

**EINSATZ DES
SHAREPOINT-WORKFLOW-MANAGERS
ZUR AUTOMATISIERUNG DES
BESTELL- UND
KONFIGURATIONSPROZESSES VON
SOFTWARE FÜR
FORMGEBUNGSMASCHINEN**

Eingereicht von
Winter Andreas BSc

Angefertigt am
Institut für Wirtschaftsinformatik - Data & Knowledge Engineering/flushleft

Betreuer
**o. Univ.-Prof. Dipl.-Ing.
Dr. techn. Michael Schrefl**

Mitbetreuung
Ass.-Prof. Mag. Dr. Christoph Schütz

Mai 2018



Masterarbeit
zur Erlangung des akademischen Grades
Master of Science
im Masterstudium
Wirtschaftsinformatik

Eidstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Masterarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Die vorliegende Masterarbeit ist mit dem elektronisch übermittelten Textdokument identisch.

Ort, Datum

Unterschrift

Kurzfassung

Sowohl die Aufrüstung der Software von Formgebungsmaschinen um neue Optionen als auch das Beziehen einer neuen Softwareversion ist ein komplizierter Prozess, der derzeit mehrere Wochen bis Monate dauern kann. Um diese Durchlaufzeit zu verkürzen wird die Automatisierung des Geschäftsprozesses mittels eines Workflows vorangetrieben. Die Modellierung ebenjenes Workflows kann mit verschiedensten Modellierungssprachen durchgeführt und mittels unterschiedlichsten Workflow-Management-Systemen automatisiert werden.

Im Zuge dieser Arbeit wird sowohl der Vergleich der beiden Modellierungssprachen Windows Workflow Foundation und BPMN als auch der Workflow-Management-Systeme Camunda, Bizagi und Share-Point gezeigt. Daraufhin erfolgt die Auswahl der Modellierungssprache und des Workflow-Management-Systems, welche ideal für die Automatisierung des Bestell- und Konfigurationsprozesses von Software für Formgebungsmaschinen im betrachteten Unternehmen geeignet sind. Zuletzt wird noch die Modellierung des Workflows in der ausgewählten Modellierungssprache gezeigt, ausgehend vom bisherigen Geschäftsprozessmodell bis zum finalen Endprodukt.

Abstract

The upgrade of software to add new options as well as the update to a new version for injection molding machines is a complicated process that can take from multiple weeks to even months. To shorten this time, process automation via automated workflows is investigated. This workflow can be modeled in different modeling languages and automated with diverse workflow automation systems.

In this paper the two modeling languages Windows Workflow Foundation and BPMN as well as the workflow management systems Camunda, Bizagi and SharePoint are compared. Subsequently, the ideal modeling language and workflow management system are chosen for the task, to automate the ordering and configuration process of software for injection molding machines. Lastly, the modeling of the workflow in the chosen modeling language is shown, starting from the current process model to the final model.

Inhaltsverzeichnis

1	Einführung	1
1.1	Aufgabenstellung	5
1.2	Vorhergehende Masterarbeit	7
1.3	State of the art	7
2	Business Process Model and Notation vs. Windows Workflow Foundation	8
2.1	BPMN	10
2.2	Windows Workflow Foundation	13
2.3	Vergleich und Beispiel	19
3	Auswahl des Workflow-Management-Systems	22
3.1	Kriterienkatalog	22
3.1.1	Kriterienauswahl	23
3.1.2	Kriterienbewertung	23
3.2	Überblick über die Workflow-Management-Systeme	24
3.2.1	Camunda	24
3.2.2	Bizagi	25
3.2.3	SharePoint	26
3.3	Vergleich der Workflow-Management-Systeme anhand des Kriterienkatalogs	28
4	Implementierung des Workflows	31
4.1	Vorhergehendes Prozessmodell in BPMN und WWF modelliert	31
4.1.1	Modellierung in BPMN	31
4.1.2	Modellierung in WWF	44
4.2	Anforderungsänderungen und der neue Workflow	58
4.3	Implementierung und Automatisierung des Workflows	71
4.3.1	User-Interfaces	71
4.3.2	Datenstruktur	72
4.3.3	Kommunikation über REST-API	74
5	Fazit - Erweiterungen und Entscheidungshilfe	75

1 Einführung

Die Firma Engel [14] ist ein weltweit tätiger Hersteller von Formgebungsmaschinen, auch Spritzgussmaschinen genannt, mit dem Firmensitz in Schwertberg in Oberösterreich. Diese Formgebungsmaschinen besitzen eine Steuerungseinheit, welche das Steuern und Überwachen der Maschine durch eine Person ermöglicht (Mensch-Maschine-Interface). Auf dieser Steuerungseinheit läuft eine Softwareversion mit individuellen Softwareoptionen, welche vom Kunden beim Kauf der Formgebungsmaschine angepasst werden kann. Um ebene Software jedoch nach dem Kauf auf eine neue Version upzudaten, oder um neue Optionen hinzuzufügen, muss ein manueller Prozess gestartet werden, welcher mehrere Wochen bis Monate dauern kann. Aufgrund dieser enorm langen Durchlaufzeit bleiben Formgebungsmaschinen regelmäßig auf dem Softwarestand der Auslieferung und werden weder upgedatet noch um neue Optionen erweitert.

Ein Ansatz, um dieses Problem zu lösen ist das Verwenden des Konzepts des BPM-Lifecycles. BPM steht für Business Process Management und ist laut [9] ein disziplinierter Ansatz, um sowohl automatisierte als auch nicht automatisierte Prozesse zu identifizieren, zu designen, auszuführen, zu dokumentieren, zu bemessen, zu überwachen und zu kontrollieren. Ein Prozess, in diesem Kontext, ist laut [9] ein Set von Aktivitäten, welches für das Erreichen eines Ziels oder das Lösen eines Problems verwendet wird. Eine sehr ähnliche Definition wird von [18] verwendet.

Da BPM laut [9] eine permanente und kontinuierliche organisatorische Verbindlichkeit, die Prozesse des Unternehmens zu verwalten, darstellt, lässt sich dieser Ansatz mittels eines Lebenszyklus-Modells darstellen. In dieser Masterarbeit wird das Lebenszyklus-Modell von [9] verwendet, aufgrund dessen großer Verbreitung und Anwendung in der Praxis.

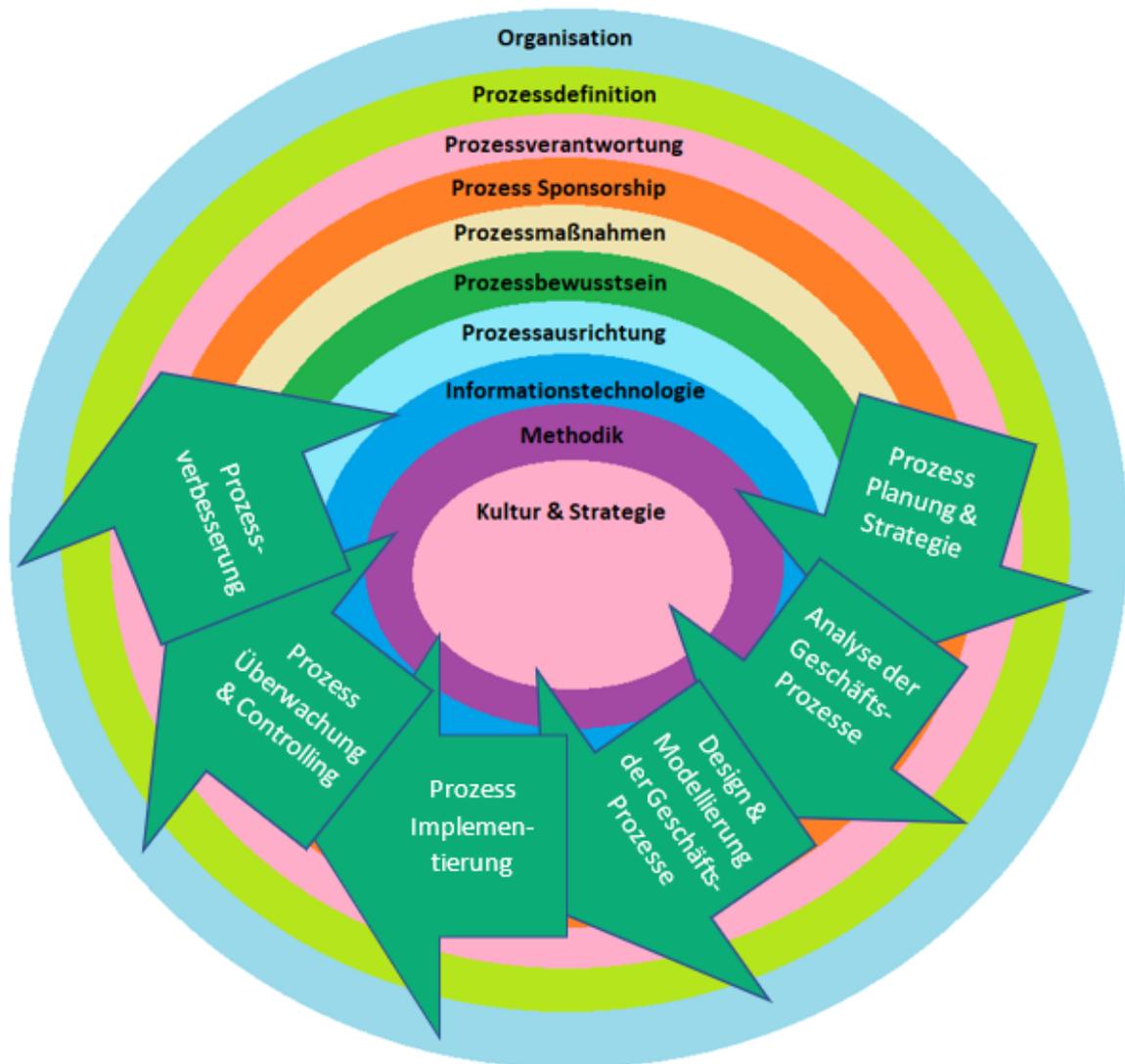


Abbildung 1: BPM Lebenszyklus-Modell nach [9]

Wie in Abbildung 1 zu sehen ist, besteht dieses Modell aus folgenden sechs Schritten:

- **Prozess Planung & Strategie:** Die erste Phase befasst sich mit dem Entwickeln einer prozessorientierten Strategie für das Unternehmen. Die Strategie gibt die Struktur und Richtung für zukünftiges kundenorientiertes Prozessmanagement vor und legt die Grundlage für einen ganzheitlichen BPM-Ansatz, der mit der Unternehmensstrategie übereinstimmt.
- **Analyse der Geschäftsprozesse:** Die zweite Phase des Lebenszyklus-Modells analysiert die Geschäftsprozesse mittels verschiedener Methodiken, um die derzeitigen Prozesse des Unternehmens zu verstehen. Dabei werden die Informationen von Strategieplänen, Prozessmodellen, Performance-Kennzahlen, Umweltänderungen und anderen Faktoren verwendet, um die Geschäftsprozesse vollständig im Kontext des gesamten Unternehmens zu verstehen.

- **Design & Modellierung der Geschäftsprozesse:** In der dritten Phase liegt der Fokus auf dem Designen und Modellieren der Geschäftsprozesse. Dabei wird die Sequenz der Aktivitäten dokumentiert. Diese umfasst, welche Arbeit durchgeführt wird, zu welcher Zeit, an welchem Ort, durch welche Person und mit welcher Methodik. Ein weiterer wichtiger Teil des Designs ist die Auswahl entsprechender Metriken, um Performance-Kennzahlen messen zu können. In den ersten Durchläufen dieses Modells wird in dieser Phase der Fokus auf dem Standardisieren/Automatisieren von Aktivitäten liegen. Nach mehrmaligem Durchlauf des Modells wird der Fokus auf dem Neu-Entwerfen beziehungsweise dem graduellen Verbessern der Aktivität liegen.
- **Prozessimplementierung:** In der Implementierungsphase werden die zuvor modellierten Prozesse implementiert und automatisiert.
- **Prozessüberwachung & Controlling:** Fortlaufendes Messen und Aufzeichnen von Geschäftsprozessen erzeugt die nötige Information, um die Ressourcen für einen Geschäftsprozess so anzupassen, dass dieser die Prozessziele erreicht. Weiters kann die Analyse der Prozessleistungsinformationen zur Verbesserung, dem Neu-Entwerfen oder dem Umstrukturieren der Aktivitäten führen.
- **Prozessverbesserung:** In der letzten Phase des Lebenszyklus-Modells werden die Ergebnisse der iterativen Analyse und der Designphase implementiert. Es werden dabei die Herausforderungen des organisatorischen Change-Management betrachtet, welche die kontinuierliche Verbesserung und die Prozessoptimierung sind. Das Ziel dabei ist, dass die Prozesse konsistent die vorgegebenen Effizienz- und Effektivitäts-Ziele erreichen.

Ein anderer Ansatz, um das Problem zu bewältigen, ist das Verwenden des Konzepts des Workflow-Managements. Um dieses Konzept zu verstehen, muss zunächst der Begriff Workflow definiert werden, da dieser unterschiedlichste Definitionen besitzt und zentral für Workflow-Management ist. Die bekannteste und am häufigsten verwendete Definition stammt von der Workflow Management Coalition [22], welche einen Workflow als die computergesteuerte Automatisierung eines Geschäftsprozesses, teilweise oder zur Gänze, definiert.

Nachdem der Begriff Workflow definiert ist, kann das Konzept Workflow-Management, welches laut [17] eine Technologie zur Unterstützung der Modellierung und Umstrukturierung von Geschäfts- als auch Informationsprozessen ist, definiert werden. Darunter fällt unter anderem die Definition von Workflows und deren Umstrukturierung und Neudesign bei sich ändernden Gegebenheiten.

[43] wiederum definiert Workflow-Management etwas allgemeiner als das Kontrollieren, Überwachen,

Optimieren und Unterstützen von Geschäftsprozessen, die mittels einer Modellierungssprache abgebildet wurden. Das Ziel dabei ist, dass die korrekte Aktivität zur richtigen Zeit durch die geeignete Person ausgeführt wird.

Um das Konzept des Workflow-Managements effizient verwenden zu können, wird meist ein Informationssystem, ein sogenanntes Workflow-Management-System (WfMS), verwendet. Gemäß der Workflow Management Coalition [22] wird ebenjenes Workflow-Management-System als ein Informationssystem definiert, das einen Workflow vollständig definiert, verwaltet und ausführt mittels der Ausführung von Software. Die Reihenfolge der Ausführung der Software wird dabei von einer digitalen Darstellung der Workflowlogik, sprich dem modellierten Geschäftsprozess, bestimmt.

Die beiden Ansätze, BPM und Workflow-Management, haben gemäß ihrer Definitionen große Übereinstimmungen und werden sehr oft als Synonyme verwendet, jedoch unterscheiden sie sich doch signifikant. Laut [7], [35] und [42] ist Workflow-Management lediglich ein Teil von BPM, da BPM nicht nur auf die Automatisierung eines Geschäftsprozesses fokussiert ist, sondern auf die stetige Verbesserung und das Zusammenspiel aller Geschäftsprozesse eines Unternehmens.

In dieser Masterarbeit liegt der Fokus auf dem Automatisieren eines Geschäftsprozesses mittels Workflow-Management unter der Verwendung eines Workflow-Management-Systems. Zunächst wird die Aufgabenstellung dieser Arbeit in Kapitel 1.1 definiert. Im nachfolgenden Unterkapitel wird die vorhergehende Masterarbeit von Karl-Heinz Stöckler vorgestellt, da diese der Ausgangspunkt für diese Arbeit ist. Weiters werden in Kapitel 1.3 Arbeiten, die thematische Überschneidungen mit dieser besitzen, dargestellt. In Kapitel 2 werden zwei verschiedene Modellierungssprachen, die für die Automatisierung von Geschäftsprozessen geeignet sind, Business Process Model and Notation (BPMN) und Windows Workflow Foundation (WWF), beschrieben und anhand eines Beispiel-Geschäftsprozesses verglichen. Danach werden in Kapitel 3 unterschiedliche Workflow-Management-Systeme vorgestellt, die sowohl die Modellierung als auch die Automatisierung des Geschäftsprozesses beherrschen. Dabei werden die Vor- und Nachteile der einzelnen Tools gegenübergestellt. In Kapitel 4 wird der Geschäftsprozess der vorhergehenden Masterarbeit umstrukturiert und neu modelliert. Zuletzt wird in Kapitel 5 eine Auswahlhilfe für die Auswahl der Modellierungssprache und für das Workflow-Management-System bereitgestellt.

1.1 Aufgabenstellung

Das Ziel dieser Masterarbeit ist die Implementierung eines Workflow-Management-Systems, welches den Softwaregenerierungsprozess für Formgebungsmaschinen automatisiert, um die Effizienz zu steigern, das Fehlerrisiko zu minimieren und die Durchlaufzeit zu senken. Abbildung 2 zeigt die Zielarchitektur des Systems.

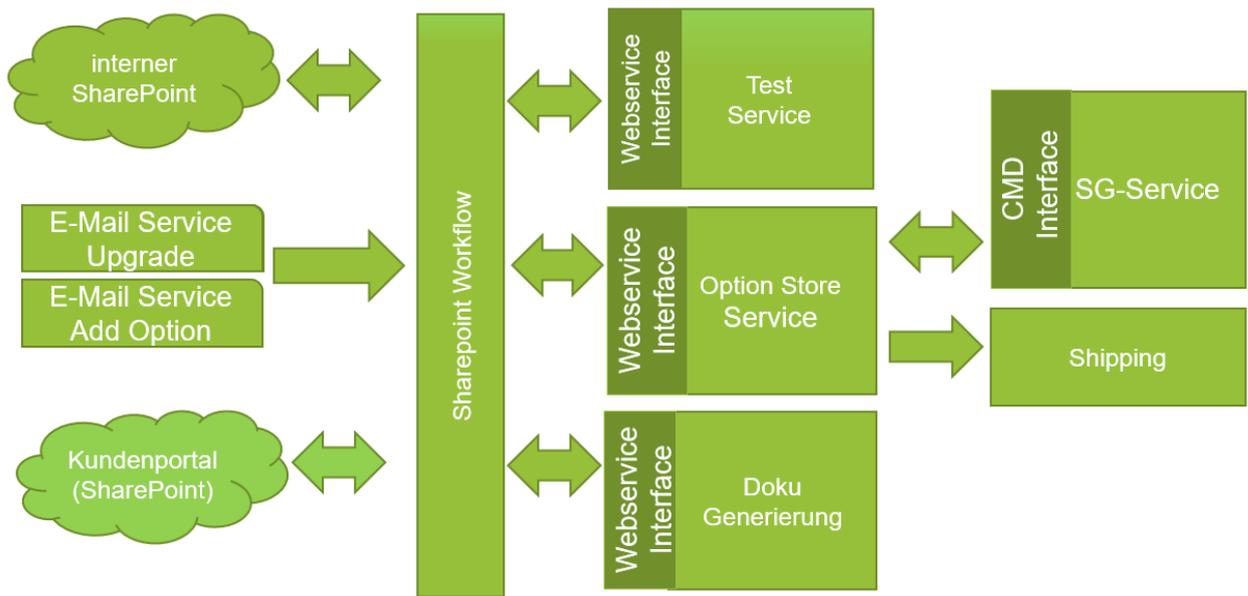


Abbildung 2: Architekturüberblick der Aufgabenstellung

Insgesamt werden drei User-Interfaces benötigt, welche das Workflow-Management-System starten können. Das erste User-Interface startet den Workflow, nachdem eine E-Mail gesendet wurde. Das zweite User-Interface ist eine Weboberfläche für interne Mitarbeiter, die in die SharePoint-Umgebung eingebunden wird. Das dritte und letzte User-Interface ist eine Weboberfläche für Kunden, die in das Engel-Portal eingebunden wird. Diese Weboberfläche wird jedoch im Zuge dieser Masterarbeit nicht umgesetzt, da die Firma Engel für die Umsetzung dieser Weboberfläche ein gesondertes Projekt einsetzt.

Neben diesen drei User-Interfaces kommuniziert der SharePoint-Workflow, wie in Abbildung 2 ersichtlich ist, mit drei weiteren Komponenten: dem Testservice, dem Optionstore und der Doku-Generierung. Der Optionstore bildet gemeinsam mit dem Workflow das Herzstück der Architektur, da der Optionstore neben der Kommunikation mit dem SG-Service, welcher die Softwaregenerierung übernimmt, auch die Kommunikation mit dem Shipping-Service enthält. Der Testservice wiederum prüft die neu generierte Software und liefert einen Bericht, der angibt, ob beim Testdurchlauf Fehler entstanden sind. Die Doku-Generierung generiert, je nach Formgebungsmaschine, manuell oder automatisch eine neue Doku-

mentation für die neue Software.

Das Ziel dieser Architektur ist der Ablauf, der in Abbildung 3 gezeigt wird.

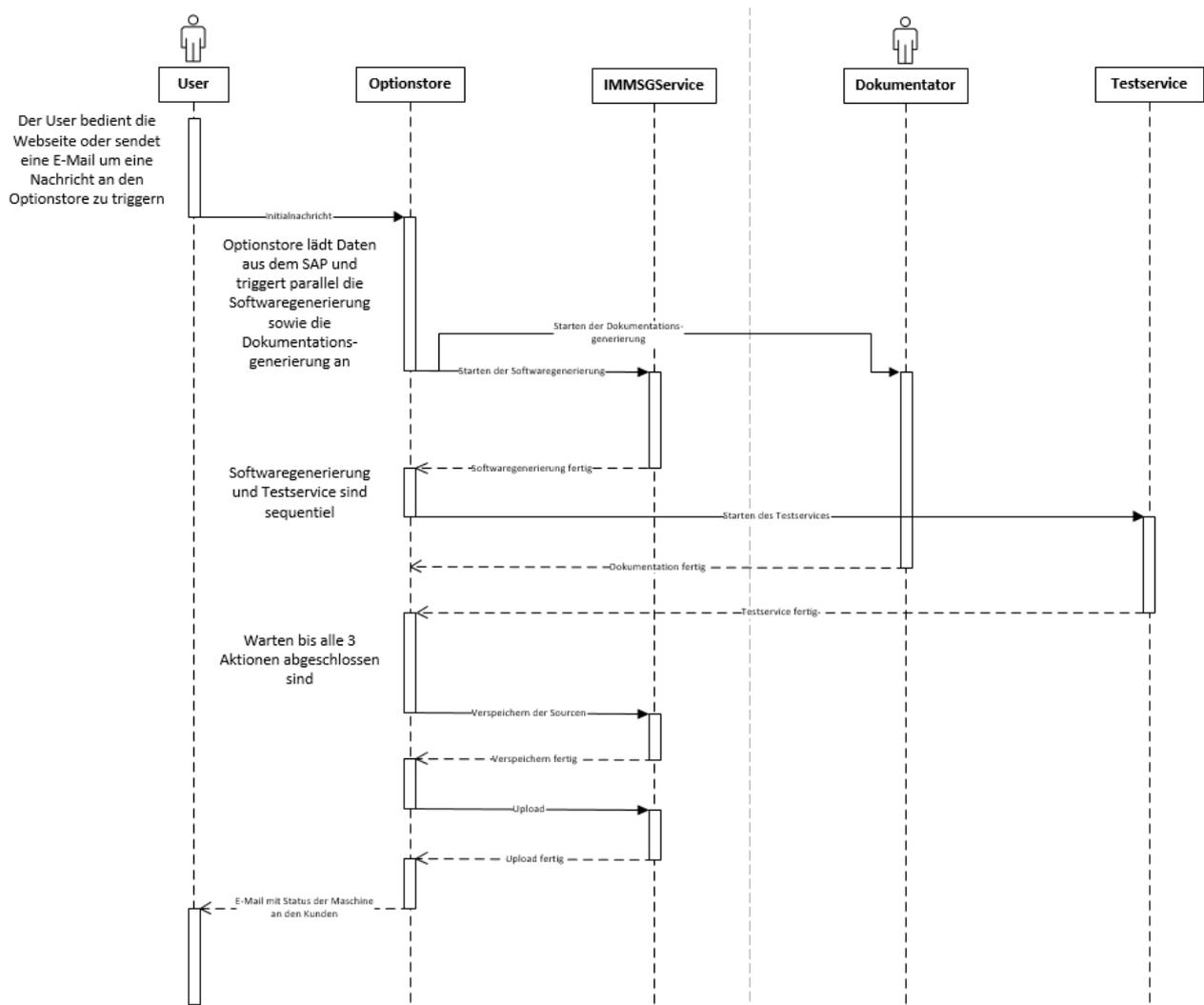


Abbildung 3: Ablauf des Workflows zur automatischen Softwaregenerierung

Der User gibt zunächst via Weboberfläche oder mittels einer E-Mail die benötigten Informationen (Maschinennummer, Kontrollversion und Optionen, die hinzugefügt/entfernt werden sollen) ein und der Workflow wird gestartet. Zunächst bezieht der Workflow mittels eines Webservices des Optionstores die benötigten Daten zum Generieren der Software aus dem SAP. Danach werden sowohl die Softwaregenerierung als auch die Dokumentationsgenerierung parallel aus dem Workflow via Webservice des Optionstores gestartet. Sobald die Softwaregenerierung abgeschlossen ist, meldet dies der Optionstore an den Workflow, welcher den Testservice mittels eines Webservices des Optionstores startet. Wenn sowohl der Testservice als auch die Dokumentationsgenerierung durchgelaufen sind, kann fortgefahren werden. Der nächste Schritt ist das Sichern der generierten Daten im SAP mittels eines Webservices des Optionstores. Zuletzt

wird noch die generierte Software inklusive Dokumentation an den Kunden gesendet, indem ein Upload-service diese Daten auf einen Webserver lädt und der Kunde eine E-Mail erhält, in der die Webadresse enthalten ist.

1.2 Vorhergehende Masterarbeit

Die Masterarbeit von Karl-Heinz Stöckler aus dem Jahr 2015 mit dem Titel "Prototypische Umsetzung der Automatisierung eines Prozesses für Nach- und Aufrüstung von Komponenten einer Formgebungsmaschine" [40] ist jene Masterarbeit, auf welcher diese aufbaut.

Die Aufgabenstellung der Arbeit war das Erzeugen eines Prototyps, welcher die Bestellung von Hard- und Softwarekomponenten für Formgebungsmaschinen mittels eines App-Stores ermöglicht.

Um dies zu erreichen, wurden aus zunächst 16 Business-Process-Management-Werkzeugen vier ausgewählt, welche mittels einer Nutzwertanalyse verglichen wurden. Die vier Werkzeuge von Bonitasoft, Bizagi, Webratio und Camunda wurden aufgrund der unterstützten Datenbanken, den Anpassungsmöglichkeiten der Formularelemente, den Anpassungsmöglichkeiten der Benutzeroberfläche, der Benutzerfreundlichkeit und der Erweiterbarkeit verglichen. Durch die zusätzlichen Ausschlusskriterien, dass das Werkzeug kostenlos und nicht cloudbasiert sein muss, fiel die Wahl auf das Werkzeug von Camunda. Dieses Werkzeug wurde verwendet, um den gewünschten Prozess zu modellieren, worauf in Kapitel 4.1 noch genauer eingegangen wird.

Weiters wurde in der Arbeit gezeigt, wie der modellierte Geschäftsprozess automatisiert wurde und welcher Code dafür verwendet wurde.

1.3 State of the art

Der BPM Lifecycle, der in [9] präsentiert wird, ist nicht der einzige BPM-Ansatz. [26] vergleicht mehrere unterschiedliche Lebenszyklus-Modelle miteinander und zeigt die Unterschiede zum Modell von [9] auf. Die BPM Lebenszyklus-Modelle, die verglichen werden, sind jene von Hallerbach et al. [19], Houy et al. [24], Netjes et al. [33], Van der Aalst [45], Verma [46], Weske [47] und Zur Muehlen and Ho [49].

Die Modellierung eines Geschäftsprozesses im Zuge des BPM-Lebenszyklus-Modells kann sowohl in verschiedensten Modellierungssprachen als auch für verschiedenste Zwecke durchgeführt werden. [1] gibt einen Überblick über einen Großteil der Anwendungsfälle und Techniken und beschreibt sowohl deren Vor- als auch Nachteile. Diese Techniken reichen von Flowcharts über DFD, RAD, RID Gantt-Diagramme bis hin zu Petri-Netzen und Workflows. Workflows werden dabei als Flüsse von Aufgaben zwischen

Personen oder Computerprogrammen in einer Organisation beschrieben. Weiters heißt es, dass Workflow nicht nur eine Technik zum Modellieren ist, es umfasst auch das Analysieren und Verbessern von Geschäftsprozessen.

Zwei mögliche Modellierungssprachen-Kategorien sind blockstrukturierte und graphenbasierte Modellierungssprachen. In [25] werden die beiden Kategorien beschrieben und deren Unterschiede dargelegt. Der Fokus der Arbeit liegt dabei auf dem Vergleich der Modellierungssprachen BPMN als graphenbasierte und BPEL als blockstrukturierte Modellierungssprache. Insbesondere werden die Unterschiede zwischen den Verknüpfungs-Bedingungen und den Schleifen der beiden Modellierungssprachen gezeigt.

Zum Modellieren eines Workflows mittels einer Modellierungssprache muss ein Workflow-Management-System ausgewählt werden. Um eine Auswahl treffen zu können, müssen diese anhand von verschiedensten Kriterien beziehungsweise einem Kriterienkatalog verglichen werden. [10] zeigt einen Kriterienkatalog auf, welcher als Ausgangsbasis für die Kriterienwahl verwendet werden kann. Dabei können unwichtige Kriterien verworfen und andere angepasst werden, um den Anforderungen des Anwendungsfalls zu genügen und die perfekte Wahl zu treffen.

[50] wiederum zeigt einen anderen Ansatz auf, um Workflow-Management-Systeme zu vergleichen. Dabei werden die jeweiligen Systeme als Metamodelle abgebildet und ebenjene werden verglichen. Dadurch ergibt sich ein Evaluierungsansatz, der einen Mittelwert zwischen horizontal-dominanten (Kriterien haben eine starke Struktur) und vertikal-dominanten Evaluierungsansätzen (Kriterien sind sehr komplex), nach [39], darstellt.

2 Business Process Model and Notation vs. Windows Workflow Foundation

Es gibt eine Vielzahl an Modellierungssprachen, die geeignet sind, um Geschäftsprozesse darzustellen. Gemäß [25] unterscheiden sich diese Modellierungssprachen nicht nur bei der Anzahl und der Auswahl der Elemente, sondern auch bei der Struktur des Modells und dem vorgesehenen Einsatzbereichs des Modells.

