

Hochschule Merseburg
Fachbereich Wirtschafts- und Informationswissenschaften
Studiengang Technische Betriebswirtschaft



Bachelorarbeit zum Thema:
**Potenziale der Open Source attended Robotic Process Automation
für kleine und mittlere Unternehmen in der Beschaffungslogistik:**
Anwendungsfall zum Einsatz von AutoHotkey

Vorgelegt von:

Viktor Neumann

XXXXXXXXXXXXXXXX XX

XXXXXX XXXXXXXXXXXX

E-Mail: vn.v.neumann@gmail.com

Matrikelnummer: xxxxxx

Zeitraum der Durchführung:
16.10.2023 - 18.12.2023

Betreuung: Dr. Alexandra Fiedler
Zweitgutachten: Prof. Dr. Jörg Döpke

Inhaltsverzeichnis

1	Einleitung	1
1.1	Herausforderungen und Chancen der Automatisierung	1
1.2	Zielsetzung und Abgrenzung	1
1.3	Verwandte Arbeiten	2
1.4	Gang der Untersuchung	2
2	Beschaffungslogistik	3
2.1	Elektronische Beschaffung	3
2.2	Kleine und mittlere Unternehmen	5
3	Robotic Process Automation	6
3.1	Mehrwert	7
3.2	Hürden	9
3.3	Potenzielle Prozesse für Robotic Process Automation	10
3.4	Open Source	14
3.4.1	AutoHotkey	15
3.4.1.1	Vorteile	16
3.4.1.2	Nachteile	17
3.4.2	Gewählter Implementierungsansatz	17
4	Anwendungsfälle	19
4.1	Lieferantenzugang und Informationserfassung	20
4.2	Vergleich vom Lieferdatum	23
4.3	Umwandlung einer Preisliste	25
4.4	Übergreifende Probleme und Lösungsansätze	29
5	Auswertungen und Ergebnisse	30
6	Zusammenfassung und Ausblick	32

1 Einleitung

1.1 Herausforderungen und Chancen der Automatisierung

Die Automatisierung von Geschäftsprozessen gewinnt in kleinen und mittleren Unternehmen (KMU) in der Beschaffungslogistik zunehmend an Bedeutung, um Effizienz und Wettbewerbsfähigkeit zu steigern (Statista, 2023). Hierbei erweist sich die Einbindung von Altsystemen und eigenständigen Anwendungen in die Prozessausführung als herausfordernd und benötigt die Einrichtung von Schnittstellen für den Informationsaustausch zwischen IT-Anwendungen (Erdmann & Sandkuhl, 2023, S. 84–85). Die hohen IT-Kosten, die mit der Schaffung von Schnittstellen oder der Neuprogrammierung von Anwendungen verbunden sind, machen eine Automatisierung jedoch oft nur für eine begrenzte Anzahl kritischer Geschäftsprozesse rentabel (Langmann & Turi, 2022, S. 1–11).

Die Beobachtung, wie Mitarbeiter während meines Praktikums diese Geschäftsprozesse aufgrund fehlender Automatisierung manuell ausführen mussten, hat mich inspiriert, über dieses Thema zu schreiben. Eine mögliche Lösung dieses Problems ist der Einsatz von Robotic Process Automation (RPA). Diese Technologie automatisiert repetitive und routinemäßige Aufgaben, die in der Regel von Mitarbeitern erledigt werden (Teunissen, 2019, S. 1).

1.2 Zielsetzung und Abgrenzung

Ziel der Arbeit ist es, die Potenziale der Open Source Attended Robotic Process Automation (RPA) für KMU in der Beschaffungslogistik zu untersuchen. Es soll ein Verfahren entwickelt werden, welches KMU ermöglicht, geeignete Prozesse für die Automatisierung mit RPA zu identifizieren. Ferner soll ein Vergleich zu anderen, in dieser Arbeit nicht gewählten RPA-Werkzeugen erleichtert werden.

Anhand von Anwendungsfällen soll die Machbarkeit von RPA mittels AutoHotkey (AHK) überprüft werden. Hierzu stellt die Skriptsprache AHK unter Windows eine vielversprechende Option dar, um Aufgaben mit variierender Komplexität zu lösen.

Anschließend soll mit drei unterschiedlichen Umsetzungsansätzen und den daraus gewonnenen Daten mittels einer Nutzwertanalyse untersucht werden, welcher der drei Ansätze sich konkret für KMU eignet.

Ein Fokus wird hierbei auf Attended RPA gelegt, da der Schwerpunkt in der Forschung größtenteils auf Unattended RPA gelegt wird. Eine nähere Erläuterung zu *Attended* und *Unattended* RPA findet sich in Kapitel 3.

1.3 Verwandte Arbeiten

Bislang sind leider nur wenige Literaturquellen zu Open Source RPA verfügbar. Shin et al. (2023) schreiben über ihre Open Source RPA mit dem Name *EnterRPA* womit die Benutzerfreundlichkeit im Vergleich zu bestehenden RPA-Plattformen verbessert werden soll, indem es ein integriertes Websystem nutzt. Mostaquim (2022) stellt Open Source *RPA Python* mit *UiPath*, einer kommerziellen RPA-Plattform und der Programmierung in Python gegenüber. Während Ramkumar, Satsish, Chetan, Patnana und Gulve (2021) auf die Evaluierung, Auswahl bzw. Verwendung von Open Source RPA *Robocorp* und auf die Planungs- und Sicherheitsmechanismen zum Schutz der Roboter eingehen.

Hüller et al. (2021) stellen *Ark Automate* vor, eine Open Source webbasierte Plattform für die Erstellung und den Betrieb von RPA-Robotern. Diese bietet austauschbare Modellierungsschnittstellen, um verschiedenen Benutzergruppen gerecht zu werden und soll Erweiterungen durch die Community fördern. Das Middleware-System mit der Bezeichnung *MIORPA* wird zur Steuerung von Open Source RPA-Robotern von Hwang et al. (2020) vorgestellt. Es enthält einen Algorithmus zur Auftragsplanung für die Zuweisung von Aufgaben an mehrere Roboter und bietet Echtzeit-Statusüberwachung.

Des Weiteren bietet das Buch von Taulli (2020) einen Einblick in die historische Entwicklung, die Geschäftsmodelle sowie die Vor- und Nachteile von Open Source Software, wobei der Schwerpunkt auf RPA liegt.

1.4 Gang der Untersuchung

Zu Beginn der Arbeit werden die grundlegenden Aspekte der Beschaffungslogistik sowie der elektronischen Beschaffung erläutert. Daraufhin sollen die Besonderheiten und Herausforderungen von KMU herausgearbeitet werden, welche von wesentlicher Bedeutung für die Gewichtung der Bewertungskriterien der Nutzwertanalyse des fünften Kapitels sind.

Im darauffolgenden Kapitel wird auf RPA eingegangen, wobei sowohl die grundlegenden Merkmale als auch der Mehrwert und die Herausforderungen dieser Technologie beleuchtet werden. Im Unterkapitel 3.3 soll anhand von Literaturrecherche herausgearbeitet werden, welche Prozesse durch den Einsatz von RPA automatisiert werden können.

Im nachfolgenden Unterkapitel 3.4 wird kurz auf die positiven und negativen Aspekte von Open Source Software eingegangen, ebenso wie auf die Definition von AHK und deren Vor- und Nachteile. Hiermit soll ein Vergleich zu alternativen RPA-Werkzeugen ermöglicht werden.

Abschließend werden die drei Ansätze für die Umsetzung präsentiert. In Kapitel 4 werden die konkreten Anwendungsfälle vorgestellt und im Anschluss mit den drei zuvor präsentierten Ansätzen umgesetzt. Dabei erfolgt eine detaillierte Ausarbeitung der jeweiligen Besonderheiten der Herangehensweisen pro individuellen Anwendungsfall. In Kapitel 5 erfolgt eine Entscheidungshilfe mittels Nutzwertanalyse, um zu bestimmen, welcher der drei Ansätze für KMU am sinnvollsten ist.

2 Beschaffungslogistik

Um das Potenzial von RPA bzw. AHK zu evaluieren, ist es notwendig zu wissen, was die Beschaffungslogistik kennzeichnet.

Nach Hausladen (2020, S. 96) umfasst die Beschaffungslogistik das Management aller logistischen Aktivitäten, die mit der Organisation, Durchführung und Überwachung des Materialtransports vom Lieferanten zum Unternehmen verbunden sind. Dazu gehört ebenfalls die Abwicklung der damit verbundenen Informationsflüsse (S. 96). Hierbei ist das vorrangige Ziel sicherzustellen, dass Materialien in der notwendigen Menge pünktlich, zielgenau und in hoher Qualität geliefert werden (S. 96).

Die Tätigkeiten der Beschaffung können in strategische und operative Bereiche aufgeteilt werden (Hausladen, 2020, S. 99–100). Hausladen nach umfasst der strategische Bereich beispielsweise:

- Ausarbeitung von Einkaufs- und Beschaffungsstrategien
- Durchführung von Wertanalysen
- Unterstützung bei Entscheidungen bezüglich Outsourcing oder Make-or-Buy
- Auswahl strategischer Lieferanten
- Vertragsverhandlungen
- Angelegenheiten des Global Sourcing
- Advanced Purchasing
- Management von Supplier Relationship (SRM)
- Maßnahmen zur Qualitätssicherung

Wohingegen der operative Bereich laut Hausladen folgende Aufgaben einschließt:

- Planung des Einkaufs
- Überprüfung der Verfügbarkeit
- Initiierung und Überwachung des Bestellprozesses; insbesondere im Hinblick auf die Einhaltung von Zeitplänen, Mengen und Qualität
- Überprüfung von Rechnungen
- Bearbeitung von Reklamationen bei Fehlern

Viele dieser Aufgaben sind standardisiert und wiederkehrend, was sie zu idealen Kandidaten für RPA macht. Unter Umständen sind jedoch nicht alle Aufgaben elektronisch umgesetzt oder häufig über mehrere Systeme verteilt, was den nächsten Unterpunkt besonders relevant macht.

