

**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" als Abschlussprojekt entwickelt. Die zentrale fachliche Problemstellung besteht darin, die komplexen und oft unvorhersehbaren Dynamiken von Wahlprozessen durch eine zeitabhängige Simulation realitätsnah abzubilden. Dabei sollen externe Faktoren wie Medienpräsenz, Kampagnenbudgets und spontane Ereignisse (z. B. Skandale oder Debatten) die Wählerpräferenzen dynamisch beeinflussen.

Die zentrale technische Herausforderung liegt in der realistischen Modellierung von Zufallsereignissen mittels verschiedener Verteilungsformen (Normal-, Gleich-, Exponentialverteilung) sowie der performanten Echtzeit-Visualisierung bei großen Datenmengen (bis zu 2.000.000 Wählern). Die Anwendung muss als Windows-Desktop-Anwendung mit grafischer Benutzeroberfläche konzipiert werden und auf den von der Schule bereitgestellten PCs lauffähig sein.

Das Projekt dient sowohl der Vertiefung von Kenntnissen im Bereich der Software-Entwicklung und Projektmanagements als auch der Demonstration der erworbenen Kompetenzen gemäß IHK-Standard.

## Projektziele

Die Projektziele lassen sich in **funktionale Kernziele** und **qualitative Anforderungen** unterteilen:

**Funktionale Kernziele:**

* **Simulation:** Entwicklung einer zeitabhängigen Wahlsimulation, die Interaktionen zwischen Wählern, Parteien und externen Einflüssen (z. B. Medien, Kampagnenbudgets) in diskreten Zeitschritten nachvollziehbar darstellt.
* **Zufallsmodellierung:** Integration von mindestens drei verschiedenen Zufallsverteilungen (Normal-, Gleich-, Exponentialverteilung) zur realistischen Abbildung unberechenbarer Wahlprozesse.
* **Konfigurierbarkeit:** Bereitstellung einer grafischen Benutzeroberfläche (GUI) mit mindestens sieben frei konfigurierbaren Eingabeparametern (z. B. Anzahl der Wähler, Parteien, Medieneinfluss).
* **Visualisierung:** Echtzeit-Darstellung der Simulationsergebnisse durch dynamische Diagramme und eine thematisch passende Animation.
* **Auslieferung:** Bereitstellung der Anwendung als einzelne, ausführbare Windows-Datei (.exe), die ohne Installation auf den Schulrechnern lauffähig ist.

**Qualitative Anforderungen:**

* **Usability:** Gestaltung einer intuitiven und benutzerfreundlichen Oberfläche gemäß ISO 9241-110 (Grundsätze der Dialoggestaltung).
* **Performance:** Gewährleistung stabiler Systemleistung auch bei hoher Last (Belastungstests mit bis zu 2.000.000 simulierten Wählern).
* **Codequalität:** Einhaltung von Clean-Code-Prinzipien zur Sicherstellung der Wartbarkeit und Erweiterbarkeit des Systems.

## Rahmenbedingungen und Abgrenzung

**Rahmenbedingungen:**

**Die Entwicklung des Projekts unterliegt organisatorischen und technischen Rahmenbedingungen. Das Projekt muss innerhalb eines Zeitrahmens von 23 Wochen (06.09.2025 bis 06.02.2026) abgeschlossen werden. Der feste Abgabetermin erfordert eine präzise Zeitplanung und konsequente Einhaltung der Meilensteine.**

**Aufgrund der stabilen Anforderungen und des festen Abgabetermins wird das Wasserfallmodell als sequenzielles Vorgehensmodell gewählt. Dieses ermöglicht eine strukturierte, phasenbasierte Entwicklung mit klarer Dokumentation jeder Projektphase (siehe 2.1** Vorgehensmodell**).**

**Die technische Umsetzung basiert auf Java SE (JDK 21 LTS) und dem Framework JavaFX für die grafische Benutzeroberfläche. Diese Wahl begründet sich in der Stabilität der Long-Term-Support-Version sowie der guten Eignung von JavaFX für komplexe Desktop-Anwendungen mit Echtzeitvisualisierung. Die Projektverwaltung erfolgt über Maven als Build-Management-Tool.**

**Das Projekt muss als einzelne, ausführbare Windows-Anwendung (.exe) bereitgestellt werden, die ohne Installation auf den Schulrechnern lauffähig ist. Die Kompatibilität mit der schulischen IT-Infrastruktur ist zwingend erforderlich.**

