



Projektkonzeption und -realisierung

Konzeption und prototypische Implementierung eines B2B-Webshops

Pflichtenheft

Wintersemester 2024/2025

Semester: 5

Kurs: WI22A-AKI

Betreuer:

Prof. Dr. Alexandros Nanopoulos

Prof. Dr. Dirk Palleduhn

Mosbach, den 7. Februar 2025

0.1 Projektteilnehmer

Name	Vorname	Matrikelnummer
Christ (Projektleiter)	Colin	4359760
Spatzek (stv. Projektleiter)	Steffen	3854031
Arnold	Daniel	8627710
Bamberger	Bastian	2923282
Denz	Andreas	5428962
Jeevakanthan	Milan	9892846
Kanjo	Alan	9795498
Kunz	Paul	2338290
Schreck	David	3533132
Strohm	Julian	7956706
Swoboda	Timo	4388948
Tomanek	Lukas	5985858
Väth	Luis	8122258
Weis	Noah	1555500

Tabelle 1: Projektteilnehmer und Matrikelnummern

0.2 Organigramm

Dieses Kapitel beschreibt die Organisationsstruktur des Projekts *B2B-Webshop* und stellt die Rollen und Verantwortlichkeiten der Teammitglieder in einem übersichtlichen Organigramm (siehe Abb. 1) dar. Dabei wird zwischen dem Projektleiter und dem Co-Projektleiter unterschieden, die jeweils eigene Teams aus Entwicklern und Fachkräften führen.

Das Organigramm dient als visuelle Orientierungshilfe, um die Aufgabenverteilung und Kommunikationswege im Projekt klar darzustellen. Ziel ist es, die Zusammenarbeit zu fördern und sicherzustellen, dass alle Beteiligten ihre Verantwortlichkeiten kennen.

0.3 Teamstruktur

Das Kapitel Teamstruktur gibt einen tabellarischen Überblick über die Zuordnung der Teammitglieder zu den zentralen Verantwortungsbereichen des Projekts. Die Rollen sind in die Kategorien Organisation, Entwicklung, Solution Architect und User Experience unterteilt (siehe Tabelle 2).

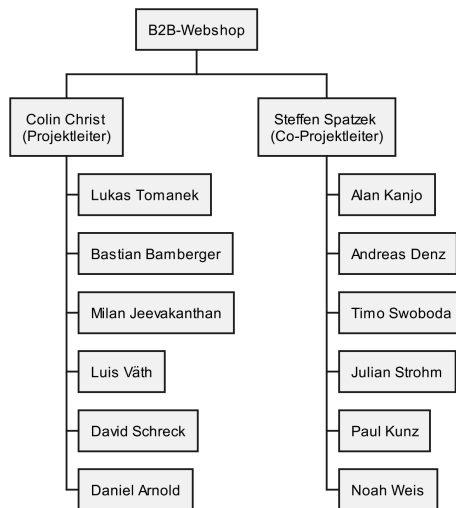


Abbildung 1: Organigramm der Projektgruppe des B2B-Webshops

Die Tabelle verdeutlicht die Zuordnung der Teammitglieder zu ihren jeweiligen Verantwortungsbereichen und unterstreicht die multidisziplinäre Zusammenarbeit innerhalb des Projekts. Die Hervorhebung einzelner Mitglieder zeigt deren Führungs- oder Spezialaufgaben, während die klare Struktur die Nachvollziehbarkeit und Effizienz innerhalb des Teams unterstützt.

Organisation	Entwicklung	Solution Architect	User Experience
Colin Christ	Steffen Spatzek	Steffen Spatzek	Alan Kanjo
Lukas Tomanek	Paul Kunz	Alan Kanjo	Lukas Tomanek
David Schreck	Timo Swoboda	Paul Kunz	David Schreck
Bastian Bamberger	Alan Kanjo	Luis Väh	Julian Strohm
Milan Jeevakanthan	Julian Strohm		Bastian Bamberger
Luis Väh	Andreas Denz		Milan Jeevakanthan
Daniel Arnold	Noah Weis		

Tabelle 2: Verantwortungsbereiche der Projektmitglieder des B2B-Webshops

Inhaltsverzeichnis

0.1	Projektteilnehmer	2
0.2	Organigramm	2
0.3	Teamstruktur	2
1	Zielsetzung	6
1.1	Musskriterien	7
1.2	Wunschkriterien	9
1.3	Abgrenzungskriterien	10
2	Produkteinsatz	12
2.1	Anwendungsbereich	12
2.2	Zielgruppen	12
2.3	Betriebsbedingungen	12
3	Produktübersicht	13
4	Detaillierte Produktfunktionen	14
4.1	User Stories	15
4.2	Funktionale Anforderungen	15
5	Datenstruktur und verwaltete Informationen	16
5.1	Produktdaten	16
5.2	Bestandsdaten	17
5.3	Kundendaten	17
6	Qualitätsanforderungen	18
7	Systemarchitektur	20
7.1	Komponenten der Architektur	20
7.2	Containerisierung und Deployment	21
8	Datenmodell	23
9	Schnittstellendefinition (API)	24

10 Benutzungsoberflächen	26
10.1 Gestaltungsrichtlinien	26
10.2 Verwendete Farben, Schriften und Layout	27
10.2.1 Farben	27
10.2.2 Schriften	27
10.2.3 Layout	27
10.3 Verzicht auf Mockups	28
10.4 Zusammenfassung	28
11 Nicht-funktionale Anforderungen	29
12 Technische Produktumgebung	30
12.1 Software	30
12.2 Hardware	31
12.3 Orgware	31
13 Konzeption der KI-Komponente	32
13.1 Idee	32
13.2 Zielgruppenanalyse	32
13.3 Funktionen und Anwendungsfälle	32
13.4 Umsetzungsmöglichkeiten	32
13.4.1 Eigenes Large Language Model (LLM)	33
13.4.2 Chatbot-Anbieter	33
13.5 Anforderungen an den Chatbot	34
13.6 Datenanforderungen	34
13.7 Auswahl des Anbieters	34
13.8 Bot-Instructions	35
13.9 Datenintegration in Botpress	35
13.10 Integration des Chatbots	36
13.11 Bearbeitungsablauf einer Kundenanfrage	36
14 Projektorganisation	37
14.1 Projektmethodik	37
14.2 Rollenverteilung	39
14.2.1 Organisatorische Leitung	39
14.2.2 Organisation	39
14.2.3 Entwicklung	40
14.2.4 Solution Architecture	41
14.2.5 User Experience (UX)	41
14.2.6 Zusammenfassung	41

1 Zielsetzung



Abbildung 1.1: Firmenlogo des Hot Hardware Hub

Der **Hot Hardware Hub** ist ein fiktives Unternehmen, das im Rahmen dieses Projektes gegründet wurde, um hochwertige und moderne IT-Hardware speziell für Geschäftskunden (B2B) anzubieten. Ziel ist es, einen innovativen Webshop zu entwickeln, der es Unternehmen ermöglicht, die benötigte Hardware schnell, einfach und bequem online zu bestellen.

Im Mittelpunkt des Systems steht das Ziel, den Kunden ein erstklassiges Einkaufserlebnis zu bieten. Eine intuitive Benutzeroberfläche sowie ein klar strukturierter und gut durchsuchbarer Produktkatalog ermöglichen es den Nutzern, mit nur wenigen Klicks die passenden Produkte zu finden und die Bestellungen mühelos abzuschließen.

