

# Product Backlog

(Piet Helzel, 19.10.2023)

## a) User Stories

1. Als Händler möchte ich den Preis einer Waren-Einheit eingeben können, um den Gesamtwert der Bestellung ermitteln zu können.
2. Als Händler möchte ich die Anzahl der Einheiten einer Ware eingeben können, um den Gesamtwert der Bestellung ermitteln zu können.
3. Als Händler möchte ich den Umsatzsteuersatz anhand meines Länderkürzels durch ein Eingabefeld bestimmen können, damit ich mir die Steuersätze nicht merken muss.
4. Als Händler möchte ich den Umsatzsteuersatz auch über einen Button eingeben können, um die Eingabe zu erleichtern.
5. Als Händler möchte ich den Typ einer Ware angeben können, damit der korrekte Steuersatz angewandt werden kann.
6. Als Händler möchte ich, dass die App auf Basis der eingegebenen Daten den Gesamtpreis automatisch anzeigt, um möglichst Zeiteffizient zu sein.
7. Als Händler möchte ich, dass die App als Webanwendung verfügbar ist, damit ich möglichst flexibel in ihrer Nutzung bin.
8. Als Händler möchte ich, dass die App auch auf Smartphones korrekt angezeigt und nutzbar ist, damit ich sie auch mobil nutzen kann.
9. Als Händler möchte ich nach der Berechnung die Parameter noch verändern können, um das Vergleichen von Produkten zu erleichtern.
10. Als Händler möchte ich die berechneten Werte abspeichern können, um sie später referenzieren zu können und das Vergleichen von Produkten zu erleichtern.

## b) Strukturierung und Priorisierung

Die Strukturierung habe ich mit drei Klassen realisiert, in denen die User Stories eingeordnet werden können:

- **Elementare Funktionalitäten**, die für die Benutzung der App unabdingbar sind. Diese haben die höchste Priorität.
- **UI/UX**: Funktionen, die die Bedienung der App deutlich vereinfachen, indem sie z.B. sehr nützliche (aber nicht essentielle) Funktionen in das User Interface integrieren. Diese haben mittlere Priorität.
- **Nice to have**: Zusätzliche Funktionen, die praktisch aber rein optional für die Funktionalität der App sind. Die Priorität ist niedrig.

### Elementare Funktionalitäten (Hohe Priorität)

- #1: Eingeben der Warenpreise
- #2: Eingabe der Anzahl einer Ware
- #3: Eingeben des Umsatzsteuersatzes anhand des Länderkürzels
- #5: Eingabe des Warentyps

### **UI/UX (Mittlere Priorität)**

- #4: Eingabe des Länderkürzels über einen Button
- #6: Automatisches Anzeigen vom Gesamtpreis
- #7: Implementation als WebApp

### **Nice to have (Niedrige Priorität)**

- #8: Mobilfreundliches User-Interface
- #9: Verändern von Parametern nach der Eingabe
- #10: Abspeichern von berechneten Werten

## **c) Planung der Sprints**

### **Sprint 1**

In diesem Sprint werden die elementaren Funktionen implementiert. Zunächst wird noch kein User Interface entworfen, sondern eine einfache Konsolenanwendung, um die korrekte Berechnung der Werte zu implementieren.

In diesem Sprint werden User Stories 1, 2, 3 und 5 bearbeitet.

### **Sprint 2**

In Sprint 2 werden die Komponenten des User Interface designet und als WebApp implementiert. Dazu gehören Eingabefelder, Textlabels und Buttons, über die der Benutzer mit dem Programm interagieren kann.

In diesem Sprint werden User Stories 4, 6 und 7 realisiert.

### **Sprint 3**

Nun werden die optionalen Features mit geringer Priorität implementiert. Da das Grundgerüst der Anwendung steht, sind diese Features mit relativ geringem Aufwand implementierbar.

Es werden in diesem Sprint die User Stories 8, 9 und 10 bearbeitet.

### **Sprint 4**

Im letzten Sprint des ersten Releases wird die App auf Bugs und Fehler überprüft und diese behoben. Features werden in diesem Sprint nicht mehr hinzugefügt. Generell wird ein gewisses Maß an Tests schon in den vorherigen Sprints durchgeführt, damit das Fundament nicht zu instabil ist.

## **d) Planung einzelner Sprints**

### **Planung Sprint 1**

Tasks:

- Sprint 1
  - Implementierung der Berechnung: *Entwickler*
  - Implementierung der Eingabemasken für Waren und Anzahl: *Entwickler*
  - Implementierung der Eingabemaske für Länderkürzel: *Entwickler*
  - Testen, ob die Eingabemaske nur valide Länderkürzel akzeptiert: *Tester*
  - Eingabemaske für Warentyp implementieren: *Entwickler*

- Testen, ob die Eingabemaske nur valide Warentypen akzeptiert: *Tester*
- Sprint 2
  - Design des allgemeinen Interfaces entwerfen: *GUI-Designer*
  - Implementierung der Eingabemasken in der WebApp: *Entwickler*
  - Implementierung der Ländereingabe als Button: *Entwickler*
  - Automatische Anzeige des Preises implementieren: *Entwickler*
  - Integration Tests entwerfen und ausführen, um die Zusammenarbeit zwischen UI und Logik zu gewährleisten.