

**Inhaltsverzeichnis**

[Abkürzungsverzeichnis 1](#_Toc208559824)

[Abbildungsverzeichnis 2](#_Toc208559825)

[1. Einleitung 3](#_Toc208559826)

[1.1. Ausgangssituation und Problemstellung 3](#_Toc208559827)

[1.2. Projektziele 3](#_Toc208559828)

[1.3. Rahmenbedingungen und Abgrenzung 3](#_Toc208559829)

[1.4. Anforderungen an die Anwendung 4](#_Toc208559830)

[1.4.1. Funktionale Anforderungen 4](#_Toc208559831)

[1.4.2. Nicht-funktionale Anforderungen 4](#_Toc208559832)

[2. Projektplanung 6](#_Toc208559833)

[2.1. Vorgehensmodell 6](#_Toc208559834)

[2.2. Ressourcen- und Ablaufplanung 6](#_Toc208559835)

[2.2.1. Zeitplanung 6](#_Toc208559836)

[2.2.2. Kostenplanung 6](#_Toc208559837)

[2.3. Risikoanalyse 6](#_Toc208559838)

[3. Technische Umsetzung 7](#_Toc208559839)

[3.1. Architektur und Design 7](#_Toc208559840)

[3.1.1. Architektur-Konzept (MVC) 7](#_Toc208559841)

[3.1.2. Datenfluss 7](#_Toc208559842)

[3.2. Auswahl der Technologien 7](#_Toc208559843)

[3.2.1. Software- und Hardware 7](#_Toc208559844)

[3.2.2. Programmiersprache und Frameworks 7](#_Toc208559845)

[3.3. Implementierung 7](#_Toc208559846)

[3.3.1. Kernkomponenten der Simulation 7](#_Toc208559847)

[3.3.2. Zufallselemente und Verteilung 7](#_Toc208559848)

[3.4. Benutzeroberfläche 7](#_Toc208559849)

[3.4.1. UI-Konzept und Usability 7](#_Toc208559850)

[3.4.2. Elemente und Animation 7](#_Toc208559851)

[4. Qualitätssicherung 8](#_Toc208559852)

[4.1. Testkonzept 8](#_Toc208559853)

[4.2. Testdurchführung 8](#_Toc208559854)

[4.3. Validierung der Zufallsmodelle 8](#_Toc208559855)

[4.4. Soll-Ist-Vergleich 8](#_Toc208559856)

[5. Zusammenfassung und Ausblick 9](#_Toc208559857)

[5.1. Fazit 9](#_Toc208559858)

[5.2. Ausblick 9](#_Toc208559859)

[Literaturverzeichnis 10](#_Toc208559860)

[Anhang i](#_Toc208559861)

# 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 ein Simulationsprojekt namens „Das unberechenbare Volk“. Ziel des Projektes ist die Wissensabfrage und Prüfungsvorbereitung der einzelnen Auszubildenden. Aufgrund dessen umfängt das Projekt folgendes Ziel:

Eine Visualisierung einer komplexen Wahlsimulation zu erstellen, welche den Einfluss von externen Faktoren auf die Wahlergebnisse auswertet und simuliert.

Das Projekt soll als Windows-Anwendung mit einer grafischen Benutzeroberfläche umgesetzt werden, die Ausführung des Projektes soll auf den, von der Schule bereitgestellten, PCs umgesetzt werden. Die Kernherausforderung des Projektes ist es, die Unvorhersehbarkeit der politischen Prozesse durch Zufallsereignisse zu modellieren und diese in einer performanten Echtzeit-Anwendung darzustellen.

## Projektziele

Die für dieses Projekt fokussierten Projektziele sind eine intuitive, moderne und benutzerfreundliche grafische Benutzeroberfläche und eine Darstellung von Interaktionen zwischen Wählern, Parteien und externen Einflüssen. Das Projekt soll die Auswirkung medialer und anderer externer Faktoren in festgelegten Zeitabschnitten nachvollziehbar visualisieren. Die Abgabe der Anwendung am **06.02.2026** erfolgt mit einer einzelnen Ausführbaren, auf Windows lauffähigen Datei.

## Rahmenbedingungen und Abgrenzung

Um eine strukturierte und phasenbasierte Entwicklung sicherzustellen, wird im Rahmen dieser Projektarbeit das Wasserfallmodell verwendet. Die technische Umsetzung des Projektes erfolgt ausschließlich auf Basis von Java SE, die grafische Benutzeroberfläche des Projektes wird mit JavaFX realisiert. Um entstehende Abhängigkeiten angebracht Verwalten zu können wird Maven eingesetzt.

Die Entscheidung zu Verwendung von JDK 21 beruht auf mehreren Faktoren. Die Verwendung von JDK 21 bietet einen Long-Term-Support in Vergleich zu anderen JDKs die den Support nur zur nächsten Veröffentlichungsversion beibehalten. Sie versichert eine hohe Sicherheit durch stetige Sicherheitsupdates und eine langfristige Stabilität und Zuverlässigkeit. Sowie zur Prävention möglicher Kompatibilitätsprobleme der Schulrechner, welche ggf. nicht die benötigte JDK Version vorweisen.

Dieses Projekt beschränkt sich bewusst auf die im Projektauftrag definierten Funktionen. Um den Projektumfang nicht zu überschreiten werden komplexere Zusatzfunktionen wie die Anbindung einer Datenbank zur Speicherung der Simulationsergebnisse oder eine künstliche Intelligenz ausgeschlossen. Die Anwendung wird zudem als reine Windows-Anwendung um die Lauffähigkeit auf der Zielhardware zu gewährleisten.

## Anforderungen an die Anwendung

Das Projekt muss im Rahmen der Anforderungen, zur Definierung der Softwarequalität und Softwarebetrieb eine Reihe funktionaler und nicht-funktionaler Anforderungen erfüllen.  
Diese im Folgenden detailliert aufgeführt.

### Funktionale Anforderungen

Die Projektbedingungen sehen die Entwicklung einer zeitabhängigen Simulation vor. Des Weiteren muss die Systemlogik die Generierung von mindestens drei Zufallswerten beinhalten, welche mit 3 verschiedenen Verteilungsformen Normal-, Gleich- und Exponentialverteilung implementiert werden soll. Die Benutzeroberfläche muss die Konfiguration der Simulation über mindestens 7 Parametern ermöglichen. Es soll zudem eine Simulationsgeschwindigkeitseinstellung implementiert werden, welche mindestens 3 Stufen enthält oder frei variable einstellbar sein soll.

Die Ergebnisse des Projektes sollen in Echtzeit visuell dargestellt und ausgewertete werden. Zur Unterstützung der visuellen Darstellung soll eine, zur Thematik passende Animation implementiert werden.

Das gesamte Projekt ist als ausführbare Windows-Datei zu konzipieren, um die Lauffähigkeit auf den Schulrechnern zu gewährleisten.

### Nicht-funktionale Anforderungen

Neben den funktionalen Anforderungen müssen auch die nicht-funktionalen Anforderungen erfüllt werden. Die gesamte Entwicklung des Codes erfolgt unter strengen Clean-Code Kriterien. Um die Benutzerfreundlichkeit der Benutzeroberfläche zu gewährleisten, verläuft die Konzeption und Entwicklung nach ISO-9241-110. Das zusätzliche Einfügen thematisch passender Animationen soll die Benutzer Erfahrung verbessern. Ein weiterer grundlegender Aspekt ist die Gewährleistung der Stabilität, Performance und Datenintegrität in Folge dessen werden Belastungstest mit mindestens 2.000.000 Wählern durchgeführt

# Projektplanung

## Vorgehensmodell

## 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