

Ostbayerische Technische Hochschule Amberg-Weiden
Fakultät Elektrotechnik, Medien und Informatik

Studiengang Elektro- und Informationstechnik

Bachelorarbeit

von

Berkay Yurdagül

**evoSTRIVE – Ein digitaler Ansatz zur
Wertstrommodellierung: Integration von
Industrie-Standards und Domänenmodellen in einem
Web-Editor**

evoSTRIVE – A Digital Approach to Value Stream
Modeling: Integration of Industry Standards and Domain
Models in a Web Editor

Ostbayerische Technische Hochschule Amberg-Weiden
Fakultät Elektrotechnik, Medien und Informatik

Studiengang Elektro- und Informationstechnik

Bachelorarbeit

von

Berkay Yurdagül

**evoSTRIVE – Ein digitaler Ansatz zur
Wertstrommodellierung: Integration von
Industrie-Standards und Domänenmodellen in einem
Web-Editor**

evoSTRIVE – A Digital Approach to Value Stream
Modeling: Integration of Industry Standards and Domain
Models in a Web Editor

Bearbeitungszeitraum: von 15. Oktober 2023
bis 17. Januar 2024

1. Prüfer: Prof. Dr.-Ing. Christoph P. Neumann

2. Prüfer: Prof. Dr. Dieter Meiller

Selbstständigkeitserklärung

Name und Vorname
der Studentin/des Studenten: **Yurdagül, Berkay**

Studiengang: **Elektro- und Informationstechnik**

Ich bestätige, dass ich die Bachelorarbeit mit dem Titel:

**evoSTRIVE – Ein digitaler Ansatz zur Wertstrommodellierung: Integration von
Industrie-Standards und Domänenmodellen in einem Web-Editor**

selbstständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine
anderen als die angegebenen Quellen oder Hilfsmittel benutzt sowie wörtliche und
sinngemäße Zitate als solche gekennzeichnet habe.

Datum: 15. Januar 2024

Unterschrift: *B. Yurdagül*

Bachelorarbeit Zusammenfassung

| | |
|---|-------------------------------------|
| Studentin/Student (Name, Vorname): | Yurdagül, Berkay |
| Studiengang: | Elektro- und Informationstechnik |
| Aufgabensteller, Professor: | Prof. Dr.-Ing. Christoph P. Neumann |
| Durchgeführt in (Firma/Behörde/Hochschule): | evosoft GmbH |
| Betreuer in Firma/Behörde: | Adrian Strauss |
| Ausgabedatum: 15. Oktober 2023 | Abgabedatum: 17. Januar 2024 |

Titel:

evoSTRIVE – Ein digitaler Ansatz zur Wertstrommodellierung: Integration von Industrie-Standards und Domänenmodellen in einem Web-Editor

Zusammenfassung:

In dieser Bachelorarbeit soll ein Value Stream Assistant (VSA) - Editor entwickelt werden, der in der Cloud für alle Endgeräte verfügbar ist, um an einem Wertstrom arbeiten zu können. Ein Value Stream Assistant ist ein Werkzeug im Value Stream Management. Das Ziel ist den Prozess- und Informationsfluss in Unternehmen zu optimieren. Der Editor soll den Benutzern die Möglichkeit bieten, Wertströme schnell aufzunehmen und anzupassen, sowie Key Performance Indicators (KPIs) und Grafiken anzubinden. Die Entwicklung der Applikation erfolgt mit React und der Bibliothek ReactFlow. Als Oberflächen-Komponenten werden Material UI (MUI) und hauseigene Webkomponenten eingesetzt. Es wird eine Application Programming Interface (API) definiert, um Zustände und Einstellungen im Backend zu speichern. Das Backend wird von einem anderen hauseigenen Team umgesetzt. Das Deployment des gesamten Systems wird auf Amazon Web Services (AWS) stattfinden. Das Hauptziel ist eine effiziente und benutzerfreundliche Plattform zur Wertstromanalyse und -anpassung.

Schlüsselwörter: VSA - Editor, Wertstrom Editor, Wertstromanalysetool

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Einführung und Motivation | 1 |
| 2 | Wertstromanalyse und Stand der Forschung | 4 |
| 2.1 | Vorgehensweise zur Erstellung einer Wertstromanalyse | 5 |
| 2.2 | Ziel der Wertstromanalyse | 5 |
| 2.3 | Stand der Forschung | 5 |
| 3 | Nutzerforschung | 6 |
| 3.1 | Kundeninterviews | 6 |
| 3.2 | Zielgruppen und Personas | 7 |
| 3.3 | Anforderungen aus Benutzersicht | 11 |
| 3.3.1 | Benutzerszenarien | 11 |
| 3.3.2 | User Stories | 12 |
| 4 | Architektur | 18 |
| 4.1 | RESTful-API | 18 |
| 4.1.1 | Schnittstellen | 19 |
| 4.2 | Branching Strategie | 20 |
| 4.2.1 | Bedeutung der Branching Strategie | 20 |
| 4.2.2 | Umgesetzte Branching Strategy | 20 |
| 5 | Auswahl der Technologien | 24 |
| 5.1 | Anforderung an die Auswahl der Technologie | 24 |
| 5.2 | React, ReactFlow und Typescript | 25 |
| 5.3 | State Management | 29 |
| 5.4 | Qualitätsabsicherung | 30 |
| 5.4.1 | Ziele von Tests | 30 |
| 5.4.2 | Testabdeckung | 31 |
| 6 | Technische Umsetzung | 32 |
| 6.1 | Projektstruktur | 32 |
| 6.2 | UX/UI | 33 |
| 6.2.1 | UI-Komponenten | 33 |
| 6.2.2 | Theming | 34 |

| | | |
|----------|---|-----------|
| 6.3 | Funktionalitäten und Features | 34 |
| 6.3.1 | Basisfunktionalitäten des Editors | 34 |
| 6.3.2 | Erstellung von Prozesskarten | 35 |
| 6.3.3 | Verbindung von Elementen | 43 |
| 6.3.4 | Erstellung von Notizen | 44 |
| 6.3.5 | Konfiguration von Prozesskarten | 49 |
| 6.4 | Pipeline Konfiguration | 53 |
| 6.4.1 | Hintergrund der Pipeline | 54 |
| 6.4.2 | Pipeline-Konfigurationsdatei | 54 |
| 7 | Zusammenfassung und Ausblick | 57 |
| | Abkürzungsverzeichnis | 59 |
| | Abbildungsverzeichnis | 60 |
| | Tabellenverzeichnis | 63 |
| | Literaturverzeichnis | 64 |
| A | Abbildungen | 67 |

Kapitel 1

Einführung und Motivation

In den letzten Jahren hat sich die industrielle Fertigung rasant entwickelt [Statista, 2023]. Dabei sind die Themen Effizienz und Ressourcenoptimierung zu zentralen Anliegen geworden. Deshalb hat die Wertstromanalyse (WSA) an Bedeutung gewonnen, welche ursprünglich aus dem Toyota Produktionssystem stammt und darauf abzielt, Verschwendung in Produktionsprozessen zu identifizieren und zu eliminieren. [Ohno, 1988, S. 3 ff.]

Laut einer Studie von Schoeman, Oberholster und Somerset wurde im ersten Jahr nach der Einführung von der WSA in einem Eisen- und Stahlwerk in Südafrika, eine Abfallreduktion von 28% und die Kostensenkung für die Abfallentsorgung von 45% festgestellt. Dies zeigt nicht nur die finanziellen Vorteile auf, sondern auch die allgemeinen kritischen Auswirkungen einer Wertstromanalyse. [Schoeman et al., 2021]

Die Wertstromanalyse ist ein systematischer Ansatz, um den Informations- und Materialfluss von der Rohstoffgewinnung bis zum Endkunden zu visualisieren und zu analysieren. Dabei werden nicht nur die Hauptprozesse, sondern auch die Zwischenprozesse und -lager berücksichtigt. [Rother and Shook, 2003, S. 5 ff.]

Durch die Betrachtung der gesamten Unternehmensprozesse können Engpässe, Überproduktionen oder unnötige Transporte identifiziert und Verbesserungsmaßnahmen gezielt eingeleitet werden. Da die Industrie mit steigendem Wettbewerbsdruck [Grömling, 2023], bestehender Ressourcenknappheit und wachsenden Umweltauflagen konfrontiert ist, bietet die Wertstromanalyse eine effektive Methode, um Produktionsprozesse zu optimieren und somit nachhaltige Wettbewerbsvorteile zu erzielen. Durch ihre Vorteile ist die Wertstromanalyse eine wichtige Methode für die Industrie geworden. Deshalb integrieren immer mehr Unternehmen die Methode in ihre Prozessoptimierungsstrategien. [Khalid et al., 2014]

Die vorliegende Arbeit konzentriert sich auf die technische Realisierung eines Editors, der speziell für die Beratung, Aufnahme und Optimierung von Wertströmen entwickelt wird. In der Planung und Umsetzung wird besonders auf die Bedürfnisse und Anforderungen seiner Endnutzer Wert gelegt. Hierzu zählen insbesondere Berater im Bereich Consulting, interne Wertstromexperten die in der Operational Excellence (OpEx) arbei-

ten sowie Spezialisten für Wertstromanalysen. Im Gegensatz zu der Wertstromanalyse auf dem Brown Paper [Rother and Shook, 2003, S. 16], wie in der Abbildung 1.1 zu sehen ist, soll der Web-Editor die Endnutzer bei der Aufnahme und Analyse flexibel unterstützen können.



Abbildung 1.1: Wertstromanalyse auf Brown Paper

Quelle: PROCESS FLOW DIAGRAMM / PROCESS MAPPING

Das Ziel des Editors ist es, den Prozess der Wertstromaufnahme so flexible und intuitiv wie möglich zu gestalten. Die Zielgruppe nutzt aktuell nach unseren Kundenbefragungen die traditionelle Methode Stift und Papier zur Dokumentation, da diese oft als flexibler und effizienter als bestehende Lösungen angesehen wird. Die Nutzerforschung zeigt bereits in den frühen Stadien, dass bestehende Softwarelösungen zum modellieren von Wertstromanalyse nicht den Anforderungen zum modellieren unserer Endnutzer entsprechen. Daher liegt der Fokus der Entwicklung auf der Benutzerfreundlichkeit des Editors und einem Templatesystem für Notizen, welches das schnelle Erstellen von digitalen Einträgen im Wertstrom erleichtern soll.

Die Bachelorarbeit ist in folgende Kapitel unterteilt: Das zweite Kapitel beschäftigt sich mit dem aktuellen Vorgehen bei Wertstromanalysen und gibt einen groben Einblick in den Stand der Forschung. Im dritten Kapitel werden die verschiedenen Nutzersichten beleuchtet, d.h. die Wahrnehmung des Editors vom Endnutzer, verschiedene Anwendungsfälle und Erkenntnisse aus den Umfragen mit der Zielgruppe.

„Architektur“ als viertes Kapitel gibt einen Überblick die API und die Strategie für das Branching im Versionskontrollsystem. Das fünfte Kapitel „Auswahl der Technologie“ setzt sich mit den Anforderungen an die Technologie auseinander und beschreibt die Kriterien für die Auswahl der Technologie.

Im sechsten Kapitel „Technische Umsetzung“ wird die konkrete Umsetzung erläutert und auf Aspekte der User Experience/User Interface (UX/UI) als auch auf die Implementierung eingegangen. Abschließend wird im siebten Kapitel „Zusammenfassung und Ausblick“ eine Zusammenfassung der Ergebnisse sowie ein Ausblick auf mögliche zukünftige Entwicklungen gegeben.

Kapitel 2

Wertstromanalyse und Stand der Forschung

Die Wertstromanalyse ist ein Methodik des Lean Manufacturing zur Visualisierung und Optimierung von Produktionsprozessen. Sie soll dabei helfen, die eigene Produktion bzw. Produktionsprozesse zu verstehen und zu verbessern. Dies wird ermöglicht durch die Visualisierung in einem Flussdiagramm. [Rahani and al-Ashraf, 2012]

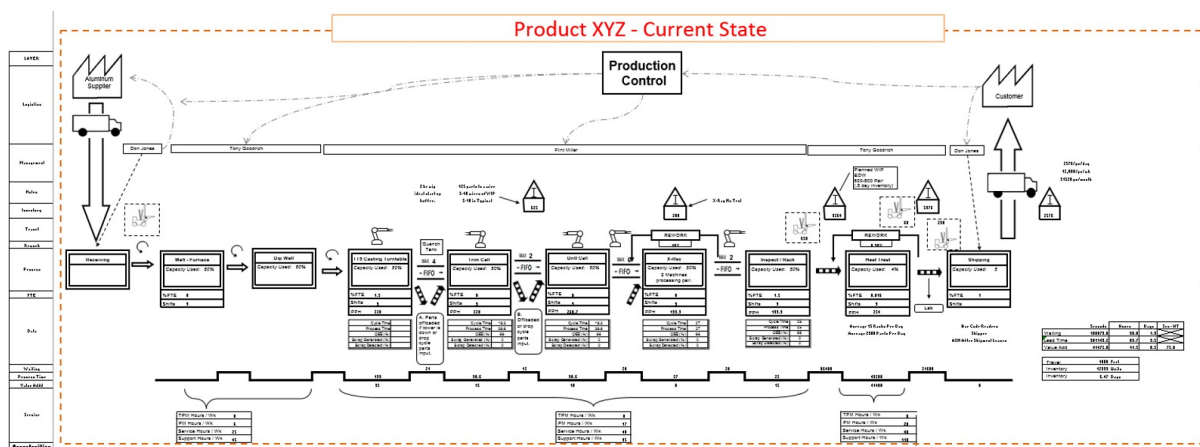


Abbildung 2.1: Ein Beispiel einer Wertstromanalyse
 Quelle: Value Stream Mapping - Show Me the Money
 Value Stream Mapping Syntax: Value Stream Mapping

Die Visualisierung, wie sie in der Abbildung 2.1 zu sehen ist, enthält den gesamten Produktionsablauf mit den Material- und Informationsflüssen. Dabei werden Wartezeiten, Transport, Prozess und Lagerbestände genauer betrachtet. Dadurch ist es möglich, die nicht wertschöpfende Aktivitäten und Bottlenecks ausfindig zu machen, um Optimierungspotenzial zu erkennen. [Gunaki et al., 2022]

2.1 Vorgehensweise zur Erstellung einer Wertstromanalyse

Die Vorgehensweise zur Erstellung einer Wertstromanalyse beginnt mit der Aufnahme des Ist-Zustandes der aktuellen Prozessabläufe. Dabei werden typische Informationen wie z.B. Zykluszeiten, Rüstzeiten, Verpackungsgröße, usw. der Produktion schriftlich festgehalten. [Rother and Shook, 2003, S. 16 ff.]

Die Modellierung des aktuellen Zustandes ermöglicht die Wertstromanalyse des Ist-Zustandes. Dieser Zustand beschreibt die aktuellen Abläufe, Materialflüsse und Informationsprozesse.

Dieser Ist-Zustand wird als Basis für die Modellierung des Soll-Zustandes genutzt. Im Soll-Zustand sind bereits alle nicht wertschöpfenden Abläufe bzw. Aktivitäten eliminiert. Das hierbei entstehende Diagramm dient nun als Grundlage für die Implementierung von Verbesserungsmaßnahmen. [Rahani and al-Ashraf, 2012]

2.2 Ziel der Wertstromanalyse

Das Hauptziel der Wertstromanalyse ist die Reduktion aller nicht wertschöpfenden Aktivitäten in den Produktionsprozessen. [Gunaki et al., 2022] Neben des Hauptzieles gibt es noch weitere Ziele und Zwecke die eine Wertstromanalyse erfüllt. Sie hilft dabei, die Prozesse besser zu verstehen und dadurch Optimierungsmaßnahmen abzuleiten. [Reichert et al., 2018, S. 59 ff.] Damit ist es möglich, eine Grundlage für die kontinuierlichen Verbesserungen der Produktionsprozesse zu schaffen.

2.3 Stand der Forschung

Der aktuelle Stand der Forschung zur Wertstromanalyse zeigt eine Vielzahl von Entwicklungen. Die Wertstromanalyse, ursprünglich aus der Automobilindustrie, wird seit dem Jahr 1999 auch in anderen Sektoren der Industrie verwendet. Vorallem in den Sektoren Fertigung, Gesundheitswesen, Bauwesen, Produktentwicklung und Dienstleistung wird die Wertstromanalyse angewendet. [Shou et al., 2017] Sie wird hauptsächlich genutzt, um nicht wertschöpfende Prozesse zu identifizieren.

Auch die Digitalisierung der Wertstromanalyse ist ein Schwerpunkt der Forschung. Hierbei werden Echtzeit-Daten der Unternehmensprozesse eingebunden und Simulationen verwendet, um die Analysen zu präzisieren bzw. zu erweitern. [Horsthofer-Rauch et al., 2022]

Es wird deutlich, dass die Wertstromanalyse kontinuierlich weiterentwickelt wird. Dadurch ergeben sich neue Herausforderungen und Möglichkeiten, die Wertstromanalyse zu verbessern.

Kapitel 3

Nutzerforschung

In diesem Kapitel wird der Endnutzer genauer betrachtet, da dieser ein zentrales Element in der Konzeption und Entwicklung von neuen Produkten und Dienstleistungen ist. Es ist wichtig ein gemeinsames Verständnis von dem zu lösenden Problem zu entwickeln, damit das Risiko eines Scheiterns der Geschäftsidee minimiert werden kann. [Eckert, 2017, S. 8]

3.1 Kundeninterviews

Die Zielgruppe wird von Beginn an und in regelmäßigen Abständen in die Entwicklungsprozesse eingebunden, da dies der zentrale Bestandteil der Strategie zum entwickeln eines neuen Produktes ist. [Eckert, 2017, S. 10] Die Identifizierung der tatsächlichen Kundenprobleme geschieht durch gezielte Interviews. In diesen wird durch gezieltes Fragestellen versucht einen tieferen Einblick in die Herausforderungen, Einstellungen, Vorgehensweisen bei der Wertstrommodellierung und Bedürfnisse der Zielgruppe zu erhalten.

Die Durchführung dieser Interviews erfolgt in einem strukturierten Ansatz. [Goodman et al., 2012, S. 129 ff.] Das Interview beginnt zunächst mit einer Einführung, in der sich alle Teilnehmer vorstellen. Dadurch soll eine vertrauensvolle Atmosphäre geschaffen werden. Anschließend folgt eine Aufwärmphase, damit der Teilnehmer sich von seinem Alltag distanzieren und die Konzentration auf das Interview gelenkt werden kann.

Nach der Aufwärmphase werden allgemeine Themen rund um das Problem behandelt. Dabei wird der Fokus auf Erfahrungen, Einstellung, Erwartungen und Annahmen gelegt. Ist der Kontext klargestellt, wird das Problem im Detail betrachtet. Der Teilnehmer wird nach seinen unmittelbaren Erfahrungen gefragt, damit ein tieferer Einblick gewonnen werden kann.

Im Anschluss folgt ein Rückblick und es werden alle wichtigen Details zusammengefasst. Das Interview endet schließlich mit einem formalen Teil und es werden administrative Themen besprochen.

Die Interviews mit der Zielgruppe im Rahmen der Bachelorarbeit wurden mit einem Moderator und zwei passiven Teilnehmern geführt. Der Moderator hat das Gespräch geleitet und die Fragen gestellt. Die passiven Teilnehmer hatten die Aufgabe, während des Gesprächs zu beobachten und „Wie können wir“-Fragen zu notieren. „Wie könne wir“-Fragen oder auch „How might we“-Fragen kommen aus der Methodik des Design Thinking. [Brown and Katz, 2011] Diese Formulierungen sind darauf ausgerichtet kreative Lösungsansätze zu fördern, indem sie Probleme in Möglichkeiten umwandeln. Dabei ist es wichtig, die Formulierungen präzise, aber nicht zu detailliert zu beschreiben und keine Lösungsansätze vorwegzunehmen.

Ein Beispiel für eine „Wie können wir“-Frage wäre: Wie können wir effiziente Tools bereitstellen, um die Vorbereitung auf die Wertstromaufnahme zu erleichtern? Durch diese Formulierung ist es möglich, die Probleme in Chancen umzuwandeln.

Nach allen Kundengesprächen wurden die Fragen, welche für die weitere Untersuchung nicht relevant waren, aussortiert und die übrigen Fragen auf einen digitales Whiteboard übertragen. Auf diesem Whiteboard wurden die Ergebnisse in verschiedene Kategorien zusammengefasst, um sie anschließend priorisieren zu können. Die Priorisierung hat ergeben, dass die „Zusammenarbeit mit/beim Kunden Vorort“, „mit der Visualisierung arbeiten“ und „die Nutzbarkeit des Tools“ für die Umsetzung im Vordergrund stehen wird im Minimum Viable Product (MVP).

3.2 Zielgruppen und Personas

Es gibt kein Produkt, welches alle Kunden gleichermaßen zufriedenstellen kann, da der Wirtschaftsmarkt sehr komplex und heterogen ist. Durch die Spezialisierung auf einen Teil der ganzen Zielgruppe ist es möglich, die Bedürfnisse und Erwartungen so gut wie möglich erfüllen zu können. [Moser, 2012, S. 46] Nach den Kundenbefragungen sind die folgenden Zielgruppen entstanden:

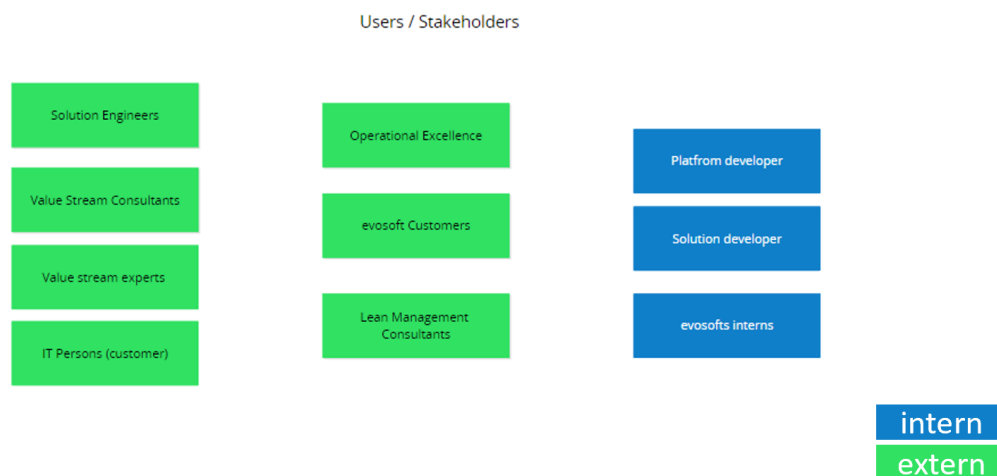


Abbildung 3.1: Brainstorming der Zielgruppen für die Applikation

In der Abbildung 3.1 sind zwei unterschiedliche Kategorisierungen zu sehen. Die blauen Karten für interne Zielgruppen und die grünen Karten für externe Zielgruppen. In den folgenden Untersuchungen werden nur die externen Zielgruppen untersucht, da aus Zeitgründen nicht alle Zielgruppen im Rahmen dieser Arbeit behandelt werden können. Die externen Zielgruppen können zusammengefasst werden wie folgt: externe Berater und interne Berater.

Nach der Definition der Zielgruppe wurden Personans beschreiben, um dem Team ein klares Verständnis für die Bedürfnisse und Anforderungen der Zielgruppe zu vermitteln. Eine Persona ist eine fiktive Person mit ähnlichen Bedürfnissen wie die Zielgruppe die stellvertretend für eine Zielgruppe steht. [Moser, 2012, S. 78]

| | |
|-----------------------|---|
| Bild |  <p>Fiktionale Darstellung von „Bernd Bauer“ Quelle: AI created images</p> |
| Name | Bernd Bauer |
| Alter | 40 |
| Hintergrund | <ul style="list-style-type: none"> • Master in Betriebswirtschaftslehre mit dem Schwerpunkt Marketing • Sieben Jahre Berufserfahrung in der Wertstromanalyse • Beschäftigung in einer international agierenden Beratungsfirma in Deutschland |
| Bedürfnisse und Ziele | <ul style="list-style-type: none"> • Optimierung und Verbesserung von Kundenwertströmen • Kunden sind ihm sehr wichtig und stehen immer an erster Stelle |
| Herausforderungen | <ul style="list-style-type: none"> • Zeitdruck und hohe Erwartungen vom Kunden • Verständnis und Optimierung von Komplexe Daten und Informationsflüsse • Schämt sich über den aktuellen Informationsaufnahmeprozess des Wertstromes mit Stift und Papier |

Tabelle 3.1: Persona von Bernd Bauer


| | |
|-----------------------|--|
| Bild |  <p>Fiktionale Darstellung von „Otto Oster“ Quelle: AI created images</p> |
| Name | Otto Oster |
| Alter | 32 |
| Hintergrund | <ul style="list-style-type: none"> • Master im Wirtschaftsingenieurwesen • Analytische, methodische und konzeptionelle Fähigkeiten sind sehr ausgeprägte • Beschäftigung bei einem Automobilzulieferer als Operational Excellence Spezialist |
| Bedürfnisse und Ziele | <ul style="list-style-type: none"> • Verbesserung der Projektdurchführung • Analyse mehrstufiger Prozesse und ihre Auswirkung auf die Produkt- und Prozessleistung |
| Herausforderungen | <ul style="list-style-type: none"> • Archivierung der bereits analog durchgeführten Wertstromanalysen in einem digitalen Tool • Einführung der neuen Kollegen im Team der Operational Excellence • Durchführung von Workshop mit Papier und Stift |

Tabelle 3.2: Persona von Otto Oster

3.3 Anforderungen aus Benutzersicht

Durch die Betrachtung der verschiedenen Benutzersichten konnten die Bedürfnisse und Herausforderungen bei der Durchführung einer Wertstromanalyse der Nutzer erkannt werden. Damit ist es möglich geworden, eine maßgeschneiderte Lösung zu entwickeln, die nicht nur technisch funktional, sondern auch benutzerfreundlich ist. Die Herangehensweise stellt gleichzeitig sicher, dass das Endprodukt den Anforderungen und Bedürfnissen der Zielgruppe entspricht. [Eckert, 2017, S. 9 ff.]

3.3.1 Benutzerszenarien

Die folgenden Abschnitte behandeln zwei ausgewählte Szenarien, welche auf den Erkenntnissen der Kundeninterviews basieren. Sie sollen ein umfassendes Verständnis für die Interaktionen, Erwartungen und möglichen Hürden der Nutzer ermöglichen. [Cooper et al., 2014, S. 102 ff.]

Benutzerszenario 1: Consulting Projekt beim Endkunden

In diesem Szenario wird das Consulting-Projekt vor Ort beim Kunden veranschaulicht.