2.1 Elektronische Beschaffung

Die grundlegende Konzeption des elektronischen Einkaufs (E-Procurement) besteht darin, die Interaktionen zwischen Unternehmen (Käufern) und Lieferanten (Verkäufern) sowie die damit verbundenen Einkaufsprozesse durch das Internet und den elektronischen Informationsaustausch mittels Computervernetzung zu optimieren (Kollmann, 2020, S. 160). Das heißt, ein Prozess der elektronischen Beschaffung wird durch den Einsatz

spezialisierte Software effizienter und sichtbarer gestaltet. Laut Turban et al. (2018, S. 140) ergeben sich Vorteile wie etwa:

- Einkäufer profitieren von zusätzlicher Zeit für wertschöpfende Aufgaben, einem reduzierten Arbeitsdruck und möglicherweise einer Senkung der Gemeinkosten für die gesamte Einkaufsabteilung.
- Aufbau effizienterer und kooperativer Partnerbeziehungen durch den Austausch von Informationen.
- Sicherstellung einer pünktlichen Lieferung und weniger Fehlbestände.
- Optimierung des Abgleichs von Rechnungen und der Streitbeilegung.
- Auffinden neuer Lieferanten, die Waren und Dienstleistungen schneller und/oder kostengünstiger bereitstellen können.
- Reduzierung von Fehlern, die während des Einkaufs- oder Versandprozesses durch menschliches Versagen auftreten können.

Konsequent dazu ergeben sich beispielhaft folgende Nachteile (Angeles & Nath, 2007, S. 112; Turban et al., 2018, S. 141).

- Die Gesamtkosten sowie alle anfallenden Kosten der Anschaffung können zu hoch sein.
- Das Risiko für Hackerangriffe steigt.
- Verschiedene Stakeholder leisten Widerstand gegen Veränderungen.
- Es gestaltet sich herausfordernd, Lieferanten zur digitalen Kooperation zu motivieren.
- Die Komplexität des Systems kann zu hoch sein.
- Unzureichende Kenntnisse im Bereich IT.
- Die Realisierung einer internen und externen Integration kann herausfordernd sein, insbesondere aufgrund von möglichen Unterschieden in den Standards.
- Technologische Entwicklungen können zu häufigen Änderungen führen.

Zusammengefasst lässt sich feststellen, dass die elektronische Beschaffung eine digitale Transformation der Beschaffungsprozesse darstellt, die erhebliche Vorteile für Unternehmen in Bezug auf Kostensenkung, Effizienzsteigerung und Lieferantenmanagement bieten kann (Turban et al., 2018, S. 140). Die Einführung von E-Procurement birgt jedoch auch Herausforderungen wie den Widerstand gegen Veränderungen, mangelnde IT-Kompetenz, Schwierigkeiten bei der Integration von Altsystemen und Unterschieden in den Standards (Angeles & Nath, 2007, S. 112; Turban et al., 2018, S. 141). Insgesamt kann E-Procurement zu einer Verbesserung der Lieferkette und der Unternehmensleistung beitragen, was in der heutigen digitalen Wirtschaft von zunehmender Bedeutung ist.

Diese Vor- und Nachteile gewinnen weiter an Bedeutung, wenn RPA in Kapitel 3 näher betrachtet wird.

2.2 Kleine und mittlere Unternehmen

Das IfM Bonn definiert KMU als solche, die einen Jahresumsatz von maximal 50 Millionen Euro und eine Mitarbeiteranzahl von weniger als 500 aufweisen (Institut für Mittelstandsforschung, 2016). Die Europäische Kommission hingegen grenzt Unternehmen ab, die eine Mitarbeiteranzahl von weniger als 250 Personen sowie einen Jahresumsatz von höchstens 50 Millionen Euro oder eine Jahresbilanzsumme von höchstens 43 Millionen Euro aufweisen (Amtsblatt der Europäischen Union, L124 vom 25. Mai 2003, 2003/2003, S. 39).

Im Jahr 2021 existierten weltweit etwa 327 Millionen KMU (Statista, 2022b), während es rund 351 Tausend Großunternehmen gab (Statista, 2022a). Trotz der unterschiedlichen Definitionen wird das Verhältnis zwischen KMU und Großunternehmen eindeutig. Mit einem Anteil von 99,89 % zeigt sich, dass der überwiegende Teil der Wirtschaft aus KMU besteht.

Umso wichtiger ist es, dass KMU den technologischen Anschluss in der globalen Wirtschaft nicht verlieren.

Besonderheiten und Herausforderungen

KMU zeichnen sich durch ihre Flexibilität und schnelle Anpassungsfähigkeit aus, doch zugleich sind sie aufgrund ihrer beschränkten absoluten Größe auch mit Nachteilen konfrontiert (Hausman, 2005, S. 777; Narula, 2004, S. 155).

Deshalb kommt es vor, dass KMU möglicherweise über gar keine eigene IT-Abteilung verfügen oder nur eine begrenzte Anzahl von IT-Experten beschäftigen (Teunissen, 2019, S. 4). Dies führt dazu, dass der Fokus der IT ausschließlich auf eine begrenzte Anzahl an Aufgaben gelegt wird.

Außerdem ist es eine Herausforderung, auf dem Arbeitsmarkt qualifiziertes Personal zu rekrutieren, gerade vor dem Hintergrund des Mangels im eigenen Unternehmen (Lee, Park, Yoon & Park, 2010, S. 296).

Die Unterstützung seitens der Geschäftsleitung spielt in KMU eine zentrale Bedeutung, da diese in der Regel Eigentümer des Unternehmens sind und unmittelbar in die Investitionsentscheidungen des Unternehmens eingebunden sind (Gupta & Barua, 2018, S. 1597; Rik Donckels & Erwin Fröhlich, 1991, S. 159).

Laut Sánchez-Rodríguez, Martínez-Lorente und Hemsworth (2020, S. 848) sind Führungskräfte in KMU oft aktiver in die täglichen Aktivitäten eingebunden als in größeren Organisationen. Darüber hinaus weisen kleinere Unternehmen generell eine schlankere Struktur des mittleren Managements auf, die einen direkten Berichtsweg von den Mitarbeitern zur obersten Führungsebene begünstigt (S. 848). Folglich wird die Entscheidungsfindung in KMU stärker von der obersten Führungsebene beeinflusst und gelenkt (S. 848). Dies steht im Gegensatz zur beobachtenden Dynamik in großen Unternehmen (S. 848).

Eine der Herausforderung für KMU besteht darin, der Konkurrenz voraus zu sein, wobei der Erwerb von neuer technologischer Ressourcen essenziell ist (Gupta & Barua, 2018, S. 1597). Weiterhin ist die Akquisition finanzieller Mittel erschwert, da Investitionen in Technologien

mit einer ungewissen Marktaussicht und technologischer Unreife das Risiko erhöht (Lee et al., 2010, S. 296). Aufgrund dessen ist es von besonderer Bedeutung, die richtige Investitionsentscheidung zu treffen, da eine schmalere Produktpalette und konsequent eine geringere Produktdiversifizierung für KMU eine verringerte Risikostreuung darstellt (Hausman, 2005, S. 777).

Lee et al. (2010, S. 292) betonen, dass KMU bedingt durch begrenzte Ressourcen einen starken Anreiz haben, nach alternativen Optionen zu suchen, um Skaleneffekte zu erzielen. Eine vielversprechende Alternative könnte *Open Source Attended RPA* sein, da das Risiko durch die niedrigeren Investitionskosten reduziert wird (Narula, 2004, S. 156; Taulli, 2020, S. 263). Gleichzeitig ist der Komplexitätsgrad bei Attended RPA geringer. Was genau RPA ausmacht, wird in dem folgenden Kapitel näher erläutert.

3 Robotic Process Automation

RPA ist eine Technologie, mit der routinemäßige Aufgaben von Mitarbeitern automatisiert werden können (Teunissen, 2019, S. 1; Willcocks, Lacity & Craig, 2015, S. 14), indem diese menschliche Handlungen in verschiedenen Anwendungen nachahmt (Langmann & Turi, 2022, S. 6). Hierbei handelt es sich um Software-Roboter, welche im Gegensatz zu gewöhnlichen Roboter nicht physisch mit der Welt interagieren (Willcocks et al., 2015, S. 5). Stattdessen handelt es sich bei diesen um speziell konfigurierte Software, welche Aufgaben und Prozesse gemäß vordefinierten Regeln ausführen. Dies kann je nach RPA-Plattform und Anbieter entweder vor Ort auf dem Computer oder in der Cloud erfolgen (Langmann & Turi, 2022, S. 65). Willcocks et al. (2015, S. 14) erwähnen, dass die Technologie nicht invasiv sei, da es die Benutzeroberfläche und Funktionalität vorhandener Systeme nutzt. Demzufolge bedarf es keiner zusätzlichen Schnittstellenprogrammierung, um mit den Systemen zu interagieren.