**Abgrenzung:**

**Um den Projektumfang klar zu definieren und eine Überschreitung des Zeitplans zu vermeiden, wird auf bestimmte Funktionen bewusst verzichtet. Es erfolgt keine Anbindung einer externen Datenbank zur dauerhaften Speicherung von Simulationsergebnissen, da alle Daten ausschließlich im Arbeitsspeicher verwaltet werden. Ebenso wird auf die Implementierung KI-gestützter Algorithmen zur Wählersteuerung verzichtet – die Simulation basiert ausschließlich auf probabilistischen Zufallsmodellen. Weitere nicht umgesetzte Features umfassen Netzwerkfunktionalität (Multi-User-Unterstützung) und erweiterte Export- oder Reporting-Funktionen über die grundlegende Visualisierung hinaus. Diese Einschränkungen stellen sicher, dass das Projekt innerhalb des vorgegebenen Zeitrahmens realisierbar bleibt, während alle Kernfunktionen gemäß Projektauftrag vollständig implementiert werden.**

## Anforderungen an die Anwendung

Um die Funktionalität und Qualität der Anwendung sicherzustellen, muss das Projekt eine Reihe von funktionalen und nicht-funktionalen Anforderungen erfüllen, die aus dem Projektauftrag abgeleitet und im Folgenden detailliert beschrieben werden.

### Funktionale Anforderungen

Funktionale Anforderungen beschreiben die konkreten Operationen und Fähigkeiten, die das System für den Benutzer bereitstellen muss. Ein Anwendungsfalldiagramm (siehe Anhang A1, Abbildung 1) visualisiert die wichtigsten Interaktionen zwischen Benutzer und System und verdeutlicht die Systemgrenzen:

* **Simulation:** Entwicklung einer zeitabhängigen Simulation.
* **Zufallslogik:** Die Systemlogik muss mindestens drei Zufallswerte mit unterschiedlichen Verteilungsformen generieren. Konkret werden Normalverteilung, Gleichverteilung und Exponentialverteilung eingesetzt.
* **Konfiguration:** Die Simulation muss über eine grafische Benutzeroberfläche (GUI) mit mindestens sieben Eingabeparametern konfigurierbar sein.
* **Steuerung:** Es muss eine Simulationsgeschwindigkeitseinstellung implementiert werden, welche mindestens drei Stufen oder frei variabel eingestellt werden kann.
* **Visualisierung:** Die Simulationsergebnisse sind in Echtzeit visuell darzustellen und auszuwerten. Dazu gehören dynamische Diagramme und ein Ereignis-Feed, der bedeutende Vorkommnisse protokolliert.
* **Animation:** Zur visuellen Unterstützung ist eine zur Thematik passende Animation zu implementieren.
* **Auslieferung:** Das Projekt muss als einzelne, ausführbare Windows-Datei bereitgestellt werden, die ohne Installation auf den Schulrechnern lauffähig ist.

### Nicht-funktionale Anforderungen

Nicht-funktionale Anforderungen definieren die Qualitätsmerkmale und Randbedingungen des Systems, die über die reine Funktionalität hinausgehen:

* **Codequalität:** Die Entwicklung erfolgt nach strengen Clean-Code-Kriterien, um Wartbarkeit, Lesbarkeit und Erweiterbarkeit des Quellcodes sicherzustellen.
* **Usability:** Die Konzeption der Benutzeroberfläche (UI) orientiert sich an der Norm ISO 9241-110 (Grundsätze der Dialoggestaltung), um eine hohe Benutzerfreundlichkeit zu gewährleisten.
* **User Experience (UX):** Die integrierte Animation soll die User Experience aktiv verbessern, indem sie die Simulation visuell unterstützt und das Verständnis komplexer Zusammenhänge erleichtert.
* **Stabilität & Performance:** Das System muss stabil und performant arbeiten, auch unter hoher Last. Um dies zu validieren, werden Belastungstests mit bis zu 2.000.000 simulierten Wählern durchgeführt, um die Datenintegrität und Antwortzeiten unter Extrembedingungen zu überprüfen.

