

Pflichtenheft

# Praxis der Softwareentwicklung

## Gruppe 1

Entwicklung einer Software zur Berechnung  
der Mandatsverteilung im Deutschen

Philipp Löwer, Anton Mehlmann, Manuel Olk, Enes Ördek,  
Simon Schürg, Nick Vlasoff



WS 2013 / 14

# Inhaltsverzeichnis

<b>1</b>	<b>Produktübersicht</b>	<b>2</b>
1.1	Lizenz . . . . .	3
<b>2</b>	<b>Zielbestimmung</b>	<b>3</b>
2.1	Musskriterien . . . . .	3
2.2	Sollkriterien . . . . .	3
2.3	Kannkriterien . . . . .	3
2.4	Abgrenzungskriterien . . . . .	3
<b>3</b>	<b>Produkteinsatz</b>	<b>3</b>
3.1	Anwendungsbereiche . . . . .	3
3.2	Zielgruppen . . . . .	3
3.3	Betriebsbedingungen . . . . .	3
<b>4</b>	<b>Produktumgebung</b>	<b>3</b>
4.1	Software . . . . .	3
4.2	Hardware . . . . .	4
4.3	Orgware . . . . .	4
4.4	Schnittstellen . . . . .	4
<b>5</b>	<b>Funktionale Anforderungen</b>	<b>4</b>
5.1	GUI (View) . . . . .	4
5.2	Schnittstellen (Controller) . . . . .	5
5.3	Datenhaltung und Verarbeitung (Model) . . . . .	5
<b>6</b>	<b>Produktdaten</b>	<b>5</b>
<b>7</b>	<b>Produktleistungen</b>	<b>6</b>
<b>8</b>	<b>Nicht-funktionale Anforderungen</b>	<b>6</b>
<b>9</b>	<b>Qualitätsanforderungen</b>	<b>6</b>
<b>10</b>	<b>Globale Testfälle und Szenarien</b>	<b>7</b>
<b>11</b>	<b>Systemmodelle</b>	<b>7</b>
11.1	Systemarchitektur . . . . .	7
<b>12</b>	<b>Benutzungsoberfläche</b>	<b>8</b>
<b>13</b>	<b>Spezielle Anforderungen an die Entwicklungsumgebung</b>	<b>8</b>
<b>14</b>	<b>Zeit- und Ressourcenplanung</b>	<b>8</b>
<b>15</b>	<b>Ergänzungen</b>	<b>8</b>
<b>16</b>	<b>Glossar</b>	<b>8</b>

## 1 Produktübersicht

Bei dem Produkt handelt es sich um ein Programm, das die Sitzverteilung im Deutschen Bundestag gemäß der gesetzlichen Bestimmungen exakt berechnet und deren Zustandekommen verständlich darlegt. Erreicht wird dies durch eine minimalistische und intuitive Benutzeroberfläche.

## **1.1 Lizenz**

Der Quellcode des Programms wird der Öffentlichkeit frei zur Verfügung gestellt. Es wird die GPL V3 Lizenz verwendet, damit das Projekt nach einer Modifizierung weiterhin öffentlich bleibt.

# **2 Zielbestimmung**

## **2.1 Musskriterien**

- Auswertung von Wahlergebnissen nach gesetzlicher Bestimmung
- Grafische Benutzeroberfläche
- Vergleich mehrerer Wahlausgänge
- Importmöglichkeit von Wahlergebnissen (.csv)
- Manipulation von Eingabedaten

## **2.2 Sollkriterien**

- Auffinden paradoxer Wahlausgänge
- Kartographische Darstellung der Bundesländer

## **2.3 Kannkriterien**

- Hilfe (Benutzerhandbuch)

## **2.4 Abgrenzungskriterien**

- Keine Mobile Anwendung oder Web-App
- Keine namentliche Nennung von Abgeordneten

# **3 Produkteinsatz**

## **3.1 Anwendungsbereiche**

- Überprüfung der Wahlergebnisse

## **3.2 Zielgruppen**

- Menschen, die kritisch gegenüber Wahlergebnissen sind
- (Unabhängige) Medien
- Politisch Interessierte

## **3.3 Betriebsbedingungen**

- Keine Verbindung zum Internet nötig

# **4 Produktumgebung**

## **4.1 Software**

- Java Runtime Environment SE 1.7 oder neuer.
- Betriebssystem z.B. Windows, Linux, Mac OS

## 4.2 Hardware

Mindestanforderungen:

- Bildschirmauflösung: 1024x768

## 4.3 Orgware

- Keine weiteren Rahmenbedingungen notwendig

## 4.4 Schnittstellen

- Importieren/ Exportieren von Zuständen und Daten

# 5 Funktionale Anforderungen

Funktionale Anforderungen werden durch eine vierstellige Nummer gekennzeichnet. Die erste Nummer kennzeichnet den folgenden Bereich:

- GUI (View)
- Schnittstellen (Controller)
- Datenhaltung (Model)

Die restlichen Nummern dienen zur Durchnummerierung.