Laut Taulli (2020, S. 265) bestehen RPA-Plattformen typischerweise aus zwei Hauptkomponenten: einem Studio oder Designer, in dem Roboter erstellt werden (S. 331), und einem Orchestrator, in welchem Roboter verwaltet werden (S. 330). Das beinhaltet beispielsweise die Planung der Aktivitäten sowie die Überwachung und Beendigung des Roboterbetriebs (S. 330). Hierbei setzen sich die Kosten einer kommerziellen RPA-Plattform aus dem Studio und Orchestrator sowie den Software-Lizenzen zusammen (Taulli, 2020, S. 108), wobei in der Regel pro Lizenz die Nutzung eines Roboters ermöglicht wird (Taulli, 2020, S. 14; Willcocks et al., 2015, S. 5).

Ein weiterer zu beachtender Aspekt, wie von Taulli (2020, S. 108) hervorgehoben, ist ein signifikanter Anstieg in Lizenzkosten von einem Roboter, der die Interaktion eines Menschen erfordert (Attended RPA) zu einer vollständig automatisierten Version (Unattended RPA). Im Weiteren werden die Unterschiede der beiden Varianten veranschaulicht.

Attended RPA

Gelegentlich wird Attended RPA auch als Robotic Desktop Automation (RDA) bezeichnet (Taulli, 2020, S. 6). Hier operiert der Roboter unmittelbar auf dem Desktop des Anwenders, wobei die Nutzer den Roboter direkt über ihre eigenen Bildschirme starten, überwachen und mit diesem interagieren (Langmann & Turi, 2022, S. 6). Während der Roboter eine Aufgabe ausführt, steht der Computer für die Mitarbeiter nicht zur Verfügung (Koch & Fedtke, 2020, S. 8).

In der Beschaffung kann der Roboter beispielsweise eine Preisliste formatieren. Die Mitarbeiter starten den Prozess und haben die Möglichkeit, diesen jederzeit zu beenden. Weiterhin werden die Nutzer bei Fehlern benachrichtigt und können darauf reagieren.

Unattended RPA

In diesem Fall arbeitet der Roboter generell auf einem Server im Hintergrund und startet sowie bewältigt die vorgegebenen Aufgaben selbstständig, ohne Nutzerinteraktion (Langmann & Turi, 2022, S. 6). Die Verarbeitung durch den Roboter wird durch bestimmte Ereignisse gesteuert, wie z. B. die Ablage von Dokumenten in einem spezifischen Ordner (Taulli, 2020, S. 6).

In der Beschaffung kann der Roboter darauf warten, dass alle Berichte vom Lieferanten in einem Ordner eingegangen sind. Anschließend werden diese zu einem einzelnen Dokument zusammengeführt und es wird entweder direkt eine Präsentation erstellt oder das Dokument wird an die dafür vorgesehenen Mitarbeiter weitergeleitet.

Laut Langmann und Turi (2022, S. 6–7) existiert weiterhin die hybride Variante, die Attended- und Unattended-Roboter kombiniert. Ihnen nach kann ein solcher Roboter größtenteils eigenständig im Hintergrund arbeiten und den Menschen gegebenenfalls um Anweisungen oder Lösungen bitten, bevor die Arbeit fortgesetzt wird.

Optical Character Recognition

Optical Character Recognition (OCR) ist eine Technologie, die nicht digitale oder schlecht elektronisch lesbare Daten verarbeitet, sodass diese maschinell lesbar werden (Taulli, 2020, S. 34–35). So kann Text aus handgeschriebenen oder generell eingescannten Dokumenten sowie Bildern mit textuellem Inhalt gewonnen werden. Hierdurch können die entstandenen Daten elektronisch bearbeitet, durchsucht und manipuliert werden (Koch & Fedtke, 2020, S. 124; Taulli, 2020, S. 34–35).

3.1 Mehrwert

Im weiteren Verlauf wird der Mehrwert von RPA anhand der Literatur untersucht.

Brückentechnologie für die Systemintegration

RPA dient als Verbindungselement zwischen verschiedenen Anwendungen, wo eine gewöhnliche Systemintegration oft durch hohe Kosten oder begrenzte IT-Ressourcen nicht durchgeführt werden kann und erleichtert somit die Einführung (Langmann & Turi, 2022, S. 10; Taulli, 2020, S. 14).

Kosteneffiziente Einführung und Betrieb

Geringe Kosten, die mit der Einführung und dem Betrieb von RPA verbunden sind, werden als ein zentraler Grund für die Einführung hervorgehoben (Lacity, Willcocks & Craig, 2015, S. 4).

Benutzerfreundliche Implementierung

Die Einrichtung von Robotern erfolgt häufig durch die Mitarbeiter und nicht durch die zentrale IT-Abteilung (Koch & Fedtke, 2020, S. 8). Die Einfachheit der RPA-Software und die Tatsache, dass keine tiefen Programmiersprachenkenntnisse erforderlich sind, tragen zu der benutzerfreundlichen Implementierung bei (Scheer, 2017, S. 30).

Hoher ROI

Hat einen hohen und schnellen ROI (Lacity et al., 2015, S. 4). Laut Lhuer (2016, S. 2) kann dieser im ersten Jahr zwischen 30 % bis 200 % liegen. Zusätzlich ist die Geschwindigkeit der Amortisation deutlich höher (S. 4).

Skalierbarkeit

RPA bietet eine hohe Skalierbarkeit durch die hohe Arbeitsgeschwindigkeit der Roboter (Langmann & Turi, 2022, S. 11) und der vielen wiederkehrenden Prozessen im Unternehmen (Koch & Fedtke, 2020, S. 15). Darüber hinaus lässt sich die Anzahl der Roboter jederzeit erhöhen (Taulli, 2020, S. 4).

Wiederverwendbarkeit von Modulen

Die Wiederverwendung von Modulen in anderen Prozessen beschleunigt die Entwicklung (Sutherland, 2013, S. 5). Langmann und Turi (2022, S. 11) nach bieten einige RPA-Plattformen auch ihre eigenen Module an.

Operative Effizienz

RPA ersetzt manuelle Tätigkeiten, die zuvor von Mitarbeitern ausgeführt wurden, durch Roboter (Scheer, 2017, S. 29–30). Dies führt zu höherer betrieblicher Effizienz und weniger manueller Arbeit (Sutherland, 2013, S. 5; Taulli, 2020, S. 91). Roboter zeichnen sich durch verkürzte Bearbeitungszeiten aus, gewährleisten eine konstante Arbeitsgeschwindigkeit sowie -qualität und eliminieren Fehler, die auf menschliche Einflüsse zurückzuführen sind (Koch & Fedtke, 2020, S. 11; Langmann & Turi, 2022, S. 11).

Kurze Implementierungszeit

RPA wird wegen seiner relativ kurzen Implementierungszeit favorisiert (Langmann & Turi, 2022, S. 12). Die technische Entwicklung eines Roboters mit moderner RPA-Software kann in nur wenigen Wochen abgeschlossen werden, was zu schnellen Effizienzsteigerungen führt (Lacity et al., 2015, S. 6).

Mitarbeiterzufriedenheit und -bindung

Die Einführung von RPA kann dazu beitragen, die Zufriedenheit der Mitarbeiter, die in den Prozessen involviert sind, zu erhöhen (Langmann & Turi, 2022, S. 12). Die Automatisierung ermöglicht es den Mitarbeitern, sich auf Aufgaben mit höherem Mehrwert zu konzentrieren, was zu einer höheren Arbeitszufriedenheit und einer stärkeren Bindung an das Unternehmen führt (Koch & Fedtke, 2020, S. 12).

Insourcing

RPA kann Kosten senken, indem ausgelagerte Prozesse aus Niedriglohnländern zurückgeführt werden (Fersht & Slaby, 2012, S. 11).

Verbesserte Prozesse ohne direkte Automatisierung

Der Prozess der Einführung von RPA umfasst häufig die Standardisierung und Optimierung von Prozessen vor der Automatisierung (Koch & Fedtke, 2020, S. 11). Dies führt zu verbesserten Prozessen ohne direkte RPA-Implementierung und trägt so zur allgemeinen betrieblichen Effizienz bei (S. 11).

Analytische Einsicht

Der Einsatz von Robotern in Prozessen liefert ein präzises Protokoll der Aktivitäten, wodurch es möglich ist, genau zu verfolgen, wie lange jede Aktivität gedauert hat (Koch & Fedtke, 2020, S. 11). Mit den Erkenntnissen, die aus den generierten Daten gewonnen werden, können Unternehmen Maßnahmen zur Prozessverbesserung einleiten (Koch & Fedtke, 2020, S. 11; Sutherland, 2013, S. 5). Dies kann die Optimierung von Arbeitsabläufen, die Behebung von Engpässen und die Verbesserung der allgemeinen betrieblichen Effizienz umfassen (Sutherland, 2013, S. 5).

3.2 Hürden

Der folgerichtige nächste Schritt besteht darin zu untersuchen, welche Herausforderungen RPA laut der Literatur mit sich bringt.

Verschleierung von IT-Infrastrukturproblemen

Unternehmen entscheiden sich möglicherweise für RPA als schnelle Lösung und vermeiden die notwendigen Investitionen zur Überholung veralteter Systeme (Koch & Fedtke, 2020, S. 18). Dies kann zu einer Verzögerung bei der Lösung grundlegender Probleme mit den IT-Systemen führen (S. 18).