# Projektplanung

## Vorgehensmodell

Für die Durchführung dieses Projekts wurde das Wasserfallmodell als lineares, sequenzielles Vorgehensmodell gewählt. Diese Entscheidung basiert auf dem festen Abgabetermin am 06.02.2026 und dem klar definierten, unveränderlichen Anforderungskatalog gemäß Projektauftrag.

Das Wasserfallmodell sieht vor, dass die Entwicklung in aufeinanderfolgenden, klar abgegrenzten Phasen erfolgt: Anforderungsanalyse und Planung, Entwurf und Design, Implementierung, Testen und Qualitätssicherung sowie Dokumentation und Abnahme. Jede Phase wird erst nach vollständigem Abschluss der vorhergehenden Phase begonnen. Eine detaillierte Zeitplanung mit Meilensteinen ist im Anhang A2 (Gantt-Diagramm) dargestellt.

Dieser strukturierte Ansatz ermöglicht eine präzise Planung und Kontrolle des Projektfortschritts sowie eine umfassende Dokumentation jeder Entwicklungsphase entsprechend der IHK-Anforderungen. Zudem minimiert das Modell das Risiko einer schleichenden Umfangserweiterung (Scope Creep), da alle Anforderungen zu Projektbeginn festgelegt werden.

Im Vergleich zu agilen Vorgehensmodellen wie Scrum, die auf iterative Entwicklung und flexible Anforderungen ausgelegt sind, bietet das Wasserfallmodell für dieses Projekt mit stabilen Vorgaben die bessere Eignung. Die Risiken des Wasserfallmodells, insbesondere die späte Fehlererkennung, werden durch eine umfassende Planungsphase und systematische Qualitätssicherung minimiert.

## Ressourcen- und Ablaufplanung

### Zeitplanung

Die Projektdurchführung erstreckt sich über einen Zeitraum von 23 Wochen (06.09.2025 bis 06.02.2026) und ist in fünf Hauptphasen unterteilt, die sequenziell nach dem Wasserfallmodell ablaufen. Die folgende Tabelle gibt einen Überblick über die zeitliche Verteilung der Projektphasen:

|  |  |  |
| --- | --- | --- |
| **Epic** | **Beschreibung** | **Zeitraum** |
| Epic 1 | Projektplanung (Mockups, UML, Spezifikation, Risikoanalyse) | Sep. 2025 |
| Epic 2 | Technische Umsetzung (Implementierung: Zufallselemente, Kernkomponenten, Backend, UI) | Okt. - Nov. 2025 |
| Epic 3 | Qualitätssicherung (Testkonzept, Validierung, Testdurchführung, Fehlerbehebung) | Dez. 2025 - Jan. 2026 |
| Epic 4 | Projektabschluss (Dokumentation, Ausführbare Datei, Präsentation) | Jan. - Feb. 2026 |

Tabelle 1: Zeitliche Verteilung der Projektphasen

Eine detaillierte Visualisierung der Zeitplanung mit allen Arbeitspaketen, Abhängigkeiten und Meilensteinen ist im Anhang A2, Abbildung 2 als Gantt-Diagramm dargestellt. Die Epics sind in einzelne User Stories unterteilt, die nacheinander bzw. teilweise parallel abgearbeitet werden.

### Kostenplanung

Die Kostenplanung basiert auf einer Kalkulation der Personalkosten sowie der benötigten Sachmittel. Für die Projektdurchführung wird von einem Gesamtaufwand von 120Arbeitsstunden ausgegangen, verteilt auf die vier Projekt-Epics. Bei einem Stundensatz von 6,50€ ergibt sich folgende Kostenübersicht:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kostenart | Beschreibung | Menge/Stunden | Einzelpreis | Gesamtkosten |
| Personalkosten | Entwicklung & Dokumentation | 120h | 6,50€ | 780,00€ |
| Sachmittelkosten |  |  |  |  |
| - Software | JDK 21, JavaFX, Maven, IntelliJ IDEA CE | - | 0,00€ | 0,00€ |
| - Hardware | Entwicklungsrechner (vorhanden) | - | 0,00€ | 0,00€ |
| - Lizenzen | Microsoft Office 365 (vorhanden) | - | 0,00€ | 0,00€ |
| Gesamtkosten |  |  |  | **780,00€** |