Gemäß dieser Eigenschaften können Modellierungssprachen nach ihrer Struktur (Sequenzflussmodellierung ist Block- oder Graphen-orientiert) in graphenbasierte oder blockstrukturierte Modellierungssprachen unterteilt werden. Bei graphenbasierten Modellierungssprachen, wie Business Process Model and Notation, kurz BPMN, oder Eventdriven Process Chains, kurz EPC [38], wird der Sequenzfluss (legt

die Reihenfolge der Aktivitäten fest) zwischen den Aktivitäten explizit mittels Kontrolllinks festgelegt. Blockstrukturierte Modellierungssprachen, wie Windows Workflow Foundation, kurz WWF, oder Web Service Business Process Execution Language, kurz BPEL [3], hingegen definieren den Sequenzfluss wie herkömmliche Programmiersprachen mit Block-Konstrukten wie If oder While. Dies wird in Abbildung 4 verdeutlicht, indem die Struktur der beiden Modellierungssprachen BPMN und WWF gegenübergestellt wird.

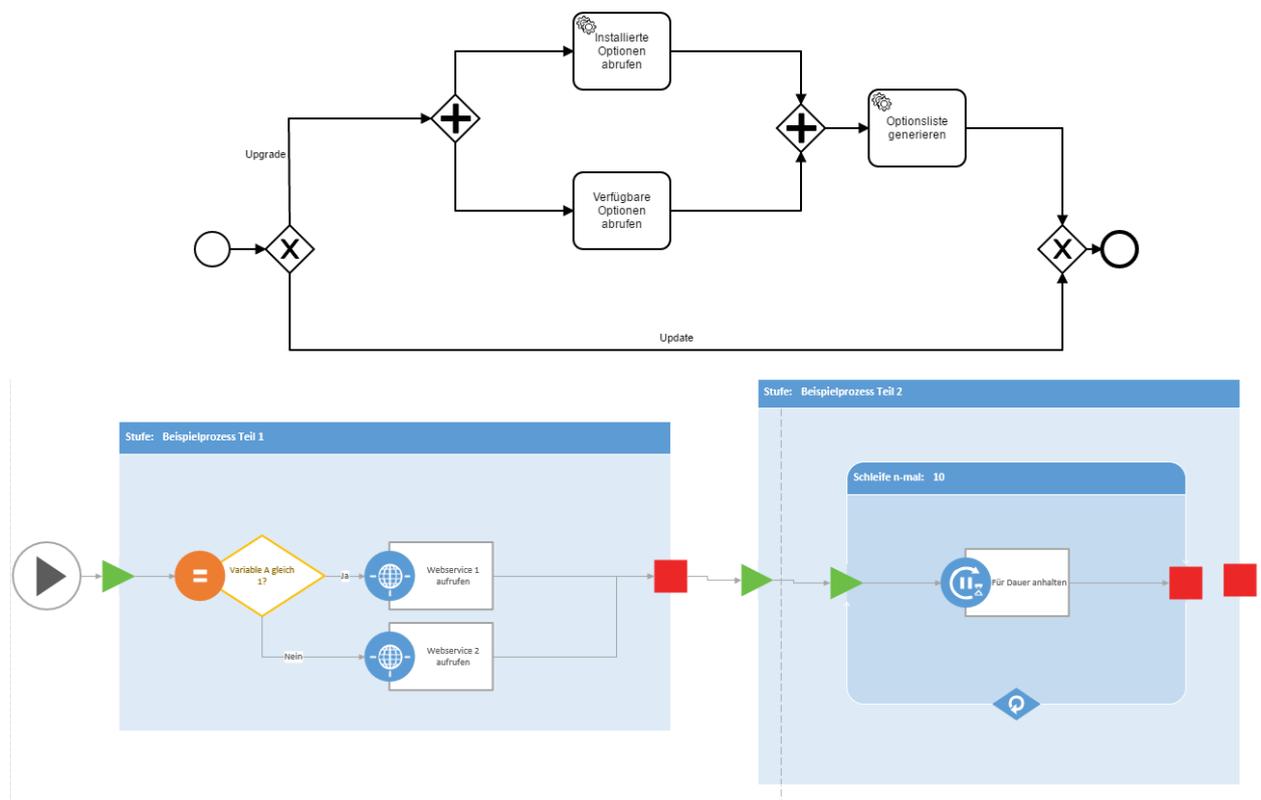


Abbildung 4: Gegenüberstellung blockstrukturierter vs. graphenbasierter Modellierungssprachen

Weiters können Modellierungssprachen auch nach ihrem vorgesehenen Einsatzbereich kategorisiert werden. So gibt es auf der einen Seite Modellierungssprachen zum Darstellen und besseren Erklären eines Geschäftsprozesses und auf der anderen Seite Modellierungssprachen, die auf das Ausführen und Automatisieren des modellierten Geschäftsprozesses abzielen. Gemäß [37] ist diese Unterteilung auf konzeptionelle Unterschiede zwischen den beiden Modellierungssprachenkategorien zurückzuführen. Zur ersten Kategorie gehören BPMN und EPC, zur zweiten WWF und BPEL.

In den folgenden Unterkapiteln wird zunächst ein Vertreter der beschreibenden graphenbasierten Modellierungssprachen beschrieben, dann ein Vertreter der automatisierenden blockstrukturierten Modellierungssprachen und zuletzt wird der Unterschied zwischen den beiden Notationen gezeigt. Als graphenbasierte Modellierungssprache wurde BPMN ausgewählt, da BPMN von Karl-Heinz Stöckler in der vorherge-

henden Masterarbeit [40] verwendet wurde. Als blockstrukturierte Modellierungssprache wurde WWF ausgewählt, da die Firma Engel, bei welcher die praktische Umsetzung dieses Projektes durchgeführt wurde, diese bereits in Verwendung hat.

2.1 BPMN

BPMN ist laut [11] die Standardnotation, um Geschäftsprozesse graphisch abzubilden. Die durch das Modellieren der Geschäftsprozesse entstehenden BPMN Prozessmodelle bestehen aus:

- **Aktivitätsknoten**, welche Events oder Leistungen, die von einer Person oder einem Programm ausgeführt wurden, darstellen
- **Kontrollknoten**, welche den Kontrollfluss zwischen den Aktivitätsknoten darstellen

Aktivitäts- und Kontrollknoten können dabei, im Bezug auf den Ablauf, in nahezu willkürlicher Art und Weise miteinander verbunden werden.

Durch die Zugehörigkeit BPMNs zur Klasse der graphenorientierten Prozessmodellierungssprachen ergibt sich gemäß [11] die Eigenschaft, dass BPMN-Modelle semantische Fehler aufweisen können. Dies können Dead- als auch Livelocks [21] sein. Weitere Eigenschaften BPMNs, die von verschiedenen Quellen wie Workflow Patterns [48] und Business Process Execution Language (BPEL) abgeleitet wurden und eher untypisch für graphenorientierte Prozessmodellierungssprachen sind, sind:

- die Möglichkeit, einen Subprozess zu definieren, welcher mehrmals gleichzeitig ausgeführt wird
- die Möglichkeit, einen Subprozess zu definieren, welcher aufgrund eines Fehlers abgebrochen wird
- die Möglichkeit, Nachrichten zwischen Prozessen zu versenden

BPMN umfasst insgesamt mehr als 100 Symbole, weshalb BPMN laut [13] eine komplexe Notation ist. Unter diesen 100 Elementen befindet sich eine Vielzahl an Elementen, die laut [11] von anderen Geschäftsprozessmodellierungs-Notationen, wie der XML Process Definition Language (XPDL) [44] und die Aktivitätsdiagramm-Komponente von UML (Unified Modeling Language) [16], übernommen wurden. Die wichtigsten und essentiellen BPMN-Elemente zum Bauen von Modellen gehören dem Kernset an. Dieses umfasst laut [11] folgende Elemente:

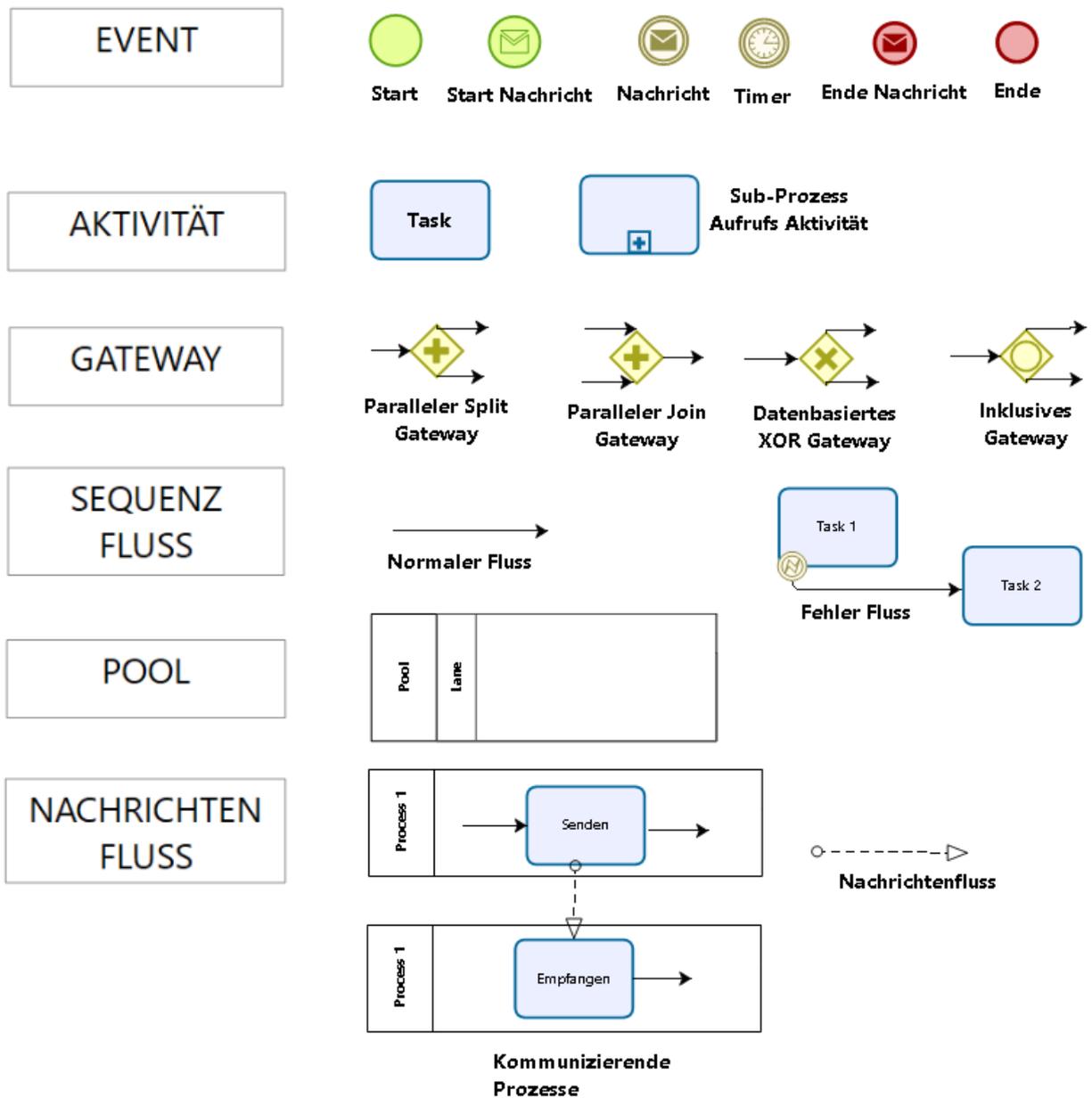


Abbildung 5: Kernset der BPMN-Elemente nach [11]

- **Objekte:**

- **Event:**

- * **Start** signalisiert den Beginn einer Instanz des Prozesses
 - * **Start Nachricht** signalisiert den Beginn eines Nachrichtenaustauschs
 - * **Nachricht** signalisiert das Senden oder Erhalten einer Nachricht
 - * **Timer** signalisiert, dass ein gewisses Datum erreicht wurde
 - * **Ende Nachricht** signalisiert das Ende eines Nachrichtenaustauschs

- * **Ende** signalisiert das Ende einer Instanz des Prozesses

– **Aktivität:**

- * **Task** ist eine atomare Aktivität und repräsentiert Arbeit, die während des Prozesses durchgeführt wird. Es gibt dabei 7 Tasktypen, die in Abbildung 6 zu sehen sind.

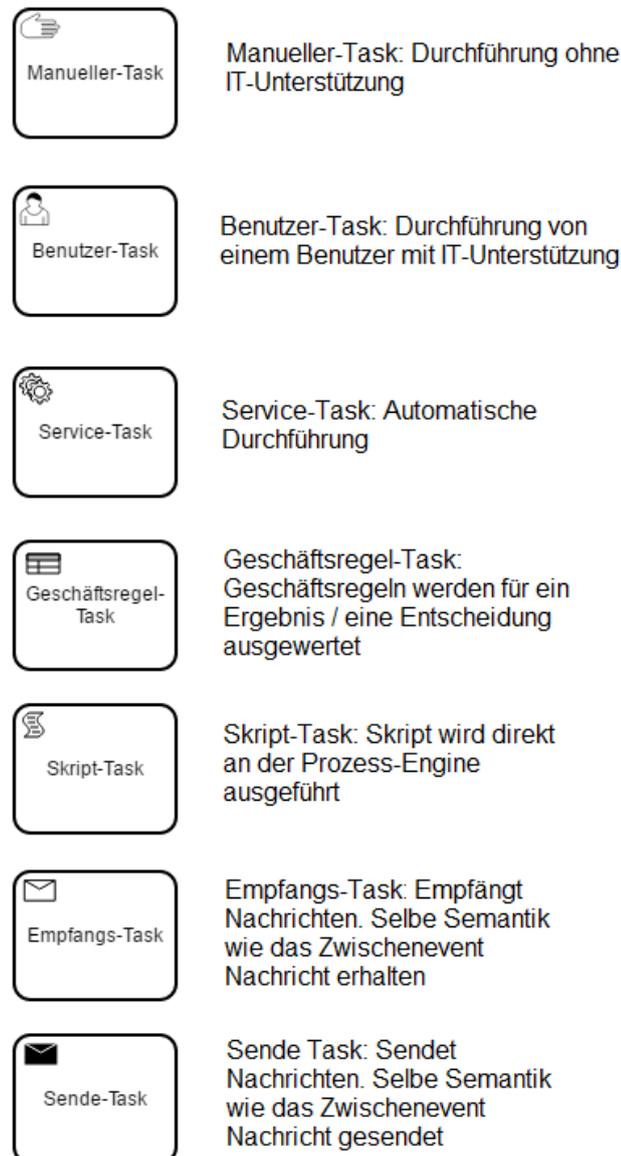


Abbildung 6: BPMN Tasktypen nach [31]

- * **Sub-Prozess** ist eine Verbunds-Aktivität und wird durch den Ablauf anderer Aktivitäten definiert. Teilprozesse können dabei eingebettet (Teil des Prozesses) oder eigenständig (kann auch von anderen Prozessen aufgerufen werden) sein. Aufgerufen werden können diese Teilprozesse mittels der **Sub-Prozess Aufrufs Aktivität**.
- **Gateway** ist ein Steuerungskonstrukt, um die Aufspaltungen und Zusammenführungen des

Sequenzflusses zu kontrollieren

- * **Paralleler Split Gateway** spaltet den Sequenzfluss in 2 parallele Sequenzflüsse
 - * **Paralleler Join Gateway** vereinigt 2 parallele Sequenzflüsse und führt den Sequenzfluss fort, sobald beide fertig sind
 - * **Datenbasiertes XOR Gateway (XOR-split)** wählt einen Sequenzfluss aus mehreren sich einander ausschließenden Optionen aufgrund von Prozessdaten aus
 - * **Inklusives Gateway** spaltet den Sequenzfluss in einen oder mehrere Sequenzflüsse
- **Sequenzfluss:** verbindet zwei Objekte im BPD und zeigt die Ablafrichtung an
 - **Fehler Fluss:** Wenn ein Zwischenevent wie **Nachricht**, **Timer** oder **Error** am Rahmen einer Aktivität angeheftet ist, so stellt dies eine Exception dar. Ein solches Event wird als Exception-Event bezeichnet. Tritt solch ein Event ein, so wird die Aktivität unterbrochen und der Sequenzfluss wird vom Exception-Event aus fortgeführt.
 - **Nachrichtenfluss:** zeigt die Übertragung von Nachrichten zwischen zwei kommunizierenden Prozessen via Kommunikationsaktionen wie send/receive oder einem message-Event an. Die kommunizierenden Prozesse müssen in unterschiedlichen Pools sein.

All diese Elemente müssen im Zuge des Modellierens in einen Pool beziehungsweise in eine Lane verpackt werden, da diese, laut [2], als Container für einen abgeschlossenen Prozess dienen und festlegen, wer die Elemente in der Lane beziehungsweise dem Pool ausführt. Gemäß [13] repräsentiert ein Pool eine Ressourcenklasse, wie etwa ein Unternehmen oder ein Server, und eine Lane eine Ressource, wie etwa eine Abteilung des Unternehmens oder ein Programm des Servers. Eine Lane muss dabei immer in einem Pool verpackt sein, ein Pool muss jedoch keine Lane enthalten und Lanes können verschachtelt werden, um komplexe Strukturen darzustellen.

Beim Bauen eines BPMN-Modells mit diesen Elementen entsteht ein Business Process Diagram, kurz BPD, welches eine Art der Flowchart-Diagramme ist. Im Gegensatz zu anderen Flowchart-Diagrammen kann gemäß [13] ein BPD mehrere Start- und Stopp-Events beinhalten. Diese Eigenschaft kann dazu verwendet werden, um unterschiedliche Start- und Stopp-Ereignisse abzubilden.

2.2 Windows Workflow Foundation

[25] beschreibt die Windows Workflow Foundation als eine blockstrukturierte Modellierungssprache mit dem Ziel, ein ausführbares Modell zu erzeugen. [34] hingegen beschreibt die Windows Workflow Foundati-

on als ein Set an Bibliotheken, Typen und Tools, um workflowbasierte Softwareapplikationen zu erstellen. Die aktuelle Version ist Version 4.5, welche auf dem .Net Framework 4.5[36] basiert.

Vor dem Modellieren eines Workflows in WWF muss zunächst eines der drei Workflowmodelle Sequenz, Flowchart und State machine ausgewählt werden. Eine Sequenz ist dabei ein sequentieller Workflow, für welchen ein expliziter Start- und Endpunkt definiert wird. Der Workflow wird vom Startpunkt bis zum Endpunkt entlang des Sequenzflusses ausgeführt, ohne der Möglichkeit, einen Schritt zu einer vorherigen Aktion zu machen. Ein Flowchart definiert den Sequenzfluss wie den eines Flussdiagrammes, welcher vom Start- bis zum Endpunkt durchquert wird, mit der Möglichkeit auf vorherige Aktionen zurückzuspringen, wenn gewisse Bedingungen gegeben sind. Eine State machine definiert den Sequenzfluss mittels eines Sets an Stati und eines Sets an Regeln.

Zwischen diesen drei Workflowmodellen gibt es jedoch keine klare Trennung und so können alle drei Modelle in einem Workflow kombiniert werden, was in der Praxis gemäß [34] meist vorkommt. Im Zuge dieser Arbeit ist beim Verwenden des Terms WWF-Workflow immer die Rede von einem WWF-Flowchart-Workflow.

Jeder Workflow, der mit WWF modelliert wurde, besteht aus mindestens einer Aktivität, welche die kleinstmögliche Ausführungseinheit darstellt. Die verwendeten Aktivitäten können entweder selbst erstellt werden, oder es können standardmäßige Aktionen (mehrere Aktivitäten zusammengefasst) von WWF verwendet werden.

Beim Selbsterstellen einer Aktivität kann von verschiedensten Klassen geerbt werden, um deren Funktionalitäten verwenden zu können. Die wichtigsten Basisklassen gemäß [34] sind

- `CodeActivity`: Zum Erstellen einer synchronen benutzerdefinierten Aktivität.
- `AsyncCodeActivity`: Zum Erstellen einer asynchronen benutzerdefinierten Aktivität.
- `CodeActivity<TResult>`: Zum Erstellen einer synchronen benutzerdefinierten Aktivität, die ein Ergebnis zurückliefert.
- `AsyncCodeActivity<TResult>`: Zum Erstellen einer asynchronen benutzerdefinierten Aktivität, die ein Ergebnis zurückliefert.
- `NativeActivity`: Zum Erstellen einer benutzerdefinierten Aktivität, die mit der Windows Workflow Foundation Runtime interagieren kann.
- `NativeActivity<TResult>`: Zum Erstellen einer benutzerdefinierten Aktivität, die mit der Windows Workflow Foundation Runtime interagieren kann und die ein Ergebnis zurückliefert.

Mithilfe dieser Basisklassen kann in Visual Studio eine benutzerdefinierte Aktivität erstellt werden, welche dann in den Workflow eingebunden werden kann. Ist dies erledigt, so kann diese wie jede andere Aktivität verwendet werden.

Benötigt man jedoch keine benutzerdefinierten Aktivitäten, so können die standardmäßigen Aktionen der Windows Workflow Foundation verwendet werden. Diese decken ein großes Spektrum an Anforderungen ab und werden in den Abbildung 7-10 dargestellt.

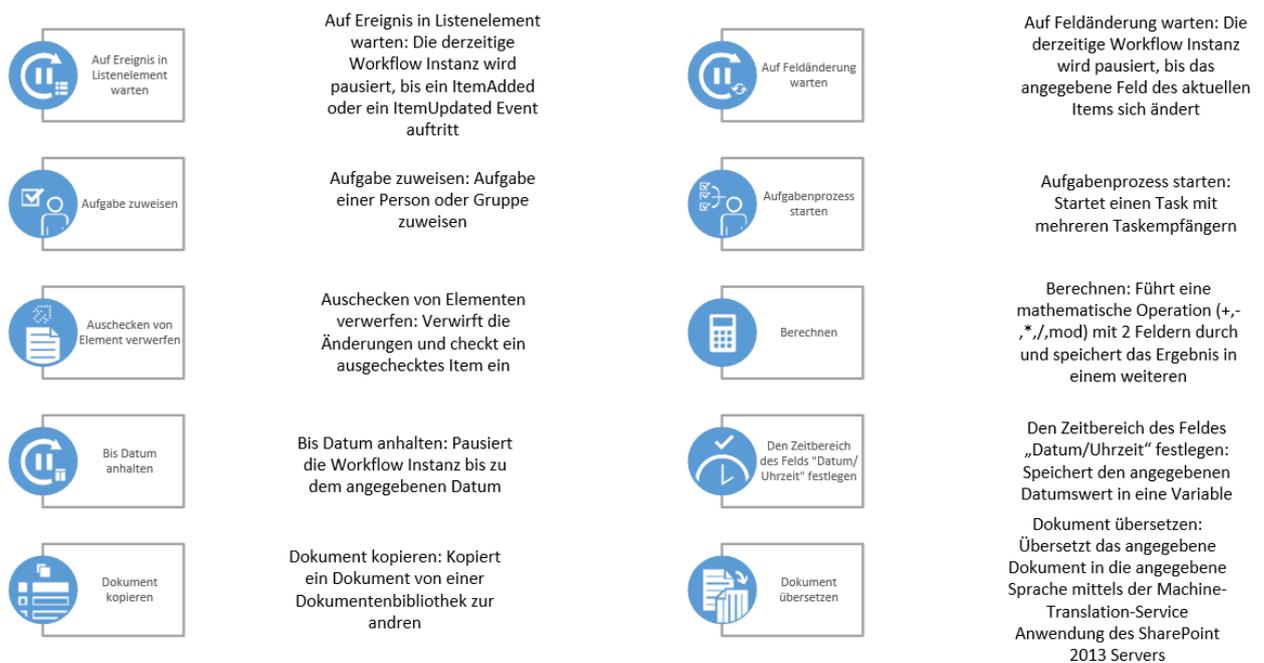


Abbildung 7: WWF-Elemente: Aktionen 1

	Element aus Wörterbuch abrufen: Speichert das angegebene Element des angegebenen Wörterbuchs in eine Variable		Element auschecken: Checkt das angegebene Item der angegebenen Dokumentenbibliothek aus
	Element einchecken: Checkt das angegebene Item der angegebenen Dokumentenbibliothek ein		Element löschen: Löscht das angegebene Item aus der angegebenen Liste
	Elemente im Wörterbuch zählen: Zählt die Anzahl der Elemente im angegebenen Wörterbuch		E-Mail senden: Sendet eine E-Mail an einen User oder eine Gruppe an Usern
	Feld im aktuellen Element festlegen: Legt den Wert des angegebenen Feldes im aktuellen Element fest		Für Dauer anhalten: Pausiert die Workflow Instanz für die angegebene Dauer
	HTTP-Webdienst aufrufen: Ruft einen externen REST Service mittels HTTP auf. GET, POST, PUT und DELETE möglich.		In Verlaufsliste protokollieren: Loggt den angegebenen Text in die History-Liste

Abbildung 8: WWF-Elemente: Aktionen 2

	Intervall zwischen Datumsangaben suchen: Berechnet den Zeitintervall in Tagen, Stunden und Minuten		Kommentar hinzufügen: Ein Kommentar wird auf der Workflowdesign-Oberfläche hinzugefügt
	Listenelement aktualisieren: Aktualisiert die angegebenen Items der angegebenen Liste auf die angegebenen Werte		Listenelement erstellen: Erstellt die angegebenen Items der angegebenen Liste mit den optionalen Default-Werten
	Listenworkflow starten: Startet eine neue Listenworkflow Instanz auf dem angegebenen Item		Teilzeichenfolge ab Anfang der Zeichenfolge extrahieren: Kopiert n Zeichen vom Anfang des Strings in eine Variable
	Teilzeichenfolge ab Ende der Zeichenfolge extrahieren: Kopiert n Zeichen vom Ende des Strings in eine Variable		Teilzeichenfolge ab Index der Zeichenfolge extrahieren: Kopiert alle Zeichen ab dem Index in eine Variable
	Teilzeichenfolge der Zeichenfolge anhand des Index mit bestimmter Länge extrahieren: Kopiert n Zeichen ab dem Index in eine Variable		Teilzeichenfolge in Zeichenfolge ersetzen: Ersetzt einen Substring in einem String durch den angegebenen Wert

Abbildung 9: WWF-Elemente: Aktionen 3

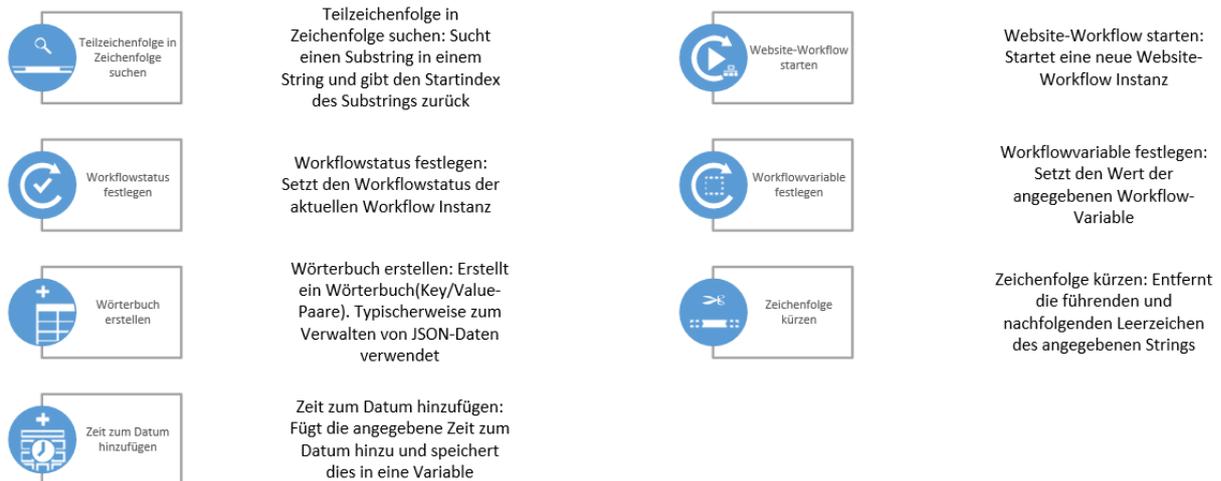


Abbildung 10: WWF-Elemente: Aktionen 4

Um den Sequenzfluss zwischen den Aktivitäten zu steuern, können Bedingungen verwendet werden. WWF bietet neben der weit verbreiteten und aus Programmiersprachen bekannten If-Anweisung noch weitere, die in Abbildung 11 zu sehen sind. Zu beachten ist jedoch, dass die Bedingung, die der If-Anweisung entspricht (wenn ein beliebiger Wert gleich dem Wert ist), nur auf Gleichheit prüfen kann und nicht, ob der Wert größer oder kleiner ist. Weiters ist es nicht möglich, mit einer If-Anweisung mehrere Werte zu prüfen.



Abbildung 11: WWF-Elemente: Bedingungen

Die bisher beschriebenen Aktionen müssen von einer Stufe umschlossen sein, da diese den äußersten Container darstellt. Einzig Bedingungen können außerhalb einer Stufe modelliert werden, um Abbruchbedingungen beziehungsweise Übergänge zwischen ebenen Stufen modellieren zu können. Diese Stufen können nicht geschachtelt werden, jedoch ist es möglich, mehrere Stufen miteinander zu verbinden, um einen Workflow in mehrere Phasen zu unterteilen. Da Stufen nicht geschachtelt werden können, müssen Schritte verwendet werden, um eine logische Trennung innerhalb von Stufen zu erzeugen. Schritte können dabei in beliebiger Anzahl in einer Stufe vorkommen und können ineinander geschachtelt werden. Vor der ersten Stufe des Workflows muss sich immer ein Start-Element befinden, welches den Startpunkt des Workflows darstellt. Zwei weitere Containerobjekte, die dem Schritt-Objekt ähnlich sind, sind die "Schleife mit Bedingung" und die "Schleife n-mal". Diese führen die sich im Container befindlichen Aktivitäten, Aktionen oder Bedingungen aus, bis eine Bedingung erreicht ist oder n-mal. Um parallele Sequenzflüsse zwischen ebenen Containerobjekten, Aktionen und Aktivitäten zu ermöglichen, können die beiden Elemente "parallelen Prozess starten" und "parallelen Prozess beenden" verwendet werden. In Abbildung 12 werden diese Containerelemente grafisch dargestellt und nochmals kurz erklärt.