Unübersichtliche Kosten

Gemäß Taulli (2020, S. 14) existieren verschiedene Geschäftsmodelle von RPA-Verkäufern. Während einige auf einem Abonnementprinzip basieren, setzen andere auf die Abrechnung nach der Anzahl der Roboterlizenzen (S. 14). Es könnten auch zusätzliche Kosten für Wartung, Schulungen für Mitarbeiter, Erwerb von Software und Hardware sowie Beraterhonorare anfallen (S. 14).

Kontinuierliche Wartung

RPA-Lösungen erfordern eine kontinuierliche Überarbeitung, Aktualisierung und Anpassung (Taulli, 2020, S. 14), da jegliche Veränderung an der Benutzeroberfläche oder im Prozess dazu führen kann, dass der Roboter nicht mehr ordnungsgemäß funktioniert (Koch & Fedtke, 2020, S. 18). Dieser Wartungsbedarf kann zu zusätzlichem Aufwand und Kosten führen (S. 18).

Förderung von ineffizienten Prozessen

Koch und Fedtke (2020, S. 19) nach, können Unternehmen ineffiziente Prozesse durch den Einsatz von RPA lösen, anstatt diese grundlegend zu überarbeiten.

Sicherheitsrisiko

Mit dem zunehmenden Einsatz von RPA in unternehmenskritischen Bereichen steigt das Risiko von Sicherheitsverletzungen, die zur Preisgabe hochsensibler Informationen führen könnten (Taulli, 2020, S. 15). Das Risiko der Informationspreisgabe besteht in der Regel hauptsächlich bei Unattended Robotern, da bei Attended Robotern das Risiko ähnlich hoch ist wie bei manueller Ausführung durch Mitarbeiter.

Limitierte Einsatzmöglichkeit

RPA ist auf die Nachahmung manueller digitaler Aufgaben beschränkt und kann keine Prozesse bewältigen, die kreatives Denken oder adaptive Maßnahmen erfordern (Koch & Fedtke, 2020, S. 19; Taulli, 2020, S. 16).

Skalierung von RPA

Die Skalierung von RPA im gesamten Unternehmen kann eine Herausforderung darstellen und erfordert eine effektive Koordination und Zusammenarbeit zwischen IT-Teams, um zahlreiche Roboter effektiv zu verwalten (Taulli, 2020, S. 15).

Negative Wahrnehmung und mangelnde Differenzierung

Koch und Fedtke (2020, S. 19) warnen das Kritiker von RPA Misserfolge möglicherweise auf die Technologie selbst zurückführen und nicht auf die grundlegenden IT-Probleme. Somit entsteht die Wahrnehmung, dass Roboter versagen, unabhängig davon, ob das Problem bei der Leistung des Roboters oder der IT-Infrastruktur liegt (S. 19).

Vorbereitungsanforderungen

Eine erfolgreiche RPA-Implementierung erfordert ein eingehendes Verständnis der aktuellen Aufgaben und Prozesse (Taulli, 2020, S. 16). Das Übergehen dieses Schrittes kann dazu führen, dass ineffiziente oder unwirksame Ansätze automatisiert werden (S. 16).

Herausforderungen beim Desktop-Zugriff

RPA-Systeme können in virtualisierten Umgebungen, in denen Desktops aus der Ferne auf Anwendungen zugreifen, auf Probleme stoßen (Taulli, 2020, S. 16). Dies kann die Fähigkeit des Systems beeinträchtigen, Text auf dem Bildschirm zu erfassen (S. 16).

3.3 Potenzielle Prozesse für Robotic Process Automation

Kriterien

Um potenzielle Prozesse für die Implementierung von RPA zu identifizieren, ist es erforderlich, die Kriterien der Automatisierung zu erarbeiten. Koch und Fedtke (2020) teilen die Kriterien in „zwingend erforderliche“ und „von Vorteil für Ihren Business-Case“ (S. 5) auf.

Ein „zwingend erforderliches“ Kriterium nach Koch und Fedtke (2020, S. 5) ist die Auslesbarkeit von digitalen Daten. Nicht alle Dateiformate, insbesondere PDF-Dateien, sind zwangsläufig maschinenlesbar, da die Daten mitunter als Bild gespeichert sind und somit nicht mit dem Mauscursor markiert werden können (Koch & Fedtke, 2020, S. 5). In diesem Fall ist OCR ein möglicher Lösungsansatz (Langmann & Turi, 2022, S. 22).

Des Weiteren müssen die Daten laut Koch & Fedtke (2020) strukturiert und einheitlich aufbereitet sein. Langmann und Turi (2022, S. 21) jedoch argumentieren, dass die initialen Daten nicht einheitlich sein müssen, solange alle möglichen Variationen der Inkonsistenzen bekannt sind. So kann ein Prozessschritt diese in eine einheitliche Form bringen. Ein gängiges Beispiel sind Zahlen mit verschiedenen Interpunktionsregeln, wie etwa bei Geldbeträgen oder Datumsangaben (Koch & Fedtke, 2020, S. 5).

Abschließendes Kriterium laut Koch und Fedtke (2020, S. 5) sind regelbasierte Arbeitsabläufe. Ihnen zu Folge, arbeitet der Roboter mit WENN-DANN-Anweisungen, daher ist es wichtig, dass die Arbeitsabläufe klare Regeln und Strukturen aufweisen. Komplexe Verschachtelungen können die Entwicklung des Robots zeitaufwendiger gestalten und erfordern einen erhöhten Betreuungsaufwand (Koch & Fedtke, 2020, S. 5).

Zu den Kriterien, die „von Vorteil für den Business-Case“ sind, gehören häufig wiederkehrende Aufgaben (Koch & Fedtke, 2020, S. 5). Je regelmäßiger eine Aufgabe durchgeführt wird, desto größer ist das Potenzial für Effizienzsteigerungen durch RPA (Langmann & Turi, 2022, S. 22).

Weiterhin eignen sich Abläufe mit wenig Ausnahmen besonders gut für die Automatisierung (Koch & Fedtke, 2020, S. 5). Je klarer und vorhersehbarer der Arbeitsablauf ist, desto reibungsloser kann die Automatisierung erfolgen (Langmann & Turi, 2022, S. 23).

Zusätzlich bieten sich Prozesse an, die über mehrere Applikationen Daten austauschen müssen (Koch & Fedtke, 2020, S. 5). RPA kann dabei helfen, die nahtlose Integration zwischen verschiedenen Anwendungen zu gewährleisten (Langmann & Turi, 2022, S. 23). Langmann und Turi (2022, S. 23) weisen darauf hin, dass die steigende Anzahl von Anwendungen zu einer Beeinträchtigung der Stabilität des Roboters führen kann. Je mehr Anwendungen im Prozess beteiligt sind, desto wahrscheinlicher ist es, dass Änderungen an diesen vorgenommen werden (Langmann & Turi, 2022, S. 23).

Schließlich sind auch Aktivitäten, die einen erheblichen Zeitbedarf von Mitarbeitern in Anspruch nehmen, potenziell für die Automatisierung durch RPA geeignet (Koch & Fedtke, 2020, S. 5). Durch die Übernahme solcher zeitintensiven Aufgaben kann die Effizienz gesteigert und Ressourcen für strategischere Aufgaben freigesetzt werden (Fung, 2014, S. 2; Krakau, Feldmann & Kaupe, 2021, S. 18).

Weitere Kriterien, die von Koch und Fedtke (2020, S. 5) nicht ausdrücklich erwähnt werden, sind Aufgaben, die fehleranfällig sind und somit indirekte Kosten verursachen, wie beispielsweise erneute Bearbeitung, Strafkosten oder Lieferantenunzufriedenheit (Sutherland, 2013, S. 6).

Hohes Prozessvolumen ist nach Langmann und Turi (2022, S. 22) ein weiteres Kriterium, dass oft zu den Hauptkriterien zählt. Um einen signifikanten Return on Investment (ROI) zu erzielen, sollten die zu automatisierenden Prozesse ein entsprechend hohes Volumen aufweisen (Langmann & Turi, 2022, S. 22). Weiter argumentieren diese, dass die Investitionen und laufenden Betriebskosten für RPA maßgeblich von Faktoren wie dem

spezifischen Anbieter, dem gewählten Lizenzmodell und dem vorhandenen Know-how im Unternehmen beeinflusst werden. Wenn diese Kosten entsprechend niedrig sind, kann sogar die Automatisierung von Prozessen mit mittlerem oder geringem Volumen in kurzer Zeit zu einem positiven ROI führen (Langmann & Turi, 2022, S. 22).

Generalisierung von Prozessschritten

In der Regel können einzelne Prozesse in mehrere Prozessschritte unterteilt werden. Jeder Prozessschritt entspricht einer konkreten Aktivität, welche kombiniert werden können, um eine Vielzahl an Aufgaben zu beschreiben. Es soll versucht werden, die einzelnen Prozessschritte weiter zu unterteilen, um so die kleinsten Einheiten eines Prozessschrittes zu erhalten. Diese können jetzt nochmals auf die Hauptkriterien von Koch und Fedtke (2020, S. 5) überprüft werden.

So nennen Koch und Fedtke (2020, S. 5) in Abbildung 1 gängige Prozessschritte, die in RPA aufkommen.