## 5.1 GUI (View)

- /F10010/ Programmstart  
Initialisierung des Hauptfensters. Es wird, falls vorhanden, der letzte Programmzustand aufgerufen (/F20010/). Andernfalls lädt das Programm die Daten der letzten Bundestagswahl.
- /F10020/ Menü  
Im Menü sind folgende Punkte gelistet:
  - Datei
    - \* Speichern des aktuellen Zustandes /F20010/
    - \* Laden eines Zustandes /F20011/
    - \* Importieren von Daten /F20020/
    - \* Exportieren von Daten /F20030/
  - Bearbeiten
    - \* Rückgängig /F20040/
  - Extras
  - Hilfe /F10030/
- /F10030/ Hilfe  
Durch den Klick auf diesen Menüpunkt öffnet sich ein Fenster, in der Handbücher zum Programm zu finden sind. Des weiteren ein kleines About, mit den wichtigsten Informationen zum Programm.
- /F10050/ Kartografische Darstellung der Länder  
Die Länder werden nach einer Überprüfung (/F30010/), Kartografisch im Fenster dargestellt.
- /F10100/ Programmende  
Das Programm speichert den aktuellen Zustand (/F20010/) und schließt das Programm.

## 5.2 Schnittstellen (Controller)

- /F20010/ Speichern des aktuellen Programmzustandes  
Es wird das Zustands-Objekt (/PD01/) in einer Datei abgelegt. Dabei kann der Benutzer beim Speichern einen internen Namen und ein Kommentar abgeben, welche mit gespeichert werden.
- /F20011/ Laden eines Programmzustandes  
Der Benutzer wählt eine Datei aus. Es wird überprüft, ob es sich um ein gültiges Objekt handelt. Falls die Version unterschiedlich ist, wird die Datei konvertiert und geladen.
- /F20020/ Importieren von Daten  
Die .csv-Dateien des Bundeswahlleiters können importiert werden.
- /F20025/ Filtern der relevanten Daten  
Die benötigten Dateien werden aus der ausgewählten .csv-Datei geladen und zu der Model-Klasse geschickt.
- /F20030/ Exportieren von Daten  
Daten können als .csv-Dateien oder als JSON/XML exportiert werden.
- /F20040/ Rückgängig machen  
Es können Operationen rückgängig gemacht werden. Hierfür wird das History-Objekt (/PD05/) verwendet.

## 5.3 Datenhaltung und Verarbeitung (Model)

- /F30010/ Überprüfen der Ländernamen  
Überprüft ob die eingegebenen Ländernamen in dem Daten-Objekt (/PD03/) korrekt sind. Falls alle Ländernamen gefunden werden, wird die Kartografische Darstellung (/F10050/) aktiviert.
- /F30020/ Überprüfen der Stimmen  
Überprüft ob die eingegebenen Stimmen in dem Daten-Objekt (/PD03/) korrekt sind. Falls die Anzahl der Stimmen  $\geq 0$  sind, kann die Sitzverteilung mit dem Wahlgesetz-Objekt (/PD04/) berechnet werden.

## 6 Produktdaten

- /PD01/ Zustands-Object  
Repräsentiert den aktuellen Zustand des Programms. Es beinhaltet alle Informationen, um den genauen Zustand des Programms wiederherzustellen. Es handelt sich um eine serialisierbare Klasse.
  - Version des Programms (um Abwärtskompatibilität zu gewährleisten)
  - Datum und Uhrzeit der Erstellung
  - Name/ID
  - Kommentar
  - Fenster-Objekt als Liste
  - History-Objekt
- /PD02/ Fenster-Objekt
  - Wahlergebnisse (Daten-Objekt)
  - Wahlgesetz-Objekt
  - Name/ID
- /PD03/ Daten-Objekt  
Beinhaltet die Anzahl der (Erst- und Zweit-)Stimmen je Wahlkreis und Partei.

- Name/ID
- Kommentar (Quelle)
- Wahlkreise mit Stimmen als Array.
- /PD04/ Wahlgesetz-Objekt  
Beinhaltet das Algorithmus zur Berechnung der Sitze mit dem Daten-Objekt als Eingabedatum. Des weiteren überprüft das Objekt, ob mit dem Datenobjekt eine Wahl überhaupt simuliert werden kann (wenn ein Datenobjekt beispielsweise keine Erststimmen enthält wie in sehr alten Wahlen).
- /PD05/ History-Objekt  
Dieses Objekt zeichnet alle Veränderungen am Programm auf. Mithilfe dieses Objektes können Operationen über das Menü Bearbeiten oder STRG+Z rückgängig gemacht werden (/F10020/ und /F20040/).

