

Projekthandbuch

Learn to innovate

Eine spielerische Applikation zum Vorstellen der
sovanta Innovation Factory for SAP BTP

Auftragnehmer: Sepia



Auftraggeber: sovanta AG



Softwareentwicklungsprojekt im Sommersemester 2023
an der Hochschule Mannheim



Version 2.0 vom 23.05.2023
Verantwortlich: Stephan Halder

Änderungsverzeichnis

Das Projekthandbuch entwickelt sich iterativ weiter und Änderungen in den Versionen werden hier protokolliert.

Versionsnummer	Datum	Änderungen	Autor	Prüfer
2.0	23.05.2023	Kleinere Anpassungen und Korrekturen, Feedback von Tutorin eingearbeitet	Stephan Halder	Fabian Hoppe, Mike Menzel
1.4	19.05.2023	Deckblatt angepasst, aktuellste Version im Änderungsverzeichnis an erste Stelle sortiert,	Stephan Halder	
1.3	15.05.2023	Rollenbeschreibungen angepasst, Auswahl der Risiken angepasst und Beschreibung des Risikomanagements eingefügt	Stephan Halder	
1.2	12.05.2023	Referenzen zu den Tools hinzugefügt, Abschnitt Deadlines hinzugefügt, Teilziele in Kapitel Projektzieleplan angepasst	Stephan Halder	
1.1	11.05.2023	Änderungsverzeichnis über Inhaltsangabe geschoben, Projektauftraggeber von Deckblatt entfernt, Kapitel Projektmeilensteinplan entfernt, Einleitung hinzugefügt, Rechtschreibfehler korrigiert, Tabellen für Kontakte in Kapitel 1 eingefügt, fehlende Kontaktdaten nachgetragen, Referenzen eingefügt, Glossar erweitert	Stephan Halder	
1.0	25.04.2023	Fehler korrigiert, Meilenstein für den 23.05.2023 hinzugefügt	Stephan Halder	
0.3	25.04.2023	Änderungen aus der Problemliste von Dokument-Review hinzugefügt.	Stephan Halder	
0.2	20.04.2023	Weitere Inhalte hinzugefügt, Struktur angepasst, Meilensteinplan hinzugefügt.	Stephan Halder	
0.1	17.04.2023	Struktur festgelegt, erste Inhalte hinzugefügt	Stephan Halder	

Inhaltsverzeichnis

1. Einleitung.....	4
2. Projektüberblick.....	4
2.1 Projektauftraggeber.....	4
2.2 Projektziel.....	4
2.3 Projekt Aufgabenstellung.....	4
2.4 Projektteammitglieder.....	5
2.5 Tutor.....	5
2.6 Management.....	5
2.7 Projektdetails.....	6
3. Der Kunde.....	7
3.1 Ansprechpartner.....	7
3.2 Kundensprechstunde.....	7
4. Prozessmodell.....	8
5. Meetings.....	9
6. Deadlines.....	10
6.1 Vom Management vorgeschriebene Deadlines.....	11
6.2 Vom Kunden vorgeschriebene Deadlines.....	11
6.3 Teaminterne Deadlines.....	11
7. Projektzieleplan.....	12
8. Rollenverteilung.....	13
9. Qualitätsmanagement.....	14
9.1 Review von Dokumenten.....	14
9.2 Testen von Code.....	14
9.2 Reviewen von Code / Git Richtlinien.....	15
9.3 Coding Richtlinien.....	15
10. Projektrisiken.....	16
10.1 Aktuelle Risiken.....	16
10.2 Risikomatrix.....	17
11. Tools.....	18
11.1 Jira.....	18
11.2 GitHub.....	18
11.3 Discord.....	18
11.4 WhatsApp.....	18
11.5 Google Drive.....	19
11.6 Microsoft Teams.....	20
12. Übersicht Dokumente.....	21
13. Glossar.....	22

1. Einleitung

In diesem Projekthandbuch sind alle relevanten Informationen bezüglich des Vorgehens und verwendeter Tools sowie Informationen über unser Team beschrieben. Dieses Projekt findet im Rahmen des Projektsemesters der Hochschule Mannheim statt.

Wörter, die im Glossar beschrieben werden, werden bei erster Benutzung in diesem Handbuch kursiv geschrieben.

