

SERVICE-BESCHREIBUNG

Konzeptentwicklung

ERFINDEN

ERFORSCHEN

ENTWICKELN

ERHALTEN

Das **Richtige** richtig entwickeln

Über 25 Jahre haben wir unseren Entwicklungsprozess verfeinert und jede Projektphase iterativ optimiert. Unser Ziel: Wir wollen Sie dabei unterstützen das perfekt zu Ihrem Business Case, zu Ihren Anwendern und zu Ihrer Systemlandschaft passende Softwareprodukt zu entwickeln. Und das auf effiziente, nachhaltige und transparente Art und Weise.



Wie wir im Detail vorgehen, erfahren Sie in diesem Dokument



Inhalt

1	Warum eine vorgelagerte Konzeptentwicklung?	3
1.1	Das Ziel	3
1.1.1	Ein validiertes Lösungskonzept	3
1.1.2	Plan für die Umsetzung	3
1.2	Ihre Vorteile durch diese Vorgehensweise	4
1.2.1	Sie bekommen ein erstes „Bild“ des späteren Produkts	4
1.2.2	Ihre Anforderungen sind in entwickelbare Arbeitspakete übersetzt	4
1.2.3	Die Entwicklung Ihres Produkts ist durchgeplant	5
1.2.4	Ihr Produkt ist „dev-ready“	5
1.2.5	Sie erhalten eine Kosteneinschätzung für die Softwareentwicklung	5
1.2.6	Projektrisiken sind aufgedeckt	5
1.2.7	Produktrisiken werden minimiert	5
2	Mögliche Deliverables der Konzeptentwicklung	6
2.1	Project Setup	7
2.1.1	Product Vision	7
2.1.2	Project Canvas	7
2.1.3	Risiko-Management mittels RAID-Log	8
2.1.4	RACI-Matrix	9
2.1.5	Ressourcenbedarfsplan	9
2.2	Verstehen der Ausgangssituation	10
2.2.1	Proto-Personas	10
2.2.2	Prozesslandkarte	11
2.2.3	Affinity Map (Ergebnisse der Benutzerforschung & Stakeholder Interviews)	12
2.3	Analyse und Ausarbeitung der Anforderungen	13
2.3.1	Ideenkatalog	13
2.3.2	User Stories & User Story Map	13
2.3.3	Validierungsplan	14
2.3.4	User Journey	14
2.3.5	Use Cases (Definition der Anwendungsfälle)	15
2.3.6	Interaktions-Design und Konzeptvalidierung mittels Wireframes und UX-Prototypen	16
2.3.7	Entity-Relation-Modell (ER-Modell)	18
2.3.8	System-Umwelt-Diagramm	19
2.3.9	Technisches Systemkonzept	20
2.5	Vorbereitung der Entwicklung	21
2.5.1	Validiertes Backlog	21
2.5.2	Product Roadmap	22
3	Fazit	23

1 Warum eine vorgelagerte Konzeptentwicklung?

Oft gibt es zu Beginn eines Softwareprojekts lediglich eine grobe Idee der späteren Lösung. Das „Warum“ aus Business-Sicht ist meist klar, jedoch sind das „Was“ und das „Wie“ noch nicht ausgearbeitet. Trotzdem besteht in solch einer sehr frühen Phase des Entwicklungsprozesses bereits der Wunsch und der Bedarf, Antworten auf diese Fragen zu erhalten:

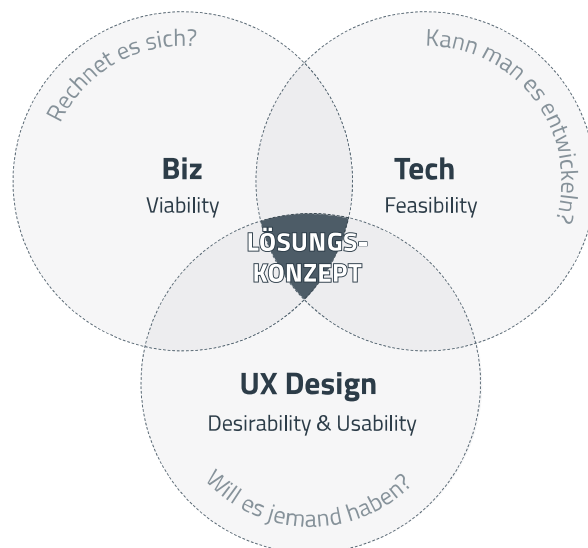
- Ist die Lösung realisierbar, und in welcher Form?
- Wie könnte das spätere Softwareprodukt aussehen?
- Wie lange wird die Entwicklung ungefähr dauern?
- Mit welchem Investitionsvolumen ist zu rechnen?

Unser Konzeptentwicklungs-Service gibt Antworten auf diese Fragen und unterstützt Sie dabei, **das richtige Softwareprodukt für Ihren Business Case zu konzipieren.**

1.1 Das Ziel

1.1.1 Ein validiertes Lösungskonzept

Bei der Erarbeitung des Lösungskonzepts legen wir die Bereiche Business, Technologie und UX Design übereinander. Dort, wo sich alle drei Bereiche treffen und überschneiden, findet sich das ideale Konzept für Ihr Produkt. Die Ideen und Lösungsansätze, die im Laufe der Konzeptentwicklung entstehen, werden dabei kontinuierlich über Prototypen und Tests geprüft und validiert.



1.1.2 Plan für die Umsetzung

Ziel der Konzeptentwicklung ist, dass unmittelbar danach mit der Softwareentwicklung begonnen werden kann. Dazu erarbeiten wir einen Umsetzungsplan, der im Detail zeigt, was die erste Zeit nach der Konzeptentwicklung im Projekt passieren wird:

Aufwandsdimensionierung	Planung der Meilensteine & Releases	Erstellung des Product Backlogs für das erste Release
Wie lange wird die Umsetzung etwa dauern, und wie stellen wir das Team zusammen?	Wir starten mit dem Wichtigsten, so dass mit möglichst wenig Entwicklung möglichst viel Business Value entsteht.	Wir beschreiben die Aufgaben für das Team im Detail, und bringen sie in die richtige Reihenfolge.

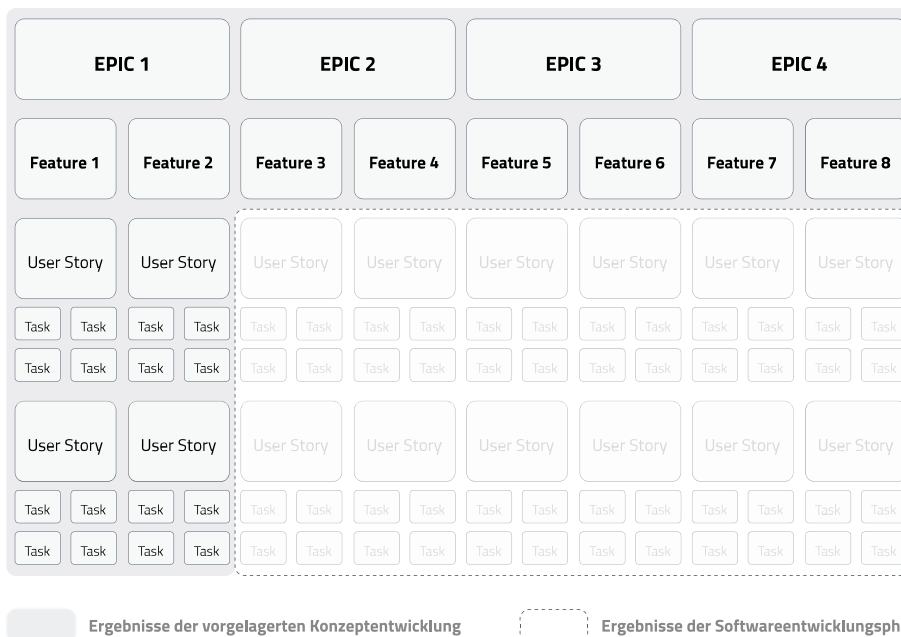
1.2 Ihre Vorteile durch diese Vorgehensweise

1.2.1 Sie bekommen ein erstes „Bild“ des späteren Produkts

All die gewonnen Erkenntnisse der Konzeptentwicklung fließen in erste Click Dummies ein, die auf Basis Ihrer validierten Anforderungen entwickelt werden. Damit wird das Lösungskonzept nicht nur erfahrbar und überzeugend vermittelbar, Sie erhalten dadurch in dieser noch frühen Phase bereits ein erstes Bild Ihres späteren Produkts – und ein repräsentatives Mittel, um die Projektbeteiligten sowie die Stakeholder abzuholen und zu überzeugen.