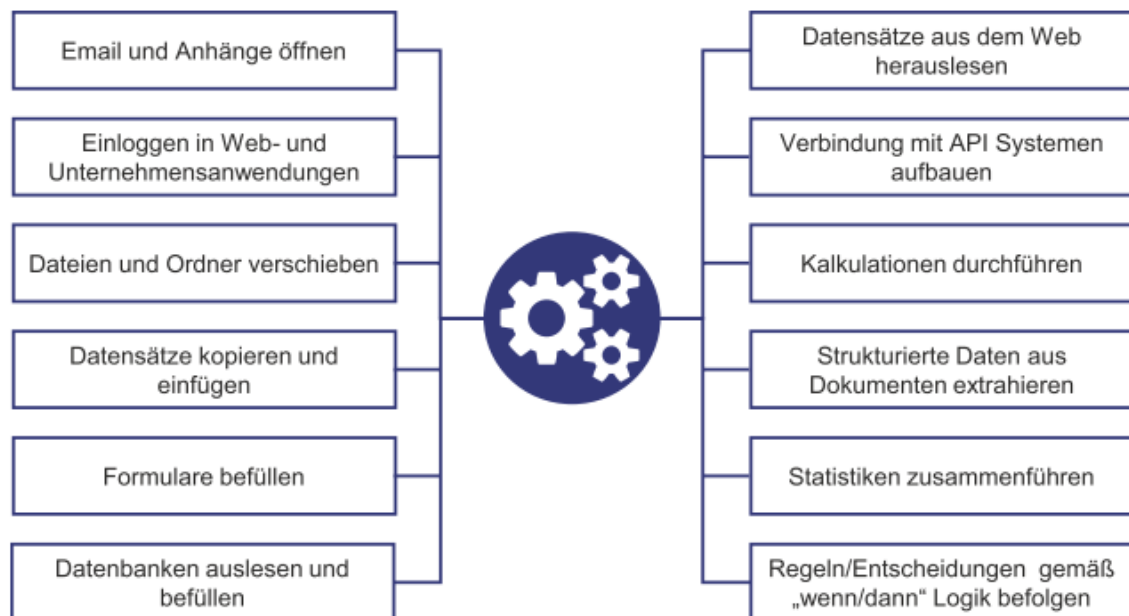


Abbildung 1. Typische RPA-Prozessschritte nach Koch & Fedtke, 2020, S. 5

Aus diesen typischen Schritten lassen sich fünf Operationen ableiten, die in jeder RPA-Anwendung Einsatz finden:

1. **Extrahieren**

Spezifische Daten aus Quellen wie Dokumenten, Websites oder Anwendungen zu lesen oder anderweitig zu entnehmen.

2. **Verarbeiten**

Transformieren von Daten in eine nutzbringende Form. Beispielsweise Berechnungen durchführen oder Texte formatieren.

3. **Einfügen**

Information an einem temporären oder finalen Ort speichern. Wie etwa in Datenbanken oder Formularen.

4. **Generelle Systeminteraktion**

Interaktionen, die nicht unter die ersten drei Operationen fallen. So zum Beispiel das Öffnen von Anwendungen, das Navigieren durch Menüs und Ausführen von Befehlen in anderen Systemen.

5. **Änderungen am Kontrollfluss**

Das Verhalten des Roboters regelbasiert ändern. Hierbei kann eine Aktion bedingt oder mehrfach ausgeführt werden. Wie zum Beispiel bei WENN-DANN-Anweisung oder Wiederholungen.

Mit diesen 5 Operationen lassen sich sämtliche Prozessschritte und folglich Prozesse für RPA generieren. Nach den folgenden Hauptkriterien soll geprüft werden. (Koch & Fedtke, 2020, S. 5; Langmann & Turi, 2022, S. 21):

1. Daten digital auslesebar.
2. Daten strukturiert und entweder bereits einheitlich oder in eine solche Form überführbar.
3. Abläufe regelbasiert.

Folgende Hauptkriterien müssen gelten, damit die Operationen mit RPA automatisiert werden können, wobei die Zahlen für die entsprechenden Hauptkriterien stehen:

Extrahieren: [1, 2, 3]

Verarbeiten: [2, 3]

Einfügen: [2, 3]

Systeminteraktion: [3]

Kontrollfluss: [3]

Extrahieren ist die einzige Operation, für welches das erste Hauptkriterium zählt. Anschließend gilt für Extrahieren, Verarbeiten und Einfügen das zweite Hauptkriterium. Abschließend gilt für alle Operationen das dritte Hauptkriterium.

Das heißt, dass das erste Hauptkriterium nur beim Extrahieren von Daten wichtig ist. Sobald ein Prozess kein Extrahieren beinhaltet, muss das erste Hauptkriterium nicht beachtet werden. Bei Prozessen bestehend aus nur Systeminteraktion, muss nur das dritte Hauptkriterium beachtet werden. Eine Ausnahme muss beim Kontrollfluss gemacht werden, da dieses ohne die anderen Operationen nicht umsetzbar ist.

Dies ermöglicht alle Systeminteraktionen zu identifizieren und auf ein Hauptkriterium zu prüfen. Das heißt, wenn ein Prozess eine Systeminteraktion beinhaltet und dieser nicht regelbasiert ist, kann dieser Prozess nicht allein mit RPA umgesetzt werden. Es ist fragwürdig, ob dieser Ansatz dazu beiträgt, einen Prozess zur Automatisierung zu identifizieren, jedoch könnte er möglicherweise eine Hilfe sein und sollte ausführlicher getestet werden.

Zusammengefasst, um Prozesse in der Beschaffung für KMU zu identifizieren, die potenziell mit RPA automatisiert werden können, gilt es zunächst sicher zu stellen, dass:

1. Daten digital auslesbar sind,
2. Daten strukturiert sind und entweder bereits einheitlich sind oder in eine solche Form überführbar sind,
3. Abläufe regelbasiert sind (Koch & Fedtke, 2020, S. 5; Langmann & Turi, 2022, S. 21).

Wobei die Prozesse nach Extrahier und Systeminteraktion Operationen gefiltert werden können, um eine spezifische und schnellere Auswahl zu erhalten.

Alle weiteren Kriterien, je nach Unternehmen und Branche können unterstützend wirken, um einen robusten Business-Case sowie eine schnelle und kostengünstige Automatisierung zu realisieren (Fersht & Slaby, 2012, S. 7).

3.4 Open Source

Open Source ist eine Form von Software bei der je nach Lizenz, bestimmte Bedingungen jederzeit erfüllt sein müssen (Spath, Schimpf & Kugler, 2007, S. 6). Eine übliche Anforderung ist die Bereitstellung des Quellcodes der Software (S. 6). Dies ermöglicht es jedem, den Code zu verwenden, zu verändern und zu verbreiten (S. 6). Somit fördert dieser Ansatz Transparenz, Innovation und Flexibilität, da Nutzer die Software an ihre speziellen Bedürfnisse anpassen können (Taulli, 2020, S. 260–264). Die Zugänglichkeit des Quellcodes von Open Source wird durch Lizenzen sichergestellt (Open Source Initiative, 2023), wie z. B. die *GNU General Public License* oder die *MIT License*.

Gemäß Ramkumar et al. (2021, S. 3) sowie Recherchen im Internet wurden die folgenden gängigen Open Source RPA-Plattformen identifiziert:

- Robocorp
- Open RPA
- TagUI and taskt
- RobotGo
- UI.Vision (Kantu)
- RPA Python

Trotz der offensichtlichen Kostenersparnis (Lacity et al., 2015, S. 4; Taulli, 2020, S. 263) gibt es spezifische Nachteile im Zusammenhang mit der Verwendung von Open Source RPA.

Implementation und Konfiguration

Die Installation und Konfiguration von Open Source RPA könnte unter Umständen länger dauern und anspruchsvoller sein im Vergleich zu kommerziellen Lösungen (Taulli, 2020, S. 264). Ggf. wird ein IT-Experte benötigt (S. 264).

Dokumentation

Die Dokumentation von Open Source RPA entspricht möglicherweise nicht dem gleichen Qualitätsstandard wie bei kommerziellen Lösungen, was die Einführung und Wartung erschweren kann (Bohli, Kaspar & Westhoff, 2014, S. 77–78).

Kontrolle des Basiscodes

In der Regel haben Unternehmen, die Open Source RPA nutzen, keine direkte Kontrolle über den Basiscode, da dieser jederzeit von den Entwicklern verändert werden kann (Taulli, 2020, S. 263). Dies erschwert die Planung erheblich (S. 263).

Community-Support

Open Source Projekte haben möglicherweise nicht den gleichen Grad an offiziellem Support wie kommerzielle Produkte (Bohli et al., 2014, S. 78). Dies kann bedeuten, dass Unternehmen auf eigene Ressourcen angewiesen sind, um Probleme zu lösen (S. 78).

Begrenzte Funktionalität

Open Source RPA Lösungen könnten im Vergleich zu kommerziellen Alternativen eine begrenzte Funktionalität aufweisen. Dies führt zu Einschränkungen, wenn komplexe Automatisierungsanforderungen erfüllt werden müssen.

Mangelnde Aktualisierung und Weiterentwicklung

Einige Open Source RPA Projekte könnten nicht aktiv weiterentwickelt oder regelmäßig aktualisiert werden (Spath et al., 2007, S. 9). Dies könnte zu Sicherheits- und Kompatibilitätsproblemen führen (Bohli et al., 2014, S. 80).

Eingeschränkte Schulungs- und Lernressourcen

Im Vergleich zu kommerziellen Produkten könnten Open Source RPA Lösungen weniger Schulungs- und Lernressourcen bieten (Spath et al., 2007, S. 9). Dies könnte die Einarbeitung und Nutzung erschweren (S. 9).