2. Projektüberblick

2.1 Projektauftraggeber

Der Auftraggeber des Projektes ist die sovanta AG. Die sovanta AG ist ein Unternehmen mit Hauptsitz in Heidelberg, das als Experte für verbesserte User Experience in SAP Software Systemen gilt.

2.2 Projektziel

Das Ziel dieses Projektes ist, ein Konzept und eine prototypische Applikation zu entwickeln, die Messebesuchern die Arbeit der sovanta AG und die sovanta Innovation Factory for SAP BTP spielerisch näherbringen soll. Das Projektziel ist genauer in der [Anforderungsspezifikation](#) beschrieben.

2.3 Projekt Aufgabenstellung

“Experience the Power of the sovanta Innovation Factory for *SAP BTP*. Erstelle ein überzeugendes Konzept und eine prototypische Applikation, die Besucher von IT Messen spielerisch von den Möglichkeiten der *sovanta Innovation Factory* für SAP BTP überzeugt.” - sovanta.

2.4 Projektteammitglieder

Name	E-Mail	Studiengang
Aleksandra Kun	aleksandra.kun@stud.hs-mannheim.de	Design
Andreas Roth	andreas.roth@stud.hs-mannheim.de	Informatik
Bibiana Lazarová	2225009@stud.hs-mannheim.de	Design
Eren Saglam	eren.saglam@stud.hs-mannheim.de	Informatik
Fabian Hoppe	fabian.hoppe@stud.hs-mannheim.de	Informatik
Mike Menzel	mike.menzel@stud.hs-mannheim.de	Informatik
Stephan Halder	stephan.halder1@stud.hs-mannheim.de	Informatik

2.5 Tutor

Name	E-Mail
Sophie Kühn	sophie.kuehn@stud.hs-mannheim.de
Marco D'Angelo	m.dangelo@lba.hs-mannheim.de

2.6 Management

Name	E-Mail
Prof. Dr. Peter Knauber	p.knauber@hs-mannheim.de
Prof. Dr. Wolfgang Schramm	w.schramm@hs-mannheim.de

2.7 Projektdetails

Projekt Starttermin: 03.04.2023

Projekt Abschlusspräsentation: 26.06.2023

Team-Website: <https://sepia2023.github.io/sepia.github.io/>

Team E-Mail-Adresse: sepia.mannheim.2023@gmail.com
Das Team nutzt diese gemeinsame Team-E-Mail-Adresse, die von Google gehostet wird. Schriftliche Kommunikation mit dem Kunden findet über diese E-Mail-Adresse statt.

Arbeitszeiten: Die Arbeitszeiten sind in der Regel von 9:30 Uhr bis um 17:30 Uhr. Ausnahmen davon sind nach vorheriger Absprache mit dem Team möglich.

3. Der Kunde

sovanta AG

Mittermaierstraße 31
69115 Heidelberg
Deutschland



Kontakt: semesterprojekt2023@sovanta.com

3.1 Ansprechpartner

Ansprechpartner	
Name	Rolle
Jakob Frankenbach	Head of Development
Larissa Haas	Senior Data Scientist
Alina Meiseberg	Senior Data Scientist
Nils Janßen	Senior Software Engineer
Thomas Bechberger	Senior User Experience Designer
Isabel Chapero Rueda	Marketing Specialist
Louise Hebestreit	Marketing Specialist

Der Kunde ist über E-Mail erreichbar. Dazu wird die oben genannte E-Mail-Adresse verwendet, um das Anliegen zu beschreiben.

3.2 Kundensprechstunde

In der Regel findet freitags von 9:15 Uhr bis um 10:15 Uhr eine Kundensprechstunde statt. Die Kundensprechstunde wird über *Microsoft Teams* abgehalten. Ausnahmen werden von dem Kunden an die E-Mail des Teams mitgeteilt. An der Kundensprechstunde nehmen in der Regel nur die Kundenbeauftragten teil. In besonderen Fällen können aber auch andere Teammitglieder an der Sprechstunde teilnehmen.