1.2.2 Ihre Anforderungen sind in entwickelbare Arbeitspakete übersetzt

Sie erhalten nach der Konzeptentwicklung ein Product Backlog, in dem Ihre Produktvision sowie die elementaren Anforderungen in Epics und Features übersetzt, definiert und priorisiert wurden. Darüber hinaus legen wir die User Stories und Tasks für die ersten Sprints fest. Somit kann das Entwicklerteam im Anschluss direkt mit der Umsetzung starten.



Epics: strategische Ziele des Produkts

Bsp.: Vertrauensaufbau in einer Online-Einkaufsplattform

Features: eigenständige, in sich abgeschlossene Funktionalität im Produkt

Bsp.: Einkaufsplattform bekommt eine Produktbewertungsfunktion

User Story: In Alltagssprache formulierte Produktfunktionalität aus Sicht eines Anwenders

Bsp.: Als registrierter Benutzer möchte ich Produkte bewerten können, damit ich meine Erfahrungen teilen und anderen Käufern bei ihrer Entscheidung helfen kann.

Task: die einzelnen Entwicklungsaufgaben, die zur Erfüllung der User Story notwendig sind

Bsp.: Entwerfe die Eingabemaske für die Produktbewertung

Die Erarbeitung und Umsetzung der restlichen User Stories und Tasks erfolgt in der Entwicklungsphase und ist kein Bestandteil der vorgelagerten Konzeptentwicklung.

1.2.3 Die Entwicklung Ihres Produkts ist durchgeplant

Sie erhalten einen priorisierten Plan für die Umsetzung (Roadmap) der strategischen Ziele und Funktionen (Epics und Features). So ist es Ihnen möglich, ganzheitlich Transparenz für alle Stakeholder und Projektbeteiligten zu schaffen. Natürlich kann der Umsetzungsplan jederzeit angepasst werden.

1.2.4 Ihr Produkt ist „dev-ready“

Alle notwendigen Informationen, Anforderungen und Ressourcendefinitionen sind vorbereitet, um direkt mit der Entwicklungsphase Ihres Softwareprodukts zu beginnen. Unklarheiten sind beseitigt, Spezifikationen sind festgelegt. Das Entwicklungsteam kann reibungslos mit der Umsetzung beginnen.

1.2.5 Sie erhalten eine Kosteneinschätzung für die Softwareentwicklung

Softwareentwicklungsprojekte bewegen sich in ihrem Umsetzungsaufwand und damit ihrer monetären Investition immer in einem gewissen Bereich. Dieser anfangs breite Unsicherheitskegel wird zum Ende der Konzeptentwicklung deutlich schmaler, wodurch wir Ihnen eine realitätsnahe Kosteneinschätzung bereitstellen können.

1.2.6 Projektrisiken sind aufgedeckt

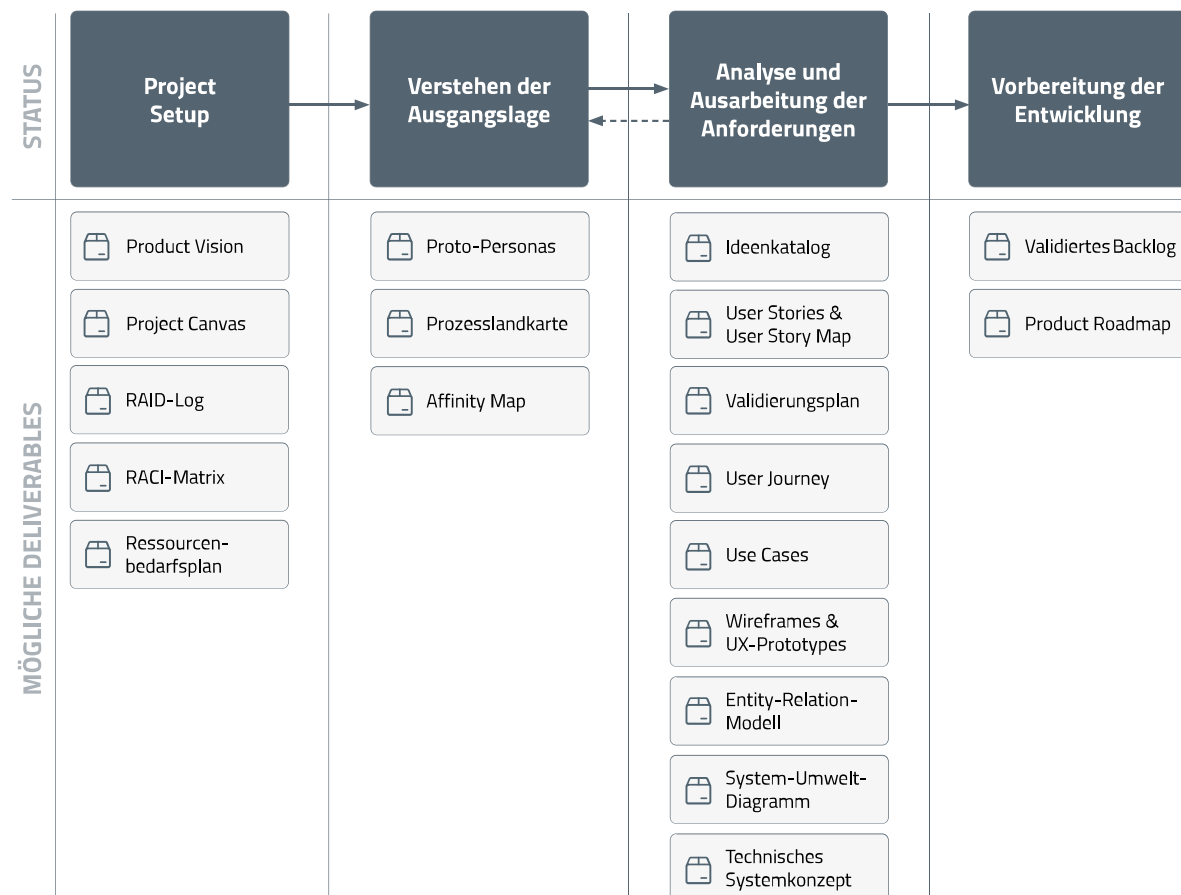
Wir haben alle zum jetzigen Zeitpunkt ersichtlichen Risiken für das Projektvorhaben und die Umsetzung definiert und transparent gemacht. Außerdem stimmen wir gemeinsam mit Ihnen erste Maßnahmen ab, wie mit den Risiken umgegangen wird und wie sie eliminiert werden können.

1.2.7 Produktrisiken werden minimiert

Durch das Validieren des Konzepts bieten wir Ihnen die Sicherheit, dass wir das richtige Produkt für den derzeitigen Markt und die richtige Zielgruppe entwickeln werden und Ihre Investition in eine sinnvolle, erfolgsversprechende Software fließt.

2 Mögliche Deliverables der Konzeptentwicklung

Das vorgestellte Lösungskonzept bündelt diverse Arbeitsergebnisse, die wir als Artefakte oder Deliverables bezeichnen. Im Folgenden werden die zentralen Arbeitsergebnisse skizziert. Es ist jedoch zu betonen, dass jedes Projekt individuelle Anforderungen hat und somit **nicht jedes aufgeführte Artefakt zwangsläufig relevant ist**.



2.1 Project Setup

2.1.1 Product Vision

In der agilen Softwareentwicklung werden unterschiedliche Planungshorizonte betrachtet. Jeder Horizont orientiert sich an dem nächsthöheren und beeinflusst damit die Inhalte des darunterliegenden. Die Product Vision ist der am weitesten blickende Horizont. Eine gut formulierte Product Vision hilft zu überprüfen, ob alle Aufgaben innerhalb der Planungsebenen, dem obersten Ziel – den Business Outcomes – dienen.

Product Vision Board Musterprojekt			
VISION What is your motivation for creating the product? Which positive change should it bring about?			
TARGET GROUP Which market or market segment does the product address? Who are the target customers and users?	NEEDS Which problem does the product solve? What benefit does it provide?	PRODUCT What product is it? What makes it stand out? Is it feasible to develop the product?	BUSINESS GOALS How is the product going to benefit the company? What are the business goals?
COMPETITORS Who are your main competitors? What are their strengths and weaknesses?	REVENUE STREAMS How can you monetise your product and generate revenues?	COST FACTORS What are the main cost factors to develop, market, sell, and service the product?	CHANNELS How will you market and sell your product? Do the channels exist today?

2.1.2 Project Canvas

Ein Project Canvas stellt die wichtigsten Informationen und Elemente eines Projekts kompakt und übersichtlich auf einer einzigen Seite dar. Das Deliverable unterstützt die Effektivität, Transparenz und Zusammenarbeit im Projektteam und trägt dazu bei, erfolgreiche Ergebnisse zu erzielen.