Bernd Bauer, ein international tätiger Berater für Wertstromanalysen in seinen 40ern, ist für viele Produktionsunternehmen der Ansprechpartner für die Effizienzsteigerung in der Produktion. Bei jedem Kundenprojekt muss er vor Ort sein. Zur Vorbereitung sendet er eine Liste an Fragen zur Produktionslinie, die optimiert werden soll, an den Kunden. Nach der Rückmeldung des Kunden wird ein individuelles Template für die schriftlichen Notizen angefertigt. Diese schriftlich gefertigten Templates dienen dazu, bereits vorgefertigte Felder für notwendige Prozessdaten zu haben, um den Fokus auf die Aufnahme von Prozessdaten zu legen. Für den aktuellen Kunden wurden folgenden Felder vorbereitet: Anzahl der Mitarbeiter, Zykluszeit, Bearbeitungszeit, Rüstzeiten und Notizenfeld.

Schon bei der Ankunft beim Kunden, beginnt bereits der Informationsfluss und Bernd wird zur Linie geführt. Während des Rundganges werden detaillierte Notizen anfertigen. Es werden wichtige Key Performance Indicators berechnet und aufgeschrieben. Er kann den Kunden nicht unterberechnen, um den Informationsfluss des Kunden nicht zu stören, muss aber gleichzeitig sicherstellen, dass alle relevanten Daten erfasst werden.

Nach dem Rundgang setzt er sich nochmals mit dem Kunden zusammen und bespricht die verschiedenen schriftlich angefertigten Notizen und ergänzt diese während des Gespräches. Es werden bereits Lösungsansätze für die Optimierungsmaßnahmen formuliert und der Wertstrom wird auf dem Brown Paper graphisch modelliert.

Später im Hotel geht er die Notizen durch und ergänzt sie erneut. Sein gesamter Besuch dauert 5-Tage. In diesen Tagen werden die Notizen fortgeführt, Workshops für die Optimierungsmaßnahmen gegeben und ein

Soll-Zustand des Wertstromes modelliert. Trotz seiner langjährigen Erfahrung empfindet er, die Doppelbelastung aus Zuhören und Notieren als herausfordernd und stressig.

Benutzerszenario 2: Optimierung von internen Wertstromprojekten

Als nächstes wird das interne Optimierungsprojekt betrachtet. Dieses wird von der Operational Excellence durchgeführt. In diesem Szenario spielt besonders die interne Vorgehensweise eine bedeutende Rolle.

Otto Oster, ein Mitarbeiter der OpEx, soll den Wertstrom der Linie 7 optimieren, da der Ausschuss und die Produktionszahlen nicht einem geforderten Optimum entsprechen. Daher versammelt er verschiedene Personengruppen der Linie 7 zusammen. Das soll für die Transparenz im Team sorgen und die Akzeptanz für die Anpassung der Linie erhöhen. Im Workshop achtet er sehr darauf, dass eine gelassene Atmosphäre herrscht und es eine flache Hierarchie gibt. Dadurch soll es ermöglicht werden, zusätzliche Informationen zu dem Prozess der Linie 7 zu erfassen.

Nun muss das Team während des Workshops mit Stift und Papier durch die gesamte Produktion der Linie 7 laufen und KPIs notieren. Anschließend bringt Otto die Notizen auf einem sieben Meter breiten und zwei Meter langen Brown Paper an der Wand an. Diese Notizen werden anschließend gemeinsam im Workshop mit Materialflüssen und Kommentaren ergänzt.

Während des Workshops tauschen alle Teilnehmer ihre Erfahrungen aus und Otto kann weitere wichtige Informationen notieren. Nachdem der Workshop abgeschlossen ist, bereitet er Lösungen für die Optimierungsmaßnahmen vor. Diese werden schließlich vom Team der Linie 7 umgesetzt.

Bevor Otto das gesamte Projekt abschließen kann, muss er die wichtigsten Daten digitalisieren und das Brown Paper archivieren. Otto empfindet es mühsam, den Überblick über die bereits durchgeführte Optimierungsmaßnahmen zu behalten. Außerdem ist das Übertragen der Daten in ein digitales Format, wie z.B. Microsoft Visio ein Flussdiagramm Designer, sehr zeitaufwendig und redundant.

Basierend auf den Personas und Nutzungsszenarien können nun funktionale als auch nicht-funktionale Anforderungen an das Produkt formuliert werden. Diese beschreiben, was das Produkt können muss und welche Softwarequalität verlangt wird. [Moser, 2012, S. 86]

3.3.2 User Stories

User Stories beschreiben in wenigen Sätzen, wie der Benutzer die Software nutzen wird. Die Struktur des Satzes folgt immer dem Schema „Als [Rolle] möchte ich [Ziel], damit [Begründung]“ [Moser, 2012, S. 96], wie in der folgenden Abbildung zu sehen ist:

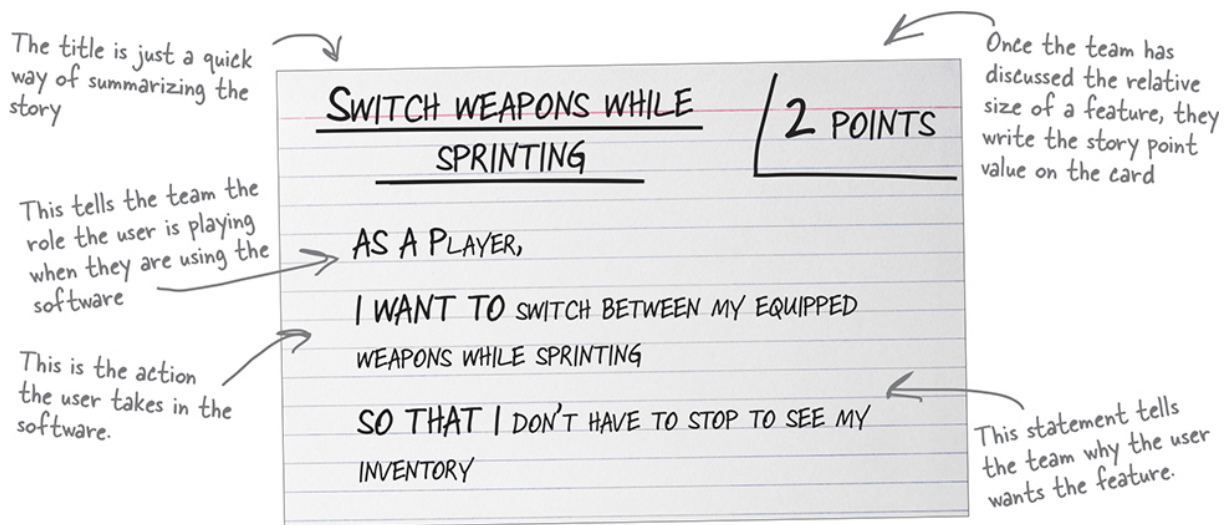


Abbildung 3.2: Beispiel für die Ausgestaltung einer Anforderung nach der Satzschablone

Quelle: Switch Weapons While Sprinting

Zusätzlich kann eine User-Story „Punkte“ erhalten, um den Aufwand zur Umsetzung abzuschätzen. Eine hohe Punktzahl steht für einen hohen Aufwand und eine niedrige Punktzahl für einen niedrigen Aufwand. Dadurch ist es möglich, eine bessere Planung und Priorisierung der Anforderungen zu ermöglichen. In der Bachelorarbeit wird auf diesen Schritt der zugrundeliegende agilen Methodik verzichtet, da alle vorgegeben Stories umgesetzt werden müssen. Außerdem enthält jeder User-Story Akzeptanzkriterien, um die Richtigkeit der Umsetzung überprüfen zu können.

Die Anforderungen an das System evoSTRIVE — Ein digitaler Ansatz zur Wertstrommodellierung: Integration von Industrie-Standards und Domänenmodellen in einem Web-Editor werden durch die folgenden User-Stories spezifiziert:

| Kennung | Name | Beschreibung |
|---------|-------------------|---|
| US-0 | Formeln im Editor | <p>Als Berater möchte ich Berechnungslogik im Editor hinterlegen, damit Berechnung innerhalb der Anwendung durchgeführt und Arbeitsabläufe effizienter gestaltet werden können.</p> <p>Akzeptanzkriterien:</p> <ul style="list-style-type: none"> • Vorgefertigte KPI-Funktionen können ausgewählt werden. • Erfasste Daten können als Eingabe für die KPI-Funktionen genutzt werden. • Statische KPI-Funktionen können hinterlegt werden. |

| | | |
|------|---------------------------------|---|
| US-1 | Datenerfassung in der Anwendung | <p>Als Berater möchte ich im Shopfloor die Prozessinformationen der Produktion erfassen können, damit ich Wertströme in Echtzeit erstellen kann, um Optimierungspotenziale schnell identifizieren zu können.</p> <p>Akzeptanzkriterien:</p> <ul style="list-style-type: none"> • Erfassung von Prozessinformationen erfolgt in zwei Modi: einem Notizen und einem Editor-Modus • Notizen-Modus ermöglicht die Prozessinformationen durch Input-Felder aufzunehmen • Editor-Modus ermöglicht die Prozessinformationen graphisch aufzunehmen • Anwendung ist responsiv, sodass sie mit dem Smartphone genutzt werden kann |
| US-2 | Erstellung von Prozesskarten | <p>Als Berater möchte ich Prozesskarten erstellen können, damit ich einen Wertstrom digital abbilden kann.</p> <p>Akzeptanzkriterien:</p> <ul style="list-style-type: none"> • Prozesskarten können erstellt werden • Prozesskarten können gruppiert werden • Prozesskarten können gelöscht werden • Prozesskarten können verknüpft werden • Prozesskarten können ein Haupt- und Subicon enthalten • Prozesskarten können Titel, Beschreibung und KPIs enthalten • Prozesskarten können eine Detailansicht mit Graphen für KPIs enthalten |

| | | |
|------|-------------------------------|--|
| US-3 | Erstellung von Prozessgruppen | <p>Als Berater möchte ich eine Prozessgruppe erstellen können, damit ich komplexe Strukturen im Wertstrom modellieren kann.</p> <p>Akzeptanzkriterien:</p> <ul style="list-style-type: none"> • Prozessgruppen können erstellt werden • Prozessgruppen können gruppiert werden • Prozessgruppen können gelöscht werden • Prozessgruppen können verknüpft werden • Prozessgruppen können Titel, Beschreibung und KPIs enthalten • Prozessgruppen können andere Karten gruppieren • Prozessgruppen können die Breite und Höhe anpassen • Prozessgruppen können eine Detailansicht mit Graphen für KPIs enthalten |
| US-4 | Verknüpfen der Karten | <p>Als Berater möchte ich im Wertstrom verknüpfen können, damit ich den Materialfluss visualisieren kann.</p> <p>Akzeptanzkriterien:</p> <ul style="list-style-type: none"> • Karten können mit Linien verknüpft werden |
| US-5 | Konfigurieren der Prozesse | <p>Als Berater möchte ich die Details meiner Prozesse anpassen können, um eine detaillierte Darstellung des Wertstroms erstellen zu können.</p> <p>Akzeptanzkriterien:</p> <ul style="list-style-type: none"> • Prozessname kann erfasst werden • Prozessdaten können erfasst werden • Prozessbeschreibung kann erfasst werden • Prozess-KPIs können erfasst werden • Prozesskommentare können erfasst werden • Prozesskarten können mit einem Haupt-Icon und einen Sekundär-Icon versehen werden • Prozess-KPIs können auf den Prozessen angezeigt werden • Prozess-KPIs können in einer Detailansicht mit Graphen angezeigt werden |

| | | |
|------|----------------------------------|---|
| US-6 | Erstellung von Notizen | <p>Als Berater möchte ich Notizen anfertigen können, damit ich nur auf die Aufnahme der Prozessinformationen konzentrieren kann.</p> <p>Akzeptanzkriterien:</p> <ul style="list-style-type: none"> • Notizen können erstellt werden • Notizen können gruppiert werden • Notizen können gelöscht werden • Notizen können bearbeitet werden • Notizenübersicht verhält sich wie eine Ordnerstruktur |
| US-7 | Erstellung von Notizen-Templates | <p>Als Berater möchte ich Notizen-Templates anfertigen können, damit ich beim Kunden bereits vorgefertigte Felder zum ausfüllen habe.</p> <p>Akzeptanzkriterien:</p> <ul style="list-style-type: none"> • Notizen-Templates können angelegt werden • Notizen-Templates können gelöscht werden • Notizen-Templates können bearbeitet werden • Notizen-Templates Inhalte können bearbeitet werden • Notizen-Templates Inhalte können die Reihenfolge verändern werden • Notizen-Templates können für die Erstellung von Notizen ausgewählt werden |
| US-8 | Import der Notizen | <p>Als Benutzer möchte ich bereits angefertigte Notizen im Editor-Modus zu Verfügung haben, damit ich schnell mit den Notizen einen Wertstrom modellieren kann.</p> <p>Akzeptanzkriterien:</p> <ul style="list-style-type: none"> • Notizen können importiert werden im Editor-Modus • Notizen stellen eine Prozesskarte im Editor dar • Ordner in der Notizenübersicht stellt eine Prozessgruppe im Editor da • Die Ordnerstruktur in der Notizenübersicht wird beibehalten in der Darstellung im Editor • Notizenfelder werden als Prozessdaten übernommen |

| | | |
|------|---------|---|
| US-9 | Theming | Als Benutzer möchte ich zwischen einem Dunkel- und Hellmodus im Web-Editor umschalten können, damit ich das Farbschema an meine Umgebung anpassen kann. Akzeptanzkriterien: <ul style="list-style-type: none">• Dunklmodus kann aktiviert werden• Hellmodus kann aktiviert werden |
|------|---------|---|

Tabelle 3.3: User-Stories

Die Nutzerforschung durch Kundeninterviews ermöglichte die Definition und Eingrenzung der relevanten Zielgruppen. Dadurch konnten die Personas erstellt werden, um die Anforderungen aus den Benutzerszenarien zu gewinnen. Dies diente als Basis für die Entwicklung der User Stories. Hierbei wurden die Bedürfnisse und Erwartungen an die Software-Lösung zur Wertstrommodellierung der Endnutzer analysiert.

Der in diesem Kapitel beschriebene Ansatz aus dem Design Thinking unterstützt die benutzerorientierte Lösungsentwicklung. Diese Methodik fördert den langfristigen Erfolg des Projekts.

Kapitel 4

Architektur

Eine Softwarearchitektur ist darauf ausgerichtet, die Geschäftsziele eines Unternehmens zu erfüllen. Sie fungiert als Brücke zwischen diesen oft abstrakten Geschäftszielen und dem konkreten System. Konkret ist eine Softwarearchitektur der Plan für ein Software-System, indem festgelegt wird, wie die Teile der Software zusammengesetzt und organisiert sind. [Carnegie Mellon University, 2023]

Der Weg von abstrakten Zielen zu konkreten Systemen ist meistens eine Herausforderung. Deshalb gibt es etablierte Techniken, um Softwarearchitekturen zu entwerfen, zu analysieren und zu dokumentieren. [Bass et al., 2021]

Ein etabliertes Hilfsmittel für die Dokumentation von Architekturen ist das arc42-Template. Es bietet einen strukturierten Rahmen, der den Prozess der Architekturdokumentation vereinfacht und unterstützt. [Starke, 2023]

Die folgenden Dokumentation enthält die REST-API und Branching-Strategie. Da bereits eine interne Architekturdokumentation mit dem arc42-Template existiert, ist die folgende Ausarbeitung als Ergänzung zu verstehen.

4.1 RESTful-API

Eine Application Programming Interface definiert die Kommunikation und den Datenaustausch zwischen unterschiedlichen Softwarekomponenten oder Systemen. Außerdem legt sie eine Menge von Regeln für den Aufbau der Interaktion fest. APIs sind unverzichtbar für die Entwicklung von modularen und skalierbaren Systemen. [Geewax, 2021] Es ist wichtig, beim Entwurf klare und gute Schnittstellendefinition zu haben.

Roy Fielding hat im Jahre 2000 in seiner Dissertation zum erstenmal die Representational State Transfer (REST) API vorgestellt. Diese Art von API nutzt ein zustandloses client-serverbasiertes Kommunikationsmodell. Es ist besonders im Kontext von Webdiensten und der Entwicklung von Internetanwendungen weit verbreitet. [Fielding, 2000]

Die RESTful-API nutzt Hypertext Transfer Protocol (HTTP)-Methoden (GET, POST, PUT, PATCH, DELETE) und ist darauf ausgerichtet Ressourcen zu manipulieren. Die Übertragung von Zuständen passiert in der Form des JavaScript Object Notation (JSON)- oder Extensible Markup Language (XML)-Formats. Ein Beispiel für einen GET-Anfrage sieht wie folgt aus:

```

1 GET /users/1 HTTP/1.1
2 Host: test.example.com
3 Accept: application/json

```

Die Antwort würde wie folgt aussehen:

```

1 HTTP/1.1 200 OK
2 Content-Type: application/json
3 Content-Length: ...
4
5
6 {
7   "users": {
8     "id": "1",
9     "name": "Test Person",
10    "links": {
11      "balances": "/accounts/1/balances",
12      "pets": "/accounts/1/pets",
13    }
14  }
15 }

```

Für die Entwicklung der REST-Schnittstellen soll auf Hypermedia as the Engine of Application State (HATEOAS) verzichtet werden, da diese Verlinkungen sehr zeitaufwendig sind. Die Antworten werden im JSON-Format erfolgen, da aktuell kein Bedarf für das XML-Format besteht.

4.1.1 Schnittstellen

Im Folgenden werden die RESTful-Schnittstellen aufgelistet und erklärt:

| Endpunkt | Methode | Beschreibung |
|-----------------------|---------|--|
| /projects | GET | Abruf aller Projekte |
| /projects | POST | Erstellung eines neuen Projekts |
| /projects/{projectId} | GET | Detailansicht eines spezifischen Projekts |
| /projects/{projectId} | PATCH | Aktualisierung eines spezifischen Projekts |
| /projects/{projectId} | DELETE | Löschung eines spezifischen Projekts |
| /views | GET | Abruf aller Ansichten |
| /views | POST | Erstellung einer neuen Ansicht |
| /views/{viewId} | GET | Detailansicht einer spezifischen Ansicht |

| | | |
|-------------------------|--------|---|
| /views/{viewId} | PATCH | Aktualisierung einer spezifischen Ansicht |
| /views/{viewId} | DELETE | Löschung einer spezifischen Ansicht |
| /kpis | GET | Abruf aller KPIs |
| /notes | GET | Abruf aller Notizen |
| /notes | POST | Erstellung einer neuen Notiz |
| /notes/{noteId} | GET | Detailansicht einer spezifischen Notiz |
| /notes/{noteId} | PATCH | Aktualisierung einer spezifischen Notiz |
| /notes/{noteId} | DELETE | Löschung einer spezifischen Notiz |
| /templates | GET | Abruf aller Vorlagen |
| /templates | POST | Erstellung einer neuen Vorlage für Notizen |
| /templates/{templateId} | GET | Detailansicht einer spezifischen Vorlage für Notizen |
| /templates/{templateId} | PATCH | Aktualisierung einer spezifischen Vorlage für Notizen |
| /templates/{templateId} | DELETE | Löschung einer spezifischen Vorlage für Notizen |

Tabelle 4.1: API Endpunkte und Create Read Update Delete (CRUD)-Operationen

4.2 Branching Strategie

Die Branching Strategie ist notwendig für ein Versionskontrollsystem, da dieses eine strukturierte und effiziente Arbeitsweise im Team ermöglicht. Außerdem hilft sie im Entwicklungsprozess Konflikte zu vermeiden und eine kontinuierliche Integration und Lieferung, auch bekannt als Continuous Integration and Continuous Delivery/Continuous Deployment (CI/CD), zu erleichtern. [Bryan, 2017] CI/CD ist eine Methodik der Software Entwicklung. Sie ermöglicht die kontinuierliche Erstellung, Testung, Bereitstellung und Überwachung von Codeänderungen. [GitLab, 2023]

4.2.1 Bedeutung der Branching Strategie

Die Branching-Strategie legt die verschiedenen Arten von Branches innerhalb eines Versionierungssystems fest. Dem Entwicklungsteam soll es ermöglicht werden gleichzeitig an verschiedenen Features, Bugfixes oder Experimenten zu arbeiten ohne sich gegenseitig zu stören. Ist die Arbeit in einem Branch abgeschlossen, kann sie in den Hauptbranch (main) integriert werden. Dies soll für reibungslose Prozesse sorgen und dabei unterstützen, Fehler und Konflikte zu vermeiden. [Bryan, 2017]

4.2.2 Umgesetzte Branching Strategy

In diesem Projekt wird die Branching Strategie von „GitFlow“ angepasst und verwendet. Diese soll einen guten Kompromiss zwischen Flexibilität und Struktur bieten. GitFlow definiert folgende Arten von Branches:

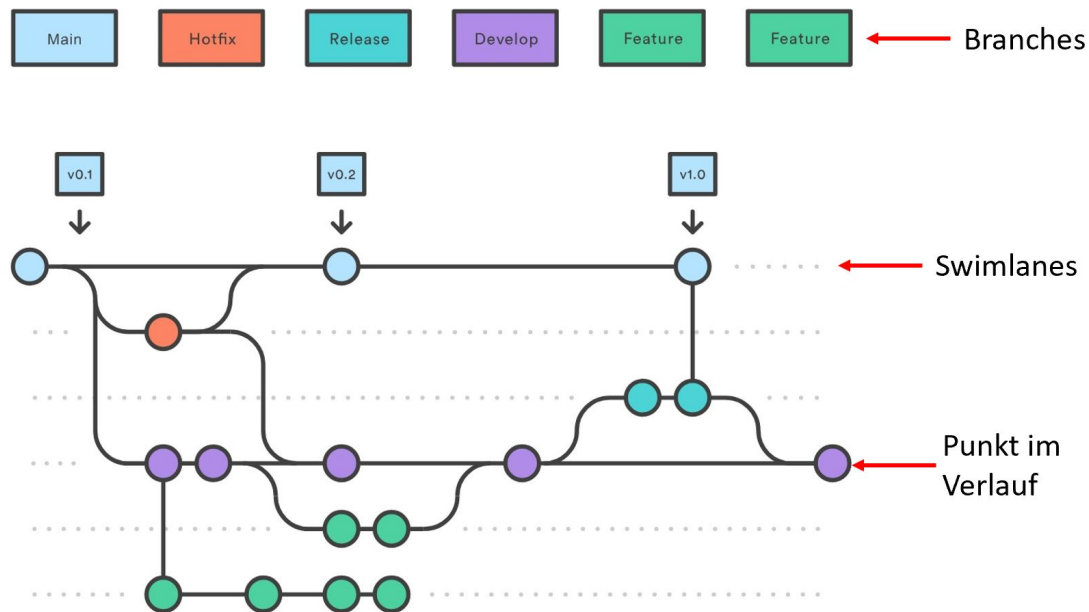


Abbildung 4.1: Visualisierung des Workflows von GitFlow

Quelle: GitFlow workflow

- **Main Branch:** Der Hauptbranch enthält immer einen deploybaren Stand des Codes.
- **Develop Branch:** Der „Develop Branch“ dient als Integrationsschicht für Features und Bugfixes.
- **Feature Branches:** Die Feature Branches werden vom „Develop-Branch“ abgeleitet und für die Entwicklung neuer Features genutzt.
- **Release Branches:** Der Release Branch dient zur Vorbereitung von neuen Versionen der Applikation und werden vom „Develop-Branch“ abgeleitet und in den „Main-Branch“ und „Develop-Branch“ gemergt.
- **Hotfix Branches:** Die Hotfix Branches dienen für dringende Bugfixes und werden vom „Main-Branch“ abgeleitet und wieder in den „Main-Branch“ und „Develop-Branch“ gemergt.

Die Strategie in diesem Projekt sieht wie folgt aus:

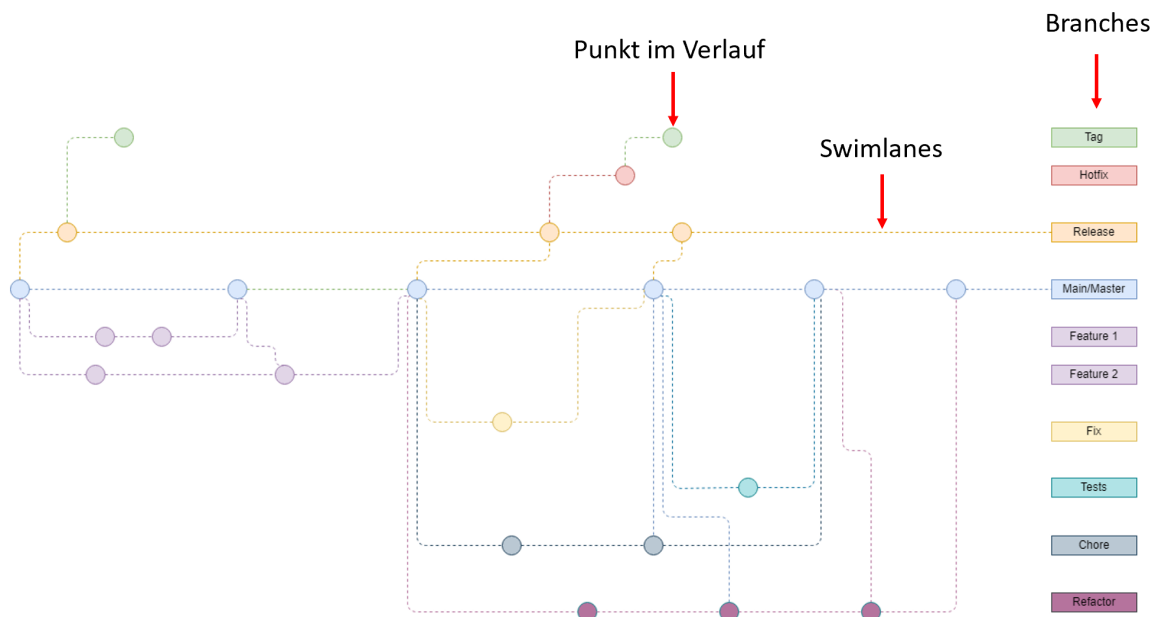


Abbildung 4.2: Visualisierung des eigenen Workflows

- **Tag:** Tags sind spezifiziert Punkt im Git-Verlauf und können nur in Produktionsumgebungen bzw. Kundenumgebungen deployt werden.
- **Hotfix Branches:** Die Hotfix Branches sind für dringende Bugfixes und werden vom „Release-Branch“ abgeleitet. Das Hotfix-Update kann nicht in den Hauptbranch zurückgeführt werden. Die langfristige Lösung muss in einen separierten Branch (Fix/Feature) erledigt werden.
- **Release Branch:** Der Release Branch enthält immer einen deploybaren Stand des Codes. Dieser kann in die Quality Assurance (QA)-Umgebung hochgeladen werden.
- **Main Branch:** Der Hauptbranch enthält immer einen deploybaren Stand des Codes. Dieser kann nur in die Entwicklungsumgebung hochgeladen werden.
- **Feature Branches:** Die Feature Branches sind vom „Main-Branch“ abgeleitet und werden für die Entwicklung neuer Features genutzt.
- **Fix Branches:** Die Fix Branches werden für bekannte Bugfixes genutzt und sind vom „Main-Branch“ abgeleitet und werden wieder in den „Main-Branch“ gemergt.
- **Test Branches:** Die Test Branches sind vom „Main-Branch“ abgeleitet und werden für die Entwicklung von Tests genutzt.
- **Chore Branches:** Die Chore Branches werden zum Anpassen von Buildprozessen oder Projektkonfigurationen genutzt. Diese werden vom „Main-Branch“ abgeleitet und wieder in „Main-Branch“ gemergt.
- **Refactor Branches:** Die Refactor-Branches werden zum Anpassen von Code genutzt, die weder ein Feature oder ein Bugfix sind. Diese sind vom „Main-Branch“ abgeleitet und werden wieder in den „Main-Branch“ gemergt.

