

# Konzeptionierung und Implementierung einer standardisierten RPA-Schnittstelle für eine Analytics-On-Demand-Self-Service-Lösung zur Bereitstellung von einkaufsbezogenen Daten in der Krones AG

Nadine Riedl  
Ostbayerische Technische Hochschule  
Regensburg  
Krones AG  
CP Processes and Systems  
Böhmerwaldstraße 5  
93073 Neutraubling  
nadine.riedl@krones.com

Stefan Janoschek  
Krones AG  
CP Processes and Systems  
Böhmerwaldstraße 5  
93073 Neutraubling  
stefan.janoschek@krones.com

Prof. Dr. Frank Herrmann  
Ostbayerische Technische Hochschule  
Regensburg  
Labor Wirtschaftsinformatik,  
SAP und Produktionslogistik  
Galgenbergstraße 32  
93053 Regensburg  
frank.herrmann@oth-regensburg.de

## ABSTRACT

Many companies, including Krones AG's Corporate Procurement (CP) Processes and Systems department, are using Robotic Process Automation (RPA) solutions to automate and optimize repetitive, time-consuming, human activities and processes to realize efficiency and time gains. An area of application of RPA that offers a lot of potential are data collection, preparation and provisioning tasks to enable Business Intelligence. An opportunity to accelerate the decision-making process by reducing the dependence on data specialists are Self-service analytics. Therefore new IT solutions like software-based process automations must be designed and implemented.

## Keywords

Robotic Process Automation, Business Intelligence

## EINFÜHRUNG

Robotic Process Automation gewinnt in den letzten Jahren immer mehr an Bedeutung. Während in einer Studie der IDG Research Services der Anteil der befragten Unternehmen, der RPA eine hohe bis sehr hohe Bedeutung beigemessen hat nur bei knapp 48 Prozent lag [1], hat sich der Anteil in der durch das Custom Research Team von CIO, CSO und COMPUTERWOCHE durchgeführten Folgestudie im Jahr 2022 deutlich, um rund 22 Prozent, auf knapp 70 Prozent erhöht. [2] Aktuelles und voraussichtliches Marktvolumen des globalen Markts für Robotic Process Automation belegen diesen Trend. Laut eines Berichts der Grand View Research wuchs der Markt zwischen 2021 und 2023 um insgesamt 81 Prozent auf einen Gesamtwert von 2,9 Milliarden US-Dollar.

Bis 2030 wird eine durchschnittliche jährliche Wachstumsrate (CAGR) in Höhe von 39,9 Prozent prognostiziert. [3]

CP Processes and Systems verwenden aktuell auf Python basierende RPA-Lösungen. Mithilfe dieser werden im Fachbereich Einkauf der Krones AG auftretende, zeitintensive, repetitive manuelle Tätigkeiten und Prozesse automatisiert, um Effizienz- und Zeitgewinne zu realisieren.

Dieser Artikel thematisiert die Konzeptionierung und Implementierung einer RPA-Schnittstelle eingebettet in ein existierendes Framework zur zeitgesteuerten Ausführung von Automatisierungsprozessen als Teil einer Analytics-On-Demand-Self-Service-Lösung. Nach einer genaueren Beschreibung der Problemstellung, werden zunächst die theoretischen Grundlagen formuliert und die technischen Rahmenbedingungen erläutert. Anschließend werden die Anforderungen an die RPA-Schnittstelle erarbeitet, um diese als Ausgangspunkt für die Konzeptionierung und Implementierung der einzelnen Komponenten, die für einen funktionierenden Analytics-On-Demand-Self-Service erforderlich sind, verwenden zu können.

## PROBLEMSTELLUNG

Corporate Procurement (CP) Processes and Systems ist im Einkauf der Krones AG als Single Point of Contact die primäre Anlaufstelle für die Bereitstellung operativer Daten in aufbereiteter Form. Erfüllen die zentral über Business-Intelligence-Berichte zur Verfügung gestellten Daten nicht die Bedürfnisse der Mitarbeitenden des Einkaufs, muss von diesen ein Ticket mit Angaben zu den benötigten Daten im internen Ticketsystem erstellt werden. Die Bearbeitung einer solchen Analytics-Aufgabe ist infolge manueller Verarbeitungsschritte und manueller Schnittstellen verschiedener IT-Anwendungen mit einer hohen Bearbeitungszeit verbunden. Zusätzlich führen eine hohe Arbeitslast und Personalengpässe bei der Ticket-Bearbeitung einer Analytics-Aufgabe zu einer langen Durchlaufzeit von durchschnittlich 48 Stunden und somit im Vergleich zur durchschnittlichen Durchlaufzeit eines regulären Tickets in Höhe von 12 Stunden zu einer vierfachen Antwortzeit.

Somit stellt sich die Frage, wie Analytics-Aufgaben zukünftig so verarbeitet werden können, dass ihre Bearbeitungszeit und die Antwortzeit ihrer Tickets reduziert werden können. Angesichts der vorhandenen RPA-Infrastruktur empfiehlt es sich aus wirtschaftlicher Perspektive eine RPA-Lösung zu entwickeln. Dadurch, dass in einem Ticket die Daten in unstrukturierter Form vorliegen, wurde eine klassische RPA-Lösung, die nur strukturierte Daten verarbeitet, ausgeschlossen. Die Komplexität der in der Literatur vorgeschlagenen KI-basierten Vorgehensweise zur Verarbeitung unstrukturierter Daten [4][5] übersteigt die verfügbaren Ressourcen. Eine Alternative stellt ein auf RPA gestützter Prozess dar, der in Form eines Analytics-On-Demand-Self-Services strukturierte Anfragen zu Analytics-Aufgaben über ein Graphical User Interface (GUI) entgegennimmt und ihre Bearbeitung über RPA-Lösungen automatisiert, sodass dem Endanwender das Ergebnis der Analytics-Aufgabe in kurzer Zeit zur Verfügung gestellt werden kann.

Ziel ist die Konzeptionierung und Implementierung eines RPAs, der im Sinne eines Präprozessors und Controllers als Schnittstelle zwischen GUI und individueller RPA-Lösungen agiert. Die Evaluation und Verifizierung der Funktionalität der RPA-Schnittstelle soll anhand der als Proof-Of-Concept zu entwickelnden Analytics-Use-Cases der Orderbuchabfrage und Abfrage von Einkaufsinfosätzen mit Konditionen erfolgen.

## BEGRIFFSDEFINITIONEN

### Business Intelligence

Zu Business Intelligence existieren in der Fachliteratur verschiedene Definitionen. Die Fachgruppe Business Intelligence & Analytics der Gesellschaft für Informatik unterscheidet die unterschiedlichen Definitionsansätze nach BI als Klasse von IT-Systemen, BI als IT-Architektur, BI als Sammelbegriff für Technologien und Konzepte zur Entscheidungsunterstützung, BI als integrierter Ansatz zur Entscheidungsunterstützung und BI als Wissensgenerierungsprozess. [6] Allen Definitionen gemein ist das auf IT basierende Treffen von Entscheidungen. Für ein in diesem Artikel geltendes einheitliches Begriffsverständnis wird Business Intelligence nach Chamoni und Linden als „IT-basierter Prozess der Informationsgenerierung, der die Entscheidungsfindung innerhalb der unternehmensrelevanten Planung, Steuerung und Kontrolle unterstützt“, definiert. [7] zit. [8]

### Robotic Process Automation

RPA ist eine Technologie, bei der digitale Roboter zur Automatisierung ganzer Geschäfts- und Verwaltungsprozesse oder einzelner Prozessschritte eingesetzt werden. [5] [9] Im Gegensatz zu den physisch existenten Robotern, die aus der industriellen Fertigung bekannt sind, handelt es sich bei den RPA-Robotern um Softwareprogramme, die darauf abzielen, einen Menschen bei der Ausführung diverser informationstechnischer Tätigkeiten zu unterstützen oder ihn vollständig zu ersetzen. [10] Eine Automatisierung erfolgt durch die Imitation manueller Benutzerinteraktionen unter Anwendung von Wenn-Dann-Regeln auf strukturierten Daten. [11]

Bei RPA wird zwischen „Attended RPA“ und „Unattended RPA“ differenziert. Während „Attended RPA“ auch als „Robot Desktop Automation“ (RDA) bezeichnet wird und einen

Software-Roboter beschreibt, der direkt auf dem Computer der nutzenden Person durch diese selbst im Sinne eines digitalen Assistenten ausgeführt wird, handelt es sich bei „Unattended RPA“ um einen Software-Roboter, der automatisiert gestartet wird und aufgrund eines eigenen Users mit Passwort im Sinne eines virtuellen Mitarbeitenden agiert. [9] [12]

In Abgrenzung zu herkömmlichen Lösungen zur Prozessautomatisierung geht RPA nicht nach der Inside-Out-Methode vor, sondern verfolgt einen Outside-In-Ansatz, bei dem keine programmiertechnischen Interventionen in bestehende IT-Systeme und IT-Anwendungen vorgenommen werden. [13] Stattdessen setzt RPA auf die vorhandene IT auf und verwendet zur Automatisierung eine Kombination von Zugriffen auf grafische Benutzeroberflächen, Bildschirmkoordinaten, Programmierschnittstellen und Zugriffsschichten von Betriebssystemen und Datenbanken. [5]

## TECHNISCHE RAHMENBEDINGUNGEN

### RPA-System

Die Umsetzung von Automatisierungsprojekten bei CP Processes and Systems erfolgt unter Verwendung des von Hr. Janoschek eigenentwickelten RPA-Systems.

### RPA-Bots

Ein jeder RPA-Bot ist ein sog. Standard-RPA, der entweder über einen Trigger oder zeitgesteuert über einen Scheduler gestartet werden kann. Eine Sonderform ist der Timer-RPA, der Bestandteil der RPA-Steuerungsumgebung ist und die Aufgabe besitzt andere RPAs zu starten.

### Entwicklungsumgebung

Die Entwicklung von RPA-Lösungen findet über den Quellcode-Editor Visual Studio Code von Microsoft statt. Zur Programmierung der RPAs wird die Programmiersprache Python verwendet. Abgegrenzt von der Entwicklungsumgebung wird die Produktivumgebung. Während die Entwicklung lokal auf dem Rechner eines Entwicklers stattfindet, erfolgt die Produktivsetzung auf dem Server.