Project Canvas Musterprojekt				
Zweck Welche Absicht sieht hinter dem Projekt? (Zweck / Herausforderung, Ursache) Warum ist das Projekt wichtig und bedeutsam – und für wen? Inwiefern wird das Projekt die Zukunft verändern – und für wen?				
Budget Wieviel Geld wird benötigt? ... für das Team ... für den Kunden? ... für die notwendigen Ressourcen?	Team Wer ist dabei/sollte dabei sein? ... im Kernteam? ... im erweiterten Team? ... als externer Partner? ... als Projektsponsor/-manager?	Umfeld Bekanntes Kräftefeld, bestehendes Engagement und Menschen, die das Projekt beeinflussen. ... Wen/was unterstützt das Projekt? ... Wen/was behindert das Projekt?	Meilensteine Welche Meilensteine sind wichtig? (Anker für Zwischenergebnisse ... wichtige Entscheidungen ... sicht- und messbare Erfolge)	Qualität Was macht die Kunden wirklich glücklich bezogen auf das Ergebnis des Projekts? ... die Meilensteine auf dem Weg? ... die Art der Informationen/Mitarbeiter im Projekt
Ressourcen Welche Ressourcen werden benötigt? ... Projektschulung ... Gesprächsgrundlage ... Werkzeuge (z.B. Software/Materialien)	Risiken + Chancen Welche unvorhergesehenen Ergebnisse werden im Falle eines Scheiterns, den Projektfortschritt gefährden bzw. befruchten? Welche externen Ereignisse und beeinflussbare Ergebnisse sind als sinnvolle Reaktionsmöglichkeiten zu berücksichtigen?	Ergebnis Was genau soll das Projekt für die Kunden liefern? Was ist es an sich? ... ein neues Produkt/ein neuer Service ... neue Erkenntnisse/Wissen	Kunde Wer ist gegenüber der Kunde? Menschlich die das Projekt finanzieren (Sponsoring) ... das Projekt starten & beenden (Eigenarbeit) ... die Projektergebnisse erhalten (Empfänger) Bei mehreren Kunden: Gibt es überlappende Kontakte?	
Zeit Wann startet das Projekt tatsächlich? Was wird dafür benötigt? (z.B. Vorbereitungen, Dokumente, Freigaben). Wann es das Projekt wirklich abgeschlossen? Was wird dafür benötigt? (z.B. Dokumente, Freigaben). Wie flexibel sind die Start- und Endtermine? Wie flexibel sind die Termine der Meilensteine?				

2.1.3 Risiko-Management mittels RAID-Log

Ein RAID-Log ist ein effektives Werkzeug zur Identifizierung und Verwaltung von potenziellen Herausforderungen und Chancen im Projekt. Das Akronym steht für: **R**isks (Risiken), **A**ssumptions (Annahmen), **I**ssues (Probleme) und **D**ependencies (Abhängigkeiten). Die Führung eines RAID-Logs ermöglicht es den Projektverantwortlichen, sich systematisch und proaktiv mit Unsicherheiten und potenziellen Hindernissen auseinanderzusetzen. Dadurch werden mögliche Risiken frühzeitig erkannt, was die Wahrscheinlichkeit von Verzögerungen oder Kostenüberschreitungen reduziert. Das Deliverable ermöglicht Ihnen:

- Transparenz über aktuelle und potenzielle Herausforderungen
- Aktive Teilnahme am Problemmanagement
- Besseres Verständnis der Projektrisiken
- Fundierte Entscheidungsfindung

RAID-Log | Musterprojekt generic||de

Projekt 11.01.2022 15:15

Projekt - Prozessschritte
Aktiv für 12 Monate

In Anbahnung In Umsetzung (12 Mo.) Beendet

Allgemein Aktivitäten **Risiken** Mitglieder Budget Ansprechpartner Leistungspakete Fortschritt Verkaufschancen Risiko Historie Dokumente Technologien Aufträge ...

+ Projektrisiko erstellen Aktualisieren Flow

Nach Schlüsselwort filtern

Name	Eintrittsw...	Auswirkung	Ursache	Geltungsbereich	Beschreibung der Auswirkung	Beschreibung des Risikos	Besitzer	Erstellt am	Status	Statusgru...
Abhängigkeiten zu externen DAQ Systemen	50	mittel	Kunde		Projektverzug / Budgetrisiko	Die zu entwickelnde Anw...		03.02.202...	Aktiv	Aktiv
Reporting an ...	50	mittel	Kunde			Das Reporting von gener...		03.02.202...	Aktiv	Aktiv
Terminierte Deadline des Projektes mit ho...	50	mittel	Kunde		Projektverzug / Budgetrisiko	Klare Vorstellung von		03.02.202...	Aktiv	Aktiv
Unbekannt Software-Qualität und Umsetz...	50	mittel	generic.de		Übergebene Leistungen (Source...	Die Arbeit von		03.02.202...	Aktiv	Aktiv
Ungeklärte Rollen bei ...	80	hoch	Kunde		Projektverzug durch * Personalh...	Bei		03.02.202...	Aktiv	Aktiv

2.1.4 RACI-Matrix

Die RACI-Matrix definiert die Verantwortlichkeiten und Beteiligungen von Personen oder Teams in einem Projekt. Das Deliverable ermöglicht:

- Klare Zuweisung von Verantwortlichkeiten & Verbindlichkeiten
- Förderung der Kommunikation und Zusammenarbeit
- Erhöhung der Effizienz und Transparenz
- Unterstützung bei der Konfliktvermeidung
- Unterstützung bei der Entscheidungsfindung
- Unterstützung bei der Projektsteuerung

Zur besseren Vorstellung, hier eine exemplarische RACI-Matrix:

	Kunde	Projekt Manager	Product Owner	Scrum Master	Dev-Team	UX Designer	Software-architekt
Anlegen Systemzugänge	A	R	C	I	C	I	C
Projektsteuerung	C	A & R	C	C	I	I	I
Agiler Prozess	C	I	C	A & R	C	C	I
Rollout-Unterstützung	A	I	R	I	C	I	I
Requirements Engineering	A	C	R	I	C	C	I
Testing & Releases	A	C	R	I	C	C	C
Qualitätssicherung	C	A & R	C	C	C	C	C
UX-Konzepte	C	I	C	I	C	A & R	C
Quellcode	I	I	I	I	A & R	I	C
...							

R = Responsible (zuständig) | **A** = Accountable (verantwortlich) | **C** = Consultant (konsultiert) | **I** = Informed (informiert)

2.1.5 Ressourcenbedarfsplan

Ein Ressourcenbedarfsplan ist wichtiger Bestandteil der kompletten Projektabwicklung und bezieht sich auf die Identifizierung und Zuweisung von verschiedenen Ressourcen, die für die erfolgreiche Umsetzung benötigt werden. Dies umfasst sowohl menschliche als auch materielle Ressourcen wie Arbeitskräfte, Zeit, Finanzen, Ausrüstung und Materialien. Ein Ressourcenbedarfsplan legt fest, welche Ressourcen benötigt werden, wann sie benötigt werden und wie sie im Verlauf des Projekts eingesetzt werden.

2.2 Verstehen der Ausgangssituation

2.2.1 Proto-Personas

Personas sind Beschreibungen fiktiver, aber realitätsnaher Benutzer. Sprechen wir von Proto-Personas sind Personas gemeint, die auf Basis von Annahmen entwickelt wurden und noch validiert werden müssen. Das Deliverable dient dazu, den typischen Benutzer besser zu verstehen. Dafür wird er mittels Hintergrundinformationen, Zielen, Verantwortungen, Tätigkeiten, Herausforderungen (Pains) und Bedürfnissen (Needs), Kompetenzen sowie Kollaborationen spezifiziert. Das Ziel ist die Erarbeitung konkreter Benutzeranforderungen an das spätere Softwareprodukt.

Auf Projektebene ermöglicht die Persona Einigkeit darüber zu schaffen, über wen konkret gesprochen und für wen konkret etwas geschaffen wird.

Proto-Persona | Musterprojekt

Persona: UX Designer Emilia Macintosh

"Details sind nicht nur Details, sie sind zusammen das Design"

Main Goals

UX im agilen Entwicklungsprozess integrieren und etablieren	Interaktionen und erweiterbar / erweiterlich machen, um das gemeinsame Verständnis zu beschleunigen	Gute Designs für Produkte erarbeiten und bei der Realisierung unterstützen	Tolle Software für Benutzer entwickeln
Entwicklung mit evaluierten Designs führen	Kommunikation zwischen Dev, PO & Kunde ermöglichen	Endbenutzer einbinden während Konzept und Entwicklung	

Kompetenzen

Welche Rollen und Bereiche in denen die Persona, sich selbst einschätzt

Teamplayer	Einzelkämpfer
pro-aktiv	reaktiv
100%	0%
Innovativ	Konservativ
Globaler Fokus	Lokaler Fokus
digital	analog

Hintergrund

Was hat Persona gelebt? Welche Arbeit hat sie gemacht und welche Qualitäten sind charakteristisch?