Diese Strategie gewährleistet die strikte Trennung zwischen den verschiedenen Arbeitsaufgaben wie Features, Bugfixes, Hotfixes. Für die klare Trennung zwischen der Produktions- und Entwicklungsarbeiten sorgen die Tags. Durch diese Trennungen ist eine strukturierte Arbeitsweise möglich.

Die in diesem Kapitel beschriebene Branching-Strategie und Schnittstellendefinition dient als Basis für die Implementierung. Sie tragen zur Organisation und Qualitätssicherung bei und sind essentiell für eine strukturierte und effiziente Vorgehensweise innerhalb des Teams.

Kapitel 5

Auswahl der Technologien

Die Auswahl der Technologie in einem Softwareentwicklungsprojekt ist eine wichtige und komplexe Aufgabe, da sie die Produktivität, Schnelligkeit und Qualität der Entwicklung sehr stark beeinflussen kann. [Vazquez et al., 2022] Deshalb sollen in diesem Kapitel die Kriterien für die Auswahl der Technologie, die Entscheidung, Statemanagement und die Qualitätsabsicherung dargelegt werden.

5.1 Anforderung an die Auswahl der Technologie

Die dargelegten Kriterien leiten sich aus der Analyse aller bekannten und bisher erarbeiteten Anforderungen des Projekts ab. Dazu gehören funktionale und nicht funktionale Anforderungen sowie bereits zusammengefasste Kundeninterviews. Da der Fokus der Funktionalitäten der Software auf das Editieren und Erstellen eines Wertstromdiagramms gelegt wird, sind folgende Kriterien ein entscheidender Faktor für die Auswahl der Technologie:

- Bewegung von Elementen können
- Rotation von Elementen
- Veränderung der Größe von Elementen
- Minimap
- Zoom-In/Out
- Verbindungen von Elementen durch Linien
- Anpassbarkeit der Inhalte von Elementen

Diese Kriterien sind grundlegende Funktionalitäten, die für die Umsetzung des Editors benötigt werden. Sekundäre Kriterien sollen dabei helfen, die Qualität und Weitsicht der Entscheidung zu verbessern. Diese sind:

- **Performanz:** Dies bedeutet, dass die ausgewählte Technologie auch bei starker Auslastung, beispielsweise wenn 100 Elemente gleichzeitig bewegt werden, die Geschwindigkeit und der Ressourcenverbrauch des Endgerätes nicht kritisch ins Schwanken kommen.

- **Lizenz der Bibliothek:** Die Bibliotheken müssen eine Open-Source-Lizenz besitzen, da diese jegliche Freiheiten bieten. Das bedeutet, dass diese in Kundenprojekte einsetzbar sind und frei von sogenannten Copyleft sind.
- **Dokumentation der Bibliothek:** Eine gute Dokumentation sorgt für die effiziente Nutzung und das Verständnis der Technologie.
- **Reifegrad der Bibliothek:** Eine stabile Codebase zeichnet sich meist durch folgende Punkte aus: weniger Fehler im Code, eine größere Community, die bei Problemen unterstützen kann und ein engagiertes Team, die hinter dem Projekt steht. [Bahamdain, 2015]
- **Browserkompatibilität:** Diese ist besonders wichtig in der Webentwicklung, da sichergestellt werden muss, dass alle Zielbrowser der Kunden mit der Lösung arbeiten können.

Die aufgezählten Kriterien sortieren die meisten Technologie bzw. Bibliotheken aus und erhöhen die Produktivität, Qualität und besonders die Geschwindigkeit bei der Softwareentwicklung.

5.2 React, ReactFlow und Typescript

Ausgewählt wurden die Technologien React, ReactFlow und Typescript, da sie alle vorgegeben Kriterien erfüllen. React ist eine Bibliothek von Meta die im Jahr 2013 veröffentlicht wurde. React ermöglicht es durch komponentenbasierte Entwicklung, wiederverwendbare User Interface (UI)-Elemente zu erstellen. Dadurch ist es möglich die Zustände (State) und Datenflüsse (Props. bzw Properties) innerhalb der Anwendung zu managen. [Meta, 2023b]

Ein wichtiger Teil von React ist JavaScript XML (JSX), eine Syntaxerweiterung für JavaScript (JS). Diese ermöglicht es, die HTML-Strukturen in JS einzubetten. Damit wird die Verknüpfung der JavaScript-Logik mit der UI vereinfacht.

Ein Codebeispiel für React sieht wie folgt aus:

```
1  function Video({ video }) {
2    return (
3      <div>
4        {
5          // Thumbnail-Komponente: Zeigt ein Vorschaubild des
          // Videos an
6        }
7      <Thumbnail video={video} />
8
9      {
10     // Video Informationen: Ein Link zum Video
11   }
12   <a href={video.url}>
13     <h3>{video.title}</h3>
14     <p>{video.description}</p>
```

```

15     </a>
16
17     {
18         // LikeButton-Komponente: Ermöglicht dem Benutzer,
           das Video zu liken
19     }
20     <LikeButton video={video} />
21 </div>
22 );
23 }

```

Listing 5.1: Codebeispiel für eine Video-Komponente
Quelle: React

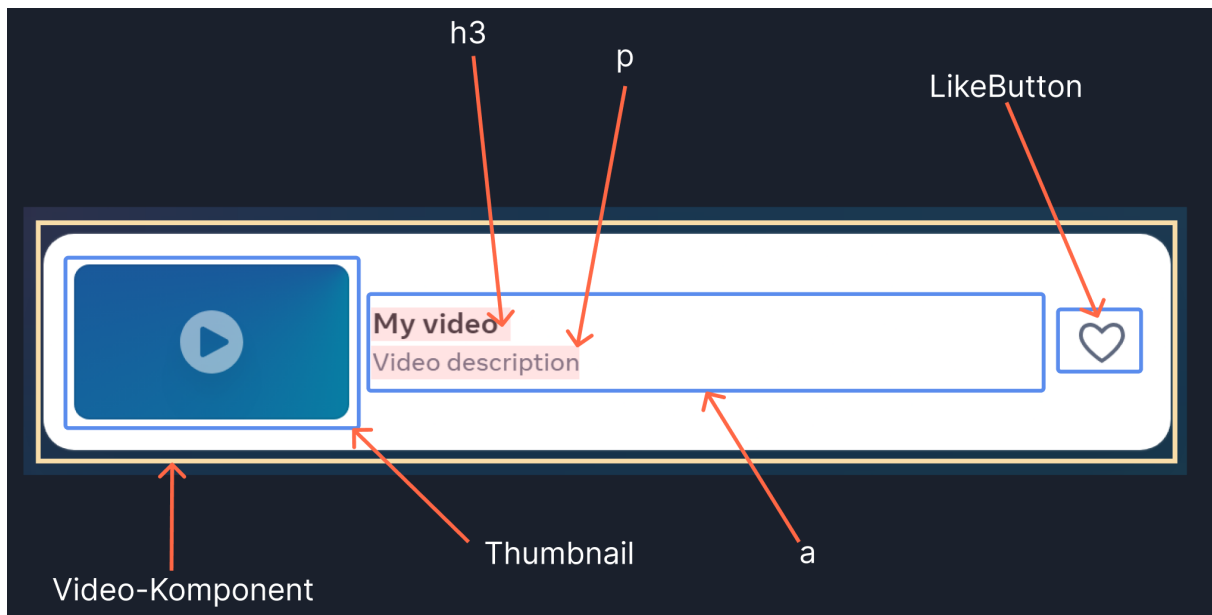


Abbildung 5.1: Visualisierung des Codebeispiels
Quelle: React Video Komponente

Die Komponente aus dem Codebeispiel erzeugt die UI-Komponente, die Abbildung 5.1 zu sehen ist. Dabei sind der „Thumbnail“ aus Zeile vier und „LikeButton“ aus Zeile neun eine eigene Komponente. Durch die verschiedenen Komponenten können Implementierungslogiken abstrahiert werden. Der Inhalt für die Komponente wird über das Property „video“ übergeben.

ReactFlow ist eine Bibliothek im React-Ökosystem. Sie ermöglicht die einfache Visualisierung von Diagrammen bis hin zu visuellen Editoren. ReactFlow ermöglicht interne Komponenten wie Knoten (Nodes) und Kanten (Edges) anzupassen durch eigene React-Komponenten.

Zusätzlich bietet ReactFlow Plugins wie z.B. Hintergrundmuster, eine Minikarte, Steuerungselement für die Karte und Werkzeugleiste für Nodes bzw. Edges an. Die Biblio-

theK wurde mit Typescript entwickelt, dies sorgt für eine einfache Integration mit einem Typescript-Projekt. [xyflow, 2023c]

Codebeispiel für eine einfache Darstellung von zwei Nodes mit einer Edge:

```
1 // Importieren der notwendigen Komponenten und Funktionen
2 import React, { useCallback } from 'react';
3 import ReactFlow, {
4   MiniMap,
5   Controls,
6   Background,
7   useNodesState,
8   useEdgesState,
9   addEdge,
10 } from 'reactflow';
11
12 // Importieren der Standard-Styles von ReactFlow
13 import 'reactflow/dist/style.css';
14
15 // Initialisierung der Startknoten und -kanten
16 const initialNodes = [
17   { id: '1', position: { x: 0, y: 0 }, data: { label: '1' } },
18   { id: '2', position: { x: 0, y: 100 }, data: { label: '2' } },
19 ];
20 const initialEdges = [{ id: 'e1-2', source: '1', target:
21   '2' }];
22
23 // Definition der App-Komponente
24 export default function App() {
25   // State-Hooks der Knoten und Kanten
26   const [nodes, setNodes, onNodesChange] = useNodesState(
27     initialNodes);
28   const [edges, setEdges, onEdgesChange] = useEdgesState(
29     initialEdges);
30
31   // Callback-Funktion, um neue Kanten hinzuzufügen
32   const onConnect = useCallback(
33     (params) => setEdges((eds) => addEdge(params, eds)),
34     [setEdges],
35   );
36
37   return (
38     <div style={{ width: '100vw', height: '100vh' }}>
39       <ReactFlow
40         nodes={nodes}
```

```

38     edges={edges}
39     onNodesChange={onNodesChange}
40     onEdgesChange={onEdgesChange}
41     onConnect={onConnect}
42   >
43     {// Steuerungselemente der Visualisierung}
44     <Controls />
45     {// Miniaturansicht der Visualisierung}
46     <MiniMap />
47     {// Hintergrundmuster der Visualisierung}
48     <Background variant="dots" gap={12} size={1} />
49   </ReactFlow />
50 </div>
51 );
52 }

```

Listing 5.2: Codebeispiel für einen einfachen Flow
Quelle: ReactFlow

Der Code aus 5.2 generiert die folgende Darstellung:

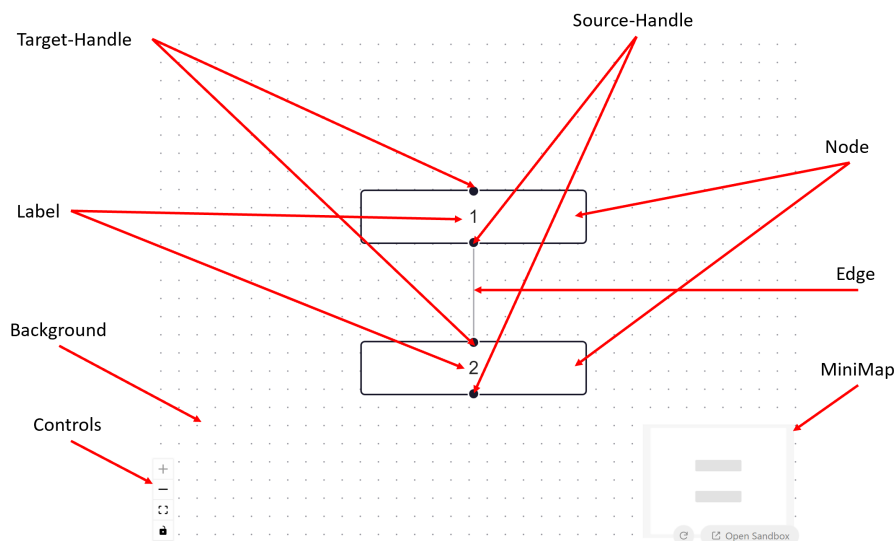


Abbildung 5.2: Visualisierung des Codebeispiels aus dem Listing 5.2
Quelle: ReactFlow: Some Extra Goodies

Mit diesem minimalen Codebeispiel aus 5.1 ist es bereits möglich, die Nodes zu bewegen, zu löschen, zu verbinden, eine Minimap und die Steuerungselement für die Karte anzuzeigen.

Die Umsetzung des Projektes erfolgt mit Typescript als Programmiersprache. Typescript wurde von Microsoft entwickelt und im Oktober 2012 mit der Version 0.8 veröffentlicht. [Turner, 2014] Typescript ist ein Superset von JavaScript. Das bedeutet, das Typescript eine Implementierung von ECMAScript [Ecma International, 2023] ist,

jedoch die Sprache mit optionaler Typisierung und Funktionen erweitert. Durch die statische Typisierung können Fehler während der Entwicklungsphase leichter erkannt werden.

Typescript bietet zusätzliche Features wie Enums, Interfaces und Dekoratoren an, welche in JavaScript nicht vorhanden sind. Diese Erweiterungen ermöglichen es, Entwicklern die Codestruktur einer Anwendung sinnvoller zu gestalten. Dies führt zu robusterem, skalierbarerem und wartungsfreundlicherem Code. [Black, 2020]

5.3 State Management

In der Webentwicklung kann das Verwalten des Anwendungszustandes (State) sehr mühsam werden, durch die große Anzahl Anwendungszuständen einer Single Page Application (SPA). Dies sorgt für eine bessere Benutzererfahrung, aber steigert die Komplexität der Verwaltung von Zuständen. Dies kann schnell zu Konflikten und Inkonsistenzen führen, deshalb soll im Projekt die Bibliothek Redux eingesetzt werden. [Bugl, 2017, S. 7]

Redux

Redux ist eine Bibliothek zur Zustandsverwaltung für React/JavaScript-Anwendungen. Sie bietet eine Möglichkeit den State der Applikation konsistent zu verwalten.

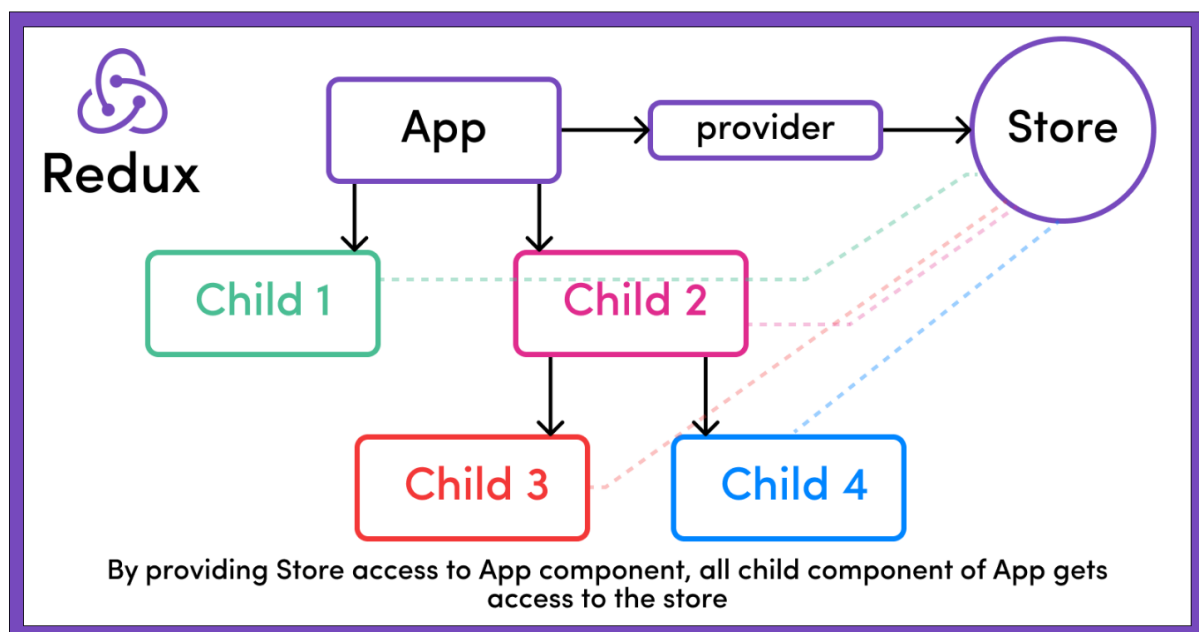


Abbildung 5.3: Zugriff auf die globale Zustandsverwaltung mit Redux

Quelle: freeCodeCamp: Core Principles of Redux

Durch eine zentralen Speicher (Store), ein unveränderliches Objekt, werden alle Zustandsinformationen an einem einzigen Ort gespeichert und können dort von Kom-

ponenten abgerufen werden [siehe Abbildung 5.3]. Dies führt dazu, dass das Testen und die Fehlersuche vereinfacht wird, da Zustand zentral an einem Ort verwaltet wird. Zudem hat Redux ein strenges Regelwerk, welches zur Konsistenz bei der Entwicklung der Applikation beiträgt. [Abramov and et al., 2023b]

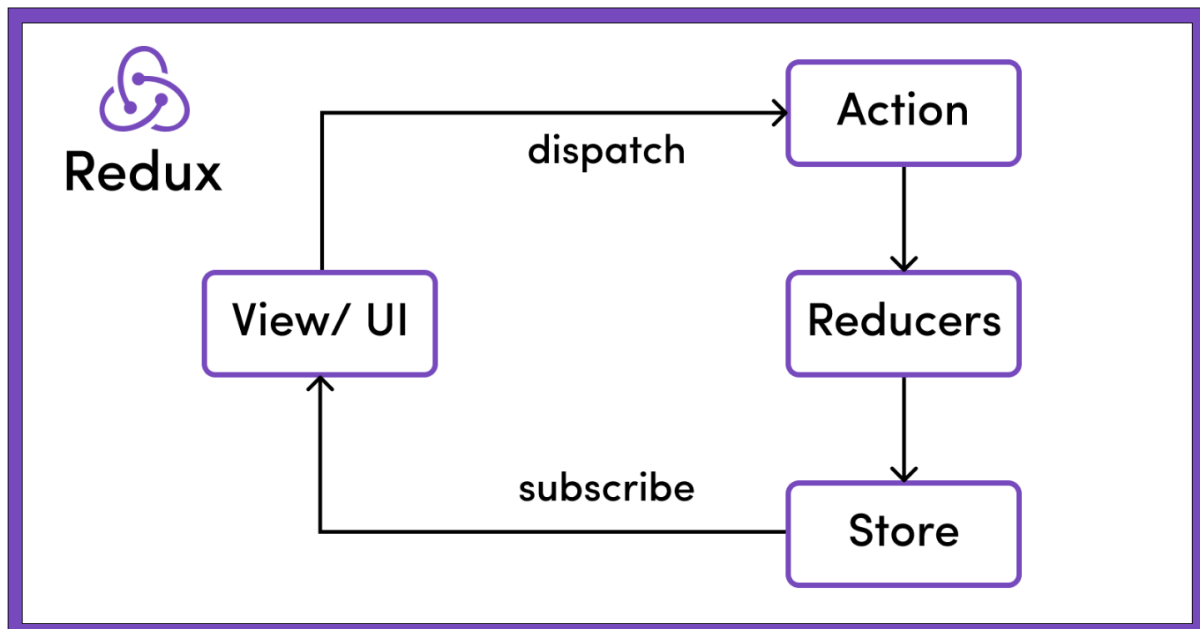


Abbildung 5.4: Visualisierung der Zustandveränderung in der Applikation mit Redux

Quelle: freeCodeCamp: Core Principles of Redux

Redux implementiert die Flux Architektur, um den Zustand zu manipulieren. Durch „Actions“ werden Veränderungen an den Store übermittelt. Actions sind JavaScript Objekte, die Informationen über die Veränderung enthalten. Die Veränderung wird an alle „Reducer“ weitergegeben. Reducer sind reine Funktionen, die den aktuellen State und die Action übergeben bekommen und als Ergebnis einen neuen State zurückgeben. Dieser wird an die Benutzeroberfläche weitergeleitet die Veränderung angezeigt. [Abramov and et al., 2023a]

5.4 Qualitätsabsicherung

Damit die Qualität, Zuverlässigkeit und Funktionalität von Software sichergestellt werden kann, muss getestet werden. Das Testen von Software sorgt dafür, dass Fehler frühzeitig erkannt werden und dadurch langfristig Zeit sowie Kosten gespart werden können. [Schmitz, 2013, S. 13 ff.]

5.4.1 Ziele von Tests

Das Ziel des Testens ist die dynamische Analyse von Software. Dadurch kann das System beobachtet und analysiert werden, sodass auf das gewünschte Verhalten getestet werden kann. Damit wird es möglich, eine Schlussfolgerung zuziehen. [Naik and

Tripathy, 2011] Es gibt viele Arten, ein System zu testen. Im Rahmen der Bachelorarbeit werden nur Unit-Tests geschrieben, da sie eine isolierte Überprüfung einzelner Funktionalität bzw. Komponenten ermöglichen.

Unit-Test

Ein Unit-Test ermöglicht es, einen Teil des Systems isoliert auf seine Funktionalität zu testen. Eine etablierte Methode zum Unit-Test schreiben ist die „Red-Green-Refactor“-Methode. In der „Red-Green-Refactor“-Methode werden zuerst Unit-Tests für einen bestimmten Teil des Systems geschrieben und schlagen zunächst fehl (Red-Phase). Danach wird der Programmcode für das Bestehen des Tests geschrieben (Green-Phase). In der letzten Phase (Refactor) wird der Code verbessert, optimiert und aufgeräumt. [Gorman, 2022, S. 576 ff.]

5.4.2 Testabdeckung

Die Testabdeckung beschreibt die Quantität an Programmcode die durch Tests abgedeckt wird. [Shahid et al., 2011] Bei der Abdeckung wird nicht zwischen den Testtypen unterschieden. Testtypen können z.B. Unit-, Integration- und End-to-End-Tests sein.

Die Testabdeckung unterscheidet jedoch zwischen folgenden Abdeckungen:

- **Funktionsabdeckung:** Die Anzahl der definierten Funktionen, die aufgerufen wurden
- **Anweisungsabdeckung:** Die Anzahl der Anweisungen im Programm, die ausgeführt wurden
- **Zweigabdeckung:** Die Anzahl der Zweige der Kontrollstrukturen (wie Schleifen und If-Anweisungen), die ausgeführt wurden
- **Bedingungsabdeckung:** Die Anzahl der booleschen Teil-Ausdrücke, die auf einen wahren und einen falschen Wert getestet wurden
- **Zeilenabdeckung:** Die Anzahl der Zeilen des Quellcodes, die getestet wurden

Für die gesamte Testabdeckung wird sich an die internen Vorgaben der evosoft GmbH gehalten. Diese sorgen für ein ausgewogenes Verhältnis zwischen Testabdeckung und effiziente Umsetzung des Editors.

Die Technologieauswahl React, Redux und ReactFlow soll für eine einfache Umsetzung der Softwarelösung sorgen. Parallel sollen Typescript sowie eine ausreichende Testabdeckung die Qualität der Softwarelösung sicherstellen.

Kapitel 6

Technische Umsetzung

Dieser Abschnitt beschäftigt sich mit der technischen Umsetzung des Editors. Es wird zuerst die Projektstruktur dargelegt. Im Anschluss wird die Gestaltung und das Benutzererlebnis beschrieben. Anschließend wird auf die spezifische Entwicklungen der UI-Komponent eingegangen.

6.1 Projektstruktur

Im Gegensatz zu anderen Frameworks zur Frontend-Entwicklung, schreibt die Bibliothek React keine spezifische Ordnerstruktur vor. [Meta, 2023a] Diese Freiheit ermöglicht, eine eigene Struktur, welche passend an die Anforderung eines Projektes gewählt werden kann, vorzugeben. Im folgenden wird die Struktur des Projektes vorgestellt:

| Ordner | Beschreibung |
|-----------------|---|
| root | Enthält grundlegende Konfigurationsdateien für das gesamte Projekt. |
| /dist | Enthält den optimierten Programmcode für die Produktionsumgebung. |
| /public | Enthält statische Assets wie ein „favicon.ico“, welches ein kleines Symbol für den Browser-Tab ist. |
| /src | Enthält den Quellcode. |
| /src/assets | Enthält Ressourcen wie Bilder und Schriftarten für die Gestaltung der Applikation. |
| /src/components | Enthält alle wiederverwendbare React-Komponenten, die in der Applikation genutzt werden. |
| /src/config | Enthält Konfigurationsdateien und Einstellungen wie z.B. für externe APIs oder Umgebungsvariablen. |
| /src/context | Enthält React Contexts zur globalen Zustandsverwaltung der Applikation. |

| | |
|---------------|---|
| /src/hooks | Enthält benutzerdefinierte React Hooks, die wiederkehrende Funktionalitäten beinhalten. |
| /src/services | Enthält Geschäftslogik die kritisch für die Funktionen der Applikation sind. |
| /src/store | Enthält das Zustandsmanagement für Redux. |
| /src/types | Enthält TypeScript-Typendefinitionen für die Applikation. |
| /src/utils | Enthält alle Hilfsfunktionen für die gesamte Applikation. |

Tabelle 6.1: Ordnerstruktur des MVPs für die Bachelorarbeit

6.2 UX/UI

Die Benutzeroberfläche (UI) und die Benutzererfahrung (UX) sind von großer Wichtigkeit in der Umsetzung des Editors. Da die Gestaltung nicht nur eine ästhetische Rolle spielt, sondern auch die Benutzerfreundlichkeit und Nutzungsgeschwindigkeit der Anwendung beeinflusst sowie deren Akzeptanz des Nutzers. Es werden hausinterne UI-Komponenten [SIEMENS, 2023] in Kombination mit der MUI [MUI, 2023b] Bibliothek genutzt, um eine anwenderfreundliche Oberfläche zu schaffen.

6.2.1 UI-Komponenten

Die Nutzung von bestehenden UI-Komponenten trägt zur Effizienz beim Entwicklungsprozess bei. Der Fokus bei der Entwicklung kann auf die Kernfunktionalitäten des Editors gelegt werden, anstatt auf die Erstellung von grundlegenden UI-Elementen. [Delgado et al., 2016] Außerdem gewährleistet die Verwendung der UI-Komponenten eine hohe Benutzerfreundlichkeit und Zuverlässigkeit, da sie von eigenen Entwicklungsteams getestet und instandgehalten werden.

