



Gökhan Witteborn-Demir, Marcus Koppelman, Jan Felix Jacobsen, Dennis Podkolsin

# Agenda

// 01

## Einführung

Mit der Motivation erläutern wir Hintergründe und beschreiben theoretische Erkenntnisse.

// 02

## Hauptteil

Was? Wie? Wer? Wann?  
Wozu? Womit? Warum?

// 03

## Zusammenfassung

Wir ziehen ein Fazit über bisherige Arbeiten und geben über die Programmausführung hinaus Ausblick für die weitere Arbeit am Projekt.

# 01 // Einführung



# Einführung // Motivation

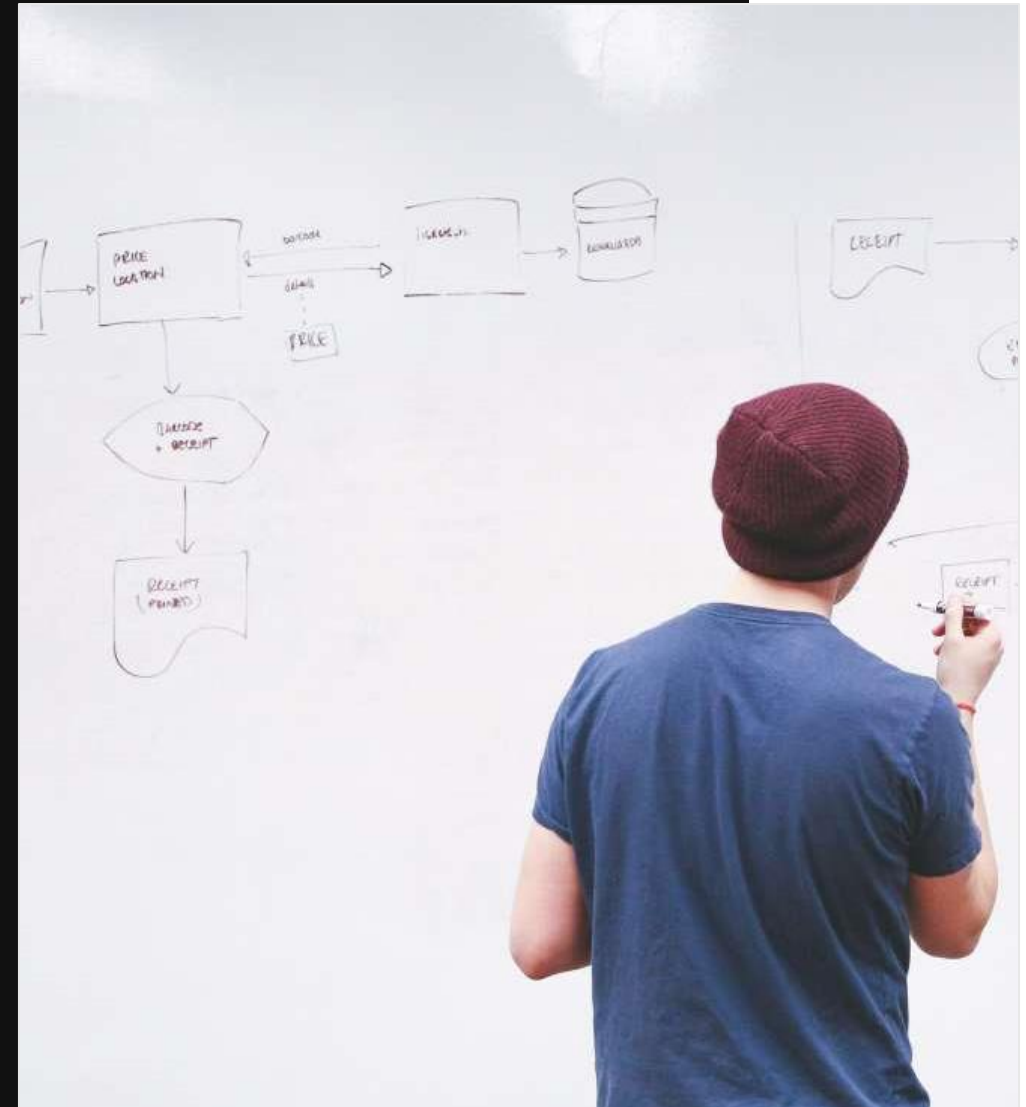
- > Viele Geschlechter (Gender)
- > Allgemeinheitliches Ziel: In Texten alle Leser\*innen ansprechen
- > Uneinheitliches Gendern: Übersichtlichkeit in großen Texten schnell verloren
- > Fokus des Autors bzw. der Autorin primär auf Textinhalt

**Es besteht klarer Bedarf für automatisierte Hilfe beim Gendern.**



## 02 //

# Hauptteil



# Gesellschaftlichen Mehrwert leisten.

Warum?

Wer?

Wie?

Wann?

Wozu?

Womit?