Die Systementwicklung reagiert auf die steigende Nachfrage nach digitalen Beschaffungslösungen im IT-Bereich. Viele Unternehmen suchen nach effizienten Möglichkeiten, schnell und unkompliziert hochwertige Hardware zu beschaffen. Der Webshop des **Hot Hardware Hub** stellt hierfür eine zuverlässige und benutzerfreundliche Plattform bereit, die die IT-Beschaffung deutlich vereinfacht. Das Ziel ist es, nicht nur Zeit und

Aufwand zu sparen, sondern auch die Zufriedenheit und Effizienz der Geschäftskunden nachhaltig zu steigern.

1.1 Musskriterien

Produktkatalog

- Kunden können den Produktkatalog mit der IT-Hardware einsehen.
- Produkte werden in Kategorien darstellbar angezeigt.
- Produkte müssen die wesentlichen Merkmale (Preis, Menge, Lagerbestand und Produktdetails) für den Kunden sichtbar machen.
- Produkte können über eine Suchfunktion mit Filtermöglichkeiten gezielt gefiltert werden.

Benutzerverwaltung

- Kunden müssen sich registrieren und sich mit ihren Daten anmelden können.
- Kunden müssen ihre Benutzerdaten einsehen und verändern können.
- Kunden müssen ihr Konto deaktivieren bzw. löschen können.
- Der Anmeldeprozess im Webshop muss mit einer sicheren Authentifizierungsmethode gestaltet werden.

Bestellprozess

- Kunden müssen ihre Produkte in den Warenkorb legen und diesen einsehen können.
- Der Kunde muss eine Bestellübersicht vor dem finalen Abschluss sehen.
- Der Kunde muss in minimalen Schritten zum erfolgreichen Kaufabschluss geführt werden.

- Kunden müssen im Kundenbereich getätigte Bestellungen und deren Status einsehen können.

Zahlung und Rechnungsstellung

- Dem Kunden müssen gängige Zahlungsmethoden verfügbar gemacht werden (z. B. Kauf auf Rechnung).
- Der Kunde muss nach erfolgreichem Abschluss eine Rechnung per E-Mail erhalten oder diese im Kundenbereich einsehen können.

Shop-Betreiber

- Administratoren können über ein Dashboard Produkte und zugehörige Daten erstellen, bearbeiten und löschen.
- Der Shop-Betreiber kann Produktbilder und Dokumente mit den Produktseiten verknüpfen.

KI-Komponente

- Es soll eine KI-Komponente integriert werden, die den Kunden per Chat beim Einkauf unterstützt.
- Über eine ML-Komponente soll erreicht werden, dass Kundenpreise je nach Einkaufsvolumen bzw. -verhalten individuell rabattiert werden.

Technische Aspekte

- Der Webshop soll plattformunabhängig von den gängigsten Geräten aufgerufen werden können.
- Der Webshop soll Anfragen schnell abarbeiten und schnell erreichbar sein.
- Die Webapplikation soll eine intuitive Bedienung aufweisen.
- Der B2B-Shop soll durch steigende Produktmengen schnell skalierbar sein.

1.2 Wunschkriterien

Benutzerverwaltung

- Es soll ermöglicht werden, dass eine eigene Einkaufsgruppe für einen gewissen Kundenkreis erstellt werden kann.
- Der Kunde soll mehr als nur einen Warenkorb anlegen, befüllen und speichern können.
- Mehrere Benutzerkonten oder -gruppen für ein Unternehmen sollen unterstützt werden.
- Es sollen Kennzahlen für einen bestimmten Kunden oder eine Einkaufsgemeinschaft bereitgestellt werden.
- Zwei-Faktor-Authentifizierung oder eine No-Password-Authentication (z. B. über Passkeys) soll dem Kunden ermöglicht werden.

Produktkatalog

- Kundenbenachrichtigungen bei wieder verfügbaren Artikeln.
- Eine noch detailliertere Filterfunktion bei der Produktsuche.
- Produktvergleich-Funktion zwischen zwei oder mehreren Produkten.
- Zeitlich begrenzte Aktionen oder individuelle Gutscheine.

Sicherheitsaspekte

- Das System muss alle Aspekte der DSGVO erfüllen.
- Es ist eine dem Stand der Technik entsprechende Datenverschlüsselung zu verwenden.

Bestellprozess

- Ein wiederkehrendes Bestellmodell soll angeboten werden.

- Individuelle Mengenrabatte je nach Menge oder Einkaufsvolumen in einem bestimmten definierten Zeitraum.

KI-Komponente

- Auf Basis von Wunschlisten, Kaufhistorie oder neuen Artikeln im Sortiment werden KI-gestützte Produktempfehlungen gegeben.

Technische Aspekte

- Darstellung von Daten wie Traffic, Besucherzahlen und Kundenaktionen in einem Dashboard für den Shopbetreiber.
- Logging und Monitoring des Webshops.

1.3 Abgrenzungskriterien

1. Funktionale Abgrenzungen:

- a) **Umfang des Produktangebots:** Der Shop beschränkt sich auf Hardwareprodukte, keine Dienstleistungen.
- b) **Kein Marktplatzmodell:** Der Shop dient nicht als Plattform für andere Anbieter.

2. Technische Abgrenzungen:

- a) **Keine mobile Anwendung:** Es wird keine App entwickelt. Der Shop soll als Webservice genutzt werden.
- b) **Keine Mehrsprachigkeit:** Der Shop wird ausschließlich in deutscher Sprache betrieben.

3. Rechtliche Abgrenzungen:

- a) **Keine rechtliche Anpassung für Nicht-EU-Länder:** Der Shop wird nicht an Steuer- und Rechtssysteme außerhalb der EU angepasst.

4. Gestalterische Anpassung:

1 Zielsetzung

- a) **Keine vollständige Barrierefreiheit:** Der Shop wird nicht vollständig barrierefrei entwickelt (z. B. keine Optimierung für Screenreader oder spezielle Kontrasteinstellungen).

2 Produkteinsatz

2.1 Anwendungsbereich

Der Anwendungsbereich des Webshops umfasst den Verkauf von IT-Hardware an Geschäftskunden. Die Kunden erhalten Zugang zum Shop und können dort die benötigte Hardware bestellen.

2.2 Zielgruppen

Der B2B-Onlineshop für IT-Hardware richtet sich vor allem an drei Hauptzielgruppen: IT-Dienstleister, Großunternehmen und Konzerne sowie Wiederverkäufer:

- IT-Dienstleister und Systemhäuser benötigen regelmäßig Hardware wie Server, Netzwerktechnik und Speichersysteme für den Aufbau und die Wartung von IT-Infrastrukturen bei ihren Kunden. Diese Zielgruppe verlangt nach großen Bestellmengen, maßgeschneiderten Lösungen und zuverlässiger Lieferung.
- Große Unternehmen und Konzerne beschaffen IT-Hardware für ihre Mitarbeiter und Abteilungen. Sie benötigen eine breite Produktpalette und einfache Bestellprozesse.
- Reseller (Wiederverkäufer) hingegen kaufen IT-Produkte in großen Mengen ein, um sie weiterzuverkaufen. Sie benötigen wettbewerbsfähige Preise, detaillierte Produktinformationen sowie eine effiziente Bestell- und Lieferabwicklung.

2.3 Betriebsbedingungen

Die Anwendung läuft auf einem Webserver in einer eigenen containerisierten Docker-Umgebung. Sie läuft rund um die Uhr, mit Ausnahme von Wartungsfenstern.