Grundlegende UI-Elemente

Der Web-Editor für Wertströme nutzt folgende grundlegende UI-Elemente:

- **Buttons und Schaltflächen:** Werden verwendet, um Aktionen wie das Speichern von Änderungen oder das Öffnen von Menüs auszuführen.
- **Textfelder und Eingabefelder:** Werden verwendet, um den Benutzern die Möglichkeit Informationen einzugeben oder zu bearbeiten.
- **Dropdown-Menüs:** Werden verwendet, um aus einer Liste von Optionen zu wählen, insbesondere im Konfigurationsbereich des Editors.
- **Tabs und Navigationsleisten:** Werden verwendet, um den Editor zu strukturieren und eine schnelle und einfache Navigation zwischen verschiedenen Abschnitten und Funktionen zu ermöglichen.
- **Dialogfenster und Pop-ups:** Werden verwendet, um Warnungen anzuzeigen oder zusätzliche Informationen bereitzustellen, ohne den aktuellen Arbeitsfluss zu unterbrechen.

- **Tooltips und Hilfetexte:** Werden verwendet, um kleine Textfelder erscheinen zu lassen, wenn der Benutzer mit dem Mauszeiger über ein Element schwebt. Diese bieten nützliche Informationen zur jeweiligen Funktion.
- **Icons:** Visuelle Symbole, die verwendet werden, um Funktionen oder Aktionen intuitiv darzustellen. Dies soll die Benutzerführung innerhalb des Editors erleichtern.
- **Typography:** Wird verwendet, um die Schrift konsistent und responsiv für das Endgerät darzustellen. Es gibt einige vorgefertigte Varianten von Schriftgrößen, aus ausgewählt werden kann.
- **Grundlegende Layout-Container:** Strukturelle Elemente, die dazu dienen, den Inhalt und die UI-Komponenten räumlich zu organisieren.

6.2.2 Theming

Das Theming des Web-Editors ermöglicht die visuelle Oberfläche flexibel zu gestalten, um sich an verschiedenen Benutzeranforderungen und unternehmenseigenen Farbpaletten anpassen zu können. Es wird ein Dunklmodus als auch einen Hellmodus angeboten. Der Dunklmodus ist ideal für Umgebungen mit geringer Beleuchtung und reduziert die Augenbelastung. Der Hellmodus ist eine helle Ansicht des Editors für gut beleuchtete Arbeitsplätze.

Neben den Modi für dunkle und helle Designs gibt es eine globale Schnittstelle zur Farbanpassung. Damit wird es ermöglicht das Farbschema des Editors zu konfigurieren, um es an die Corporate Identity des Kunden anzupassen. Das Theming-Feature des Web-Editors, ist aus den Kundeninterviews hervorgegangen.

6.3 Funktionalitäten und Features

Dieses Kapitel konzentriert sich auf die zentralen Funktionalitäten des Web-Editors für Wertströme. Es werden grundlegende Werkzeuge und Funktionen des Editors vorgestellt.

6.3.1 Basisfunktionalitäten des Editors

Der Editor bietet vielseitige Funktionen zur Erstellung und Handhabung von Wertströmen. Diese Funktionalitäten sind wichtig für die effiziente Modellierung und Verwaltung von Wertströmen. Im folgenden werden die Funktionalitäten aufgelistet, die ergänzend zu den Basisfunktionalitäten von ReactFlow implementiert wurden:

- Erstellung von Prozesskarten
- Gruppierung von Prozesskarten
- Größenanpassung von Gruppen
- Import von Notizen aus dem Notizenmodus
- Konfiguration des Inhaltes von Prozesskarten
- Templateerstellung für Notizen
- Erstellung von Notizen

Diese grundlegenden Funktionalitäten bilden das Fundament für den MVP-Web-Editor. Sie sollen ein hohes Maß an Flexibilität und Kontrolle über die Darstellung von Wertströmen bieten. Außerdem unterstützt der Notizenmodus dabei, die Prozessdaten in kurzer Zeit und digital aufzunehmen.

6.3.2 Erstellung von Prozesskarten

Über eine Seitenleiste kann man Prozesskarten erstellen. Diese sieht wie folgt aus:

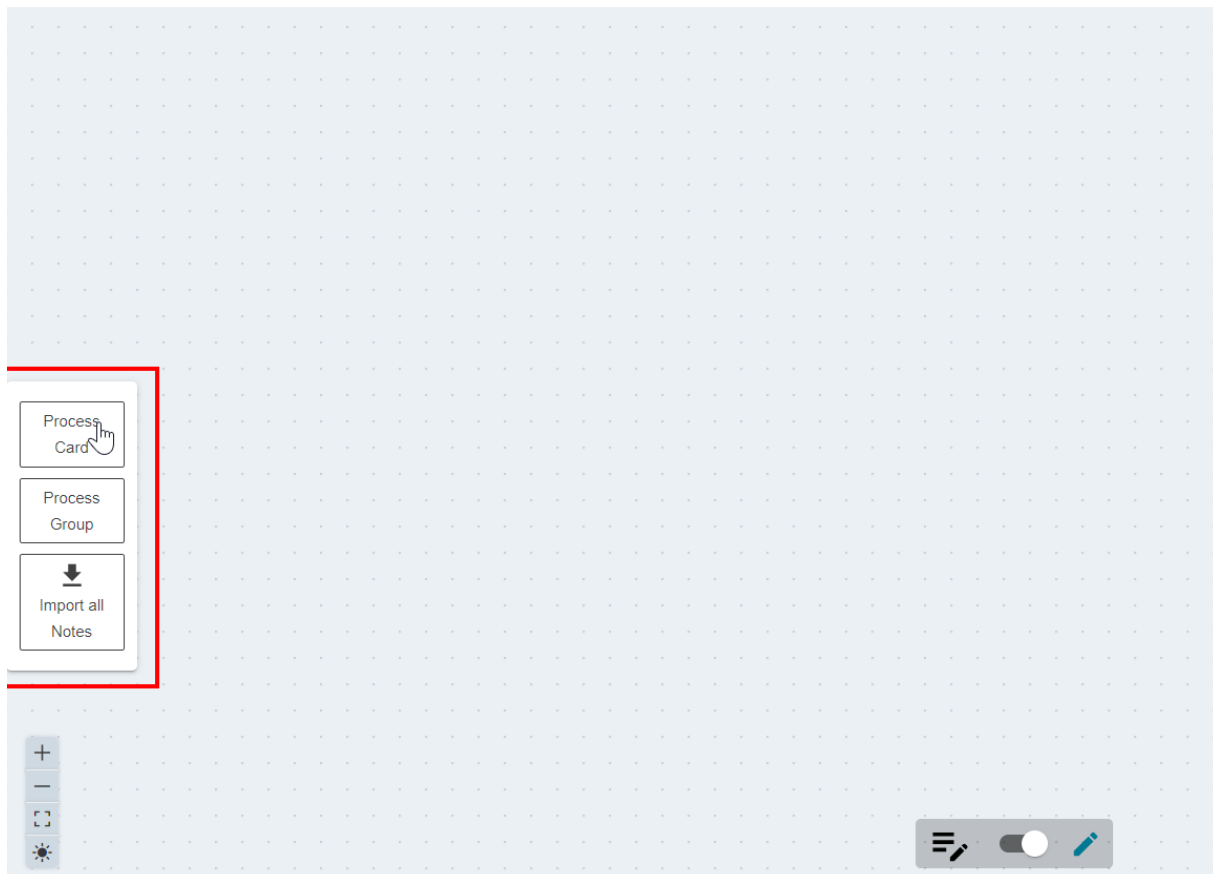


Abbildung 6.1: Seitenleiste zur Erstellung von verschiedenen Prozesskarten

Die Seitenleiste, wie sie links in der Abbildung 6.1 zu sehen, verfügt über drei Buttons. Diese ermöglichen die Generierung von Prozesskarten. Der erste Button erstellt eine simple Prozesskarte. Diese enthält als Standardkonfiguration ein Haupt- und Subicon zur visuellen Kennzeichnung. Der zweite Button erstellt eine Prozesskarte welche andere Prozesskarten gruppieren kann und dadurch komplexe Strukturen visualisiert werden können. Die Prozessgruppenkarte besitzt keine Icons und kann auch keine Icons enthalten. Der dritte Button ermöglicht die Konvertierung von Notizen zu Prozesskarten und Prozessgruppenkarten aus aufgenommenen Notizen. Diese Funktionalität ermöglicht es, die Notizen schnell in die Visualisierung zu übertragen.

Art von Prozesskarten

Es gibt insgesamt zwei verschiedene Prozesskarten, die sich in einer Funktionalität unterscheiden. Die erste Art von Karte ist die Prozesskarte und die zweite Art ist eine Prozessgruppe. Die Prozesskarte repräsentiert einen Prozess oder eine Ressource. Eine Prozesskarte sieht wie folgt aus:

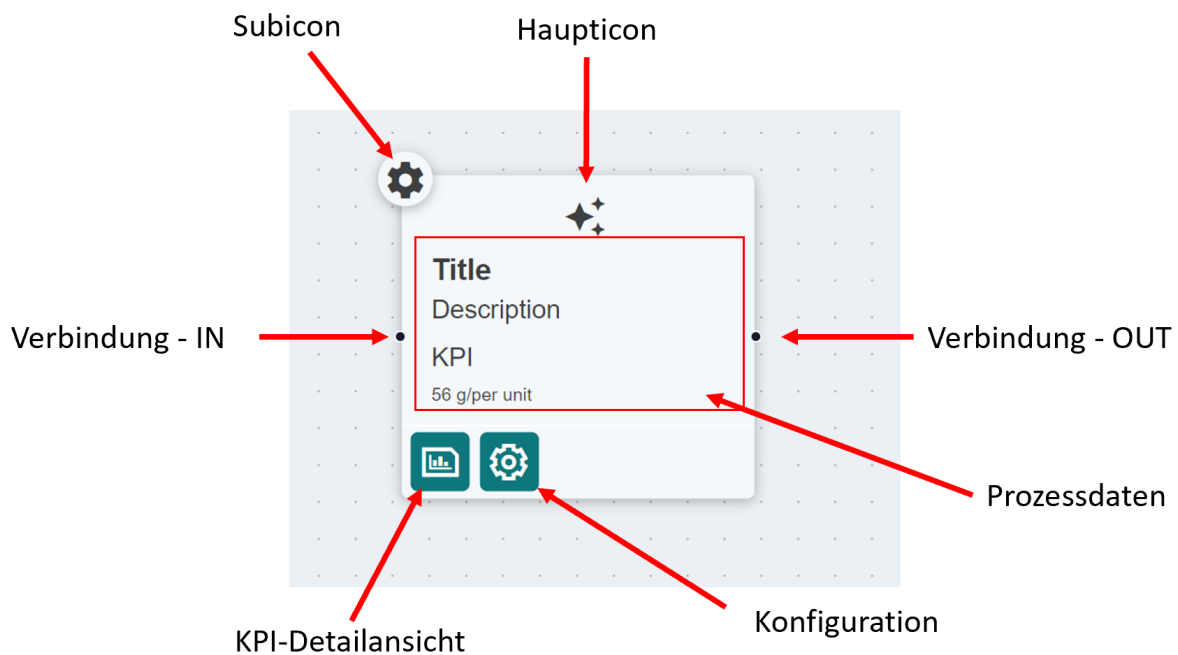


Abbildung 6.2: Ein Beispiel einer vorkonfigurierten Prozesskarte

Die Prozesskarte, wie in der Abbildung 6.2 zu sehen, besitzt zwei Icons. Das Haupticon repräsentiert die primäre Funktion bzw. Kategorie der Prozesskarte. Das Subicon wird verwendet, um zusätzliche Funktionen oder Unterkategorien der primären Funktion bzw. Kategorie zu repräsentieren.

Jede Karte besitzt eine IN- und OUT-Verbindung. Durch die IN-Verbindung kann eine Prozesskarte mit einer anderen verknüpft werden. Durch die OUT-Verbindung kann die Prozesskarte mit anderen Karten verbunden werden. Eine Verbindung stellt einen Materialfluss dar.

Jede Prozesskarte enthält einen Konfigurationsbutton und einen Graphenbutton, wenn dieser konfiguriert wurde. Der Konfigurationsbutton ermöglicht die Bearbeitung der Prozessdaten bzw. die Ergänzung von Informationen eines Prozesses. Die Prozessdaten sind Titel, Beschreibung, statische Daten, KPI-Anbindungen und die KPI-Graphenanbindung. Visuell sind nur Titel, Beschreibung und KPIs auf der Prozesskarte zusehen, da diese Informationen primär wichtig sind für unsere Zielgruppe.

Die Prozessgruppe kann alle Karten gruppieren und hat dieselben Funktionalitäten, wie eine Prozesskarte. Die Prozessgruppe sieht wie folgt aus:

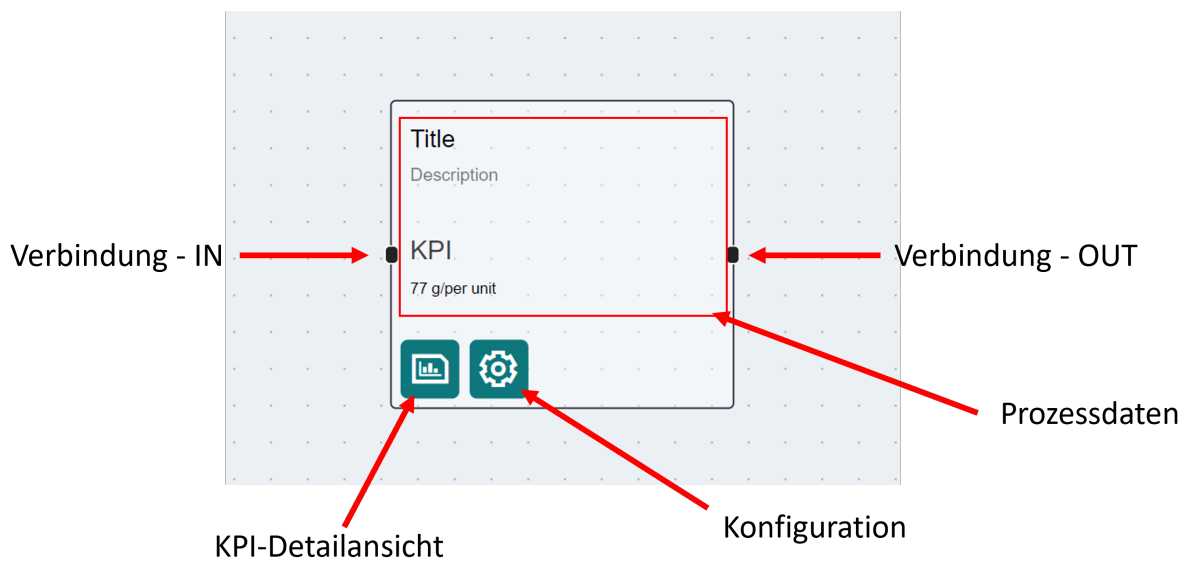


Abbildung 6.3: Das Aussehen einer Prozessgruppe

Die Prozessgruppe besitzt kein Haupt- oder Subicon, jedoch sind alle restlichen Funktionalitäten der Prozesskarte vorhanden. Zusätzlich kann die Prozessgruppe ihre Größe dynamisch verändern, damit sie sich automatisch auf ihren Inhalt anpassen kann. Diese Funktionalität wird grafisch wie folgt dargestellt:

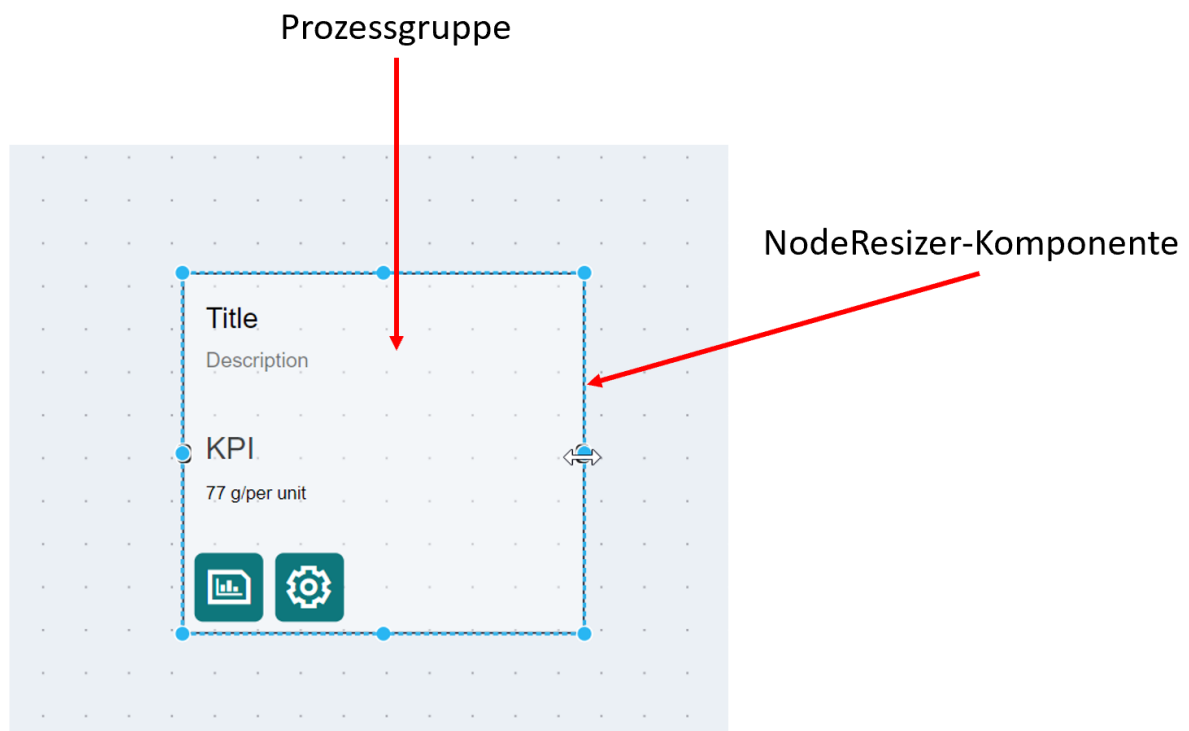


Abbildung 6.4: Anpassung der Größe einer Prozessgruppe

Das Anpassen der Größe von Prozessgruppen soll dem Benutzer die individuelle Gestaltung ermöglichen. Die Anpassung erfolgt durch die Bibliothek „node-resizer“, welche eine Erweiterung von ReactFlow ist. Die Integration wird in Listing 6.1 abgebildet:

```
1 // Importieren von Komponenten
2 import { Handle, Position, NodeResizer } from 'reactflow';
3
4 // Definition der ResizableNode-Komponente
5 const ResizableNode = ({ data }) => {
6   return (
7     <>
8       {/* ermöglicht das Ändern der Größe der Node */}
9       <NodeResizer minWidth={100} minHeight={30} />
10
11      {/* Handle für eingehende Verbindungen auf der linken
12       Seite*/}
13      <Handle type="target" position={Position.Left} />
14
15      {/* Inhalt der Node */}
16      <div style={{ padding: 10 }}>{data.label}</div>
17
18      {/* Handle für ausgehende Verbindungen auf der rechten
19       Seite */}
20      <Handle type="source" position={Position.Right} />
21    </>
22  );
23 }
```

Listing 6.1: Codebeispiel für eine Node mit einem Resizer
Quelle: ReactFlow - NodeResizer

Das Codebeispiel aus Listing 6.1 ermöglicht das Erstellen einer individuell angepassten Node. Diese kann nun ihre Größe dynamisch verändern und besitzt eine Mindestbreite von 100 und Mindesthöhe von 30 Pixeln. Die Implementierung einer Custom-Node wird im folgenden Kapitel näher erläutert.

Zusätzlich sollen Prozessgruppen die unterschiedlichen Detailebenen in einem Wertstrom ermöglichen. Eine Detailebene ist der Grad der Genauigkeit von Informationen in einer Darstellung. Die Erstellung von Detailebenen ist durch dynamisches Drag & Drop von anderen Karten in Prozessgruppen möglich. Die Darstellung von Detailebenen sieht wie folgt aus:

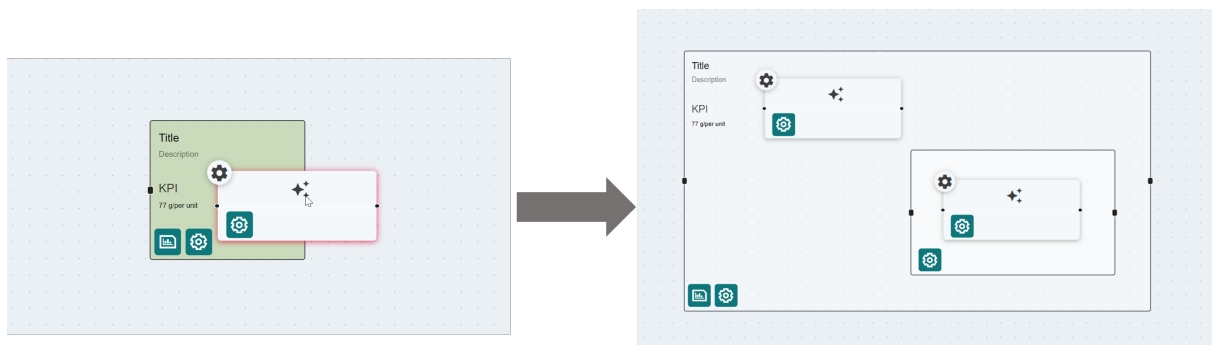


Abbildung 6.5: Gruppierung von Elementen zur Erschaffung von Detailebenen

Die Gruppierung oder auch rekursive Gruppierung von Karten soll es ermöglichen, komplexe Strukturen auf eine übersichtliche Weise darzustellen. Ist eine Gruppierung entstanden, kann diese durch das Elternelement bewegt werden.

Die Implementierung erfolgt durch die Modifizierung der internen Nodekonfiguration. Eine Node kann die Referenz zu einem Elternelement halten, indem sie die ID des Elternelements unter ihrer Konfiguration 'parentNode' referenziert. Folgend wird ein Codebeispiel aufgeführt:

```

1 // Importieren von Komponenten und Funktionen
2 import ReactFlow, {Background, useNodesState} from 'reactflow
  ';
3
4 // Definition der NestedFlow-Komponente
5 const NestedFlow = () => {
6   // Verwendung der useNodesState Hooks
7   const [nodes, setNodes, onNodesChange] = useNodesState([
8     {
9       id: '2', // ID der Node
10      data: { label: 'Group A' }, // Inhalt der Node
11      position: { x: 100, y: 100 }, // Position der Node im
        Flow
12      className: 'light', // Klassenname für zusätzliche CSS-
        Styles
13      style: { backgroundColor: 'rgba(255, 0, 0, 0.2)', width:
        200, height: 200 }, // Styles der Node
14    },
15    {
16      id: '2a',
17      data: { label: 'Node A.1' },
18      position: { x: 10, y: 50 },
19      parentNode: '2', // Referenzierung der Eltern-Node
20    },
21  ]);
22

```

```
23   return (
24     <ReactFlow
25       nodes={nodes} // Nodes werden übergeben
26       onNodesChange={onNodesChange} // Handler für Änderungen
          an den Nodes
27       className="react-flow-subflows-example"
28       fitView
29     >
30       <Background />
31     </ReactFlow>
32   );
33 };
34
35 export default NestedFlow;
```

Listing 6.2: Codebeispiel für eine Node mit einem Resizer
Quelle: ReactFlow - Subflow

In dem Codeausschnitt aus Listing 6.2 ist zu sehen, wie eine Node eine andere Node als Elternelement referenzieren kann (Zeile 16). Diese initiale Nodekonfiguration wird an ReactFlow übergeben (Zeile 22). Dies schafft einen Bezug und sorgt dafür, dass die Position von Kinderelementen immer im Bezug auf das Elternelement berechnet wird.

Dadurch ist es möglich Elternelemente gleichzeitig mit ihren Kinderelementen neu zu positionieren, wenn sie bewegt werden. Die Entgruppierung entsteht durch die Löschung der Referenz zum Elternelement.

Custom-Nodes

„Custom-Nodes“ sind React-Komponenten, die für die individuelle Anpassung von Nodes verwendet werden können. Das bedeutet, dass jede beliebige Funktionalität oder jedes Design implementiert werden kann, welches mit einer React-Komponente möglich ist. [xyflow, 2023b] Diese individuell angepassten Custom-Nodes werden an ReactFlow übergeben und erhalten ihre Grundfunktionalitäten wie z.B. die Drag & Drop-Funktionalität.

Das Einbinden von Custom-Nodes erweitert die Properties einer React-Komponente. [xyflow, 2023a] Dadurch ist es möglich das Standardverhalten von Nodes anzupassen bzw. zu erweitern. Eine Beispielimplementierung einer Custom-Node sieht wie folgt aus:

```
1 // Importieren von Funktionen und Komponenten
2 import React from 'react';
3 import { Handle, Position } from 'reactflow';
4
5 // Definition der CustomNode -Komponente
6 const CustomNode = ({ data, isConnectable }) => {
7   return (
```



```
8      <>
9      { /* Erster Handle (Target) links am Knoten positioniert
10         */
11         <Handle
12             type="target"
13             position={Position.Left}
14             style={{ background: '#555' }}
15             onConnect={(params) => console.log('handle onConnect
16                 ', params)}
17             isConnectable={isConnectable}
18         />
19         { /* Inhalt des Knotens */
20         <div>
21             Custom Node
22         </div>
23         { /* Zweiter Handle (Source) rechts am Knoten
24             positioniert */
25         <Handle
26             type="source"
27             position={Position.Right}
28             id="a"
29             style={{ top: 10, background: '#555' }}
30             isConnectable={isConnectable}
31         />
32         { /* Dritter Handle (Source) rechts am Knoten
33             positioniert */
34         <Handle
35             type="source"
36             position={Position.Right}
37             id="b"
38             style={{ bottom: 10, top: 'auto', background: '#555'
39                 }}
40             isConnectable={isConnectable}
41         />
42     </>
43 );
44 };
45 export default CustomNode;
```

Listing 6.3: Codebeispiel für eine Custom Node

Das Codebeispiel in dem Listing 6.3 erzeugt nun eine Node mit einer IN-Verbindung und zwei OUT-Verbindungen. Diese werden durch die Komponente „Handle“ implementiert und konfiguriert. Zusätzlich enthält sie den statischen Namen „Custom Node“, die in einem Div-Element verpackt ist. Diese neue Komponente kann nun in

ReactFlow eingebunden werden. Dies geschieht durch die Ergänzung von „nodeTypes“. Die Property „nodeTypes“ sieht wie folgt aus:

```
1 // Importieren von ReactFlow und Komponenten
2 import ReactFlow, {Background, useNodesState} from 'reactflow
  ';
3
4 // Importieren einer benutzerdefinierten Knotenkomponente
5 import CustomNode from './CustomNode.jsx';
6
7 // Definition der NestedFlow-Komponente
8 const NestedFlow = () => {
9   // Verwendung des useNodesState Hooks
10  const [nodes, setNodes] = useNodesState ([
11    {
12      id: '1', // eindeutige ID
13      position: { x: 100, y: 100 }, // Position der Node
14      type: 'newNode', // Typ der Node
15    },
16  ]);
17
18  // Definition der benutzerdefinierten Nodetypen
19  const nodeTypes = {
20    newNode: CustomNode, // neu erstellte Komponente
21  }
22
23  return (
24    <ReactFlow
25      nodes={nodes}
26      nodeTypes={nodeTypes}
27      fitView
28    >
29      <Background />
30    </ReactFlow>
31  );
32 };
33
34 export default NestedFlow;
```

Listing 6.4: Codebeispiel für die Ergänzung von Knotentypen

Dadurch ist es möglich, eine Custom-Node einzubinden und zu nutzen. Dies geschieht durch die Referenzierung des neuen Types in Zeile 15.