- > Ungleichheit der Geschlechter auch in der Sprache
- > Auf der ganzen Welt werden Frauen oft benachteiligt
- > **Gegenbewegung:** Gendern bekommt immer mehr Bedeutung im akademischen, politischen und auch journalistischen Bereich

**Es ist schlicht schwierig den Überblick bei diesem Schreibstil zu behalten, wenn man noch nicht damit aufgewachsen ist, auf genderneutrale Sprache zu achten.**

# Stakeholder & Zielgruppen identifizieren.

Warum?

Wer?

Wie?

Wann?

Wozu?

Womit?

## Die 4 Gruppen:

- > Gesetzgeber
- > Product Owner
- > Anwender\*innen des Systems
  - > Personen aus wissenschaftlichen Einrichtungen bzw. Bildungseinrichtungen, die Texte verfassen und veröffentlichen wollen
  - > **Student\*innen**
  - > Lehrkräfte & Schüler\*innen
  - > Dozent\*innen
  - > Journalist\*innen
  - > Politiker\*innen
- > Entwickler\*innen
  - > Scrum-Master

**Priorisierung durch Entscheidungskriterien: 1. Macht, 2. Legitimität und 3. Dringlichkeit**

# Problemlösungen gestalten.

Warum?

Wer?

Wie?

Wann?

Wozu?

Womit?

## Unsere Idee: **Genderly**

→ Web-Anwendung zur Anpassung gegebener Texte hinsichtlich inklusiver Wortwahl

- > Nimmt Text mittels Eingabe auf, sucht für genderbare Wörter, korrekt gegenderte Alternativen
- > Eingabe und Wortalternativen werden als neuer Text angezeigt
- > Text kann vom Benutzer noch weiter angepasst oder herauskopiert und genutzt werden
- > Für die Textanalyse und Textanpassung werden Eingabewörter mit Einträgen einer Datenbank verglichen
  - Idee: Vermerkte Alternativwörter ersetzen genderbare Eingabewörter



# Nutzergewohnheiten ausmachen.

Warum?

Wer?

Wie?

Wann?

Wozu?

Womit?

- > Beim Verfassen wissenschaftlicher Texte
- > Reden halten
- > Verfassen von Gesetzesentwürfen
- > Verfassen von Artikeln

Allgemein ist es für Alltagssituationen grundlegend besser, sich korrekt/freundlich/höflich auszurücken. **Immer.**

# Einen tieferen Zweck erfüllen.

Warum?

Wer?

Wie?

Wann?

Wozu?

Womit?

- > Etablierung eines inklusiven Sprachstils
  - Beitrag zur **Geschlechtergleichheit** im Umgang miteinander
- > Nutzer\*innen können einfacher genderneutrale Sprache anwenden und beachten
- > Alle Personen werden angesprochen, niemand wird ausgeschlossen
- > Neutraler Ton / neutrale Ausdrucksweise erfahren und nutzen
- > Nutzer\*innen können genderneutralen Sprachstil einfacher erlernen

# Die Anwendung verfügbar machen.

Warum?

Mit unserer Web-Anwendung: **Genderly** ;)

Wer?

**Nutzer\*in fragt: Womit?**

- > Auch mit mobilen Endgeräten überall flexibel nutzbar
- > Hohe Gerätekompatibilität durch Einhaltung von Web-Standards
- > Barrierefreiheit durch einfache Gestaltung

Wie?

Wann?

Wozu?

**Entwickler\*in fragt: Womit?**

- > Java mit Spring Framework im Backend
- > SQLite-Datenbank als Persistenz für Wörter und ihre Alternativen
- > HTML5, JavaScript, Thymeleaf und Bootstrap im Frontend

**Womit?**

# Must Haves

Texterfassung, -analyse und Generierung eines genderneutralen Vorschlagstexts.

## Voraussetzungen:

- > Eingegebener Text ist in deutscher Sprache
- > Eingegebener Text ist syntaktisch korrekt
- > Wörter werden durch Alternativworte ersetzt.
- > Bedienung ist mittels einer Weboberfläche möglich



