



OctoPi

Projekthandbuch
Softwareentwicklungsprojekt SS 2023

Version: 1.0 (25.04.2023)

Auftraggeber: sovanta AG

Verantwortliche:

Thomas Martin

Julian Wernz

Inhaltsverzeichnis

1. Versionsverzeichnis.....	4
2. Versionierung.....	5
3. Einleitung.....	6
4. Projektbeschreibung.....	7
4.1 Projekthintergrund.....	7
4.2 Projektziel.....	7
5. Stakeholder.....	8
5.1 Team.....	8
5.1.1 Entwickler.....	8
5.1.2 Designer.....	8
5.2 Auftraggeber.....	8
5.3 Betreuung.....	9
6. Projektmanagement.....	10
6.1 Organisation.....	10
6.1.1 Arbeitszeiten.....	10
6.1.2 Vorgehensmodell.....	10
6.2 Meetings.....	10
6.1.2 Sprint-Planning.....	10
6.1.3 Sprint-Retrospektive.....	11
6.1.4 Sprint-Daily.....	11
6.1.5 Kundensprechstunde.....	11
6.2 Projektplanung.....	12
6.2.1 Arbeitspakete.....	12
6.3 Tools.....	13
6.3.1 Jira.....	13
6.3.2 Dokumentenspeicherung (Google Docs).....	14
6.3.3 VS Code.....	14
6.3.4 GitHub.....	14
6.3.5 Discord.....	14
6.3.6 WhatsApp.....	14
6.4 Rollendefinition.....	15
6.4.1 Projektleiter/Product Owner.....	15
6.4.2 Kundenkontakt.....	15
6.4.3 Dokumentation.....	15
6.4.4 Qualitätssicherung.....	16
6.4.6 Entwicklung und Design.....	16
6.5 Risikomanagement.....	17
7. Dokumente / Unterlagen.....	19
7.1 Webseite.....	19



7.2 Protokolle.....	19
7.3 Vorträge.....	19
8. Glossar.....	21

1. Versionsverzeichnis

Dieses Dokument entsteht in einem iterativen Prozess und wird insbesondere um neue Anforderungen erweitert, sobald sich Auftraggeber und Auftragnehmer auf eine konkrete Umsetzung der Aufgabe geeinigt haben.

Version	Änderung	Datum	Autor(en)
1.0	Überarbeitung für Abgabe 1	25.04.2023	Philip Dell, Thomas Martin, Julian Wernz,
0.4	Projektmanagement überarbeitet	24.04.2023	Thomas Martin, Julian Wernz
0.3	Projektmanagement (Tools, Rollenverteilung) hinzugefügt	23.04.2023	Thomas Martin
0.2	Projektmanagement (Organisation, Meetings, Projektplanung) hinzugefügt	21.04.2023	Thomas Martin
0.1	Projektbeschreibung und Stakeholder hinzugefügt	20.04.2023	Thomas Martin
0.0	Inhaltsverzeichnis und Struktur des Dokuments definiert	19.04.2023	Jasmin Tschernoch

2. Versionierung

Für die Versionierung dieses Dokuments gelten folgende Regeln, welche teamintern beschlossen wurden:

1. Die Versionsnummer besteht aus zwei Zahlen, getrennt durch einen Punkt.
2. Die erste Zahl der Versionsnummer wird ganzzahlig erhöht, sobald das Dokument für eine Abgabe bereit ist. Die Zahl nach dem Punkt wird zusätzlich auf 0 gesetzt.
3. Die Erhöhung der zweiten Zahl der Versionsnummer erfolgt, sobald der Inhalt um mindestens ein Kapitel erweitert oder der Inhalt bestehender Abschnitte geändert wurde.
4. Die initiale Version eines Dokuments ist 0.0 .

3. Einleitung

Das vorliegende Projekthandbuch entsteht im Rahmen des Softwareentwicklungsprojektes im Sommersemester 2023 an der Hochschule Mannheim und dient allen Beteiligten des Projektes als Leitfaden. Dieses Dokument enthält alle projektrelevanten Informationen von der Teambeschreibung, Team, Vorgehensweise, bis hin zur Teamstruktur.