6.3.3 Verbindung von Elementen

Damit ein Material- oder Informationsfluss dargestellt werden kann, muss es möglich sein, Knoten miteinander zu verbinden. Diese Funktionalität ist essenziell für die Erstellung von Wertströmen. Eine Verbindung kann erstellt werden, indem eine OUT-Verbindung mit einer IN-Verbindung verbunden wird.

Custom-Edges

Eine Edge kann durch eine React-Komponente individualisiert werden. Dabei wird zwischen zwei Darstellungen unterschieden und zwar der Edge selbst und der Verbindungslinie - „Connection Line“.

Die Edge ist die Darstellung der Verbindung zwischen zwei Elementen und die Verbindungslinie ist die Darstellung der Möglichkeit zur Verbindungsherstellung. Dadurch ist es möglich, verschiedene Verbindungsdarstellungen zu visualisieren. Ein Codebeispiel für eine Custom-Edge sieht wie folgt aus:

```
1 // Importieren von notwendigen Komponenten und Funktionen
2 import React from 'react';
3 import { BaseEdge, EdgeProps, getBezierPath } from 'reactflow'
4   ;
5 // Definition der CustomEdge-Komponente
6 export default function CustomEdge({
7   id, // ID der Kante
8   sourceX, // x-Koordinate des Startknotens
9   sourceY, // y-Koordinate des Startknotens
10  targetX, // x-Koordinate des Zielknotens
11  targetY, // y-Koordinate des Zielknotens
12  sourcePosition, // Position des Startknotens
13  targetPosition, // Position des Zielknotens
14  style = {}, // optionale Styles
15  markerEnd, // Pfeilspitze
16 }: EdgeProps) {
17   // Berechnung des Bezier-Pfades für die Kante
18   const [edgePath, labelX, labelY] = getBezierPath({
19     sourceX,
20     sourceY,
21     sourcePosition,
22     targetX,
23     targetY,
24     targetPosition,
25   });
26
27   // Rendern der Kante mit übergebenen Parametern
28   return (
```

```
29     <>
30     <BaseEdge path={edgePath} markerEnd={markerEnd} style={
31         style} />
32     </>
33 );
}
```

Listing 6.5: Codebeispiel für eine Custom Edge

Dieses Codebeispiel aus Listing 6.5 ermöglicht, dass die Edge einen Marker und ein neues Aussehen erhält. Zusätzlich berechnet sie die Kurve für die Darstellung der Edge durch die übergeben EdgeProps. [Meta, 2023a] Die Custom Edge kann nun auch über edgeTypes referenziert und an ReactFlow übergeben werden.

6.3.4 Erstellung von Notizen

Die Erstellung von Notizen bezieht sich nur auf die Aufnahme von Prozessdaten. Dadurch kann der Fokus des Benutzers auf die Aufnahme von KPIs bzw. andere relevante Daten gelegt werden, ohne dass Karten im Editor erstellt werden müssen. Somit ist eine Datenaufnahme vereinfacht und der Nutzer kann sich auf eine Analyse und Bewertung von Kundendaten konzentrieren.

Die allgemeine Bearbeitung und Erstellung von Notizen geschieht im Notizenmodus. Der Notizenmodus ist durch die Applikationssteuerung zu erreichen, wie hier zu sehen ist:

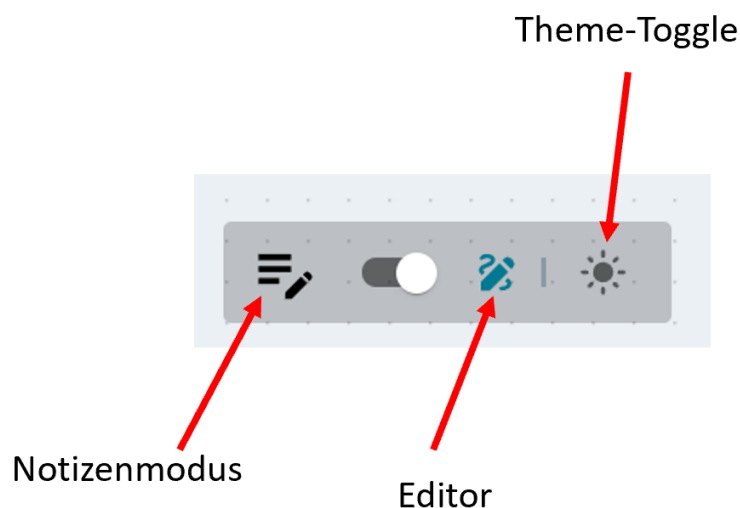


Abbildung 6.6: Applikationssteuerung

Durch den Schalter kann zwischen dem Notizenmodus und dem Editor gewechselt werden. Außerdem kann der Benutzer in der Applikationssteuerung zwischen dem dunklen und hellen Design wechseln.

Im Notizenmodus gibt es folgende Funktionalitäten:

- Erstellen von Notizen
- Erstellen von Gruppen
- Erstellen von Templates
- Ein- und ausklappen von Gruppen
- Ordnen der Reihenfolge von Notizen bzw. Gruppen
- Umbenennen von Gruppen
- Ausfüllen von Notizen
- Importieren eines bestehenden Standardtemplates für Notizen
- Gruppieren der Notizen bzw. Gruppen
- Entgruppieren der Notizen
- Löschen von Templates, Notizen und Gruppen
- Ordnen der Reihenfolge des Inhaltes eines Templates

Der Notizenmodus sieht wie folgt aus:

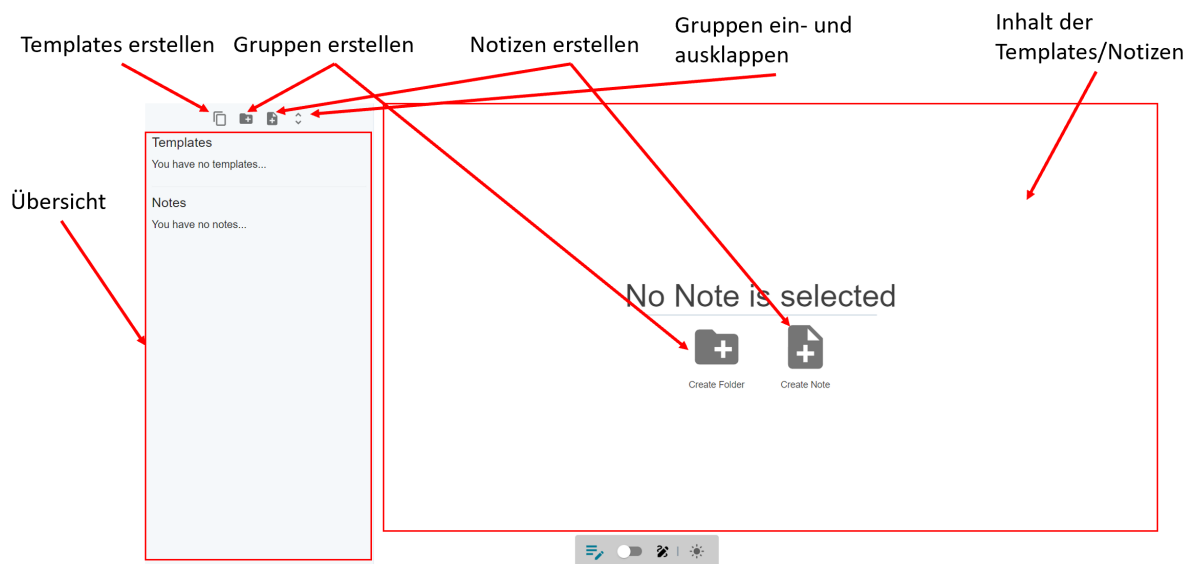


Abbildung 6.7: Die Ansicht im Notizenmodus

Auf der linken Seite sind alle Templates, Notizen und Gruppen in Ordnerstrukturen zu finden. Oben in der linken Leiste sind Buttons. Diese ermöglichen es, Templates, Gruppen und Notizen zu erstellen. Der letzte Button besitzt die Funktionalität alle Gruppen ein- und ausklappen zu lassen. Diese soll dem Benutzer ermöglichen, schnell zwischen einer detaillierteren und einer kompakten Ansicht zu wechseln.

Ist ein Template oder eine Notiz ausgewählt, wird dies sichtbar auf der rechten Seite. Der Inhalt kann modifiziert werden. Die Änderungen werden automatisch gespeichert.

Notizen erstellen/aufnehmen

Eine Notiz kann nur bearbeitet werden, wenn sie ausgewählt wurde. Ist sie ausgewählt, können folgende Felder vorhanden sein:

- **Stopuhr:** Die Stopuhr besitzt vier Textfelder. Diese sind in Stunde, Minute, Sekunde und Millisekunden unterteilt. Durch diese Aufteilung wird eine detaillierte Zeitaufnahme ermöglicht. Die Stopuhr kann durch zwei Buttons aktiviert, pausiert und gestoppt werden. Jedes Textfeld ist manuell veränderbar. Die Stopuhr soll den Benutzer innerhalb der Applikation bei der Zeitmessung unterstützen sowie ein Toolbruch reduzieren.
- **Textfeld - Number:** Dieses Textfeld erlaubt die Eingabe von Zahlen und soll als Input für die KPI-Funktionen dienen.
- **Textfeld - String:** Dieses Element ist ein normales Textfeld, welches jegliche Eingaben ermöglicht.

Untitled 1

IMPORT TEMPLATE ▾

Lead Time

Hour: 0 : Minutes: 0 : Seconds: 0 : Milliseconds: 0 ▶ ↻

Process Time

Hour: 0 : Minutes: 0 : Seconds: 0 : Milliseconds: 0 ▶ ↻

Transit Time

Hour: 0 : Minutes: 0 : Seconds: 0 : Milliseconds: 0 ▶ ↻

Rework Time

Hour: 0 : Minutes: 0 : Seconds: 0 : Milliseconds: 0 ▶ ↻

Number of operators

☰ 🔍 🗑️ ☀️

Abbildung 6.8: Ein Beispiel an Notizenfeldern

Die in Abbildung 6.8 dargestellten Felder müssen nicht ausgefüllt werden. Jedes zusehende Textfeld kann manuell ausgefüllt werden bzw. angepasst werden. Die Anzahl und die Reihenfolge der Felder kann nur in den Templates angepasst werden.

Templates für Notizen

Die Templates sollen den Nutzer bei der Vorbereitung der Aufnahme von Prozessdaten unterstützen. Dem Benutzer stehen bei der Anfertigung von Templates drei verschiedene Typen zur Auswahl. Diese können wie folgt ausgewählt werden:

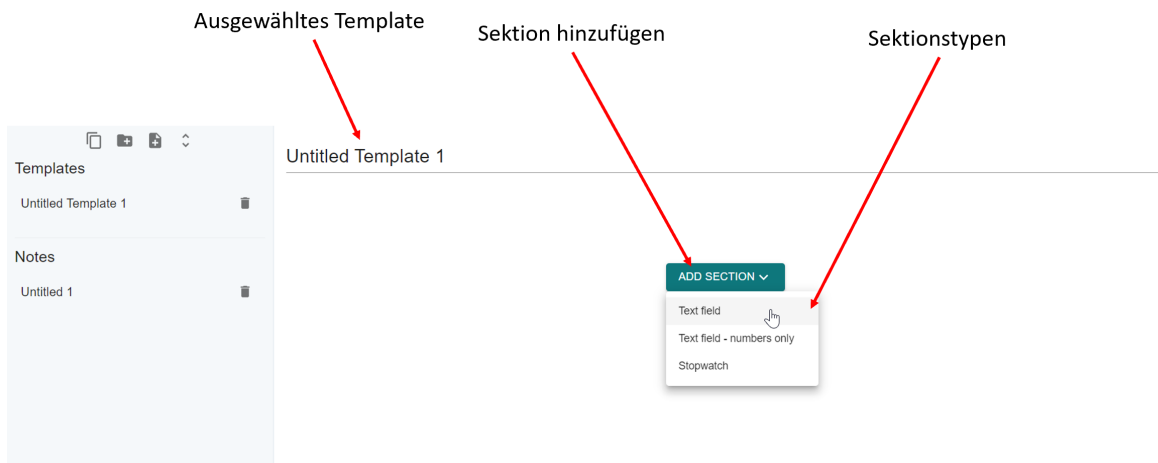


Abbildung 6.9: Feldtypen die in einem Template erstellt werden können

Der Name von Templates kann jederzeit angepasst werden. Erstellte Felder enthalten einen automatisch generierten Namen und Badges die den Typen des Feldes anzeigen. Die Sektionen können umbenannt werden, sodass sie als Titel für die Textfelder dienen können. Zudem können die verschiedenen Felder durch Drag & Drop geordnet werden:

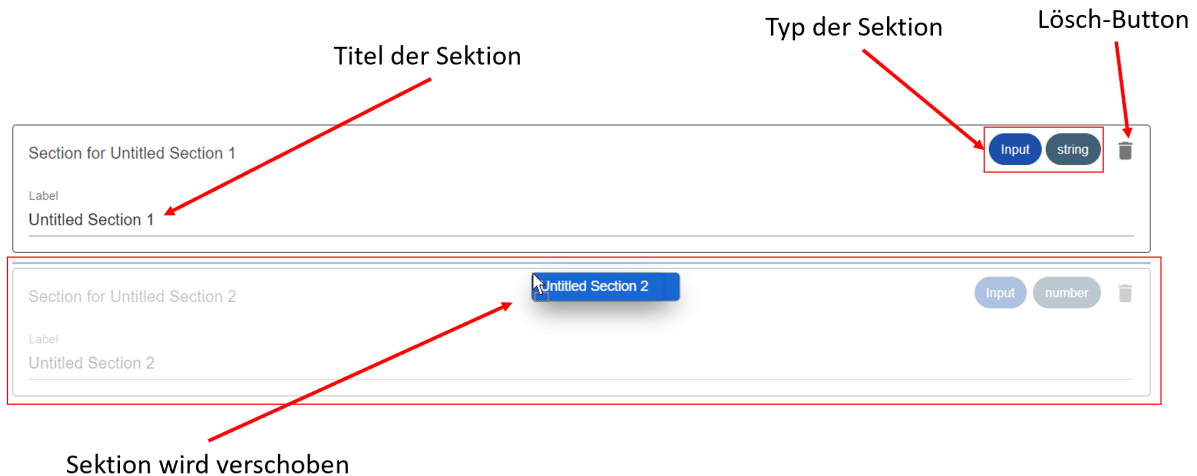


Abbildung 6.10: Veränderung der Reihenfolge von Feldern in einem Template

Damit soll dem Benutzer eine einfache Individualisierung ermöglicht werden. Außerdem besitzen alle Sektionen einen Lösch-Button und eine Typen, mit dem die verschiedenen Textfelder unterschieden bzw. wieder entfernt werden. Die Typen von Feldern sind oben rechts in Abbildung 6.10 zusehen.

Nutzung von Templates

Sind Templates erstellt können diese wie folgt bei der Erstellung einer Notiz ausgewählt werden:

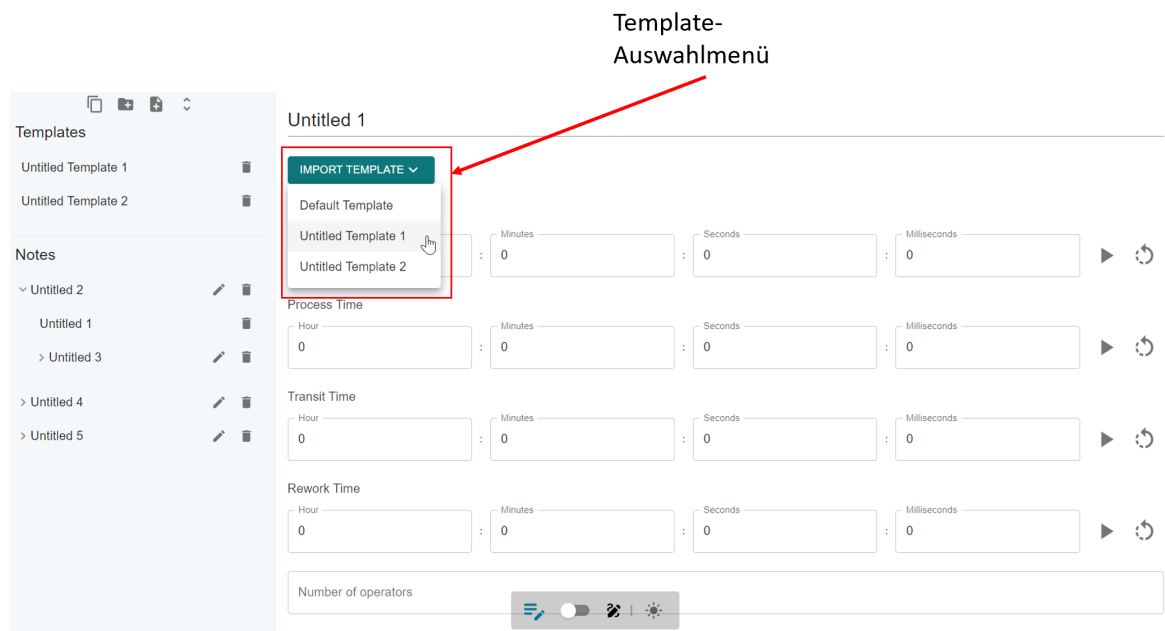


Abbildung 6.11: Das Auswahlmenü von Templates in einer Notiz

Das Template-Menü ist ausschließlich über eine ausgewählte Notiz erreichbar. Es ist in Abbildung 6.11 unter dem Button „IMPORT TEMPLATE“ zusehen. Dieses Menü enthält standardmäßig ein vorgefertigtes Templates, das automatisch zum Einsatz kommt, sobald eine neue Notiz erstellt wird. Zusätzlich listet es alle erstellten Templates auf.

Bei einer Anpassung von Templates bzw. Löschung geschieht das unabhängig von der Notiz. Wird ein Template ausgewählt, werden bereits ausgefüllte Textfelder entweder ersetzt durch ein leeres Feld oder für Zahlenfelder mit einer „0“ ersetzt. Der alte Zustand geht verloren.

Importierung von Notizen in den Wertstrom

Sind alle Notizen erstellt und strukturiert worden, können diese in die Wertstromdarstellung im Editor-Modus importiert werden. Die Struktur im Notizenmodus wird beim Importieren von Notizen beibehalten und sieht wie folgt aus:

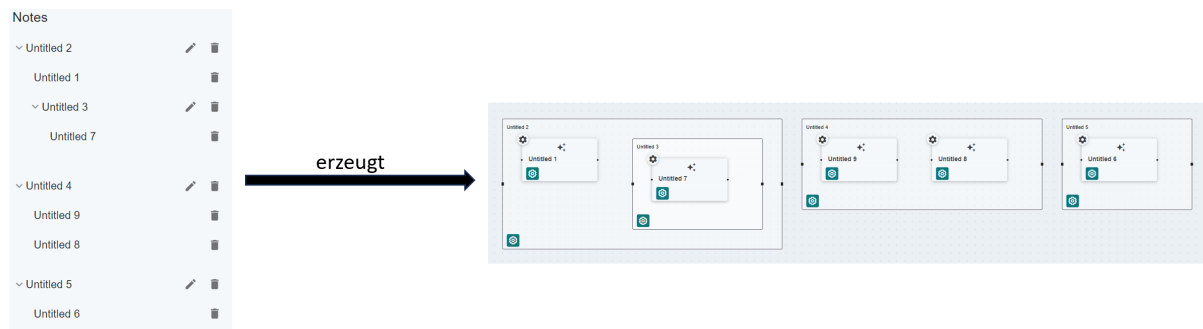


Abbildung 6.12: Import der Notizen im Editor

Die Struktur, die im Editor automatisch generiert wurde kann durch Drag & Drop angepasst werden. Dadurch ist es möglich, schnell Wertströme aus Notizen zu erstellen und anzupassen. Die Notizen können erneut importiert werden, jedoch bleiben die zuvor importierten Notizen vorhanden. Dadurch ist es möglich neue Informationen hinzuzufügen, ohne die existierenden Daten zu verlieren.

Die ausgefüllten Prozessdaten sind im Konfigurator zu finden und können dort bearbeiten bzw. erweitert werden. Der Konfigurator ist durch den Konfigurationsbutton mit dem Zahnradsymbol in den Prozesskarten zu erreichen.

6.3.5 Konfiguration von Prozesskarten

Alle Anpassungen von Prozessdaten und ergänzenden Informationen eines Prozesses erfolgen im Konfigurator. Dieser wird als Dialogfenster angeboten, um den Kontext in der Wertstromvisualisierung beizubehalten. Das Dialogfenster kann durch den Konfigurationsbutton mit dem Zahnrad als Icon auf der Prozesskarte bzw. -gruppe geöffnet werden.

Wird der Button auf der Prozesskarte und -gruppe gedrückt, befüllt sich der Konfigurator mit den Informationen der zugehörigen Prozesskarte und -gruppe. Der Konfigurator sieht wie folgt aus:

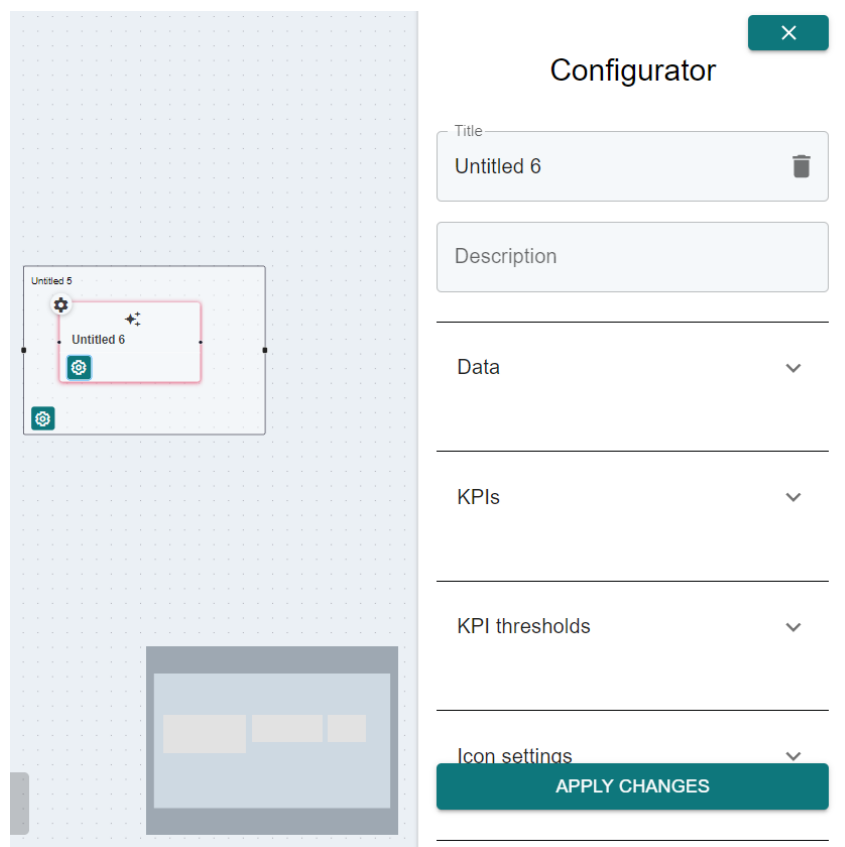


Abbildung 6.13: Anpassung von Informationen im Konfigurator

Die Abbildung 6.13 zeigt den Inhalt der Prozesskarte „Untitled 6“. Der Konfigurator ermöglicht es, folgende Daten zu bearbeiten:

- **Titel:** Anpassen des Prozessnamens
- **Beschreibung:** Anpassen der Beschreibung
- **Data:** Anpassen bzw. Erweitern von Daten einer Prozesskarte und -gruppe
- **KPIs:** Anbinden von KPI-Funktionen oder die Eingabe von statischen KPIs
- **Icons:** Anpassen von dem Haupt- und Subicon

Weitere Features zur Validierung wurden integriert, die jedoch nicht implementiert wurden. Diese dienen dafür, einen Ausblick auf zukünftige Erweiterungen und Verbesserungen zu geben. Diese sind:

- **KPI Thresholds:** Angabe von Schwellenwerten für KPIs die konfiguriert wurden
- **Target Values:** Angabe von Zielwerten für bestimmte Kennzahlen
- **Shift Plan:** Integration von Schichtplänen
- **Setup Effort Matrix:** Automatische Bewertung der Aufwandsmatrix zur Einrichtung und Anpassung von Prozessen

Der Konfigurator steigert die Effizienz und Effektivität in der Prozessverwaltung bzw. -optimierung durch das einfache Verwalten in einem Dialogfenster.

Daten-Anbindung

Die Daten-Anbindung ist unter „Data“ im Konfigurator zu finden. Diese Sektion sieht wie folgt aus:



Abbildung 6.14: Daten-Anbindungssektion im Konfigurator

In diesem Menü können Daten verwaltet und erweitert werden. Jedes Datenfeld besteht aus zwei Teilen: einem Textfeld für den Titel des Datenfeldes und einem zweiten Textfeld für den Wert. Durch diese Strukturierung soll eine geordnete Informationserfassung von Daten ermöglicht werden. Die Daten können durch den „ADD DATA“ Button erweitert werden oder durch den Lösch-Button, der in jeder Sektion enthalten ist, entfernt werden. Somit kann der Benutzer seine Daten flexibel verwalten. Der Titel des Datenfeldes ist ein Pflichtfeld, welches ausgefüllt werden muss.

Sind Daten erstellt bzw. vom Notizenmodus übernommen worden, könne diese mit den KPI-Funktionen verbunden werden. Die Datenfelder dienen als Eingabe für die KPI-Funktionen.

Die Flexibilität der Daten-Anbindung ermöglicht es, dass die Daten aktualisiert werden können. KPIs können kontinuierlich angepasst werden, sodass die Verbesserungen durch die Optimierungsmaßnahmen beobachtet bzw. sichtbar gemacht werden können.