### Framework

Zur Lösung wiederkehrender Aufgaben wurde ein Framework aufgebaut. Jeder RPA, der über das RPA-Framework erstellt wird, wird auf dem Server nach derselben Grundstruktur angelegt. Um den neu erstellten RPA den spezifischen Anforderungen der zu erledigenden Aufgabe entsprechend zu erweitern, wird der RPA lokal installiert und der Quellcode in der Datei „rpa\_module.py“ implementiert. Darüber hinaus stellt das Framework unter Nutzung verschiedener Python-Bibliotheken allgemeine Software-Module, die als Klassen-Module in einem Paket gruppiert wurden, zur Verfügung. Diese können durch das Hinzufügen neuer Funktionalität erweitert und durch Parametrisierung an die Aufgabe eines RPAs angepasst werden.

### Steuerungsumgebung

Zur automatischen Ausführung eines RPA-Bots wird in der Windows Aufgabenplanung in einer virtuellen Maschine eine neue Aufgabe angelegt, die zu einem vorgegebenen Zeitpunkt und in einem wiederkehrenden Zeitraum (alle fünf

Minuten) den Timer-RPA über die Ausführung einer Batch-Datei startet. Zur Steuerung, welcher RPA-Bot, wie gestartet werden soll, ist eine Datenbank angebunden. Generell existieren zum Start eines Standard-RPAs die beiden Möglichkeiten „via Trigger“ und „via Timer“. Während sich bei „Start via Trigger“ an einem definierten Ort ein Trigger-File befinden muss, damit der Start eines RPAs getriggert wird, signalisiert bei „Start via Timer“ ein Timer in Form einer gesetzten Uhrzeit, wann die Ausführung eines RPAs überfällig ist. Der eigentliche Ausführungsstart eines RPA-Bots wird durch den RPA-Timer über ein Start-File initiiert. Wurden mehrere Start-Files für jeweils verschiedene RPA-Bots erzeugt, erfolgt eine parallele Abarbeitung des Programm-codes der RPAs.

## Workflow-Automatisierung mit Microsoft Power Plattform und Microsoft Office 365

Für die Automatisierung von Prozessen mit Userinteraktionen wird neben dem RPA-Framework auf eine Kombination der Microsoft-Anwendungen Power Apps, SharePoint und Power Automate zurückgegriffen. Über eine Power App können die für den Workflow erforderlichen Eingaben entweder über Eingabefelder durch den Anwender oder automatisiert über die Power App erfasst werden. Während bei der manuellen Erfassung exemplarisch die Eingabe einer Lieferantennummer genannt werden kann, kann bei der automatisierten Erfassung die E-Mail-Adresse des Anwenders angeführt werden. Als Datenbank fungieren SharePoint-Listen, die als Datenquellen in der Power App zu hinterlegen sind. Dadurch können die in der SharePoint-Liste enthaltenen Daten über die Power App sowohl geladen als auch durch die Erfassung neuer Daten oder die Modifikation bestehender Daten transformiert werden. Zugleich fungiert die SharePoint-Liste als Basis für die robotergesteuerten Prozessautomatisierung mit Power Automate. Ein Power Automate Workflow, der die einzelnen zu automatisierenden Schritte enthält, kann zum einen durch das Erzeugen oder Verändern eines SharePoint-Listeneintrags ausgelöst werden und zum anderen bei der Realisierung der Geschäftslogik auf die SharePoint-Liste als Datenbank zurückgreifen. Das Zusammenspiel der einzelnen Komponenten der Workflow-Automatisierung wird in der nachfolgenden Abbildung 1 veranschaulicht.

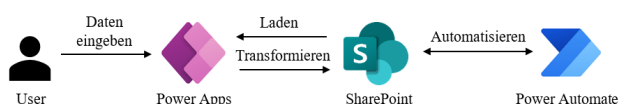


Abbildung 1: Workflow-Automatisierung mit Userinteraktion

## ANALYTICS-USE-CASES

### Orderbuch

Das Orderbuch ist ein Werkzeug des Enterprise-Resource-Planning (ERP)-Systems von SAP für die Bezugsquellenermittlung. Systemtechnisch wird das Orderbuch als Liste, in der die für ein Material in einem bestimmten Werk und einem vorgegebenen Zeitraum erlaubten und nicht erlaubten Bezugsquellen definiert sind, abgebildet. [14] Die Daten des Orderbuchs werden in der SAP-Standardtabelle „EORD“ gespeichert.

Die Bezugsquellen eines bestimmten Materials eines bestimmten Werks können über die SAP-Standardtransaktion „ME03“ angezeigt werden. Eine Abfrage des Orderbuchs zu einem Lieferanten ist über diese Transaktion nicht möglich. [15]

Benötigen Mitarbeitende des Einkaufs die Daten dennoch, müssen sie den mit einem administrativen und zeitlichen Aufwand in Höhe von fünf bis zehn Minuten verbundenen Weg über das interne Ticketsystem gehen. Eine Antwort auf ihr Ticket erhalten sie nach durchschnittlich zwei Tagen. Die Bearbeitung des Tickets nimmt durchschnittlich 15 Minuten in Anspruch.

Für die Bearbeitung eines Tickets zur Orderbuchabfrage muss von einem Mitarbeitenden von CP Processes and Systems anfrageabhängig eine ein bis dreimalige Abfrage der Datenbanktabelle „EORD“ über die SAP-Standardtransaktion „SE16“ mit anschließendem Download des Ergebnisses als CSV-Datei durchgeführt werden. Das Ergebnis der Abfrage muss anschließend als Excel-Datei aufbereitet als Anlage des Tickets dem Anfragestellendem bereitgestellt werden.

Relevant für einen Einkaufenden ist die Möglichkeit eine Abfrage zu

- Material und Lieferanten,
- Material, Lieferanten und Werk,
- für ein Werk und Lieferanten  
als fixiert definierten Orderbuchsätzen

zu starten.

### Einkaufsinfosatz mit Konditionen

Ein Einkaufsinfosatz fungiert als Informationsquelle des Einkaufs, indem er unter anderem Auskunft darüber erteilt, zu welchen Preisen und Konditionen ein Lieferant liefern kann. Grundsätzlich wird dazu ein Einkaufsinfosatz mit Einkaufsinformationen zu einem bestimmten Material und dem Lieferanten des Materials in SAP definiert. [16]

Für ein zielorientiertes Führen einer Jahrespreisverhandlung ist eine fundierte Vorbereitung des verhandlungsführenden Mitarbeitenden unumgänglich. Grundlage dafür ist ein Abzug der im SAP-System hinterlegten, aktuell gültigen Einkaufsinfosätze zusammen mit ihren Konditionen.

Die Ausführungsdauer einer solchen Infosatzabfrage mit Konditionen ist mit einem zeitlichen Aufwand, der je nach den Fähigkeiten eines Nutzers zwischen 10 und 30 Minuten variiert, verbunden.

Für einen Abzug muss zunächst ein lieferanten- und einkaufsorganisationsspezifischer und gegebenenfalls materialspezifischer Export der über einen BI-Bericht zur Verfügung gestellten Einkaufsinfosätze erfolgen. Im Ergebnis der Infosatzabfrage tritt ein Einkaufsinfosatz mehrmalig auf, weil jede im Einkaufsinfosatz hinterlegte Kondition eine Zeile besitzt. Für eine lesbarere Darstellung und um das unmittelbare Einspielen von Preis- und Konditionsänderungen über eine vorhandene Automatisierungsschnittstelle in SAP zu ermöglichen, müssen die Einkaufsinfosätze über das sogenannte „Single-Liner-Tool“, bei dem es sich um eine auf Excel VBA basierende Robot Desktop Automation Anwendung handelt, aufbereitet werden.

## ANFORDERUNGSANALYSE

Die Anforderungsanalyse ist im Kontext eines systematischen Software-Entwicklungsprozess in verschiedene Vorgehensmodelle zur System- und Softwareentwicklung integriert. [17] Die Qualität der Anforderungen gilt als Garant für den Erfolg eines Projekts. [18] Das folgende Kapitel beschäftigt sich damit, einen geeigneten Anforderungskatalog als Basis für die Konzeptionierung und Implementierung der genannten Komponenten zu erarbeiten. Für eine RPA-Lösung wird es dabei als vorteilhaft erachtet den Funktionsumfang auf ein Minimum Viable Product (MVP) zu beschränken. [4] Ähnliche, teilweise weitaus umfangreichere Anforderungsanalysen werden im Zuge eines Requirements Engineerings in Funktionsbereichen und Unternehmen, die auf Software-Entwicklung spezialisiert sind, durchgeführt.

Pohl und Rupp (s. [18]) legen verschiedene Techniken zur Ermittlung von Anforderungen fest und betonen dabei die situationsgerechte Anwendung, um ein an ein Projekt angepasstes, möglichst vollständiges Anforderungsprofil zu erhalten. Aus Perspektive des zu realisierenden Projekts wurde das Führen eines Experteninterviews und die Darstellung des Anwendungsablaufs des Analytics-On-Demand-Self-Services angewandt. Die Anforderungen wurden in zwei Granularitätsstufen spezifiziert, wobei eine Kombination aus modellbasierter und natürlichsprachiger Dokumentation, die in der Fachliteratur zur Nutzung der Vorteile beider Dokumentationsformen empfohlen wird, verwendet wurde. [18]

Für die modellbasierte Dokumentation als Use-Case-Diagramm, das in Abbildung 2 vereinfacht dargestellt ist, fungieren Struktur und Verhalten des Analytics-On-Demand-Self-Services als primäre Quellen. Der User, der als eine Person, die den Analytics-On-Demand-Self-Service in Anspruch nimmt, definiert wird, startet eine Anfrage bzw. einen Request, indem er meldet, welche Daten er benötigt. Daraufhin startet zeitgesteuert ein RPA, der die Vorverarbeitung des Requests übernimmt und deshalb im weiteren Verlauf als Preprocessor-RPA bezeichnet wird. Der Preprocessor-RPA erzeugt aus den Requestdaten eine Datei für den Transfer zu einem sog. Analytics-RPA, der, als ein für eine Analytics-Aufgabe spezifischer RPA, nach einem durch den Preprocessor-RPA initiiertem, zeitgesteuertem Start die Verarbeitung und Ergebnisbereitstellung übernimmt.