Benutzerfreundlichkeit

Die Benutzeroberflächen von Open Source RPA sind in der Regel schwerer zu bedienen, was den Einstieg und die Nutzung erschwert (Taulli, 2020, S. 263–264).

Es ist wichtig zu beachten, dass die genannten Nachteile nicht unbedingt auf alle Open Source RPA Projekte zutreffen. Bei der Entscheidung für eine RPA-Lösung ist es wichtig zu berücksichtigen, dass die verfügbaren Ressourcen, Qualität der Dokumentation, genauen Anforderungen und die Unterstützung der Community vorhanden sind.

3.4.1 AutoHotkey

AHK ist eine Programmiersprache für Windows, durch die Nutzer repetitive Aufgaben automatisieren können. Es ist unter der Open-Source GNU GPLv2 lizenziert (AutoHotkey Foundation LLC, n. d.d) und ist sowohl für Einzelpersonen als auch für kommerzielle Zwecke kostenfrei erwerblich (AutoHotkey Foundation LLC, n. d.a).

Wie jede andere Programmiersprache dient diese als ein Weg, mit Computern zu kommunizieren. Hierbei beschreibt die Syntax die genaue Struktur sowie mögliche Kombination einzelner Zeichen, um ein valides Programm zu generieren (Slonneger & Kurtz, 1995, S. 1). Konkret handelt es sich bei AHK um eine Skriptsprache, welche durch die einfache Nutzung und schnelle Entwicklung gekennzeichnet wird. Die entstehenden Programmdateien werden daher Skripte genannt.

Ein Fokus dieser Sprache wird auf das simple Belegen von Tastenkombinationen mit beliebigen Funktionen gelegt. Hierdurch kann die Produktivität der Arbeiter erhöht werden, indem Unterbrechungen im Workflow minimiert werden.

Weiterhin umfassen häufig genutzte Verwendungszwecke das Simulieren von Maus- und Tastaturaktionen, das Ausführen von Windows-Anwendungen und die Interaktion mit deren Anwendungsfenstern. Zusätzlich können grafische Benutzeroberflächen (Graphical User Interfaces, GUI) erstellt werden, um den Dialog mit dem Programm intuitiver zu gestalten.

Skripte können entweder von der Quellcodedatei oder einer kompilierten Datei ausgeführt werden. Ersteres ermöglicht die einfache Bearbeitung des Programms, benötigt jedoch die Installation von AHK auf dem ausführenden Computer. Letzteres kann nicht weiter verändert werden, kann aber ohne Aufsetzen der Skriptsprache ausgeführt werden. So kann dies genutzt werden, um Programme von einem Computer im gesamten Unternehmen zu verteilen, ohne dass eine Konfiguration auf jeder Maschine notwendig ist.

Um ein besseres Verständnis über den Nutzen von AHK als RPA-Werkzeug zu erlangen, werden im Weiteren einige Stärken und Schwächen geschildert.

3.4.1.1 Vorteile

Die Skriptsprache bringt einige Vorteile mit sich:

Geringer Konfigurationsaufwand

Das Aufsetzen der Sprache ist nach der Installation abgeschlossen und das Entwickeln eines Skriptes kann direkt im Anschluss in einem beliebigen Textverarbeitungsprogramm begonnen werden.

Open Source

AHK ist Open Source, d. h. die Benutzer können ihre Skripte unter den gegebenen Lizenzbeschränkungen ändern und weitergeben.

Flache Lernkurve

Eine simple Syntax vereinfacht die Entwicklung. Selbst Anwendern mit geringer Programmiererfahrung wird die Sprache zugänglich gemacht.

Fortlaufende Entwicklung

AHK wird seit 2003 kontinuierlich entwickelt (AutoHotkey Foundation LLC, n. d.b). So wird mit der Zeit die Funktionalität der Sprache erweitert und Fehler behoben. Weiterhin kann hierdurch eine große Menge an Online-Ressourcen gefunden werden.

Community-Support

Durch eine aktive und hilfsbereite Community auf Internetforen wie *Stackoverflow* und dem AHK-Forum wird den Benutzern das Lernen der Sprache sowie die Suche nach weiteren Ressourcen leicht gemacht.

Systemintegration

Die Interaktion mit anderen Windows-Anwendungen ist unkompliziert und kann damit als eine nahtlose Schnittstelle zwischen verschiedenen Systemen dienen.

AHK-Kostenfrei

Das Runterladen und Benutzen von AHK ist kostenfrei (AutoHotkey Foundation LLC, n. d.a).

3.4.1.2 Nachteile

Gleichzeitig zieht die Nutzung mehrere Nachteile nach sich:

Skalierbarkeit

Während AHK für kleine bis mittelgroße Automatisierungsaufgaben geeignet ist, so skaliert diese Sprache schlecht bei der Entwicklung komplexer Projekte. Dedizierte RPA-Tools bieten oft eine bessere Alternative.

Windows-exklusiv

AHK wurde für Windows als Zielplattform entwickelt und ist nicht mit anderen Betriebssystemen wie MacOS oder Linux kompatibel.

Umständliche GUI-Entwicklung

Die Sprache unterstützt zwar die Entwicklung grafischer Benutzeroberflächen, erlaubt jedoch im Vergleich zu anderen Programmiersprachen den Aufbau einer komplexeren Oberfläche nur schwierig.

Unausgereifte Werkzeuge zur Fehlerdiagnose

Werkzeugen zum Identifizieren von Fehlern fehlen einige Funktionen, um die Arbeit mit diesen effizient auszuführen. Die Fehlersuche kann deshalb in komplexen Skripten mehr Aufwand und Zeit erfordern.

Lernkurve fortgeschrittener Funktionen

Während die grundlegende Skripterstellung leicht verständlich ist, erfordern einige der fortgeschrittenen Funktionen ein Wissen über die C-Programmiersprache sowie einiger Windows Programmierschnittstellen.

Begrenzte RPA-Funktionalität

AHK wurde nicht mit dem konkreten Ziel von RPA entwickelt. Als solches fehlen einige der erweiterten Funktionen, die bei umfassenden RPA-Werkzeugen vorhanden sind. Dies könnte die Eignung für komplexe Automatisierungsprojekte beeinträchtigen.

3.4.2 Gewählter Implementierungsansatz

Einfacher Ansatz

Als simpler Lösungsansatz wurde der Makrorekorder mit dem Namen *Macro Recorder v2.1+* (AutoHotkey Foundation LLC, n. d.c), erstellt vom AHK-Forum Benutzernamen *FeiYue*, gewählt.

Der Begriff Makro beschreibt die Möglichkeit, mit einem einzelnen Aufruf eine Menge an Anweisungen auszuführen. Auf dieser Idee aufbauend, zeichnet ein Makrorekorder mehrere Maus- und Tastatureingaben eines Nutzers auf und ermöglicht eine spätere Wiedergabe dieser Aktionen. Aufgezeichnete Aktionen werden automatisch zu AHK-Code umgewandelt, welcher bei Bedarf manuell bearbeitet werden kann. Weiterhin ist ein Exportieren zu einer Skriptdatei möglich, welche ein Ausführen des Makros unabhängig vom Rekorder ermöglicht.

Nach dem Start des *Macro Recorder v2.1+* wird am oberen Bildschirmrand eine grafische Benutzeroberfläche mit den jeweiligen Funktionstasten angezeigt (siehe Abbildung 2).

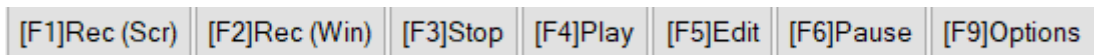


Abbildung 2. GUI Macro Recorder v2.1+

Nach dem Drücken der Taste *F1* oder *F2* wird jede von Benutzern durchgeführte Aktion aufgezeichnet. Der Unterschied zwischen den beiden Aufzeichnungsfunktionen liegt in der Robustheit. Mit *F2* werden die spezifischen Fensterbezeichnungen mitaufgenommen, wodurch sichergestellt wird, dass das Makro erst beim Fokus eines Fensters mit demselben Namen weiterläuft. So wird mit *F1* nie sichergestellt, ob ein Fenster im Fokus bzw. ob das richtige Fenster ausgewählt ist.

Korrekturen an aufgenommenen Makros lassen sich am einfachsten durchführen, indem der gesamte Prozess von neuem aufgezeichnet wird. Eine manuelle Anpassung bereits aufgezeichneter Aktionen ist ebenfalls möglich. Hier kann weiterhin die Wiedergabegeschwindigkeit des Makros angepasst werden, indem die Variable *Playspeed* in der obersten Zeile des Skriptes verändert wird.

Mittlerer Ansatz

Ein weiter fortgeschrittener Lösungsansatz bietet ein Makrorekorder mit erweiterter Funktionalität. Hierzu wurde *Pulover's Macro Creator* (Batista, n. d.), erstellt von Rodolfo Urbano Batista, gewählt.

Neben der grundlegenden Aufnahme von Nutzeraktionen können hier Aktionen auch ohne die Aufzeichnung von Maus und Tastatur erzeugt werden, indem mit den GUI-Elementen interagiert wird. Ähnlich zum ersten Ansatz kann der aufgezeichnete Code editiert bzw. erweitert werden, wobei dies direkt in der Anwendung geschieht. Die Möglichkeit, den Code in eine Skriptdatei zu exportieren, ist vorhanden.