4. Prozessmodell

Prozessmodell	
Prozessmodell:	<ul style="list-style-type: none">• Adaptiertes Scrum
Sprints:	<ul style="list-style-type: none">• Sprint Beginn: Mittwoch 09:30 Uhr• Sprint Ende: Dienstag 17.30 Uhr
Meetings:	<ul style="list-style-type: none">• Daily Team Meeting• Sprint Planning Meeting• Sprint Retrospektive
Anpassungen an dieses Projekt	<ul style="list-style-type: none">• Jeder ist Teil des Development Teams (auch Teamleiter)• Weitere Rollen im Development Team• Kein Product Owner• Kein Scrum Master

Das Team benutzt einen iterativen Prozess mit Scrum Elementen. Dabei beginnt ein Sprint mittwochs. Mittwochs nachmittags, nach dem Jour Fixe oder nach dem Review (siehe Kapitel [5 Meetings](#)), wird eine Sprint Retrospektive durchgeführt, um den letzten Sprint zu reflektieren. Der nächste Sprint wird montags im Sprint Planning Meeting geplant, dabei werden auch Arbeitspakete definiert. Nähere Informationen zu den Meetings sind in Abschnitt [5. Meetings](#) zu finden.

Im Gegensatz zu Scrum haben wir keinen Product Owner und keinen Scrum Master. Außerdem ist jeder, auch der Teamleiter, Teil des Development Teams. Innerhalb des Development Teams können noch weitere Rollen verteilt werden, dazu gehören Rollen wie Tester, Frontend Entwickler und Backend Entwickler. Diese Rollen werden je nach Bedarf während des Sprint Planning Meetings für den nächsten Sprint zugeteilt.

5. Meetings

Meetings finden in der Regel vor Ort an der Hochschule im Raum L303 statt. Ausnahmen sind nach vorheriger Kommunikation im Team möglich. An den Meetings nehmen in der Regel alle Teammitglieder des Studiengangs Informatik teil (siehe [Projektteammitglieder](#)).

Bezeichnung	max. Dauer	Tag Uhrzeit	Beschreibung
Sprint Planning Meeting	2 Stunden	Montag 13 Uhr	<p>Beim Sprint Planning Meeting wird der nächste Sprint geplant. Das Sprint Planning Meeting findet montags um 13 Uhr statt, da danach noch genügend Zeit ist, um die Folien für einen Jour fixe oder ein Review vorzubereiten. (siehe Jour fixe und Review)</p> <p>Dabei bestimmt der Teamleiter zusammen mit dem Team die Ziele für den nächsten Sprint und es werden Arbeitspakete mit Akzeptanzkriterien vom Team definiert. Danach wird der Zeitaufwand für die Arbeitspakete von jedem Teammitglied mit <i>Fibonacci Zahlen</i> anonym geschätzt. Im Anschluss werden die Schätzungen besprochen und gemeinsam eine Schätzung für das Arbeitspaket festgelegt. Die Arbeitspakete werden in mithilfe des Tools Jira verwaltet (siehe Kapitel 9.1 Jira).</p> <p>Abschließend werden die Verantwortlichkeiten für die Arbeitspakete verteilt und gegebenenfalls die Rollenverteilung für den nächsten Sprint angepasst.</p>
Daily Team Meeting	15 Minuten	täglich 13 Uhr	<p>Es wird besprochen, was man seit dem letzten Daily Meeting geschafft hat und was man bis zum nächsten Daily Meeting schaffen will und welche Schwierigkeiten man hatte. Außerdem gibt es eine Stimmungsabfrage beim Meeting, um einen Überblick über die Stimmung im Team zu bekommen. Die Stimmung wird im gleichnamigen Dokument dokumentiert.</p> <p>An Tagen, an denen ein Sprint Planning Meeting oder eine Sprint Retrospektive stattfindet, wird das Daily Meeting im Rahmen des entsprechenden Meetings durchgeführt.</p>
Sprint Retrospektive	1 Stunde	mittwochs nach dem Jour fixe / Review	<p>Bei der Sprint Retrospektive wird innerhalb des Teams über die Zusammenarbeit des Teams im letzten Sprint reflektiert und gegebenenfalls Verbesserungsmaßnahmen erarbeitet.</p>