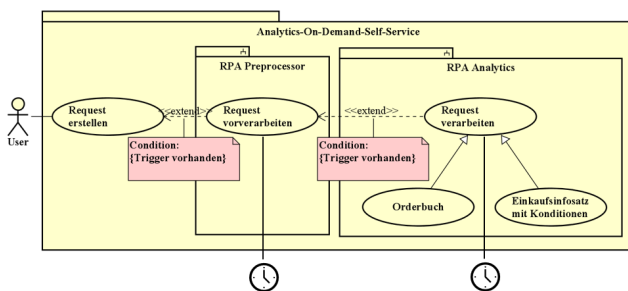


Abbildung 2: Use-Case-Diagramm zum Analytics-On-Demand-Service

Im nächsten Schritt wurden daraus Anforderungen, die sich aus den einzelnen Use Cases ergeben und auf die Konzeptionierung und Implementierung der RPA-Schnittstelle und der

Verifizierung ihrer Funktionalität Einfluss haben, abgeleitet. Die Anforderungen wurden gruppiert nach den Systemkomponenten, auf die sie sich beziehen natürlichsprachlich dokumentiert und zur Nachverfolgbarkeit und eindeutigen Identifizierung fortlaufend nummeriert. Um Unterschiede hinsichtlich der Relevanz der Anforderungen zu berücksichtigen, wurde eine dreistufige Priorisierung der Anforderungen im Hinblick auf den Erfolg des Projekts vorgenommen. [18] Dazu wurden die Anforderungen in Muss- Soll- und Kann-Anforderungen klassifiziert und entsprechend gewichtet: [19]

- **Muss-Anforderung** (Gewichtung: 3)  
Die Anforderung ist zwingend zu erfüllen, um den Erfolg des Projekts nicht zu gefährden.
- **Soll-Anforderung** (Gewichtung: 2)  
Die Anforderung hat eine hohe Relevanz, ist aber nicht zwingend für den Erfolg des Projekts erforderlich.
- **Kann-Anforderung** (Gewichtung: 1)  
Die Anforderung hat eine geringe Relevanz und gefährdet den Erfolg des Projekts nicht.

Die Anforderungen werden im Folgenden in verkürzter Form aufgelistet.

Preprocessor-RPA:

- A01: Zeitgesteuerter Start via Timer (Muss, 3)
- A02: Vorverarbeitung bei neuem Request (Soll, 2)
- A03: Lesen der Request-Daten (Muss, 3)
- A04: Validierung der Eingabedaten (Muss, 3)
- A05: Reklamation fehlender Daten (Kann, 1)
- A06: Aufbereitung der Request-Daten (Muss, 3)
- A07: Initiierung Start des Analytics-RPAs (Muss, 3)

Analytics-RPA:

- A08: Zeitgesteuerter Start via Trigger (Muss, 3)
- A09: Lesen der aufbereiteten Request-Daten (Muss, 3)
- A10: Verarbeitung des Requests (Muss, 3)
- A11: Bereitstellung des Ergebnisses (Muss, 3)

Allgemein:

- A12: Zentrale Datenhaltung (Muss, 3)
- A13: Identifikation eines Requests (Muss, 3)
- A14: Feste und variable Request-Daten (Soll, 2)
- A15: Statustracking (Muss, 3)
- A16: Nachrichtendienst (Soll, 2)
- A17: Berechtigungskonzept (Soll, 2)

## KONZEPTIONIERUNG

Um die speziellen Anforderungen eines usergesteuerten, automatisierten Prozesses zur Abarbeitung von Analytics-Aufgaben zu erfüllen, bedarf es einer individuellen Software-Lösung. Im Folgenden wird das aus den Anforderungen entworfene Konzept für die zu implementierende RPA-Lösung präsentiert.

### Zentrale Datenhaltung

Die Basis für einen funktionierenden Analytics-On-Demand-Self-Service, der sich über mehrere Komponenten erstreckt, ist eine gemeinsame Datengrundlage. Diese soll in Bezug auf

die Requestverwaltung als Single Point of Truth fungieren (A12). Werden die Daten zentral gespeichert, sind alle Komponenten zur Vereinheitlichung verpflichtet und eine effektive Datenhaltung wird ermöglicht. [20]

Als Datenquelle dient, wie bei der Workflow-Automatisierung bei CP Processes and Systems üblich, eine SharePoint-Liste. Die SharePoint-Liste enthält die durch einen User aufgegebenen Requests und wird durch den Preprocessor-RPA und die Analytics-RPAs als Datenbasis zur Vorverarbeitung und Verarbeitung eines Requests verwendet. Neben festen Spalten, die für die Bearbeitung des Requests erforderlich sind, zu denen u.a. die Request-ID gehört (A13), enthält die SharePoint-Liste eine beliebig erweiterbare Anzahl an Eingabespalten, in denen die Benutzereingaben gespeichert werden (A14).

### Zeitgesteuerter Start

Zur automatischen, zeitgesteuerten Ausführung des Preprocessor-RPAs (A01) und zum automatischen Start des Analytics-RPAs bei vorhandenem Trigger nach Zeitplan (A08), ist es erforderlich beide RPAs manuell in der Datenbank, die für die Steuerung der RPAs verantwortlich ist, zu erfassen.

### Trigger

Nachhaltigkeitsaspekte haben zunehmend Einfluss auf die Entwicklung und den Betrieb von Software. Eine Steigerung von Performance und Verfügbarkeit ohne zusätzlicher Ressourcen lässt sich insbesondere durch einen ressourceneffizienten Betrieb von Software erreichen. [21] Um Rechenressourcen effizient zu nutzen und sie nicht unnötigerweise zu belasten, soll ein RPA nur ausgeführt bzw. gestartet werden, wenn ein Request vorverarbeitet oder verarbeitet werden muss. Aus diesem Grund bedarf es sowohl eines Triggers für den Preprocessor-RPA als auch eines Triggers für den Analytics-RPA.

Der Trigger für den Preprocessor-RPA soll einfacherweise über einen Power Automate Workflow gesetzt werden, wenn ein neuer Request in der Sharepoint-Liste erfasst wird. Das Vorhandensein des Triggers signalisiert dem Preprocessor-RPA, dass mindestens ein neuer Request in der SharePoint-Liste erfasst wurde und er deshalb mit der Vorverarbeitung starten kann (A02).

Der Trigger für den Analytics-RPA muss durch den Preprocessor-RPA gesetzt werden, wenn ein Request vorverarbeitet wurde. Das Vorhandensein des Triggers muss dem Analytics-RPA signalisieren, dass er seine Analytics-Aufgabe auf Basis der von dem Preprocessor-RPA erzeugten Datei zum Datenaustausch ausführen kann (A07). Der Trigger muss dem Trigger zur Steuerung eines RPAs entsprechen, d.h. es muss sich um eine leere Text-Datei handeln, deren Name sich aus der auszuführenden Maschine und dem auszuführenden RPA zusammensetzt, und die an einem vordefinierten Speicherort erstellt wird.

### RPA-Steuerungsliste

Für die Initiierung des Starts eines Analytics-RPAs muss der Preprocessor-RPA Kenntnis darüber besitzen, welcher RPA mit welcher Aufgabe betraut ist. Zur anwenderfreundlichen Erweiterung, die eine gute RPA-Software auszeichnet [5], wird die Zuordnung nicht fest in den Quellcode des

Preprocessor-RPAs implementiert, sondern über eine Tabelle in Form einer Excel-Liste zur Verfügung gestellt. Darüber hinaus ist es für die Validierung der Benutzereingaben erforderlich, zu wissen, welche Eingabedaten für welche Analytics-Aufgabe benötigt werden. Um die aus einer weiteren Datenquelle resultierende Komplexität des Analytics-On-Demand-Self-Services zu reduzieren, bietet es sich an die Verknüpfung der Inputparameter mit der Verknüpfung eines RPAs zur Aufgabe vorzunehmen. Dabei entstehen durch die unterschiedlichen Konstellationen von Eingabeparametern verschiedene Varianten einer Aufgabe.

### Statustracking

Um den Status eines Requests zu überwachen und um die richtige Reihenfolge bei der Abarbeitung eines Requests einzuhalten (A15), ist es notwendig den Request-Lebenszyklus in mehrere Einzelphasen zu unterteilen: Initial, In Progress, Done und Fulfilled. Wie aus der schematischen Abbildung 3 ersichtlich wird, kann jede Phase, die systemintern durch eine Ziffer von null bis drei repräsentiert wird, einer Komponente des Analytics-On-Demand-Services zugeordnet werden. Während ein Request beim Anlegen den initialen Status 0 erhält, wird der Status durch den Preprocessor-RPA zu Beginn der Vorverarbeitung auf den Status 1 – In Progress – und am Ende auf den Status 2 – Done – gesetzt. Der Analytics-RPA kennzeichnet nach Verarbeitung eines Requests den Request mit Status 3 – Fulfilled – als abgeschlossen. Tritt innerhalb des Preprocessor-RPAs oder des Analytics-RPAs ein Fehler auf, ist vorzeitig das Ende des Request-Lebenszyklus erreicht und der Status 4 – Cancelled – kennzeichnet den Abbruch des Requests.

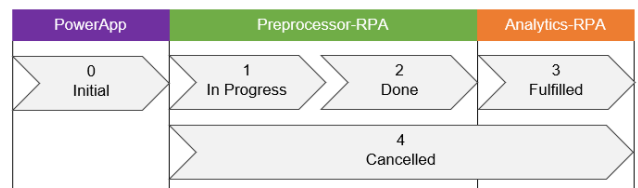


Abbildung 3: Request-Lebenszyklus im Kontext des Analytics-On-Demand-Services

### Validierungsprüfungen

Um bei der Verarbeitung eines Requests Fehler durch Benutzereingaben zu vermeiden, werden die Benutzereingaben durch den Preprocessor-RPA zunächst auf Plausibilität getestet (A04).

Zu den möglichen Eingabedaten des Analytics-On-Demand-Self-Services, die bei der Verwendung zur Orderbuch- und Einkaufsinfosatzabfrage mit Konditionen validiert werden müssen, gehören eine oder mehrere Lieferantennummern, Materialnummern, Werke und Einkaufsorganisationen. Eine Lieferantenummer ist gültig, wenn sie entweder aus dem 600000er oder aus dem 650000000er Nummernkreis stammt. Eine Materialnummer ist gültig, wenn sie zehn Stellen besitzt. Ein Werk und eine Einkaufsorganisation müssen auf Vierstelligkeit überprüft werden. Außerdem ist es für die Orderbuchabfrage erforderlich, dass überprüft wird, ob das FIX-Kennzeichen gesetzt ist.