Beim Start öffnet sich ein Fenster mit vielen Icons und zwei Fensterabschnitten (siehe Abbildung 3). Im linken Teil werden die Anweisungen und Funktionen in grafischer Form dargestellt, während im rechten Teil der reine AHK-Code angezeigt wird.

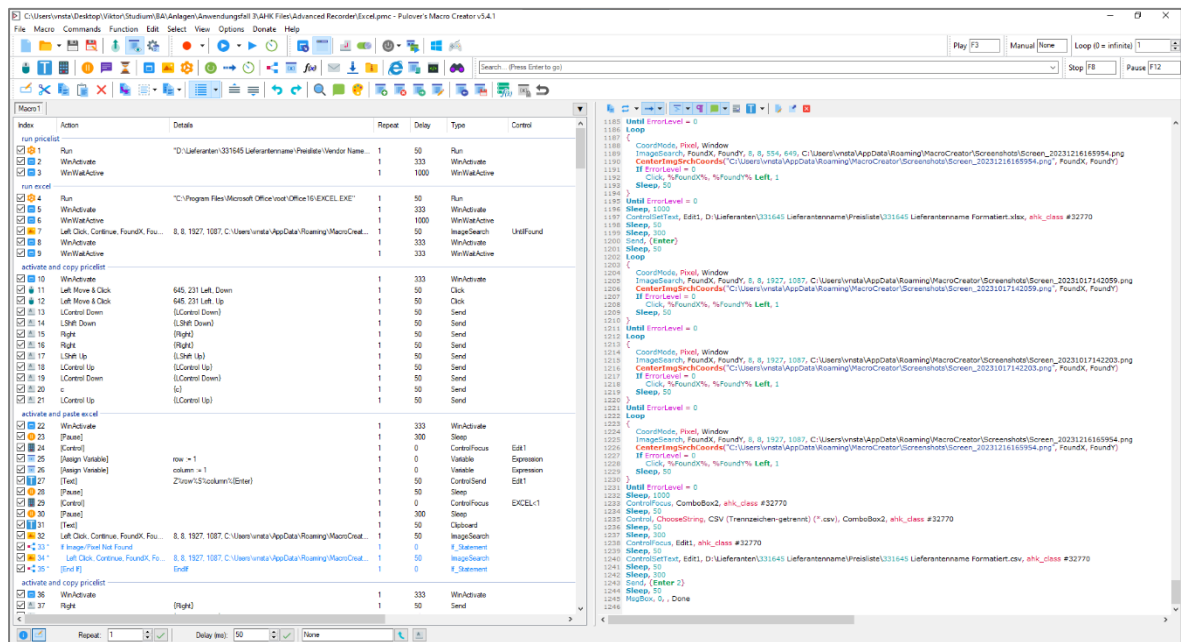


Abbildung 3. GUI Pulovert's Macro Creator

Das Programm bietet darüber hinaus ein breites Spektrum an Funktionen wie OCR, die durch Anklicken aktiviert werden können. Jede dieser Funktionen kann in einem Dialogfenster weiter konfiguriert werden, um ein gewünschtes Verhalten zu erzielen.

Fortgeschrittener Ansatz

Als anspruchsvollsten Ansatz wird das direkte Programmieren über AHK gewählt. Hierzu wird die Open-Source Entwicklungsumgebung *Visual Studio Code* verwendet. Obwohl der Editor die Entwicklung und Fehlersuche stark vereinfacht, ist dieser nicht zwingend erforderlich, um Anwendungen in dieser Sprache zu verfassen.

Dieser Ansatz gilt als fortgeschritten, da hierbei ein größerer Wert auf Robustheit, d. h. Anwendungen sollen aus verschiedenen Kontexten erfolgreich dieselbe Aufgabe bewältigen können sowie weniger Fehler produzieren.

Es wird weiterhin eine Windowsprogrammierschnittstelle, die Component Object Model (COM) Schnittstelle genutzt, um direkter mit Anwendungen kommunizieren zu können. Diese erlaubt beispielsweise das Ausführen von Funktionen in Excel in der anwendungsspezifischen Programmiersprache.

4 Anwendungsfälle

Es wurden drei Anwendungsfälle für die Beschaffungslogistik eines KMU entwickelt, welche die Kriterien aus dem Unterkapitel 3.3 nach Koch und Fedtke (2020, S. 5) erfüllen und somit durch RPA automatisierbar sind. Diese werden mit den drei Implementierungsansätzen aus Unterabschnitt 3.4.2 umgesetzt, wobei die Machbarkeit dieser zu bewerten ist. Die Anwendungsfälle sind dabei nach zunehmendem Umfang sortiert.

Es wird in diesem Szenario von einem KMU mit den folgenden Softwarelösungen ausgegangen: Windowsbetriebssystem, ERP-System und Office-Anwendungen.

In den anschließenden Unterkapiteln folgt eine Beschreibung des jeweiligen Anwendungsfalls sowohl gewöhnlich durch Menschen als auch mit einem RPA-Roboter. Da diese Arbeit in keinem echten Unternehmen abgeschlossen wird, müssen Limitationen durch Software wie ERP-System entsprechend simuliert werden. Hierzu wird kurz auf die genutzten Alternativen eingegangen.

Für jeden Lösungsansatz wird dann die Vorgehensweise veranschaulicht, gefolgt von den dabei beobachteten Besonderheiten und Problemen. Ebenso ist hier die benötigte Entwicklungszeit dokumentiert, welche in Wissensaneignung, Implementierung und Testen aufgeteilt werden kann. Konkrete Zahlen zu diesen drei Zeitaufteilungen sind jedoch nicht gemessen worden, da diese oftmals sehr eng miteinander verknüpft waren.

4.1 Lieferantenzugang und Informationserfassung

Es gibt verschiedene Gründe dafür, dass ein Lieferant nicht direkt an das ERP-System angebunden ist. Dies führt dazu, dass Informationen auf alternative Weise beschafft werden müssen. Zu diesem Zweck erfolgt ein Zugriff auf die Internetseite des Lieferanten unter Verwendung der entsprechenden Zugangsdaten. Nach erfolgreichem Login wird durch das Suchfeld nach der Bestellnummer gesucht und ein zugehöriges Lieferdatum wird extrahiert. Diese Informationen werden abschließend von Mitarbeitern in das ERP-System eingetragen. Dieser Prozess wird für jedes Bauteil wiederholt.

Sobald der RPA-Roboter aktiviert wurde, navigiert dieser zur Internetseite des Lieferanten und authentifiziert sich mithilfe der Benutzerdaten der Mitarbeiter. Der Roboter pausiert nun und wartet auf eine Tastenkombination zum Fortsetzen der Aufgabe. Während dieser Periode kopieren die Mitarbeiter die gewünschte Bestellnummer eines Bauteils. Anschließend wird die Arbeit des Roboters durch die Tastenkombination fortgesetzt, wobei dieser die kopierte Bestellnummer in das Suchfeld der Website eingefügt. Daraufhin liest der Roboter das Lieferdatum aus der entsprechenden Spalte, welches dann von den Mitarbeitern in das ERP-System eingetragen wird. Dieser Prozess kann wiederholt werden, beginnend ab dem Kopieren der nächsten Bestellnummer.

Da keine Lieferantenseite zur Verfügung steht, wird der Prozess simuliert: Hierzu greift der RPA-Roboter auf die E-Mail-Adresse der Hochschule im Internet zu und authentifiziert sich. Nachdem eine Bestellnummer kopiert und die Fortsetzungstaste betätigt wurde, navigiert der Roboter zum Suchreiter und führt eine Suche nach der Bestellnummer durch. Eine E-Mail mit dem gesuchten Betreff *Bestellnummer* und *Lieferdatum* erscheint, welche dieser in der Zwischenablage speichert.

Es ist wichtig zu erwähnen, dass die Login-Sicherheit in diesem Fall nicht berücksichtigt wurde, da Benutzername und Passwort unverschlüsselt im Quellcode zu sehen sind.

Macro Recorder v2.1+

Entwicklungszeit: 30 Minuten.

Vorgehensweise:

1. Mit der Tastenkombination *Windows + R* den *Ausführen-Dialog* starten.
 - a. Die Internetadresse eingeben und den Dialog bestätigen.
2. Benutzername und Passwort eingeben und bestätigen.
3. Auf die Tastenkombination zum Fortsetzen der Arbeit warten.
4. Auf die Schaltfläche *Suchen* drücken.
 - a. In die Suchleiste drücken und die von den Mitarbeitern kopierte Bestellnummer einfügen und suchen.
 - b. Das Datum aus dem Betreff der E-Mail auswählen, kopieren und wieder zur Startseite zurückkehren.
5. Erneut warten bis die Tastenkombination zum Fortsetzen betätigt wird.

Besonderheiten:

Mit der Tastenkombination *Windows-Taste + R* wird der *Ausführen-Dialog* in deutscher Systemsprache oder *Run-Dialog* in englischer Systemsprache aufgerufen. Dieses bietet eine schnelle Möglichkeit, Befehle auszuführen oder Programme zu öffnen, ohne mit der grafischen Benutzeroberfläche interagieren zu müssen. Die Eingabe einer Internetadresse in diesem Dialog führt zum Start des als Standard festgelegten Internetbrowsers und der Navigation zur eingegebenen Internetadresse.