Dieses Dokument wird über den Projektverlauf, bei Veränderung der Teamstruktur oder Teamabläufen, kontinuierlich überarbeitet und gegebenenfalls erweitert.

4. Projektbeschreibung

4.1 Projekthintergrund

Das *SEP* ist ein Modul im 4. Semester im Studiengang IB, in dem kleinere Gruppen (fünf bis acht Studierende) an der Realisierung eines Produktes, deren Anforderungen von einem externen Kunden gestellt werden, arbeiten. Als Teil des SEPs im Sommersemester 2023 wurde die Firma „sovanta AG“ als externer Kunde gewählt (im Folgenden als sovanta adressiert). Sovanta ist im Bereich der Softwareentwicklung und *UX* tätig. Ihre Spezialgebiete umfassen SAP User Experience, Data Science und die *SAP BTP*.

Viele Kunden von sovanta nutzen bereits SAP Software für eine Vielzahl ihrer verschiedenen Geschäftsbereiche. Dennoch wird die SAP BTP, wegen fehlender Aufklärung über die SAP BTP, laut sovanta, nicht ausreichend von ihren Kunden genutzt.

4.2 Projektziel

Ein überzeugendes Konzept und eine prototypische Applikation, um Besuchern von Messen spielerisch die Möglichkeiten der SAP BTP vorzustellen und potenzielle Kunden von der 'sovanta Innovation Factory' überzeugen zu können. Die Software soll die Besucher auf den Stand von sovanta aufmerksam machen. Weiterhin besteht das Ziel, das Interesse an dem Unternehmen und deren Dienstleistungen, insbesondere der *Innovation Factory* for SAP BTP, zu wecken.

5. Stakeholder

5.1 Team

Das Team besteht aus acht Mitgliedern, davon studieren zwei Kommunikationsdesigns und die restlichen sechs Informatik.

Das Team ist über die Team-Mail erreichbar: octopi-sep@proton.me

5.1.1 Entwickler

Name	Studiengang	Kontakt
Philip Dell	Informatik	2122604@stud.hs-mannheim.de
Thomas Martin	Informatik	2121321@stud.hs-mannheim.de
Steven Schmitt	Informatik	2025448@stud.hs-mannheim.de
Jasmin Tschernoch	Informatik	2120914@stud.hs-mannheim.de
Timo Wenz	Informatik	2025014@stud.hs-mannheim.de
Julian Wernz	Informatik	2123602@stud.hs-mannheim.de

5.1.2 Designer

Name	Studiengang	Kontakt
Johanna Neuendorf	Kommunikationsdesign	2020427@stud.hs-mannheim.de
Julia Stumpe	Kommunikationsdesign	2020434@stud.hs-mannheim.de

5.2 Auftraggeber

Der Kontakt zu den sovanta-Mitarbeitern wird über einen E-Mail-Verteiler geregelt. Die durchschnittliche Rückmeldezeit beträgt zwei bis drei Tage.

Name, Vorname	Arbeitsbereich	Kontakt
Frankenbach, Jakob	Head of Development	semesterprojekt2023@sovanta.com
Haas, Larissa	Senior Data Scientist	semesterprojekt2023@sovanta.com
Meiseberg, Alina	Senior Data Scientist	semesterprojekt2023@sovanta.com
Janßen, Nils	Senior Software Engineer	semesterprojekt2023@sovanta.com
Bechberger, Thomas	Senior User Experience Designer	semesterprojekt2023@sovanta.com
Rueda, Rueda	Marketing Specialist	semesterprojekt2023@sovanta.com
Hebestreit, Louise	Marketing Specialist	semesterprojekt2023@sovanta.com

5.3 Betreuung

Name, Vorname	Rolle	Kontakt
Prof. Dr. Schramm, Wolfgang	Professor	w.schramm@hs-mannheim.de
Prof. Dr. Knauber, Peter	Professor	p.knauber@hs-mannheim.de
Marco, D'Angelo	Lehrbeauftragter (TEW)	m.dangelo@lba.hs-mannheim.de
Haas, Daniel	Tutor	daniel.haas@stud.hs-mannheim.de

6. Projektmanagement

6.1 Organisation

6.1.1 Arbeitszeiten

Die Informatiker arbeiten werktags von 09:30 Uhr bis 17:30. Um 12:00 Uhr machen sie eine gemeinsame Mittagspause, die eine dreiviertel Stunde dauert. Nachmittags ist jeder frei, eine fünfzehnminütige Pause zu machen.

