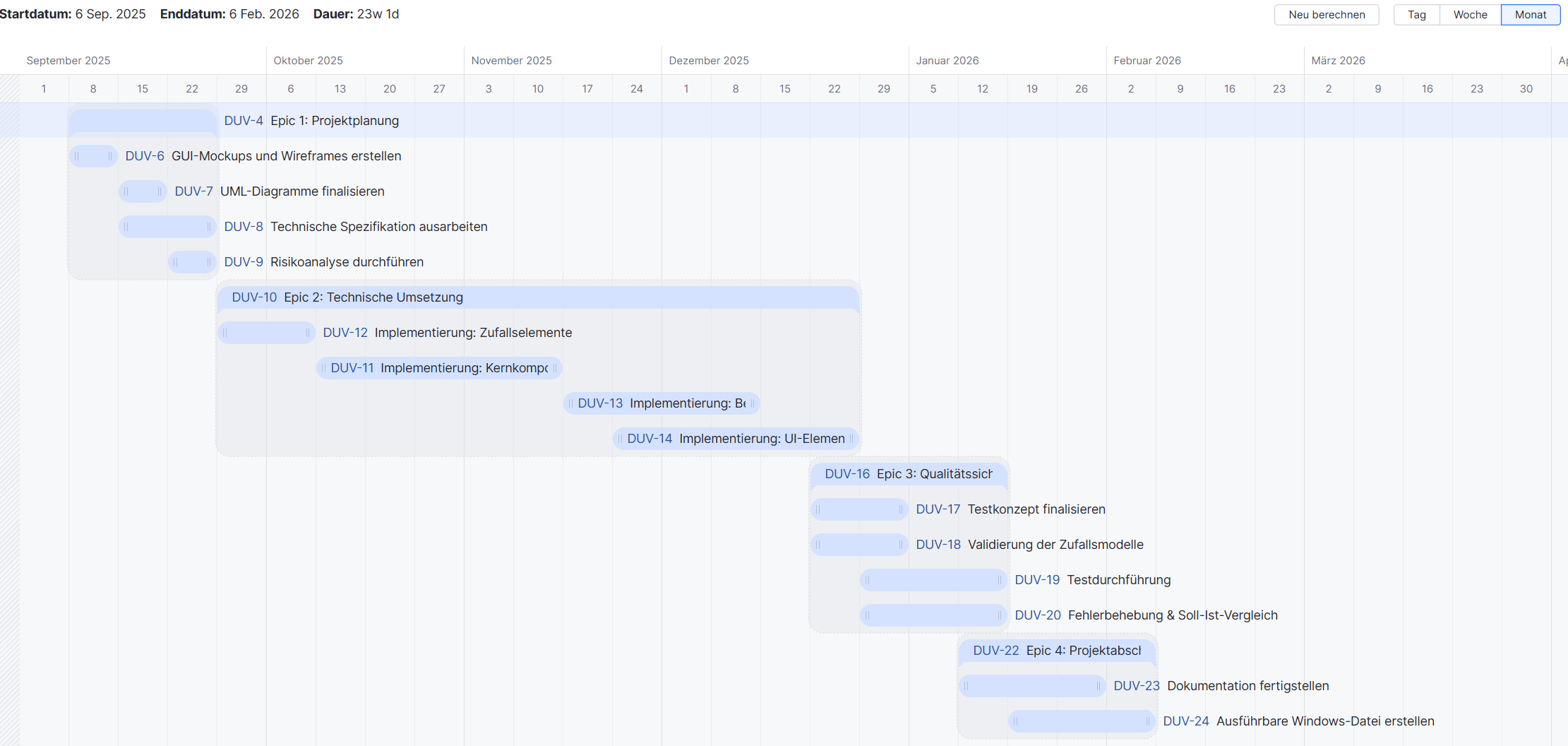


Abbildung 2: Gantt-Diagramm

## A3