32 Jahre	Kreativbereich / Psychologiebereich	Mediengestalter / Grafiker	Fokus auf den Menschen
dynamisch, interessiert, Trends beobachten	handlich und ausdauernd	Perfektionist, akribisch, gewissenhaft	Gut in Kommunikation, überzeugend

Verantwortung

Welche Bereiche der Aufgabenstellung?

UX und UI der entwickelten Anwendung	Design-Konzepte gewährleisten	Konzept und Anfertigung der final Discovery Phase	Unbekannte Fehler im Kundenkonzept aufdecken und lösen
Design-Abklärung für die Entwicklung sicherstellen, so dass Stories für Entwickler bereit sind	Designs, die umgesetzt werden, sind validiert	Über Research Plan, Durchführung, Auswertung, Dokumentation	Endbenutzer in Diskussionsrunden einbinden, Fokus immer wieder auf Benutzer lenken

Tätigkeiten

Welche Aufgabenstellungen sind typisch?

Problemstellung verstehen	Lösungen erarbeiten	UX Qualitätsmanagement, Lösungen validieren	Dokumentation der Ergebnisse (Personas, etc.)
Über Research	Ideation, Konzept entwickeln	Design Reviews	Styleguide
Desk Research, Fachwissen aufbauen	Ideen einstimmen von allen möglichen Beteiligten	Usability Tests	
User Requirements erheben	Workshops konzipieren, vorbereiten, moderieren, auswerten	Design aller entwickelten Software reviewen und prüfen	
Personas erarbeiten und validieren	Prototypen erstellen		
	Übergabe von Designs an Entwickler		Beitrag bei Prototypprozess
	Interaction Design		Sparringspartner für PO/PM

Pains & Needs

Welche Herausforderungen hat die Persona? Welche Bedürfnisse hat sie?

Abhängigkeit von Prozessen und Zuständigkeiten beim Kunden	Fehlender Zugang zu Nutzern	Vorwissen des Kunden auf UX, Anzweifeln von Anfang an
Kein Verständnis für gutes UX/UI bei Kunden / Entwicklern / ...	Raum für Fragen schaffen, damit eigene, bessere Zusammenhänge im Team möglich ist	Unterstützung durch "Etablierte" für Sicherstellung der technischen Mischbarkeit
Kein gutes Konzept wie UX im Projekt integriert ist	Klare Verantwortlichkeiten kommunizieren, wenn im Projekt & was nicht (Pro, User or Dev...)	Ansprechpartner für Design beim Kunden
Zu wenig Zeit für Requirements Engineering, bevor Entwicklung anfragt	Erwartung des Kunden, dass sofort mit Coding angefangen wird	Zugang zu Design-Tools
Weggenommene Arbeit durch ständige Änderungen und dadurch zu wenig Zeit für Design	Kommunikation, Feedbackmechanismen entlang nicht vernachlässigen. Brauch Zeit um sie aufzubauen	Zugang zu CI und Sparringspartner vom Kunden
	Ausreichend Vorlauf, um Zeit für Fehler aber auch bessere Ideen zu haben	
	Fachwissen über Dozine des Projekts	

Arbeitet mit

Welche Rollen sind typisch? Wie interagiert die Persona mit anderen?

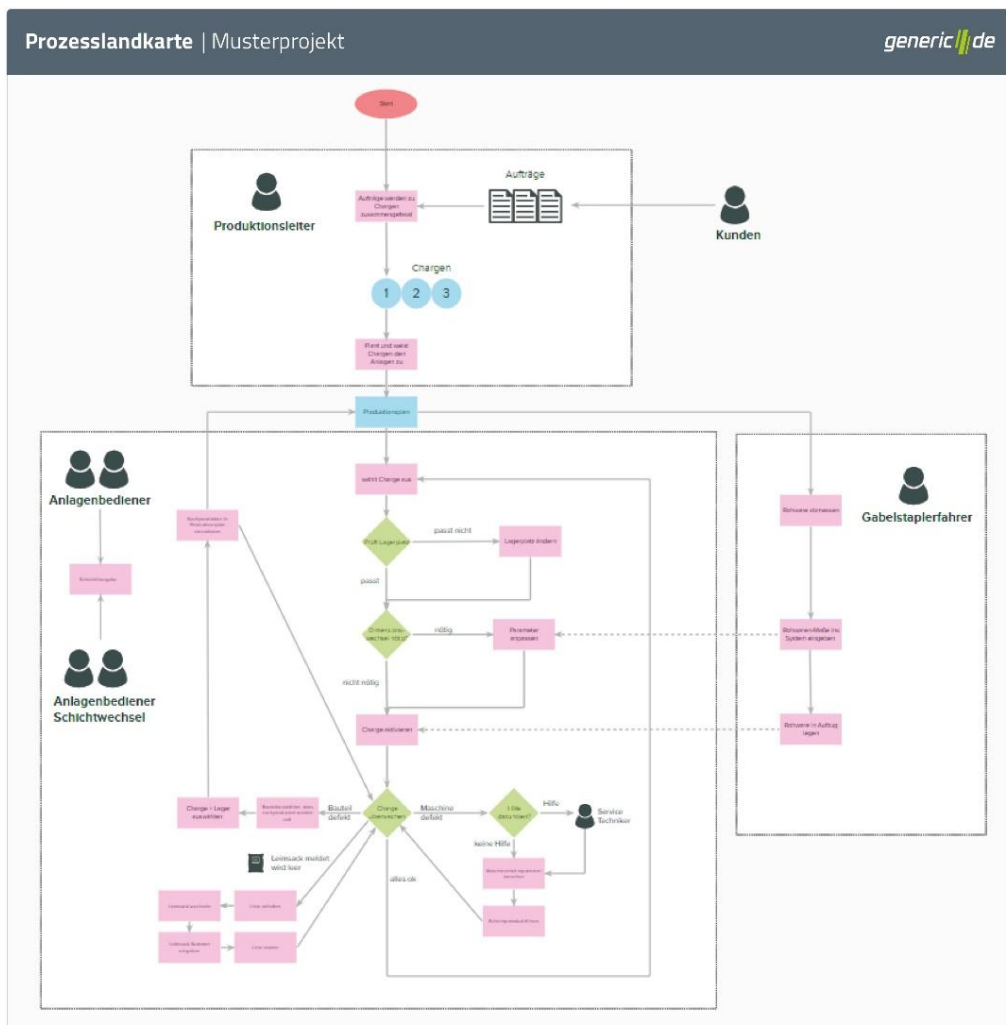
Entwickler	PO (generic)
PO (Kunde)	Fachexperte beim Kunden
Anderer UXDs	Endbenutzer
Scrum Master	Projektleiter
	Vertriebler

Service-Beschreibung: Konzeptentwicklung

Seite 10

2.2.2 Prozesslandkarte

Die Prozesslandkarte ist eine übergeordnete Sicht auf die Prozesse einer Software mitsamt der Schnittstellen nach außen. Sie beinhaltet in unserem Kontext meistens die Geschäftsprozesse inklusive des Task Flows – die Schritte, die ein Benutzer durchlaufen muss, um eine Aufgabe zu erfüllen. Die Prozesslandkarte bildet den Ist-Zustand ab und dient dazu, die größten Pain Points und Hürden im Prozess zu identifizieren.