## 7 Produktleistungen

- Zeit
  - Starten + Laden des letzten Zustandes: unter 5 Sekunden.
  - Beenden + Speichern des aktuellen Zustandes: unter 5 Sekunden.
  - Exportieren/Importieren von Daten: unter 10 Sekunden.
- Genauigkeit  
Die Genauigkeit des Algorithmus zur Sitzberechnung muss dem Wahlgesetz entsprechen und exakte Ergebnisse liefern.

## 8 Nicht-funktionale Anforderungen

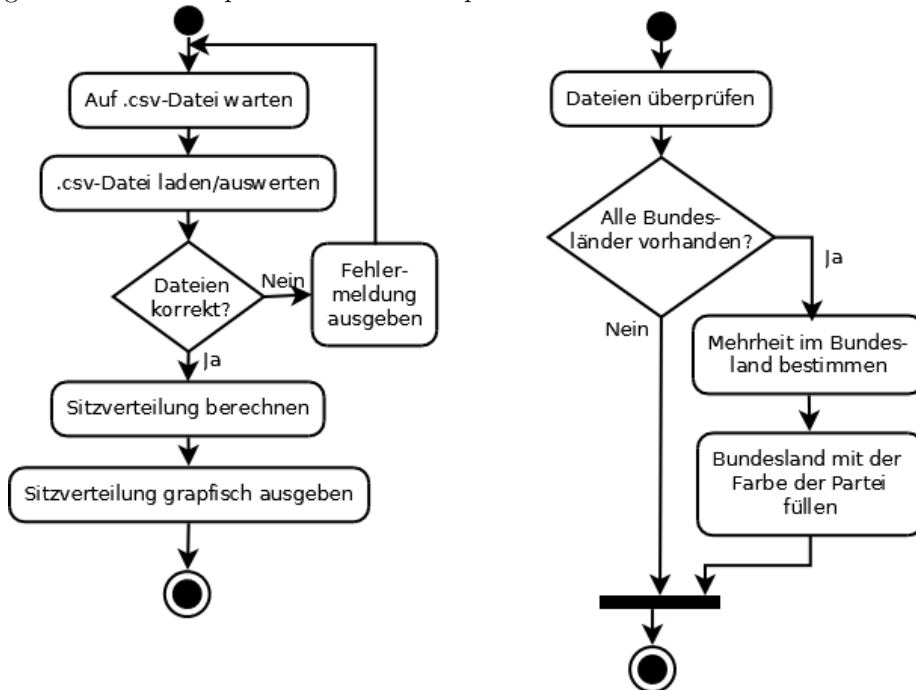
- Allgemeine Anforderungen:
  - Die Sitzverteilung muss für den Benutzer transparent und nachvollziehbar dargestellt werden.
- Sicherheitsanforderungen:
  - Die Eingabedaten dürfen während der Berechnung nicht verändert werden.
- Plattformunabhängigkeit:  
Das Programm muss auf der offiziellen Oracle JRE laufen.

## 9 Qualitätsanforderungen

- Hilfreiche Fehlermeldungen
- Kein Datenverlust (auch nach Programmabstürzen)

## 10 Globale Testfälle und Szenarien

Folgende Funktionssequenzen sind zu überprüfen:



Folgende Datenkonsistenzen müssen eingehalten werden:

- noch nichts...

Folgende unzulässige Aktionen müssen korrekt behandelt werden:

- Negative Stimmenanzahl
- Buchstaben als Stimmen

Testszenarien:

- Falsche Daten importieren:  
Starten des Programms → Im Hauptmenü auf Datei klicken → Datei importieren auswählen → Im Dateibrowser die falsche .csv-Datei auswählen → Mit dem Button Laden bestätigen → Eine Fehlermeldung taucht auf → Programm befindet sich wieder im Startzustand
- Manuell Daten modifizieren:  
Starten des Programms → Korrekte Daten laden → Die Sitzverteilung wird angezeigt → Den Wert zweier Parteien miteinander tauschen → Die Sitzverteilung erneut berechnen → Eine mögliche Veränderung der Sitzverteilung wird angezeigt

## 11 Systemmodelle

### 11.1 Systemarchitektur

Das Programm basiert auf der MVC- Architektur, wobei auf eine saubere Trennung der Einheiten Model, View und Controller geachtet wird. Dies sorgt nicht nur für einen flexiblen Programmentwurf, so dass spätere Änderungen bzw. Erweiterungen erleichtert werden, sondern garantiert auch die Trennung kritischer Komponenten, wie der Algorithmusimplementierung, von weniger sensiblen Komponenten, wie der GUI, und dient allgemein der Übersichtlichkeit.

## **12 Benutzungsoberfläche**

## **13 Spezielle Anforderungen an die Entwicklungsumgebung**

- Allgemein
  - Latex
  - Versionskontrolle mit SVN
- Entwicklung
  - IDE: Eclipse
- Entwurf
  - DIA für Diagramme
- Validierung
  - JUnit
- Teamkommunikation
  - Google-Groups Mailingliste

## **14 Zeit- und Ressourcenplanung**

## **15 Ergänzungen**

## **16 Glossar**