Die Designer arbeiten in der Woche jeweils fünf Stunden. Diese verrichten sie donnerstags und / oder freitags.

6.1.2 Vorgehensmodell

Als Vorgehensmodell nutzen wir abgewandeltes Scrum. Zum einen gibt es bei uns keinen Scrum-Master. Diese Rolle ist in unserer Situation überflüssig, da bei unseren Meetings keine Moderatorenrolle gebraucht wird und der Aufwand für die Priorisierung des Backlogs, durch wöchentliche Abgaben, reduziert ist. Zum anderen sind die Aufgaben des Product Owners (Produktfunktionen definieren, User Stories erstellen, ...) an das ganze Team verteilt. Die Hauptverantwortung des Projektstatus und des Backlogs liegen aber immer noch mit dem Product Owner. Da die Sprint-Länge nur eine Woche beträgt, halten wir kein Sprint-Review, sondern fassen während des Sprint-Retro den Sprint-Inkrement zusammen.

Die Länge des Sprints wurde auf eine Woche festgelegt, damit zu jedem Jour fixe/Review, die jede Woche mittwochs zu halten sind, ein Sprint in seiner Gesamtheit präsentiert werden kann. Daher sind Sprint Anfang und Ende für 14 Uhr dienstags angesetzt.

6.2 Meetings

6.1.2 Sprint-Planning

Das Sprint-Planning-Meeting ist für 14 Uhr dienstags angesetzt. In diesem Meeting wird das Ziel für den anliegenden Sprint definiert, was durch wöchentliche Abgaben erleichtert wird. Das Team teilt die Aufgaben in Arbeitspakete auf und es wird zusätzlich der gebrauchte Arbeitsaufwand geschätzt. Falls beim Schätzen, ein großer Bei größeren Diskrepanzen beim

Schätzen wird im Team diskutiert, bis sich ein Kompromiss bildet. Arbeitspakete, die im letzten Sprint nicht vervollständigt wurden, bekommen eine neue Schätzung. Dieses Meeting wird vom Product Owner geleitet.

Das Meeting dauert ca. eine Stunde.

6.1.3 Sprint-Retrospektive

12:30 Uhr beginnt dienstags das Sprint-Retro-Meeting. Es werden alle teaminterne Probleme, die im letzten Sprint aufgetreten sind, aufgelistet und besprochen. Teammitglieder stellen Lösungsansätze vor, mit denen die Kommunikation in Zukunft verbessert werden kann.

Die Retro dauert ca. 30 Minuten, kann aber auch, wenn mehr Zeit benötigt wird, in der Mittagspause weitergeführt werden, die Teammitglieder essen zusammen in der Hochschul Mensa.

6.1.4 Sprint-Daily

Im Sprint-Daily-Meeting stellt jedes Teammitglied kurz vor, was derjenige am Vortag geleistet hat. Es wird geklärt, ob sich jeder der Arbeitsteilung für den jetzigen Tag bewusst ist und ob jemand Probleme bei einer Aufgabe hat.

Das Daily-Meeting ist maximal 15 Minuten lang.

6.1.5 Kundensprechstunde

Es wird einmal in der Woche eine Kundensprechstunde von der sovanta angeboten, in der Teams ihre Ideen vorstellen können und Fragen stellen. An der Sprechstunde beteiligen sich diejenigen, die für den Kundenkontakt verantwortlich sind. Danach stellen diese die Ergebnisse der Sprechstunde dem Rest des Teams vor und es wird besprochen, wie wir das Kundenfeedback in unser Projekt einbauen können.

Die Sprechstunde beginnt um 9.15 Uhr freitags und dauert meist 1 Stunde. Das Meeting danach dauert gewöhnlich 1 Stunde.