## Nachrichtendienst

Unter Nachrichtendienst werden die Tätigkeiten, die im Rahmen des Versendens von Nachrichten an den User ergriffen werden müssen, subsumiert. Um den User über den Erfolg oder Misserfolg eines Requests zu informieren, werden zwei verschiedene Nachrichtenarten definiert – eine „Fulfilled-Message“, wenn der Request erfolgreich verarbeitet werden konnte und eine „Cancelled-Message“, wenn während der Request Vorverarbeitung oder Verarbeitung ein Fehler aufgetreten ist, der zum Abbruch geführt hat (A16).

Jede Nachricht besteht aus einem standardisierten Nachrichtentext mit Platzhaltern für dynamischen Inhalt, um eine Nachricht request- und userspezifisch formulieren zu können.

Weil der Versand von Nachrichten sowohl von dem Preprocessor-RPA als auch von einem Analytics-RPA erfolgen muss, wird der Nachrichtentext für eine bessere Wartbarkeit nicht direkt in den Programmcode eingebunden, sondern in Form von HTML-Dateien den einzelnen RPAs als Templates bereitgestellt.

Die beiden üblichen Wege des digitalen Nachrichtenaustauschs in der Krones AG sind der Versand von E-Mails über Outlook und der Versand von Nachrichten über Microsoft Teams. Aufgrund der Vielzahl an bei Mitarbeitenden eingehenden E-Mails aus dem Tagesgeschäft und des auf 5000 zu versendende E-Mails beschränkten Limits eines Outlook-Postfachs [22] wurde Microsoft Teams als Standard-Kommunikationskanal für die Benachrichtigung eines Users des Analytics-On-Demand-Self-Services gewählt und eine Benachrichtigung per E-Mail wird dem User als Auswahloption zur Verfügung gestellt.

Zur Benachrichtigung eines Users über den aktuellen Status des Requests werden zwei existierende, standardisierte Power-Automate-Workflows genutzt. Während der eine Workflow für den Versand von E-Mails über Outlook zuständig ist, wird der andere für den Versand von Nachrichten über Microsoft Teams eingesetzt. Es handelt sich dabei jeweils um einen automatisierten Cloud-Flow, der ausgelöst wird, wenn in einer vorgegebenen SharePoint-Liste ein neuer Eintrag erstellt wird.

## Datenaustausch

Dadurch, dass die durch den Preprocessor-RPA vorverarbeiteten Daten von verschiedenen Analytics-RPAs als Eingabedaten verwendet werden müssen, ist es notwendig, dass die Daten standardisiert in einer Datei zum Datenaustausch aufbereitet werden, sodass sie von einem beliebigen an den Prozess angebotenen RPA verarbeitet werden können (A06). Die Vereinheitlichung auf semantischer Ebene (Inhalt) und syntaktischer (Format) sind Voraussetzungen für die Gewährleistung von Interoperabilität verschiedener Systeme und einer maschinellen Weiterverarbeitung von Daten. [23]

### Semantische Ebene

Die Datei für den Datenaustausch muss die Request-Daten enthalten, die ein Analytics-RPA für die Verarbeitung eines Requests benötigt. Neben den validierten Eingabewerten eines Users, sind das die grundsätzlichen Request-Daten „ID des Requests“, „E-Mail-Adresse des Users“, „Wunsch

des Users nach E-Mail-Benachrichtigung“ und „Variante der Analytics-Aufgabe“. Um die grundsätzlichen Daten von den Benutzereingaben unterscheiden zu können, soll deren Bezeichnung jeweils ein „r\_“ vorangestellt werden.

### Syntaktische Ebene

Für das Format der Datei zum Datenaustausch bieten sich verschiedene Datenaustauschformate, wie JSON, XML und CSV an. [24] Als wichtige Kriterien, die es bei der Auswahl des Dateiformats zu beachten gilt, wurde die programmertechnische Handhabbarkeit mit Python, die Eignung für die Struktur und Komplexität der auszutauschenden Daten und die Les- und Schreibbarkeit zur Vereinfachung von Entwicklung und Wartung identifiziert.

Um die verschiedenen Datenaustauschformate im Hinblick auf die Erfüllung der drei Kriterien vergleichend gegenüberzustellen, wurde eine Entscheidungsmatrix erstellt, in der eine zahlenbasierte Bewertung von 1 bis 5 vorgenommen wurde. Je höher die Bewertung eines Formats, desto höher der Erfüllungsgrad des Kriteriums im Vergleich zu den anderen Formaten. Die Bewertung der einzelnen Kriterien kann in der nachfolgenden Tabelle 4 der Entscheidungsmatrix eingesehen werden.

Kriterium	JSON	XML	CSV
Handhabbarkeit mit Python	5	3	3
Eignung für die Struktur und Komplexität der auszutauschenden Daten	5	4	5
Les- und Schreibbarkeit	4	3	5
	14	10	13

Abbildung 4: Entscheidungsmatrix für das Format zum Datenaustausch

Für die Bewertung wurden nachfolgende Aspekte berücksichtigt:

#### Handhabbarkeit mit Python

Die Python-Standardbibliothek stellt für jedes der drei Datenformate ein Modul zur Verarbeitung zur Verfügung. Dadurch, dass JSON im Vergleich zu den anderen Datenformaten den Vorteil bietet, dass es sich aufgrund korrespondierender Datentypen, wie beispielsweise dem JSON Object, das einem Python Dictionary entspricht, leichter verarbeiten lässt, erhielt das Format eine höhere Bewertung.

#### Eignung für die Struktur und Komplexität der auszutauschenden Daten

Die im Rahmen des Analytics-On-Demand-Self-Service auszutauschenden Daten sind einfach und klar strukturiert, weshalb die Formate JSON und CSV ausreichend sind. Komplexe Datenstrukturen lassen sich besser in XML abbilden.

#### Les- und Schreibbarkeit

Am einfachsten für den Menschen les- und schreibbar, ist das CSV-Format, das aus Zeilen und Spalten aufgebaut ist. Es folgt JSON, dessen Syntax im Vergleich zu XML kompakter ist. JSON verwendet geschweifte Klammern für Objekte und eckige Klammern für Arrays, wohingegen XML öffnende und schließende Tags verwendet und die Daten hierarchisch strukturiert speichert.

Nach Auswertung der Entscheidungsmatrix, bei der die vergebenen Bewertungspunkte aufsummiert wurden, wurde sich für die standardisierte Aufbereitung mittels JSON, das mit 14 Bewertungspunkten die höchste Punktzahl erhalten hatte, entschieden.

### **Preprocessor-RPA**

Ziel des Preprocessor-RPAs ist es die Datei zum Datenaustausch zu erstellen und den Start für die Analytics-Aufgabe zuständigen Analytics-RPA zu initiieren. Um nur mit der Vorverarbeitung zu starten, wenn ein neuer Request erstellt wurde, muss zunächst geprüft werden, ob ein Trigger vorhanden ist. Wenn ein Trigger existiert, müssen zunächst alle neuen Requests gelesen werden (A03). Um das Risiko einer Mehrfachverarbeitung durch verschiedene Instanzen zu minimieren, muss der Status eines jeden Requests aktualisiert werden. Die Vermeidung einer Mehrfachverarbeitung, ist dadurch nicht möglich, da bei einem nahezu simultanen Start verschiedener Instanzen des Preprocessor-RPAs ein Request bereits mehrfach erfasst sein kann. Anschließend muss für jeden Request die Datei zum Datenaustausch erzeugt werden. Dazu müssen zunächst die Eingabedaten des Users validiert werden. Bei nicht erfolgreicher Validierung ist es nach A05 wünschenswert, dass die Eingabe beim User reklamiert wird. Jedoch kann die Konzeption und Implementierung eines Reklamationsprozess im Rahmen des Projekts nicht realisiert werden. Stattdessen wird bei ungültigen Eingaben der Request als abgebrochen gekennzeichnet und eine Nachricht an den User gesendet, in der auf den Abbruch und seinen Eingabefehler hingewiesen wird. Für die Verarbeitung des Requests durch den entsprechenden Analytics-RPA muss ein Trigger zum Start des Analytics-RPA gesetzt werden. Abschließend muss der Status in der Steuerungs-Sharepoint-Liste aktualisiert werden.

### **Ergebnisbereitstellung**

Um das Ergebnis der Analytics-Aufgabe dem User bereitzustellen (A11), ist ein Ablageort erforderlich, auf den sowohl der Analytics-RPA und der User zugreifen können. Ein online verfügbarer Ablageort, auf den über das RPA-Framework gut zugegriffen werden kann, ist SharePoint. Deshalb soll das Ergebnis eines Requests einem User in einem persönlichen SharePoint-Ordner bereitgestellt werden. Für die Bezeichnung des Ordners soll der Name des Users verwendet werden. Zur Erleichterung der Suche und des Zugriffs sollen alle Ordner in derselben Bibliothek einer SharePoint-Teamwebseite angelegt werden. Damit ein User nur auf das Ergebnis der selbst erstellten Requests zugreifen kann, ist die Vergabe von Berechtigungen erforderlich.

Um Berechtigungen auf einen SharePoint-Ordner zu erteilen, ist der Zugriff auf die „SharePoint CSOM API“ über einen HTTP-POST-Request notwendig. Über die Spezifikation des Ortes des Ordners, den es zu berechtigen gilt, und der Methode „UpdateDocumentSharingInfo“ in der URI kann für einen User über die User-ID, die der E-Mail-Adresse entspricht, die Benutzerrolle geändert werden.

Aufgrund von Restriktionen hinsichtlich Webseitenzugriff mit verbundenem Office-Konto aus einem RPA heraus, muss für die automatisierte Freischaltung eines Users auf einen Ordner auf einen Power Automate Workflow zurückgegriffen werden. In Power Automate ist es über die Hinterle-

gung einer „Verbindung“ zu SharePoint möglich, dass über einen Konnektor direkt mit der SharePoint-API kommuniziert werden kann. Damit der Workflow auch von anderen RPAs genutzt und auf beliebige SharePoint-Ordner angewendet werden kann, wurde bei der Konzeptionierung des Workflows auf Portabilität geachtet. Im Zuge dessen soll eine Sharepoint-Liste zur Steuerung des Berechtigungsprozesses verwendet werden, die es ermöglicht, den Power Automate Workflow über dynamische Werte zu steuern. Der Flow muss folglich über einen Eintrag eines RPAs in der Berechtigungs-SharePoint-Liste ausgelöst werden.