3 Produktübersicht

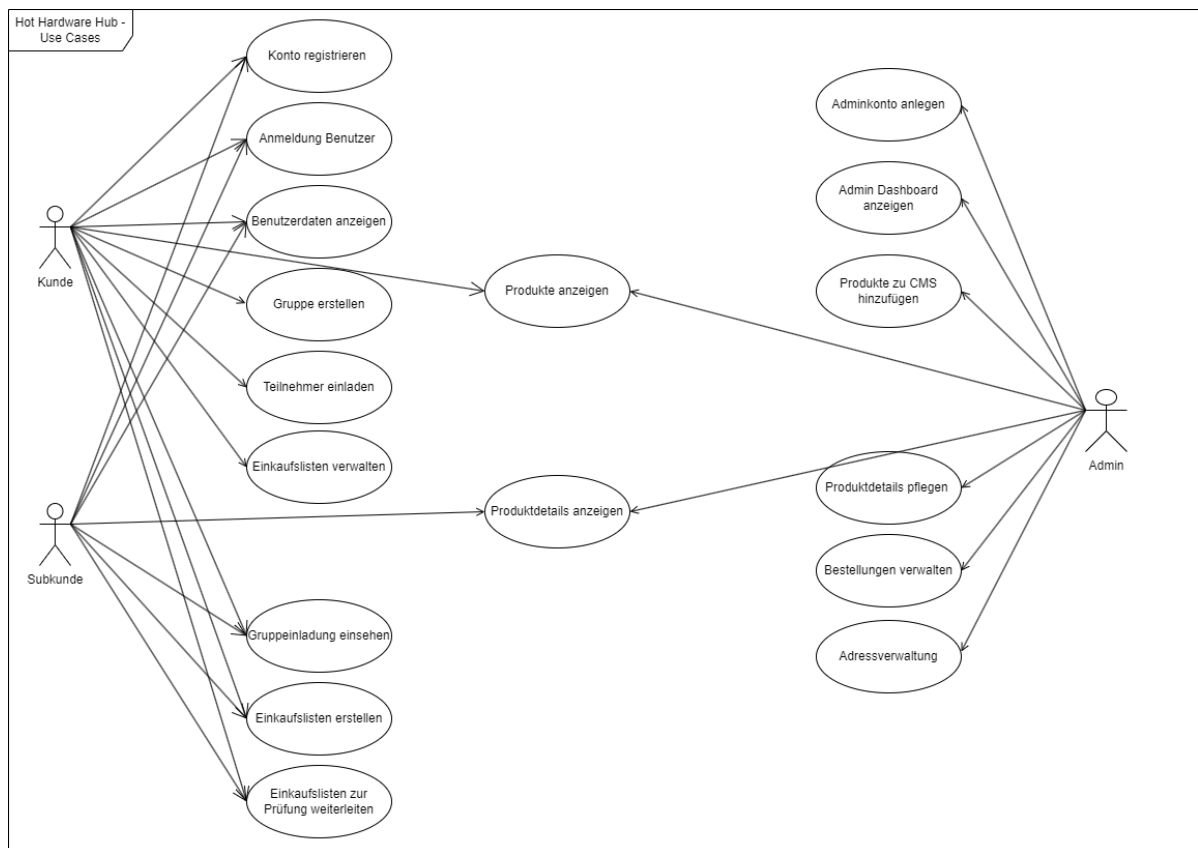


Abbildung 3.1: Use Case-Diagramm

Für das Produkt *Hot Hardware Hub* sind drei Akteure vorgesehen:

Admin Verwaltung und Pflege des Kundenstamms und des Produktkatalogs

Kunde Verantwortlicher Einkäufer des Kunden, der weitere Mitarbeiter für seinen Einkäuferstamm berechtigen kann

Subkunde Berechtigter Einkäufer eines Kunden, der Produkte im *Hot Hardware Hub* einkauft

4 Detaillierte Produktfunktionen

F1 Anmeldung

F1.1 Kunde kann sich registrieren

F1.2 Kunde kann sich nach Registrierung anmelden

F1.3 Kunde kann seine Benutzerdaten anzeigen

F1.4 Kunde kann gezielt ein Produkt in der Detailansicht öffnen

F2 Anpassung Benutzerdaten (Subkunde)

F2.1 Subkunde kann Benutzerdaten ändern

F2.2 Subkunde kann Passwort zurücksetzen

F2.3 Subkunde kann Benutzerkonto löschen

F3 Admin-Benutzer

F3.1 Adminkonto registrieren

F3.2 Admindashboard anzeigen

F3.3 Produkte zu CMS hinzufügen

F3.4 Produktdetails pflegen

F3.5 Erweiterung Item-Model

F4 Produktkatalog

F4.1 Kunde kann Produktkatalog ansehen

F4.2 Kunde kann im Produktkatalog nach Suchkriterien suchen

F4.3 Kunde kann per Filterfunktion gewisse Artikel ausblenden/einblenden

F4.4 Kunde kann gezielt ein Produkt in der Detailansicht öffnen

4.1 User Stories

An dieser Stelle sei auf das Scrum-Board verwiesen, das unter folgendem Link zu finden ist:

<https://tree.taiga.io/project/ssptzk-b2b-webshop>

Damit werden Redundanzen in der Dokumentation vermieden.

4.2 Funktionale Anforderungen

Die funktionalen Anforderungen des Webshops orientieren sich an gängigen Geschäftsfällen, um die Prozesse der IT-Beschaffung effizient abzubilden. Dazu zählen unter anderem:

- **Benutzerverwaltung:** Registrierung, Anmeldung und Verwaltung von Geschäftskunden sowie Unterstützung von Einkäufergruppen bei Freigabeprozessen.
- **Produktverwaltung:** Hinzufügen, Ändern und Löschen von Produkten durch Administratoren.
- **Warenkorb- und Bestellprozess:** Produkte in den Warenkorb legen, Bestellungen aufgeben und Bestellhistorien einsehen.
- **Such- und Filterfunktionen:** Detaillierte Suche und Filterung nach Produktkategorien, Eigenschaften und Verfügbarkeit.
- **E-Mail-Benachrichtigungen:** Automatisierte E-Mails für Bestellbestätigungen, Rückmeldungen und Freigabeprozesse.

Diese Anforderungen decken die zentralen Funktionen ab, die für einen effizienten Betrieb des Webshops und die Erfüllung der Geschäftsziele notwendig sind.

5 Datenstruktur und verwaltete Informationen

Dieses Kapitel beschreibt die strukturierten Daten, die im Webshop verarbeitet und verwaltet werden. Dazu gehören Produktdaten, Bestandsinformationen sowie Kundendaten, die für die Geschäftsprozesse relevant sind.