6.2 Projektplanung

6.2.1 Arbeitspakete

Werden im laufenden Sprint und beim Sprint Planning Meeting angelegt.

Jedes Arbeitspaket wird mit den folgenden Inhalten beschrieben:

- Inhalt - was soll alles im Arbeitspaket geschehen
- Begründung - aus welchen Gründen ist das Arbeitspaket wichtig und warum gehört dieser Inhalt dazu
- Nach Fertigstellung dieser Änderung - was sind die Outputs nachdem das Arbeitspaket beendet wurde
- Akzeptanzkriterium - wann/woran kann festgemacht werden, ob das Arbeitspaket beendet ist
- Was nicht enthalten ist - was ist explizit nicht in diesem Arbeitspaket enthalten
- Getroffene Entscheidungen - wurde beim Bearbeiten eine besondere oder wichtige Entscheidung getroffen?

Außerdem ist jedes Arbeitspaket einem 'Meilenstein', in dem mehrere Arbeitspakete gruppiert werden, untergeordnet. Dies kann sich auf eine Aktion beziehen, die in jedem Sprint ähnlich ausgeführt wird(z.B. Organisation) , oder einen Produktzustand, der nur einmal während des Projektverlaufs bearbeitet wird(z.B. 'Vertikaler Prototyp').

6.3 Tools

6.3.1 Jira

Wir nutzen *Jira*, um unsere Arbeitspakete zu verteilen und unseren Fortschritt zu verfolgen. Bei den Sprint-Planning-Meetings entscheiden wir, welche Pakete in den aktuellen Sprint kommen. Auf Jira haben wir zu jedem Paket eine weitere Beschreibung, was alles ein Paket beinhaltet und eine *DOD* (Definition of Done). Außerdem können wir mithilfe von Jira zu jedem Paket den Arbeitsaufwand schätzen und sogar zu jedem Paket eine Person als Verantwortlichen zuweisen. Jedes Paket im aktuellen Sprint hat einen von drei Zuständen:

- 'ZU ERLEDIGEN' : Das Arbeitspaket wurde noch nicht angefangen, sollte aber bis zum Sprintende bearbeitet sein. Beim Sprint Anfang sind alle Pakete in diesem Zustand, auch wenn ein Paket im vorherigen Sprint nicht fertiggestellt wurde. Solche Pakete werden bei Sprint Anfang überarbeitet und bekommen eine neue Schätzung.
- 'IN ARBEIT' : Ein Arbeitspaket, an dem jemand schon gearbeitet hat, aber noch nicht fertiggestellt werden konnte, ist in diesem Status. Dieses Paket muss auch eine gewisse Arbeitszeit protokolliert haben.
- 'FERTIG' : Ein Arbeitspaket, welches ihrer DOD übereinstimmt. Es ist egal, ob die protokollierte Zeit für solche Pakete über oder unter der geschätzten Zeit liegt. Wenn ein Arbeitspaket sich als überflüssig erweist, darf es auch mit null protokollierter Zeit auf 'FERTIG' gestellt werden.

Um die Zeit für jedes Paket zu überwachen, benutzen wir *Clockify*. Clockify ist eine Zeit Tracking Software, mit der wir unsere Zeiten zu den jeweiligen Arbeitspaketen messen können. Da 'Clockify' sich in Jira integrieren lässt, können wir direkt in Jira unsere Zeiten eintragen und auch so einfach erkennen, ob ein Arbeitspaket schneller fertig wird als gedacht oder sich im Verzug befindet. Die Zeit lässt sich entweder live messen oder manuell eintragen.

Ein weiteres Add-On, welches wir für die Zeiterfassung nutzen, ist '*Epic Sum Up*'. Mit 'Epic Sum Up' lassen sich eingetragene Zeiten summieren. Dadurch lassen sich protokollierte Zeiten einfacher kontrollieren.

6.3.2 Dokumentenspeicherung (Google Docs)

Google Docs wird genutzt, um unsere verschiedenen Dokumente untereinander auszutauschen. *Google Docs* ist unser Speicherort für Präsentationsfolien, Feedback, Pflichtenheft, Projekthandbuch, Protokolle, Risikomanagement und alle Dokumente, bezogen auf den Kundenkontakt.