Tabelle 2: Kostenplanung

## Risikoanalyse

Die Risikoanalyse identifiziert potenzielle Gefährdungen für den Projekterfolg und definiert präventive sowie reaktive Maßnahmen zur Risikominimierung. Die Risiken werden nach ihrer Eintrittswahrscheinlichkeit (gering, mittel, hoch) und ihren Auswirkungen auf Projekttermin, -kosten oder -qualität bewertet. Die folgende Übersicht zeigt die identifizierten Risiken mit entsprechenden Gegenmaßnahmen:

|  |  |  |  |
| --- | --- | --- | --- |
| Risiko | Wahrschein-lichkeit | Auswirkung | Prävention/ Reaktive Maßnahmen |
| Performance-Probleme bei hoher Wähleranzahl | Mittel | Hoch | Frühzeitige Belastungstests, Optimierung der Datenstrukturen, Profiling |
| Kompatibilitätsprobleme auf Schulrechnern | Gering | Hoch | Klärung IT-Infrastruktur, Test auf Schulrechner, jpackage mit JRE |
| Fehlerhafte Zufallsverteilungen | Mittel | Mittel | Statistische Validierung, Unit-Tests, Vergleich mit Referenzimplementierungen |
| Zeitverzug durch unterschätzte Komplexität | Mittel | Hoch | Pufferzeiten eingeplant, wöchentliche Fortschrittskontrolle, Priorisierung |
| Krankheitsbedingte Ausfälle | Gering | Mittel | Zeitpuffer, frühzeitige Information, Fokus auf Kernfunktionen |
| Unzureichende Dokumentation | Gering | Mittel | Kontinuierliche Dokumentation, frühzeitiges Review durch Betreuer |
| Nichterfüllung der IHK-Anforderungen | Gering | Hoch | Checkliste erstellen, regelmäßiger Abgleich mit Projektauftrag |

Tabelle 3: Risikoauflistung

# Technische Umsetzung

## Architektur und Design

### Architektur-Konzept (MVC)

Das Projekt folgt einer Model-View-Controller (MVC)-Architektur. Dieses Entwurfsmuster wurde gewählt, um eine klare Trennung der Verantwortlichkeiten zu gewährleisten und die Wartbarkeit sowie Erweiterbarkeit des Systems zu erhöhen.

Das Model enthält die gesamte Simulationslogik und ist vollständig vom User Interface entkoppelt. Die View ist ausschließlich für die Darstellung der Daten und die Erfassung von Benutzereingaben verantwortlich und wird mit JavaFX umgesetzt. Der Controller dient als Vermittler zwischen Model und View, verarbeitet Benutzereingaben und aktualisiert beide Komponenten entsprechend.

Als Alternative wurde das Model-View-ViewModel (MVVM)-Pattern in Betracht gezogen, das durch Data Binding eine noch stärkere Entkopplung zwischen View und Logik ermöglicht. MVVM wird häufig in modernen UI-Frameworks wie JavaFX eingesetzt und würde die automatische Synchronisation von Daten und UI-Elementen vereinfachen. Allerdings wurde MVC bevorzugt, da es für die Projektkomplexität ausreichend ist, eine geringere Einarbeitungszeit erfordert und eine klarere, explizite Kontrolle über den Datenfluss bietet. Der zusätzliche Overhead durch Data Binding würde bei diesem Projektumfang keinen signifikanten Mehrwert bieten.

Diese Architektur bietet den Vorteil, dass Änderungen an der Benutzeroberfläche die Simulationslogik nicht beeinträchtigen und umgekehrt. Zudem ermöglicht die klare Struktur eine parallele Entwicklung der Komponenten und erleichtert das Testen einzelner Module.

### Datenfluss

Der Datenfluss im System folgt einem unidirektionalen Prinzip und ist im Sequenzdiagramm (siehe Anhang A1, Abbildung 3) visualisiert. Die Daten durchlaufen drei zentrale Phasen: Eingabe, Verarbeitung und Ausgabe.

In der Eingabephase erfasst die View Konfigurationsparameter vom Benutzer, wie die Anzahl der Wähler und Parteien, Anfangspräferenzen, Medieneinfluss und Kampagnenbudgets. Diese Daten werden vom Controller validiert und als strukturierte Parameter an das Model übergeben.

Das Model führt in der Verarbeitungsphase die Simulation in diskreten Zeitschritten durch. Dabei werden die Eingabeparameter mit den Zufallsverteilungen (Normal-, Gleich-, Exponentialverteilung) kombiniert, um Meinungsänderungen der Wähler, Kampagneneffekte und zufällige Ereignisse zu berechnen. Das Ergebnis sind aktualisierte Zustandsdaten, die die Verteilung der Wählerpräferenzen, Unterstützerzahlen der Parteien und aufgetretene Ereignisse beschreiben.