5.1 Produktdaten

Die folgenden Informationen werden für jedes Produkt im Webshop gespeichert:

- **Bezeichnung:** Produktname (z. B. Dell Latitude 5520)
- **Kurze Beschreibung:** Prägnante Angabe des Produkts (z. B. Hochwertiger Business-Laptop mit Intel i7)
- **Ausführliche Beschreibung:** Detaillierte Produktspezifikationen, z. B.: Intel Core i7-1165G7, 16 GB RAM, 512 GB SSD, 15,6-Zoll-Display, Windows 11 Pro, geeignet für anspruchsvolle Geschäftsanwendungen.
- **Tatsächlicher Verkaufspreis:** 1.450 €
- **Herstellerinformationen:**
 - Herstellername (z. B. Dell)
 - Modellnummer (z. B. Latitude 5520)

5.2 Bestandsdaten

Folgende Bestandsinformationen werden verwaltet:

- **Verfügbare Anzahl im Lager:** 150 Einheiten von Artikel X
- **Produkte im Warenkorb:** 10 Artikel X in 5 verschiedenen Warenkörben
- **Bestellte, aber noch nicht gelieferte Artikel:** 20 Artikel X sind in ausstehenden Lieferungen in Bearbeitung

5.3 Kundendaten

Für registrierte Geschäftskunden werden die folgenden Daten gespeichert:

- **Name und Firmenname:** Jürgen IT GmbH
- **Liefer- und Rechnungsadresse:** Hauptstraße 15, 74821 Mosbach
- **E-Mail-Adresse:** juergenIT@web.de
- **Bisheriges Einkaufsvolumen:** 25.000 €
- **Aktueller Warenkorb:** 3 Artikel (z. B. 2 Monitore und 1 Drucker)
- **Einkaufsgruppen und Freigabeprozesse:** Mitarbeiter mit entsprechender Berechtigung haben Zugriff auf den Bestellverlauf und die Bestellliste.

6 Qualitätsanforderungen

In diesem Kapitel werden die Qualitätsmerkmale des zu entwickelnden Produkts und deren erforderliche Qualitätsstufen definiert. Die Qualitätsmerkmale müssen in operationalisierter Form vorliegen.

1. Benutzbarkeit (Verständlichkeit)

Die Benutzeroberfläche des Webshops muss klar strukturiert und intuitiv verständlich sein.

- Geschäftskunden sollen ohne lange Einarbeitungszeit schnell die benötigte Hardware finden und bestellen können.
- Dies erfordert eine durchdachte Navigation, verständliche Produktbeschreibungen und eine optimierte Kaufabwicklung.
- Der Webshop muss ein konsistentes UI/UX-Design aufweisen und mit gängigen B2B-Shopsystemen vergleichbar sein, um die Usability zu maximieren.

2. Effizienz (Zeitverhalten)

Da Geschäftskunden effiziente Beschaffungslösungen benötigen, muss der Webshop schnelle Ladezeiten und eine reibungslose Performance gewährleisten.

- Produktseiten, Suchanfragen und der Checkout-Prozess sollen innerhalb von **maximal zwei Sekunden** geladen werden.
- Verzögerungen beim Einkaufsprozess sind zu minimieren, um Kaufabbrüche zu vermeiden.
- Ein Caching-Mechanismus und eine performante Datenbankanbindung sind erforderlich, um Lastspitzen abzufangen.

3. Interoperabilität

Die angebotene Hardware muss sich nahtlos in die bestehende IT-Infrastruktur der Geschäftskunden integrieren lassen.

- Der Webshop muss umfangreiche Filter- und Suchfunktionen bieten, um eine gezielte Produktauswahl zu ermöglichen.
- Kunden sollen nach Betriebssystemen, Ports, Dockingstation-Kompatibilität und anderen relevanten Kriterien filtern können.
- Eine API-Schnittstelle soll bereitgestellt werden, um die Anbindung an externe ERP-Systeme zu ermöglichen.

4. Übertragbarkeit (Installierbarkeit)

IT-Hardware für Unternehmen muss oft einfach zu installieren sein, insbesondere Server, Netzwerktechnik oder Arbeitsplatzlösungen.

- Der Webshop muss umfassende technische Spezifikationen, Treiberinformationen und Installationsanleitungen bereitstellen.
- Plug-and-Play-Funktionalitäten sollten klar ausgewiesen werden.
- Kompatibilitätsinformationen mit anderen Systemen sind transparent darzustellen.

5. Zuverlässigkeit (Wiederherstellbarkeit)

Die IT-Beschaffung muss auch bei Problemen reibungslos funktionieren.

- Der Webshop muss Rückgabeoptionen, Garantiebedingungen und Service Level Agreements (SLAs) klar und verständlich kommunizieren.
- Ein robustes Bestell- und Transaktionsmanagementsystem soll sicherstellen, dass Bestellungen auch im Falle technischer Probleme nicht verloren gehen.
- Im Fehlerfall muss eine automatische Wiederherstellung von abgebrochenen Bestellprozessen gewährleistet sein.

7 Systemarchitektur

Die nachfolgende Abbildung zeigt die Systemarchitektur des Webshops für den *Hot Hardware Hub*. Der gesamte Aufbau basiert auf einer containerisierten Umgebung, die mithilfe von *Dokploy* auf einem VPS bereitgestellt wird. Die Webshop-Komponenten, Datenbank sowie der Mailservice werden in einem separaten Container ausgeführt, um eine klare Trennung der Komponenten und eine bessere Skalierbarkeit zu gewährleisten.

7.1 Komponenten der Architektur

1. Traefik als Reverse Proxy und Loadbalancer

Traefik dient als zentrale Schnittstelle für den Datenverkehr zwischen den Clients und dem System. Es übernimmt die Verteilung der Anfragen an die entsprechenden Komponenten und ermöglicht eine sichere Kommunikation.

2. Frontend – Angular und NGINX

Die Benutzeroberfläche des Webshops wurde mit Angular realisiert. Diese wird über einen NGINX-Webserver ausgeliefert, der statische Inhalte wie HTML, CSS und JavaScript effizient bereitstellt.

3. Backend – Python Django Framework

Das Backend basiert auf dem Python-Django-Framework und wird durch einen *Uvicorn*-Server bereitgestellt. Django stellt die REST-API bereit, über die das Frontend mit dem Backend kommuniziert. Die API wird durch einen dedizierten API-Gateway abgesichert, das die Anfragen verwaltet und gegebenenfalls weiterleitet.

4. Datenbank – PostgreSQL

Die relationalen Daten des Webshops werden in einer PostgreSQL-Datenbank gespeichert. Diese bietet eine zuverlässige und skalierbare Lösung für die Verwaltung der Geschäftsdaten wie Produkte, Bestellungen und Kundendaten.

5. Statische Ressourcen und Testdaten

Statische Inhalte wie Medienoder andere statische Ressourcen werden zentral gespeichert und von Django bereitgestellt.

6. E-Mail-Service – Go-Mailservice

Für die Abwicklung von Benachrichtigungen und E-Mails wird ein separater Go-Mailservice verwendet. Dieser ist über eine REST-Schnittstelle in die Systemarchitektur integriert und stellt sicher, dass alle E-Mails zuverlässig und performant versendet werden.

7.2 Containerisierung und Deployment

Alle genannten Dienste werden in Docker-Containern ausgeführt, was eine portable und konsistente Umgebung gewährleistet. Durch den Einsatz von *Dokploy* wird das Deployment automatisiert und in verschiedene Umgebungen wie Test- und Produktionssysteme orchestriert. Diese Containerisierung sorgt für eine klare Trennung der Dienste und ermöglicht es, Updates oder Skalierungen gezielt durchzuführen, ohne das Gesamtsystem zu beeinträchtigen.