### **Analytics-RPA**

Ziel eines Analytics-RPAs ist es die Analytics-Aufgabe auszuführen und dem User das Ergebnis bereitzustellen. Infolgedessen, dass sich verschiedene Analytics-RPAs nur anhand der spezifischen Analytics-Aufgabe voneinander unterscheiden, wurde ein allgemeines Konzept zum Ablauf eines Analytics-RPAs ausgearbeitet. Zur Verarbeitung eines Requests muss zunächst die für den jeweiligen Analytics-RPA vorgesehene Datei zum Datenaustausch eingelesen werden (A09). Damit keine Mehrfachverarbeitung durch verschiedene Instanzen des Analytics-RPAs möglich ist, muss über einen Sperrmechanismus der Zugriff auf die Dateien synchronisiert werden. Anschließend muss die Analytics-RPA spezifische Funktionalität ausgeführt werden. (A10) Bei erfolgreicher Verarbeitung wird das Ergebnis des Requests dem User bereitgestellt und der Request als erfüllt gekennzeichnet. Bei Auftritt eines Fehlers wird der User informiert und der Request als abgebrochen gekennzeichnet.

### **Sperrmechanismus**

Ein etabliertes, einfaches Verfahren zur Vermeidung von Mehrbenutzeranomalien bei parallelem Zugriff auf eine Ressource ist das Setzen einer Sperre zur Sicherung des exklusiven Zugriffs eines Prozesses auf eine Ressource. [25] Damit nicht verschiedene Instanzen eines Analytics-RPAs auf dieselben Dateien zum Datenaustausch zugreifen und diese mehrfach verarbeiten, wird der Dateizugriff mit einem Sperrmechanismus synchronisiert, der einen exklusiven Zugriff auf die Datei ermöglicht. Existiert noch keine Sperre, wird eine Sperre in Form einer leeren Text-Datei gesetzt. Die Sperre bleibt solange bestehen, bis ein Analytics-RPA die Request-Daten eingelesen hat. Vor der Freigabe der Sperre, wird die Datei zum Datenaustausch verschoben.

### **Analytics-RPAs: Use Case 1 und 2**

Vor Automatisierung eines Prozesses ist es zur erfolgreichen Nutzung von RPA im Rahmen eines aktiven Prozessmanagements erforderlich zu prüfen, ob der zu automatisierende Prozess optimiert werden kann. [5] [12]

Bei Analyse des bestehenden Prozesses zur Beschaffung von Einkaufsinfosätzen mit Konditionen wurde als Potenzial zur Optimierung des automatisierten Prozessablaufs die Reduzierung von Schnittstellen identifiziert. Schnittstellen können sich zu einem Problemfaktor bei der Prozessautomatisierung entwickeln, da mit zunehmender Anzahl an Schnittstellen die Komplexität des zu automatisierenden Prozesses steigt. [26]

Im Prozess zur Beschaffung von Einkaufsinfosätzen mit Konditionen kann die Anzahl der erforderlichen Schnittstellen um zwei reduziert werden. Zum einen kann der Download und Import der Daten für die Weiterverarbeitung durch einen direkten Zugriff auf das Data Lakehouse, das als Datenquelle des BI-Berichts dient, eingespart werden. Zum anderen kann anstelle der Imitation der menschlichen Arbeitsschritte bei Bedienung des Single-Liner-Tools, die Funktionalität des Single-Liner-Tools unmittelbar über den Analytics-RPA realisiert werden.

Im Prozess zur Abfrage des Orderbuchs soll die Abwicklung über die SAP Standardtransaktion „SE16“ aufgrund bestehender Framework-Strukturen und dem nicht Vorhandensein der SAP-Datenbanktabelle „EORD“ im Data Lakehouse beibehalten werden.

Die Ergebnisse der Analytics-Aufgaben sollen jeweils als Excel-Datei mit Hinweis auf die Eingabedaten dem User bereitgestellt werden. Zur eindeutigen Identifizierung der Ergebnisse soll die Datei die Request-ID als Namen erhalten. Die Excel-Datei, die das Ergebnis der Einkaufsinfosatzabfrage mit Konditionen beinhaltet, soll entsprechend dem für die SAP-Automatisierungsschnittstelle benötigten Formats aufgebaut werden.

Gemäß Ausnahmebehandlung soll ein Laufzeitfehler, der während der Programmausführung auftritt abgefangen werden und an die höhere Programmebene zur Behandlung weitergereicht werden. [24]

### Berechtigungskonzept

Gemäß einem zur Gewährleistung von Informationssicherheit zu integrierendem Berechtigungsmanagement ist auf eingeschränktem Zugang und Zugriff der Mitarbeitenden auf Informationen und Dienste zu achten. [27] Über ein Berechtigungskonzept soll die Benutzung des Analytics-On-Demand-Self-Services auf einen eingeschränkten Personenkreis restriziert werden (A17). Zur frühzeitigen Unterbindung eines Requests einer nicht zur Ausführung der Analytics-Aufgabe berechtigten Person soll die Berechtigungsprüfung Teil der GUI werden. Das Berechtigungskonzept ist somit nicht Teil der im Rahmen des Projekts zu implementierenden Komponenten.

## IMPLEMENTIERUNG

Im Zuge der Implementierung wurde das erarbeitete Konzept für die einzelnen Komponenten des Analytics-On-Demand-Services in ausführbaren Programmcode umgesetzt.

Die Implementierung umfasst die Implementierung einer Hub-Klasse, des RPA Preprocessors, des RPA Analytics, der das Orderbuch abfragt, des RPA Analytics, der die Einkaufsinfosätze mit Konditionen abfragt und zwei Power Automate Workflows zum Erzeugen eines Triggers und zur Vergabe von SharePoint-Ordner-Berechtigungen.

### Hub-Klasse

In der Fachliteratur wird zur Reduzierung der Komplexität der RPA-Implementierung der modulare Aufbau von RPA-Software empfohlen, sodass durch Wiederverwendung eine schnellere Entwicklung erreicht werden kann. [28] Die für

die Abwicklung eines Request erforderliche, mehrfach benötigte Funktionalität wurde deshalb modular in Funktionen implementiert und zentral über eine sog. Hub-Klasse bereitgestellt.

## Zentrale Datenhaltung

Auf SharePoint wurde eine SharePoint-Liste, in der in Abbildung 5 dargestellten Form, angelegt.

Name	Typ	Default	Inhalt
ID	Zahl		ID des Requests
Created	Datum und Uhrzeit		Zeitpunkt der Request-Erfassung
TYP	Einzelne Textzeile		Analytics-Aufgabe
VAR	Einzelne Textzeile		Variante der Analytics-Aufgabe
OPTION1	Mehrere Textzeilen		Eingabe des Request-Erstellers
OPTION2	Mehrere Textzeilen		Eingabe des Request-Erstellers
OPTION3	Mehrere Textzeilen		Eingabe des Request-Erstellers
OPTION4	Mehrere Textzeilen		Eingabe des Request-Erstellers
STATUS	Zahl	0	Status des Requests
Requester	Einzelne Textzeile		E-Mail-Adresse des Users
Email_Response	Ja/Nein		Wunsch des Users nach E-Mail-Benachrichtigung

Abbildung 5: Spalten der SharePoint-Liste zur Datenhaltung

## Trigger

Für die Erzeugung des Triggers für den Preprocessor-RPA wurde in Power Automate ein automatisierter Cloud-Flow angelegt. Der Flow startet automatisch, wenn ein neues Element in der SharePoint-Liste, die die Request-Daten enthält, erstellt wird. Aufgabe des Flows ist es, eine Text-Datei mit dem Namen „trigger.txt“, die als Trigger fungiert, in einem speziellen OneDrive-Verzeichnis zu erzeugen.

Um den Trigger für den Analytics-RPA zu erzeugen, werden in einer dafür implementierten Methode, die bestehenden Strukturen des RPA-Frameworks genutzt. Unter Angabe des Namens des Analytics-RPAs, dessen Ausführung getriggert werden soll, wird ein den zur Steuerung eines RPAs entsprechender Trigger erzeugt.

## RPA-Steuerungsliste

In Excel wurde eine RPA-Steuerungsliste, in der in Abbildung 6 dargestellten Form angelegt. In der RPA-Steuerungsliste wurden für die Orderbuchabfrage drei Varianten erfasst, sodass die für einen Einkaufenden relevanten Möglichkeiten einer Abfrage realisiert werden können. Für die Infosatzabfrage mit Konditionen wurden zwei Varianten erfasst, sodass eine Abfrage entsprechend der Filterbedingungen des zu Grunde liegenden Ist-Prozesses möglich ist.

	A	B	C	D
1	TASK	TASK_VARIANTE	TASK_PARAMETER	RPA
2	Orderbuch	Orderbuch1	LIFNR;WERKS;FIX	rpa_analytics_
3	Orderbuch	Orderbuch2	LIFNR;MATNR	rpa_analytics_
4	Orderbuch	Orderbuch3	LIFNR;MATNR;WERKS	rpa_analytics_
5	Infosatz Konditionen	InfosatzKonditionen1	LIFNR;EKORG	rpa_analytics_
6	Infosatz Konditionen	InfosatzKonditionen2	LIFNR;EKORG;MATNR	rpa_analytics_

Abbildung 6: RPA-Steuerungsliste



## Nachrichtendienst

Sowohl für „Fulfilled-Message“ als auch für „Cancelled-Message“ wurde jeweils eine HTML-Datei als Template erstellt. Die Templates sind nach dem von dem W3C (World Wide Web Consortium) für HTML5 vorgegebenen Standards aufgebaut.

In Abbildung 7 und 8 ist jeweils ein Beispiel einer über Microsoft Teams versendeten „Fulfillment“ und „Cancellation“-Nachricht abgebildet.