In der Ausgabephase werden die berechneten Daten vom Controller abgerufen und an die View weitergegeben. Diese transformiert die numerischen Daten in visuelle Darstellungen (dynamische Diagramme, Animation) und textuelles Feedback (Ereignis-Feed). Die Aktualisierung erfolgt in Echtzeit, sodass Änderungen unmittelbar für den Benutzer sichtbar werden.

Dieser unidirektionale Datenfluss stellt sicher, dass die Komponenten lose gekoppelt bleiben und Daten niemals direkt von der View zum Model oder umgekehrt fließen, sondern immer über den Controller vermittelt werden

## 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

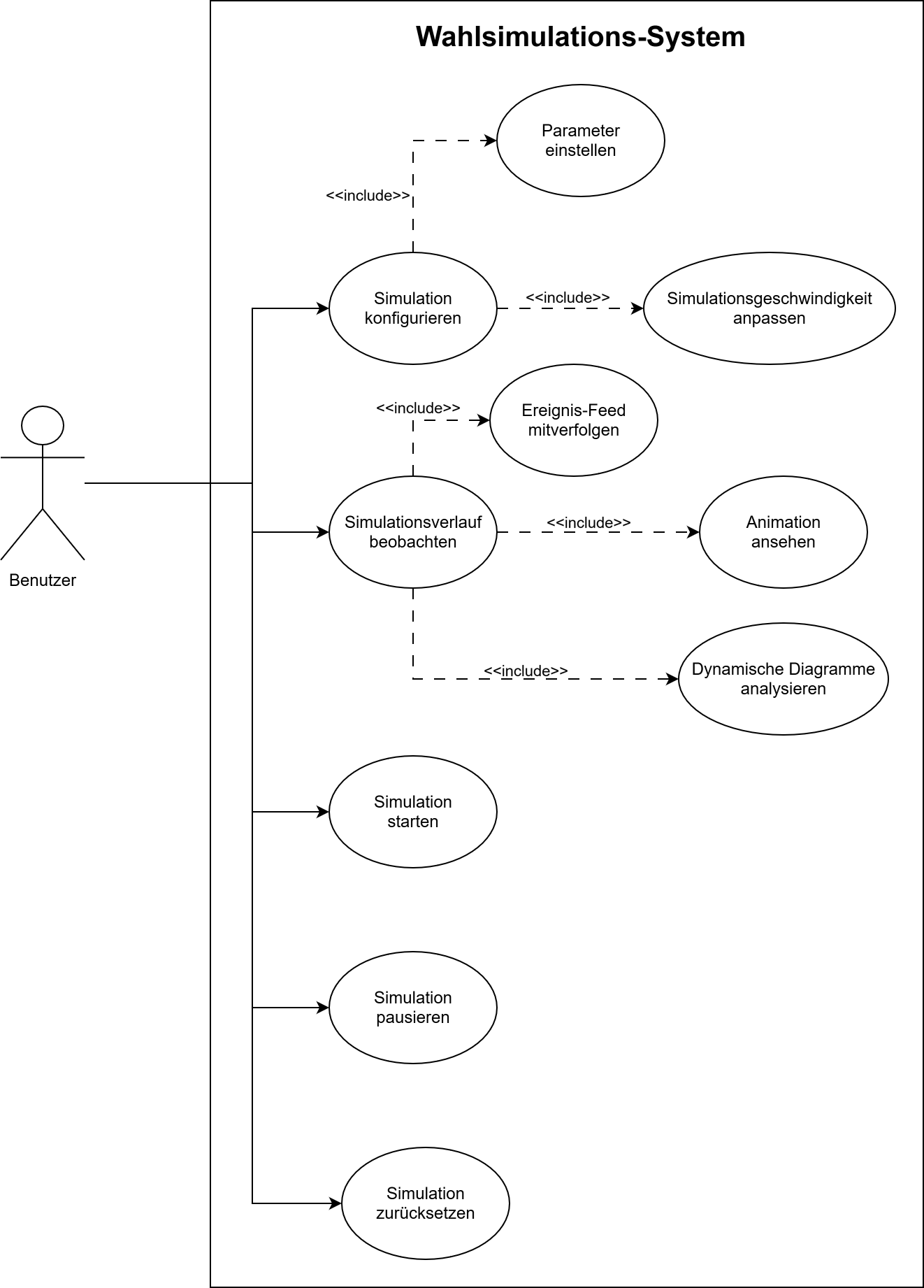


Abbildung 1: Anwendungsfalldiagramm des Wahlsystems

## A2 Zeitplanung

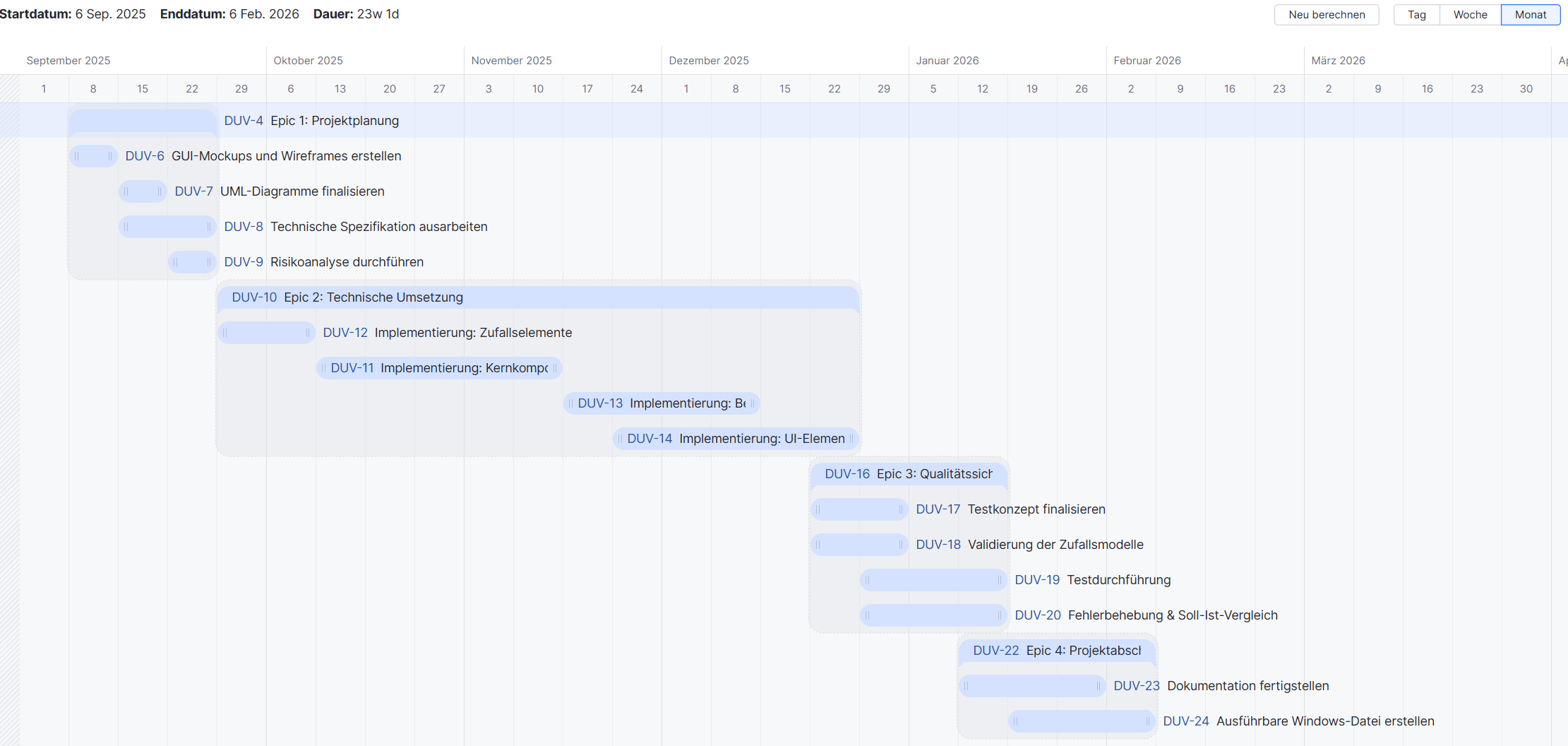


Abbildung 2: Gantt-Diagramm

## A3