2.2.3 Affinity Map (Ergebnisse der Benutzerforschung & Stakeholder Interviews)

„Wer mit Benutzern spricht, spart Zeit und Geld. Jede Stunde, die für Nutzerforschung aufgewendet wird, spart mehr als 5 Stunden Programmier- und Designzeit.“

(Cindy Alvarez - Lean Customer Development)

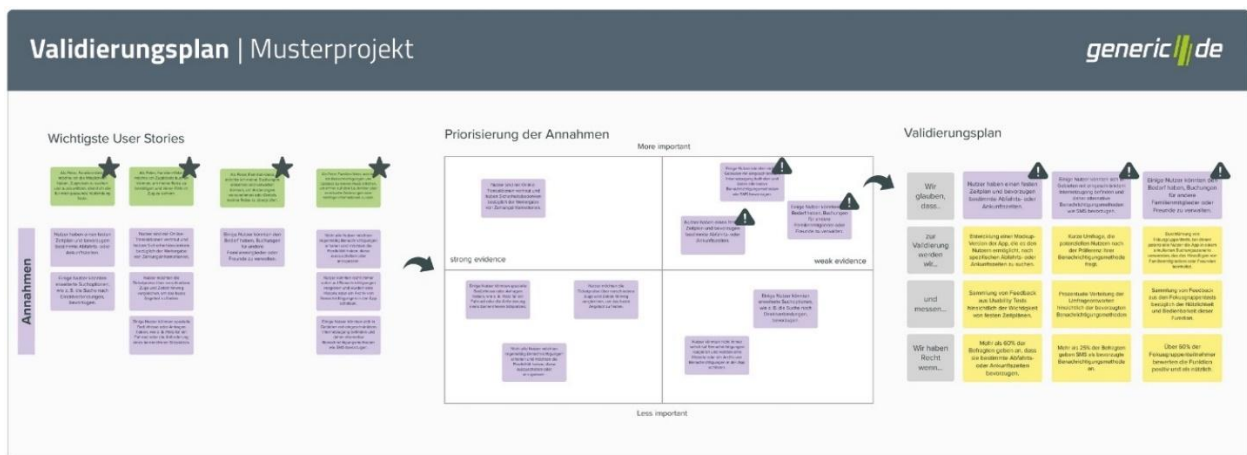
Bei Bedarf werden Inhalte der Prozesslandkarte und der Proto-Personas durch User Research und Stakeholder Interviews validiert und angereichert. So können beispielsweise durch Beobachten des Benutzers an dessen Arbeitsplatz die Rahmenbedingungen in Bezug auf die Arbeitsumgebung identifiziert werden, die den Erfolg des Produkts gefährden. Die Ergebnisse werden in einer Affinity Map zusammengetragen und sortiert, bevor sie in andere vorhandene Artefakte mit einfließen.



2.3.3 Validierungsplan

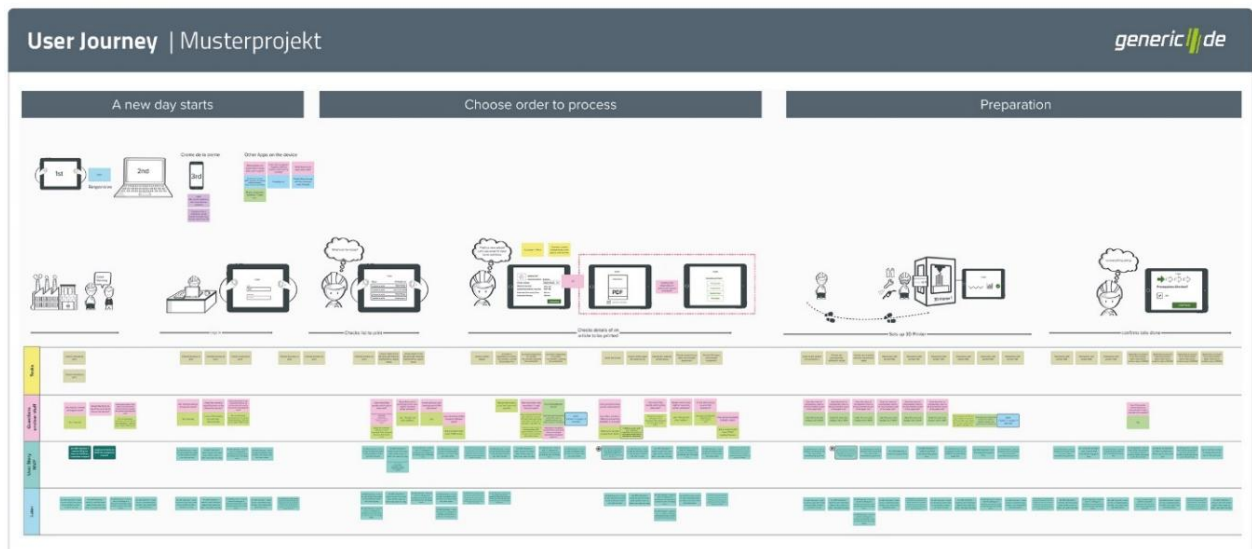
Die wichtigsten User Stories werden nun priorisiert und zugehörige Annahmen und die zugehörigen Risiken identifiziert. Wir erstellen einen Validierungsplan, in dem entschieden wird, welche der Risiken minimiert werden sollen in Hinblick auf:

- **Desirability:** Hat das Produkt für die Benutzer eine Bedeutung und Relevanz?
- **Usability:** Können die Benutzer das Produkt effizient nutzen?
- **Feasibility:** Kann das Produkt mit neuen/bestehenden Technologien verwirklicht werden?



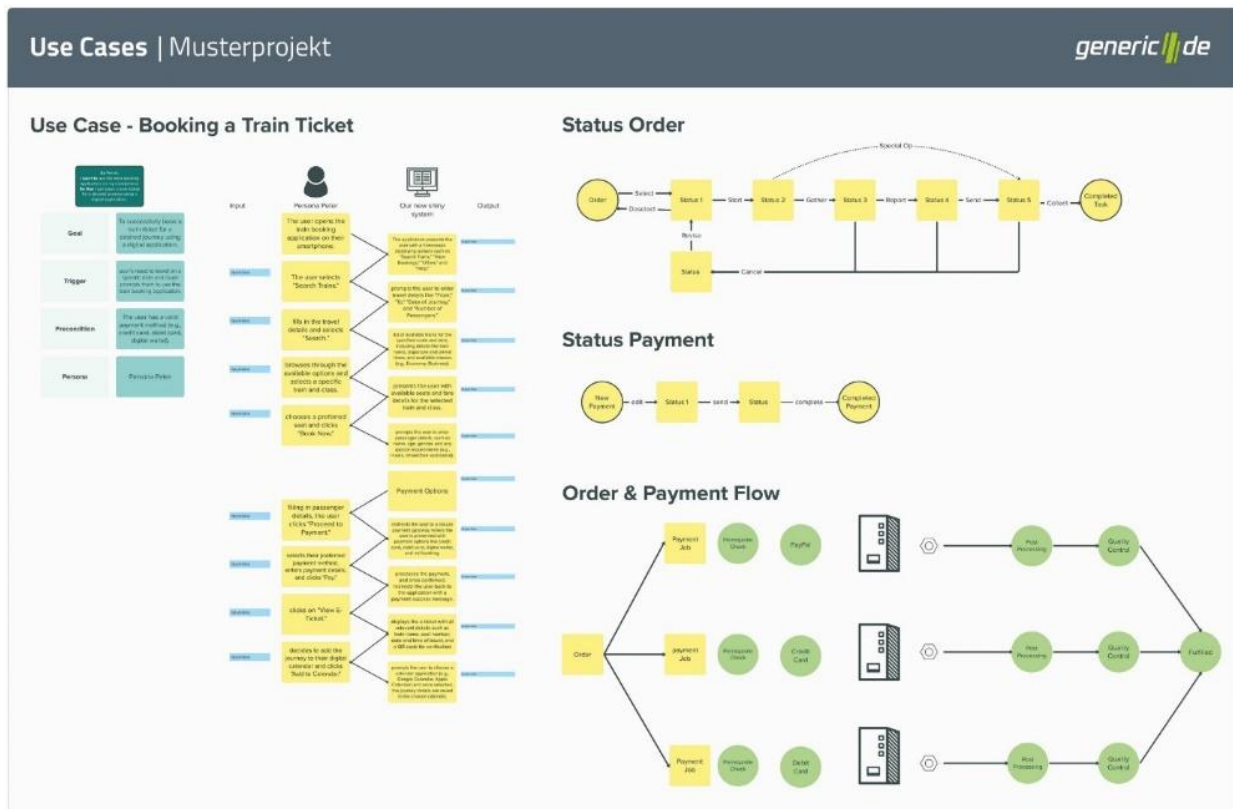
2.3.4 User Journey

Bei der Beschreibung der User Journey geht es darum, die Erfahrungen eines Benutzers über mehrere Interaktionspunkte hinweg zu verstehen. Dabei werden alle Kanäle und Informationsquellen berücksichtigt, die der Benutzer im Prozess verwendet. Durch die User Journey wird die User Experience der vorher priorisierten User Stories im Detail beschrieben.