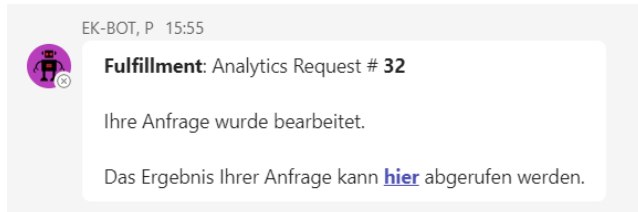


Abbildung 7: „Fulfillment“-Nachricht

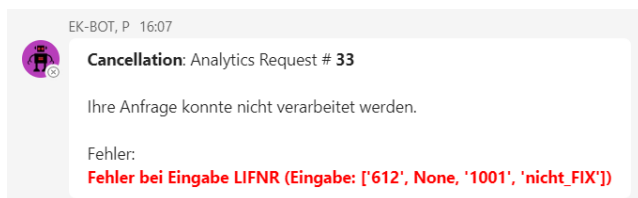


Abbildung 8: „Cancellation“-Nachricht

## Validierungsprüfungen

Jede Validierungsprüfung findet in einer eigenen Methode statt. Zur Kennzeichnung, dass es sich um eine Methode zur Validierungsprüfung handelt, wurde eine einheitliche Benennung gewählt. Alle Methodennamen beginnen mit „check“ und werden durch einen Unterstrich von dem Namen des zu überprüfenden Eingabeparameters getrennt, zum Beispiel „check\_LIFNR“. Die Benutzereingabe des Users, die es zu validieren gilt, wird der Methode als Übergabeparameter übergeben. Für jede Validierung mit Ausnahme der Validierung des FIX-Kennzeichens wird zunächst geprüft, ob es sich bei der Benutzereingabe um kein leeres Objekt oder keinen Leerstring handelt. Anschließend werden die Eingabedaten des Users bereinigt. Bei der Bereinigung werden Leerzeilen, Leerzeichen, Punkte, Kommas und nicht ASCII-Zeichen durch einen Leerstring ersetzt. Dann findet für jeden in der Benutzereingabe enthaltenen Wert eine eingabeparameterabhängige Validierungsprüfung statt, ehe bei erfolgreicher Validierung die Benutzereingabe und ansonsten eine leere Variable zurückgegeben wird.

Zur Überprüfung, ob es sich bei dem zu überprüfenden Wert um eine Lieferantenummer handelt, wird zunächst geprüft, ob es sich um einen numerischen Wert handelt. Zur Überprüfung auf die Nummernkreise wird die Lieferantenummer auf zehn Stellen mit 0en aufgefüllt. Von rechts ausgehend wird anschließend geprüft, ob die Lieferantenummer mit „00006“ oder „065“ beginnt.

Zur Überprüfung, ob es sich bei dem zu überprüfenden Wert um ein Werk handelt, wird geprüft, ob die Länge des Strings vier ist. Ebenso findet die Überprüfung auf eine Einkaufsorganisation statt. Für die Validierung einer Materialnummer wird geprüft, ob die Länge des Strings zehn ist. Bei der Validierung des FIX-Kennzeichens findet je nach Eingabe zur späteren Überprüfung eine Anpassung der Eingabe auf „X“ (nur fixierte Einträge) oder einen Leerstring (alle Einträge) statt.

Anhand der aus der RPA-Steuerungsliste ermittelten Eingabeparametern findet eine Validierung der in den SharePoint-Eingabespalten vorhandenen Eingabedaten des Users statt. Dazu wird für jeden Eingabeparameter die eingabeparameterabhängige Funktion zur Validierung aufgerufen, die prüft, ob es sich um eine dem Eingabeparameter entsprechende Eingabe handelt.

Die Methoden zur Validierungsprüfungen wurden als verschachtelte Funktionen implementiert, sodass die dem Eingabeparameter entsprechende, aufzurufende Methode nicht erst über eine Kontrollstruktur identifiziert werden muss, sondern direkt über die locals()-Funktion von Python aufgerufen werden kann.

War die Validierung erfolgreich wird der Eingabeparameter als Key und die Benutzereingabe als Value in einem Dictionary als neues Key-Value-Paar hinzugefügt. Entsprechend keine der in den Eingabespalten enthaltenen Benutzereingaben den Vorgaben des Eingabeparameters zur Gültigkeit, wird zur Benachrichtigung eines Users in einem String spezifiziert, bei welchem Eingabeparameter die Benutzereingaben nicht erfolgreich validiert werden konnten.

Sollte keine Methode zur Validierungsprüfung für den Eingabeparameter vorhanden sein, wenn beispielsweise aufgrund der Art des Eingabeparameters keine benötigt wird, wird mit der Benutzereingabe analog einer erfolgreichen Validierung verfahren.

## Preprocessor-RPA

Zu Beginn der Ausführung des Preprocessor-RPAs wird geprüft, ob ein Trigger vorhanden ist. Wenn nein, wird die Ausführung des Preprocessor-RPAs beendet. Wenn ja, wird der vorhandene Trigger gelöscht und die Methode „preprocessing\_routine“, die die Routine zur Vorverarbeitung eines Requests enthält aufgerufen.

Um die neu erfassten Requests zu erhalten, startet die Vorverarbeitungs-Routine mit dem Einlesen der Einträge der Sharepoint-Liste mit dem Status 0 in ein Dataframe. Wenn es neue Requests gibt, wird zur Kennzeichnung, dass die Requests bearbeitet werden der Status eines jeden Requests auf 1 aktualisiert.

Anschließend wird die Methode zur Erstellung der Datei zum Datenaustausch aufgerufen. Innerhalb der Methode wird für jeden neuen Request eine Methode zum Lesen des Namens des für die Ausführung der Analytics-Aufgabe erforderlichen RPAs und die dafür notwendigen Eingabeparameter aus der RPA-Steuerungsliste aufgerufen.

Anschließend wird eine weitere Methode zur Erstellung eines Dictionarys mit den grundsätzlichen Request-Daten aufgerufen. Nach der Validierung der Eingabedaten in einer speziell dafür implementierten Methode werden das Dictionary mit den grundsätzlichen Request-Daten und das Dictionary mit den validierten Eingabedaten zu einem Dictionary vereinigt.

Das Dictionary wird durch eine Methode zur Speicherung der Daten im JSON-Format serialisiert als JSON-File (sh. Abbildung 9) mit der Request-ID als Dateinamen in einem identisch benannten Ordner gespeichert.

```

1 {
2   "r_ID": 32,
3   "r_Reqester": "Nadine.Riedl@krones.com",
4   "r_Email": true,
5   "r_Var": "InfosatzKonditionen2",
6   "LIFNR": [
7     " "
8   ],
9   "EKORG": [
10    " "
11   ],
12  "MATNR": [
13    " ",
14    " "
15  ]
16 }

```

Abbildung 9: JSON-File

Wenn kein Fehler bei der Speicherung aufgetreten ist, wird in einer weiteren Methode der Trigger zum Start eines Analytics-RPA gesetzt. Außerdem wird der Status des Requests auf 2 aktualisiert.

Ist während der Validierung oder an einer anderen Stelle ein Fehler aufgetreten, wird der Fehler abgefangen, die Request-Verarbeitung abgebrochen, eine „Cancelled“-Message an den User gesendet und der Status des Requests auf 4 aktualisiert. Je nach Art des Fehlers enthält die Nachricht entweder einen Hinweis auf einen Fehler bei der Validierung oder weist allgemein darauf hin, dass es einen Fehler bei der JSON-File-Erstellung gab.

### Ergebnisbereitstellung

Auf SharePoint wurde die SharePoint-Liste zur Steuerung des Berechtigungsprozesses, in der in Abbildung 10 dargestellten Form angelegt.

Name	Typ	Inhalt
ID	Zahl	ID des Eintrags
Email	Einzelne Textzeile	E-Mail-Adresse der zu berechtigenden Person(en)
Sharepoint	Einzelne Textzeile	SharePoint-Webseite, auf der sich der Ordner befindet, auf den berechtigt werden soll
Library	Einzelne Textzeile	Bibliothek, in der sich der Ordner befindet, auf den berechtigt werden soll
Ordner	Einzelne Textzeile	Ordner, auf den berechtigt werden soll

Abbildung 10: Spalten der Berechtigungs-SharePoint-Liste

Um einen User auf einen SharePoint-Ordner zu berechtigen, wurde in Power Automate ein automatisierte Cloud-Flow angelegt. Der Flow startet automatisch, wenn ein neues Element in der Berechtigungs-SharePoint-Liste erstellt wird.

Aufgabe des Flows ist es eine HTTP-Anforderung an SharePoint zu senden. Dazu werden zunächst die für den HTTP-Request erforderlichen Daten aus der SharePoint-Liste in Variablen gespeichert. Als nächstes werden die Ordnermetadaten des zu berechtigenden Ordners abgerufen. Anschließend wird eine POST-Anforderung an SharePoint gesendet, die die E-Mail-Adresse der zu berechtigenden Person und die Berechtigung zur Anzeige beinhaltet.

Zur Bereitstellung des Ergebnisses auf SharePoint wird durch den Analytics-RPA eine Methode aufgerufen, die zunächst prüft, ob bereits ein persönlicher Ordner zur Ergebnisablage existiert. Falls nicht, wird ein Ordner mit dem Namen des Users als Ordner-Bezeichnung angelegt. Die Vergabe der Berechtigung auf den Ordner erfolgt über eine eigene Methode, die einen SharePoint-Listeneintrag in der Berechtigungs-SharePoint-Liste erzeugt. Anschließend wird die Ergebnisdatei, die in OneDrive in einem „Ergebnis“-Ordner gespeichert wurde, auf SharePoint in den persönlichen Ordner des Users hochgeladen und in OneDrive in einen „Veröffentlicht“-Ordner verschoben.

### Analytics-RPA

Zu Beginn der Ausführung des Preprocessor-RPAs wird die Methode „analytics\_routine“, die die Routine zur Verarbeitung eines Requests enthält aufgerufen.

Damit Änderungen an Files und Templates, die von verschiedenen Analytics-RPAs verwendet werden, nur einmalig und zentralisiert erfolgen müssen, sollen diese in den Verzeichnissen des Preprocessor-RPAs durchgeführt werden. Um bei Ausführung eines RPAs immer über den aktuellen Stand der Dateien zu verfügen, werden die im Preprocessor-RPA hinterlegten Dateien zu Beginn der Verarbeitungsroutine kopiert und in die entsprechenden Verzeichnisse des Analytics-RPAs eingefügt.