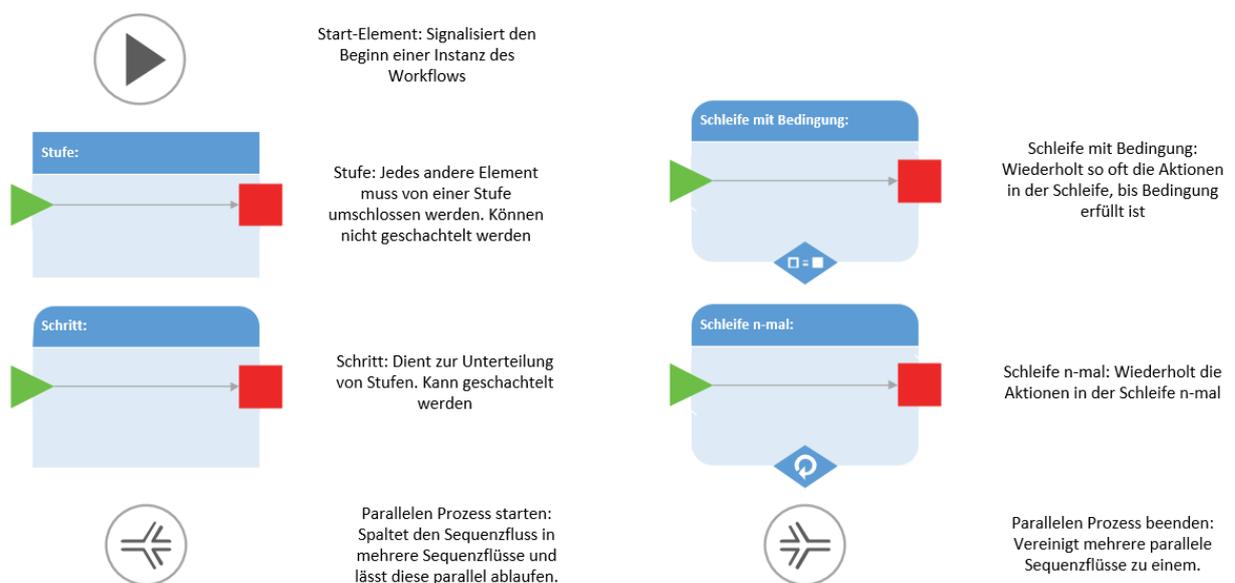


Abbildung 12: WWF-Elemente: Container und Abschlusszeichen

Die eben gezeigten graphischen Darstellungen von Elementen eines Workflows basieren auf einem Dokument, der sogenannten Workflow-Definition, welche in XAML (Extensible Application Markup Language) geschrieben ist. XAML ist gemäß [28] eine XML-basierte Markup-Sprache, die von Microsoft entwickelt wird und ist für die graphische Darstellung des Workflows dasselbe, wie HTML für die graphische Dar-

stellung der Webseite. In Abbildung 13 ist die XAML-Datei des Beispiels aus Kapitel 2.3 dargestellt, um den Aufbau einer XAML-Datei zu zeigen.

```
<Activity mc:Ignorable="mwaw" x:Class="MasterarbeitBeispiel1.MTW" this:MTW.UpdateOderUpgrade="True"
xmlns="http://schemas.microsoft.com/netfx/2009/xaml/activities"
xmlns:local="clr-namespace:Microsoft.SharePoint.WorkflowServices.Activities"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:mwaw="clr-namespace:Microsoft.Web.Authoring.Workflow;assembly=Microsoft.Web.Authoring"
xmlns:scg="clr-namespace:System.Collections.Generic;assembly=mscorlib"
xmlns:this="clr-namespace:MasterarbeitBeispiel1"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <x:Members>
    <x:Property Name="Maschinennummer" Type="InArgument(x:String)" />
    <x:Property Name="UpdateOderUpgrade" Type="InArgument(x:Boolean)" />
    <x:Property Name="responseCode" Type="InArgument(x:String)" />
    <x:Property Name="responseCode1" Type="InArgument(x:String)" />
    <x:Property Name="responseCode2" Type="InArgument(x:String)" />
    <x:Property Name="responseCode3" Type="InArgument(x:String)" />
    <x:Property Name="responseCode4" Type="InArgument(x:String)" />
  </x:Members>
  <Sequence>
    <Sequence>
      <mwaw:SPDesignerXamlWriter.CustomAttributes>
        <scg:Dictionary x:TypeArguments="x:String, x:String">
          <x:String x:Key="InitBlock">InitBlock-7751C281-B0D1-4336-87B4-83F2198EDE6D</x:String>
        </scg:Dictionary>
      </mwaw:SPDesignerXamlWriter.CustomAttributes>
    </Sequence>
    </Sequence>
    <Flowchart StartNode="{x:Reference __ReferenceID0}">
      <FlowStep x:Name="__ReferenceID0">
        <Sequence>
          <mwaw:SPDesignerXamlWriter.CustomAttributes>
            <scg:Dictionary x:TypeArguments="x:String, x:String">
              <x:String x:Key="StageAttribute">StageContainer-8EDBFE6D-DA0D-42F6-A806-F5807380DA4D</x:String>
            </scg:Dictionary>
          </mwaw:SPDesignerXamlWriter.CustomAttributes>
          <local:SetWorkflowStatus Disabled="False" Status="Stage 1">
            <mwaw:SPDesignerXamlWriter.CustomAttributes>
              <scg:Dictionary x:TypeArguments="x:String, x:String">
                <x:String x:Key="StageAttribute">StageHeader-7FE15537-DFDB-4198-ABFA-8AF8B9D669AE</x:String>
              </scg:Dictionary>
            </mwaw:SPDesignerXamlWriter.CustomAttributes>
            </local:SetWorkflowStatus>
            <Sequence DisplayName="Stage 1" />
            <Sequence>
              <mwaw:SPDesignerXamlWriter.CustomAttributes>
                <scg:Dictionary x:TypeArguments="x:String, x:String">
                  <x:String x:Key="StageAttribute">StageFooter-3A59FA7C-C493-47A1-8F8B-1F481143EB08</x:String>
                </scg:Dictionary>
              </mwaw:SPDesignerXamlWriter.CustomAttributes>
            </Sequence>
          </Sequence>
        </FlowStep>
      </Flowchart>
    </Sequence>
  </Activity>
```

Abbildung 13: WWF XAML Datei

2.3 Vergleich und Beispiel

In diesem Kapitel wird anhand eines Beispielprozesses gezeigt, inwieweit sich die Modellierungssprachen BPMN und WWF unterscheiden.

Der Beispielprozess ist ein vereinfachter Teil des praktischen Geschäftsprozesses, welcher in Kapitel 4 beschrieben wird. Zunächst wird dieser textuell beschrieben, danach in BPMN und zuletzt in WWF.

Prozessbeschreibung: Der Kunde möchte ein Update oder ein Upgrade seiner Formgebungsmaschine durchführen. Da er nicht alle Maschinendetails besitzt, möchte er sich über seine Maschine informieren, weshalb er diesen Geschäftsprozess anstößt.

Der Kunde gibt zuerst die Maschinenummer seiner Formgebungsmaschine ein und wählt danach aus, ob er ein Update (eine neue Softwareversion der Maschinensoftware wird installiert) oder ein Upgrade (neue Optionen werden hinzugefügt/entfernt und optional wird eine neue Softwareversion der Maschinensoftware installiert) ausführen möchte. Wählt er Upgrade aus, so werden sowohl die installierten Optionen als auch die verfügbaren Optionen parallel abgerufen. Nachdem beide Aktionen fertig sind, wird eine Optionsliste mit allen Optionen generiert. Wählt der Kunde Update aus, so werden alle Aktionen mit dem Wort Option ignoriert. Danach wird für Update und Upgrade die Maschinenversion abgefragt. Am Ende des Prozesses werden die generierten Informationen angezeigt.

In Abbildung 14 wird das BPMN-Modell für diese Prozessbeschreibung gezeigt, welches in Camunda modelliert wurde:

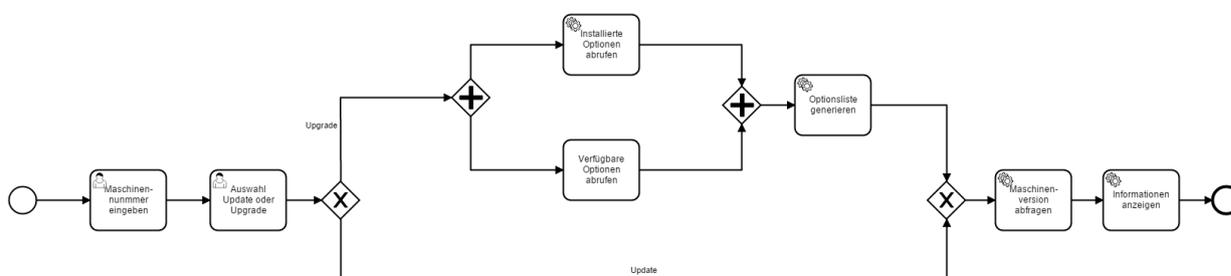


Abbildung 14: BPMN Beispiel-Prozess: Maschineninformationen darstellen

In Abbildung 15 wird das WWF-Modell für diese Prozessbeschreibung gezeigt, welches in SharePoint Designer modelliert wurde:

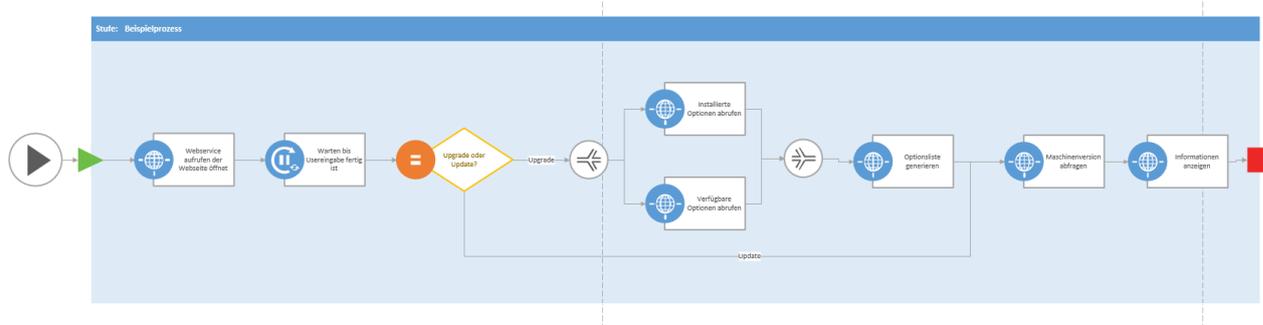


Abbildung 15: WWF Beispiel-Prozess: Maschineninformationen darstellen

Anhand der beiden Abbildungen sind die Unterschiede zwischen den beiden Notationen ersichtlich.

Zunächst ist es in WWF nicht möglich, abstrakte Aktionen, wie "Maschinennummer eingeben" und "Auswahl Update oder Upgrade", effizient und sauber darzustellen, da WWF eine Modellierungssprache mit Fokus auf Ausführbarkeit ist. Um dies jedoch trotzdem abbilden zu können, gibt es zwei verschiedene Ansätze, die verwendet werden können. Der erste Ansatz ist, dass dem User die Eingabe der Informationen vor dem Starten des Geschäftsprozesses ermöglicht wird und die dadurch erhaltenen Daten werden dem Geschäftsprozess als Startparameter übergeben. Dabei würden die ersten beiden Aktionen (Maschinennummer eingeben und Auswahl Update oder Upgrade) des Geschäftsprozesses wegfallen. Bei Variante zwei, welche in Abbildung 15 dargestellt wird, wird als erste Aktion ein Webservice aufgerufen, welcher die Webseite zum Eingeben der Informationen anzeigt. Danach wartet der Workflow, dass der Benutzer die Daten in die Webseite eingibt und diese an ihn via Webservice übergeben werden. Dies geschieht, indem der Webservice mittels der SharePoint-REST-API die Daten in das Listenelement der aktuellen Workflowinstanz schreibt. Die Aktion, die auf den Aufruf des Webservices folgt, wartet auf eine Änderung in jenem Feld, in das geschrieben wird, wodurch der Workflow weiterläuft sobald die Daten durch den Benutzer eingegeben wurden.

Ein weiterer Unterschied zwischen WWF und BPMN ist, dass es in WWF keine unterschiedlichen Gateways gibt. Es gibt lediglich eine If-Anweisung und mehrere Typen von Bedingungen, die Prüfungen bezüglich der Liste, dem Benutzer oder dem Datum durchführen. Gateways wie XOR oder Inklusive sind dort nicht vorhanden.

Weiters gibt es in der Windows Workflow Foundation keine expliziten Events wie bei BPMN. Es gibt zwar Aktionen in WWF, die sich wie BPMN-Events verhalten (BPMN-Zwischenevent-Timer entspricht WWFs "Für Dauer anhalten"-Aktion), diese werden jedoch nicht wie Events in BPMN abgearbeitet.

Die Anzahl der verfügbaren Elemente ist ein weiterer Punkt, indem sich beide Modellierungssprachen stark unterscheiden. Wie bereits erwähnt, gibt es in BPMN über 100 Elemente. In WWF jedoch, ist diese Anzahl unbegrenzt, da man selbst Codeaktionen schreiben und verwenden kann. In graphenbasierten Modellierungssprachen wäre dies undenkbar, da dadurch ein Modell schwer lesbar bis unlesbar wird. In einer blockstrukturierten Modellierungssprache wie WWF ist dies jedoch von Vorteil, da der Fokus auf der Ausführbarkeit liegt und dadurch die Flexibilität der Modellierungssprache stark steigt.

Des Weiteren gibt es das Konzept der Pools und Swimlanes nur in BPMN und nicht in WWF. Dort lässt sich auf reiner Modellierungsebene nicht festlegen, wer für die Ausführung einer Aktion zuständig ist. Dies kann erst bei der Automatisierung festgelegt werden.

3 Auswahl des Workflow-Management-Systems

In diesem Kapitel werden unterschiedliche Workflow-Management-Systeme gezeigt, die sowohl für die Modellierung als auch die Automatisierung eines Geschäftsprozesses in BPMN oder WWF geeignet sind. Weiters werden diese anhand von Kriterien, die aus der Literatur bezogen werden, verglichen. Die Auswahl fiel dabei auf folgende drei Workflow-Management-Systeme:

- Camunda: Wurde ausgewählt, da die vorherige Masterarbeit von Karl-Heinz Stöckler [40] dieses System verwendet.
- Bizagi: Wurde ausgewählt, da es eine Modellierung des Geschäftsprozesses in BPMN ermöglicht, welche sich sehr einfach in eine SharePoint-Umgebung einbinden lässt.
- SharePoint: wurde ausgewählt, da eine SharePoint-Umgebung bereits vorhanden ist, weshalb ein SharePoint-Workflow die naheliegendste Lösung ist.

3.1 Kriterienkatalog

Um ein Workflow-Management-System auswählen zu können, müssen Kriterien definiert werden, anhand welcher die Systeme verglichen werden. Die Kriterien werden dabei in die drei Kategorien Ausschlusskriterien, Workflow-Management-System-spezifische Kriterien und anwendungsspezifische Kriterien unterteilt. Ausschlusskriterien besitzen einen immens hohen Stellenwert und bei Nichterfüllen eines solchen Kriteriums wird das jeweilige System von der restlichen Evaluierung ausgeschlossen. Workflow-Management-System-spezifische Kriterien betreffen die Qualität des Systems und anwendungsspezifische Kriterien beziehen sich auf Eigenschaften, die aufgrund des Anwendungsfalls von Bedeutung sind. Die Kriterien der jeweiligen Kategorie werden bewertet, um deren Bedeutsamkeit für die Auswahl des Workflow-Management-Systems zu spezifizieren. Sowohl die Skala für die Bewertung der Kriterien als auch für die Ausprägung eines Kriteriums eines Workflow-Management-Systems verläuft von 1 bis 10, wobei 1 der schlechteste und 10 der beste Wert ist. Ausschlusskriterien sind sowohl von der Bewertung der Kriterien als auch der Ausprägung des Kriteriums eines Workflow-Management-Systems ausgeschlossen, weil diese aufgrund ihres Stellenwertes nicht bewertet werden müssen und da es nur binäre Antworten, wie ja oder nein, auf diese Kriterien gibt.

3.1.1 Kriterienauswahl

Ausschlusskriterien:

- Graphische Darstellung: Der modellierte Workflow muss im jeweiligen Workflow-Management-System graphisch darstellbar sein. Dabei muss die Qualität der Darstellung ausreichen, um auch komplexe Prozesse verständlich abzubilden.
- Unterstützung durch das Unternehmen: Das Unternehmen Engel muss die Technologie, die das Workflow-Management-System nutzt, verwenden oder unterstützen, da die Implementierung einer neuen Technologie nicht erwünscht ist.

Workflow-Management-System-spezifische Kriterien nach [4], [32] und [50]:

- Benutzerfreundlichkeit: Das Workflow-Management-System soll dem Benutzer die Erstellung eines Workflows so einfach wie möglich machen, durch aufgeräumte und intuitive Oberflächen.
- Wartbarkeit: Workflows sollen einfach wartbar und anpassbar sein. Änderungen sollten problemlos durchführbar sein.

Anwendungsspezifische Kriterien:

- Effizienz der Modellierungssprache: Der abgebildete Workflow sollte dem Geschäftsprozess so genau wie möglich entsprechen. Weiters sollten so wenig Elemente wie möglich dafür verwendet werden, um die Komplexität gering zu halten.
- Implementierung: Die Implementierungskosten des Workflows in die vorhandene Windows-Umgebung der Firma Engel sollten so gering wie möglich sein.
- Erweiterbarkeit: Die Erweiterung des Workflows durch benutzerdefinierte Aktionen sollte möglich sein. Des Weiteren ist die Kommunikation mit externen Systemen wichtig.

3.1.2 Kriterienbewertung

Um die Bedeutung der einzelnen Kriterien für die Auswahl des Workflow-Management-Systems festzulegen, müssen diese bewertet werden. Die beiden Ausschlusskriterien werden dabei nicht bewertet, da

diese eine Voraussetzung, welche erfüllt oder nicht erfüllt wird, darstellen. In Tabelle 5 werden die zu bewertenden Kriterien gezeigt.

Kriterium	Bewertung	Begründung
Benutzerfreundlichkeit	3/10	Da in diesem Anwendungsfall nur eine kleine Gruppe an Benutzern existiert, die mit dem Workflow-Management-System interagieren, und diese gut geschult sind, ist die Benutzerfreundlichkeit nicht von besonderer Bedeutung.
Wartbarkeit	6/10	Da der Optionstore jene Teile enthält, die sich regelmäßig ändern, ist der Workflow selbst nicht von regelmäßigen Änderungen betroffen. Deshalb hat die Wartbarkeit nicht allerhöchste Priorität.
Effizienz	9/10	Aufgrund der häufigen Ausführung des Workflows, ist die Effizienz der Modellierung von hoher Bedeutung.
Implementierung	8/10	Eine nahtlose Integration in die bestehende SharePoint-Umgebung ist von hoher Priorität, da dadurch die Akzeptanz der Benutzer erhöht wird.
Erweiterbarkeit	10/10	Der Workflow kommuniziert hauptsächlich mit heterogenen Systemen (Optionstore, Testservice und Dokumentationservice), weshalb die Erweiterbarkeit maximale Priorität genießt.

Tabelle 1: Bewertung der Kriterien zur Auswahl des Workflow-Management-Systems

3.2 Überblick über die Workflow-Management-Systeme

In den nachfolgenden Kapiteln werden die unterschiedlichen Workflow-Management-Systeme beschrieben.

3.2.1 Camunda

Camunda[8] ist eine Open-Source-Plattform für Workflow- und Entscheidungsautomation, welche BPMN für die Workflowautomation und DMN[5] für die Entscheidungsautomation unterstützt. Der Camunda-Technologiestack besteht aus folgenden Elementen:

- **Workflow Engine:** Führt die modellierten BPMN-Prozesse aus und unterstützt dabei nahezu alle Symbole des BPMN 2.0 Standards. Weiters kann die Workflow Engine via REST-API gestartet und als REST-Service verwendet oder in eine Java-Anwendung direkt eingebunden werden.
- **Decision Engine:** Führt die definierten Entscheidungstabellen aus. Die Decision Engine ist in die Workflow Engine eingebettet, kann aber auch alleine via der REST-API gestartet und als REST-Service verwendet oder in eine Java-Anwendung direkt eingebunden werden.
- **Modeler:** Ermöglicht das Modellieren von BPMN-Prozessen und DMN-Entscheidungstabellen. Diese können direkt nach dem Modellieren von der Workflow Engine beziehungsweise der Decision Engine ausgeführt werden.
- **Tasklist:** Ermöglicht Benutzern das Überblicken und Abarbeiten von Aufgaben, die sie durch die Ausführung des Workflows zugeteilt bekommen. Dieses User-Interface kann entweder direkt von Camunda übernommen oder selbst gestaltet werden.
- **Cockpit:** Erlaubt das Überwachen der Workflows und Entscheidungstabellen im produktiven Betrieb.
- **Admin:** Erlaubt das Organisieren der Benutzer in Gruppen und das Regeln der Zugriffsrechte auf Benutzer-, als auch auf Gruppenebene. Die Zugriffsrechte wirken auf Benutzer unabhängig der Zugriffsart (Java-Anwendung oder REST-API).
- **Optimize:** Erlaubt es Benutzern, Reports, Dashboards und Alarmer, die beim Überschreiten eines Schwellwertes auslösen, zu erstellen.

All jene Elemente kommunizieren je nach Implementierung mittels REST-API oder Java-API miteinander.

3.2.2 Bizagi

Bizagi[6] ist eine Plattform zum Modellieren, Definieren, Überwachen und Betreiben von Geschäftsprozessen. Die Plattform unterteilt sich in drei Produkte, welche nachfolgend beschrieben werden:

- **Modeler:** Ermöglicht das Modellieren und Optimieren von Geschäftsprozessen in BPMN mit vollständiger Unterstützung aller BPMN-Elemente. Weiters können die Prozess-Modelle simuliert werden, um eine optimale Implementierung zu ermöglichen.

- Studio: Erlaubt das Automatisieren von modellierten Geschäftsprozessen, welche auch im Studio selbst modelliert werden können, mit minimalen Programmieraufwand. Formulare können per Drag&Drop erstellt werden, wodurch Interfaces simpel und flink generiert werden können. Ressourcen und Aufgaben können an Mitarbeiter manuell oder via Auslastungs-Algorithmus verteilt werden. Weiters ist eine einfache Anbindung sowohl an SharePoint als auch SAP mittels SOAP oder REST möglich.
- Engine: Führt die Workflows aus und ermöglicht Benutzern das Ansehen und Abarbeiten von Aufgaben mittels Arbeits-Portal, welches von nahezu allen Endgeräten mit Internetzugang aufgerufen werden kann. Des Weiteren werden Reports und Dashboards automatisch erstellt, welche personalisiert werden können. Eine weitere Funktionalität der Engine ist die Trendprognose, um den optimalen Betrieb zu garantieren.

Zusätzlich können derzeit sowohl der Modeler als auch die Engine in der Cloud betrieben werden. Dies ermöglicht den einfachen Austausch von Geschäftsprozessmodellen und die optimale Skalierung und Nutzung der Engine-Ressourcen.

3.2.3 SharePoint

SharePoint [29] bietet die grundlegendsten Funktionen eines Workflow-Management-Systems an. Alle weiteren Funktionalitäten, die benötigt werden, müssen selbst implementiert werden. Die Workflows selbst basieren auf Windows Workflow Foundation 4, welche wiederum auf den Nachrichtenfunktionen der Windows Communication Foundation [27] basiert. Die Architektur von SharePoint wird in Abbildung 16 gezeigt.

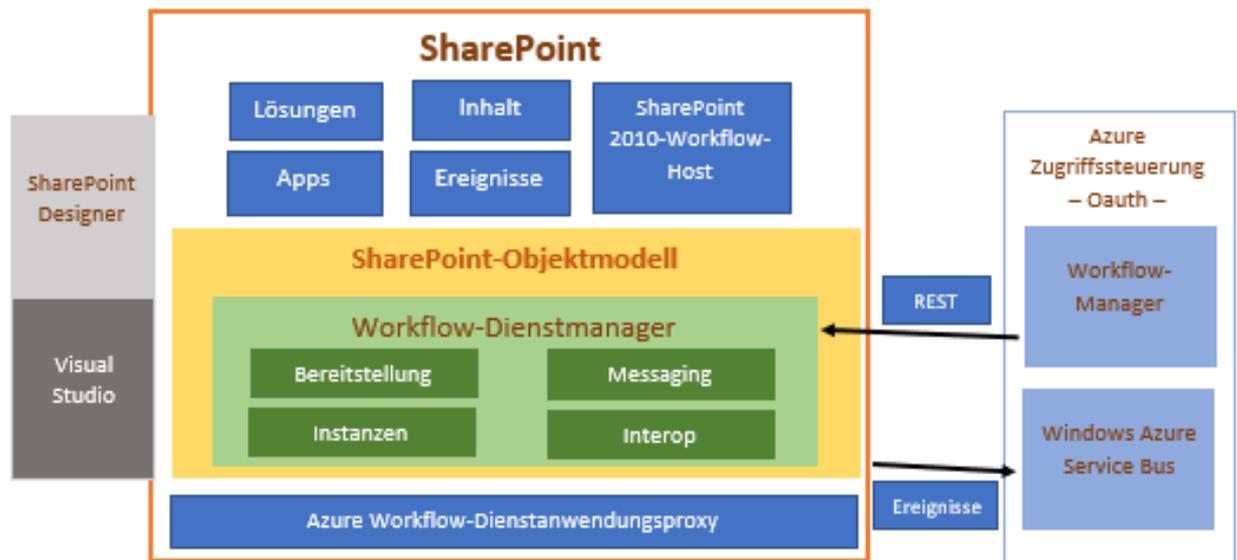


Abbildung 16: SharePoint Architektur nach [29]

Die Ausführung der Workflows findet im Workflow-Manager, außerhalb von SharePoint, statt. Er kommuniziert mit SharePoint mit allgemeinen Protokollen über den Windows Azure Service Bus. Dadurch verbessert sich die Skalierbarkeit, da der Workflow-Manager unabhängig von SharePoint selbst skaliert werden kann. Der Workflow-Dienstmanager ist der Client, mit dem der Workflow-Manager kommuniziert. Dessen Aufgaben sind die Bereitstellung, das Messaging, die Instanzensteuerung und die Interoperabilität mit SharePoint-2010-Workflows. Diese 2010-SharePoint-Workflows werden nicht im Workflow-Manager selbst ausgeführt, welcher nur 2013-Workflows ausführt, sondern im SharePoint 2010-Workflow-Host. Dieser ist weiterhin in SharePoint vorhanden, um die Ausführung alter Workflows zu gewährleisten. Modelliert werden können SharePoint-Workflows entweder mittels dem SharePoint Designer oder Visual Studio. Mit beiden Anwendungen ist es möglich, sowohl 2010-SharePoint-Workflows als auch 2013-SharePoint-Workflows zu erstellen. Weiters ist es in beiden Anwendungen möglich, Workflows graphisch zu modellieren, jedoch ist dies in Visual Studio nicht optimal. Zusätzlich bietet Visual Studio die Möglichkeit, Workflows mittels Code zu definieren sowie das Erweitern von Workflows mit benutzerdefinierten Aktionen.

3.3 Vergleich der Workflow-Management-Systeme anhand des Kriterienkatalogs

Die eben beschriebenen Workflow-Management-Systeme werden nun anhand des Kriterienkatalogs verglichen und es wird eine Auswahl getroffen, welches für diesen Anwendungsfall am besten geeignet ist.

Camunda

Kriterium	Ausprägung	Begründung
Graphische Darstellung	Ja	Graphische Darstellung der Modelle ist ausreichend.
Unterstützung durch das Unternehmen	Nein	Weder verwendet das Unternehmen, noch unterstützt es die Technologie von Camunda, weshalb die Implementierung und Wartung des Systems zu einem hohen Aufwand führen würde.

Tabelle 2: Ausprägung der Kriterien für Camunda

Aufgrund des nicht Erfüllens der Ausschlusskriterien, wird Camunda nicht als Auswahloption für diesen Anwendungsfall in Betracht gezogen.

Bizagi

Kriterium	Ausprägung	Begründung
Graphische Darstellung	Ja	Graphische Darstellung der Modelle ist ausreichend.
Unterstützung durch das Unternehmen	Ja	Durch die einfache Implementierung in SharePoint und SAP würde die Implementierung beziehungsweise Wartung zu einem minimalen Mehraufwand führen.
Benutzerfreundlichkeit	10/10	Bizagi ist ein klar strukturiertes Workflow-Management-System, welches die einzelnen Bereiche des Workflow-Managements in Prozessmodellierung, Datenmodellierung, Formdesign, Geschäftsregelmodellierung, Aufgabenzuweisung, Integration mit heterogenen Systemen und Ausführung unterteilt. Durch diese Unterteilung und die aufgeräumten Oberflächen ist die Verwendung simpel und benutzerfreundlich.
Wartbarkeit	8/10	Aufgrund der Trennung in einzelne Bereiche sind Wartung und Änderungen einfach und effizient durchzuführen.
Effizienz	9/10	Trotz der Ineffizienz von BPMN im Bezug auf die Automatisierung von Geschäftsprozessen, ist Bizagis-Engine höchst effizient und skalierbar.
Implementierung	6/10	Die Integration in eine SharePoint-Umgebung funktioniert gut, jedoch muss ein eigenständiges System implementiert werden, welches zu Kosten führt.
Erweiterbarkeit	8/10	Bizagi lässt sich unkompliziert an andere heterogene Systeme anbinden. Jedoch ist es nicht möglich, den Workflow selbst um neue Aktionen zu erweitern.

Tabelle 3: Ausprägung der Kriterien für Bizagi

SharePoint

Es werden sowohl der SharePoint Designer als auch Visual Studio als Modellierungstools in Betracht gezogen, da der Austausch von Workflows zwischen den Tools, wie hier beschrieben [20], möglich ist.

Kriterium	Ausprägung	Begründung
Graphische Darstellung	Ja	Graphische Darstellung der Modelle ist beim SharePoint Designer ausreichend. Dies ist bei Visual Studio nicht der Fall, weshalb das Tool nicht zur Modellierung geeignet ist.
Unterstützung durch das Unternehmen	Ja	SharePoint wird im Unternehmen bereits verwendet.
Benutzerfreundlichkeit	2/10	Die Oberfläche des Designers ist zwar aufgeräumt und übersichtlich, jedoch wird dem Benutzer kaum Hilfe geboten. Weiters ist der Workflow Designer teilweise instabil und stürzt ab.
Wartbarkeit	7/10	Die Anpassung eines Workflows funktioniert einfach und schnell. Weiters kann die darunterliegende Datenstruktur angepasst werden, ohne dass der Workflow dadurch beeinflusst wird. Eine Trennung, wie bei Bizagi, gibt es nicht.
Effizienz	9/10	Die Notation der Windows Workflow Foundation ist für die Automatisierung von Workflows ausgelegt, weshalb diese äußerst Effizient ist.
Implementierung	10/10	Es besteht bereits eine SharePoint-Umgebung, in welche die neuen Workflows implementiert werden können.
Erweiterbarkeit	10/10	SharePoint erlaubt sowohl die einfache Anbindung an heterogene Systeme mittels Webservice-Calls als auch die Erweiterung von Workflows mittels benutzerdefinierten Aktionen, die in Visual Studio geschrieben wurden.

Tabelle 4: Ausprägung der Kriterien für SharePoint

Workflow-Management-System-Auswahl

Aufgrund der Ausschlusskriterien fiel Camunda aus der Auswahl und die Entscheidung fiel zwischen Bizagi und SharePoint. Es ist dabei zu beachten, dass bei SharePoint nur der Designer als Modellierungstool geeignet ist und nicht Visual Studio, da dessen graphische Darstellung der Workflows unzureichend für diesen Anwendungsfall ist.