2.3.5 Use Cases (Definition der Anwendungsfälle)

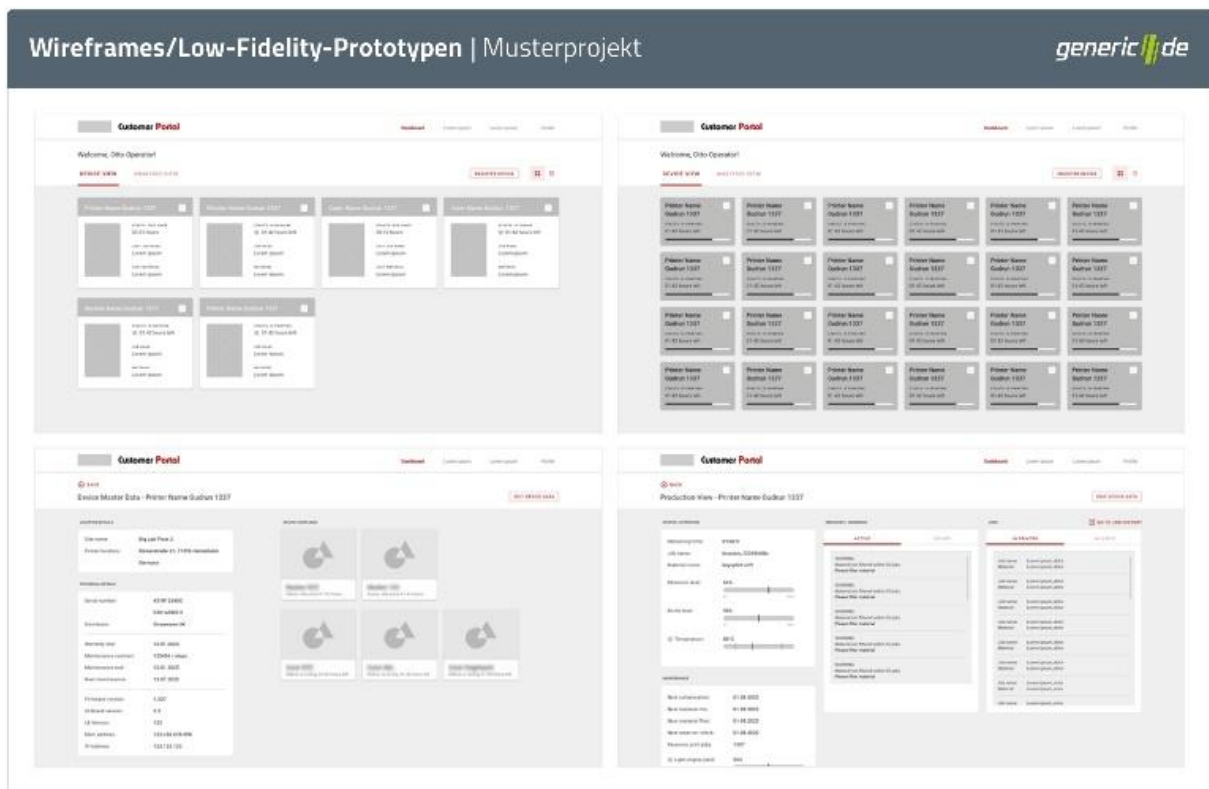
Ein Use Case beschreibt, wie ein Benutzer mit dem Produkt interagiert, um ein bestimmtes Ziel zu erreichen. Im Vergleich zur User Story zeigt ein Use Case in detaillierter Form, wie der Benutzer den Nutzen erzielt. Ein Use Case beschreibt neben dem Benutzer auch Vorbedingungen, Auslöser und Ergebnis.



2.3.6 Interaktions-Design und Konzeptvalidierung mittels Wireframes und UX-Prototypen

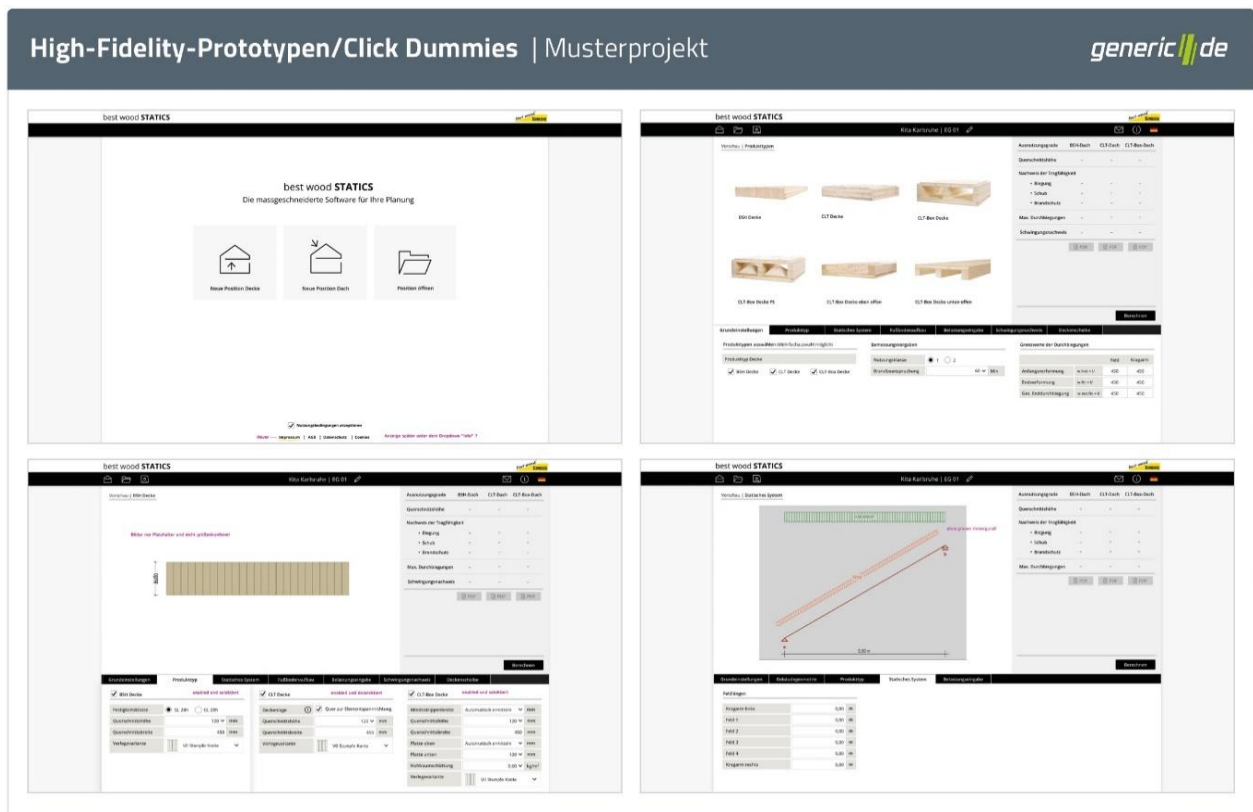
Je nach Validierungsplan werden Low- oder High-Fidelity-Prototypen zur Konzeptvalidierung verwendet. Dabei wird untersucht, ob die wichtigsten Features für die Benutzer den erhofften Mehrwert liefern (Desirability) und ob die Benutzer das Feature effizient nutzen können (Usability).

Erste Skizzen des User Interfaces (Wireframes): Um ein besseres Bild zu bekommen und um das gemeinsame Verständnis zu erleichtern, visualisieren wir bereits während der Ideenfindung erste Skizzen der wichtigsten User Interfaces (UI) – sogenannte Wireframes.



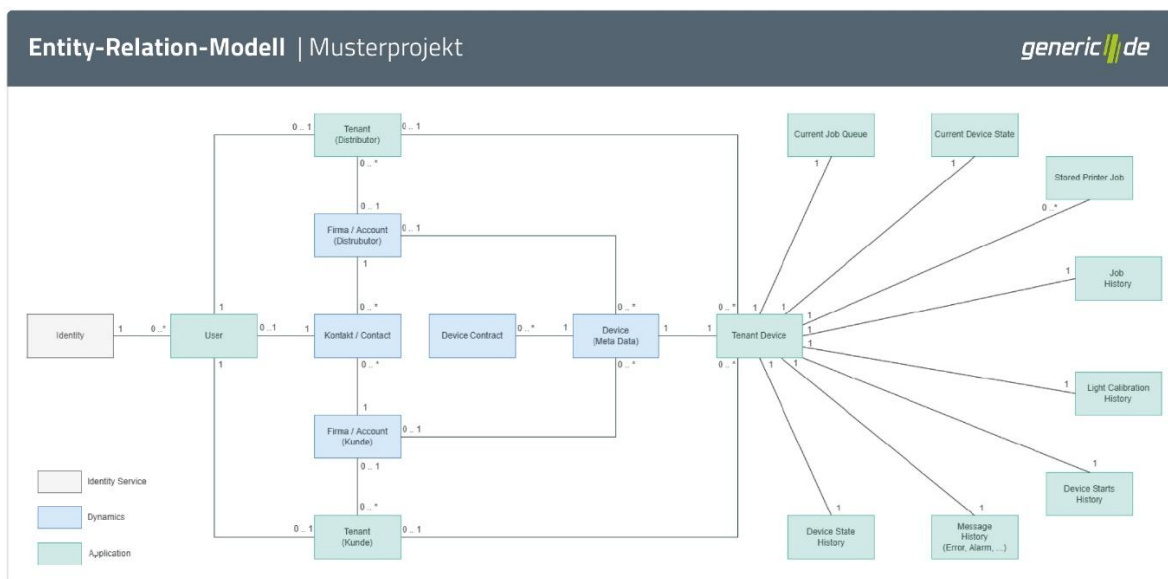
High-Fidelity-Prototypen/Click Dummies: High-Fidelity-Prototypen simulieren die Applikation in Aussehen und Funktion und machen die Lösung erfahrbar. D.h. Benutzer können sich hier, z.B. während eines Usability Tests, durch ein konkretes Nutzungsszenario klicken, ohne dass dafür Code geschrieben werden muss.