Bezeichnung	max. Dauer	Tag Uhrzeit	Beschreibung
Jour fixe	4 Stunden (Vortrag 5 Minuten)	mittwochs 10 Uhr	Beim Jour fixe wird dem Management der aktuelle Stand des Projektes präsentiert. Dafür müssen Folien mithilfe eines Templates erstellt werden, die bis um 17 Uhr am Dienstag vor dem Jour fixe auf der Teamwebsite abgegeben werden müssen. Inhalt dieser Folien ist ein Rückblick auf den vergangenen Sprint, ein Blick auf den aktuellen Sprint, aktuelle Risiken und eine Projektampel. Der Vortrag sollte 5 Minuten lang sein.
Review	5 Stunden (Vortrag 20 Minuten)	alle zwei Wochen mittwochs 10 Uhr	Das Review besteht aus einem Jour fixe Teil und einem Produktteil. Der Jour fixe Teil sollte neben den normalen Inhalten eines Jour fixes noch einen Rückblick zum letzten Review und ein Ausblick zum nächsten Review haben. Im Produktteil soll dem Management unser aktuelles Teilprodukt gezeigt werden. Der Review Vortrag sollte etwa 20 Minuten dauern, dabei sollte der Jour fixe Teil etwa 7 Minuten und der Produktteil etwa 13 Minuten dauern. Auch für diese Präsentation gibt es ein Template .

6. Deadlines

Es gibt verschiedene Deadlines während diesem Projekt.
Die Deadlines können in drei Kategorien eingeteilt werden:

- vom Management vorgeschriebene Deadlines,
- vom Kunden vorgeschriebene Deadlines
- und Teaminterne Deadlines

6.1 Vom Management vorgeschriebene Deadlines

Reviews und Jour fixe finden in der Regel mittwochs statt.

Folien, die für ein Jour fixe beziehungsweise für ein Review erstellt werden, müssen am Tag vor der Präsentation bis um 17 Uhr auf der Teamwebsite hochgeladen werden.

Zum Review müssen zusätzlich noch weitere Dokumente abgegeben werden. Dazu zählen die Anforderungsspezifikation, das Projekthandbuch und die Architekturdokumentation. Diese Dokumente müssen ebenfalls am Tag vor dem Review bis um 17 Uhr auf der Teamwebsite hochgeladen werden.

Nach dem Review müssen Gruppenberichte über die anderen Teams abgegeben werden. Diese müssen bis um 14 Uhr am Folgetag des Reviews abgegeben werden.

6.2 Vom Kunden vorgeschriebene Deadlines

Die Abschlusspräsentation findet am 26.06.2023 beim Kunden in Heidelberg statt.

Bis zum 16.06.2023 muss dem Kunden eine technische Dokumentation, ein UX-Konzept, und ein ausführbares Produkt dem Kunden abgegeben werden. Fehler können noch bis zur Abschlusspräsentation behoben werden.

6.3 Teaminterne Deadlines

Folien für das Jour fixe beziehungsweise das Review müssen bis Montagabend bereit zum Reviewen sein und können an die Tutorin zur Überprüfung gesendet werden.

7. Projektzieleplan

Projektzieleplan	
Zielart	Projektziele
Projektziel:	Das Ziel dieses Projektes ist, ein Konzept und eine prototypische Applikation zu entwickeln, die Messebesuchern die Arbeit der sovanta AG und die sovanta Innovation Factory for SAP BTP spielerisch näherbringen soll. Das Projektziel ist genauer in der Anforderungsspezifikation beschrieben.
Teilziele:	<p>Ziele für den 25.04.2023:</p> <ul style="list-style-type: none"> ➤ Anforderungsspezifikation ➤ Vertikaler Prototyp <p>Ziele für den 04.05.2023:</p> <ul style="list-style-type: none"> ➤ Idee anhand eines Low fidelity Prototypen dem Kunden vorgestellt <p>Ziele für den 16.05.2023</p> <ul style="list-style-type: none"> ➤ Architekturdokument dem Management abgegeben <p>Ziele für den 23.05.2023</p> <ul style="list-style-type: none"> ➤ Dem Kunde wurde ein ausführbarer Prototyp vorgeführt. ➤ Level 1 ist implementiert und getestet <p>Ziele für den 13.06.2023</p> <ul style="list-style-type: none"> ➤ Level 2 ist implementiert und getestet <p>Ziele für den 19.06.2023</p> <ul style="list-style-type: none"> ➤ Ausführbares Produkt beim Kunden abgegeben
Projektnutzen:	Der Auftraggeber kann mit unserem Produkt auf IT-Messen die Aufmerksamkeit der Besucher auf seinen Stand ziehen.