Anschließend ist der Analytics-RPA solange aktiv, bis alle für den Analytics-RPA vorgesehenen Requests verarbeitet wurden. Dazu wird zunächst durch das Analytics-RPA-spezifische Verzeichnis in OneDrive iteriert. Ist ein Request vorhanden wird eine Sperre für den exklusiven Zugriff auf den Request angefordert. Konnte die Sperre nicht gesetzt werden, weil bereits eine andere Instanz des Analytics-RPAs den exklusiven Zugriff besitzt, wird daraufhin eine Sperre für den nächsten vorhandenen Request angefordert. Konnte die Sperre erfolgreich gesetzt werden, werden die in der JSON-Datei enthaltenen Daten geladen. Anschließend wird die eingelesene Datei in einen „Eingelesen“-Ordner in OneDrive verschoben, bevor der exklusive Zugriff durch das Löschen der als Sperre fungierenden Datei freigegeben wird.

Als Voraussetzung für die Aktualisierung des Request-Status in SharePoint werden basierend auf der in den Daten vorhandenen Request-ID die in SharePoint zum Request vorhandenen Request-Daten in ein Dataframe eingelesen. Daraufhin wird die Methode, die die Analytics-RPA spezifische Funktionalität enthält, ausgeführt. Im Erfolgsfall wird das

Ergebnis des Requests dem User auf SharePoint bereitgestellt, der Status des Requests wird auf 3 aktualisiert und dem User wird eine Fulfillment-Message gesendet. Im Fehlerfall wird eine Cancelled-Message an den User gesendet, deren Nachrichtentext darauf hinweist, dass ein Fehler im ausführenden Analytics-RPA zu dem Abbruch des Requests geführt hat. Außerdem wird der Status des Requests auf 4 aktualisiert.

### **Analytics-RPA - Use Case 1: Orderbuch**

Die für die Orderbuchabfrage spezifische Funktionalität befindet sich in einer eigenen Methode, der das Dictionary mit den aus der JSON-Datei geladenen Daten als Übergabeparameter übergeben wird. Um entsprechend der Benutzereingaben die Tabelle „EORD“ über die SAP-Transaktion „SE16“ abfragen zu können, werden die Benutzereingaben ohne Information über das FIX-Kennzeichen durch Iteration und Anhängen an eine Liste zunächst von den grundsätzlichen Daten getrennt. Für die SE16-Abfrage werden anschließend verschiedene Funktionen des RPA-Frameworks verwendet. Nach dem Verbindungsaufbau zu SAP und dem Aufruf der Eingabemaske der Tabelle „EORD“ über die Transaktion „SE16“, wird die Eingabe der Filterkriterien vorbereitet, in dem geprüft wird, ob alle für die Eingabe der Filterkriterien erforderlichen Felder auf dem Selektionsbildschirm vorhanden sind. Falls nicht, werden diese hinzugefügt. Daraufhin werden die Felder des Selektionsbildschirms entsprechend der Benutzereingaben gefüllt. Anschließend findet nach Ausführung der Tabellenabfrage der Download des Abfrageergebnisses als CSV-Datei statt. Bei erfolgreichem Download wird die CSV-Datei in ein Python-Dataframe eingelesen. Im Fehlerfall wird ein leeres Python-Dataframe erzeugt. Aus dem Dataframe wird für einen Requesterstellenden als Ergebnis seiner Anfrage eine Excel-Datei mit der Request-ID als Namen im „Ergebnis“-Ordner in OneDrive erstellt. Zuvor wurden die Filterkriterien über ein weiteres Tabellenblatt eingefügt. Während der Methodenausführung auftretende Fehler werden abgefangen und in der aufrufenden Methode „analytics\_routine“ behandelt.

### **Analytics-RPA - Use Case 2: Einkaufsinfosatz mit Konditionen**

Die für die Einkaufsinfosatzabfrage mit Konditionen spezifische Funktionalität befindet sich in einer eigenen Methode, der das Dictionary mit den aus der JSON-Datei geladenen Daten als Übergabeparameter übergeben wird. Um entsprechend der Benutzereingaben die sich im Data Lakehouse befindenden Tabellen abfragen zu können, werden die Benutzereingaben durch Iteration und Anhängen an eine Liste zunächst von den grundsätzlichen Daten getrennt. Im Anschluss wird auf Basis der Benutzereingaben eine String-Konkatenation erzeugt, die als SQL-Abfragestring bei der Extraktion der Daten aus dem Data-Lakehouse dient.

Nacheinander werden die Tabellen des Data Lakehouses, die die Daten zu den Einkaufsinfosätzen und den Konditionen enthalten, jeweils in ein Python-Dataframe eingelesen. Nach der Verbindung der einzelnen Dataframes über einen Left-Join, wird das Ergebnis in ein Excel-Template, das in einem Verzeichnis des RPAs abgelegt ist und dem für die SAP-Automatisierungsschnittstelle benötigten Formats entspricht, eingefügt. Die Filterkriterien werden in einem weite-

ren Tabellenblatt abgebildet. Anschließend wird die Excel-Datei mit der Request-ID als Namen im „Ergebnis-Ordner“ in OneDrive gespeichert. Während der Methodenausführung auftretende Fehler werden abgefangen und in der aufrufenden Methode „analytics\_routine“ behandelt.

## **ERGEBNISSE**

### **Technische Sicht**

Die durchgeführte Implementierung aller auf Basis der Anforderungsanalyse konzipierten Komponenten ermöglicht über die manuelle Erfassung eines neuen Requests in Form eines Listeneintrags in der SharePoint-Liste sowohl eine Simulation einer Orderbuchabfrage als auch eine Simulation einer Infosatzabfrage mit Konditionen, womit der Nachweis für die Funktionsfähigkeit der RPA-Schnittstelle erbracht wird.

Alle Anforderungen, die verpflichtend und als hoch relevant für den Erfolg des Projekts eingestuft wurden (Muss, Soll), konnten erfüllt werden. Die weniger relevante Anforderung der Reklamation fehlender Daten bei einem User (Kann) wurde aus Zeitgründen nicht erfüllt. Somit deckt der im Zuge des Projekts implementierte Funktionsumfang insgesamt 97 Prozent der ermittelten Anforderungen ab. Die Umsetzung eines Berechtigungskonzepts, das erst bei der Entwicklung der GUI relevant wird, wurde bei der Berechnung nicht berücksichtigt.

Die standardisierte Gestaltung der RPA-Schnittstelle sorgt ohne neuen bzw. in Ausnahmefällen mit einem im Vergleich zu einer Neuentwicklung eines RPAs, geringem Implementierungsaufwand für die Erweiterbarkeit des Analytics-On-Demand-Self-Services um weitere Analytics-Use-Cases.

Zusätzlich für Erweiterbarkeit und darüberhinausgehend für Wartbarkeit und Wiederverwendung sorgt die Modularisierung von Attributen und Methoden in einer für den Analytics-On-Demand-Self-Service-Lösung angelegten RPA-Klasse.

### **Prozessuale Sicht**

Im Zuge der Evaluierung der Projektergebnisse hat sich die Entwicklung eines auf RPA gestützten Prozesses in Form eines Analytics-On-Demand-Self-Services als zielführender Ansatz zur Reduzierung der Bearbeitungszeit einer Analytics-Aufgabe, sowie zur Reduzierung der Antwortzeit auf eine Anfrage erwiesen. Die im Zusammenhang mit der RPA-Schnittstelle erzielten Ergebnisse werden nachfolgend in Abbildung 11 zusammenfassend dargestellt und im Anschluss erörtert. Die quantitative Betrachtung des Projektergebnisses erfolgte unter der Annahme einer idealtypischen Verarbeitung eines Requests, die voraussetzt, dass der Request sofort von einem Analytics-RPA verarbeitet wird. In der Realität steigen Antwort- und Bearbeitungszeit eines Requests linear mit der Anzahl der zu verarbeitenden Requests. Eine realistische Betrachtung der Zeit- und Effizienzgewinne im Hinblick auf die Ausführung einer Analytics-Aufgabe ist erst nach Produktivsetzung des Analytics-On-Demand-Self-Services möglich. Als Kernergebnis festzuhalten sind die durch den neuen Prozess gewonnenen zeitlichen Kapazitäten bei den Bearbeitenden einer Analytics-Aufgabe, sowie eine verkürzte Wartezeit der Anfragestellten angesichts der

Abwicklung einer Analytics-Aufgabe über den Analytics-On-Demand-Self-Service anstelle des Ticketsystems.

	Allgemein	Orderbuch	Infosatz m. Konditionen
	Antwortzeit (AWZ)		
Urspr. AWZ	Ø 2 Tage ± 2880 min		
Best-Case (BC)	6 min	8 min	5,5 min
Average-Case (AC)	8,5 min	10,5 min	8 min
Worst-Case (WC)	11 min	13 min	10,5 min
Δ urspr. AWZ & AC	- 2871,5 min	- 2869,5 min	- 2 bis 22 min
Effizienzsteigerung	99,7 %	99,6 %	20 bis 73 %
Zeitliche Kapazität		+ Ø 15 min	+ Ø 10 bis 30 min
	Bearbeitungszeit (BBZ)		
Man. BBZ		Ø 15 min	Ø 10 bis 30 min
Autom. BBZ	Ø 1 min	Ø 3 min	Ø 0,5 min

Abbildung 11: Prozessuale Ergebnisse

### Antwortzeit

Nach Implementierung einer GUI und erfolgreicher Pilotierung des Analytics-On-Demand-Self-Services können Mitarbeitende des Einkaufs zukünftig bei Bedarf per Anlage eines Requests einen automatischen Verarbeitungsprozess initiieren, der ihnen das Ergebnis ihrer in Auftrag gegebenen Analytics-Aufgabe zeitnah zur Verfügung stellt. Im Best Case erfolgt nach Anlage des Requests, sofort der zeitgesteuerte Start des Preprocessor-RPAs und fünf Minuten später der zeitgesteuerte Start via Trigger des Analytics-RPA. Hinzu kommt die Bearbeitungszeit der Analytics-Aufgabe, die nach Erfahrungswerten je nach Umfang der Abfrage zwischen 0,5 und 1,5 Minuten liegt und somit durchschnittlich eine Minute beträgt. Im Worst Case erfolgt der zeitgesteuerte Start des Preprocessor-RPAs erst nach fünf Minuten, sodass im ungünstigen Fall die Antwortzeit auf den Request fünf Minuten höher ist als im besten Fall. Im Best Case beträgt die Antwortzeit sechs Minuten, im Worst Case 11 Minuten und im Average Case unter der Annahme einer Gleichverteilung 8,5 Minuten. Im Vergleich zur ursprünglichen Antwortzeit in Höhe von durchschnittlich zwei Tagen bedeutet die Average Case eine Reduzierung der Antwortzeit um 2871,5 Minuten und folglich eine Effizienzsteigerung des Prozesses um 99,7 Prozent. Die verringerte Wartezeit verbessert die Handlungsfähigkeit der Anfragestellten bezüglich dem der Anfrage zu Grunde liegenden Zweck und steigert ihre Produktivität.