Außerdem messen wir unsere Motivation täglich mithilfe eines *Google Forms*, welches die Ergebnisse in einer Excel Tabelle speichert.

6.3.3 VS Code

Dass *VS Code* die gemeinsame Entwicklerumgebung von 'OctoPi' ist, ist noch nicht final festgelegt und wird im späteren Projektverlauf bestimmt. *VS Code* wird hier aufgelistet, da der vertikale Prototyp bisher innerhalb *VS Code* entwickelt wurde.

6.3.4 GitHub

Für die Versionskontrolle nutzen wir *GitHub*. Wir haben eine Organisation 'OctoPi-Organization', auf der wir zwei Repositories liegen haben. Das eine Repository ist für das Hosten der Website. Das andere ist für das Hauptprojekt.

Link: <https://github.com/OctoPi-Team>

6.3.5 Discord

Wir nutzen *Discord* als Haupt Kommunikationsplattform für die Kommunikation innerhalb des Teams, inklusive des Tutors. Auf unserem eigenen Server, sprechen wir uns über Teamaktivitäten ab, die Designer platzieren ihre Ressourcen in ihrem eigenen Kanal und wenn zu Hause gearbeitet wird, trifft sich das Team auf dem *Discord* Server.

Weiterhin wird von einem Bot, täglich eine Erinnerung, für das Motivations-Abstimmen geschickt.

6.3.6 WhatsApp

WhatsApp wird als Kommunikationsplattform für kleinere organisatorische Absprachen genutzt. Diese betreffen z.B. das Zuspätkommen und projektexterne Treffen.

6.4 Rollendefinition

Innerhalb des Teams gibt es eine Rollenverteilung. Jedes Mitglied ist einer Rolle zugewiesen, wobei auch auf Backups Wert gelegt wird. Die Rollen werden alle 2 Wochen neu aufgeteilt.

6.4.1 Projektleiter/Product Owner

Der Projektleiter ist verantwortlich für den generellen Projektstatus, Aufgaben zu delegieren und Arbeitspakete zu definieren. Gerät das Projekt in Verzug, ist es die Aufgabe des Projektleiters, Pakete neu zu priorisieren und Maßnahmen zu ergreifen, damit das Projekt sich wieder in den Projektplan befindet. Dazu kommt noch, dass der Projektleiter Verantwortung für den Sprint Backlog ist, sodass die Arbeitspakete auf Jira zu jeder Zeit verständlich sind und die protokollierten Arbeitszeiten schlüssig sind.

Hauptverantwortlicher: Thomas Martin

Backup: Philip Dell

6.4.2 Kundenkontakt

Damit während dem Projektverlauf, zwischen den Vorstellungen des Kunden und dem Projektziel entsteht, ist es die Aufgabe des Kundenkontaktes, den Kunden immer über den Projektstatus auf dem Laufenden zu halten. Der Kundenkontakt bereitet die Fragen für die Kundensprechstunden vor und vertritt das Team dort freitags. Bei Teamdiskussion vertritt der Kundenkontakt die Interessen des Kunden. Der E-Mail-Kontakt wird auch vom Kundenkontakt gepflegt.

Hauptverantwortlicher: Jasmin Tschernoch

Backup: Timo Wenz

6.4.3 Dokumentation

Teammitglieder mit der Dokumentationsrolle sind verantwortlich für die kontinuierliche Erstellung von Dokumenten. Zu externen (Jour fixe/Review, Feedback von Tutoren/Professoren, Kunden, Sprechstunde) als auch zu Team-internen Meetings (Daily-Sprint-Meeting, Sprint-Planung, ...). Dafür wird täglich ein Protokoll geführt. Bei Veränderung von älteren wichtigen Dokumenten, ist es auch die Aufgabe des Dokumentationszuständigen, diese aktuell zu halten.