High-Fidelity Prototypen dienen auch dazu, ein einheitliches Verständnis im ganzen Projektteam sicherzustellen und sowohl Interaktion als auch das visuelle Design für die Implementierung klar zu beschreiben. Das Design ist am Ende der Konzeptentwicklung zwar nicht vollständig fertig, aber viele wichtige Entscheidungen sind bereits getroffen.



2.3.7 Entity-Relation-Modell (ER-Modell)

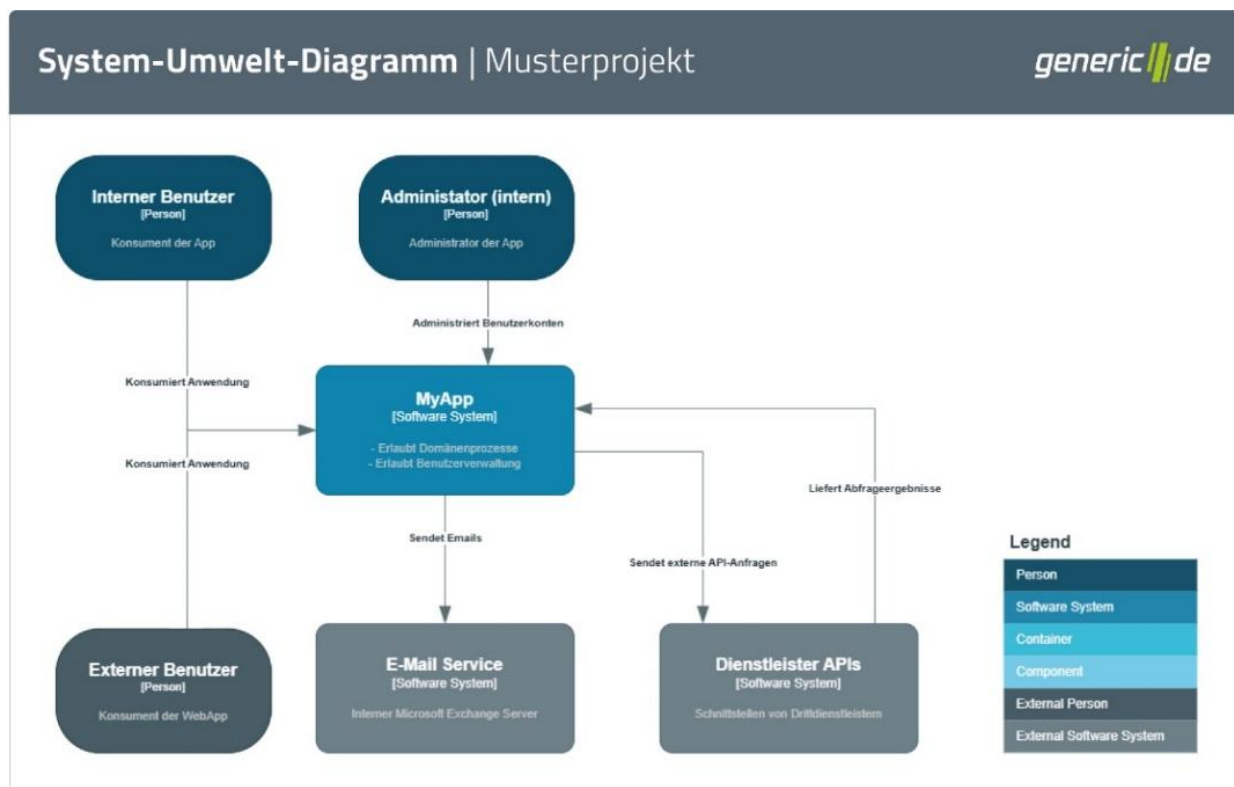
Um Produkthanforderungen möglichst präzise und frei von Interpretationsmöglichkeiten zu formulieren, müssen Begriffe eindeutig sein. Mit Hilfe eines ER-Modells werden die Entitäten der Fachdomäne und deren Zusammenhänge dargestellt. Dies ermöglicht dem gesamten Projektteam ein gleiches Verständnis von den Anforderungen sowie von der Domäne.



2.3.8 System-Umwelt-Diagramm

Ein System-Umwelt-Diagramm oder Produkt-Kontext-Diagramm betrachtet ein Softwareprodukt als Ganzes und zeigt dessen Input und Output von bzw. zu externen Faktoren. Es hilft beim Definieren der Systemgrenzen. Dabei zeigt es zum einen die Anwendungen als Interaktionsmöglichkeiten mit dem System durch einen Benutzer und zum anderen die Schnittstellen zu weiteren Systemen. Dadurch können Anforderungen klarer für das Produkt abgegrenzt und Abhängigkeiten transparenter gemacht werden.

Mit Hilfe dieser Darstellung können der Start und Endpunkt von Use-Cases leichter identifiziert bzw. verknüpft werden. Darüber hinaus gibt das Diagramm eine visuelle Vorstellung der Architekturvision und kann helfen, alle Beteiligten auf eine gemeinsame Zielvorstellung auszurichten.

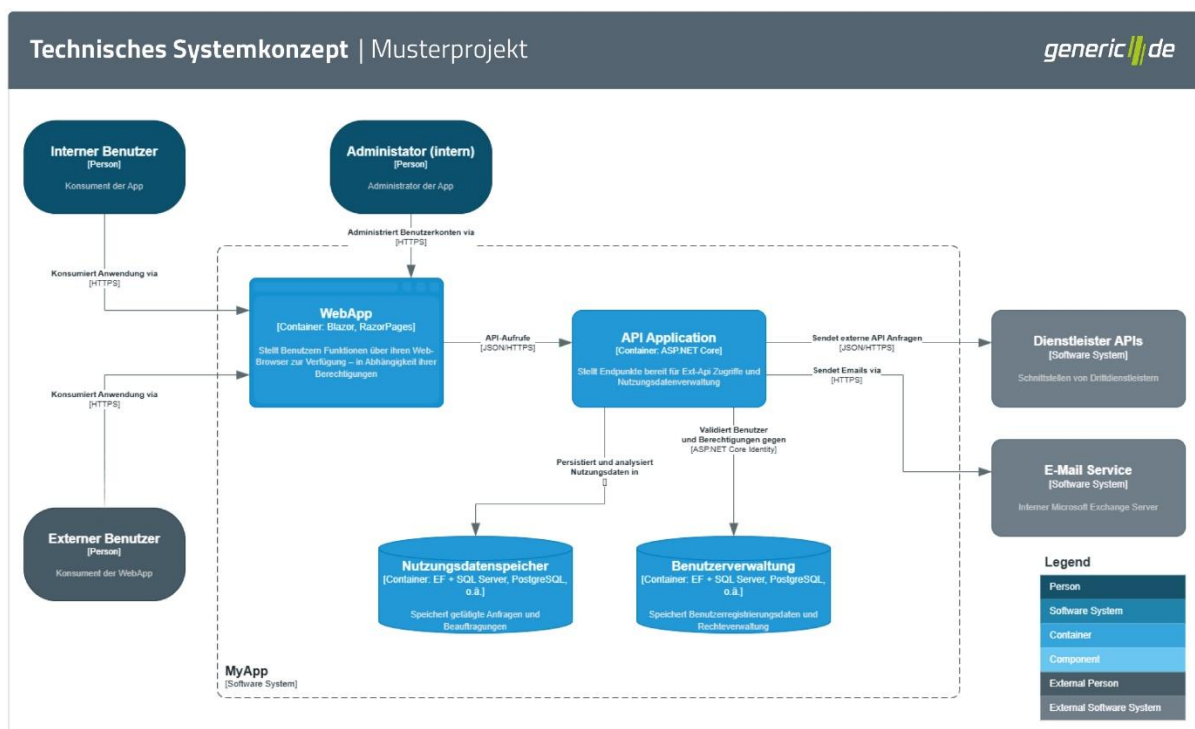


2.3.9 Technisches Systemkonzept

Ein technisches Systemkonzept bildet gemeinsam mit dem Interaktionskonzept den Plan für die konkrete Umsetzung eines Softwarevorhabens. Wesentliche Bestandteile sind:

- Analyse der technischen Rahmenbedingungen
- Technologie-Stack, PoC
- konkretes Architekturkonzept in Form eines C4-Modells

Weitere Ausarbeitungen und Detaillierungen basieren auf dieser Grundlage. Die jeweiligen Elemente können sich gegenseitig beeinflussen. Eine Anforderung zur Datenhaltung kann beispielsweise bestimmte Systembestandteile bedingen. Gleichzeitig kann eine ausgewählte Technologie gewisse Beschränkungen bedeuten. Wenn es der Zeitrahmen zulässt, können Machbarkeiten und Technologien mithilfe von Proof-of-Concept Implementierungen evaluiert werden, um das Feasibility-Risiko zu minimieren.