8. Rollenverteilung

Rollen / Verantwortlichkeiten			
Projektrolle	Rollenbeschreibung	Name	Stellvertreter
Teamleiter	<ul style="list-style-type: none"> • Team motivieren • Konflikte schlichten • Prioritäten der Arbeitspakete setzen • Moderator bei Meetings • Zielsetzung für Sprints 	Mike Menzel	Fabian Hoppe
Qualitätsbeauftragter	<ul style="list-style-type: none"> • Planung regelmäßiger Maßnahmen zur Qualitätssicherung • Erarbeitet Checklisten zur Überprüfung von Dokumenten, wobei auch Feedback von Tutoren und dem Management beachtet werden • Hat den Überblick, welche Dokumente reviewed werden müssen 	Stephan Halder	Andreas Roth
Risikomanagement	<ul style="list-style-type: none"> • Risiken mit Eintrittswahrscheinlichkeit und Risikoauswirkung ermitteln (siehe 10. Projektrisiken) • Indikatoren festlegen • präventive und reaktive Maßnahmen festlegen • Team auf Risiken aufmerksam machen 	Eren Saglam	Andreas Roth
Kundenbeauftragter	<ul style="list-style-type: none"> • Pflegt Kontakt zum Kunden • Sammlung von Fragen und Ideen (Fragenkatalog) • Nimmt an den Kundengesprächen teil 	Fabian Hoppe	Mike Menzel
Toolbeauftragter und Protokollführung	<ul style="list-style-type: none"> • Protokollführung bei Sprint Retrospektive und Sprint Planning Meeting • Pflegt Arbeitspakete in Jira ein • Sorgt für eine ordentliche Struktur im Drive-Ordner • Ist verantwortlich für die verwendeten Tools (siehe 11.0 Tools) und schaltet neue Teammitglieder für die Tools frei 	Eren Saglam	Stephan Halder

9. Qualitätsmanagement

Es wurden mehrere Maßnahmen zur Qualitätssicherung festgelegt, diese sind auch im [Qualitätssicherungshandbuch](#) zu finden, welches nach Absprache mit dem Team vom Qualitätsbeauftragten erstellt wurde, um einheitliche Standards für das Reviewen von Dokumenten, für das Schreiben für Code und Tests und für den Umgang mit *Git* festzulegen. Der Qualitätsbeauftragte ist auch dafür verantwortlich, das Qualitätssicherungshandbuch aktuell zu halten.

Dies hier stellt nur einen Auszug der wichtigsten Maßnahmen dar.

9.1 Review von Dokumenten

Jedes Dokument wird von zwei Personen unabhängig voneinander mithilfe einer Checkliste überprüft, die vom Qualitätsbeauftragten erstellt und nach Feedback vom Management sowie den Tutoren zu den jeweiligen Dokumenten angepasst werden. Beide Personen legen jeweils eine Liste mit Problemen an, die sich in diesem Dokument finden. Danach werden die Probleme auf dieser Liste mit dem Autor besprochen und eine Fehlerliste erstellt. Der Autor korrigiert diese Fehler im Anschluss. Danach wird das Dokument nochmal mit einer Checkliste überprüft. Des Weiteren werden Dokumente, die abgegeben werden müssen, auch zuvor nochmal an einen Tutor zur Durchsicht geschickt.

9.2 Testen von Code

Es werden mehrere Maßnahmen bezüglich des Testens von Codes angewandt. Folgende Testarten werden in diesem Projekt angewendet:

- Blackbox Tests
- Whitebox Tests
- Manuelle Tests

Gefundene Fehler werden dokumentiert, indem sie in Jira eingetragen werden. Dabei wird jedem Bug eine eindeutige Nummer, ein Name und eine Beschreibung des Bugs gegeben. Der Bug wird dann in eine von vier Schweregraden eingeteilt: niedrig, mittel, hoch, kritisch. Mehr Details zum Testen von Code stehen im Dokument [Konzept für Code Testing](#). In diesem Dokument steht auch, was die oben genannten Schweregrade genau bedeuten.

9.2 Reviewen von Code / Git Richtlinien

Auf dem Main-Branch des Git-Repository sollte nur ausführbarer und getesteter Code liegen. Um dies sicherzustellen, benutzen wir einen [Featurebranch-Workflow](#). Dabei müssen, bevor der neue Code auf den Main-Branch gemerged wird, zwei Personen unabhängig voneinander den Code auf Fehler überprüfen. Außerdem müssen auch alle Tests fehlerfrei abgelaufen sein. Diese Richtlinien sind auch detailliert im Dokument [Git-Richtlinien](#) zu finden.