Die Wahl fiel aufgrund der hohen Ausprägungen in den Bereichen Effizienz, Implementierung und Erweiterbarkeit auf SharePoint und den SharePoint Designer. Dieser bietet zwar keine hohe Benutzerfreundlichkeit wie Bizagi, jedoch ist diese in diesem Anwendungsfall nicht von hoher Bedeutung.

4 Implementierung des Workflows

Im folgenden Kapitel wird die praktische Implementierung der Aufgabenstellung aus Kapitel 1.1 mit dem ausgewählten Workflow-Management-System aus Kapitel 3 gezeigt. Zunächst wird in Kapitel 4.1 der BPMN-Prozess der vorhergehenden Masterarbeit von Karl Heinz Stöckler [40] gezeigt und erklärt und es wird dieser in WWF modelliert. In Kapitel 4.2 werden jene Anforderungen dargelegt, die sich seit der Masterarbeit im Jahr 2015 geändert haben und warum sich diese geändert haben. In Kapitel 4.3 wird der neu modellierte Geschäftsprozess in WWF dargestellt und beschrieben. Weiters werden wichtige Schnittstellen beziehungsweise Komponenten erklärt.

4.1 Vorhergehendes Prozessmodell in BPMN und WWF modelliert

Um die Änderungen des Workflows zu zeigen, die sich durch den Wechsel von BPMN zu WWF ergeben haben, wird in diesem Kapitel zunächst der Workflow der vorhergehenden Masterarbeit [40] dargestellt und beschrieben. Darauffolgend wird dieser BPMN-Workflow als ein WWF-Workflow modelliert und es werden die daraus entstehenden Änderungen begründet.

4.1.1 Modellierung in BPMN

Die Aufgabenstellung der Masterarbeit [40] war das teilweise Implementieren des im Patent AT 514527 A1 2015-01-15 [15] beschriebenen Verfahrens zum Nachrüsten von Komponenten für Formgebungsmaschinen. Der Kunde sollte dabei die Möglichkeit haben, mittels einer App-Store-ähnlichen Plattform,

Informationen über mögliche Zusatzoptionen für seine Formgebungsmaschine zu erhalten und diese auch zu bestellen. Zusatzoptionen können dabei sowohl Softwarekomponenten für die Steuerung der Formgebungsmaschine als auch Hardwarekomponenten entsprechen.

Um diese Aufgabenstellung zu lösen, wurde von Karl-Heinz Stöckler ein Workflow in BPMN modelliert und automatisiert, welcher die Nach- und Aufrüstung von Komponenten für Formgebungsmaschinen ermöglicht. Dieser Workflow wird in den Abbildungen 17-19 und die jeweiligen Subprozesse in den Abbildungen 20-25 gezeigt, welche nachfolgend beschrieben werden.

Im Workflow gibt es die drei Lanes CSD-VET (die Vertriebsabteilung der Firma Engel), System und Benutzer, welcher ein interner (Mitarbeiter) oder ein externer (Kunde) Benutzer sein kann. Um den Workflow zu starten, muss sich der Benutzer in der Eingabemaske einloggen und dort die Fabrikationsnummer (6-stelliger Code, der eine Formgebungsmaschine identifiziert) eingeben. Nachdem diese Informationen eingegeben wurden, wird die Sicherheitsprüfung durchgeführt, welche in Abbildung 20 gezeigt wird.

Dabei wird zwischen internen und externen Benutzern unterschieden und es wird geprüft, ob die getätigten Eingaben korrekt sind. Ist dies der Fall, so läuft der Workflow weiter wie geplant. Ist dies nicht der Fall, so wird eine Fehlermeldung am Display des internen Benutzers angezeigt oder die CSD-VET-Abteilung wird kontaktiert, um mit dem externen Benutzer Kontakt aufzunehmen. In beiden Fällen wird der Workflow danach abgebrochen.

Wurden die Benutzerdaten korrekt eingegeben, so wird die Optionsliste für die eingegebene Fabrikationsnummer generiert, was in Abbildung 21 dargestellt wird. Dabei wird zum einen die Liste der installierten Optionen einer Formgebungsmaschine und zum anderen die Liste der installierbaren Optionen vom System abgefragt. Parallel dazu werden die Preise zu den installierbaren Optionen von der CSD-VET-Abteilung bezogen. Nach dem Abschluss aller drei Aktionen werden diese drei Optionslisten zu einer kombiniert, welche sowohl Hardware- als auch Softwareoptionen umfasst.

Der Benutzer bekommt ebenjene Optionsliste angezeigt und kann daraus auswählen, welche Optionen er zu seiner Formgebungsmaschine hinzufügen möchte. Nach der Auswahl evaluiert das System die Optionen und prüft, ob diese kostenpflichtig sind. Sind die ausgewählten Optionen kostenpflichtig und handelt es sich nicht um einen Mitarbeiter (interner Benutzer), so muss ein Angebot generiert werden.

Bei der Angebotsgenerierung, welche in Abbildung 22 gezeigt wird, wird zuerst geprüft, ob sowohl Lieferzeit als auch Preis für die ausgewählten Optionen vorhanden sind. Ist dies der Fall, so kann der Benutzer die Angebotsanfrage direkt stellen. Ist dies nicht der Fall, so muss der Benutzer zunächst eine Preisanfrage an die CSD-VET-Abteilung stellen, welche diese bearbeitet und den Preis und die Lieferzeit der Optionen zurücksendet, bevor eine Angebotsanfrage gestellt werden kann. Ein Sonderfall tritt ein, wenn die Optio-

nen des Benutzers keine Standardoptionen sind, beziehungsweise wenn der Benutzer einen Sonderwunsch hat. Ist dies der Fall, so muss die CSD-VET-Abteilung zunächst den Sonderwunsch bearbeiten, bevor die Angebotsanfrage des Benutzers erfolgen kann. In allen drei Fällen wird nach der Angebotsanfrage des Benutzers das Angebot von der CSD-VET-Abteilung im System angelegt.

Nach dem Abschluss der Angebotsgenerierung wird die Bestellung der Optionen durch den Benutzer ausgelöst. Darauffolgend wird dem Benutzer eine Bestellbestätigung angezeigt, welche die bestellten Optionen inklusive Preise enthält. Als nächster Schritt wird vom System geprüft, ob es sich um einen externen Benutzer handelt beziehungsweise ob die Optionen nicht kostenlos sind. Trifft dies zu, so muss die CSD-VET-Abteilung das Angebot erneut prüfen. Handelt es sich um einen internen Benutzer oder sind die Optionen kostenlos, so kann dieser Schritt übersprungen werden. Danach wird geprüft, ob sich unter den Optionen Hardware-relevante Optionen befinden, da dies eine Prüfung durch die CSD-VET-Abteilung auslöst, ob der ePlan (Konfiguration, die die Zuordnung zwischen Hardware Ein-/Ausgängen und den Software-Variablen festlegt) geändert werden muss. Im Falle von reinen Software-Optionen ohne Hardwarerelevanz kann auch dieser Schritt übersprungen werden. Im darauffolgenden Schritt wird abermals geprüft, ob Optionen Hardware- beziehungsweise Software-relevant sind. Sind diese Hardware-relevant, so wird ein ePlan-Mapping von der CSD-VET-Abteilung erstellt und die Hardwarekomponenten werden via Einkauf, Lager oder Produktion besorgt. Im Falle von Software-relevanten Optionen wird geprüft, ob es sich um ein Sonderprojekt (Formgebungsmaschine wurde nach Kundenwünschen angepasst) oder zu installierende Sonderoptionen handelt. Trifft dies zu, so wird ein Sonderauftrag der CSD-VET-Abteilung ausgelöst. Ist die Formgebungsmaschine eine Standardmaschine (ohne kundenspezifische Anpassungen) und handelt es sich um Standardoptionen, so wird geprüft, ob die Optionen ePlan-relevant sind. Wenn die Optionen ePlan-relevant sind, muss bis zu dessen Fertigstellung gewartet werden. Sind diese nicht ePlan-relevant, so kann dies übersprungen werden. Im nachfolgenden Schritt wird die Software für die Steuerung der Formgebungsmaschine mit den neuen Optionen generiert, was in Abbildung 23 und 24 zu sehen ist.

Beim Generieren der neuen Software wird zunächst geprüft, ob die Änderungen relevant für die Dokumentation sind. Trifft dies zu, so wird ein Dokumentationsauftrag für die neue Software freigegeben. Weiters wird geprüft, ob die Änderungen sicherheitsrelevante Optionen betreffen, da dies den Druck der Dokumentation auf Papier zufolge hat. Eine elektronische Variante der Dokumentation (eHelp) wird immer erzeugt. Das Laden der Final-Sicherung aus SAP ist unabhängig davon, ob die Optionsänderungen Dokumentations-relevant sind und wird in jedem Fall ausgeführt. Nach dem Laden der Final-Sicherung wird die Projektumgebung, die für das Generieren der Software nötig ist, vorbereitet. Danach werden

die bereits installierten Optionen der Formgebungsmaschine aus der Final-Sicherung geholt und um die neuen Optionen erweitert. Mittels dieser neuen Optionsliste werden aus SAP die Source-Dateien geladen, welche für das Generieren der neuen Software benötigt werden. Nach dem Laden der Source-Dateien kann das Projekt erstellt und kompiliert werden. Der nächste Schritt ist das Generieren des Targets, bei welchem ein Update-Paket mithilfe des kompilierten Projektes und der erzeugten Dokumentation, wenn vorhanden, erstellt wird. Zum Abschluss der Software-Generierungs-Sequenz wird parallel zueinander die Final-Sicherung gespeichert, die Source-Sicherung gespeichert und das Update-Paket auf einen FTP-Server hochgeladen. Falls die Optionen sowohl Hardware- als auch Software-relevant sind, so werden die Aktionen beider Sequenzflüsse ausgeführt.

Nach dem Fertigstellen der Softwaregenerierung und der optionalen Bestellung der Hardwarekomponenten wird ein automatischer Test der neu generierten Software durchgeführt. Nach diesem Softwaretest wird die Software ausgeliefert.

Bei der Auslieferung wird, wie in Abbildung 25 zu sehen ist, als Erstes geprüft, ob die neuen Optionen nur Software-relevant und nicht Sicherheits-relevant sind. Trifft dies zu, so kann das Update-Paket per Datenfernübertragung (DFÜ) gesendet werden. Trifft dies nicht zu, so wird die Software/Hardware physisch ausgeliefert und es muss festgestellt werden, ob die Software/Hardware dringlich benötigt wird. Bei einem dringlichen Auftrag wird per Expressdienst versendet, sonst per Post-Sammelsendung.

Nach dem Ausliefern der Software/Hardware wird geprüft, ob die Bestellung fehlerfrei verarbeitet und versandt werden konnte. Im Fehlerfall wird der Benutzer darüber informiert, dass ein Problem aufgetreten ist und der Workflow wird beendet. Im Erfolgsfall wird der Benutzer über den Versand informiert und muss die Installation der neuen Version bestätigen, damit das System dies im SAP vermerken kann. Nach dem Vermerken der Installation der neuen Software durch den Benutzer ist das System auf dem aktuellen Stand und der Workflow wird beendet.

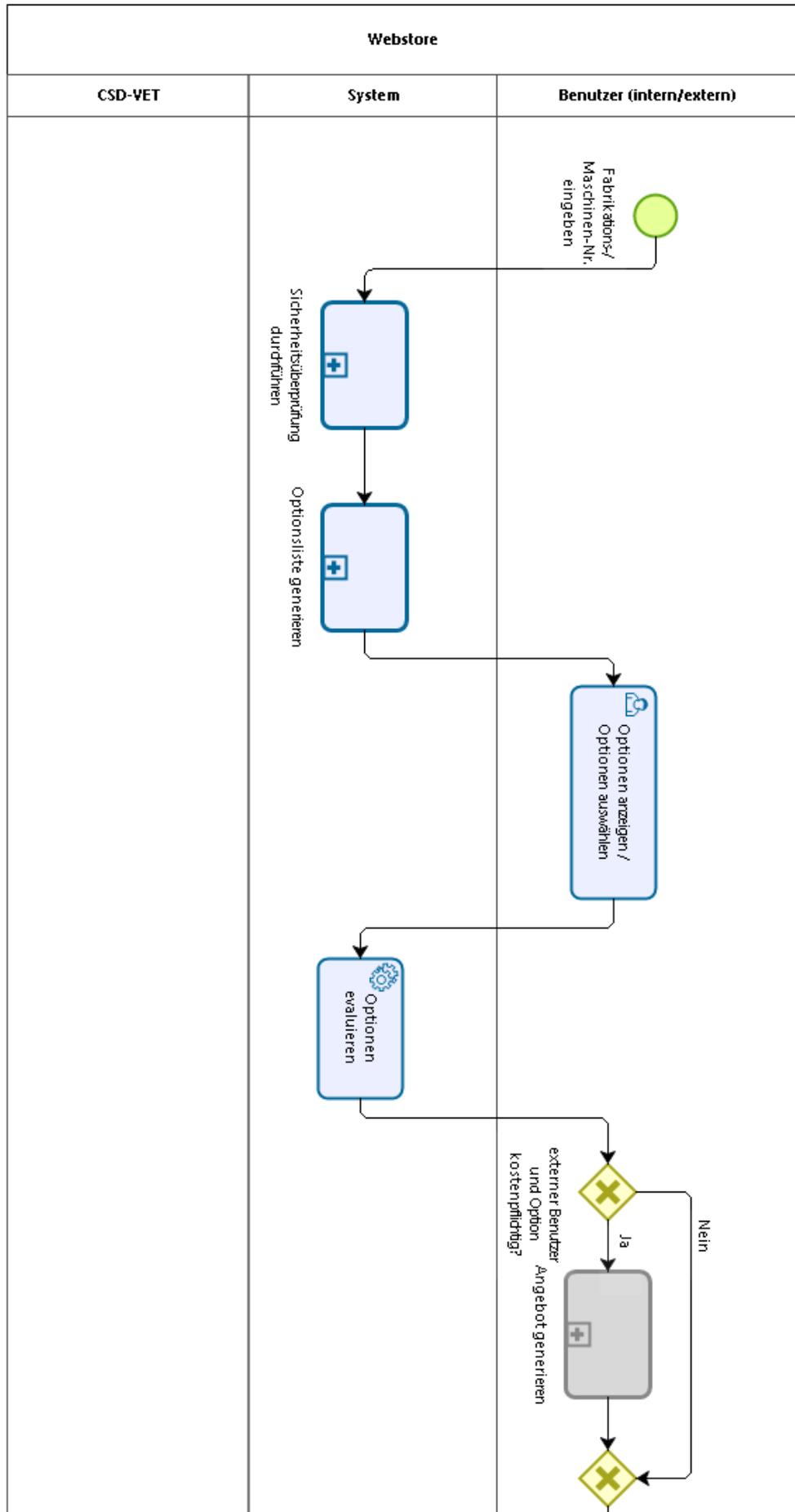


Abbildung 17: BPMN Workflow 1 nach [40]

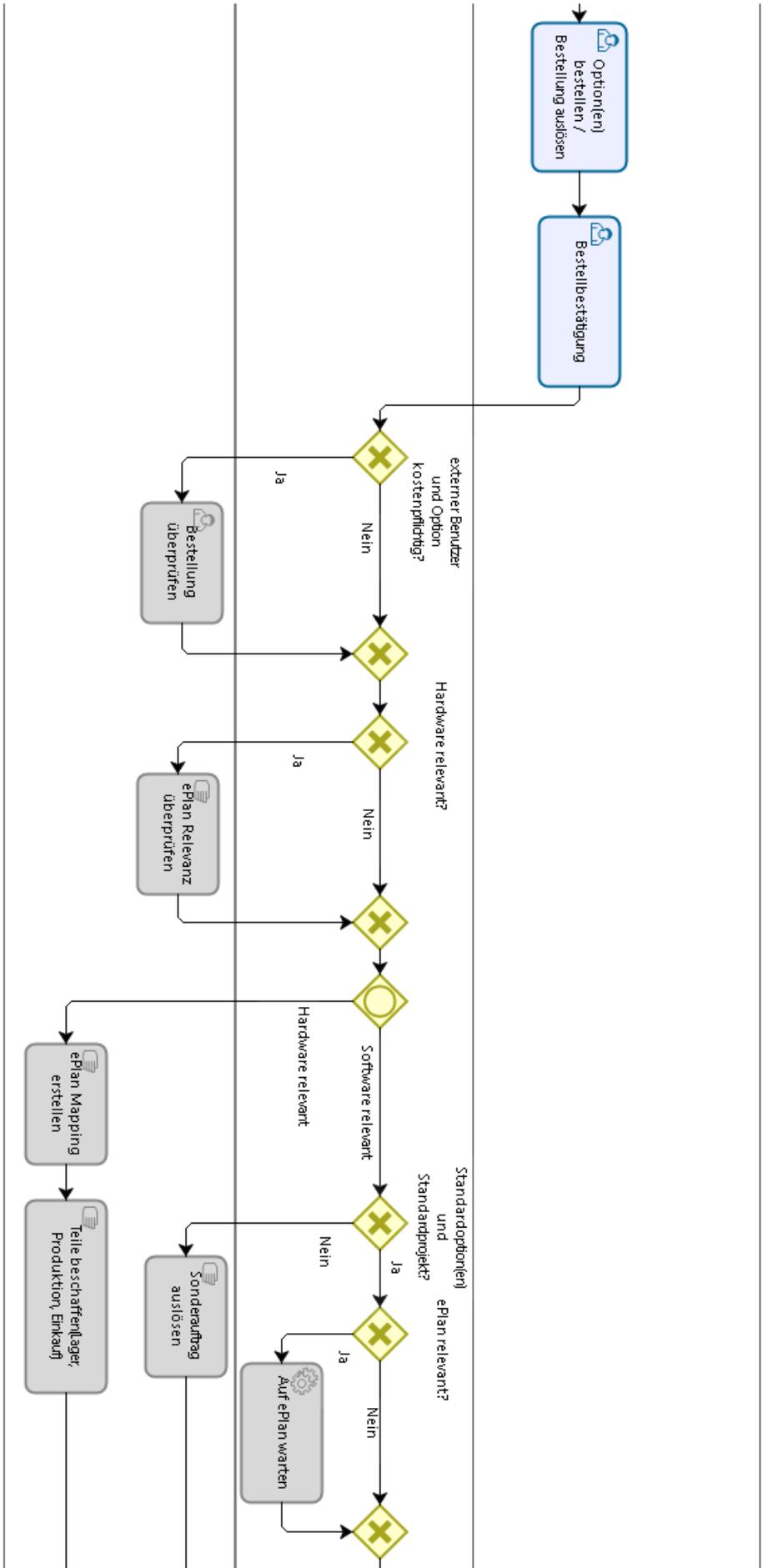


Abbildung 18: BPMN Workflow 2 nach [40]

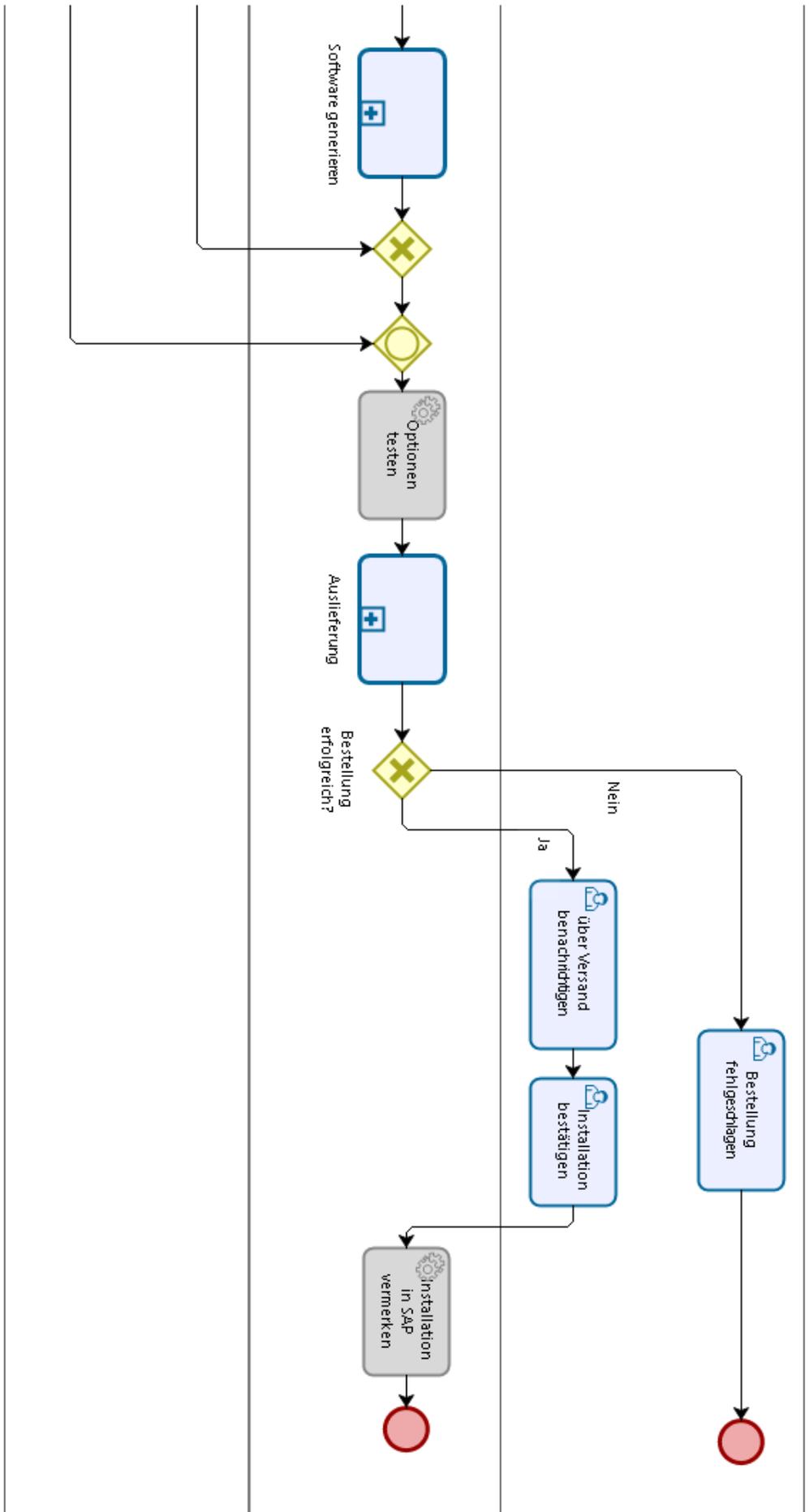


Abbildung 19: BPMN Workflow 3 nach [40]

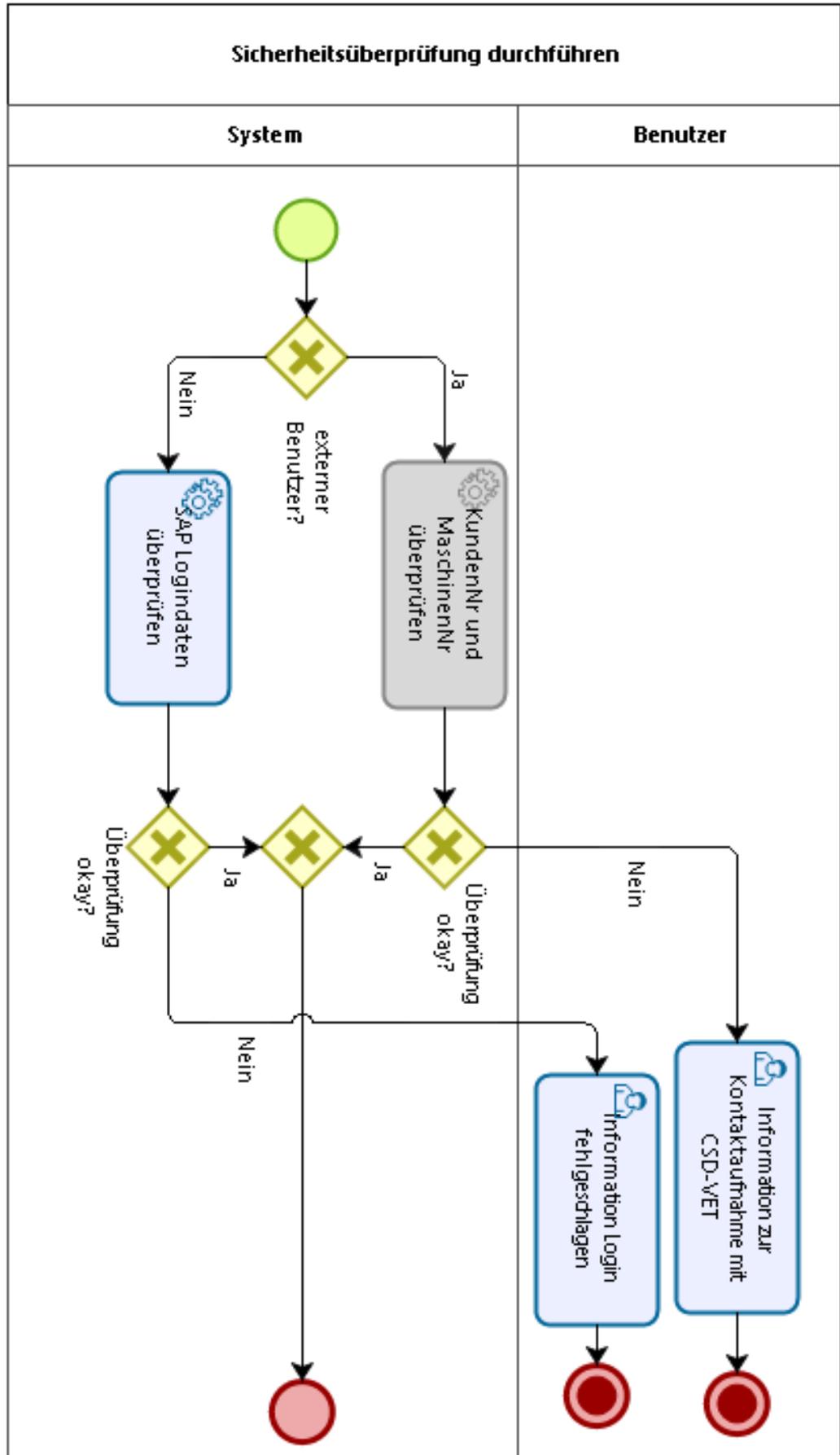


Abbildung 20: BPMN Workflow Sicherheitsüberprüfung nach [40]

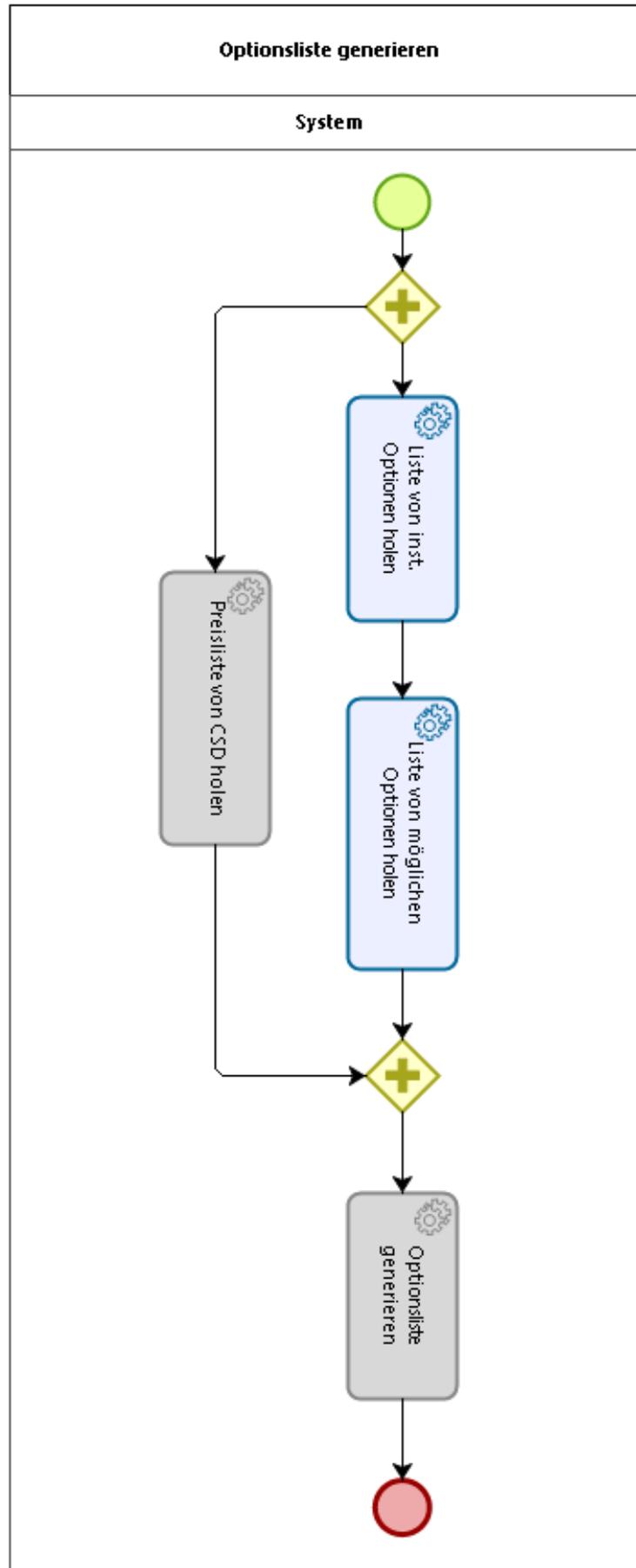


Abbildung 21: BPMN Workflow Optionsliste generieren nach [40]