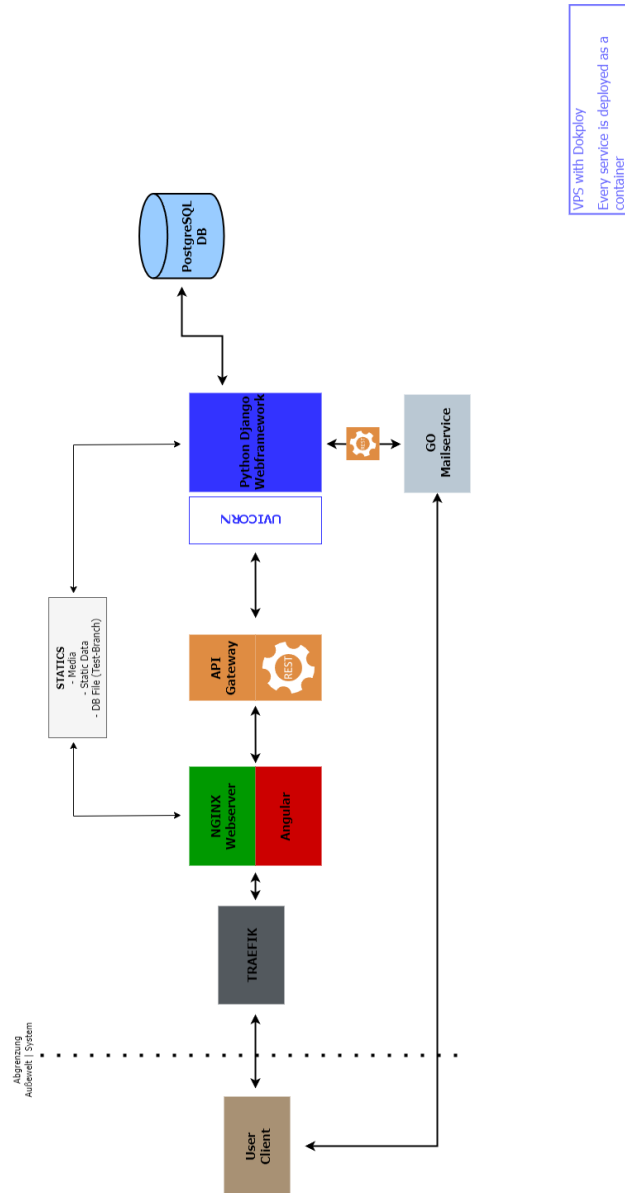


Abbildung 7.1: Systemarchitektur

8 Datenmodell

- Beschreibung Datenstruktur und deren Beziehung (ER-Diagramm erstellen)
- Datenbanktabellen beschreiben, Felder der Tables

9 Schnittstellendefinition (API)

Methode	API-Endpoint	Übergebene Daten (Request)	Antwort (Response)
POST	/web/api/auth/register/	9.1	9.2
POST	/web/api/auth/login/	9.3	9.4
POST	/web/api/auth/refresh/	9.5	9.6
GET	/web/api/me/profile/	-	9.7
POST	/web/api/me/orders/	9.8	9.9

Tabelle 9.1: API-Schnittstellendokumentation des HotHardwareHub-Shops

```
1  {
2      "email": "test@example.com",
3      "password": "securepassword",
4      "password_confirm": "securepassword"
5  }
```

Listing 9.1: Request für Registrierung

```
1  {
2      "email": "test@example.com"
3  }
```

Listing 9.2: Response für Registrierung

```
1  {
2      "email": "test@example.com",
3      "password": "securepassword"
4  }
```

Listing 9.3: Request für Login

```
1  {
2      "access": "JWT_ACCESS_TOKEN",
3      "refresh": "JWT_REFRESH_TOKEN"
4  }
```

Listing 9.4: Response für Login


```
1      {  
2          "refresh": "JWT_REFRESH_TOKEN"  
3      }
```

Listing 9.5: Request für Token Refresh

```
1      {  
2          "access": "NEW_JWT_ACCESS_TOKEN"  
3      }
```

Listing 9.6: Response für Token Refresh

```
1      {  
2          "email": "test@example.com",  
3          "company_identifier": "12345",  
4          "company_name": "Tech Corp"  
5      }
```

Listing 9.7: Response für Profilabruf

```
1      {  
2          "order_info": {  
3              "buyer_name": "Max Mustermann"  
4          },  
5          "items": [  
6              { "item_id": 1, "quantity": 2 }  
7          ]  
8      }
```

Listing 9.8: Request für Bestellungen

```
1      {  
2          "order_id": 11,  
3          "order_status": "pending",  
4          "order_total": 49.99  
5      }
```

Listing 9.9: Response für Bestellungen

Die vollständige Liste aller verfügbaren Endpunkte, inklusive detaillierter Dokumentation und interaktiver Testmöglichkeiten, kann über die **Swagger-UI** abgerufen werden. Diese stellt eine umfassende Übersicht bereit und ermöglicht es Entwicklern, die API effizient zu nutzen und Endpunkte direkt zu testen.

<http://hothardwarehubtest-shop-xco4pb-80b302-5-75-130-54.traefik.me/web/api/swagger/>

10 Benutzungsoberflächen

Dieses Kapitel beschreibt die Gestaltung der Benutzungsoberflächen des Webshops, einschließlich der zugrunde liegenden Richtlinien und Designentscheidungen. Ziel war es, eine benutzerfreundliche, ästhetische und funktionale Oberfläche zu schaffen, die den Anforderungen von Geschäftskunden gerecht wird.

10.1 Gestaltungsrichtlinien

Die Entwicklung der Benutzeroberflächen erfolgte auf Grundlage der folgenden Prinzipien:

- **Klarheit und Übersichtlichkeit:** Die Benutzeroberfläche wurde bewusst einfach und übersichtlich gestaltet, um dem Benutzer eine schnelle Orientierung zu ermöglichen.
- **Konsistenz:** Einheitliche Farben, Schriften und Layout-Elemente sorgen für ein durchgängiges Erscheinungsbild und intuitive Bedienbarkeit.
- **Fokus auf Funktionalität:** Alle Designelemente unterstützen die Nutzbarkeit und wurden auf unnötige Komplexität verzichtet.
- **Responsivität:** Die Benutzeroberfläche ist für verschiedene Endgeräte optimiert, sodass der Webshop sowohl auf Desktops als auch auf mobilen Geräten problemlos genutzt werden kann.

10.2 Verwendete Farben, Schriften und Layout

10.2.1 Farben

Das Farbschema des Webshops basiert auf den Farben des erstellten Logos, um ein einheitliches und professionelles Erscheinungsbild zu gewährleisten. Die Logo-Farben wurden gezielt auf die Benutzeroberfläche übertragen, um eine visuelle Wiedererkennung und Markenidentität zu schaffen.

- **Primärfarben:** Die Hauptfarben des Logos werden verwendet, um wichtige Elemente wie Buttons, Links und Header hervorzuheben.
- **Sekundärfarben:** Ergänzende Farben aus der Farbpalette des Logos dienen zur Unterstützung der Struktur, beispielsweise für Hintergrundbereiche oder Navigationsleisten.
- **Neutralfarben:** Dezentere Farben wie Weiß, Grau und Schwarz werden genutzt, um Inhalte klar und übersichtlich zu präsentieren, ohne die Hauptfarben zu überlagern.

Die Übertragung der Logo-Farben in die Benutzeroberfläche sorgt für eine harmonische Verbindung zwischen der Corporate Identity und dem Design des Webshops. Dies unterstützt nicht nur die Ästhetik, sondern stärkt auch die Wiedererkennbarkeit der Marke.

10.2.2 Schriften

Als Schriftart wurde *Roboto* gewählt, da sie modern und gut lesbar ist. Die Schriftgröße und -farbe wurden so angepasst, dass Inhalte auch bei längerer Nutzung gut wahrgenommen werden können.