# Nice To Haves

## Textoptionen:

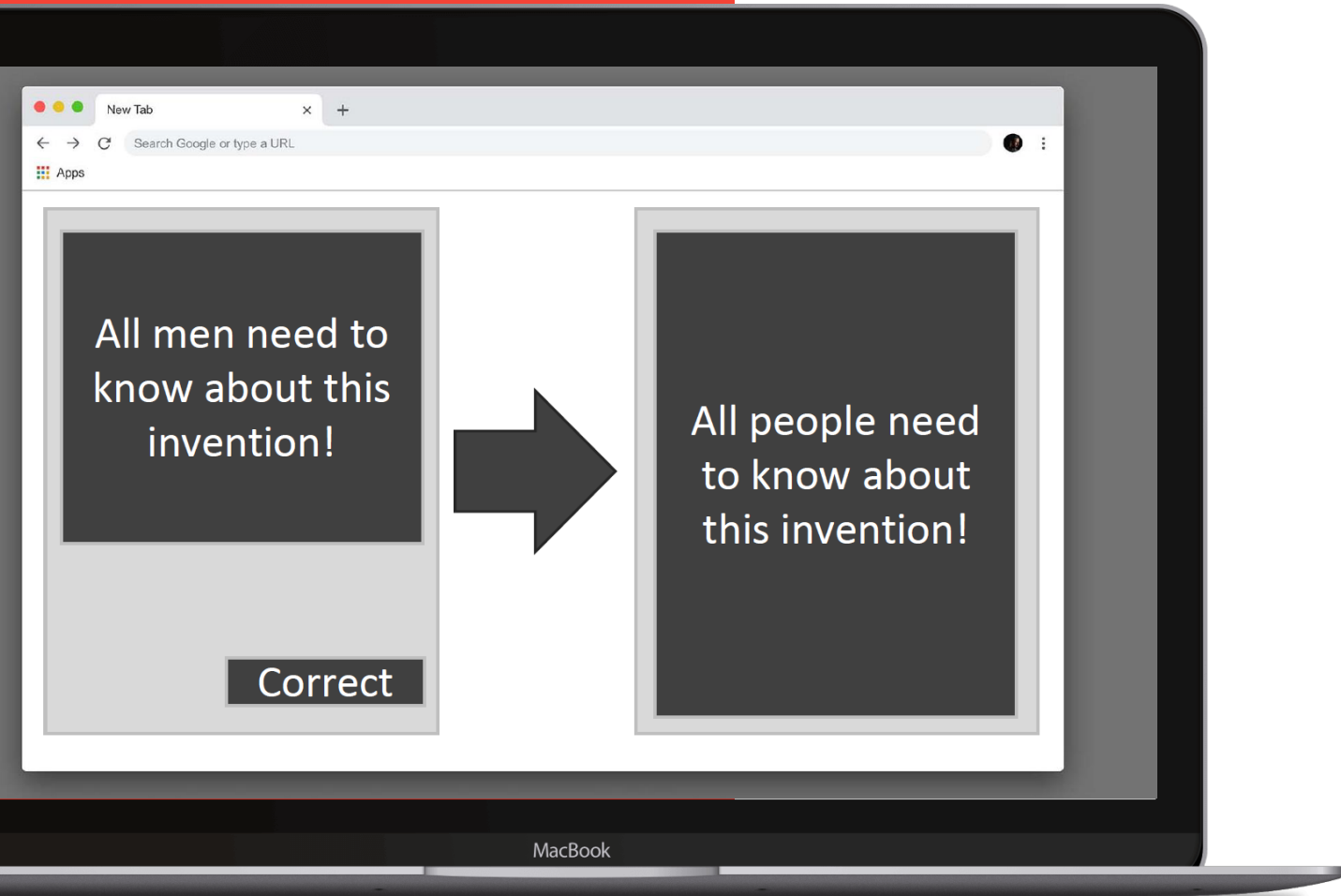
- > Wie soll gegendert werden?
- > Auswahlmöglichkeiten für Gendering-Stile, z.B.:
  - > Genderlose Ersatzworte,
  - > Schreibstil mit "\*"
  - > Schreibstil mit ":"

## Visuelle Optionen:

- > Themes: Dark/Light – Modus

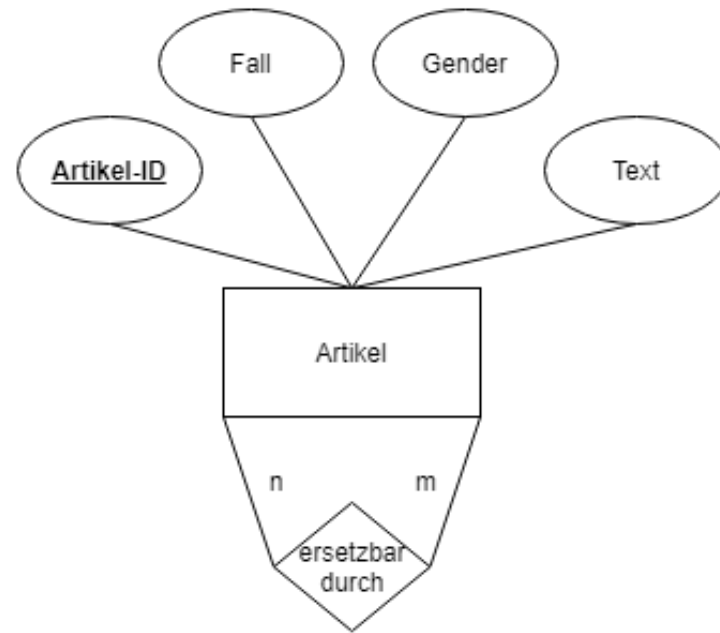
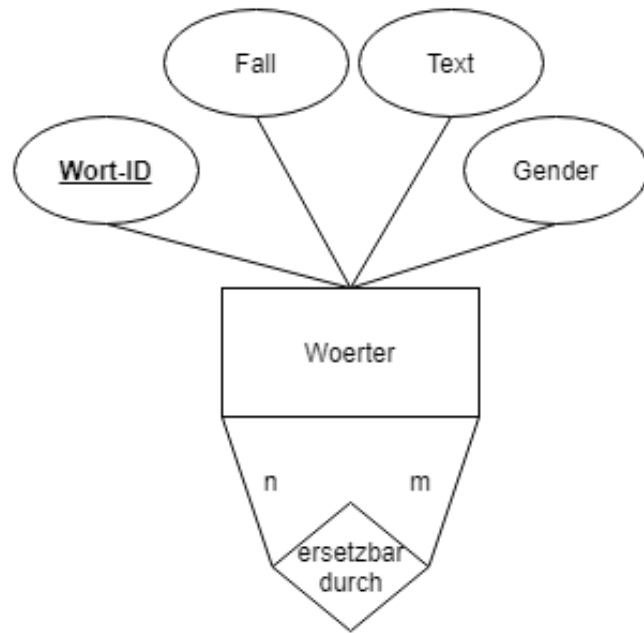
## Monetarisierung:

- > Werbeflächen vermieten
- > Premium-Modus  
(Begrenzung einer Wortanzahl aufheben)
- > Präsenz auf Social Media herstellen



# Mockups & Modelle

Die frühzeitige Verwendung von Mockups kann ein guter Beitrag dazu sein, den Fokus auf die Nutzererfahrung zu richten. Sie fördern die Kommunikation mit den Stakeholdern und bilden gleichzeitig innerhalb des Teams eine solide Grundlage für Diskussion und iterative Verbesserungsprozesse.

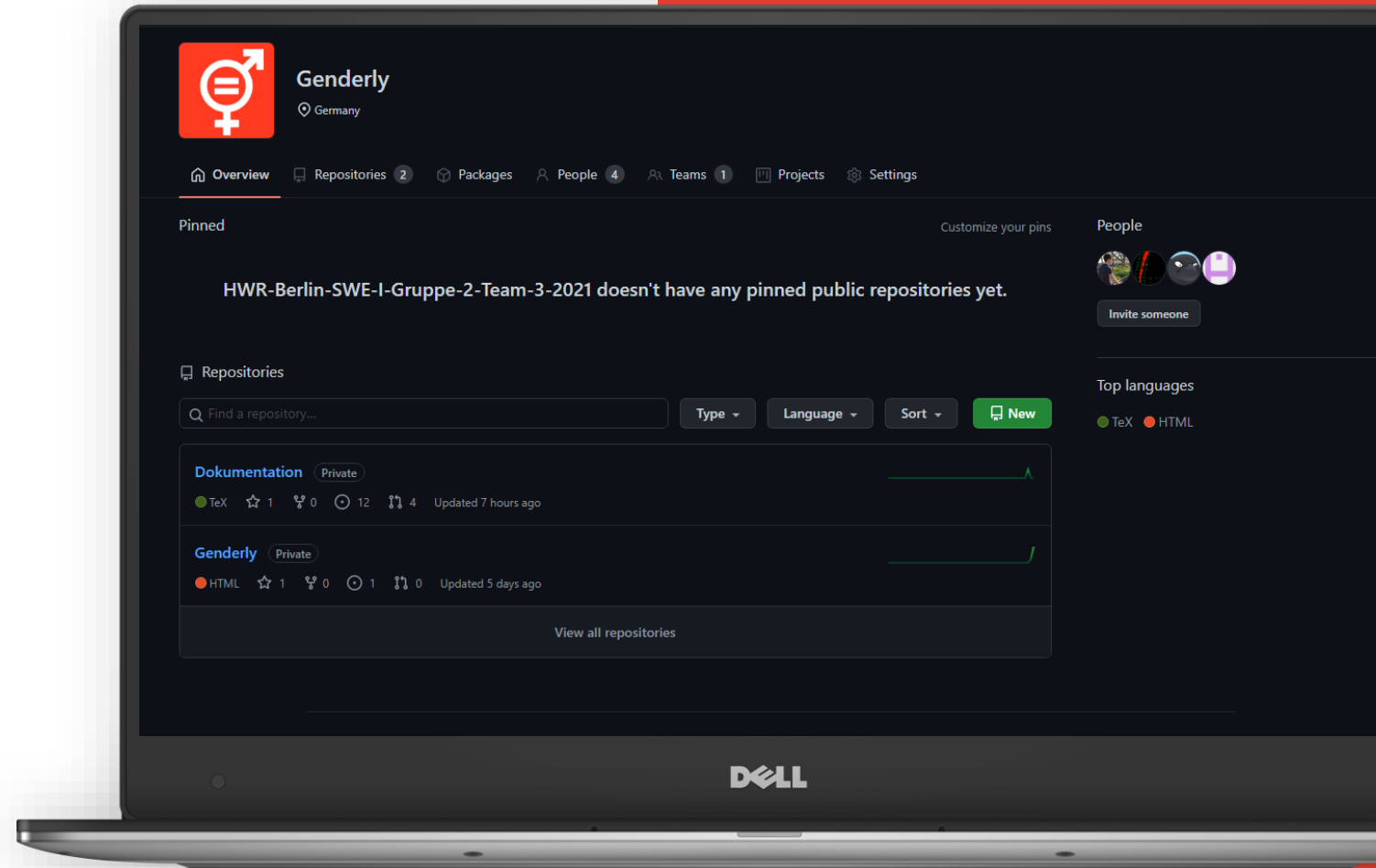


# Mockups & Modelle

Die frühzeitige Verwendung von Mockups ist ein guter Beitrag dazu, den Fokus auf die Nutzererfahrung zu richten. Sie fördern die Kommunikation mit den Stakeholdern und bilden gleichzeitig innerhalb des Teams eine solide Grundlage für Diskussion und iterative Verbesserungszyklen.