9.3 Coding Richtlinien

Es wurden auch Richtlinien zum Schreiben von Code festgelegt. Darin steht unter anderem, wie Variablen, Funktionen und Klassen zu benennen, aber auch in welchem Stil Code geschrieben werden sollte. Die Coding Richtlinien sind im gleichnamigen Dokument auf Google Drive zu finden: [Coding Richtlinien](#)

10. Projektrisiken

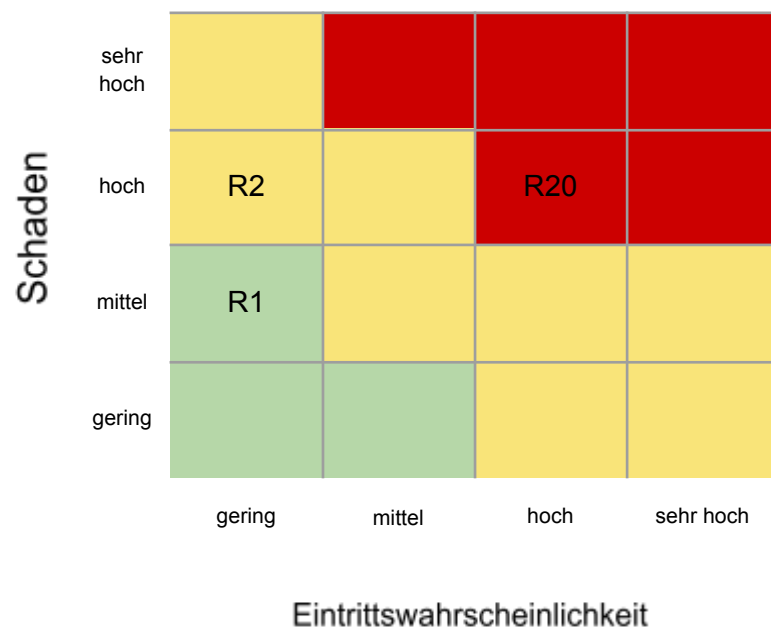
Der Risikobeauftragte sucht nach möglichen Risiken für das Projekt. Er spricht dazu mit den Teammitgliedern, ob sie ein Risiko sehen, welches relevant sein könnte. Er bespricht die gefundenen Risiken mit dem Team und ordnet die Risiken entsprechend unserer Risikomatrix (siehe [10.2 Risikomatrix](#)) ein.

10.1 Aktuelle Risiken

Dies ist ein Auszug der aktuellen Risiken des Teams. Eine vollständige Liste der Risiken ist im gleichnamigen Dokument [Risiken](#) zu finden. Die Risiko-ID ist eine fortlaufende Nummer, mit der jedes Risiko, welches im Dokument Risiken definiert wurde, eindeutig identifiziert werden kann.

Risiko ID	Risikobeschreibung	Indikator	Präventivmaßnahmen	Reaktionsmaßnahmen
R1	Mitarbeiter können nicht an die Hochschule kommen, weil öffentliche Verkehrsmittel streiken	Bahnen / Busse fahren nicht mehr	Möglichkeiten für Homeoffice haben / einrichten	Kommunikation über Discord, spontaner Umstieg auf Homeoffice
R2	Längerer Ausfall der Designer	Beide Designer fallen länger als eine Woche aus	Die Designer richtig einbinden, Stimmungsabfrage, Streitschlichtung	Kontakt der Dozenten, mehr Zeit für an die Person vergebene Arbeitspakete einplanen
R20	Unity Projekte können zu groß werden und nicht mehr auf GitHub gepusht werden, da das Repository eine Größenbeschränkung für Dateien hat	Eine Funktionalität soll auf das Repository gepusht werden, aber Github verweigert den Push, weil kein Speicher mehr zur Verfügung steht	1. Verwenden von <i>Git LFS</i> (Large File Storage), um große Dateien in Github zu verwalten und zu speichern. Link zu LFS: https://git-lfs.com/ 2. Keine Assets oder sonstige Dateien im Repository speichern, die nicht notwendig sind, um das Projekt auszuführen	1. Löschen unbenutzter Assets und großer Dateien, um die Repository Größe zu verringern.