10.2.3 Layout

Das Layout des Webshops ist klar strukturiert und folgt etablierten Standards:

- **Navigation:** Eine horizontale Hauptnavigation bietet schnellen Zugriff auf Kategorien und wichtige Funktionen.
- **Produktdarstellung:** Produkte werden in einer Rasteransicht präsentiert, um eine übersichtliche Darstellung zu gewährleisten.

- **Interaktive Elemente:** Buttons und Links sind optisch hervorgehoben, sodass sie leicht identifizierbar sind.

10.3 Verzicht auf Mockups

Für die Gestaltung der Benutzeroberfläche wurden keine Mockups verwendet. Stattdessen erfolgte die Entwicklung direkt iterativ während der Programmierung. Entscheidungen zur Gestaltung wurden im Team getroffen und kontinuierlich basierend auf Feedback optimiert. Diese Vorgehensweise ermöglichte eine flexible Anpassung an die Projektanforderungen.

10.4 Zusammenfassung

Die Benutzeroberflächen des Webshops wurden mit dem Ziel entwickelt, ein benutzerfreundliches und funktionales Design zu schaffen. Durch die Orientierung an klaren Richtlinien und Standards konnte eine Oberfläche geschaffen werden, die Geschäftskunden ein angenehmes und effizientes Nutzungserlebnis bietet.

11 Nicht-funktionale Anforderungen

NF10 Das Produkt soll plattformunabhängig sein.

NF20 Das Produkt muss anwenderfreundlich sein (intuitive Bedienbarkeit für Benutzer ohne EDV-Vorkenntnisse, umfangreiche Hilfefunktion)

NF30 Das Produkt muss mit geringem Aufwand weiterentwickelbar und wartbar sein.

NF40 Das Produkt soll fehlertolerant bezüglich Bedien- und Eingabefehler sein.

12 Technische Produktumgebung

In diesem Kapitel wird die technische Umgebung des Webshops beschrieben. Da es sich um eine Client-Server-Anwendung handelt, werden die Umgebungen für den Client und den Server getrennt betrachtet.

12.1 Software

Die Bereitstellung des Webshops erfolgt über einen Virtual Private Server (VPS). Die eingesetzten Softwaresysteme umfassen:

- **Server-Betriebssystem:** Linux (Ubuntu 24.04.1 LTS ARM64 Noble Numbat)
- **Dokploy:** Deployplattform für die Orchestrierung und Verwaltung von Services
- **Webserver:** NGINX für die Auslieferung des Frontends
- **Backend:** Python Django Framework mit Uvicorn
- **Datenbank:** PostgreSQL für die Speicherung von Produkt-, Bestands- und Kundendaten sowie SQLite für die Testumgebung
- **Containerisierung:** Docker für die Bereitstellung und Isolation der verschiedenen Komponenten
- **Reverse Proxy:** Traefik zur Verteilung und Absicherung der Anfragen
- **Frontend:** Angular für die Entwicklung der Benutzeroberfläche
- **Mailservice:** Mailservice in Go, mit REST-Schnittstelle welche vom Backend getriggert wird.

Für den Client wird lediglich ein browserfähiges Endgerät benötigt. Der Webshop ist mit allen modernen Webbrowsern (z. B. Chrome, Firefox, Edge) kompatibel.

12.2 Hardware

Die minimalen und maximalen Anforderungen an die Hardware sind wie folgt:

- **Server:** VPS mit mindestens 2 CPU-Kernen, 4 GB RAM und 50 GB Speicherplatz
- **Client:** Ein browserfähiges Endgerät mit Internetzugang

12.3 Orgware

Der Webshop wird unter den folgenden organisatorischen Randbedingungen eingesetzt:

- Der Server benötigt eine stabile Netzwerkverbindung, um alle Anfragen der Clients zu verarbeiten und mit der Datenbank zu kommunizieren.
- Der Zugriff auf den Webshop erfolgt über gängige Webbrowser, wodurch keine zusätzliche Softwareinstallation auf Client-Geräten notwendig ist.
- Administratoren und Entwickler benötigen Zugriff auf den VPS zur Wartung, Überwachung und Weiterentwicklung des Systems

13 Konzeption der KI-Komponente

13.1 Idee

Für den Webshop soll ein Chatbot erstellt werden. Dieser soll Fragen der Kunden beantworten, um die Kundenzufriedenheit zu erhöhen.

13.2 Zielgruppenanalyse

Der KI-Chatbot richtet sich vor allem an IT-Einkäufer, die nach Produkten mit bestimmten technischen Spezifikationen suchen und einzelne Produkte schnell vergleichen wollen.

13.3 Funktionen und Anwendungsfälle

Produktberatung:

- Empfehlung geeigneter Produkte auf Basis von Kundenanforderungen (z. B. „Ich suche einen Monitor mit mindestens 24 Zoll“)
- Produktvergleiche (z. B. „Was sind die Unterschiede zwischen Laptop A und Laptop B?“)

13.4 Umsetzungsmöglichkeiten

Es gibt mehrere Möglichkeiten, einen Chatbot zu erstellen:

1. Erstellung eines eigenen Large Language Models, welches mit Produktdaten trainiert wird.
2. Nutzung eines Chatbot-Anbieters, der mit Trainingsdaten aus dem B2B-Webshop arbeitet.

Diese Möglichkeiten werden im Folgenden verglichen.

13.4.1 Eigenes Large Language Model (LLM)

Vorteile	Nachteile
Maximale Kontrolle <ul style="list-style-type: none"> · Modell kann individuell trainiert werden · Vollständige Anpassung der Antworten 	Hoher Entwicklungsaufwand <ul style="list-style-type: none"> · Aufbau eines eigenen LLMs erfordert erhebliches Know-how · Längere Implementierungszeit
Datenschutz und Sicherheit <ul style="list-style-type: none"> · Alle Daten bleiben im eigenen System, was im Hinblick auf Datenschutzregelungen vorteilhaft ist - 	Wartung und Aktualisierung <ul style="list-style-type: none"> · Kontinuierliche Pflege, Optimierung und Nachtrainieren des Modells sind erforderlich, um relevante Ergebnisse zu liefern · Aufwendige Skalierung bei steigender Nutzung

Tabelle 13.1: Abwägung der Nutzung eines LLMs

Ein eigenes LLM ist dann zu empfehlen, wenn der Schutz der Privatsphäre und die vollständige Anpassbarkeit im Vordergrund stehen.

13.4.2 Chatbot-Anbieter

Ein Chatbot-Anbieter wird empfohlen, wenn eine schnelle Implementierung und der Zugang zu den neuesten KI-Technologien wichtiger sind.

Im Fall des B2B-Webshops wird der Chatbot über einen Chatbot-Anbieter bereitgestellt, da vor allem die schnelle Implementierung für das zeitlich begrenzte Projekt von Vorteil ist. Die Trainingsdaten stehen erst wenige Wochen vor Projektende zur Verfügung, so dass nur wenig Zeit für die Implementierung bleibt.