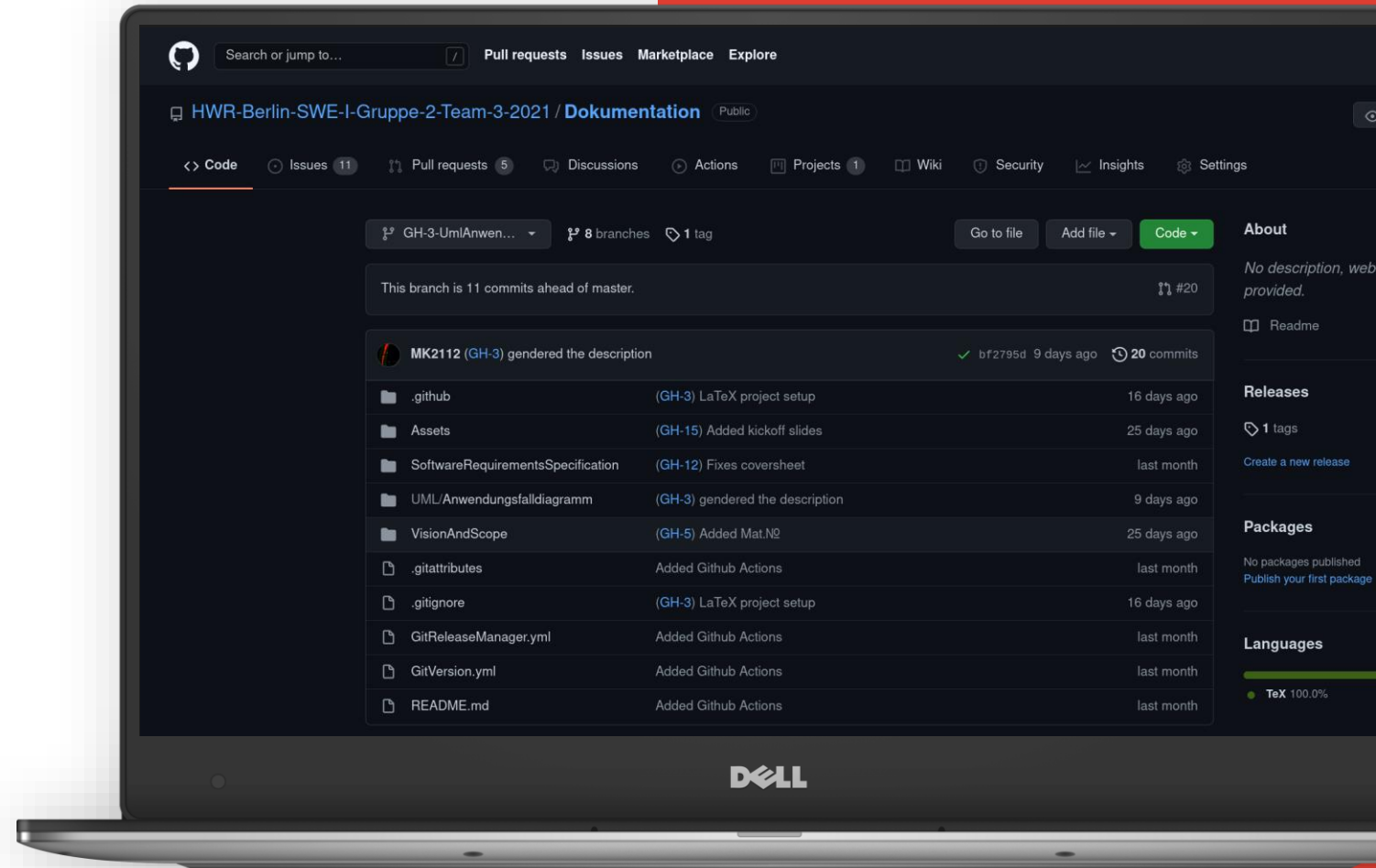
# GitHub Organisation

- > Verwaltung von Projekten
- > Arbeiten im Team
- > Öffentlich einsehbar



# Projekt- management

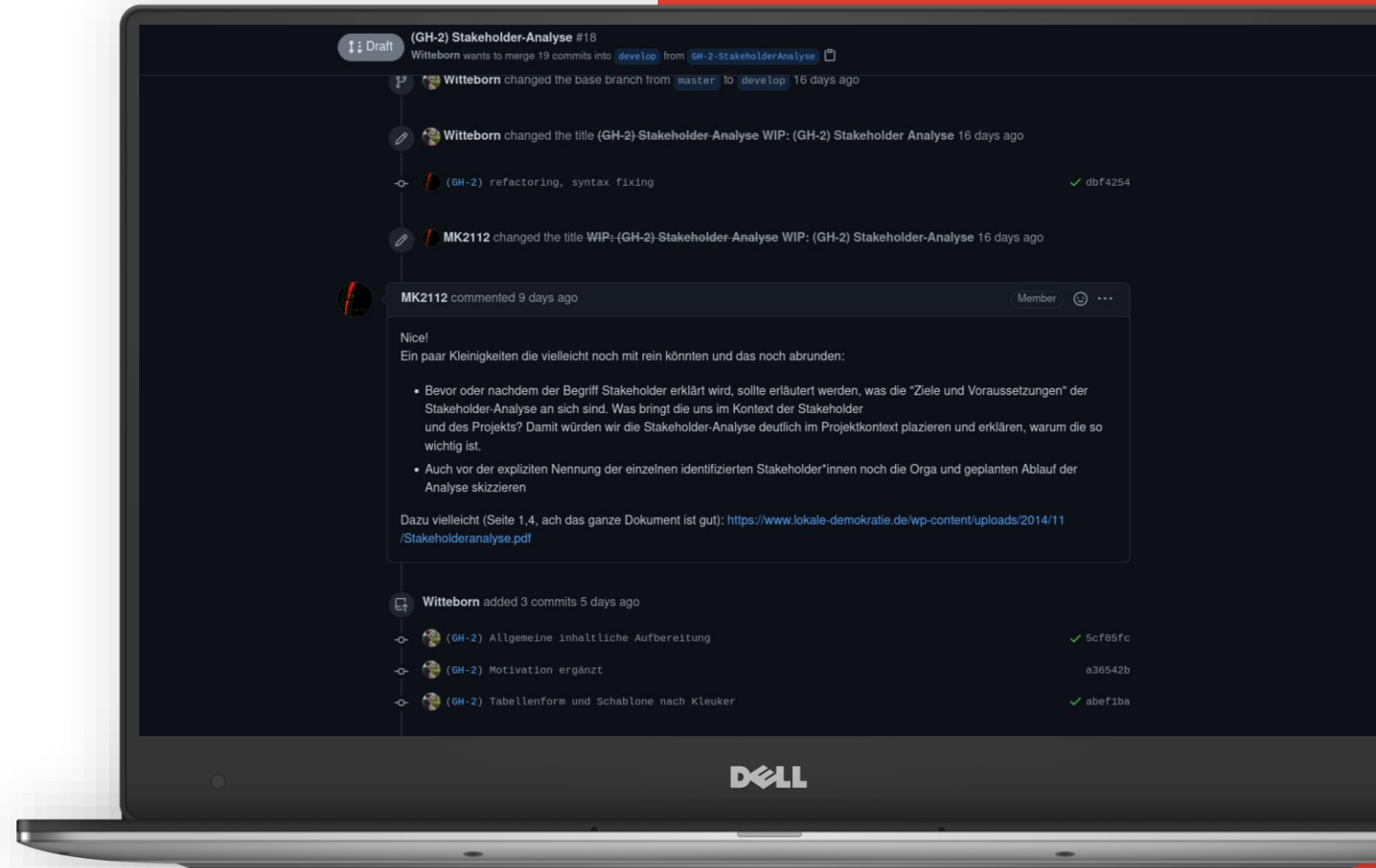
📁 .github	(GH-21) Migrating to GitHubBoilerplate 1.0.0
📁 Assets	(GH-15) Added kickoff slides
📁 SoftwareRequirementsSpecification	(GH-12) Fixes coversheet
📁 VisionAndScope	(GH-5) Added Mat.N2
📄 .gitattributes	Added Github Actions
📄 .gitignore	(GH-15) Specified .gitignore
📄 CODE_OF_CONDUCT.md	(GH-21) Migrating to GitHubBoilerplate 1.0.0
📄 CONTRIBUTING.md	(GH-21) Migrating to GitHubBoilerplate 1.0.0
📄 GitReleaseManager.yml	(GH-21) Migrating to GitHubBoilerplate 1.0.0
📄 GitVersion.yml	Added Github Actions
📄 LICENSE	(GH-21) Migrating to GitHubBoilerplate 1.0.0
📄 README.md	(GH-21) Migrating to GitHubBoilerplate 1.0.0
📄 SECURITY.md	(GH-21) Migrating to GitHubBoilerplate 1.0.0
📄 SUPPORT.md	(GH-21) Migrating to GitHubBoilerplate 1.0.0