KPI-Anbindung

Die KPI-Anbindung ist unter „KPIs“ im Konfigurator zu finden. Dieses Menü sieht wie folgt aus:

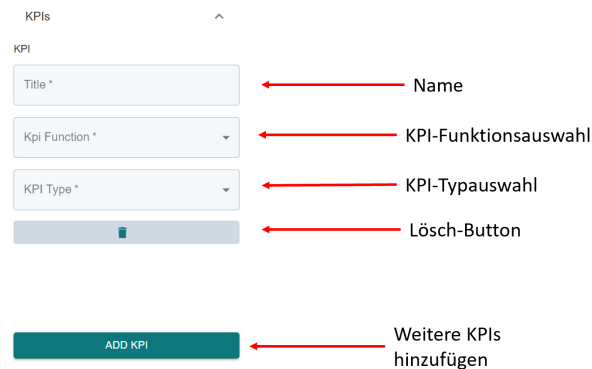


Abbildung 6.15: KPI-Anbindungssektion im Konfigurator

In diesem Menü können KPIs verwaltet werden. Durch die Anbindung von KPI-Funktionen soll es ermöglicht werden, dynamisch KPIs zu berechnen. Für die Bachelorarbeit wurden die Funktionen gemockt und Werte zufällig generiert. Es gibt vier verschiedene KPIs: CO2-Footprint, Production Cost, Quantity und Static KPI. Die erste Funktion zeigt die Funktionalität der Parameterübergabe. Die zweite und dritte Funktion sind parameterlos. Die letzte Funktion ermöglicht es, statische Zahlen als KPI anzuzeigen.

Die dynamische Berechnung der KPIs bietet die Möglichkeit der Einbeziehung von Echtzeitdaten. Dadurch kann die Relevanz der Messungen gesteigert werden. Damit soll es ermöglicht werden, dass eine fundiertere Entscheidung bei den Prozessoptimierungsmaßnahmen getroffen werden kann.

Zuletzt gibt es den Typen für eine KPI-Anbindung. Diese Unterscheidung ermöglicht es zwischen detaillierten und übersichtlichen Darstellungsform zu wählen. Es gibt folgende Typen die ausgewählt werden können: Node und Chart. Ist der Typ Node ausgewählt wird die KPI auf der Prozesskarte bzw. -gruppe mit Titel und Wert angezeigt. Wird der Typ Chart ausgewählt wird ein zusätzlicher Button auf der Prozesskarte bzw. -gruppe sichtbar. Dieser Button öffnet ein Popup-Fenster, welches sich seitlich öffnet und Werte graphisch visualisieren kann.

Icon-Picker

Die Icons können im Konfigurator in der Sektion „Icon settings“ angepasst werden. Die Sektion für Icons sieht wie folgt aus:

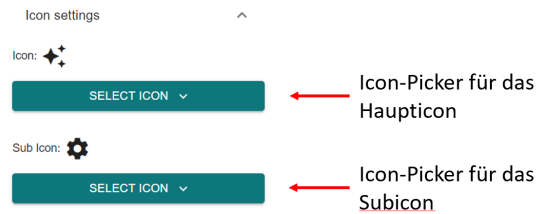


Abbildung 6.16: Icon-Picker Sektion im Konfigurator

In diesem Menü können Haupt- und Subicon angepasst werden. Die Auswahl an Icons geschieht in einem sogenannten Icon-Picker, der ein Popup-Fenster ist. Der Icon-Picker sieht wie folgt aus:

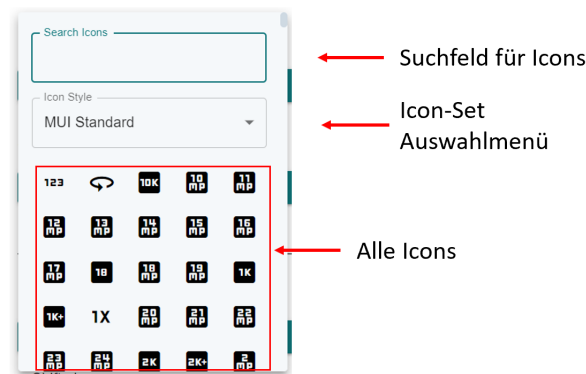


Abbildung 6.17: Icon-Picker geöffnet im Konfigurator

Der Icon-Picker bietet eine breite Palette an Icons an. Es gibt eine Auswahl von mehr als 2100 Icons. Diese Icons können dazu genutzt werden Prozesse visuell zu kennzeichnen. Damit soll die Übersichtlichkeit im Wertstrom bzw. Editor verbessert werden.

Das geöffnete Menü, wie in der Abbildung 6.17 zu sehen ist, enthält ein Suchfeld, welches die Suche von Icons vereinfacht. Wird die Suchfunktionalität genutzt, wird nur im ausgewählten Icon-Set gesucht.

Es besteht die Möglichkeit, zwischen folgenden Icons Sets auszuwählen: MUI-Standard, MUI-Sharp, MUI-Outlined, MUI-Rounded, MUI-Two-Tone [MUI, 2023a] und SIEMENS-Icons [SIEMENS, 2023]. Die Auswahl erfolgt im Icon-Style-Dropdown-Menü. Die Auswahl an Icons kann jederzeit erweitert werden.

6.4 Pipeline Konfiguration

Die Continuous Integration and Continuous Delivery/Continuous Deployment Pipeline in GitLab ermöglicht die Automatisierung von Build-, Test- und Deployment-

Prozessen. Dies sorgt für eine effiziente und konsistente Bereitstellung der Software.

6.4.1 Hintergrund der Pipeline

Das Deployment der Anwendung soll auf dem AWS-Dev Account der evosoft GmbH geschehen. Dieses Deployment wurde während der Umsetzung für die interen Abstimmung zwischen den Stakeholdern genutzt. Die Implementierung der CI/CD-Pipeline ermöglicht Folgendes:

- **Automatisierung:** Beschleunigt den Release-Zyklus durch schnellere und sichere Bereitstellung von Updates und Fixes
- **Konsistenz:** Stellt Qualität und Funktionalität durch automatisierte Testausführung sicher
- **Schnellere Feedback-Schleifen:** Vereinfacht die Kooperation mit den Stakeholdern

6.4.2 Pipeline-Konfigurationsdatei

Die „gitlab-ci.yml“-Datei ist die Konfigurationsdatei der GitLab CI/CD-Pipeline. Die Konfigurationsdatei sieht wie folgt aus:

```
1  stages :
2    - test
3    - build
4    - zip
5    - deploy
6
7  variables :
8    http_proxy: '$CODE_PROXY '
9    https_proxy: '$CODE_PROXY '
10   no_proxy: '$NO_PROXY '
11
12 test frontend:
13   stage: test
14   image: node:16-alpine3.16
15   before_script:
16     - npm install
17   script:
18     - npm run test
19
20 build frontend:
21   stage: build
22   image: node:16-alpine3.16
23   script:
24     - npm install
25     - npm run build
26   only:
```

```
27     variables:
28       - $CI_COMMIT_BRANCH == "main"
29     artifacts:
30       paths:
31         - dist
32
33
34 zip frontend:
35   stage: zip
36   image: ubuntu:latest
37   before_script:
38     - apt-get update
39     - apt-get install zip -y
40   script:
41     - mkdir zips
42     - cd dist
43     - zip ../zips/dist.zip -r .
44   only:
45     variables:
46       - $CI_COMMIT_BRANCH == "main"
47   artifacts:
48     paths:
49       - zips/*
50
51 deploy frontend:
52   stage: deploy
53   image:
54     name: amazon/aws-cli:2.4.11
55     entrypoint: ['']
56   script:
57     - aws --version
58     - aws s3 sync dist s3://$AWS_S3_BUCKET/ --delete
59     - aws cloudfront create-invalidation --distribution-id
60       $DISTRIBUTION_ID --paths "/*"
61   only:
62     variables:
63       - $CI_COMMIT_BRANCH == "main"
```

Listing 6.6: Konfigurationsbeispiel der GitLab Pipeline, Syntax für die GitLab-Pipeline:
CI/CD YAML syntax reference

Kern der Konfiguration

1. Die Schritte der Pipeline sind:

- **Test:** Testet das Projekt

- **Build:** Erstellt das Projekt
 - **Zip:** Verpackt das Projekt zum Hochladen
 - **Deploy:** Lädt das Front-End auf AWS hoch
2. Variablen: Die Variablen in der Pipeline sind notwendig Kommunikationsaufbau aus GitLab zu externen Services wie z.B. AWS oder Node Package Manager (npm).
 3. Test-Stage:
 - Für die Testumgebung wird ein Node-Image genutzt
 - Nachdem die Testumgebung aufgesetzt wurde, müssen alle Bibliotheken heruntergeladen werden, um alle Tests mit dem Skript „npm run test“ auszuführen.
 4. Build-Stage:
 - Für die Build-Umgebung wird auch eine Node-Image genutzt.
 - Nachdem das Build-Skript ausgeführt wurde, entsteht ein Artefakt. Dieses Artefakt wird in der Deploy-Stage genutzt.
 5. Zip-Stage:
 - Zum Erstellen eines ZIP-Archivs wird die Umgebung von Ubuntu genutzt.
 - Das ZIP-Archiv wird für den Upload in den S3-Bucket benötigt.
 6. Deploy-Stage:
 - Die Deploy-Stage verwendet die Command Line Interface (CLI) von AWS, um die Inhalte des S3-Buckets zu updaten und die CloudFront-Distribution zu invalidieren. Durch die Invalidierung wird sichergestellt, dass Erneuerungen sofort sichtbar sind, indem der Cache des Hostings gelöscht wird.
 7. Bedingungen der Stages: Die Build-, Zip- und Deploy-Stages werde nur ausgeführt, wenn die Änderungen auf dem „main“-Branch vorgenommen wurden. Genauso müssen die Stages nacheinander ausgeführt werden.

Die erwähnten Funktionalitäten des Projekts bieten eine solide Grundlage für die Weiterentwicklung des Projektes. Die Anwendung ermöglicht die schnelle Aufnahme von Prozessdaten, Konvertierung von Notizen zu Prozesskarten bzw. -gruppen im Editor und somit die Erstellung von Wertstromdiagrammen.

Kapitel 7

Zusammenfassung und Ausblick

Die Bachelorarbeit evoSTRIVE – Ein digitaler Ansatz zur Wertstrommodellierung: Integration von Industrie-Standards und Domänenmodellen in einem Web-Editor konzentriert sich auf die Entwicklung eines Wertstromanalysetools in Kooperation mit dem IT-Unternehmen evosoft GmbH. Die Wertstromanalyse stammt ursprünglich aus der Automobilindustrie und zielt darauf ab, die Verschwendungen in den Produktionsprozessen zu identifizieren und zu eliminieren.

Im Fokus stand die Entwicklung einer innovativen Applikation zur Erfassung und Modellierung von Wertströmen. Bei der Umsetzung waren die Methoden des Design Thinkings und die enge Zusammenarbeit mit der Zielgruppe ausschlaggebend. Dadurch konnte eine Lösung entwickelt werden, die speziell auf die Bedürfnisse, Anforderungen und Herausforderungen der Zielgruppe, bestehend aus Wertstromberatern und -spezialisten, zugeschnitten ist.

Die Entwicklung und Implementierung des Editors hat alle bekannten Stakeholder sehr zufrieden gestellt. Die gesetzten Ziele wurden mithilfe der Anwendung realisiert. Das Kernstück der Bachelorarbeit ist der Editor- und Notizen-Modus. Die Modi ermöglichen es, die Berater bei der Vorbereitung auf ein Kundenprojekt bzw. Optimierungsprojekt zu unterstützen. Besonders durch die Trennung der Aufnahme von Prozessdaten und Modellierung eines Wertstromes, konnte ein enormer Effizienzgewinn in der Datenerfassung und -verarbeitung erreicht werden.

Der Notizenmodus ermöglicht es dem Benutzer, den Fokus auf die Aufnahme von Prozessdaten zu legen. Dadurch wird eine Abstraktionsebene zur visuellen Aufnahme geschaffen, was dafür sorgt, dass Informationen effizient aufgenommen werden können. Zudem wird es dem Nutzer vereinfacht, die Gewohnheit mit Stift und Papier zu schreiben, abzulegen, und durch den benutzerfreundlichen Editor zu ersetzen.

Zusätzlich ermöglicht der Notizenmodus das Strukturieren der Notizen und das Erstellen von Templates. Mit den Templates kann der Berater sich effizient auf die Beratung bzw. das Optimierungsprojekt vorbereiten. Dadurch wird die Beratungsqualität und die Produktivität gesteigert. Ist ein Template vorbereitet, kann es in eine Notiz geladen und ausgefüllt werden.

Der Editormodus der Applikation unterstützt beim Erstellen von Wertströmen. Es ist möglich, Prozesskarten und Prozessgruppen zu erstellen. Die Prozessgruppen ermöglichen es, komplexe Strukturen zu modellieren. Jede Prozesskarte bzw. Prozessgruppe kann individuell konfiguriert werden. Es ist möglich, Titel, Beschreibung, Prozessdaten und zusätzliche Informationen zu konfigurieren. Außerdem ist es möglich, KPI-Funktionen zu integrieren und die Anpassung von Prozesskarten-Icons vorzunehmen.

Sind alle nötigen Karten erstellt, ermöglicht der Editor auch die Visualisierung von Materialflüssen, durch die Verbindung von Prozesskarten und -gruppen. Die gesamten Funktionalitäten bieten eine solide Basis zur Weiterentwicklung der Applikation. Die Applikation zeigt die Digitalisierung der Berater bei Kundenbesuchen, die Anpassbarkeit der Visualisierung, die dauerhafte Verfügbarkeit der Wertstromanalysen und die Vermeidung der redundanten Digitalisierung bereits durchgeführter Wertstromanalysen.

Ausblick

Aufgrund der Ergebnisse der Bachelorarbeit, konnten verschiedene weitere Entwicklungsrichtungen identifiziert werden. Eine Möglichkeit wäre es, branchenspezifische Lösungen anzubieten. Damit ist gemeint, dass die Branchen wie Pharma und Gesundheit oder Finanzen in Zukunft auch einen Editor zur Wertstromanalyse einsetzen könnten. Bevor eine branchenspezifische Lösung angeboten werden kann, müssen zunächst die Anforderungen angepasst werden. Besonders kritische Änderungen könnten die Sicherheitsstandards, branchenspezifische Regularien und Audit Trails sein.

Ein nächster Schritt könnte die Ermöglichung einer Nutzerkollaboration sein. Diese Funktion befähigt Teams gleichzeitig online an einem Wertstrom zu arbeiten, Feedback zu geben und Änderungen zu diskutieren.

Außerdem besteht die Möglichkeit, die Applikation mit maschinellem Lernen und künstlicher Intelligenz zu erweitern. Diese Technologien können dazu genutzt werden, die Berater bei ihren Analysen zu unterstützen, in dem sie Vorschläge zur Verbesserung von Wertströmen anbieten würden. Ein Beispiel für einen Anwendungsfall könnte sein, dass eine Mustererkennung in den Wertstromdaten erfolgt und darauf basierend Optimierungsvorschläge generiert werden können. Auch könnten historische Daten und aktuelle Prozessmetriken dazu beitragen.

Zuletzt könnte die Internationalisierung bzw. Lokalisierung der Applikation interessant sein. Dadurch wäre es möglich, internationale Kunden zu gewinnen und für die Applikation zu begeistern.

Abkürzungsverzeichnis

API Application Programming Interface

AWS Amazon Web Services

CI/CD Continuous Integration and Continuous Delivery/Continuous Deployment

CLI Command Line Interface

CRUD Create Read Update Delete

HATEOAS Hypermedia as the Engine of Application State

HTTP Hypertext Transfer Protocol

JSON JavaScript Object Notation

JS JavaScript

JSX JavaScript XML

KPI Key Performance Indicator

MUI Material UI

MVP Minimum Viable Product

npm Node Package Manager

OpEx Operational Excellence

QA Quality Assurance

REST Representational State Transfer

SPA Single Page Application

UI User Interface

UX User Experience

UX/UI User Experience/User Interface

VSA Value Stream Assistant

WSA Wertstromanalyse

XML Extensible Markup Language

Abbildungsverzeichnis

| | | |
|------|---|----|
| 1.1 | Wertstromanalyse auf Brown Paper | 2 |
| 2.1 | Ein Beispiel einer Wertstromanalyse | 4 |
| 3.1 | Brainstorming der Zielgruppen für die Applikation | 7 |
| 3.2 | Beispiel für die Ausgestaltung einer Anforderung nach der Satzschablone | 13 |
| 4.1 | Visualisierung des Workflows von GitFlow | 21 |
| 4.2 | Visualisierung des eignen Workflows | 22 |
| 5.1 | Visualisierung des Codebeispiels | 26 |
| 5.2 | Visualisierung des Codebeispiels aus dem Listing 5.2 | 28 |
| 5.3 | Zugriff auf die globale Zustandsverwaltung mit Redux | 29 |
| 5.4 | Visualisierung der Zustandveränderung in der Applikation mit Redux . | 30 |
| 6.1 | Seitenleiste zur Erstellung von verschiedenen Prozesskarten | 35 |
| 6.2 | Ein Beispiel einer vorkonfigurierten Prozesskarte | 36 |
| 6.3 | Das Aussehen einer Prozessgruppe | 37 |
| 6.4 | Anpassung der Größe einer Prozessgruppe | 37 |
| 6.5 | Gruppierung von Elementen zur Erschaffung von Detailebenen | 39 |
| 6.6 | Applikationssteuerung | 44 |
| 6.7 | Die Ansicht im Notizenmodus | 45 |
| 6.8 | Ein Beispiel an Notizenfeldern | 46 |
| 6.9 | Feldtypen die in einem Template erstellt werden können | 47 |
| 6.10 | Veränderung der Reihenfolge von Feldern in einem Template | 47 |
| 6.11 | Das Auswahlmenü von Templates in einer Notiz | 48 |
| 6.12 | Import der Notizen im Editor | 49 |
| 6.13 | Anpassung von Informationen im Konfigurator | 50 |
| 6.14 | Daten-Anbindungssektion im Konfigurator | 51 |
| 6.15 | KPI-Anbindungssektion im Konfigurator | 52 |
| 6.16 | Icon-Picker Sektion im Konfigurator | 53 |
| 6.17 | Icon-Picker geöffnet im Konfigurator | 53 |
| A.1 | Die Ansicht des Editors | 67 |
| A.2 | Die Ansicht des Editors im Darkmode | 68 |
| A.3 | Das Erstellen einer Prozesskarte | 68 |
| A.4 | Das Verändern der Größe einer Prozessgruppe | 69 |

| | | |
|------|--|----|
| A.5 | Das Öffnen des Konfigurators durch eine Prozesskarte | 69 |
| A.6 | Die Ansicht des Icon-Pickers geöffnet | 70 |
| A.7 | Die Ansicht beim öffnen des Notizenmoduses | 71 |
| A.8 | Das Selektieren einer Notiz zum bearbeiten des Inhaltes | 71 |
| A.9 | Die Reihenfolge der angelegten Notizen ändern | 72 |
| A.10 | Verschieben einer Notiz in eine Gruppe | 73 |
| A.11 | Das Umbenennen einer Gruppe | 74 |
| A.12 | Das Erstellen eines Templates | 75 |
| A.13 | Ein bereits konfigurierte Template | 75 |
| A.14 | Die Reihenfolge im Template anpassen | 76 |
| A.15 | Das Auswählen eines Templates in einer Notiz | 76 |
| A.16 | Die Mehrfachauswahl im Editor | 77 |
| A.17 | Die Gruppierung von Karten durch Drag & Drop | 77 |
| A.18 | Die Ansicht von gruppierten Karten | 78 |
| A.19 | Das Verbinden von Karten zu Visualisierung von Materialflüssen | 78 |
| A.20 | Die Ansicht der importierten Notizen im Editor | 79 |

Tabellenverzeichnis

| | | |
|-----|--|----|
| 3.1 | Persona von Bernd Bauer | 9 |
| 3.2 | Persona von Otto Oster | 10 |
| 3.3 | User-Stories | 17 |
| 4.1 | API Endpunkte und CRUD-Operationen | 20 |
| 6.1 | Ordnerstruktur des MVPs für die Bachelorarbeit | 33 |

Literaturverzeichnis

- [Abramov and et al., 2023a] Abramov, D. and et al. (2023a). Prior Art | Redux. <https://redux.js.org/understanding/history-and-design/prior-art>.
- [Abramov and et al., 2023b] Abramov, D. and et al. (2023b). Redux - A predictable state container for JavaScript apps. | Redux. <https://redux.js.org/>.
- [Bahamdain, 2015] Bahamdain, S. S. (2015). Open Source Software (OSS) Quality Assurance: A Survey Paper. *Procedia Computer Science*, 56:459–464.
- [Bass et al., 2021] Bass, L., Clements, P., and Kazman, R. (2021). Software Architecture in Practice, 4th Edition. <https://learning.oreilly.com/library/view/software-architecture-in/9780136885979/>.
- [Black, 2020] Black, N. (2020). Boris Cherny on TypeScript. *IEEE Software*, 37(2):98–100.
- [Brown and Katz, 2011] Brown, T. and Katz, B. (2011). Change by design. *Journal of Product Innovation Management*, 28(3).
- [Bryan, 2017] Bryan, J. (2017). *Excuse Me, Do You Have a Moment to Talk about Version Control?* PeerJ Preprints.
- [Bugl, 2017] Bugl, D. (2017). *Learning Redux*. Packt Publishing Ltd.
- [Carnegie Mellon University, 2023] Carnegie Mellon University (2023). Software Architecture | Software Engineering Institute. <https://www.sei.cmu.edu/our-work/software-architecture/index.cfm>.
- [Cooper et al., 2014] Cooper, A., Reimann, R., Cronin, D., and Cooper, A. (2014). *About Face: The Essentials of Interaction Design*. John Wiley and Sons, Indianapolis, IN, fourth edition edition.
- [Delgado et al., 2016] Delgado, A., Estepa, A., Troyano, J. A., and Estepa, R. (2016). Reusing UI elements with Model-Based User Interface Development. *International Journal of Human-Computer Studies*, 86:48–62.
- [Eckert, 2017] Eckert, R. (2017). *Lean Startup in Konzernen und Mittelstandsunternehmen: Ergebnisse einer Expertenbefragung und Handlungsempfehlungen*. essentials. Springer Fachmedien, Wiesbaden.
- [Ecma International, 2023] Ecma International (2023). ECMAScript® 2024 Language Specification. <https://tc39.es/ecma262/>.

- [Fielding, 2000] Fielding, R. T. (2000). *In Information and Computer Science*. PhD thesis, UNIVERSITY OF CALIFORNIA, IRVINE.
- [Geewax, 2021] Geewax, J. J. J. (2021). *API Design Patterns*. <https://learning.oreilly.com/library/view/api-design-patterns/9781617295850/>.
- [GitLab, 2023] GitLab (2023). *Get started with GitLab CI/CD* | GitLab. <https://docs.gitlab.com/ee/ci/>.
- [Goodman et al., 2012] Goodman, E., Kuniavsky, M., and Moed, A. (2012). Chapter 6 - universal tools: Recruiting and interviewing. In Goodman, E., Kuniavsky, M., and Moed, A., editors, *Observing the User Experience (Second Edition)*, pages 95–139. Morgan Kaufmann, Boston, second edition edition.
- [Gorman, 2022] Gorman, B. L. (2022). *Unit Testing, Integration Testing, and Mocking*. In Gorman, B. L., editor, *Practical Entity Framework Core 6: Database Access for Enterprise Applications*, pages 575–614. Apress, Berkeley, CA.
- [Grömling, 2023] Grömling, M. (2023). *Hartnäckige Produktionslücken der deutschen Industrie*. Research Report 41/2023, IW-Report.
- [Gunaki et al., 2022] Gunaki, P., Devaraj, S., and Patil, S. (2022). *Process optimization by value Stream Mapping*. *Materials Today: Proceedings*, 54:251–254.
- [Horsthofer-Rauch et al., 2022] Horsthofer-Rauch, J., Schumann, M., Milde, M., Ver-nim, S., and Reinhart, G. (2022). *Digitalized value stream mapping: Review and outlook*. *Procedia CIRP*, 112:244–249.
- [Khalid et al., 2014] Khalid, N. K. M., Hashim, A. Y. B., and Salleh, M. R. (2014). *On Value Stream Mapping and Its Industrial Significance*. *Journal of Industrial and Intelligent Information*, 2(2):88–90.
- [Meta, 2023a] Meta (2023a). *File Structure – React*. <https://legacy.reactjs.org/docs/faq-structure.html>.
- [Meta, 2023b] Meta (2023b). *React*. <https://react.dev/>.
- [Moser, 2012] Moser, C. (2012). *User Experience Design: Mit erlebniszentrierter Software-entwicklung zu Produkten, die begeistern*. X.media.press. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [MUI, 2023a] MUI (2023a). *Material Icons - Material UI*. <https://mui.com/material-ui/material-icons/>.
- [MUI, 2023b] MUI (2023b). *MUI: The React component library you always wanted*. <https://mui.com/>.
- [Naik and Tripathy, 2011] Naik, K. and Tripathy, P. (2011). *Software Testing and Quality Assurance: Theory and Practice*. John Wiley & Sons.
- [Ohno, 1988] Ohno, T. (1988). *Toyota Production System: Beyond Large-Scale Production*. CRC Press.

- [Rahani and al-Ashraf, 2012] Rahani, A. R. and al-Ashraf, M. (2012). Production Flow Analysis through Value Stream Mapping: A Lean Manufacturing Process Case Study. *Procedia Engineering*, 41:1727–1734.
- [Reichert et al., 2018] Reichert, D., Cito, C., and Barjasic, I. (2018). *Lean & Green: Best Practice*. Springer Fachmedien Wiesbaden, Wiesbaden.
- [Rother and Shook, 2003] Rother, M. and Shook, J. (2003). *Learning to See: Value Stream Mapping to Add Value and Eliminate Muda*. Lean Enterprise Institute.
- [Schmitz, 2013] Schmitz, P. (2013). *Software-Qualitätssicherung — Testen im Software-Lebenszyklus*. Springer-Verlag.
- [Schoeman et al., 2021] Schoeman, Y., Oberholster, P., and Somerset, V. (2021). Value Stream Mapping as a Supporting Management Tool to Identify the Flow of Industrial Waste: A Case Study. *Sustainability*, 13(1):91.
- [Shahid et al., 2011] Shahid, D. M., Ibrahim, S., and Mahrin, M. (2011). A Study on Test Coverage in Software Testing.
- [Shou et al., 2017] Shou, W., Wang, J., Wu, P., Wang, X., and Chong, H.-Y. (2017). A cross-sector review on the use of value stream mapping. *International Journal of Production Research*, 55:1–23.
- [SIEMENS, 2023] SIEMENS (2023). Icons | Siemens Industrial Experience. <https://ix.siemens.io/docs/icon-library/icons/>.
- [Starke, 2023] Starke, D. G. (2023). Erfolgreiche Softwarearchitektur. <https://www.arc42.de/>.
- [Statista, 2023] Statista (2023). Manufacturing - Worldwide | Statista Market Forecast. <https://www.statista.com/outlook/io/manufacturing/worldwide>.
- [Turner, 2014] Turner, J. (2014). Announcing TypeScript 1.0. <https://devblogs.microsoft.com/typescript/announcing-typescript-1-0/>.
- [Vazquez et al., 2022] Vazquez, H. C., Pace, J. A. D., Marcos, C., and Vidal, S. (2022). Retrieving and Ranking Relevant JavaScript Technologies from Web Repositories. [TLDR] This work proposes a two-phase approach for assisting developers to retrieve and rank JS technologies in a semi-automated fashion, which relies on a machine learning technique to infer, based on criteria used by other projects in the Web, a ranking of the output of ST-Retrieval.
- [xyflow, 2023a] xyflow (2023a). Api Reference – React Flow. <https://reactflow.dev/api-reference>.
- [xyflow, 2023b] xyflow (2023b). Custom Nodes – React Flow. <https://reactflow.dev/examples/nodes/custom-node>.
- [xyflow, 2023c] xyflow (2023c). Node-Based UIs in React – React Flow. <https://reactflow.dev/>.