Vorteile	Nachteile
Schnelle Implementierung <ul style="list-style-type: none"> · Anbieter bieten oft vorgefertigte Tools, APIs und intuitive Dashboards, die eine schnelle und einfache Integration ermöglichen · Kein Aufbau eines eigenen Modells notwendig 	Abhängigkeit vom Anbieter <ul style="list-style-type: none"> · Daten und Geschäftsprozesse werden auf die Plattform des Anbieters ausgelagert, was zu Abhängigkeiten führen kann ×
Wartung und Updates <ul style="list-style-type: none"> · Anbieter übernimmt Wartung, Optimierung und die Bereitstellung aktueller KI-Modelle · Regelmäßige Updates sorgen dafür, dass das Modell auf dem neuesten Stand der Technik bleibt 	Datenschutzrisiken <ul style="list-style-type: none"> · Daten werden an Drittanbieter übermittelt, die teilweise Server außerhalb der EU verwenden ×
Skalierbarkeit <ul style="list-style-type: none"> · Anbieter verfügen über skalierbare Infrastrukturen, die automatisch auf steigende Nutzungsanforderungen reagieren können ×	Eingeschränkte Anpassbarkeit <ul style="list-style-type: none"> · Anbieter bieten oft weniger Möglichkeiten, das Modell tiefgehend auf unternehmensspezifische Anforderungen anzupassen · Limitierte Kontrolle über den Trainingsprozess des Modells

Tabelle 13.2: Abwägung der Nutzung eines externen Chatbots

13.5 Anforderungen an den Chatbot

13.6 Datenanforderungen

Die für den Chatbot erforderlichen Produktdaten sind unter anderem die Artikelnummer, die technischen Spezifikationen und die Preise.

13.7 Auswahl des Anbieters

Nach Recherche und Vergleich mehrerer Anbieter wurde „Botpress“ als Dienstleister ausgewählt. Dieser bietet in seinem kostenlosen Modell die besten Möglichkeiten und 500 kostenlose Anfragen pro Monat. Darüber hinaus besteht die Möglichkeit, den Chatbot an die spezifischen Anforderungen anzupassen und eine reibungslose Integration in das

Funktionale Anforderungen	Nicht-funktionale Anforderungen
<ul style="list-style-type: none"> · Erkennung von Kundenanfragen in natürlicher Sprache · Unterstützung mehrstufiger Dialoge - 	<ul style="list-style-type: none"> · Vollständige Antworten in weniger als 10 Sekunden · 24/7-Zugriff · Intuitive Interaktion, auch für nicht-technische Nutzer

Tabelle 13.3: Anforderungen an den Chatbot

Frontend zu gewährleisten.

13.8 Bot-Instructions

Gemäß den Vorgaben der Bot-Instructions ist der Chatbot dazu angehalten, spezifische Antworten zu geben und festgelegte Zuständigkeiten zu erfüllen. Es wurde eine Erwartung an die Klarheit und Konsistenz der Antworten des Chatbots gestellt, wobei gleichzeitig die Interaktionsmöglichkeit des Nutzers durch Buttons festgesetzt wurde. Darüber hinaus wurde spezifiziert, dass der Bot Produktempfehlungen ausgeben soll, die den Angaben des Kunden entsprechen.

13.9 Datenintegration in Botpress

Die Daten können Botpress auf verschiedene Weise zur Verfügung gestellt werden. Die beiden einfachsten Möglichkeiten sind die Eingabe eines Links zur Website oder die Bereitstellung über eine CSV-Datei.

Bei der ersten Möglichkeit wird die Website bei einer Anfrage analysiert und das Ergebnis an den Kunden zurückgegeben. Nach mehreren Tests mit einer Website hat sich herausgestellt, dass diese Methode ungenaue Antworten liefert und nicht immer die richtigen Produkte ausgibt bzw. bei unpräzisen Anfragen keine passenden Antworten zurückgegeben werden.

Die zweite Möglichkeit besteht darin, eine CSV-Datei mit den Produktdaten in Botpress hochzuladen. Nach mehreren Tests mit Testdaten hat sich gezeigt, dass die Antworten präziser sind und auch bei ungenauen Anfragen sinnvolle Antworten zurückgegeben werden. Ein Nachteil ist, dass die Daten immer manuell aktualisiert werden müssen. Dennoch ist diese Methode für den Einsatz im B2B-Webshop besser geeignet, da auch ungenaue Anfragen sinnvoll beantwortet werden müssen.

Für die Realisierung im Zuge des Projektes wurde aufgrund der vorliegenden Vorteile die zweite Variante gewählt. Die Produktdaten welche im Webshop eingebunden werden, wurde in die Wissensdatenbank des Chat-Bots eingefügt.

13.10 Integration des Chatbots

Nach Integration der Produktdaten bei dem Anbieter und der darauffolgenden Publikation werden seitens Botpress zwei Skript-Dateien bereitgestellt. Diese Skript-Zeilen können anschließend in das Frontend eingebettet werden. Dadurch wird der Chatbot für die Webshop-Besucher sichtbar und kann verwendet werden.

13.11 Bearbeitungsablauf einer Kundenanfrage

Im Falle der Eingabe einer Frage in den Chat erfolgt eine Analyse durch das LLM. Für die Beantwortung einfacher Anfragen wird GPT-4o Mini verwendet, für eine bestmögliche Beantwortung GPT-4o und für die Generierung der Antwort. Nach Analyse wird eine Durchsicht der Knowledge Base durchgeführt, in diesem Fall der Produktdatenbank, da keine weitere Knowledge Base zur Verfügung steht. Nach dem Auffinden der passenden Informationen erfolgt die Formulierung einer Antwort und der bereitstellung für den Anfragenden.

14 Projektorganisation

14.1 Projektmethodik

Die Entwicklung des Webshops für den *Hot Hardware Hub* erfolgt nach der agilen Methode Scrum, da sie besonders gut für komplexe und dynamische Softwareprojekte geeignet ist. Die Wahl von Scrum basiert auf mehreren entscheidenden Faktoren, die sich sowohl auf die Anforderungen des Projekts als auch auf die Besonderheiten der IT-Hardware-Branche beziehen.

1. Flexibilität und Anpassungsfähigkeit an Marktveränderungen

Die IT-Hardware-Branche ist einem schnellen Wandel unterworfen. Technologien entwickeln sich stetig weiter, neue Produkte kommen auf den Markt, und Kundenanforderungen ändern sich kontinuierlich. Eine klassische, langfristig fixierte Projektplanung (z. B. nach dem Wasserfall-Modell) wäre in einem solch dynamischen Umfeld nicht optimal, da sie wenig Spielraum für kurzfristige Anpassungen lässt.

Scrum hingegen ermöglicht eine iterative Entwicklung in kurzen Sprints, sodass regelmäßig neue Funktionen ausgeliefert und Feedback berücksichtigt werden können. Sollte sich beispielsweise herausstellen, dass Geschäftskunden verstärkt eine bestimmte Filterfunktion oder eine alternative Bezahlmethode benötigen, kann dies flexibel in den nächsten Sprint aufgenommen und zeitnah umgesetzt werden.

2. Frühzeitige und kontinuierliche Lieferung von Produktinkrementen

Ein zentraler Vorteil von Scrum ist die Möglichkeit, den Webshop schrittweise zu entwickeln und dabei nach jedem Sprint ein funktionsfähiges Produktinkrement bereitzustellen. Dadurch wird verhindert, dass das gesamte Projekt erst nach Monaten oder Jahren einsatzbereit ist. Stattdessen können erste Versionen – beispielsweise als *Minimal Viable Product* (MVP) – schon früh veröffentlicht und dann fortlaufend verbessert werden.