Hauptverantwortlicher: Philip Dell

Backup: Steven Schmitt

6.4.4 Qualitätssicherung

Der Qualitätsbeauftragte trägt die Verantwortung, dass alle Abgaben an Kunden oder Professoren den Anforderungen entsprechen. Dazu gehört z.B. dass Folien zur Abgabe für das Jour fixe/Review keine Rechtschreibfehler besitzen, Projekthandbuch und Pflichtenheft inhaltlich vollständig sind oder dass der Prototyp für den Kunden lauffähig ist. Weitergehend ist es die Aufgabe des Qualitätsbeauftragten, Abläufe innerhalb des Teams zu optimieren.

Hauptverantwortlicher: Steven Schmitt

Backup: Thomas Martin

6.4.6 Entwicklung und Design

Da der Projektverlauf es noch nicht verlangt, sind Entwickler-Rollen, wie z.B. Tester; Software-Designer, noch nicht verteilt. Für Aufgaben der technischen Seite, wie z.B. Software-Architektur oder Programmierung, tragen die Designer keine Verantwortung. Sie tragen die Verantwortung, dass die Benutzeroberflächen des Endprodukts den 'User Experience Heuristiken' der sovanta entsprechen.

Verantwortung: Philip Dell, Jasmin Tschernoch, Julian Wernz, Timo Wenz

6.5 Risikomanagement

Wir betreiben Risikomanagement mithilfe eines Google Spreadsheets, in dem alle bisher gefundenen Risiken aufgelistet sind. Zu jedem Risiko stehen noch eine Beschreibung, die präventiven Maßnahmen, ein messbarer Indikator, als auch die präventiven Maßnahmen, für den Fall, dass das jeweilige Risiko eingetreten ist. Außerdem besitzt jedes Risiko noch eine Risikobewertung, die sich aus der Auswirkung auf das Projekt und der Eintrittswahrscheinlichkeit ergibt.

		Auswirkung			
		Unbe- deutend	Gering	Kritisch	Existen- ziell
Eintrittswahrscheinlichkeit	Unwahr- scheinlich				
	Selten				
	Möglich				
	Sehr wahr- scheinlich				

Risikomatrix

Eintrittswahrscheinlichkeit:

- Unwahrscheinlich: Es ist unwahrscheinlich, dass das Risiko eintritt.
- Selten: Das Risiko tritt selten auf.
- Möglich: Es besteht eine realistische Möglichkeit, dass das Risiko eintritt.
- Sehr wahrscheinlich: Es ist sehr wahrscheinlich, dass das Risiko eintritt.

Auswirkung:

- Unbedeutend: Die Auswirkungen des Risikos sind gering oder kaum wahrnehmbar.
- Gering: Die Auswirkungen des Risikos sind spürbar, aber nicht schwerwiegend.
- Kritisch: Die Auswirkungen des Risikos sind schwerwiegend und haben Auswirkungen auf den Projektverlauf.
- Existenziell: Das Risiko hat katastrophale Auswirkungen auf die Qualität des Endproduktes.

Ein Risiko ist am gefährlichsten, wenn es sehr wahrscheinlich eintreten könnte und eine existenzielle Bedrohung für das Projekt darstellt.

7. Dokumente / Unterlagen

7.1 Webseite

Die Octopi-Website ist eine grundlegende Übersicht für die Professoren und Tutoren, um vorab Informationen über das Team zu erfahren. Sie ist in Github-Pages gehostet und mit den Sprachen *Javascript*, *CSS* und *HTML 5* erstellt worden.

Die Website besteht aus vier verschiedenen Seiten.

Auf der Homepage findet man das Gruppenbild und die Namen der Mitglieder.

Die Seite „Über uns“ sagt aus, für welche Firma das Team arbeitet und wann die Arbeitszeiten sind.

Unter „Kontakt“ findet man die Kontaktdaten sowie eine direkte Verknüpfung, um direkt eine Mail an das Team zu senden.

Auf der Dokumentenseite finden Sie Downloads für unsere Jour fixe/Review Folien und Steckbriefe. So können sich Besucher der Website einen besseren Eindruck von der Arbeit und den Fähigkeiten des Teams verschaffen.