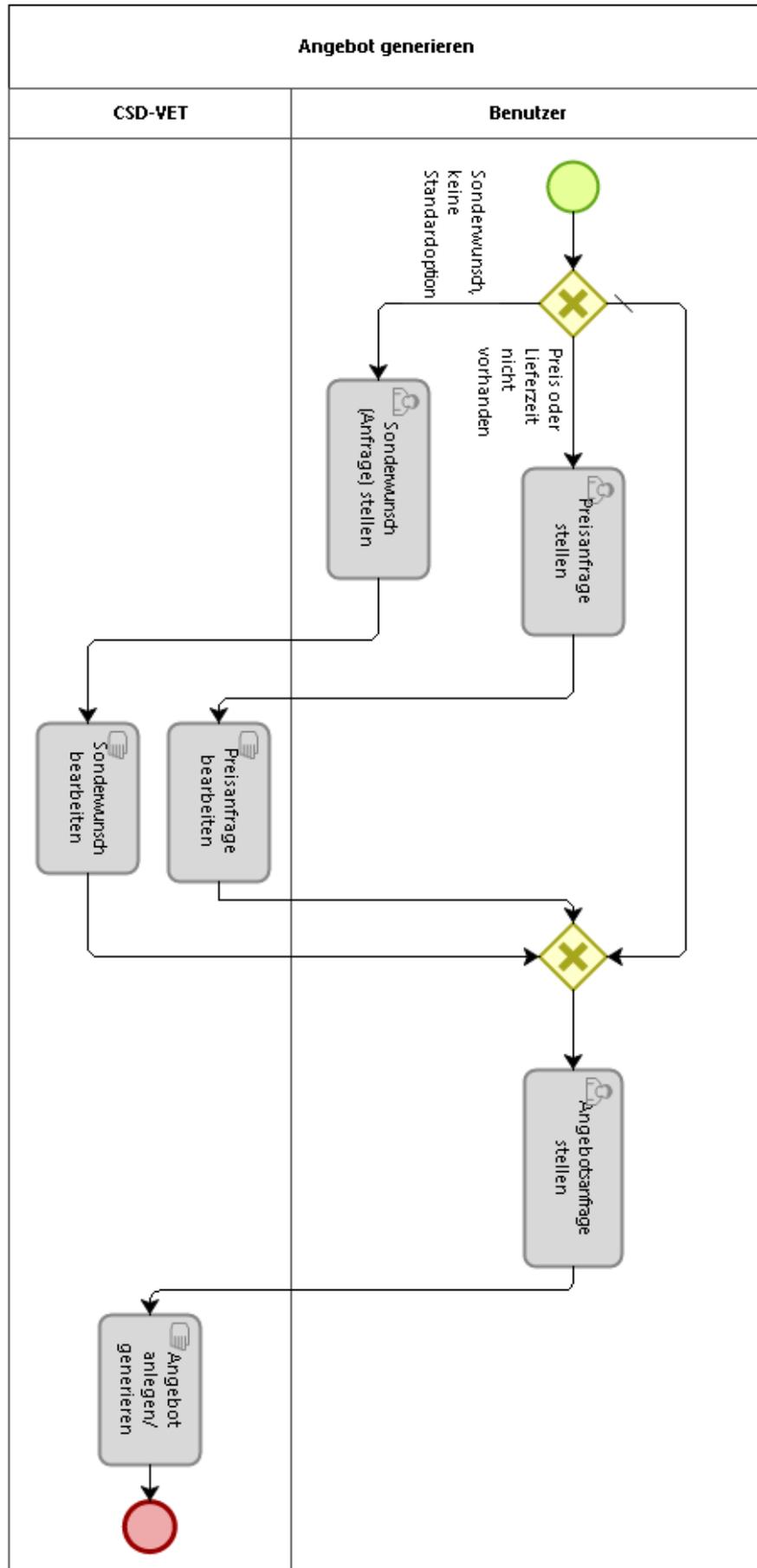


Abbildung 22: BPMN Workflow Angebot generieren nach [40]

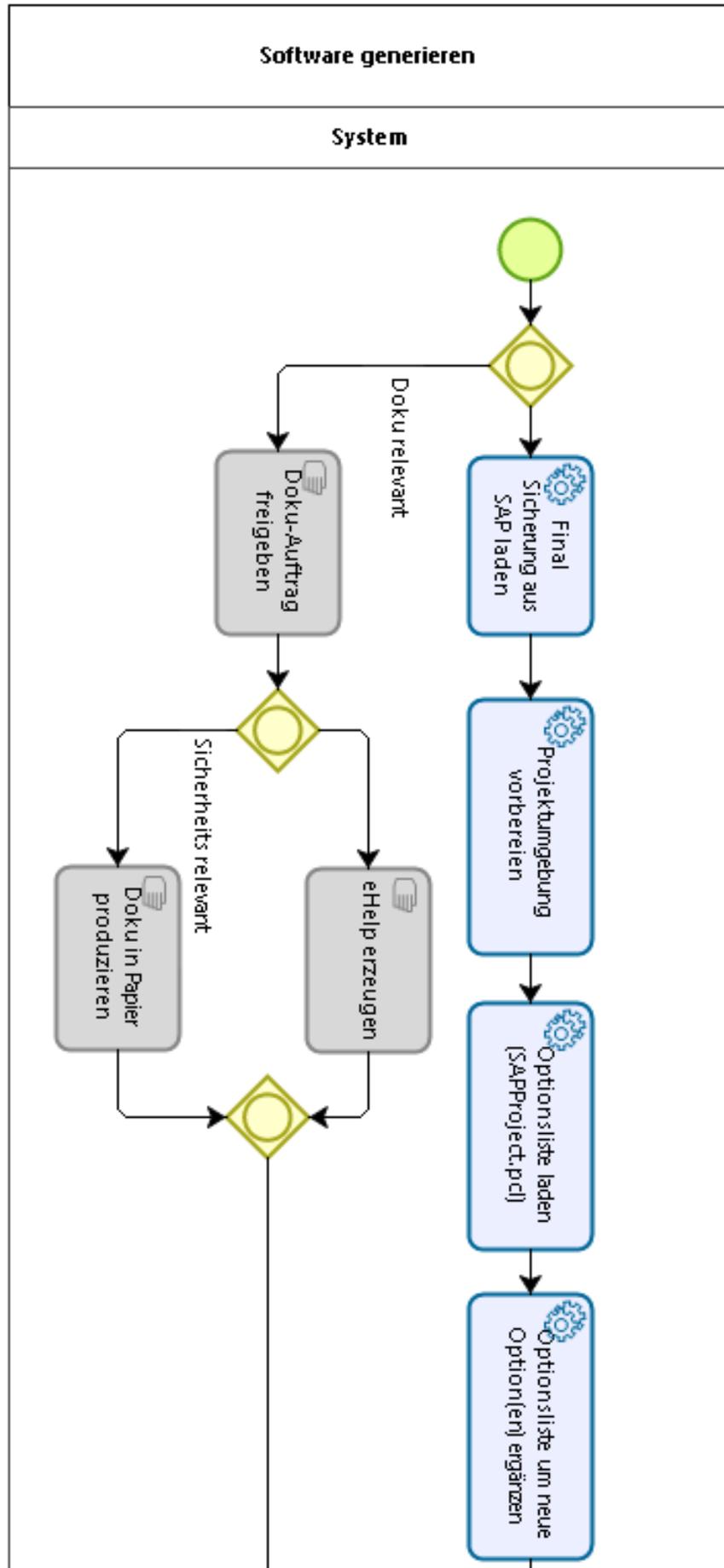


Abbildung 23: BPMN Workflow Software generieren 1 nach [40]

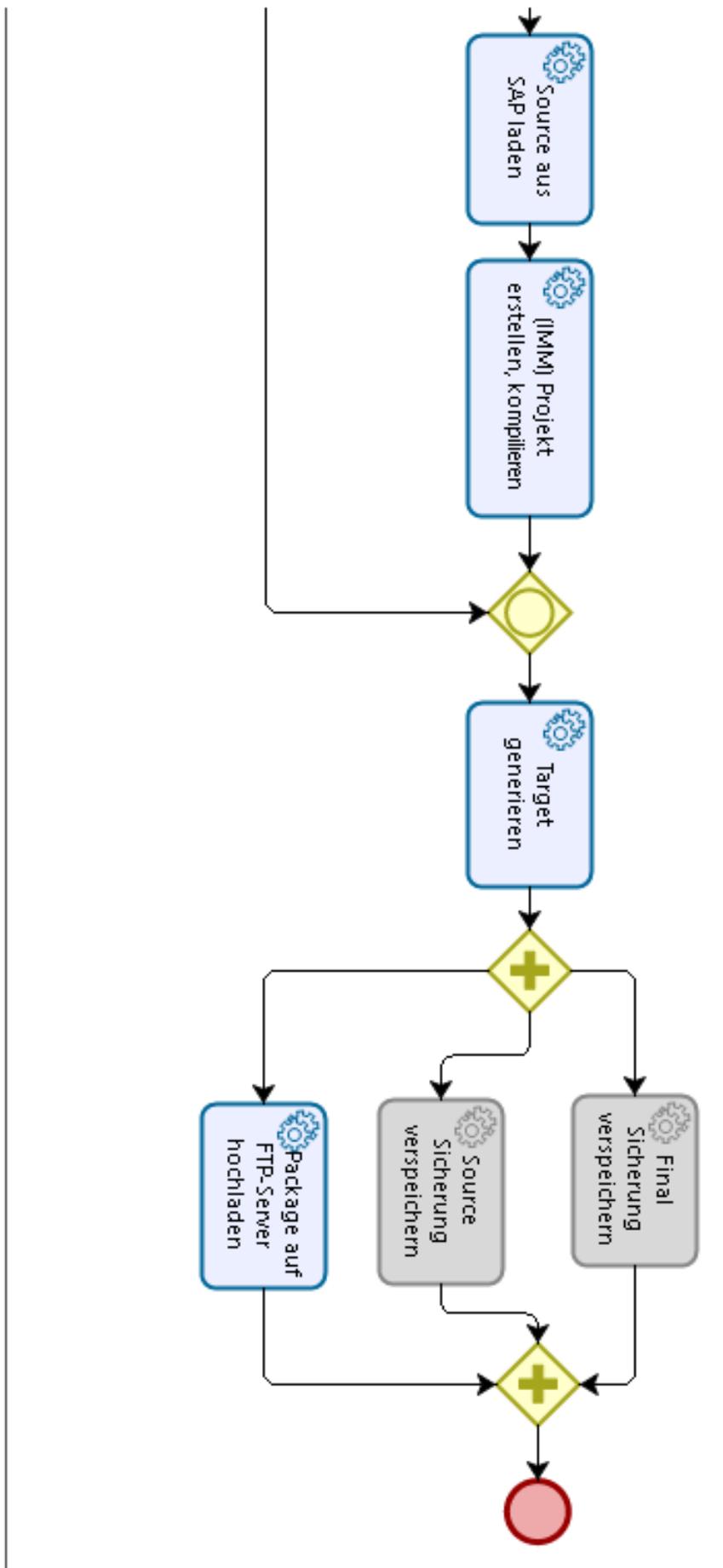


Abbildung 24: BPMN Workflow Software generieren 2 nach [40]

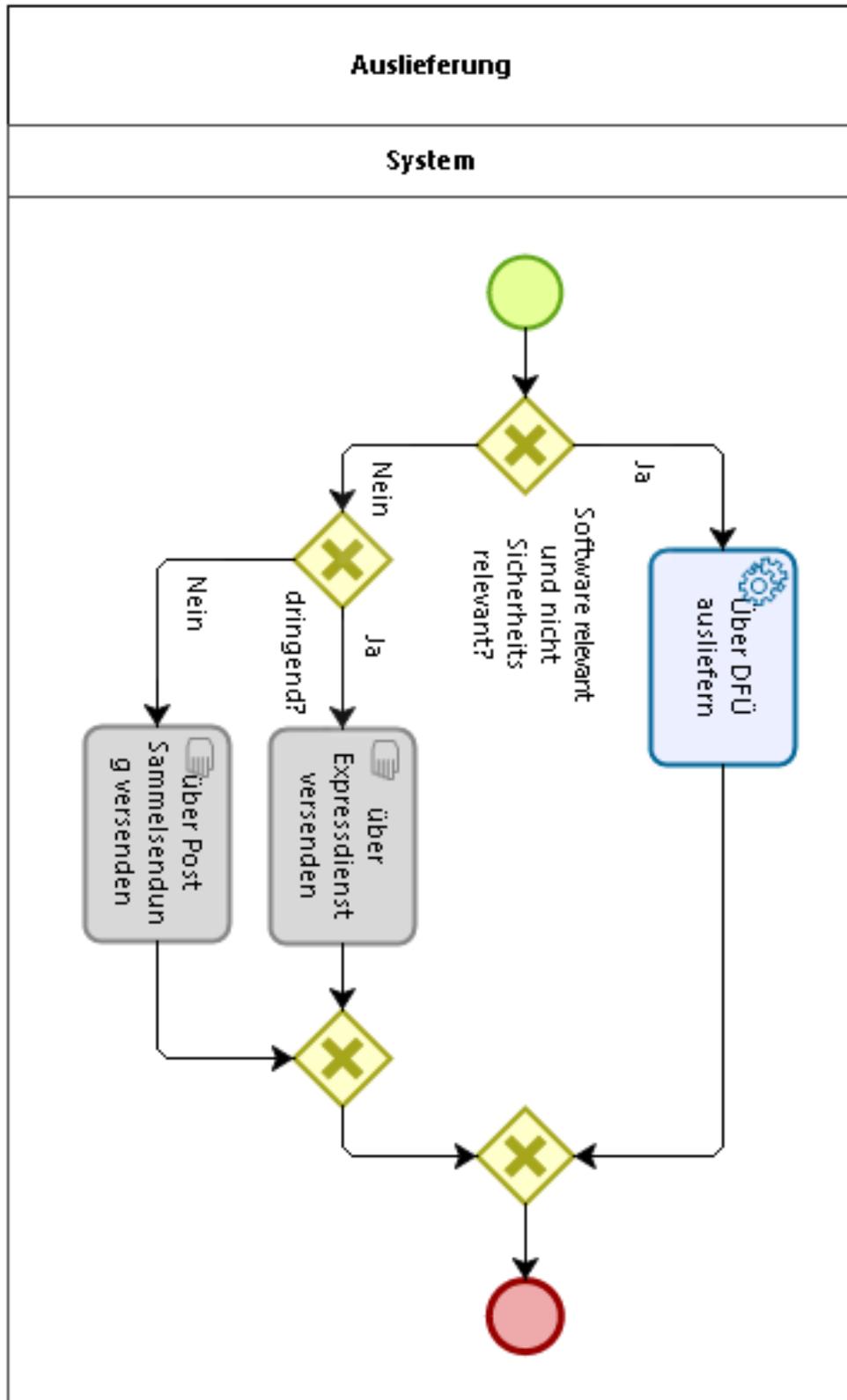


Abbildung 25: BPMN Workflow Auslieferung nach [40]

4.1.2 Modellierung in WWF

Da in diesem Anwendungsfall, dem Automatisieren des Bestell- und Konfigurations-Prozesses von Software für Formgebungsmaschinen, die Modellierungssprache Windows Workflow Foundation besser geeignet ist als BPMN, wird der eben dargestellte Workflow von Karl-Heinz Stöckler [40] in WWF neu modelliert, welcher in den Abbildungen 27-37 gezeigt wird.

Da sich durch die Neumodellierung Änderungen ergeben haben, werden diese nachfolgend beschrieben und die jeweiligen Designentscheidungen dazu erklärt.

Aufgrund der Änderung der Modellierungssprache von BPMN zu WWF gibt es keine Pools und Lanes mehr.

Weiters gibt es in WWF abstrakte Task-Typen, deren Ausführung nicht exakt definieren, wie der User-Task in BPMN, nicht. Diese werden in diesem Anwendungsfall durch das Senden einer E-Mail an den User, wie in Abbildung 27 zu sehen, oder durch das Anzeigen einer Webseite, welche mittels eines Webservice-Aufrufs geöffnet wird, wie in Abbildung 29, ersetzt. Es ist zu beachten, dass nach dem Anzeigen einer Webseite, wenn eine Benutzereingabe erfolgt, immer mittels "Auf Ereignis in Listenelement warten"- oder "Auf Feldänderung warten"-Aktion gewartet wird, da sonst der Workflow automatisch weiterläuft. Die "Auf Ereignis in Listenelement warten"-Aktion wird dabei verwendet, wenn nicht bekannt ist, welchen Wert ein gewisses Feld annimmt, beziehungsweise wenn sich unterschiedliche Felder ändern können. Die "Auf Feldänderung warten"-Aktion wird wiederum verwendet, wenn der Workflow angehalten werden muss, bis ein Feld einen gewissen Wert annimmt. Um den Workflow vom Webservice aus fortzusetzen, kann die REST-API SharePoints verwendet werden, indem das gewünschte Feld des aktuellen Items angepasst wird.

Auch Manual-Tasks und Service-Tasks werden durch Webservice-Calls ersetzt. Bei Service-Tasks ist dies sehr naheliegend, da diese in BPMN dem Aufrufen von Webservices dienen. Manual-Tasks hingegen könnten auch als "Aufgabe zuweisen"-Aktion oder "E-Mail senden"-Aktion umgesetzt werden, da mit allen Dreien das Benachrichtigen einer Person möglich ist. Die Webservice-Call-Aktion wurde in diesem Anwendungsfall gewählt, da nicht alle menschlichen Akteure Zugriff auf das SharePoint-System besitzen und E-Mails einen umständlichen Kommunikationskanal darstellen. Mittels dem aufgerufenen Webservice wird direkt über ein Interface eine Nachricht in das verwendete Anwendungsprogramm der Person, wie beispielsweise SAP, gesendet.

Eine weitere Änderung ist, dass in WWF ein Sequenzfluss nicht einfach beendet werden kann, wie dies in Abbildung 20 zu sehen ist. Um dies zu lösen, wird eine If-Anweisung verwendet, welche nach einer Stufe

prüft, ob ein Fehler aufgetreten ist. Wie in Abbildung 28 zu sehen ist, wird im Fehlerfall der Workflow beendet, indem zur Stufe "End of Workflow" weitergeleitet wird. Ist hingegen kein Fehler aufgetreten, so wird der Workflow normal fortgesetzt.

Weitere Änderungen entstehen, da es in WWF keine Gateways, sondern nur Bedingungen gibt, von welchen in diesem Anwendungsfall nur die If-Anweisung verwendet wird. Die If-Anweisung entspricht den aus Programmiersprachen bekannten If-Anweisungen, mit der Einschränkung, dass nur eine Variable in einer If-Anweisung verwendet werden kann. Weiters ist es nicht möglich, auf Größer oder Kleiner zu prüfen. Deshalb wird in Abbildung 29 die Prüfung nach externer Benutzer und kostenpflichtige Option aufgesplittet. Diese Aufspaltung ist jedoch nicht immer möglich, da es Konstrukte in Graphen gibt, die nicht mittels If-Anweisungen ohne Code-Verdoppelung oder GoTo-Statement (GoTo ermöglicht einen Sprung ausgehend von der GoTo-Anweisung zu einer gekennzeichneten Anweisung [41]) umgesetzt werden können. WWF unterstützt keinen dieser beiden Ansätze, weshalb Konstrukte, welche ohne einen der beiden Ansätze nicht dargestellt werden können, verboten sind und zu Fehlern führen. Beispiele hierfür sind das "Preis oder Lieferzeit nicht vorhanden"-Gateway in Abbildung 22 und das "Software-relevant und nicht Sicherheits-relevant?"-Gateway in Abbildung 25. Wie in Abbildung 26 zu sehen ist, würde die Aufspaltung des "Software-relevant und nicht Sicherheits-relevant?"-Gateways in mehrere If-Anweisungen einen Code erzeugen, welcher Code-Verdoppelung (in diesem Beispiel) oder ein GoTo-Statement enthält.

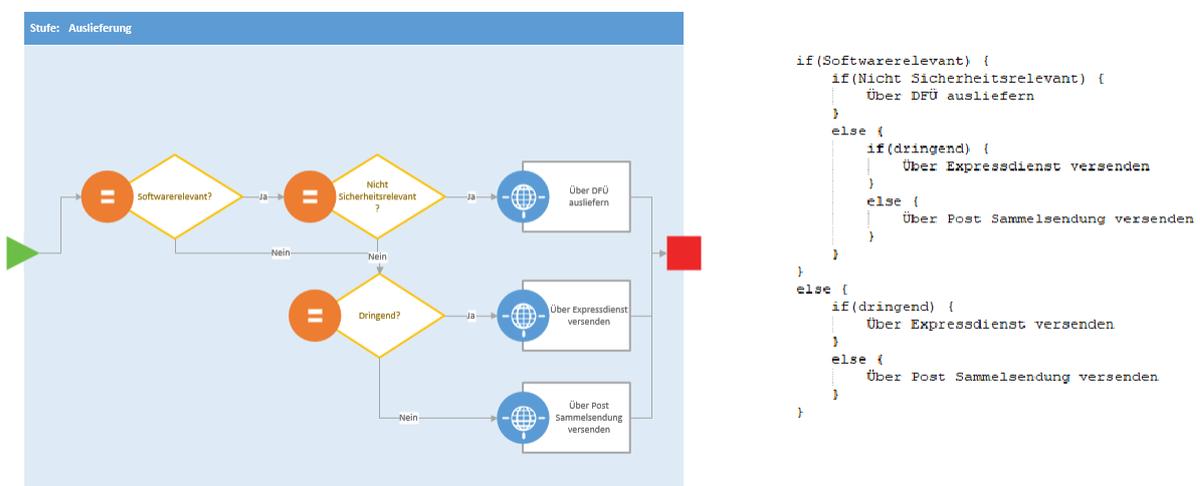


Abbildung 26: WWF Problem mit der If-Anweisung

Eine weitere Einschränkung in WWF ist, dass If-Anweisungen nicht mehr als zwei ausgehende Pfade besitzen können. Da in einer If-Anweisung nur auf Gleichheit geprüft wird, können nur die beiden Pfade Ja, wenn die Bedingung in der Anweisung erfüllt wurde, und Nein, wenn die Bedingung nicht erfüllt

wurde, von der Anweisung ausgehen. Aufgrund dessen wird das Gateway beim Generieren des Angebotes auf zwei If-Anweisungen aufgesplittet, wie in Abbildung 30 zu sehen ist.

Die letzte Änderung, die durch den Wechsel zu WWF entstanden ist, ist das Ersetzen des Inklusiven-Gateways, da es kein Äquivalent in WWF gibt. Um dieses zu ersetzen, wird ein paralleler Sequenzfluss gestartet, in welchem am Anfang die jeweilige Abfrage mittels If-Anweisung durchgeführt wird. Trifft diese zu, wird der Sequenzfluss durchlaufen. Trifft sie nicht zu, so wird der Sequenzfluss ausgelassen, wie in Abbildung 33 zu sehen ist.