2.5 Vorbereitung der Entwicklung

2.5.1 Validiertes Backlog

Das Product Backlog ist eine flexible, emergente und priorisierte Liste an Anforderungen. Das bedeutet, dass die wichtigsten, als nächstes zu implementierenden Anforderungen ganz oben positioniert und hinreichend genau besprochen wurden, sodass alle Beteiligten das gleiche Verständnis darüber haben. Die Komplexität der User Stories wird von den Entwicklern geschätzt, was bei der Lösungsfindung unterstützt sowie zur Priorisierung und für eine Vorhersage genutzt werden kann. Für die Schätzungen treffen wir gemeinsam mit Ihnen Annahmen. Das Backlog verweist auf validierte Prototypen und alle visuellen Designvorgaben.

Die Arbeit mit einem Product Backlog verbessert die Wertschöpfung, erhöht die Effizienz und Parallelisierbarkeit von Tätigkeiten und ermöglicht damit agiles Arbeiten.

Product Backlog | Musterprojekt
generic  de

Customer Portal Team ☆ 🔍

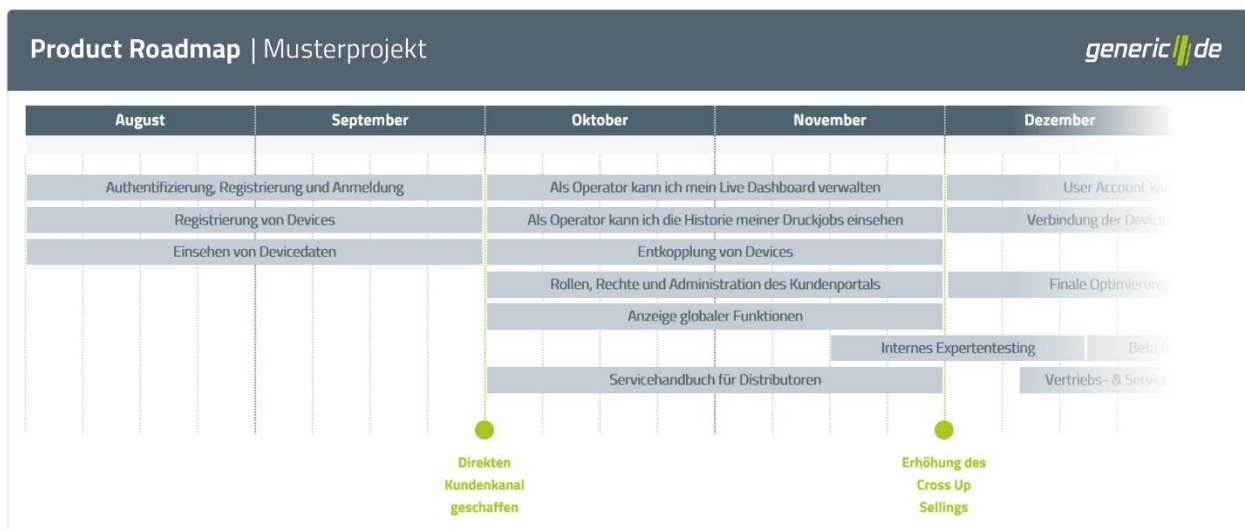
Backlog Analytics | + New Work Item 👁 View as Board 🔗 Column Options ⋮

+	Order	ID	Priority	Title
+		1134	2	👑 Online-Einkaufsplattform vertrauenswürdiger gestalten
		1135	2	📦 Produktbewertungen hinzufügen
		1136	2	🗣 Als registrierter Benutzer möchte ich Produkte bewerten können, damit ich meine Erfahrungen teilen und anderen Käufern bei ihrer Entscheidung helfen kann.
		1137	2	📄 Entwurf der Benutzeroberfläche
		1138	2	📄 Bewertungssystem implementieren
		1139	2	📄 Bewertungsformular erstellen
		1140	2	📄 Validierung und Datenschutz
		1141	2	📄 Anzeige der Bewertungen auf Produktseiten
		1142	2	📄 Filter- und Sortieroptionen
		1143	2	👑 Verbesserung des Produktfilter- und Suchsystems

2.5.2 Product Roadmap

Die Product Roadmap visualisiert die geplante Entwicklung mitsamt den Projektmeilensteinen sowie der wichtigsten Funktionen Ihres Produkts über einen bestimmten Zeitraum. Dadurch erhalten Sie einen Überblick darüber, wie sich Ihr Produkt im Laufe der Zeit entwickeln wird, um die strategischen Ziele zu erreichen und den Bedürfnissen der Benutzer gerecht zu werden.

Das Deliverable hilft Ihnen dabei, die Produktentwicklung strategisch zu lenken, die Zusammenarbeit zu verbessern, den Entwicklungsprozess zu kommunizieren und sicherzustellen, dass das Endprodukt den Anforderungen der Benutzer gerecht wird.



3 Fazit

Kern der vorgelagerten Konzeptentwicklung ist die Beantwortung zwei essenzieller Fragen:

- **Welchen Wert schaffen wir für die Benutzer? (Desirability, Usability)**
- **Wie erschaffen wir das Produkt? (Feasibility)**

Viele Projekte starten zu früh mit der Entwicklung, ohne diese zwei wesentlichen Fragen ausreichend zu beantworten.

Solange man nicht einigermaßen weiß, welchen Mehrwert man den Benutzern bietet und warum das Produkt für sie wichtig ist, tappt man im Dunkeln. Dann besteht ein hohes Risiko des Scheiterns. Dieses Anwenderrisiko wird minimiert durch Erstellung von Prototypen und Validierung des Konzepts mit Endnutzern.

Das Feasibility-Risiko wird reduziert durch die Ausarbeitung der Anforderungen. Dabei entsteht ein gemeinsames Verständnis der Anforderungen und der erwarteten Funktionalität. Nach unserem agilen Ansatz definieren wir nur die höchst-priorisierten Features im Detail. Niedrig-priorisierte Features werden aufgenommen, aber zu Beginn nur auf hoher Flugebene betrachtet. Agilität bedeutet nicht, dass es keinerlei Dokumentation gibt! Anforderungen werden allen beteiligten Personen in verständlicher Form verfügbar gemacht.

Die Ergebnisse der Konzeptentwicklung sind Arbeitsgrundlage und Blaupause Ihres Softwareprodukts. Dabei können Sie sich sicher sein, dass wir nur genau so viel Zeit und Arbeit investieren, wie nötig ist, um in die Entwicklungsphase zu starten. Wir entscheiden gemeinsam mit Ihnen, welche Deliverables und Methoden für Ihr Vorhaben am erfolgversprechendsten sind.

Ihr Kontakt bei generic.de

Kontaktieren Sie uns für weitere Informationen und lassen Sie uns gemeinsam die Zukunft Ihrer Softwareentwicklung gestalten.



Jörg Lenz

Key Account Management

Tel.: +49 721 619096-50

E-Mail: joerg.lenz@generic.de



Marcus Schönherr

Sales Manager

Tel.: +49 721 61 90 96-43

Mobil: +49 174 7026360

E-Mail: marcus.schoenherr@generic.de



Sebastian Betzin

CTO & Vorstand

E-Mail: sebastian.betzin@generic.de

generic.de software technologies AG | Johann-Georg-Schlosser-Str. 66 | D - 76149 Karlsruhe

Vorstand: Michael Puder (Vorsitzender), Michael Speer, Sebastian Betzin | Aufsichtsratsvorsitzender: Jürgen Betzin

Telefon: +49 721 619096-0 | info@generic.de | www.generic.de

HRB 705226 | Amtsgericht Mannheim | USt.-IdNr. DE202990883

Volksbank Karlsruhe | IBAN: DE73 6619 0000 0054 4641 00 | BIC: GENODE61KA1

Deutsche Bank | IBAN: DE49 6607 0024 0099 8856 00 | BIC: DEUTDEB660