Link zu der Seite: <https://octopi-team.github.io/OctoPi-Website/index.html>

7.2 Protokolle

Das Team führt täglich Protokolle, welche die relevanten Ereignisse des Tages enthalten. Jedes Protokoll enthält das Datum, den Namen der Person, welche das Protokoll erstellt hat, sowie Informationen darüber, ob die Arbeit online oder vor Ort durchgeführt wurde. Darüber hinaus enthält das Protokoll eine Agenda, welche alle besprochenen Themen und geplante Aktivitäten enthält.

7.3 Vorträge

Es gibt zwei verschiedene Arten von Vorträgen, die regelmäßig gehalten werden. Der am häufigsten gehaltene Vortrag ist das "Jour fixe", der einmal pro Woche (Mittwoch) stattfindet. Dieser Vortrag dient als Statusbericht und informiert Professoren und Tutoren über den aktuellen Stand des Projekts. Dieser sollte nicht länger als 5 Minuten dauern.

Der zweite Vortrag ist das "Review", in dem alle vorherigen Jour fixe, seit dem letzten Review zusammengefasst werden sollten. Hierbei wird auch die Planung bis zum nächsten Review sowie der Prototyp präsentiert.

8. Glossar

Begriff	Definition
Backend	Ein Server, der für die Maschineninteraktion gedacht ist. Es findet keine direkte Interaktion vom Benutzer zum Backend statt.
CDS	SAP Core Data Services https://cap.cloud.sap/docs/cds/
Clockify	eine kostenlose App zum Zeiterfassen https://clockify.me/
CSS	Cascading Stylesheet
Discord	Eine Plattform, die zur Kommunikation genutzt wird. https://discord.com/
DOD	Definition of Done
Epic Sum Up	Epic Sum Up ist ein Add-On für Jira, mit der man eingetragenen Zeiten einfach summieren kann
Express.js	Backend Web Server Framework basierend auf Node.js. https://expressjs.com/
FA	Funktionale Anforderungen
Frontend	Teil der Software, welcher mit dem Nutzer interagiert und auf dem Rechner des Nutzers läuft.
Github	Softwareversionierungsservice, welcher Git benutzt und weitere Zusatzfunktionen bietet. https://github.com/

Google Docs	Online-Service, welcher Mehrbenutzerbetrieb für Word-Dokumente, Sheets und Slides anbietet. https://docs.google.com
HTML5	Strukturbasierte Programmiersprache basierend auf XML, welche für Webseiten benutzt wird.
Human Centered Design	reale Personen sind in der Mitte der Entwicklung
Innovation Factory	Die Innovation Factory besteht aus sechs Bereichen und wird mit den vorhandenen Features der BTP realisiert. Die sechs Bereiche lauten Design, Engineering, Produktion, Parts, Shipment und Monitoring. (https://sovanta.com/innovation-factory-for-sap-btp/)
JavaScript	Eine Programmiersprache, die für den Browser entwickelt wurde und mittlerweile auch mithilfe von Node.js im Backend laufen kann.
Jira	Online Projektmanagement Tool, welches primär für Softwareentwicklung eingesetzt wird. https://www.atlassian.com/software/jira
NFA	Nichtfunktionale Anforderungen
Node.js	Eine plattformübergreifende Javascript Laufzeitumgebung. https://nodejs.org
React.js	Frontend Framework basierend auf Node.js Javascript. https://react.dev/
R	Randbedingung
SAP BTP	SAP Business Technology Platform: „Eine Technologieplattform, die Daten und Analysen, künstliche Intelligenz, Anwendungsentwicklung, Automatisierung und Integration in einer einheitlichen Umgebung vereint.“ (https://www.sap.com/germany/products/technology-platform/what-is-sap-business-technology-platform.html)

SEP	Softwareentwicklungsprojekt
Three.js	JavaScript Frontend Library für 3D Rendering. https://threejs.org/
UX	Nutzer Erfahrung - Die Erfahrung, die ein Nutzer mit einem Produkt macht.
VS Code	Visual Studio Code ist ein kostenloser Quelltext-Editor. https://code.visualstudio.com/
Web-API	Application Programming Interface, welches Interaktionen über http requests bietet.
WhatsApp	Internet Nachrichten App für Mobilgeräte. https://www.whatsapp.com/