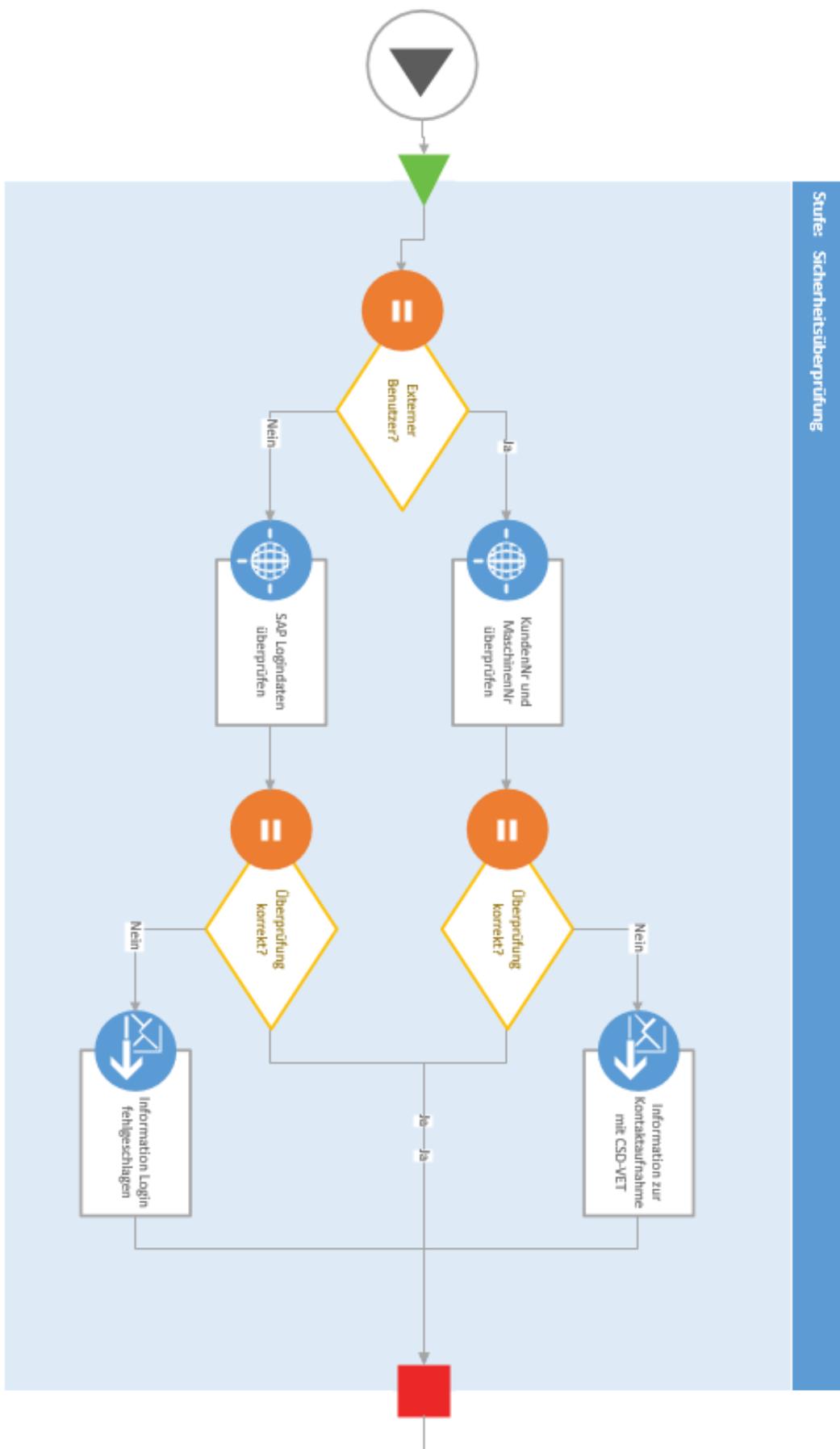


Abbildung 27: WWF Workflow Stand 2015 1

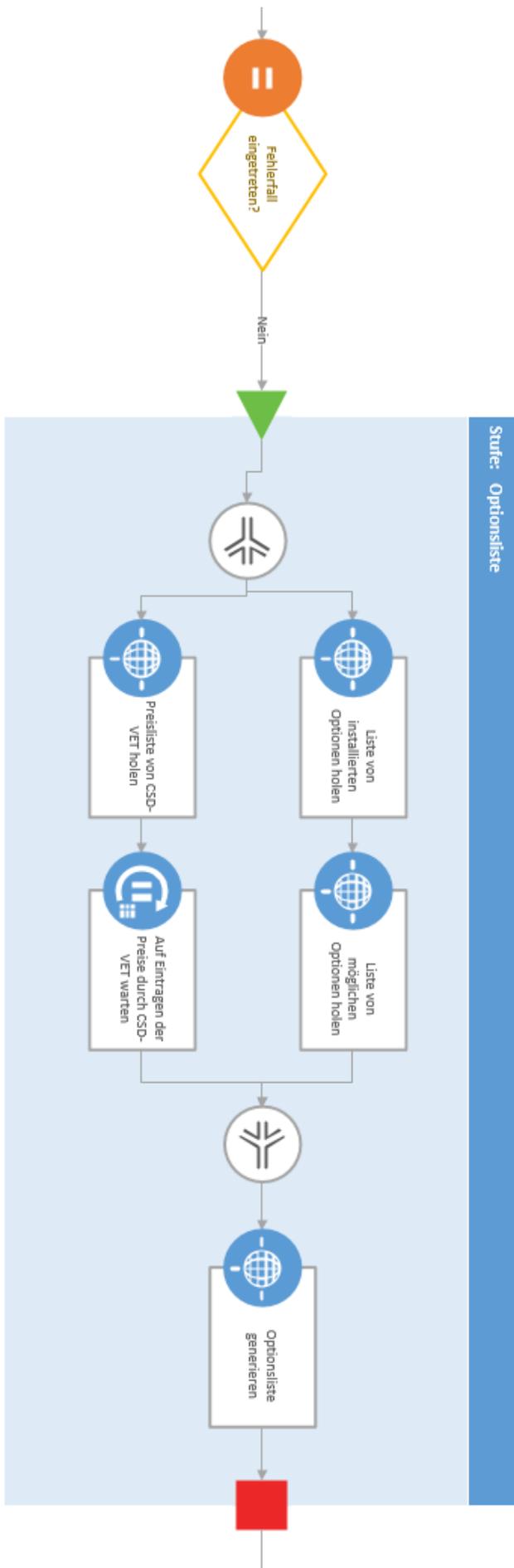


Abbildung 28: WWF Workflow Stand 2015 2

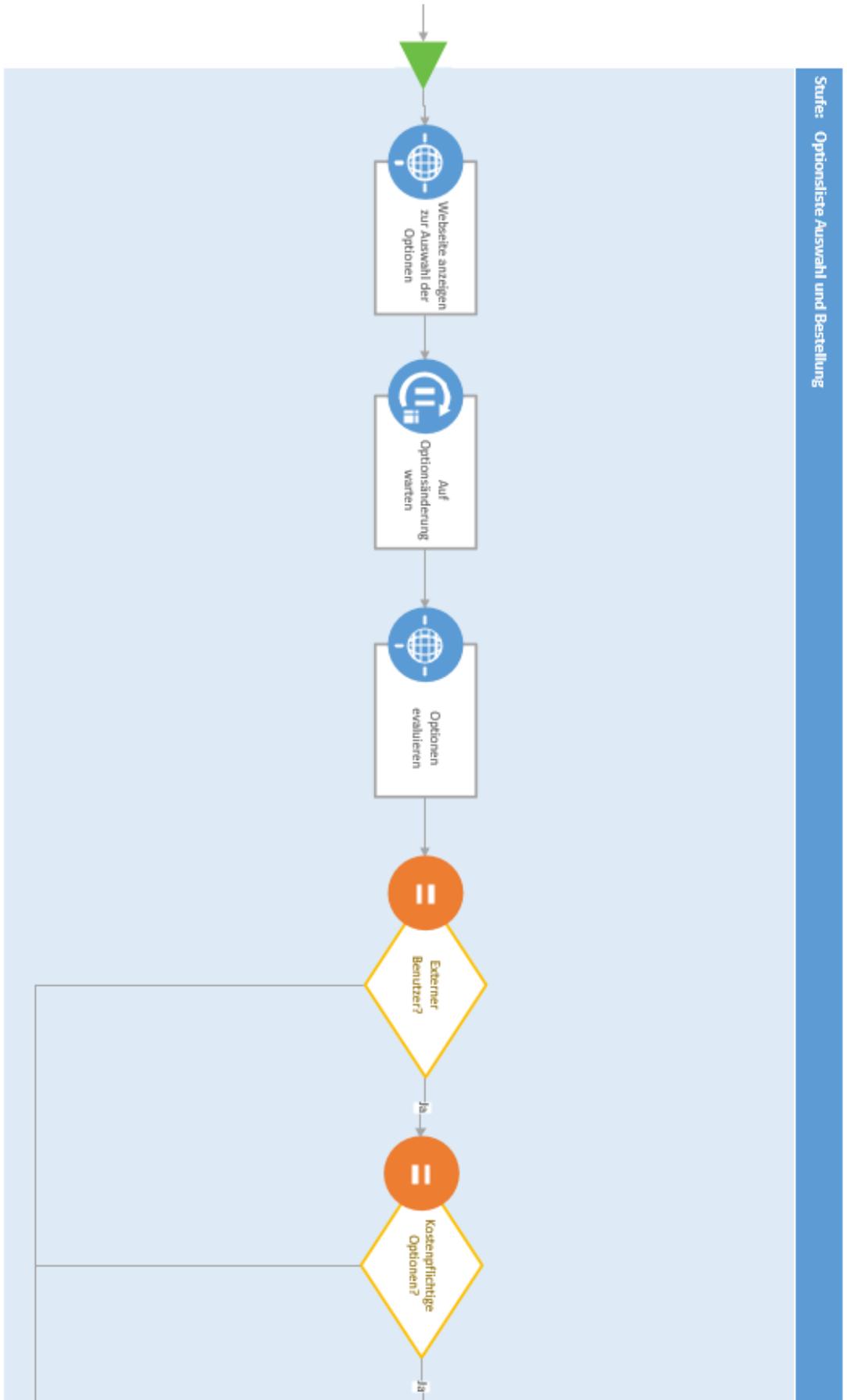


Abbildung 29: WWF Workflow Stand 2015 3

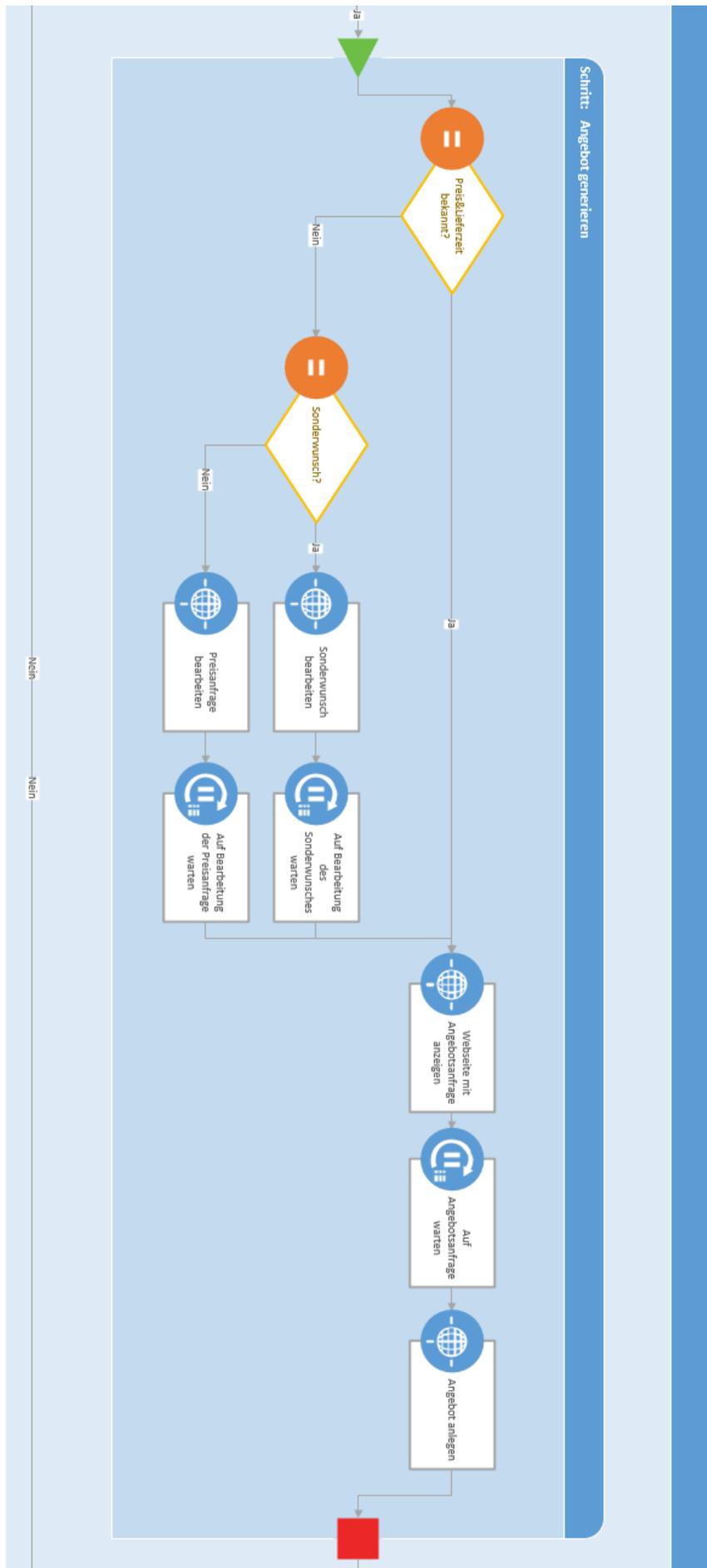


Abbildung 30: WWF Workflow Stand 2015 4

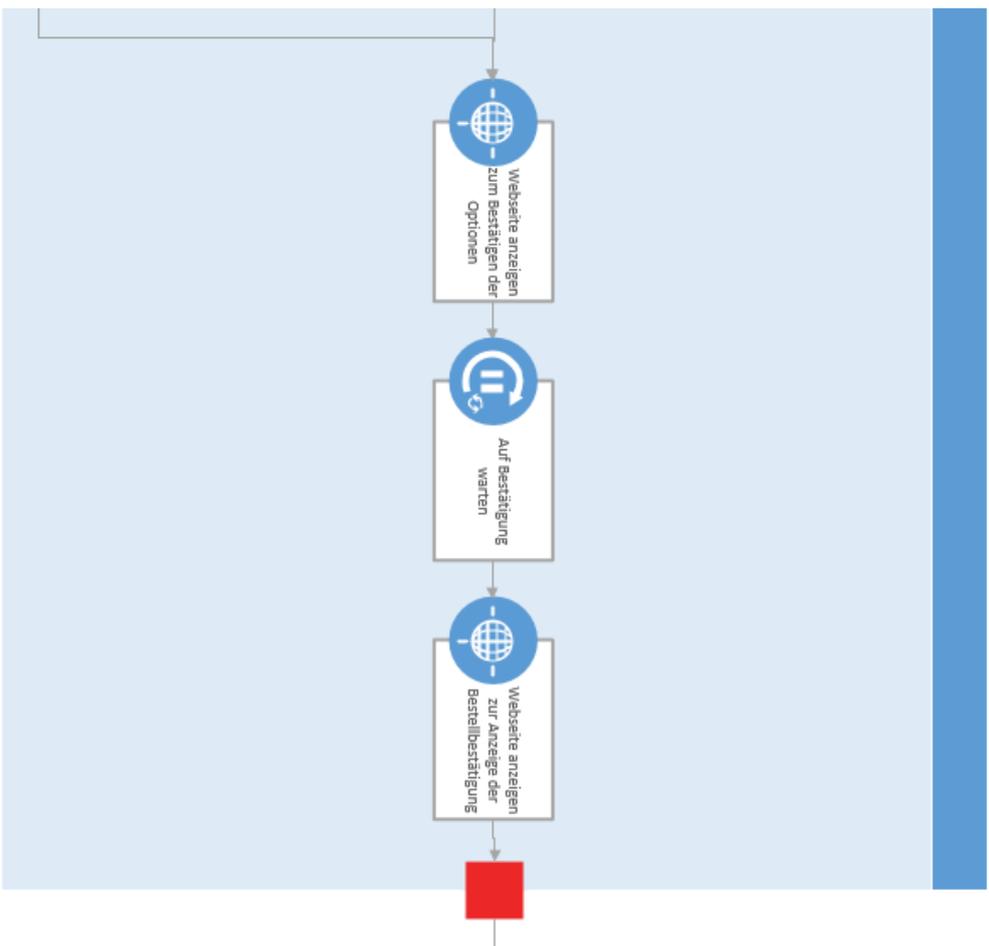


Abbildung 31: WWF Workflow Stand 2015 5

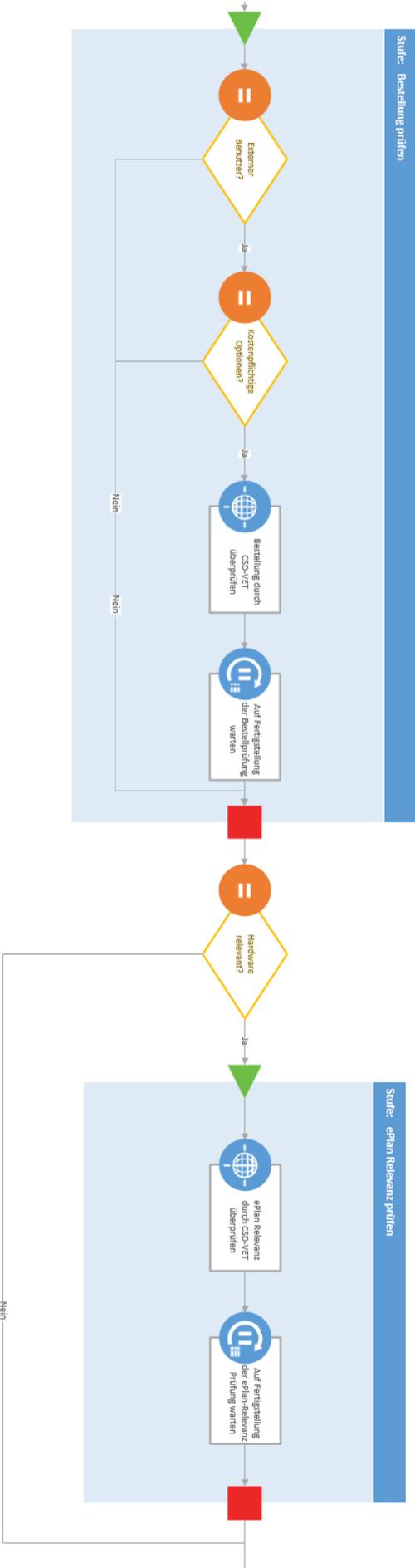


Abbildung 32: WWF Workflow Stand 2015 6

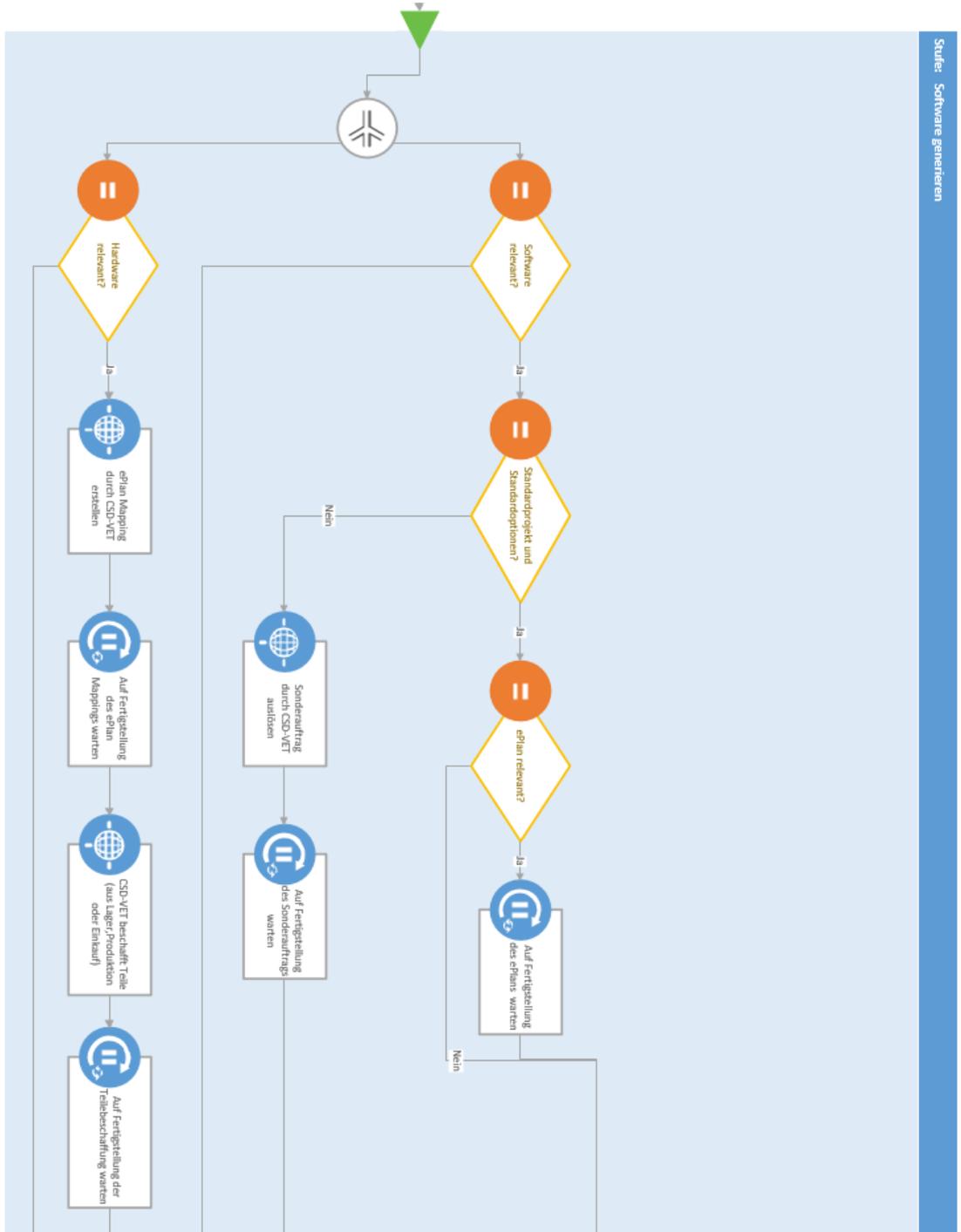


Abbildung 33: WWF Workflow Stand 2015 7

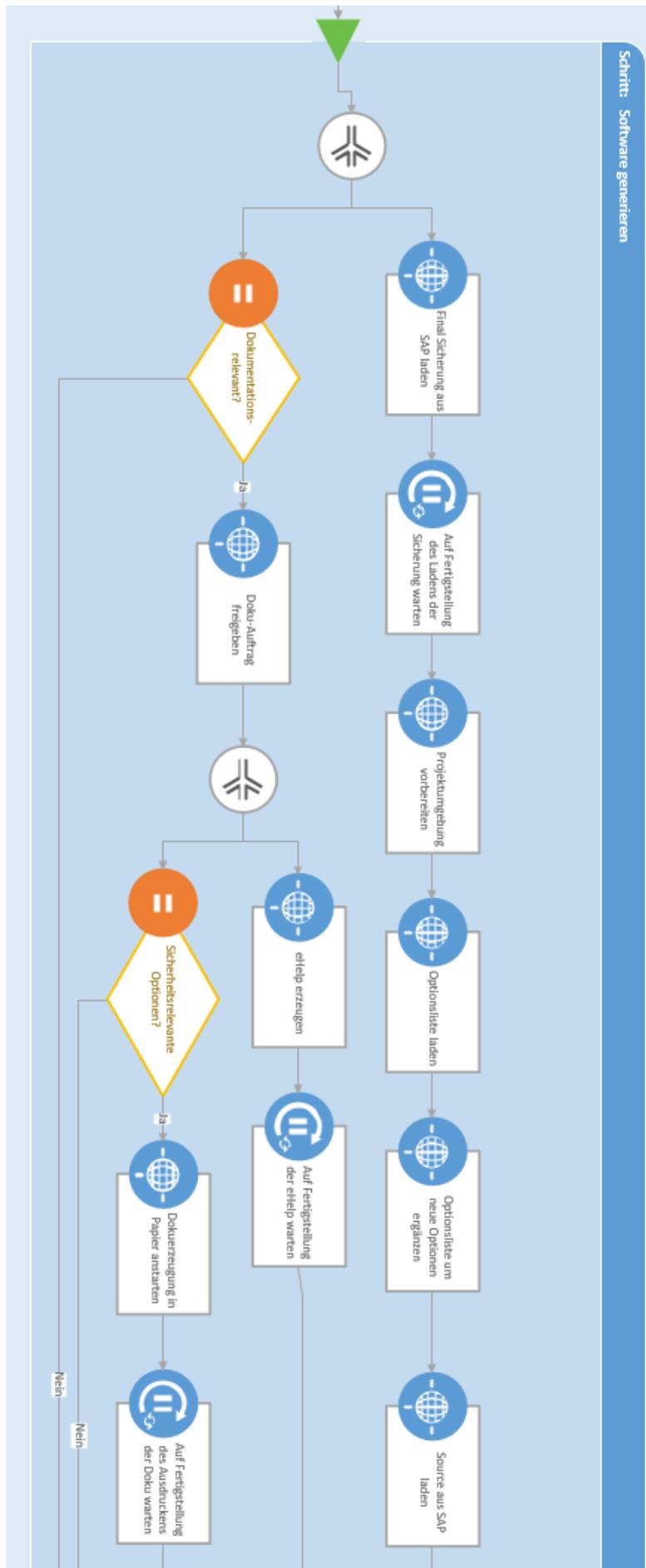


Abbildung 34: WWF Workflow Stand 2015 8

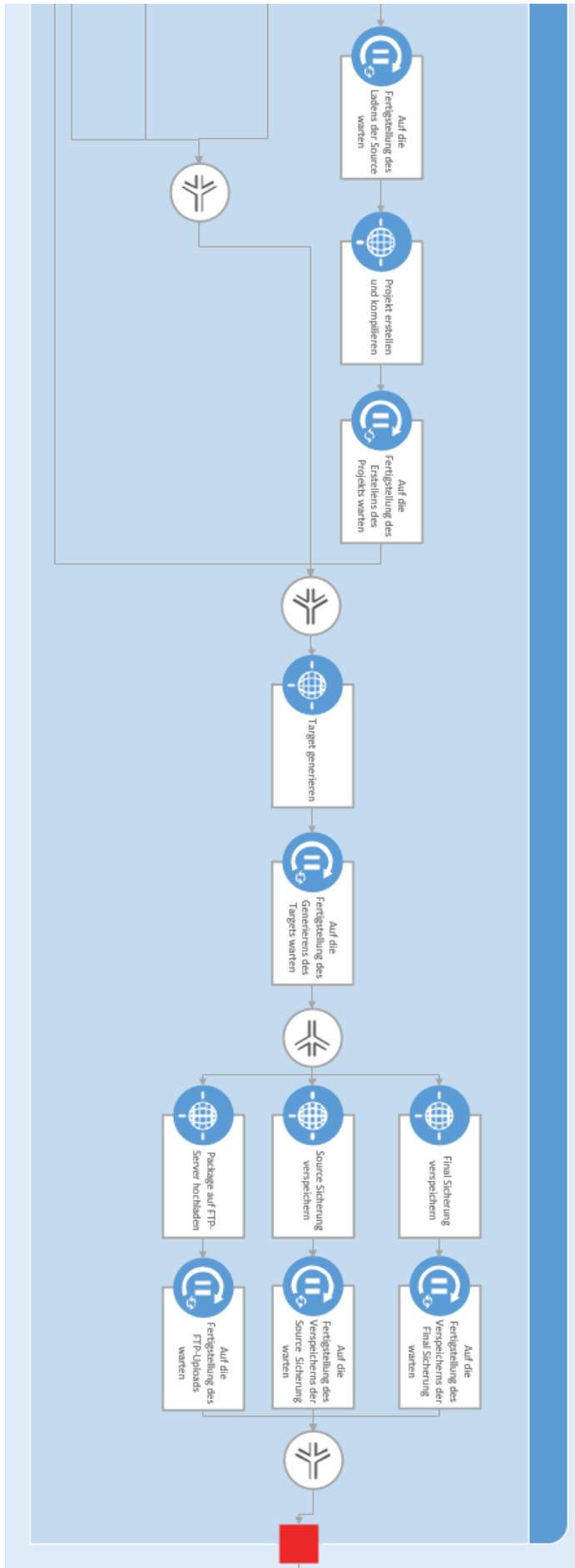


Abbildung 35: WWF Workflow Stand 2015 9

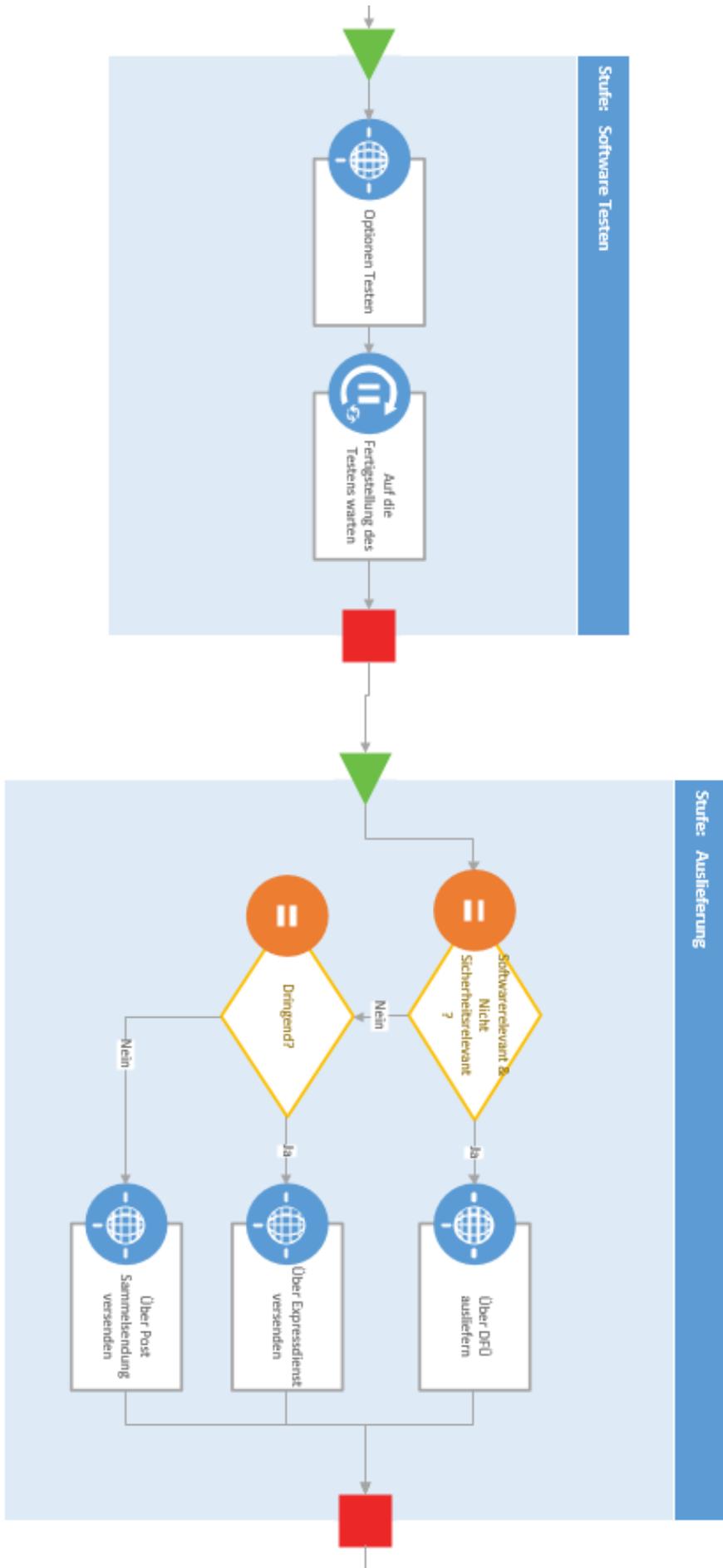


Abbildung 36: WWF Workflow Stand 2015 10

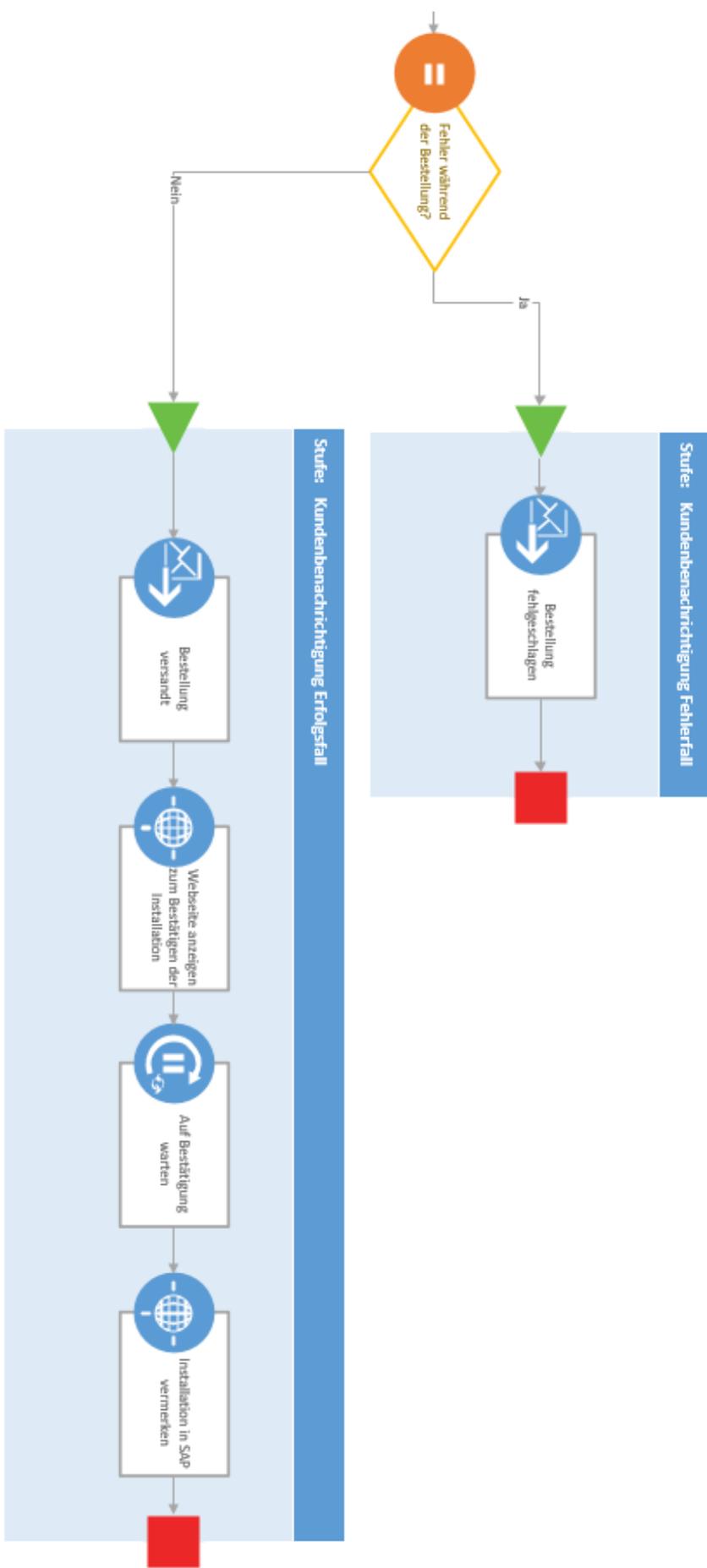


Abbildung 37: WWF Workflow Stand 2015 11

4.2 Anforderungsänderungen und der neue Workflow

Da der Workflow von Karl-Heinz Stöckler 2015 modelliert wurde, musste dieser an die heutigen Anforderungen angepasst werden. Im Zuge dieses Kapitels werden jene Anforderungen dargelegt, die sich im Laufe der Zeit geändert haben und wie sich diese auf den Workflow ausgewirkt haben.

Eine der größten Änderungen ist der Verzicht auf Hardware-Optionen, da die Bestellung dieser über einen Workflow, aufgrund von logistischen und qualitativen Problemen, weder umsetzbar noch gewünscht ist. Aufgrund dessen wurden alle Elemente des vorherigen Workflows entfernt, die einen Bezug zu Hardware-Optionen hatten. Darunter fallen auch jene Elemente, die sich mit dem ePlan beschäftigen. Der Vorteil, der durch diese Änderungen entsteht, ist, dass die Schnittstelle zur CSD-VET-Abteilung entfernt wird, wodurch ein Teil des Human-Machine-Interfaces (HMI) wegfällt, welches eine Fehlerquelle darstellt und die Durchlaufzeit des Workflows erhöht.

Eine weitere Anforderungsänderung ist der Wegfall der Sonderoptionen. Bei der Auswahl von Optionen sind nur mehr Software-relevante Standardoptionen verfügbar, da Sonderoptionen wiederum von der CSD-VET-Abteilung abgearbeitet werden. Die kundenorientierte Betreuung bei Sonderoptionen/ Sondermaschinen ist ein Qualitätsmerkmal der Firma Engel, weshalb die automatische Abarbeitung dieser Softwaregenerierungs-Aufträge nicht zielführend und gewünscht ist. Weiters wird dadurch die komplette Schnittstelle zur CSD-VET-Abteilung entfernt, was, wie oben bereits erwähnt, das Fehlerrisiko reduziert und die Durchlaufzeit optimiert.

Aufgrund des Wegfalls der Hardwareoptionen und der kompletten Kommunikation mit der CSD-VET-Abteilung, wird eine Unterscheidung zwischen externen und internen Benutzern nicht mehr benötigt, da alle Softwareoptionen kostenfrei sind und keine Sonderoptionen mehr möglich sind. Deshalb entfällt der gesamte Login-Prozess, wodurch sowohl der Workflow als auch die Benutzerinteraktion vereinfacht werden.

Eine Anforderungsänderung, die zu einem Zuwachs an Workflow-Elementen geführt hat, ist, dass Formgebungsmaschinen-spezifische Daten aus SAP vor dem Generieren der neuen Softwareversion geladen werden müssen. Dieses Laden von Formgebungsmaschinen-spezifischen Daten aus SAP wird zu Beginn des Workflows durchgeführt und ist in Abbildung 38 ersichtlich.

Eine zusätzliche Anforderung an den Workflow ist, dass dieser eine Möglichkeit bietet, einen menschlichen Operator zu kontaktieren, welcher eine der Operationen "Software generieren", "Software testen " oder "Software dokumentieren" manuell durchführt. Weiters ist im Fehlerfall einer automatischen Ausführung einer der drei Operationen, gewünscht, dass ein menschlicher Operator diese Operation manuell durchführt

und danach der Workflow normal fortgeführt wird. Durch diese Maßnahme soll die Anzahl der Durchläufe des Workflows mit einem fehlerhaften Endergebnis gegen Null gehen. In Abbildungen 40 und 41 ist dies dargestellt. Im manuellen Fall wird der Operator mittels einer E-Mail benachrichtigt (es wird ein Webservice zum Senden einer E-Mail verwendet, da der Mailserver SharePoints (Microsoft Exchange Server[23]) nicht alle benötigten Einstellungen zulässt, die der firmeneigene Mailserver ermöglicht), in der ein Link enthalten ist, auf welchen er bei erfolgreichem Durchführen der Operation klickt, um den Workflow fortzusetzen. Im automatischen Fall wird die Operation durchgeführt und im Fehlerfall wird entweder der Operator wie im manuellen Fall kontaktiert, oder der Workflow wird abgebrochen. Dies hängt von den Formgebungsmaschinen-spezifischen Daten aus SAP ab, welche festlegen, ob der Workflow abgebrochen (nicht mehr Versuchen) oder manuell fortgeführt (erneut versuchen) wird.