10.2 Risikomatrix



Dies ist die Risikomatrix des Teams. Sie wird benutzt, um die Risiken übersichtlich einzuordnen. Beispielhaft sind in der Risikomatrix die aktuellen Risiken aus [10.1 Aktuelle Risiken](#) dargestellt.

Die Bezeichnungen an der Risikomatrix *gering*, *mittel*, *hoch* und *sehr hoch* haben folgende Bedeutungen:

Bezeichnung	Eintrittswahrscheinlichkeit	Schaden
gering	Vorfall tritt innerhalb des Projektes ein	Minimale Verzögerung in den nachfolgenden Prozessen
mittel	Vorfall tritt innerhalb der nächsten zwei Monate ein	Führt zu einer Verzögerung von ca. einem Tag
hoch	Vorfall tritt vor dem nächsten Review ein	Führt zu einer Verzögerung von mehr als einem Tag
sehr hoch	Vorfall tritt innerhalb des Sprints ein	Führt zu Verzögerungen von mehr als einer Woche

11. Tools

Zugänge zu allen Tools werden vom Toolbeauftragten verwaltet. Dieser kann auch neue Benutzer zu den Tools freischalten. (siehe [Rollenverteilung](#))

Folgende Tools werden im Team benutzt:

11.1 Jira

[Jira](#) wird verwendet, um die Arbeitspakete zu strukturieren, gearbeitete Zeiten an Arbeitspaketen zu erfassen und als Überblick welche Arbeitspakete abgeschlossen sind und welche noch anstehen. Außerdem werden die Arbeitszeiten mithilfe der Erweiterung "[Clockwork Free](#)" erfasst.

11.2 GitHub

[GitHub](#) wird im Team zum Hosten der [Teamwebsite](#) und als Repository zum Coden verwendet. Dazu existieren sowohl ein [Repository für die Website](#) als auch [eines zum Coden](#).

Der Master-Branch sollte immer lauffähigen Code beinhalten. Um das sicherzustellen, benutzen wir einen Feature-Branch-Workflow. (siehe [Reviewen von Code / Git Richtlinien](#))

11.3 Discord

[Discord](#) wird zur Kommunikation im Team benutzt. Dazu gibt es einen Discord Server, der für das Team erstellt wurde. Discord ist das Hauptkommunikationsmittel im Team. Es wird sowohl dazu verwendet, im Team miteinander zu kommunizieren, als auch mit der Tutorin in Kontakt zu treten. Discord wird auch zur Kommunikation benutzt, wenn Personen im Homeoffice arbeiten. Die Designer sind auch über diesen Discord Server erreichbar.

11.4 WhatsApp

[WhatsApp](#) wird ebenfalls zur Kommunikation innerhalb des Teams benutzt. Es besteht eine WhatsApp-Gruppe mit allen Teammitgliedern, in der Informationen ausgetauscht werden. Es gibt zu Discord zusätzlich eine WhatsApp-Gruppe, um dringliche Angelegenheiten zu kommunizieren. Dies machen wir getrennt von Discord, da einige Teammitglieder Discord Benachrichtigungen stumm geschaltet haben und nur während der Arbeitszeit diese Nachrichten lesen.

11.5 Google Drive

[Google Drive](#) wird verwendet, um Dokumente, die sich auf dieses Projekt beziehen, zu speichern und zu organisieren. Dabei sind die Dokumente in folgenden Ordnern strukturiert:

- Kundengespräche:
Hier befinden sich Dokumente, die im Zusammenhang mit dem Kundengespräch stehen. (z.B. Fragebögen, Protokolle zu Kundengespräche)
- Jour fixes und Review:

Hier befinden sich Dokumente, die im Zusammenhang mit den Jour fixes und den Reviews stehen. (z.B. Präsentationsfolien, Protokolle zu den Jour fixes/ Reviews)
- Projektmanagement:

In diesem Ordner befinden sich Dokumente, die in Zusammenhang mit dem Projektmanagement stehen. (z.B. Dokumente zum Risikomanagement, Definition des Prozessmodells, Projekthandbuch, Anforderungsdokument)
- Team Meetings:

In diesem Ordner befinden sich alle Dokumente, die im Zusammenhang mit Team-Meetings entstehen. (z.B. Protokolle zu Meetings, Sprint-Planungen)
- Qualitätssicherung:

In diesem Ordner sind alle Dokumente zu finden, die in Zusammenhang mit der Qualitätssicherung entstehen. (z.B. Qualitätssicherungs-Handbuch, Checklisten)
- Ideenfindung:

In diesem Ordner sind erste Ideen und Dokumente vom gemeinsamen Brainstorming zu finden.
- SAP-BTP/sovanta Innovation Factory:

In diesem Ordner sind Dokumente, die die Funktionen der SAP-BTP zusammenfassen und die Funktionsweise der sovanta Innovation Factory erklären.
- Implementierung:

In diesem Ordner befinden sich alle Dokumente, die während der Entwicklung entstehen. (z.B Architektur, Designdokumente)

11.6 Microsoft Teams

[Microsoft Teams](#) wird für die Sprechstunden mit dem Kunden verwendet. Zum Beitritt wird nur der Link zur Besprechung benötigt. Kunden Sprechstunden finden in der Regel freitags von 9:15 Uhr bis 10:15 Uhr statt.

12. Übersicht Dokumente

Übersicht der wichtigsten Dokumente im Team:

- [Risiken](#)
- [Coding Richtlinien](#)
- [Qualitätssicherungshandbuch](#)
- [Git Richtlinien](#)
- [Anforderungsdokument](#)
- [Architekturdokument](#)
- [Konzept für Code Testing](#)
- [Jour fixe Template](#)
- [Review Template](#)
- [Stimmung](#)

13. Glossar

Begriff	Beschreibung
SAP-BTP	SAP-BTP steht für "SAP Business Technology Platform" und bezieht sich auf die Cloud-Plattform von SAP, die Unternehmen dabei unterstützt, ihre Geschäftsprozesse zu optimieren und Innovationen voranzutreiben.
sovanta Innovation Factory	Die sovanta Innovation Factory beschreibt die Vorgehensweise der sovanta AG bei Projekten. Mit der Innovation Factory stellt sovanta ihr umfassendes Tool- und Serviceportfolio ihren Kunden bereit.
Fibonacci Zahlen	Fibonacci-Zahlen sind eine Folge von Zahlen, bei der jede Zahl die Summe der beiden vorherigen Zahlen ist (mit Ausnahme der ersten beiden Zahlen, die als 0 und 1 definiert sind). Die Folge beginnt normalerweise mit 0, 1, 1, 2, 3, 5, 8, 13, 21 und setzt sich endlos fort.
Discord	Discord ist ein Onlinedienst für Sprach- und Videokonferenzen und Chatnachrichten.
Google Drive	Google Drive ist ein Filehosting-Dienst von Google, der zur Speicherung, Bearbeitung und Freigabe von Dateien benutzt werden kann.
Jira	Jira ist eine Webanwendung, mit der Fehlerverwaltung, Problembehandlung und operatives Projektmanagement betrieben werden kann.
Microsoft Teams	Microsoft Teams ist eine von Microsoft erstellte Software für Besprechungen und Chats.
Git	Git ist ein verteiltes Open-Source-Versionskontrollsystem, das Entwicklern dabei hilft, Änderungen in einem Code-Repository zu verfolgen, zu organisieren, zu verwalten und mit anderen Teammitgliedern zusammenzuarbeiten.
Github	Github ist ein Dienst zur Versionsverwaltung von Software- und Entwicklungsprojekten.

Begriff	Beschreibung
Feature-Branch-Workflow	Der Feature-Branch-Workflow ist eine Methode für die Versionskontrolle von Code, bei der jeder neue Code-Beitrag in einer separaten Branch (Feature-Branch) entwickelt und getestet wird, bevor er in die Haupt-Branch (Master-Branch) integriert wird. Dadurch können Entwickler unabhängig voneinander arbeiten und Änderungen ohne Konflikte in den Code einbringen. Der Feature-Branch-Workflow wird häufig in der Software-Entwicklung eingesetzt, um die Qualität und die Stabilität des Codes zu verbessern.
Git LFS	Git LFS (Large File Storage) ist eine Erweiterung für das Git-Versionierungssystem, die entwickelt wurde, um den Umgang mit großen Dateien effizienter zu gestalten. Anstatt große Dateien direkt in das Git-Repository einzufügen, verwendet Git LFS eine Verweisdatei, die die großen Dateien außerhalb des Repositorys speichert