Diese Vorgehensweise ermöglicht es, den Markt zu testen, erste Kunden zu gewinnen und deren Feedback in die Weiterentwicklung einfließen zu lassen. So kann sichergestellt werden, dass der Webshop von Beginn an kundenorientiert ist und

sich an deren tatsächlichen Bedürfnissen ausrichtet.

3. Kundenorientierung und Einbindung von Stakeholdern

Da der *Hot Hardware Hub* speziell für Geschäftskunden konzipiert ist, ist es essenziell, dass deren Anforderungen und Wünsche kontinuierlich in die Entwicklung einfließen. Im Scrum-Framework übernimmt der **Product Owner** diese Rolle, indem er die Interessen der Kunden vertritt, Anforderungen priorisiert und das *Product Backlog* verwaltet.

Regelmäßige Meetings, wie die *Sprint Reviews*, bieten zudem die Möglichkeit, dass Stakeholder (z. B. zukünftige Nutzer, Geschäftspartner oder interne Entscheider) den Entwicklungsfortschritt begutachten und frühzeitig Rückmeldungen geben. Dadurch wird sichergestellt, dass der Webshop von Anfang an marktgerecht entwickelt wird und spätere, kostenintensive Nachbesserungen vermieden werden.

4. Verbesserung der Zusammenarbeit im Entwicklungsteam

Scrum fördert eine transparente und effiziente Zusammenarbeit innerhalb des Teams. Durch *Daily Scrums* wird sichergestellt, dass jedes Teammitglied stets über den aktuellen Stand der Entwicklung informiert ist. Probleme oder Hindernisse werden frühzeitig erkannt und können schnell gelöst werden.

Zudem sorgen *Sprint Retrospektiven* dafür, dass das Team kontinuierlich aus vergangenen Sprints lernt und seine Arbeitsweise verbessert. Dies steigert nicht nur die Effizienz, sondern auch die Qualität des Endprodukts.

5. Risikominimierung durch regelmäßige Tests und Qualitätskontrollen

Ein weiteres zentrales Argument für die Nutzung von Scrum ist die Minimierung von Risiken durch regelmäßige Tests und Überprüfungen. Da in jedem Sprint ein testbares Produktinkrement entsteht, können Fehler frühzeitig erkannt und behoben werden.

Im Gegensatz zu traditionellen Entwicklungsmodellen, bei denen ein Produkt erst am Ende der Entwicklungsphase umfassend getestet wird, stellt Scrum sicher, dass kontinuierlich getestet wird. Dies reduziert das Risiko schwerwiegender Fehler oder technischer Probleme, die ansonsten erst spät erkannt würden.

6. Schnellere Markteinführung und höhere Wettbewerbsfähigkeit

Durch die iterative Entwicklung und die regelmäßige Auslieferung neuer Funktionen kann der Webshop schneller auf den Markt gebracht werden. Ein früher Release als *MVP* ermöglicht es, bereits erste Umsätze zu generieren und auf Basis von Kundenfeedback weiterzuentwickeln.

Gerade im B2B-Bereich, wo Unternehmen nach effizienten und unkomplizierten Beschaffungslösungen suchen, ist es wichtig, möglichst früh eine funktionierende

Plattform bereitzustellen. Durch Scrum kann der Webshop frühzeitig in Betrieb genommen und schrittweise verbessert werden, was einen klaren Wettbewerbsvorteil darstellt.

14.2 Rollenverteilung

Um eine klare Aufgabenverteilung sicherzustellen und die Effizienz des Teams zu maximieren, wurde das Projektteam in verschiedene Verantwortungsbereiche eingeteilt. Nachfolgend wird die Rollenverteilung detailliert beschrieben:

14.2.1 Organisatorische Leitung

Die organisatorische Leitung des Projekts übernehmen **Colin Christ** und **Steffen Spatzek**. Ihre Aufgaben umfassen:

- Koordination der übergeordneten Prozesse
- Führung des Teams und Sicherstellung einer effizienten Zusammenarbeit
- Verantwortung für die Kommunikation zwischen den einzelnen Bereichen
- Sicherstellung der Zielerreichung und Einhaltung von Deadlines
- Eingreifen bei Problemen während der Sprints
- Vorbereitung und Durchführung von Terminen (Zwischenstand vor dem Plenum und Dozenten)

14.2.2 Organisation

Die Verantwortung für die organisatorischen Inhalte des Projekts liegt bei:

- **Lukas Tomanek, Colin Christ, Steffen Spatzek, David Schreck, Bastian Bamberger, Milan Jeevakanthan, Luis Väth und Daniel Arnold**

Ihre Aufgaben umfassen:

- Erstellung und Pflege der Projektdokumentation
- Regelmäßige Aktualisierung des Pflichtenhefts
- Erstellung von Modellen und Diagrammen
- Planung und Bearbeitung der Sprints
- Präzise Formulierung der Use-Cases als Grundlage für die Projektarbeit
- Finale Version des Pflichtenhefts als LaTeX-Dokument

14.2.3 Entwicklung

Das Entwicklungsteam setzt sich aus den folgenden Mitgliedern zusammen:

- **Steffen Spatzek:** Führende Rolle in der Entwicklung, verantwortlich für die technische Umsetzung, insbesondere:
 - Weiterentwicklung des Django Frameworks
 - Containerisierung der Services
 - Verantwortung für den Deployprozess und die VPS-Überwachung
 - Erstellung und Bereitstellung des E-Mail-Services
 - Mitwirkung bei der Fehlerbehebung und Verbesserung des Frontends
- **Alan Kanjo:** Unterstützte maßgeblich bei der Bereitstellung des Django Frameworks und der Containerisierung.
- **Andreas Denz:** Federführend verantwortlich für die Entwicklung des Frontends in Angular und Unterstützung im Backend.
- **Paul Kunz, Julian Strohm** sowie **David Schreck:** Unterstützten bei verschiedenen Aspekten der Softwareentwicklung, darunter:
 - Unterstützung bei Backend- und Frontend-Funktionalitäten
 - Mithilfe bei der Entwicklung neuer Features für den Webshop

- Bereitstellung von HTML-Templates für den E-Mail-Versand.
- Produktdatenaufbereitung und -bereitstellung

14.2.4 Solution Architecture

Die technische Architektur des Systems wird von **Steffen Spatzek** sowie **Alan Kanjo** verantwortet. Die Aufgaben umfassen:

- Sicherstellung einer stabilen, skalierbaren und sicheren IT-Infrastruktur
- Reproduzierbare Versionen mithilfe von Docker-Containern

14.2.5 User Experience (UX)

Für die Gestaltung und Optimierung der Benutzererfahrung sind die folgenden Teammitglieder zuständig:

- **Alan Kanjo, Lukas Tomanek, David Schreck, Julian Strohm, Bastian Bamberger und Milan Jeevakanthan**

Ihre Aufgaben umfassen:

- Gestaltung und Verbesserung der Benutzeroberfläche
- Durchführung von User-Testing
- Implementierung von UX-Optimierungen zur Gewährleistung eines reibungslosen und angenehmen Einkaufserlebnisses für Geschäftskunden
- Erarbeitung eines Konzeptes für die Integration eines KI-Chatbots sowie Vorbereitungsmaßnahme für die Implementierung

14.2.6 Zusammenfassung

Diese Rollenverteilung gewährleistet, dass alle relevanten Aspekte des Webshop-Projekts professionell abgedeckt werden. Das Team kann effizient auf Anforderungen reagieren und Herausforderungen während der Entwicklung meistern.