### Bearbeitungszeit

#### Analytics-Use-Case: Orderbuch

Der Prozess zur Orderbuchabfrage wurde analog dem manuellen Prozess zur Orderbuchabfrage als RPA-Lösung implementiert. In einer isolierten Betrachtung des Abfrageprozesses zum Orderbuch bewirkt die Automatisierung eine Reduzierung der durchschnittlichen Bearbeitungszeit um 12 Minuten und impliziert eine durch Automatisierung erzielte Effizienzsteigerung um 80 Prozent. Bei einer in den Analytics-On-Demand-Self-Service integrierten Betrachtung entfällt der Bearbeitungsweg über Dritte und die Länge der Prozessausführung beträgt im Best Case acht Minuten, im Worst Case 13 Minuten und im Average Case 10,5 Minuten. Dadurch, dass die Analytics-Aufgabe nicht mehr von CP Processes and Systems bearbeitet werden muss, können sich die Mitarbeitenden strategisch wertvollere Aufgaben widmen. Über das Schaffen von zeitlichen Kapazitäten bei CP Processes and Systems in Höhe von 15 Minuten pro

Analytics-Aufgabe hinausgehend profitieren die Anfragestellten hier von den quantitativen Vorteilen einer Reduzierung der Antwortzeit um 2869,5 Minuten, die eine Effizienzsteigerung des Prozesses um 99,6 Prozent inkludiert, und den identischen qualitativen Vorteilen, die der Analytics-On-Demand-Self-Service in Bezug auf die Antwortzeit mit sich bringt.

#### Analytics-Use-Case: Einkaufsinfosatz mit Konditionen

Der Prozess zur Abfrage der Infosätze mit Konditionen wurde nicht nur automatisiert, sondern auch optimiert. Die Optimierung erfolgte durch einen direkten Zugriff auf die im Data Lakehouse gespeicherten relevanten Daten, wodurch die Verarbeitungsschritte für Download und Import der Daten für die Weiterverarbeitung entfielen. Die Automatisierung führt zu einer durchschnittlichen automatisierten Bearbeitungszeit in Höhe von 30 Sekunden. Für einen Mitarbeitenden des Einkaufs, der den Analytics-On-Demand-Self-Service in Anspruch nehmen muss, um eine automatisierte Einkaufsinfosatzabfrage mit Konditionen auszuführen, bedeutet das eine Ergebniszustellung im Best Case nach 5,5 Minuten, im Worst Case nach 10,5 Minuten und im Average Case nach acht Minuten. Im Vergleich zur ursprünglichen durchschnittlichen Bearbeitungszeit wurde für technisch weniger versierte Mitarbeitende die Dauer bis zum Erhalt des Ergebnisses von 30 auf acht Minuten und für technisch versierte Mitarbeitenden von 10 Minuten auf acht Minuten reduziert, was einer durch Automatisierung erzielten Effizienzsteigerung zwischen 20 und 73 Prozent entspricht. Abgesehen von der Reduktion der Dauer von Start der Analytics-Aufgabe bis zum Zeitpunkt, an dem ein bedarfsgerechtes Ergebnis vorliegt, können bei den Usern des Analytics-On-Demand-Self-Services je nach ihren Fähigkeiten zeitliche Kapazitäten zwischen zehn und 30 Minuten zur Bearbeitung qualitativ-höherwertiger und wertschöpfender Aufgaben geschaffen werden.

## FAZIT UND AUSBLICK

Entsprechend der des Artikels zu Grunde liegenden Zielsetzung wurde, eingebettet in ein existierendes Framework zur zeitgesteuerten Ausführung von Automatisierungsprozessen, eine standardisierte RPA-Schnittstelle für eine Analytics-On-Demand-Self-Service-Lösung konzeptioniert und implementiert.

Nach der Implementierung der für den Analytics-On-Demand-Self-Services benötigten GUI und erfolgreicher Pilotierung des Projekts kann in Zukunft jeder Mitarbeitende der Krones AG, der zur Ausführung der entsprechenden Analytics-Aufgabe berechtigt ist, gemäß der im Rahmen dieses Artikels ausgearbeiteten Use Cases eine durch einen Automatisierungsprozess gesteuerte Orderbuch- oder Einkaufsinfosatzabfrage durchführen. Als Resultat wird sich der Bearbeitungsaufwand einer Analytics-Aufgabe auf Null reduzieren, womit zeitliche Kapazitäten für die Bearbeitung qualitativ hochwertigere und strategisch wertvollere Aufgaben geschaffen werden, und die Wartezeit auf das Ergebnis einer Analytics-Aufgabe wird sich verringern. Darüber hinaus ermöglicht die standardisierte Gestaltung der RPA-Schnittstelle die Erweiterung des Analytics-On-Demand-Self-Services um weitere Use Cases.

Zukünftig gilt es auf dem Ergebnis des Projekts aufzubauen und zum einen den Analytics-On-Demand-Self-Service durch die Anbindung weitere Analytics-RPAs auszuweiten und zum anderen Analytics-unabhängige RPAs in das System zu integrieren, sodass deren usergesteuerte, automatische Abarbeitung ermöglicht wird. Außerdem ist es denkbar in einer weiteren Ausbaustufe des Analytics-On-Demand-Self-Services auf effizientere Methoden der Informatik zurückzugreifen, wie die Verwendung einer Datenbank anstelle von Excel- und SharePoint-Listen oder die Implementierung eines User Interfaces mit Hilfe eines auf Python basierenden Webframeworks, wie Flask, anstelle der bis dato üblichen Verfahrensweise mit einer Power-App.

## ZUSAMMENFASSUNG

Hohe Bearbeitungs- und Antwortzeiten von Tickets, die diverse Aufgaben aus dem Bereich Analytics inkludieren, führten CP Processes and Systems zur Konzeptionierung und Implementierung einer standardisierten RPA-Schnittstelle, um einen Analytics-On-Demand-Self-Service zur Bereitstellung von einkaufsbezogenen Daten gemäß dem Business Intelligence Ansatz zu realisieren. Dafür wurde zunächst ein Anforderungskatalog erarbeitet, der als Basis für die Konzeptionierung und Implementierung aller zur Prüfung der Funktionsfähigkeit der RPA-Schnittstelle erforderlichen Komponenten diente. Die durchgeführte Evaluation bescheinigt aus technischer und prozessualer Sicht den Erfolg des Projekts, wobei im Vordergrund die zukünftig nach Einführung des Analytics-On-Demand-Self-Service gewonnen zeitlichen Kapazitäten und verringerten Wartezeiten stehen.

## LITERATUR

- [1] Studie robotic process automation 2021, 2021.
- [2] Studie intelligent automation 2023, 2023.
- [3] Gvr report cover robotic process automation market size, share & trends analysis report by type (software, services), by deployment (cloud, on-premise), by organization (large, small & medium enterprises), by operations, by application, by region, and segment forecasts, 2024 - 2030, 2024.
- [4] Jan Krakau and Carsten Feldmann. *Robotic Process Automation (RPA) in der Logistik*. Springer Gabler, 1 edition, 2023.
- [5] Mario Smeets, Ralph Erhard, and Kaufler Thomas. *Robotic Process Automation (RPA) in der Finanzwirtschaft*. Springer Fachmedien, 2 edition, 2023.
- [6] Fachgruppe WI-BIA. Benennung der fachgruppe, o.J.
- [7] Markus Linden. *Geschäftsmodellbasierte Unternehmenssteuerung mit Business-Intelligence-Technologien*. Springer Gabler, 2016.
- [8] Peter Chamoni and Markus Linden. Business intelligence. 36(12):S. 1588 – 1592, 2007.
- [9] Christian Langmann and Daniel Turi. *Robotic Process Automation (RPA) – Digitalisierung und Automatisierung von Prozessen*. Springer Gabler, 2 edition, 2021.
- [10] Thomas Allweyer. Robotic process automation – neue perspektiven für die prozessautomatisierung, 2016.
- [11] Gartner. Robotic process automation software, o.J.
- [12] Praxishandbuch robotic process automation (rpa), 2022.
- [13] Wil van der Aalst, Martin Bichler, and Armin Heinzl. Robotic process automation, o.J.
- [14] SAP. Einkauf (mm-pur), 2023.
- [15] Jochen Benz and Markus Höflinger. *Logistikprozesse mit SAP*. Vieweg+Teubner, 3 edition, 2011.
- [16] SAP. Einkauf (mm-pur), 2023.
- [17] Manfred Broy and Marco Kuhrmann. *Einführung in die Softwaretechnik*. Springer Vieweg, 2021.
- [18] Klaus Pohl and Chris Rupp. *Basiswissen Requirements Engineering*. dpunkt.verlag, 2009.
- [19] Andrea Herrmann. *Grundlagen der Anforderungsanalyse*. Springer Vieweg, 2022.
- [20] Stephan Hoppe and Rolf Holle. Beitrag einer zentralen datenhaltung zur qualitätssicherung in kora. 8(2):158–164, 2000.
- [21] Daniel Sonnet, Gerhard Wanner, and Konrad Pfeilsticker. *Chancen einer nachhaltigen IT*. Springer Vieweg, 2023.
- [22] Microsoft. Sendegrenzwerte in outlook.com, o.J.
- [23] Handbuch digitalisierung in staat und verwaltung, 2020.
- [24] Johannes Ernesti and Peter Kaiser. Python 3, 2017.
- [25] Peter Kleinschmidt and Christian Rank. *Relationale Datenbanksysteme*. Springer-Verlag, 3 edition, 2005.
- [26] Matthias Meyer. Schnittstellenmanagement, 2016.
- [27] Bundesamt für Sicherheit und Informationstechnik. It-grundschutz, 2021.
- [28] Richard Murdoch. *Robotic Process Automation*. 2018.