Aufgrund von Anforderungsänderungen wurde auch der Auslieferungsprozess geändert. Da Hardwareoptionen komplett entfernt wurden, wurde auch die Möglichkeit der Auslieferung der neuen Software inklusive Dokumentation via Post (Express oder Sammelzustellung) entfernt, was in Abbildung 47 zu sehen ist. Dadurch verringert sich die Zeitspanne zwischen dem Bestellen der neuen Softwareversion und dem Erhalten ebenjener deutlich.

Eine weitere Änderung ist, dass die Erstellung einer Dokumentation immer durchgeführt werden muss, da jede Änderung Dokumentations-relevant ist. Weiters ist die Erstellung einer Dokumentation in Papierform nicht mehr möglich, da keine sicherheitsrelevanten Optionen über diesen Workflow hinzugefügt werden können und die Auslieferung per Post nicht mehr möglich ist.

Eine Änderung des Workflows, die nicht aufgrund von geänderten Anforderungen, sondern aufgrund einer logischen Anpassung durchgeführt wurde, ist, dass die Software vor dem Sichern getestet wird. Durch diese Änderung wird die Anzahl der fehlerhaften Sicherungen deutlich verringert. Eine weitere Anpassung ist, dass die einzelnen Softwaregenerierungs-Schritte nicht mehr aus dem Workflow heraus aufgerufen werden. Stattdessen wird ein Webservice aufgerufen, welcher die einzelnen Schritte durchführt. Dadurch wird die Flexibilität des Workflows erhöht, die Komplexität verringert und die Anzahl der Änderungen im Workflow gesenkt.

Die letzte Änderung des Workflows ist die komplette Entfernung von Interaktionen mit dem Benutzer außerhalb der Abschluss-E-Mails. Da durch die neuen Interfaces bereits alle benötigten Informationen beim Start des Workflows bekannt sind, werden keine zusätzlichen Interaktionen mit dem Benutzer benötigt, wodurch die Anzahl der HMIs sinkt. Wie bereits oberhalb erwähnt, werden dadurch die Durchlaufzeit und die Fehlerrate verringert.

Der nachfolgende Workflow, welcher in den Abbildungen 38-47 abgebildet wird, ist durch die Kombination

des Workflows der vorherigen Masterarbeit, den Änderungen bedingt durch den Wechsel von BPMN zu WWF und den geänderten Anforderungen entstanden. Dabei ist zu beachten, dass Abbildung 39 einen Überblick über die drei Hauptoperationen gibt und Abbildungen 40-45 diese genauer abbilden.