Anhang A

Abbildungen

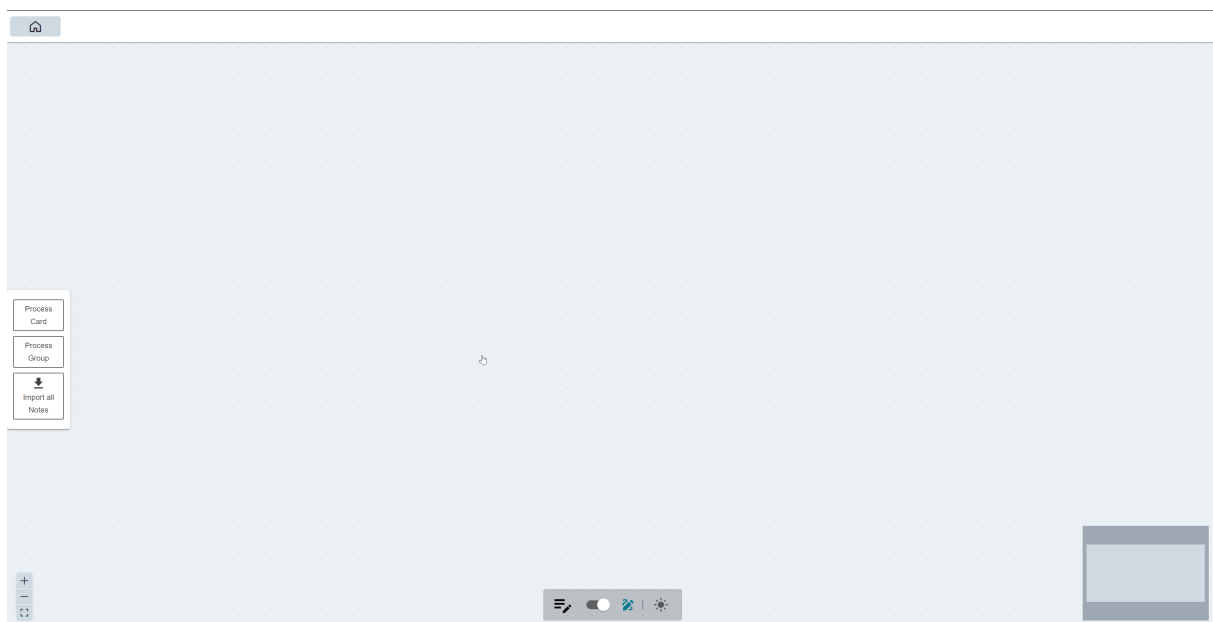


Abbildung A.1: Die Ansicht des Editors

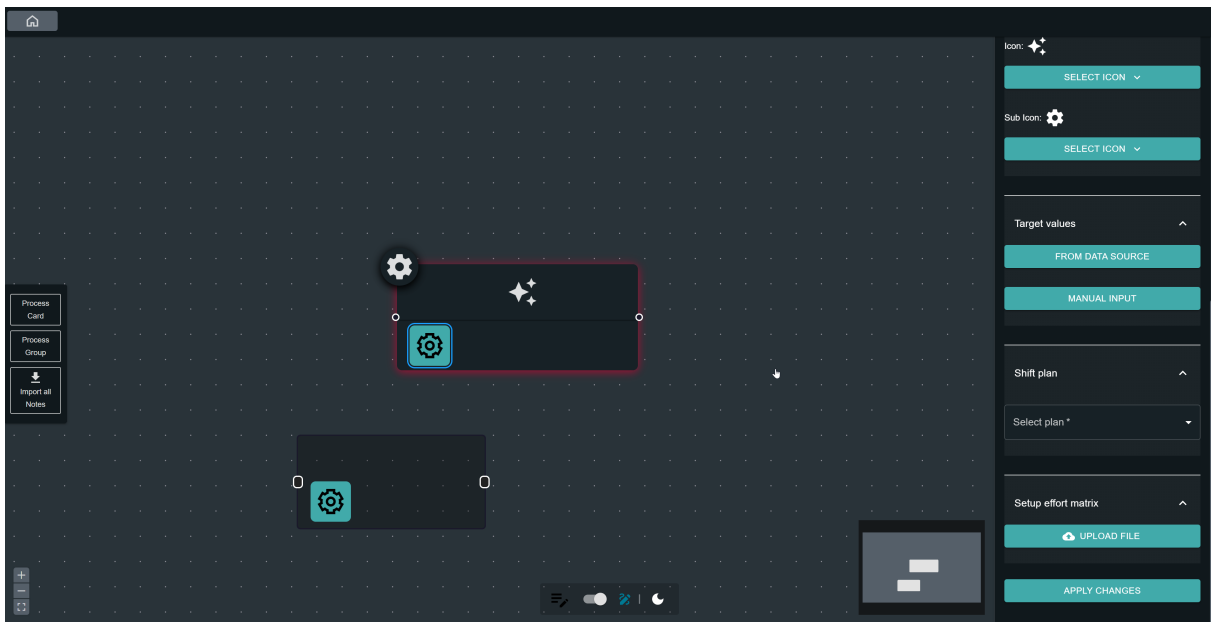


Abbildung A.2: Die Ansicht des Editors im Darkmode

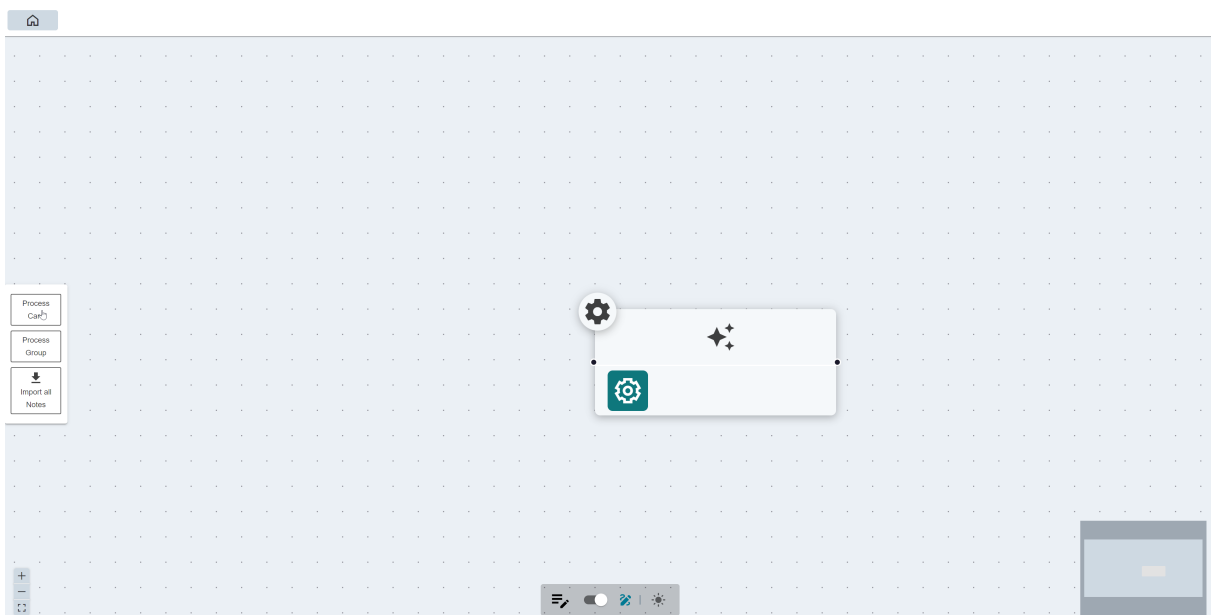


Abbildung A.3: Das Erstellen einer Prozesskarte

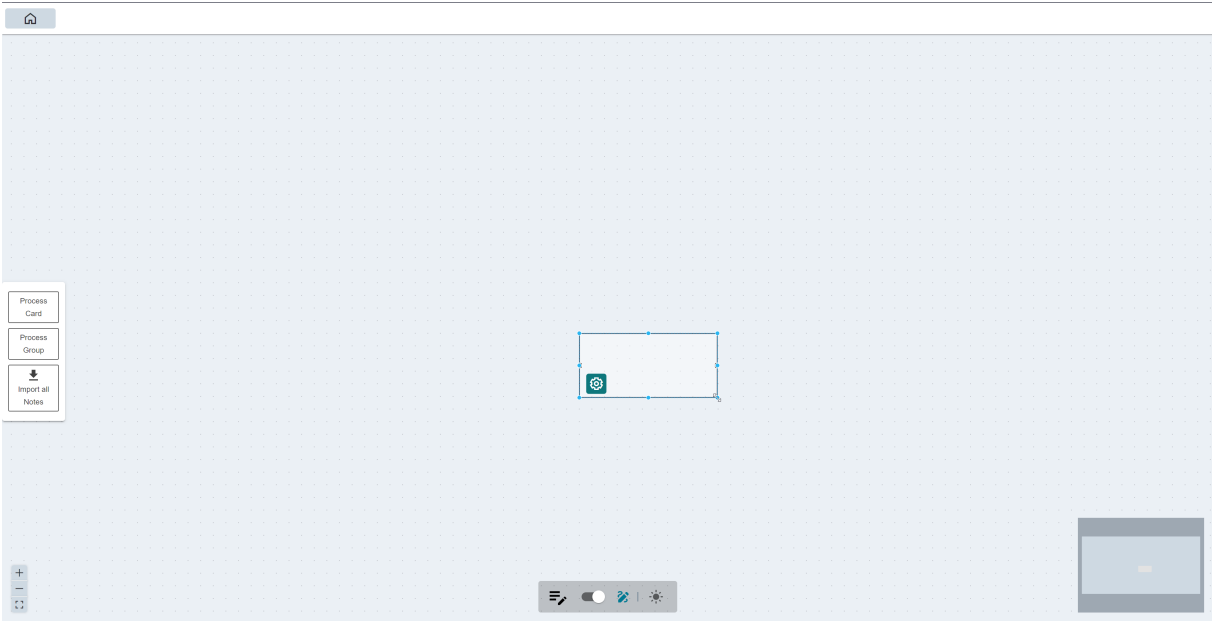


Abbildung A.4: Das Verändern der Größe einer Prozessgruppe

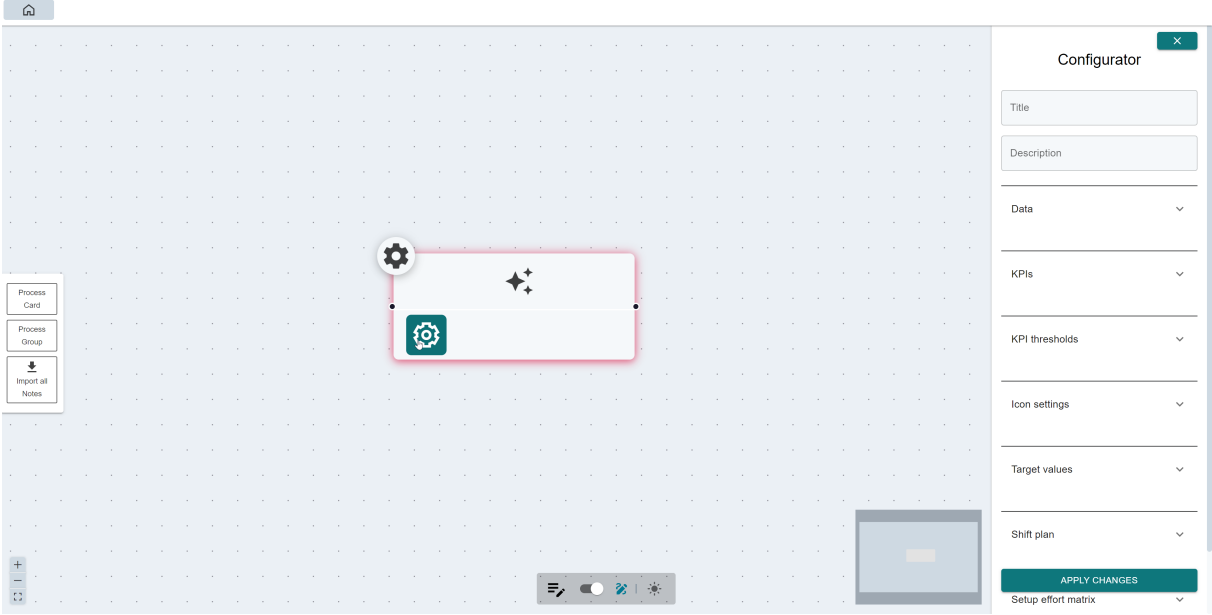


Abbildung A.5: Das Öffnen des Konfigurators durch eine Prozesskarte

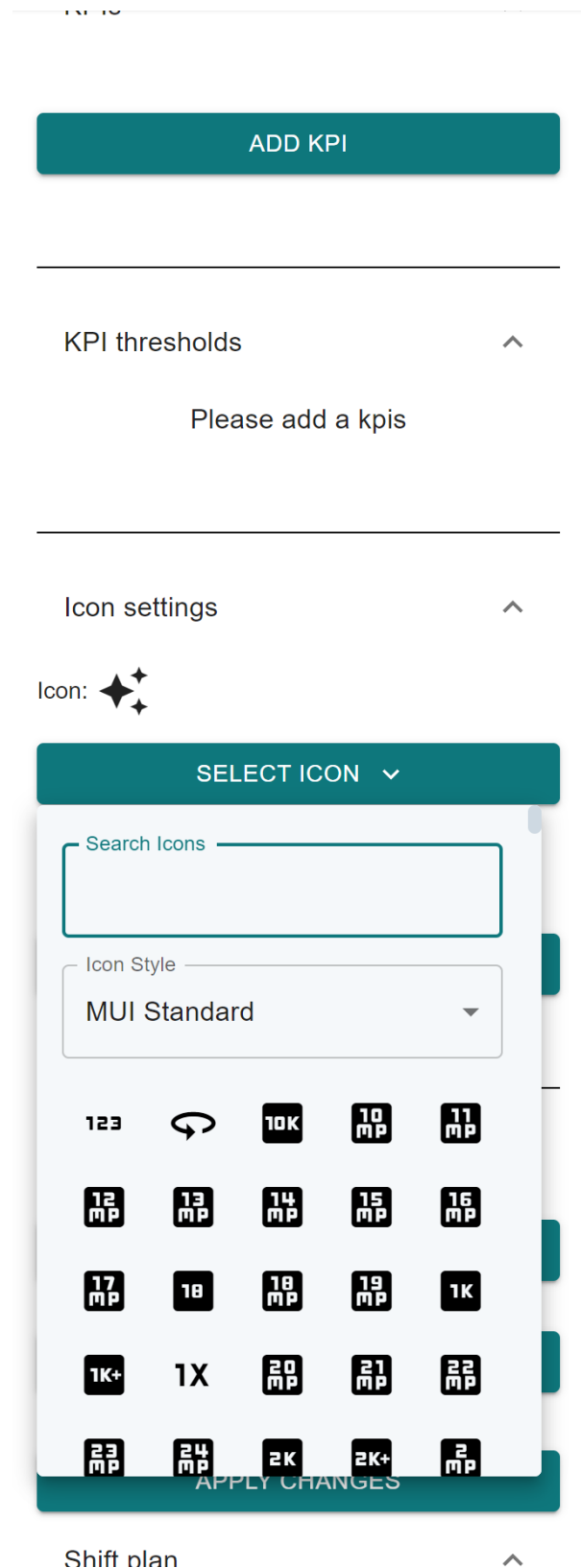


Abbildung A.6: Die Ansicht des Icon-Pickers geöffnet

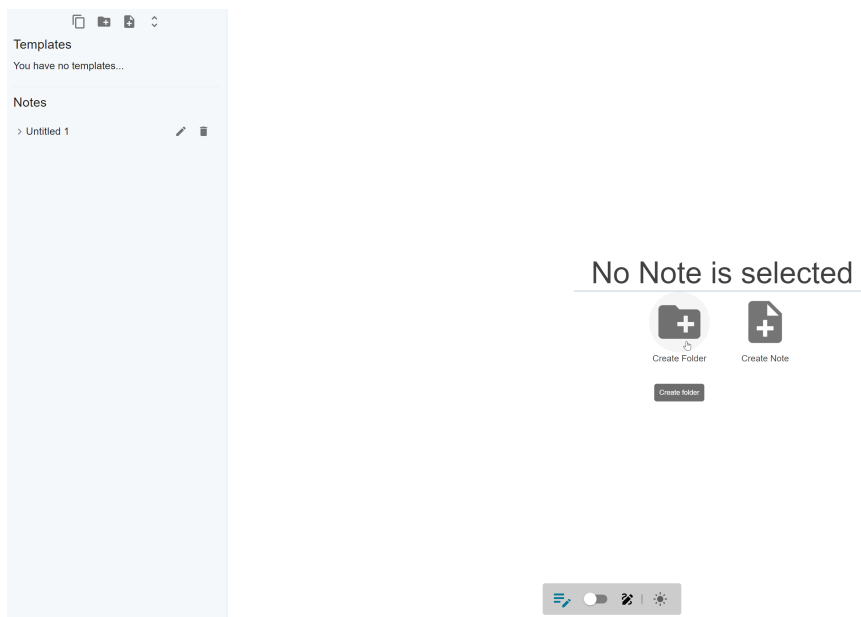


Abbildung A.7: Die Ansicht beim öffnen des Notizenmoduses

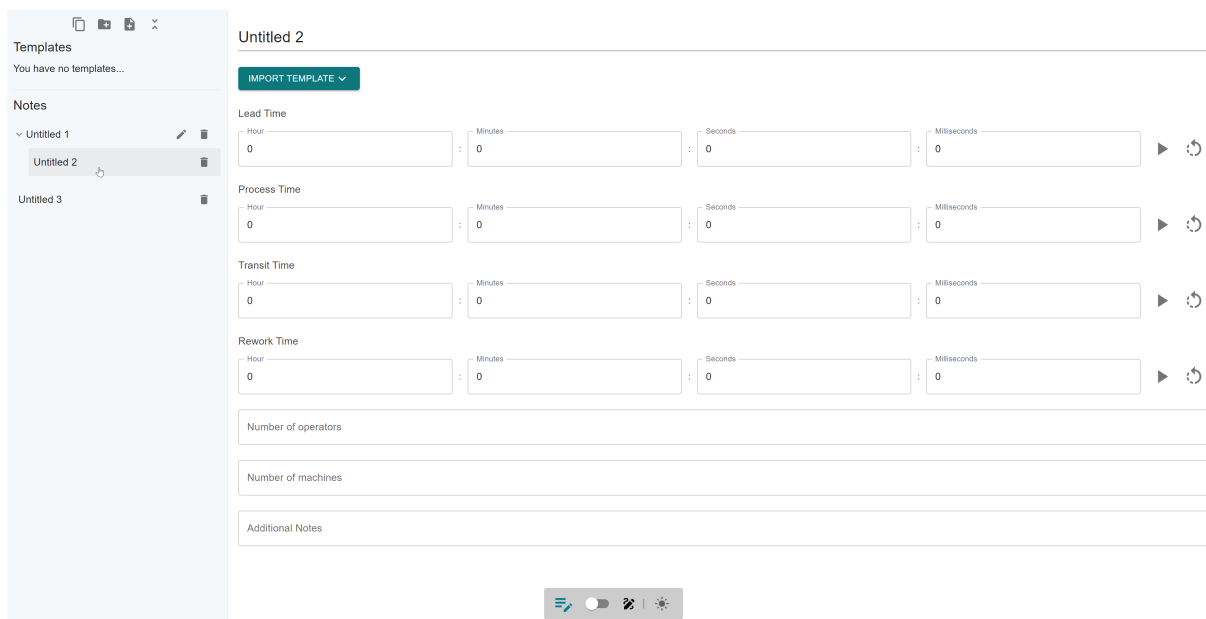


Abbildung A.8: Das Selektieren einer Notiz zum bearbeiten des Inhaltes

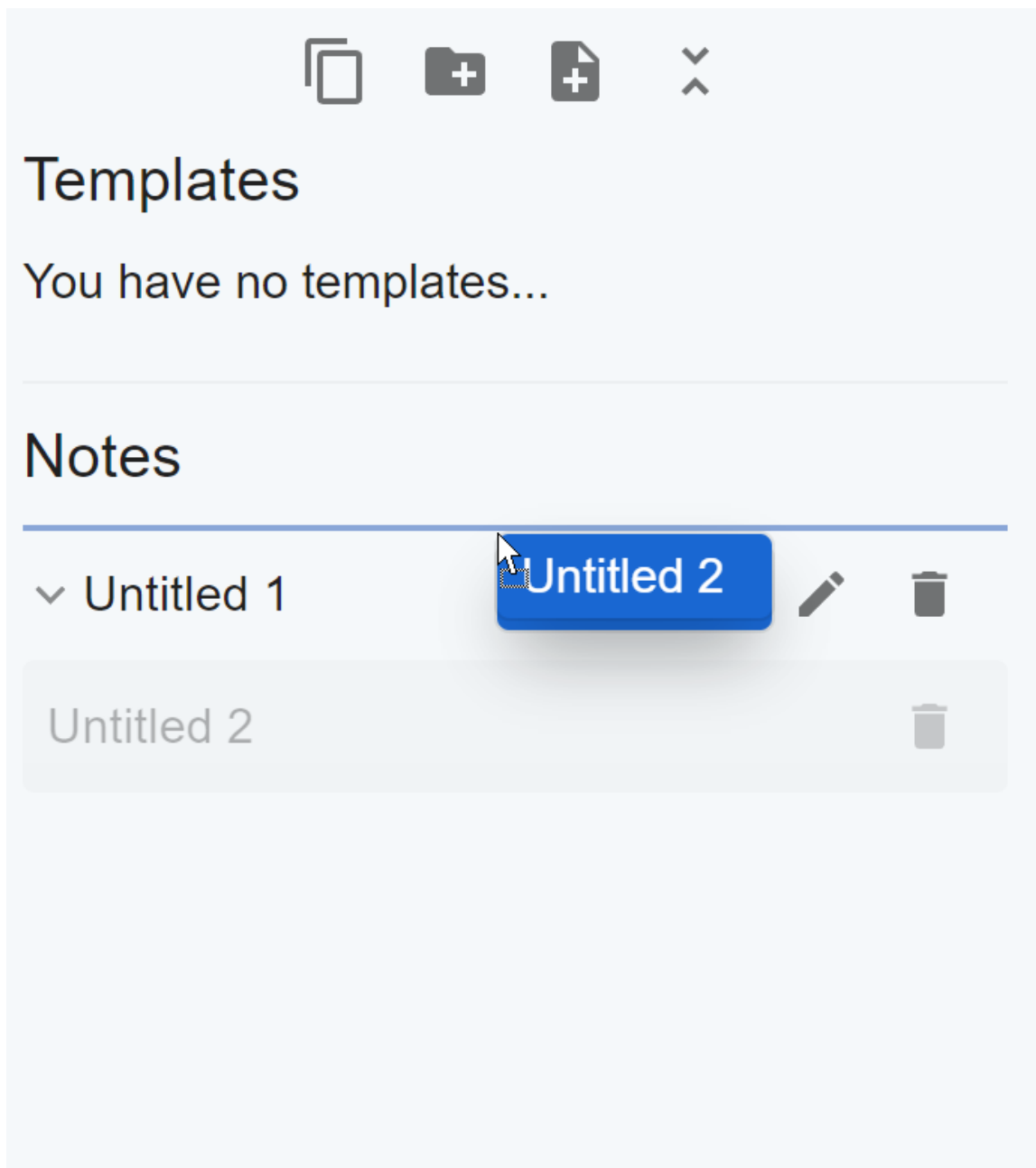


Abbildung A.9: Die Reihenfolge der angelegten Notizen ändern

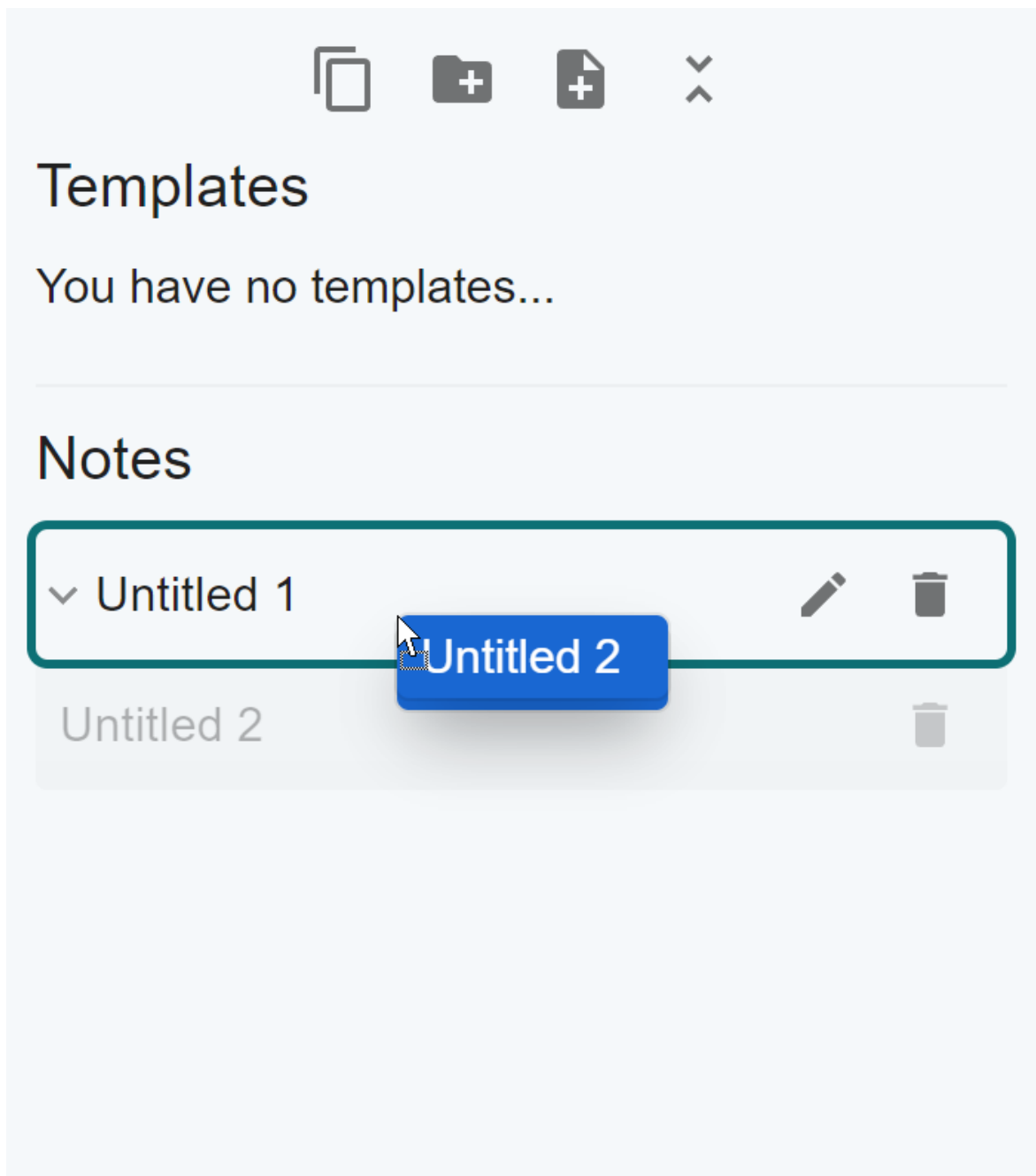


Abbildung A.10: Verschieben einer Notiz in eine Gruppe