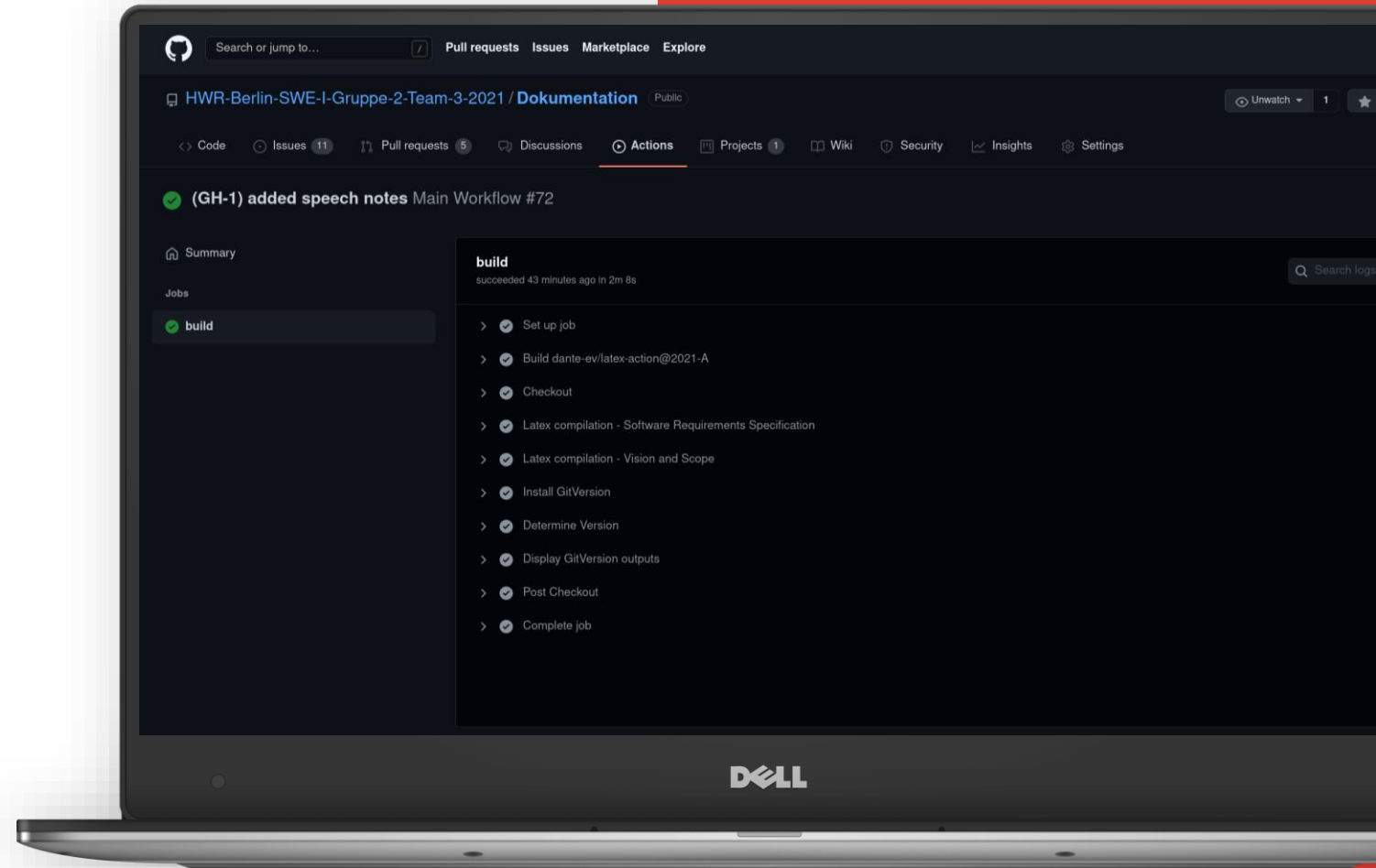
# GitHub Pull Requests

- > Vier-Augen-Prinzip durch Reviews
- > PRs haben immer ein 'Issue'
- > Zahlreiche Vorlagen, damit nichts vergessen wird



# GitHub Actions

- > Automatisiertes Bauen von Projekten
- > Automatisiertes Semantic Versioning (1.0.0)
- > Automatisch erzeugter Release mit Changelog
- > Qualitätssicherung