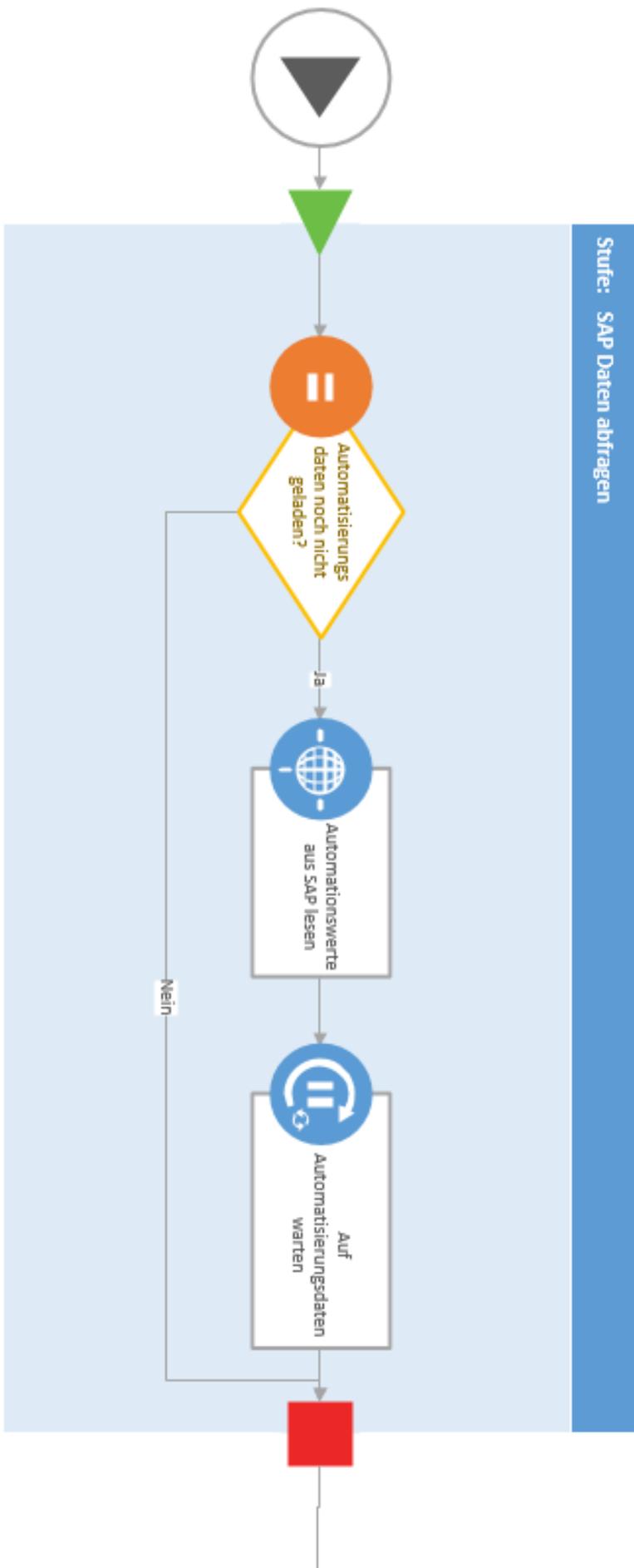


Abbildung 38: WWF Workflow aktueller Stand 1

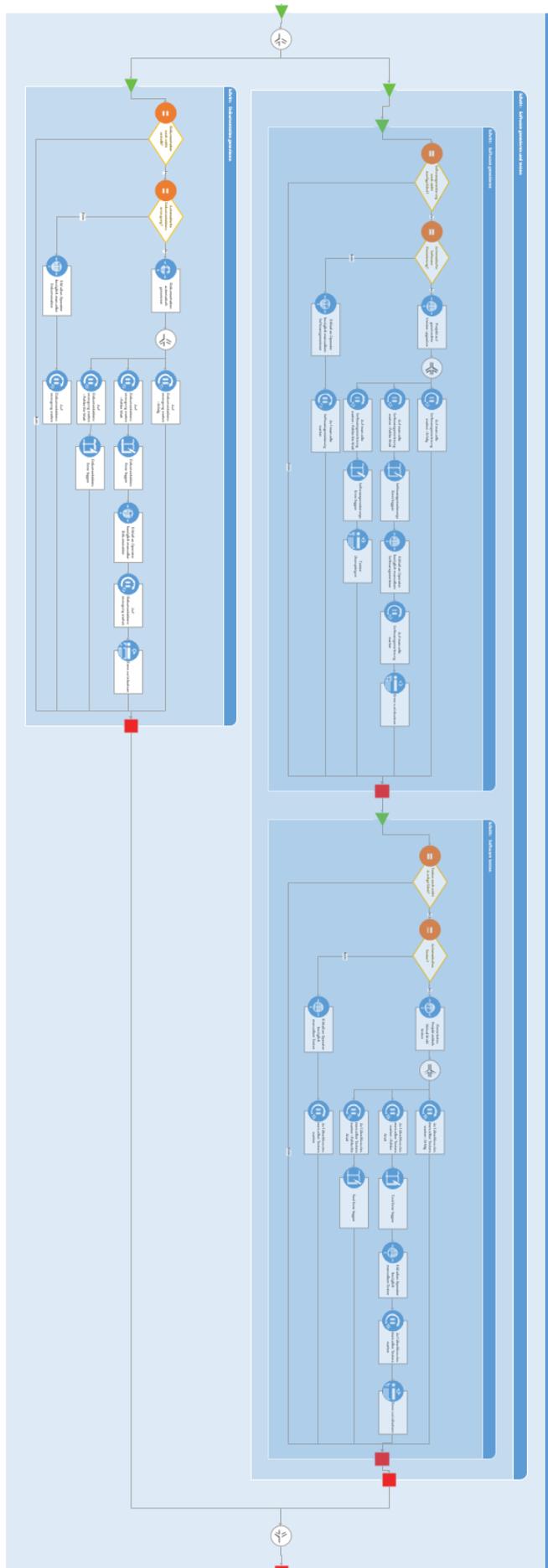


Abbildung 39: WWF Workflow aktueller Stand 2

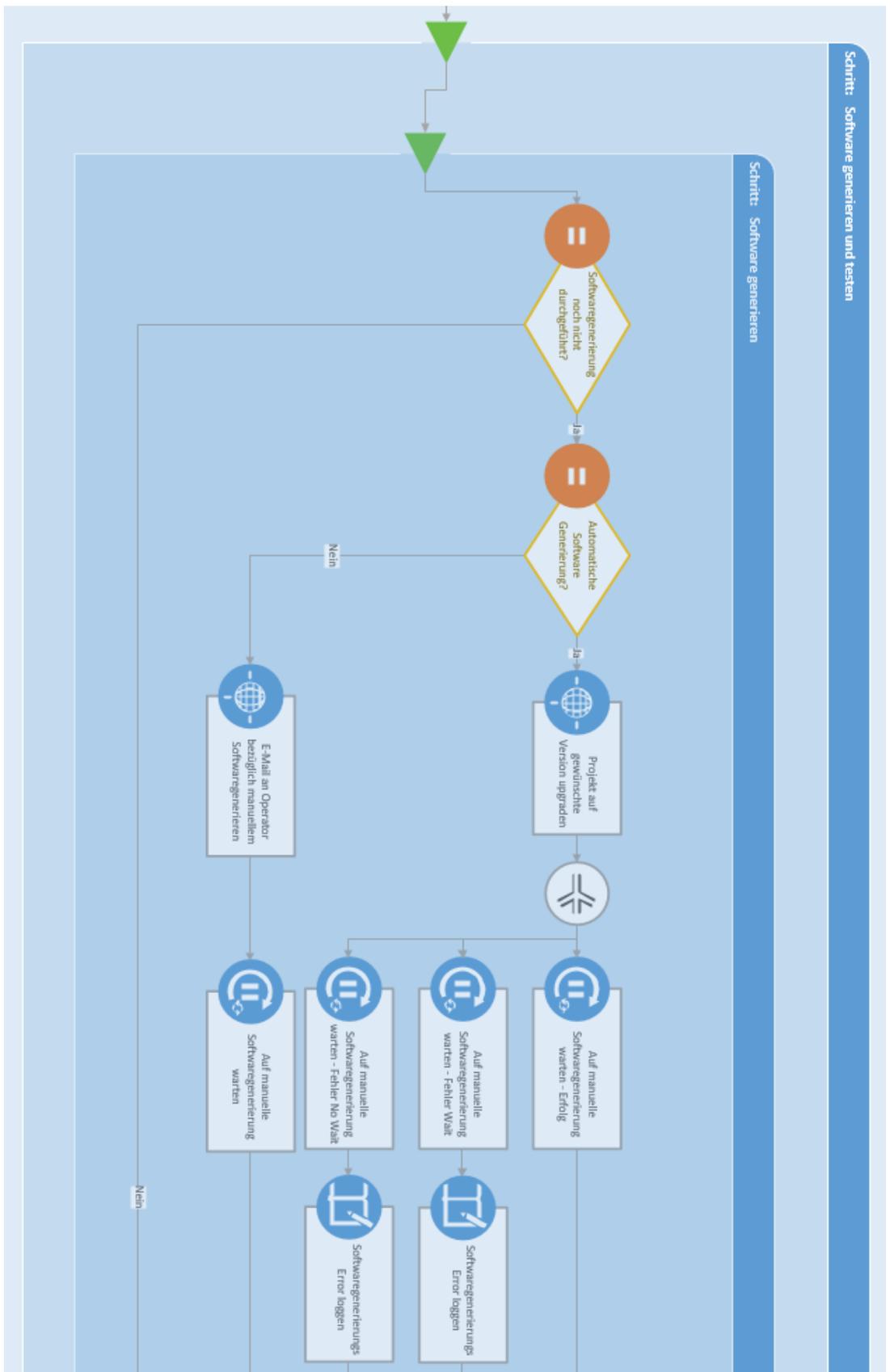


Abbildung 40: WWF Workflow aktueller Stand 3

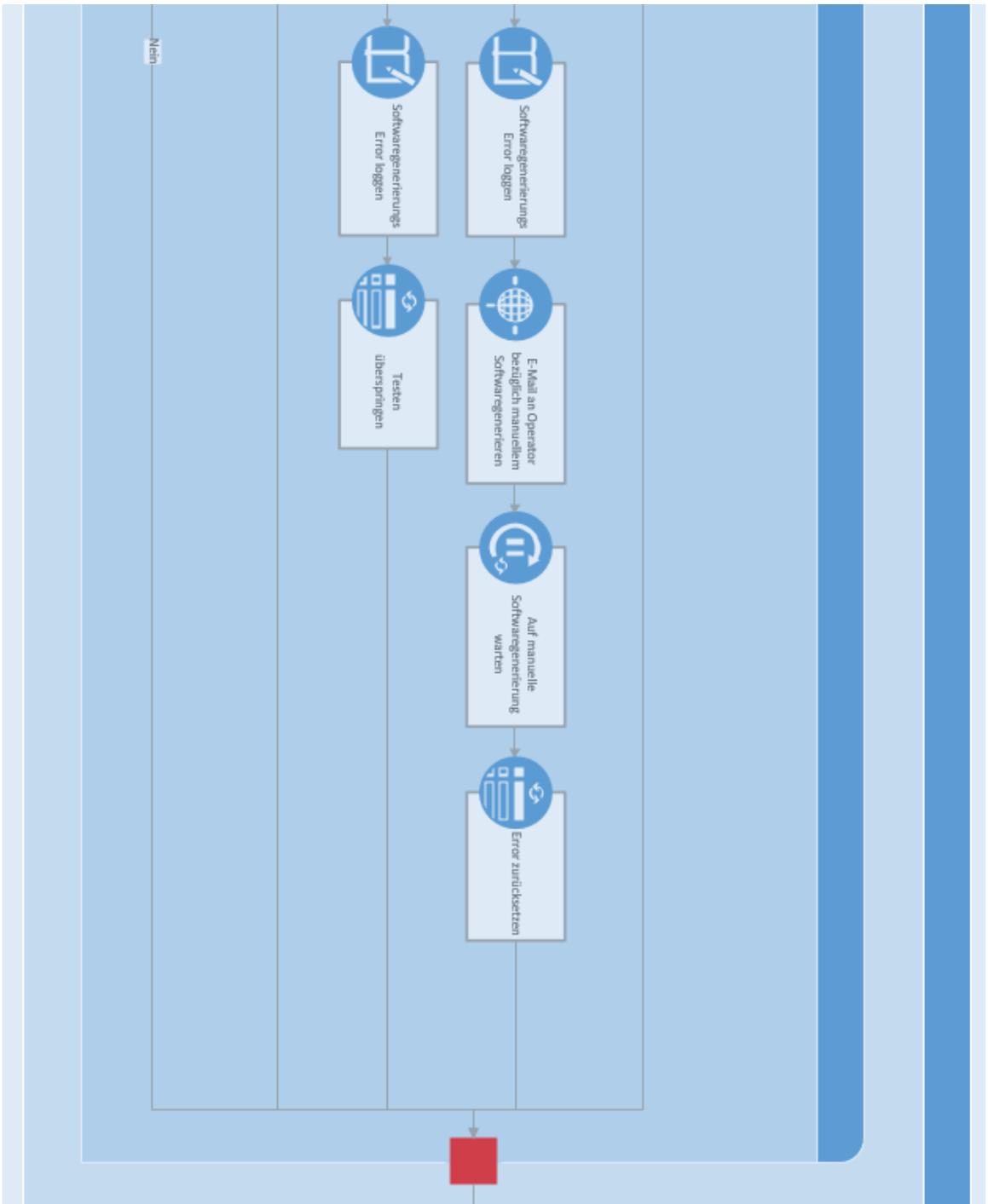


Abbildung 41: WWF Workflow aktueller Stand 4

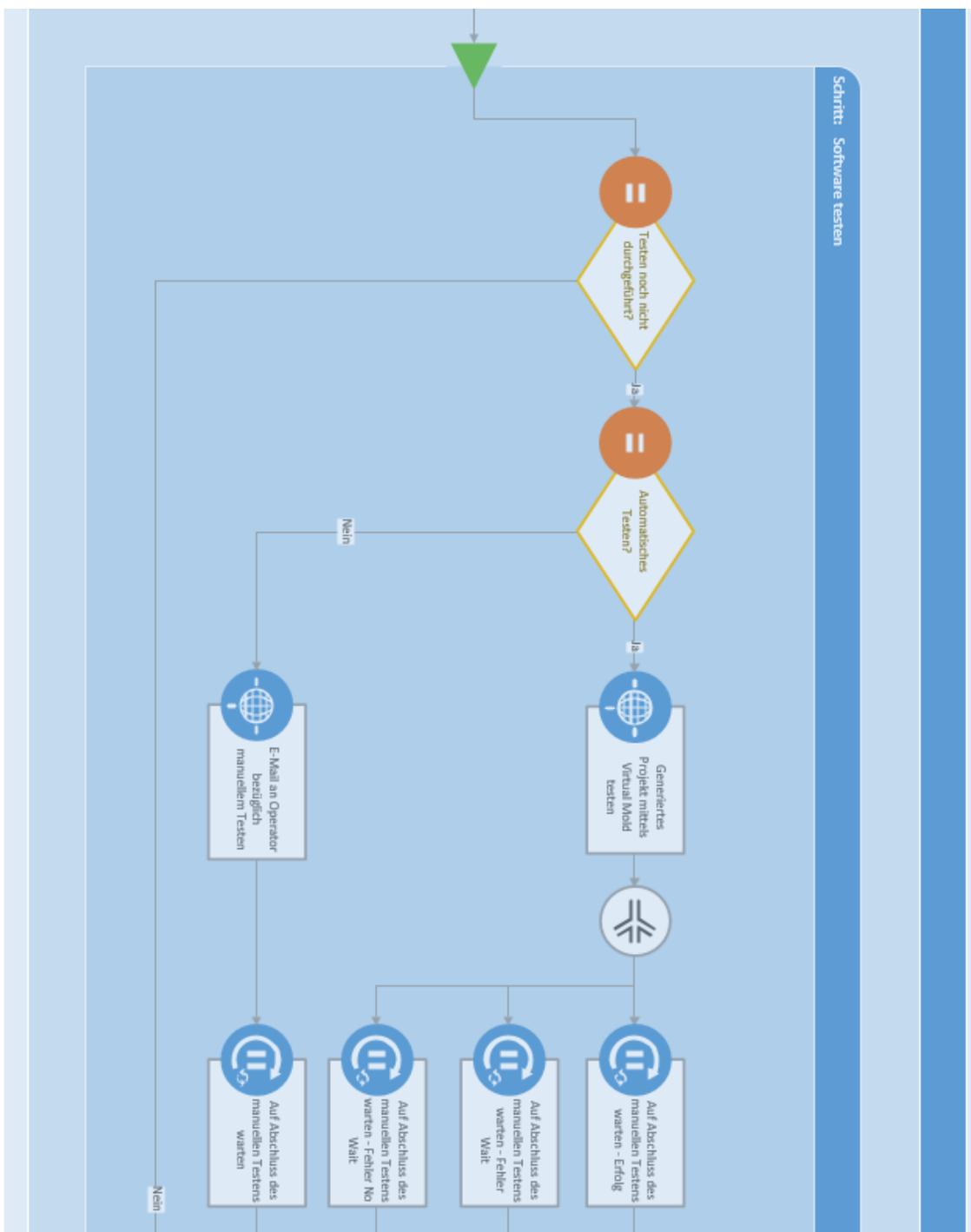


Abbildung 42: WWF Workflow aktueller Stand 5

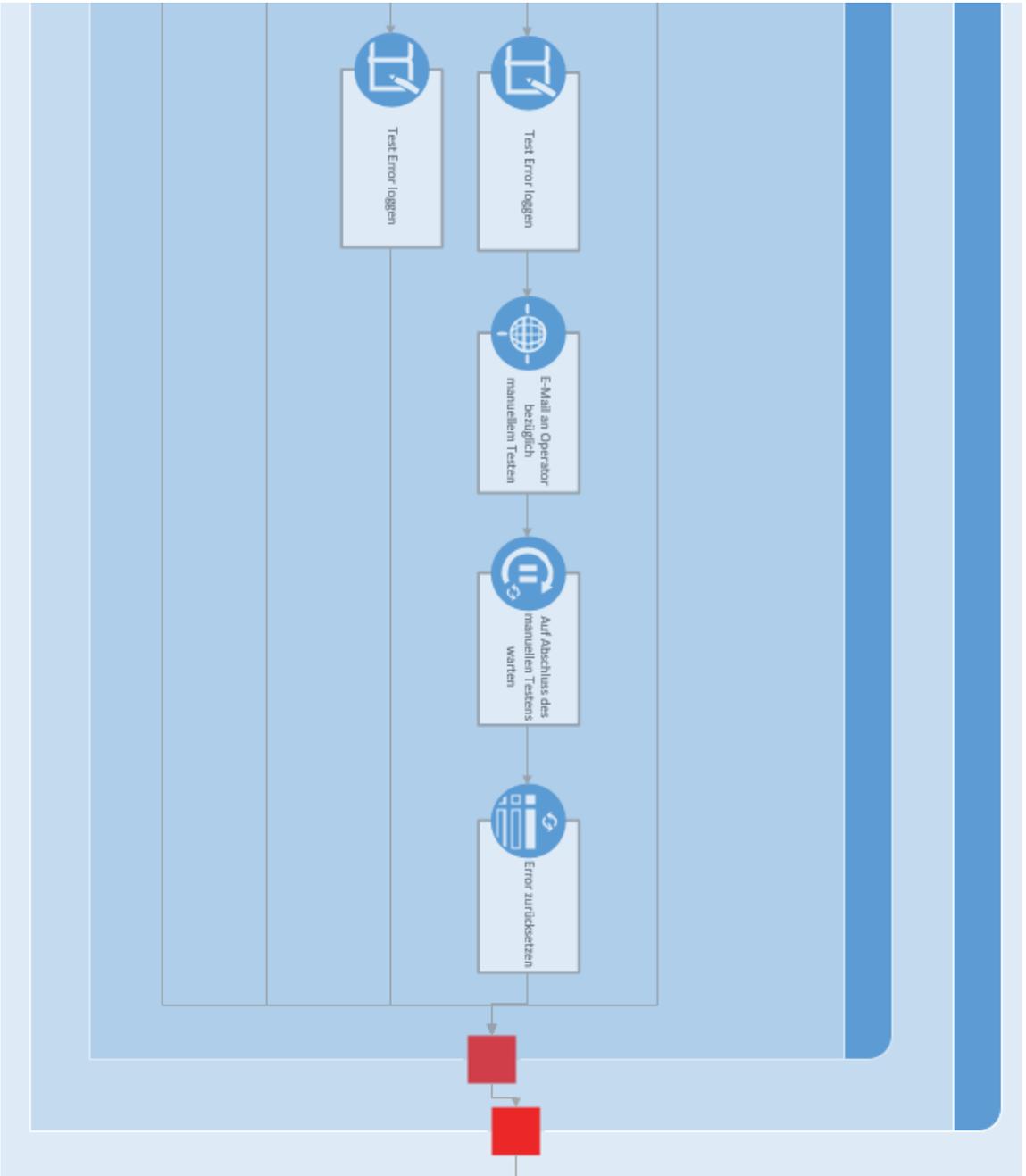


Abbildung 43: WWF Workflow aktueller Stand 6

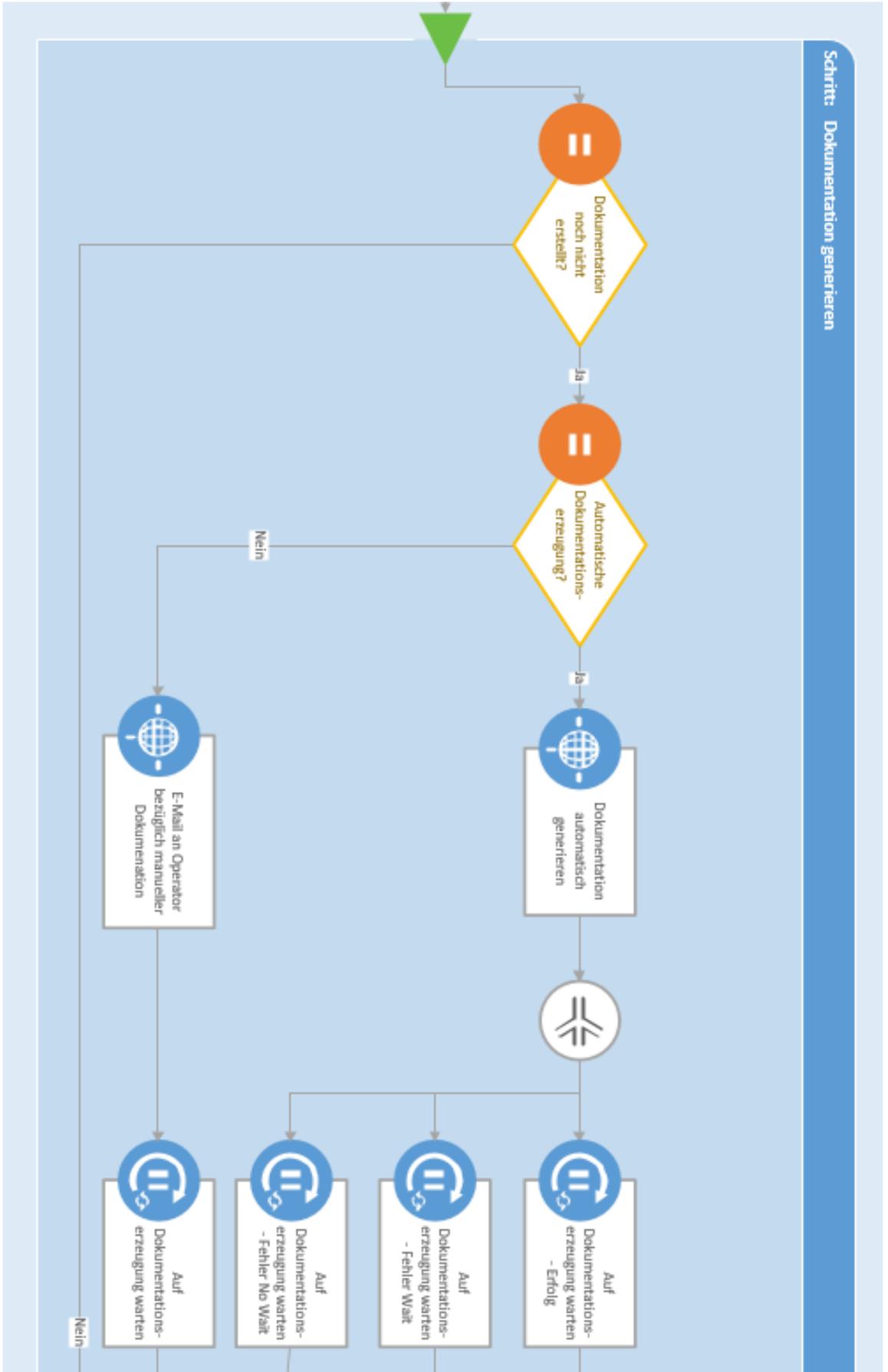


Abbildung 44: WWF Workflow aktueller Stand 7

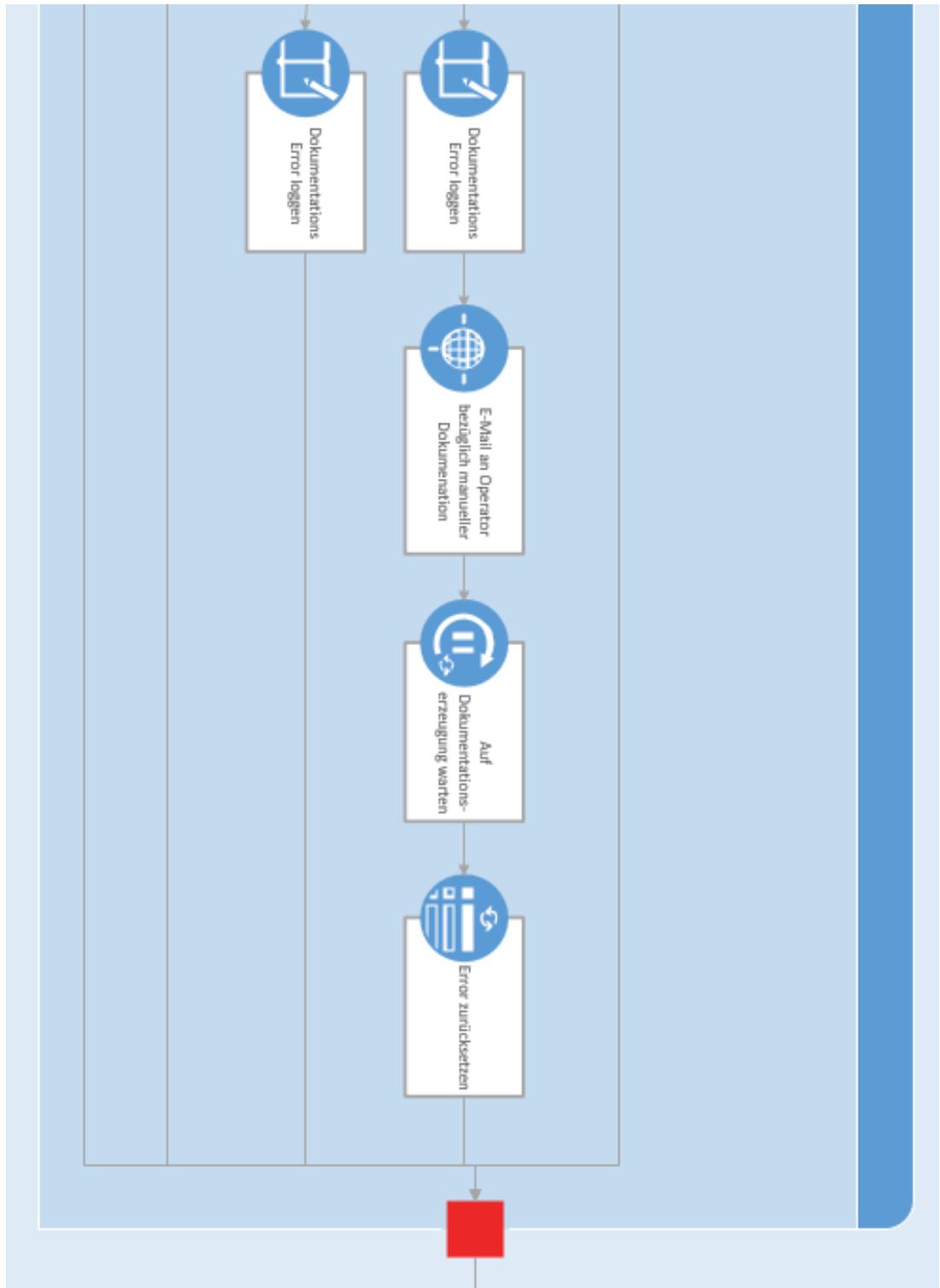


Abbildung 45: WWF Workflow aktueller Stand 8

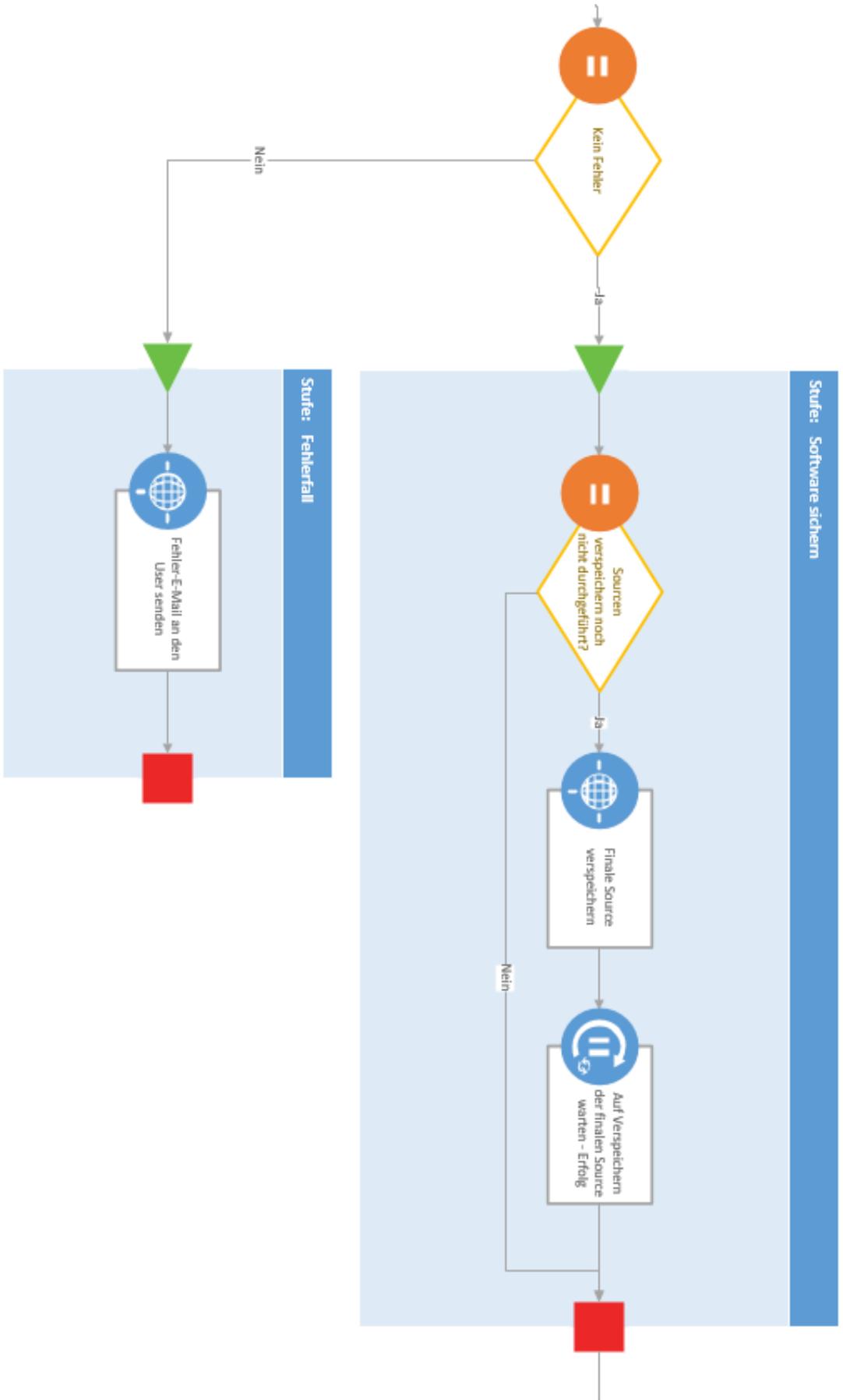


Abbildung 46: WWF Workflow aktueller Stand 9

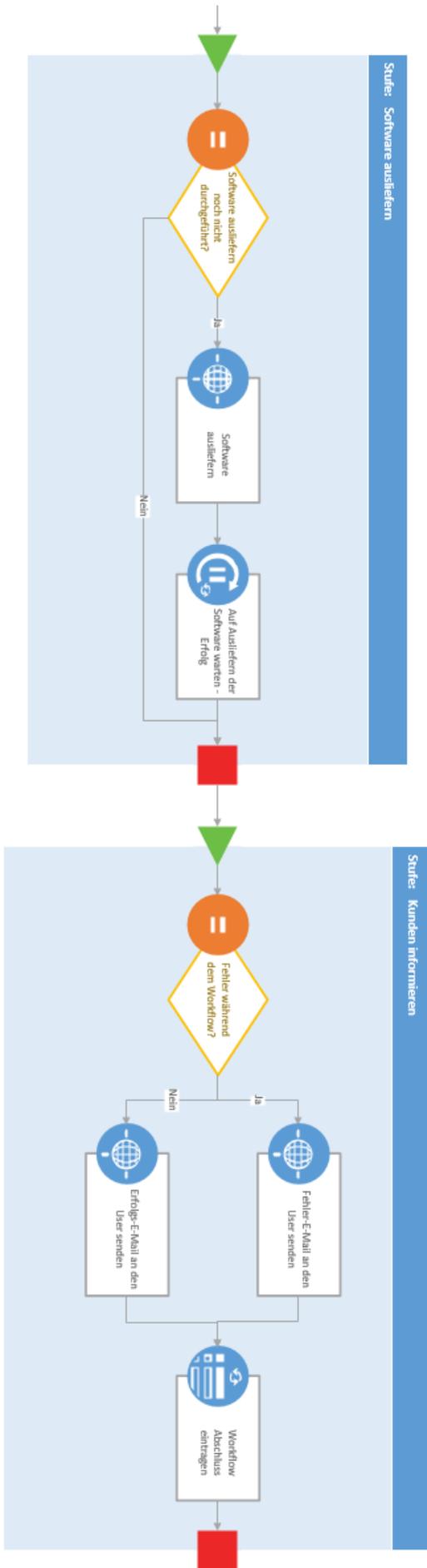


Abbildung 47: WWF Workflow aktueller Stand 10

4.3 Implementierung und Automatisierung des Workflows

Um den modellierten Workflow zu implementieren und automatisieren, mussten die User-Interfaces designt, die Datenstruktur erstellt und die Kommunikation mit dem Optionstore hergestellt werden.

4.3.1 User-Interfaces

Um den Workflow zu starten, sollten laut Aufgabenstellung zwei User-Interfaces (externes User-Interface für Kunden wird exkludiert, da es nicht im Aufgabenbereich dieser Masterarbeit ist) implementiert werden, die jeweils einem unterschiedlichen Einsatzszenario dienen. Das E-Mail-Interface ist die schnellste Variante, um den Workflow zu starten. Dem Benutzer müssen jedoch die bereits installierten beziehungsweise die verfügbaren Optionen der Formgebungsmaschine bekannt sein, da hier keine Hilfestellungen geboten werden. Zum Starten wird eine E-Mail an die jeweilige Adresse (in Abbildung 48 wird eine Beispiel E-Mail-Adresse angezeigt) mit einem Betreff, der UPD oder AOP enthält, gesendet. UPD steht für Update und führt ein Update der Software der Formgebungsmaschine auf die angegebene Version aus. AOP steht für Add Options und führt ein Upgrade der Software der Formgebungsmaschine aus, indem die angegebenen Optionen hinzugefügt/entfernt (beim Entfernen muss ein - vor die Option gesetzt werden) werden und auf die angegebene Version geupdatet wird. Die Parameter sind dabei durch “_” getrennt und die Optionen, wenn vorhanden, durch “;”. In Abbildung 48 werden beide Aufrufe dargestellt.

Senden	An...	xc200.test@engel.at
	Cc...	
	Betreff	UPD_208476_V4.40.10

Senden	An...	xc200.test@engel.at
	Cc...	
	Betreff	AOP_208476_V4.40.10_AirValve2;CoreFix1

Abbildung 48: E-Mail-Interface

Die zweite Variante, um den Workflow zu starten, ist das Verwenden der internen Webseite. Diese benötigt etwas länger, um den Workflow zu starten, da sie dem Benutzer mehr Hilfestellung bietet, indem die installierten Optionen, die verfügbaren Optionen, die verwendete Softwareversion und noch weitere Informationen angezeigt werden. In Abbildung 49 wird dargestellt, welche Informationen nach dem Eingeben der Fabrikationsnummer angezeigt werden. Im Feld “Control Version“ kann die gewünschte Versionsnummer der Software eingegeben werden und mittels dem “E-Mail Address“-Feld kann die E-Mail-Adresse, an welche die Abschluss-E-Mail des Workflows gesendet wird, festgelegt werden. In der Tabelle können

jene Optionen, die entfernt werden sollen, mittels dem Demarkieren einer Option beziehungsweise die, die hinzugefügt werden sollen, durch das Markieren einer Option festgelegt werden. Um den Workflow nach den getätigten Eingaben zu starten, reicht ein Klick auf den Update/Upgrade-Button.

SharePoint

BROWSE PAGE

ENGEL Control EDIT LINKS

MachineSoftwareUpdater

Updated Pages

MachineSoftwareUpdater

Home

Notebook

Documents

Recent

Common

Site Contents

EDIT LINKS

Fabrication Number 208476 Search

Control Version V4.40.00 V4.40.00 Reload

Machine Type victory 60/50 tech

Customer ENGEL AUSTRIA GMBH LAGER

Shipped Special Software

E-Mail Address Andreas.Winter@engel.at +

Options List Search..

Installed Not Installed

<input checked="" type="checkbox"/>	AcousticAlarm1 (V4.30.00)	Details >
<input checked="" type="checkbox"/>	AirValve1 (V4.30.00)	Details >
<input checked="" type="checkbox"/>	ChargeAmp1 (V4.30.00)	Details >
<input checked="" type="checkbox"/>	ConveyorBelt1 (V4.30.00)	Details >
<input checked="" type="checkbox"/>	CoreMove1 (V4.30.00)	Details >
<input checked="" type="checkbox"/>	CoreMove2 (V4.30.00)	Details >

Update/Upgrade Cancel

Abbildung 49: User-Interface interne Webseite

4.3.2 Datenstruktur

Da der Workflow ein Listen-Workflow ist, basiert dieser auf einer Liste und verwendet diese als seine Datenstruktur. In Abbildung 50 ist diese dargestellt.

Column Name	Type
Id	Single line of text
FabNo	Single line of text
Requesttype	Single line of text
OptionsList	Single line of text
ControlVersion	Single line of text
SenderType	Single line of text
FilePath	Single line of text
Email	Single line of text
SpecialSoftware	Yes/No (check box)
Shipped	Yes/No (check box)
SAPDataStep	Single line of text
SAPDataType	Single line of text
SoftwareStep	Single line of text
SoftwareType	Single line of text
DocuStep	Single line of text
DocuType	Single line of text
TestingStep	Single line of text
TestingType	Single line of text
BackupStep	Single line of text
BackupType	Single line of text
ShippingStep	Single line of text
ShippingType	Single line of text
FailedOnError	Yes/No (check box)
Finished	Yes/No (check box)
TriggerTime	Single line of text

Abbildung 50: Liste des Workflows

Die Id dient der Zuordnung des Listenitems zur Workflowinstanz. FabNo, Requesttype, OptionsList, ControlVersion, SenderType, FilePath und Email werden beim Erstellen eines neuen Listenitems benötigt. SpecialSoftware und Shipped werden aus dem SAP bezogen und sind für die Auslieferung der Software von hoher Bedeutung. SAPDataStep, SoftwareStep, DocuStep, TestingStep, BackupStep und ShippingStep werden verwendet, um den aktuellen Status der Workflow-Instanz festzustellen. Die jeweiligen Steps können dabei einen der Werte 0 (Default), 1 (Bevorstehend), 2 (Aktiv), 3 (Erfolgreich), 4 (Fehler, Erneut Versuchen), 5 (Fehler, Nicht mehr Versuchen) oder 6 (Ausgelassen) annehmen. Die sechs Typen SAPDataType, SoftwareType, DocuType, TestingType, BackupType und ShippingType legen fest, ob der dazugehörige Step automatisch (Type hat Wert 0) oder manuell (Type hat Wert 1) durchgeführt wird. FailedOnError indiziert, ob ein Fehler aufgetreten ist und Finished zeigt an, ob der Workflow bereits abgeschlossen ist. Die TriggerTime enthält den Zeitpunkt, an dem der Workflow gestartet wurde.

4.3.3 Kommunikation über REST-API

Um die Kommunikation zwischen der Workflow-Instanz in SharePoint und der Sammlung an Webservices im Optionstore zu ermöglichen, wird die REST-API SharePoints [30] verwendet. Diese ermöglicht das Interagieren eines heterogenen Systems mit der SharePoint-Umgebung, wodurch das Verändern von Feldern in Listenitems, das Erstellen von Listenitems und das Starten von Workflows ermöglicht wird. Im folgenden Kapitel werden ebenjene Interaktionen beispielhaft in der Programmiersprache C# gezeigt, wobei die Bibliothek Microsoft.SharePoint.Client verwendet wurde.

Als erstes wird das Erstellen eines Listenitems in Abbildung 51 gezeigt, da das Listenitem die Ausgangsbasis zum Starten eines Workflows ist (bei Listenworkflows). Zunächst wird eine Verbindung mittels ClientContext hergestellt, welche die Adresse der SharePoint-Seite, einen Benutzer und ein Passwort benötigt, um eine Verbindung mit der SharePoint-Seite herzustellen. Über diese Verbindung wird die Liste abgefragt, ein neues Listenitem für diese Liste erstellt und die geänderte Liste an die SharePoint-Seite gesendet.

```
ClientContext clientContext = GetClientContext();
List oList = clientContext.Web.Lists.GetByTitle(par.CONTROLWORKFLOWLIST);
ListItemCreationInformation itemCreateInfo = new ListItemCreationInformation();
ListItem oListItem = oList.AddItem(itemCreateInfo);
oListItem["FabNo"] = fabNr;
oListItem["Requesttype"] = requesttype;
oListItem["OptionsList"] = optionChanges;
oListItem["ControlVersion"] = version;
oListItem["SenderType"] = sendertype;
oListItem["FilePath"] = filepath;
oListItem["Email"] = email;
oListItem["TriggerTime"] = DateTime.Now;
oListItem.Update();
clientContext.ExecuteQuery();
```

Abbildung 51: Code: Listenitem erstellen

Auch beim Starten eines Workflows [12] auf einem Listenitem wird der ClientContext benötigt, um eine Verbindung mit der SharePoint-Seite herzustellen, was in Abbildung 52 gezeigt wird. Nachdem diese Verbindung hergestellt wurde, wird über die WorkflowSubscriptionCollection (beinhaltet alle veröffentlichten Workflows) der gewünschte Workflow geholt. Mittels den Initialisierungsdaten (initiationData) können Daten an den Workflow übergeben werden und die Funktion StartWorkflowOnListItem ermöglicht das Starten des Workflows auf dem erstellten Listenitem.

```

Web web = clientContext.Web;
WorkflowServicesManager wfServicesManager = new WorkflowServicesManager(clientContext, web);
WorkflowSubscriptionService wfSubscriptionService = wfServicesManager.GetWorkflowSubscriptionService();
WorkflowInstanceService wfInstanceService = wfServicesManager.GetWorkflowInstanceService();
WorkflowSubscriptionCollection wfSubscriptions = wfSubscriptionService.EnumerateSubscriptions();
string controllistworkflow = par.CONTROLLISTWORKFLOWV3;
clientContext.Load(wfSubscriptions, wfSubs => wfSubs.Where(wfSub => wfSub.Name == controllistworkflow));
clientContext.ExecuteQuery();
WorkflowSubscription wfSubscription = wfSubscriptions.First();
var initiationData = new Dictionary<string, object>
{
    { "parallelVar", false },
    { "dontwaitparallelVar", true }
};
wfInstanceService.StartWorkflowOnListItem(wfSubscription, oListItem.Id, initiationData);
clientContext.ExecuteQuery();

```

Abbildung 52: Code: Workflow starten

Um die Ausführung eines Workflows von außerhalb der SharePoint-Umgebung beeinflussen zu können, kann ein Listenelement via REST-API verändert werden, auf welches im Workflow gewartet wird. Hierbei muss abermals die Liste mittels ClientContext besorgt werden, von welcher das gewünschte Listenitem mithilfe dessen ID bezogen wird. Die Felder des Listenitems werden angepasst und das Listenitem wird zurück an die SharePoint-Seite gesendet.

```

oList = clientContext.Web.Lists.GetByTitle(par.CONTROLWORKFLOWLIST);
clientContext.Load(oList);
clientContext.ExecuteQuery();
listitem = oList.GetItemById(listItemID);
clientContext.Load(listitem);
clientContext.ExecuteQuery();
listitem[fieldname] = value;
listitem.Update();
clientContext.ExecuteQuery();
listitem = oList.GetItemById(listItemID);
clientContext.Load(listitem);
clientContext.ExecuteQuery();

```

Abbildung 53: Code: Listenitem ändern

5 Fazit - Erweiterungen und Entscheidungshilfe

Im Zuge dieser Arbeit wurde ein Workflow zum Automatisieren des Bestell- und Konfigurations-Prozesses von Software für Formgebungsmaschinen in Windows Workflow Foundation umgesetzt. Dieser wurde in einer SharePoint-Umgebung entwickelt und automatisiert und kann über zwei verschiedene Interfaces gestartet werden.

Für ebenen Workflow gibt es einige Erweiterungsmöglichkeiten, die nachfolgend dargelegt werden. Zum einen ist eine Erweiterung durch das Implementieren des dritten Interfaces möglich. Dieses wird bereits in einem anderen Projekt der Firma Engel umgesetzt und wird in näherer Zukunft hinzugefügt. Eine andere Erweiterungsmöglichkeit wäre das Implementieren des automatischen Dokumentationservices. Derzeit ist nur die Schnittstelle zum zukünftigen Service implementiert, weshalb die Dokumentation immer manuell erzeugt werden muss. Durch die Implementierung eines ausreichend qualitativen Dokumentationservices würde die Durchlaufzeit des Workflows deutlich sinken. Ein weiterer Ansatzpunkt zur Verbesserung des Workflows ist das Ersetzen des derzeit verwendeten statischen Testservices durch einen modularen Testservice. Der statische Testservice durchläuft für jede Formgebungsmaschine die selbe Testsequenz, wodurch nur die Basisfunktionen der jeweiligen Formgebungsmaschine getestet werden können. Ein modularer Testservice würde abhängig der installierten Optionen unterschiedliche Tests durchführen, wodurch alle Funktionalitäten der Formgebungsmaschine getestet, die Qualität des Workflows enorm steigen und die Fehlerrate deutlich sinken würden.

Weiters soll noch eine Entscheidungshilfe, für alle jene, die vor einer Modellierungssprachen-Entscheidung zwischen BPMN und WWF stehen, geboten werden.

BPMN	WWF
Wenn die zu modellierenden Prozesse als Kommunikationsmittel eingesetzt werden.	Wenn die Implementierung eines Workflows mit benutzerdefinierten Aktionen benötigt wird.
Wenn die Modellierung aller Prozesse eines Unternehmens im Zuge von BPM gefragt ist.	Wenn die Implementierung von Workflows in einer bestehenden Windows/SharePoint-Umgebung gefragt ist.
Wenn noch nie Geschäftsprozesse modelliert wurden und diese zum ersten Mal abgebildet werden.	Wenn das einzige Ziel der Modellierung die Automatisierung ist.
Wenn abteilungs- beziehungsweise unternehmensübergreifende Modelle benötigt werden.	
Wenn abstrakte Aktionen benötigt werden.	

Tabelle 5: Entscheidungshilfe zwischen BPMN und WWF

Tabellenverzeichnis

1 Bewertung der Kriterien zur Auswahl des Workflow-Management-Systems	24
2 Ausprägung der Kriterien für Camunda	28
3 Ausprägung der Kriterien für Bizagi	29
4 Ausprägung der Kriterien für SharePoint	30
5 Entscheidungshilfe zwischen BPMN und WWF	76

Literatur

- [1] Aguilar-Saven, R. S. (2004). Business process modelling: Review and framework. *International Journal of production economics*, 90(2):129–149.
- [2] Allweyer, T. (2016). *BPMN 2.0: introduction to the standard for business process modeling*. BoD–Books on Demand.
- [3] Andrews, T., Curbera, F., Dholakia, H., Golland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., et al. (2003). Business process execution language for web services.
- [4] Berger, M., Ellmer, E., Quirchmayr, G., and Zeitlinger, A. (1997). Evaluating workflow management systems. In *Database and Expert Systems Applications, 1997. Proceedings., Eighth International Workshop on*, pages 412–417. IEEE.
- [5] Biard, T., Le Mauff, A., Bigand, M., and Bourey, J.-P. (2015). Separation of decision modeling from business process modeling using new “Decision Model and Notation” (DMN) for automating operational decision-making. In *Working Conference on Virtual Enterprises*, pages 489–496. Springer.
- [6] Bizagi (2018). Bizagi - Digital Transformation & Business Process Management BPM.
- [7] BP Logix (2016). BPM vs Workflow: Is There a Difference? — BP Logix.
- [8] Camunda Services GmbH (2018). Workflow and Decision Automation Platform — Camunda BPM.
- [9] CBOK, B. (2009). Guide to the business process management common body of knowledge. *Versão 2.0. 2009. Disponível em: www.abmp.org. Acesso em: 25 nov 2012*, page 21.
- [10] Derungs, M., Vogler, P., and Österle, H. (1995). *Kriterienkatalog Workflow-Systeme*. Institut für Wirtschaftsinformatik.
- [11] Dijkman, R. M., Dumas, M., and Ouyang, C. (2007). Formal semantics and analysis of BPMN process models using Petri nets. *Queensland University of Technology, Tech. Rep.*
- [12] Dipongkor, A. (2015). How to start SharePoint Workflow programmatically using CSOM - SharePoint Stack Exchange.
- [13] Dumas, M., La Rosa, M., Mendling, J., Reijers, H. A., et al. (2013). *Fundamentals of business process management*, volume 1. Springer.

- [14] ENGEL AUSTRIA GmbH (2018). ENGEL Kunststoff Spritzgussmaschinen.
- [15] Fritz, H., Lettner, G., and Kern, A. (2014). Verfahren zum Nachrüsten von Komponenten für eine Formgebungsmaschine.
- [16] Gebhardt, K. F. (2017). Unified Modeling Language.
- [17] Georgakopoulos, D., Hornick, M., and Sheth, A. (1995). An overview of workflow management: From process modeling to workflow automation infrastructure. *Distributed and parallel Databases*, 3(2):119–153.
- [18] Glykas, M. M. (2011). Effort based performance measurement in business process management. *Knowledge and Process Management*, 18(1):10–33.
- [19] Hallerbach, A., Bauer, T., and Reichert, M. (2008). Managing Process Variants in the Process Lifecycle.
- [20] Hao, E. (2007). Porting SharePoint Designer Workflows to Visual Studio – Microsoft SharePoint Designer Team Blog.
- [21] Ho, A., Smith, S., and Hand, S. (2005). On deadlock, livelock, and forward progress. Technical report, University of Cambridge, Computer Laboratory.
- [22] Hollingsworth, D. and Hampshire, U. (1995). Workflow management coalition: The workflow reference model. *Document Number TC00-1003*, 19.
- [23] Hord, S., Plumley, M., and Hendrickson, J. (2018). Configure outgoing email for a SharePoint Server farm — Microsoft Docs.
- [24] Houy, C., Fettke, P., and Loos, P. (2010). Empirical research in business process management—analysis of an emerging field of research. *Business Process Management Journal*, 16(4):619–661.
- [25] Kopp, O., Martin, D., Wutke, D., and Leyman, F. (2009). The difference between graph-based and block-structured business process modelling languages. *Enterprise Modelling and Information Systems Architectures—International Journal of Conceptual Modeling*, 4(1):3–13.
- [26] Macedo de Morais, R., Kazan, S., Inês Dallavalle de Pádua, S., and Lucirton Costa, A. (2014). An analysis of BPM lifecycles: from a literature review to a framework proposal. *Business Process Management Journal*, 20(3):412–432.

- [27] Mackey, A. (2010). Windows communication foundation. In *Introducing .NET 4.0*, pages 159–173. Springer.
- [28] Microsoft (2011). Working with XAML.
- [29] Microsoft (2015a). Grundlegendes zu SharePoint 2013-Workflows.
- [30] Microsoft (2015b). REST API reference and samples.
- [31] MID GmbH (2010). MID GmbH: Ihr Partner in allen Themen der Enterprise IT.
- [32] Molina, G. (2009). *Comparison of WFMS based on Flexibility*. PhD thesis, Thesis on Master of Science in Management Operation and Logistics.
- [33] Netjes, M., Reijers, H. A., and van der Aalst, W. M. (2006). Supporting the BPM life-cycle with FileNet. In *Proceedings of the CAiSE*, volume 6, pages 497–508.
- [34] Pialorsi, P. (2013). *Microsoft SharePoint 2013 Developer Reference*. Pearson Education.
- [35] Quirk, E. (2017). Understanding the Difference Between Workflow Management and BPM.
- [36] Rajshekhar, A. (2013). *.Net Framework 4.5 Expert Programming Cookbook*. Packt Publishing Ltd.
- [37] Recker, J. C. and Mendling, J. (2006). On the translation between BPMN and BPEL: Conceptual mismatch between process modeling languages. In *The 18th International Conference on Advanced Information Systems Engineering. Proceedings of Workshops and Doctoral Consortium*, pages 521–532. Namur University Press.
- [38] Scheer, A.-W., Thomas, O., and Adam, O. (2005). Process modeling using event-driven process chains. *Process-Aware Information Systems*, pages 119–145.
- [39] Strahringer, S. et al. (1996). Konzept metamodellierter Methodenvergleiche. Technical report, Darmstadt Technical University, Department of Business Administration, Economics and Law, Institute for Business Studies (BWL).
- [40] Stöckler, K.-H. (2015). Prototypische Umsetzung der Automatisierung eines Prozesses für Nach- und Aufrüstung von Komponenten einer Formgebungsmaschine.
- [41] Tutorialspoint (2018). C++ goto statement.
- [42] Ultimus (2012). BPM vs Workflow vs. Low-Code: What are the differences?

- [43] Van der Aalst, W. M. (1998). The application of Petri nets to workflow management. *Journal of circuits, systems, and computers*, 8(01):21–66.
- [44] van der Aalst, W. M. (2003). Patterns and xpd: A critical evaluation of the xml process definition language. *BPM Center report BPM-03-09, BPMcenter.org*, pages 1–30.
- [45] van der Aalst, W. M. (2004). Business process management: A personal view. *Business Process Management Journal*, 10(2).
- [46] Verma, N. (2009). *Business process management: profiting from process*. Global India Publications.
- [47] Weske, M. (2007). *Business Process Management—Concepts, Languages, Architectures*.
- [48] White, S. A. (2004). Process modeling notations and workflow patterns. *Workflow handbook*, 2004:265–294.
- [49] Zur Muehlen, M. and Ho, D. T.-Y. (2005). Risk management in the BPM lifecycle. In *International Conference on Business Process Management*, pages 454–466. Springer.
- [50] zur Muhlen, M. (1999). Evaluation of workflow management systems using meta models. In *Systems Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on*, pages 11–pp. IEEE.

Abbildungsverzeichnis

1	BPM Lebenszyklus-Modell [9]	2
2	Architekturüberblick der Aufgabenstellung	5
3	Ablauf des Workflows zur automatischen Softwaregenerierung	6
4	Gegenüberstellung blockstrukturierte vs. graphenbasierte Modellierungssprachen	9
5	Kernset der BPMN-Elemente nach [11]	11
6	BPMN Tasktypen nach [31]	12
7	WWF-Elemente: Aktionen 1	15
8	WWF-Elemente: Aktionen 2	16
9	WWF-Elemente: Aktionen 3	16
10	WWF-Elemente: Aktionen 4	17
11	WWF-Elemente: Bedingungen	17
12	WWF-Elemente: Container und Abschlusszeichen	18
13	WWF XAML Datei	19
14	BPMN Beispiel-Prozess: Maschineninformationen darstellen	20
15	WWF Beispiel-Prozess: Maschineninformationen darstellen	20
16	SharePoint Architektur nach [29]	27
17	BPMN Workflow 1 nach [40]	35
18	BPMN Workflow 2 nach [40]	36
19	BPMN Workflow 3 nach [40]	37
20	BPMN Workflow Sicherheitsüberprüfung nach [40]	38
21	BPMN Workflow Optionsliste generieren nach [40]	39
22	BPMN Workflow Angebot generieren nach [40]	40
23	BPMN Workflow Software generieren 1 nach [40]	41
24	BPMN Workflow Software generieren 2 nach [40]	42
25	BPMN Workflow Auslieferung nach [40]	43
26	WWF Problem mit der If-Anweisung	45
27	WWF Workflow Stand 2015 1	47
28	WWF Workflow Stand 2015 2	48
29	WWF Workflow Stand 2015 3	49
30	WWF Workflow Stand 2015 4	50
31	WWF Workflow Stand 2015 5	51

32 WWF Workflow Stand 2015 6	52
33 WWF Workflow Stand 2015 7	53
34 WWF Workflow Stand 2015 8	54
35 WWF Workflow Stand 2015 9	55
36 WWF Workflow Stand 2015 10	56
37 WWF Workflow Stand 2015 11	57
38 WWF Workflow aktueller Stand 1	61
39 WWF Workflow aktueller Stand 2	62
40 WWF Workflow aktueller Stand 3	63
41 WWF Workflow aktueller Stand 4	64
42 WWF Workflow aktueller Stand 5	65
43 WWF Workflow aktueller Stand 6	66
44 WWF Workflow aktueller Stand 7	67
45 WWF Workflow aktueller Stand 8	68
46 WWF Workflow aktueller Stand 9	69
47 WWF Workflow aktueller Stand 10	70
48 E-Mail-Interface	71
49 Ulser-Interface interne Webseite	72
50 Liste des Workflows	73
51 Code: Listenitem erstellen	74
52 Code: Workflow starten	75
53 Code: Listenitem ändern	75