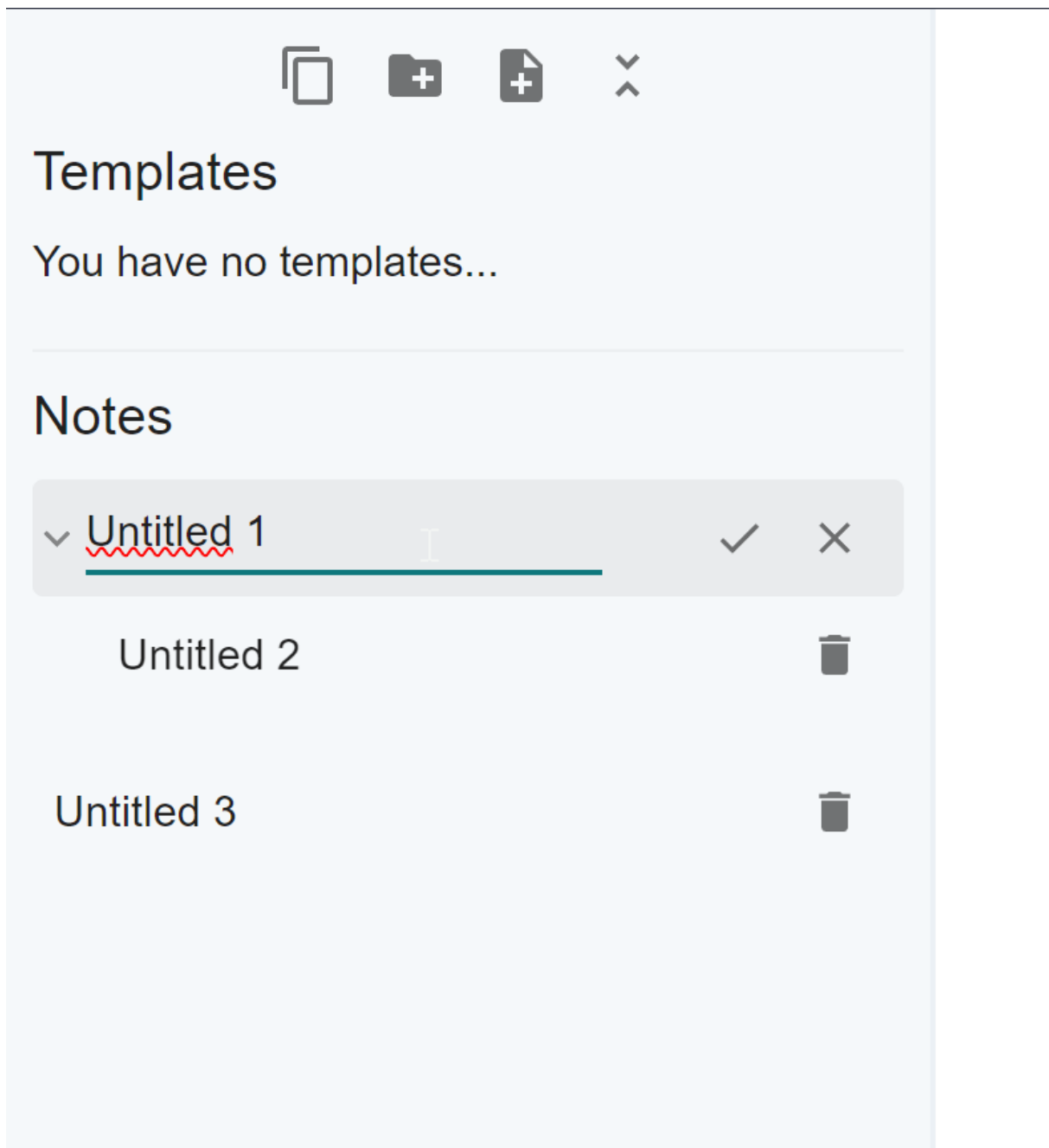


Abbildung A.11: Das Umbenennen einer Gruppe

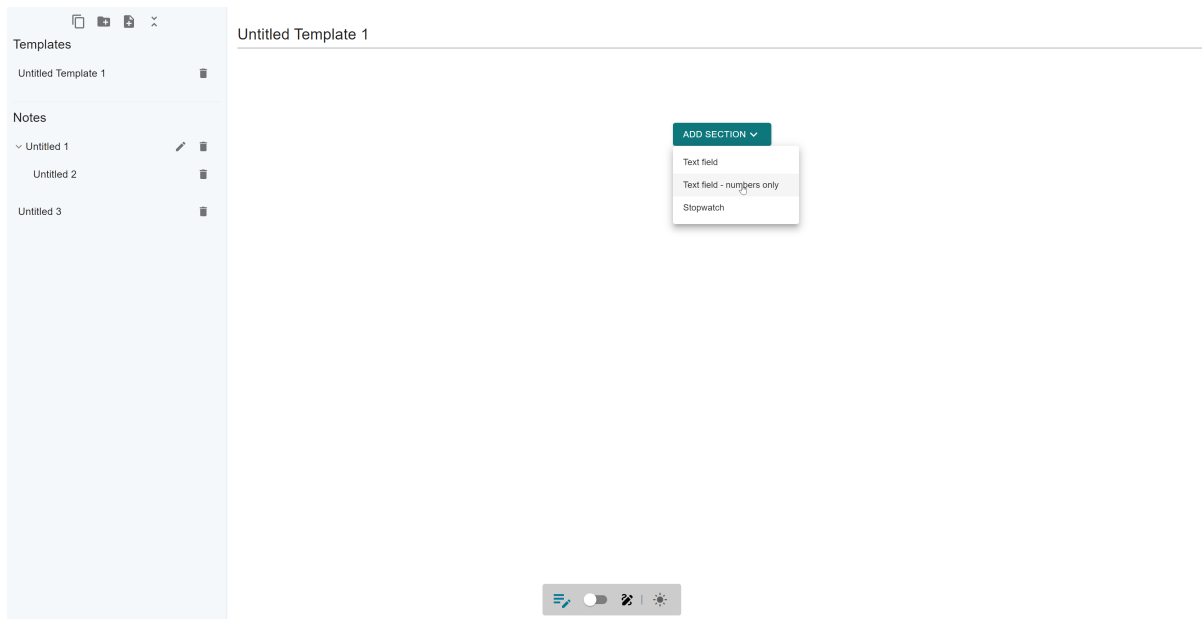


Abbildung A.12: Das Erstellen eines Templates

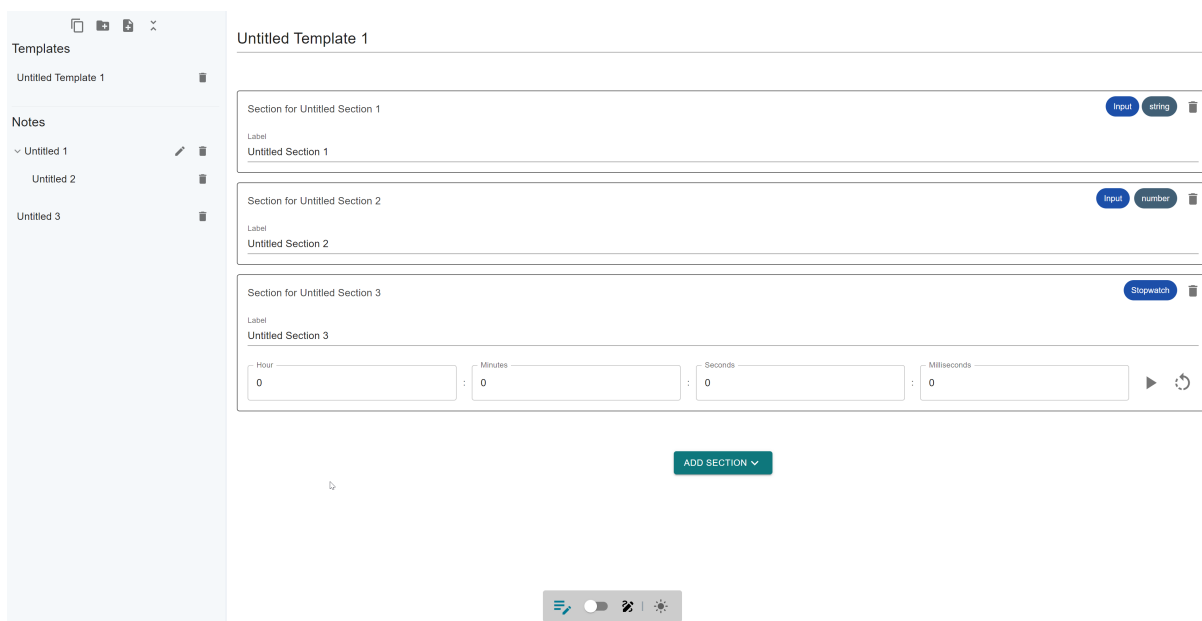


Abbildung A.13: Ein bereits konfigurierte Template

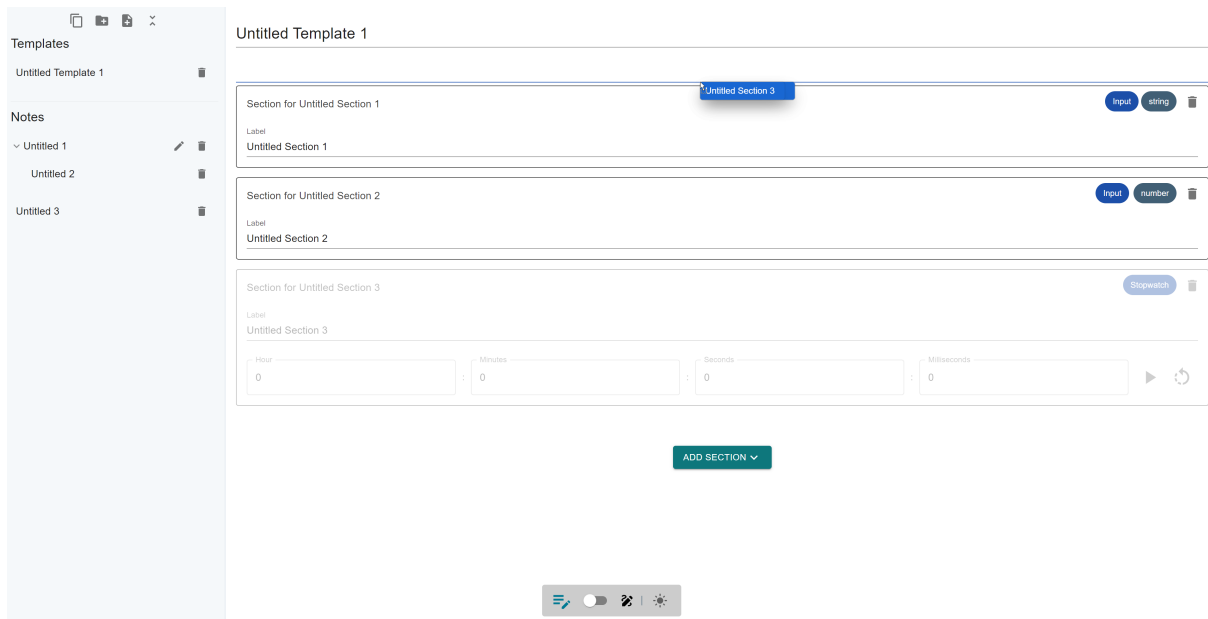


Abbildung A.14: Die Reihenfolge im Template anpassen

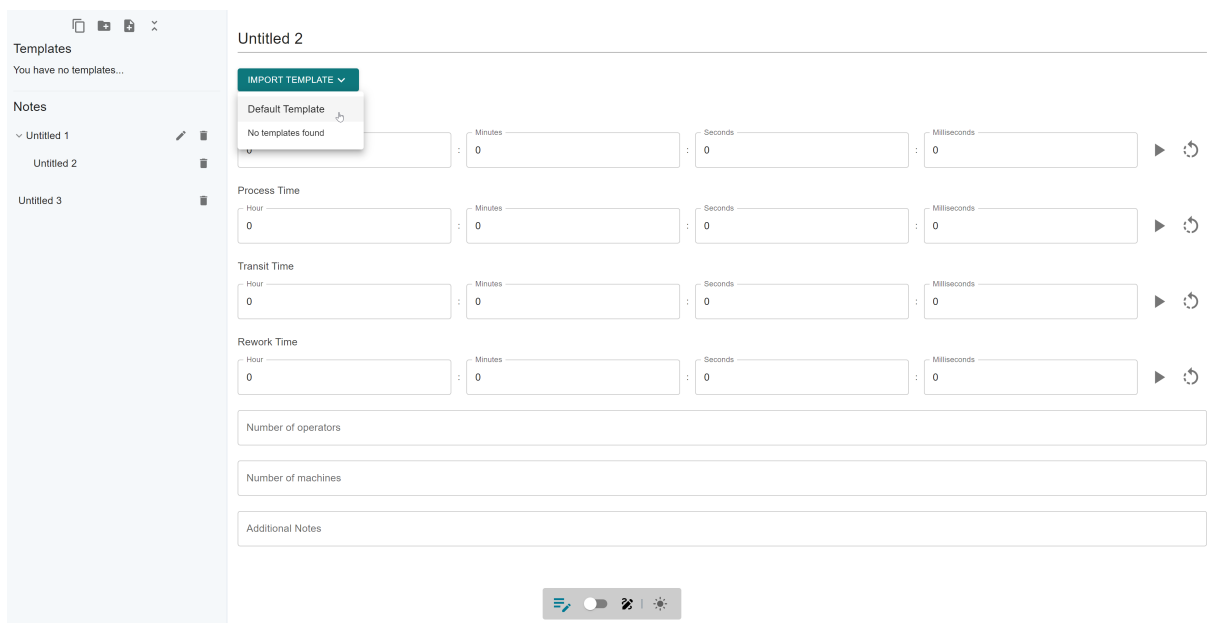


Abbildung A.15: Das Auswählen eines Templates in einer Notiz

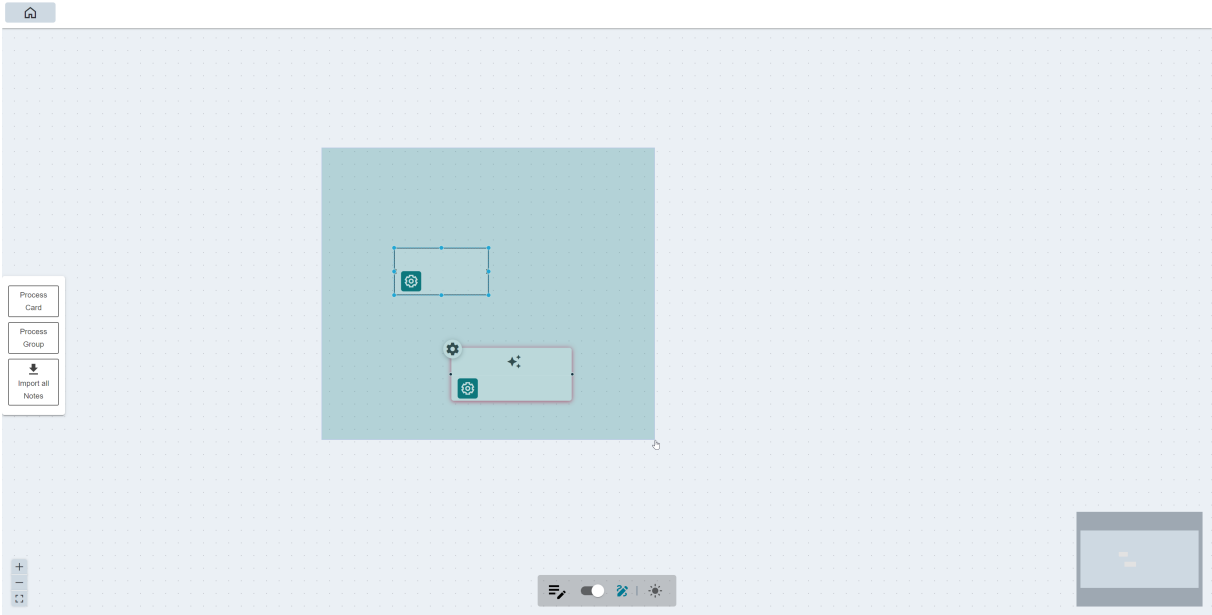


Abbildung A.16: Die Mehrfachauswahl im Editor

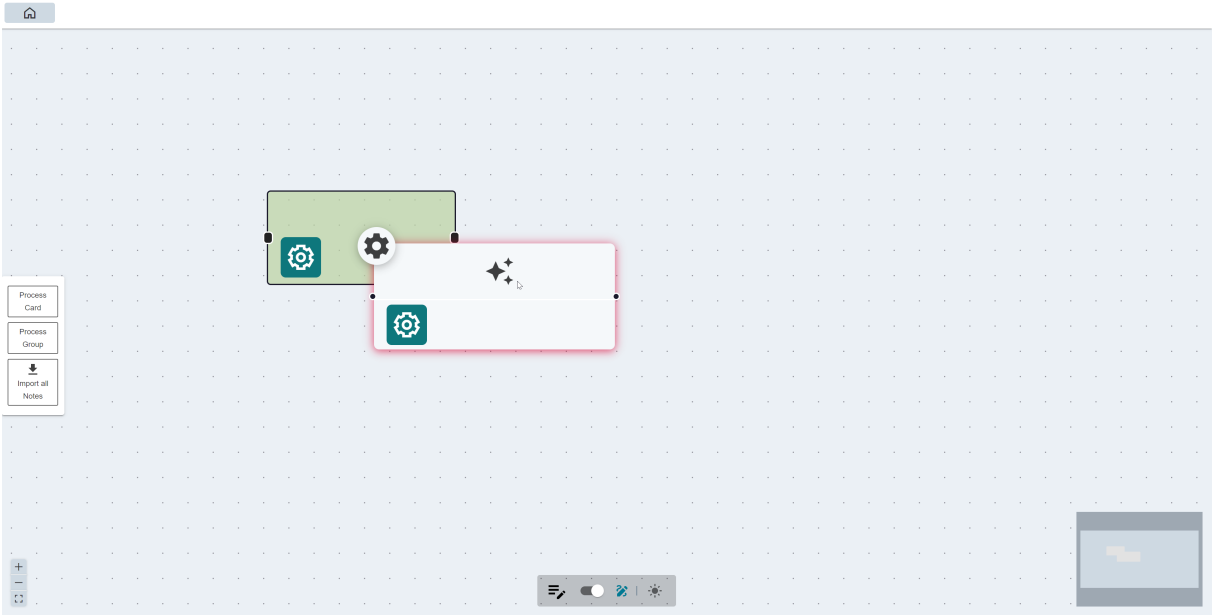


Abbildung A.17: Die Gruppierung von Karten durch Drag & Drop

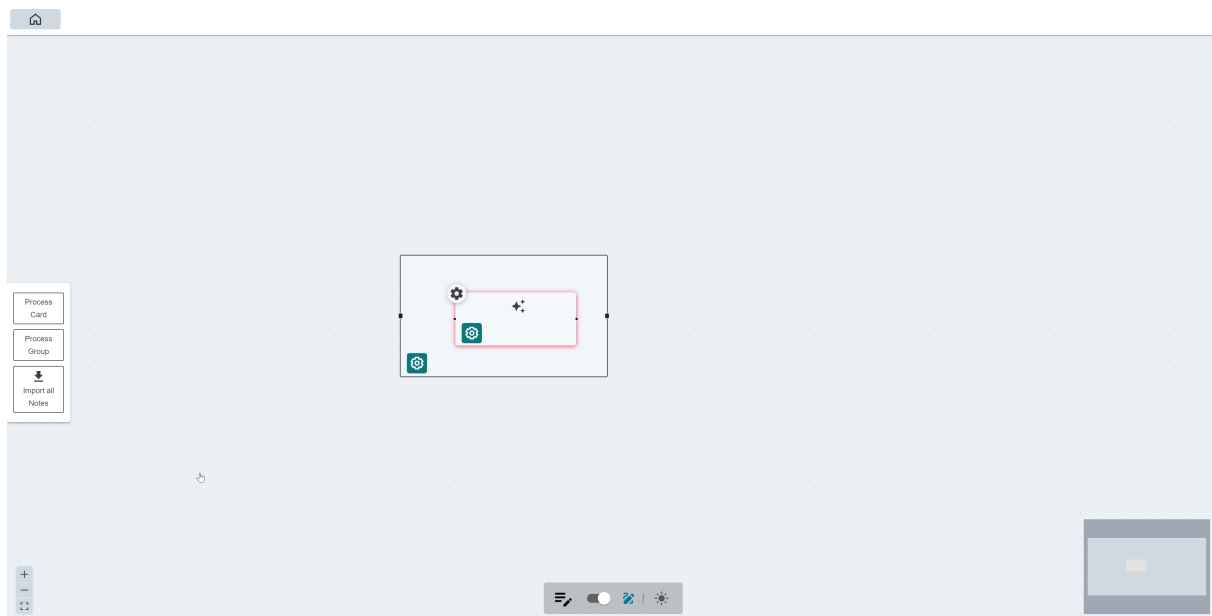


Abbildung A.18: Die Ansicht von gruppierten Karten

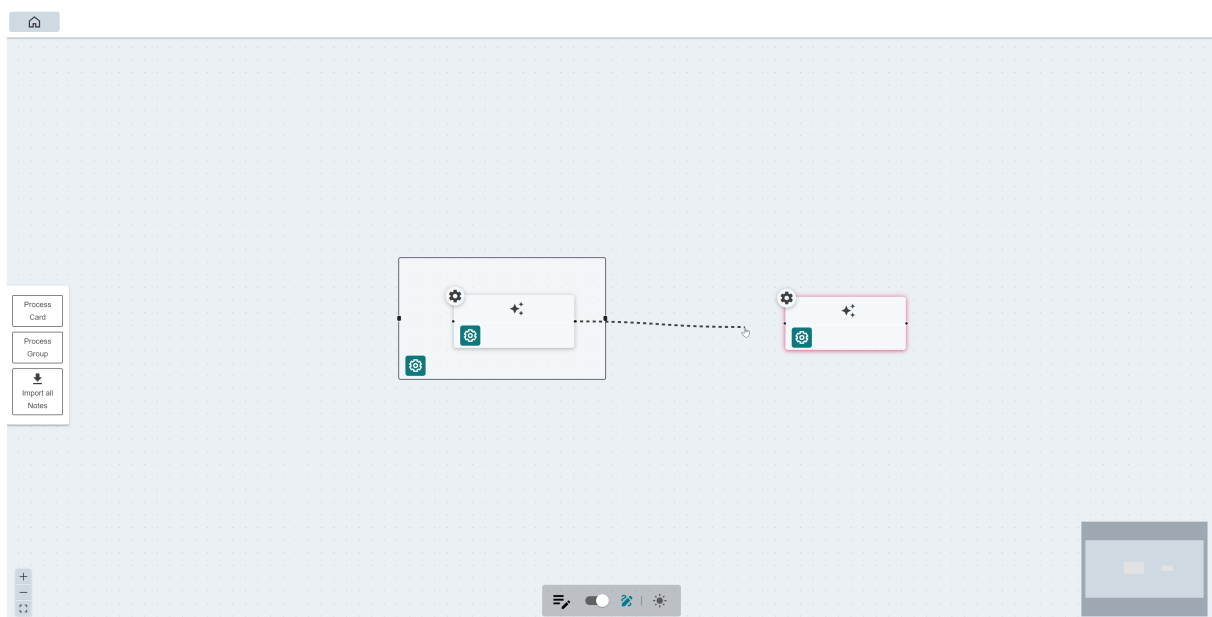


Abbildung A.19: Das Verbinden von Karten zu Visualisierung von Materialflüssen

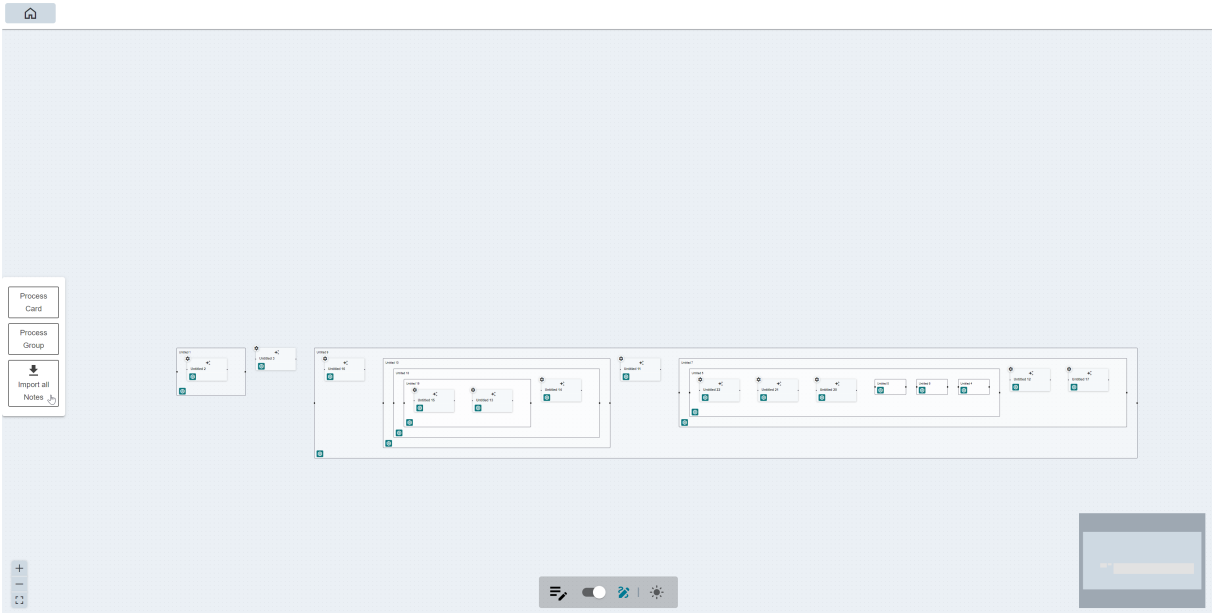


Abbildung A.20: Die Ansicht der importierten Notizen im Editor