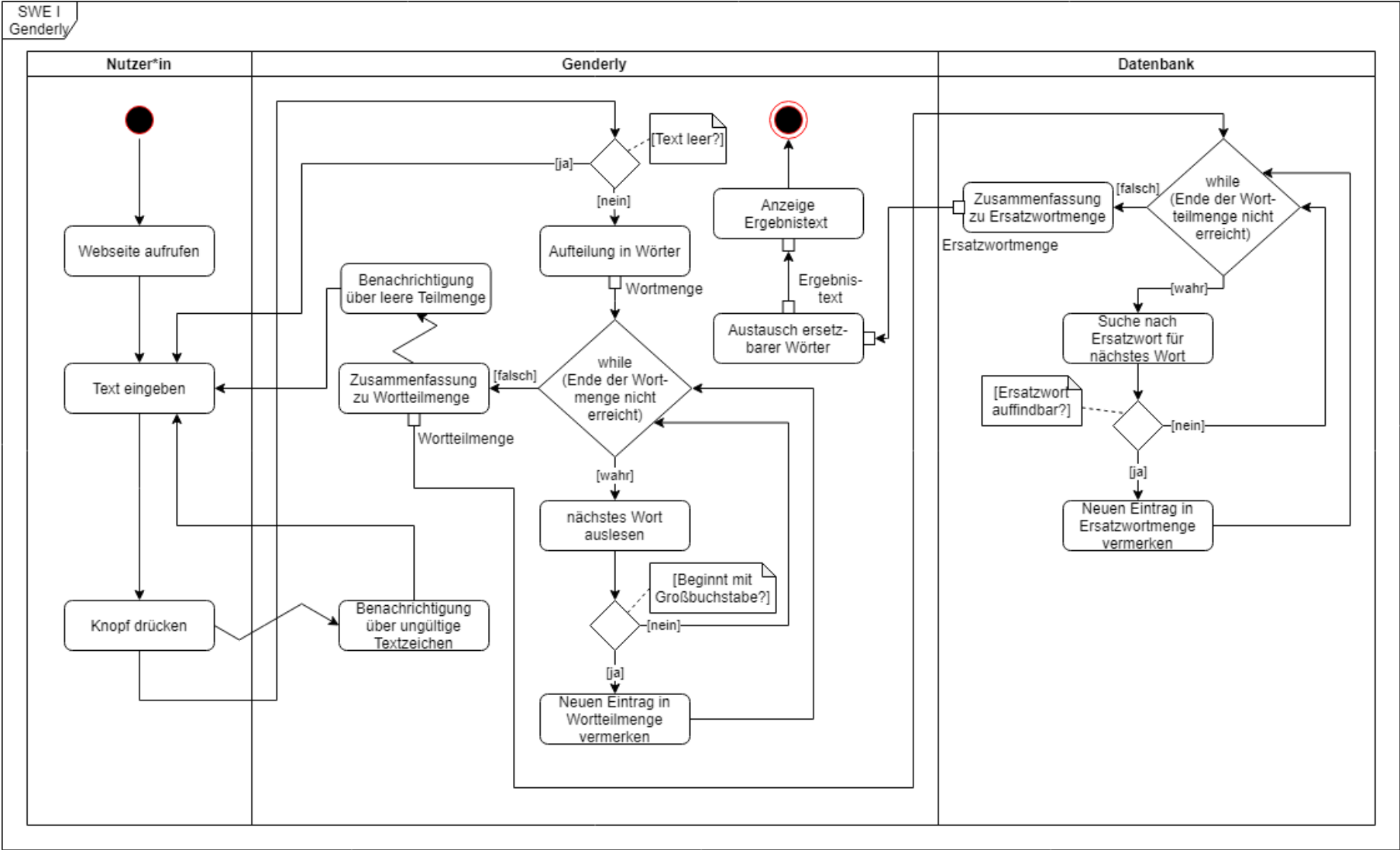


# 03 //

# Zusammenfassung

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <meta name="description" content="Gender your texts correctly.">
7
8   <!-- The appName might be underlined red, don't worry, it's IntelliJ's fault -->
9   <title th:text="${appName}"></title>
10
11   <link th:href="@{/css/bootstrap.min.css}" rel="stylesheet" />
12   <link th:href="@{/css/bootstrap-multiselect.css}" rel="stylesheet" />
13
14   <script type="text/javascript" th:src="@{/js/src/genderly.js}"></script>
15   <script type="text/javascript" th:src="@{/js/src/bootstrap.min.js}"></script>
16   <script type="text/javascript" th:src="@{/js/src/bootstrap-multiselect.min.js}"></script>
17 </head>
18 <body class="d-flex flex-column min-vh-100" style="background-color: rgba(236,236,236,255); height: 100%">
19   <!-- Thymeleaf replaces this during runtime: -->
20   <div th:replace="fragments/genderlyNavbar :: genderlyNavbar"></div>
21   <div class="container-fluid" style="margin-top: 8%;">
22     <!-- Don't yet fulfill functional -->
23     <div class="row justify-content-center">
24       <div class="col-4">
25         <label for="sel2" style="color: #e6301b; float: left;" ><b>Sprache:</b></label>
26         <select class="form-control" id="sel2" style="...">
27           <option value="0">Deutsch</option>
28         </select>
29       </div>
30       <div class="col-4">
31         <label for="sel3" style="color: #e6301b; float: left; width: 100%"><b>Gendering-Stil:</b></label>
32         <select class="form-control" id="sel3" style="color: black; margin-bottom: 20px; background-color:">
33           <option value="0">Wortersatz</option>
34         </select>
35       </div>
36     </div>
```

## Programmausführung



# Fazit

## GitHub-Organisation:

- > Public Foss Projekt
- > DevOps Environment mit GitHub Actions
- > Contribution Guidelines
- > Code of Conduct
- > Security Policy
- > Automatisiertes Semantic Versioning mit Milestones
- > Issue und Pullrequest Templates
- > Projektlizenz

## Projekt:

- > Frontend
- > Aktivitätsdiagramm
  - Konkrete Vorstellung, wie unser Produkt vorgeht
- > Erster Entwurf für die Struktur der Datenbank als ERM

# Ausblick

## **Technik-Aspekt:**

- > Erstellung des Backends mit Datenbank-Anbindung
- > Datenbank designen und befüllen

## **Projektmanagement-Aspekt:**

- > Vision und Strategie
- > Risiko-Analyse (fortlaufend)

**Vielen Dank für Ihre  
Aufmerksamkeit!**