Da ein Makro lediglich die Aktionen eines Menschen aufzeichnet und diese identisch abspielt, kann die wiederholende Funktion dieses Prozesses nicht mit diesem Ansatz alleine implementiert werden. Wird ein Makro aufgenommen, dass die Benutzer anmeldet und nach einer Bestellnummer sucht, so kann dies nur für ein einzelnes Bauteil erfolgen. Grund dafür ist, dass ein erneutes Ausführen eine erneute Anmeldung zufolge hat.

So kann dieses Problem in zwei Alternativen gelöst werden. In jedem Fall werden die Login-Aktionen von den Suchaktionen getrennt, d. h. es werden zwei Makros aufgezeichnet. Die erste Lösung trennt das Ausführen der beiden Makros voneinander ab, indem das Such-Makro bei Bedarf manuell gestartet wird. Die zweite Option beinhaltet das Hinzufügen einer Codezeile am Ende des Login-Makros, um das zweite Makro automatisch aufzurufen. Im zweiten Makro muss ebenso eine Tastenkombination festgelegt werden, die dieses Makro ausführt. Auf diese Weise kann der Prozess effektiver mit weniger Unterbrechungen genutzt werden.

Pulover's Macro Creator

Entwicklungszeit: 2 Stunden 20 Minuten.

Vorgehensweise:

Die Vorgehensweise entspricht zum großen Teil der des *Macro Recorder v2.1+* mit Ausnahme von 4.b. So wird hier die E-Mail direkt geöffnet und das Lieferdatum mit Hilfe von OCR aus dem Betreff ausgelesen. Anschließend wird erneut zur Startseite navigiert und auf die Tastenkombination gewartet.

Besonderheiten:

Dieses Programm ist näher am Code integriert, was das Aufzeichnen des gesamten Prozesses als ein Makro ermöglicht. Die einzelnen Schritte werden dann direkt im linken Fensterabschnitt weiterbearbeitet. Hierdurch kann der Abschnitt des Suchprozesses so modifiziert werden, dass auf eine Tastenkombination gewartet wird.

Der aus der Makro-Aufzeichnung generierte Code wurde stark abgeändert, indem das Warten zwischen Maus- und Tastatur-Aktionen vollständig entfernt wurde. Stattdessen wurde die *Image-Search*-Funktion eingesetzt, welche eine gezielte Suche nach festgelegten Bildern ermöglicht und anschließend eine Aktion durchführt. Diese Bilder wurden an Bereich der Website festgelegt, die lediglich einmal vorkommen. Hierdurch kann der Zustand des E-Mail-Postfaches und damit der Fortschritt des Gesamtprozesses zuverlässig festgestellt werden.

Der Roboter sucht somit nach jedem Prozessschritt nach einem Bild für den nächsten Teil. Wird dieses gefunden, so kann der nächste Schritt durchgeführt werden. Dadurch führt eine Variation in der Ladegeschwindigkeit der Internetseite zu keinen Problemen beim Ausführen des Programms.

Um das Auslesen des Datums zu gewährleisten, wurde OCR genutzt. Die Schrift des Betreffs konnte hier jedoch nicht zuverlässig erkannt werden, da eine E-Mail als Hyperlink eine blaue Schriftfarbe aufweist. Folglich musste die E-Mail zusätzlich geöffnet werden, um eine Betreffzeile in schwarzer Farbe zu erhalten.

Programmierung

Entwicklungszeit: 10 Stunden

Vorgehensweise:

Die Vorgehensweise unterscheidet sich hier kaum vom *Pullover's Macro Creator*. Da jedoch kein Makro-Rekorder genutzt wird, um Maus- und Tastatur-Eingaben zu simulieren, müssen diese per Hand geschrieben werden. Weiterhin wird die Internetseite nicht durch das Aufrufen des *Ausführen-Dialogs* gestartet, sondern direkt den Aufruf einer Funktion im Code.

Besonderheiten:

Bei der Programmierung traten gängige Herausforderungen der Softwareentwicklung auf. Darunter Missverständnisse wie eine Funktion korrekt genutzt wird sowie Flüchtigkeitsfehler im Code. Zusätzlich war dies das erste anspruchsvolle Projekt, das in AHK programmiert wurde. Mit dem Aufsetzen einer externen Programmierbibliothek für OCR ist die Entwicklungszeit über die eingeplante Zeit geschritten.

Um den Prozess weniger fehleranfällig zu gestalten, wurde versucht, auf die Schnittstelle des Internet-Browsers zuzugreifen. Da das erforderliche Wissen nicht vorhanden war, wurde sich aus Zeitgründen dagegen entschieden. Stattdessen wurde, wie beim *Pullover's Macro Creator*, die *Image-Search*-Funktion verwendet, um den Prozessfortschritt zu prüfen.

Beim Arbeiten mit der *Image-Search*-Funktion, ist die Idee entstanden, anstelle mit festgelegten Koordinaten auf dem Bildschirm direkt auf die gefundenen Bilder von GUI-

Elementen zu klicken. So wurde zum Beispiel nach einem Bild der Suchleiste gesucht und erst nach erfolgreicher Identifikation dieser wurde diese angeklickt. Hierdurch kann die Suchaktion mit weniger Fehlern durchgeführt werden.

Das Programmieren umfasst erneut das Anwenden von OCR. Hierzu wurde die *Descolada-OCR* Bibliothek für AHK genutzt ("*Descolada OCR*" [GitHub], n. d.), welche auf der OCR-Bibliothek der Windows Programmierschnittstelle basiert. Dabei gab es oft falsch ausgelesene Werte für das Lieferdatum. Um dieses Problem zu beheben, musste die Größe des Bildes in dem Funktionsaufruf stark skaliert werden. Trotz der momentanen Einstellungen liefert OCR jedoch weiterhin falsche Werte in anderen Kontexten.

4.2 Vergleich vom Lieferdatum

Für die Produktionsplanung ist es entscheidend zu erfahren, wann das letzte Bauteil einer Baugruppe eintreffen wird. Dementsprechend müssen die Mitarbeiter in der Beschaffung das ERP-System öffnen, zu der entsprechenden Position navigieren und den Filter einrichten, der ausschließlich die zu bearbeitende Elemente anzeigt. Anschließend vergleichen die Mitarbeiter die Liefertermine von Bauteil 1 und Bauteil 2, tragen das späteste Datum in die Spalte *Reales Lieferdatum* ein und wiederholen diesen Vorgang, bis alle Baugruppen ein tatsächliches Lieferdatum aufweisen.

Die Mitarbeiter öffnen das ERP-System, das die zu bearbeitende Elemente enthält, und initiieren den RPA-Roboter. Dieser vergleicht die beiden Liefertermine und trägt das spätere Datum in die Spalte *Reales Lieferdatum* ein. Der Roboter stoppt automatisch, wenn keine weiteren Lieferdaten zum Vergleichen vorhanden sind.

Da kein ERP-System zur Verfügung steht, wird der Prozess in Excel simuliert.

Im Weiteren muss die Excel-Datei mit den Lieferdaten (ERP-Datei) bereits vor dem Ausführen des Roboters geöffnet sein.

Macro Recorder v2.1+

Entwicklungszeit: 30 Minuten.

Vorgehensweise:

1. Mit der Tastenkombination *Windows + R* den *Ausführen-Dialog* starten.
 - a. Den Pfad zu Excel eingeben und den Dialog bestätigen.
2. In der neu geöffneten Excel-Datei (Zielfile) eine leere Arbeitsmappe erstellen und zu der ERP-Datei wechseln.
3. Für alle *N* Lieferdaten in der ERP-Datei:
 - a. Kopieren des n-ten Lieferdatums für Bauteil 1, wechseln zur leeren Zielfile, einfügen in die erste Spalte und Wechseln zur ERP-Datei.
 - b. Kopieren des n-ten Lieferdatums für Bauteil 2, wechseln zur leeren Zielfile, einfügen in die zweite Spalte und Wechseln zur ERP-Datei.
4. Zur Zielfile wechseln und in der dritten Spalte eine *WENN*-Funktion einfügen, welche die erste Spalte mit der zweiten nach Größe vergleicht und das größere Ergebnis ausgibt.

5. Die *WENN*-Funktion auf die anderen $N-1$ Zeilen derselben Spalte durch Ziehen der unteren rechten Ecke anwenden.
6. Die Spalte mit den Funktionen kopieren, in die vierte Spalte einfügen und in ein Datum umwandeln.
7. Die Werte der vierten Spalte kopieren und in der ERP-Datei unter der Spalte *Reales Lieferdatum* einfügen.

Besonderheiten:

In dieser Ausführung wurde die menschliche Interaktion größtenteils entfernt. So müssen die Mitarbeiter lediglich die ERP-Datei öffnen und den Roboter starten.

Wiederholt kam es zu Fehlern, da die Abspielgeschwindigkeit des Makros zu hoch eingestellt war. So hat zum Beispiel Excel noch nicht vollständig geladen, während das Makro bereits beginnt Daten einzufügen. Die Abspielgeschwindigkeit wurde hierbei solange reduziert bis keine Fehler auftraten. Zusätzlich wurde durch Limitationen in dem Makro-Rekorder das Testen erschwert, da das Abspielen des Makros auf ein Fenster mit demselben Namen wie in der Aufzeichnung wartet. So wurde in diesem Prozess eine leere Arbeitsmappe erstellt, welche automatisch von Excel den Namen *Mappel* zugewiesen bekommen hat. Wird eine weitere neue Datei in Excel erstellt, so erhält diese den Namen *Mappe2* und wird folglich nicht mehr vom Makro erkannt. Es ist anzumerken, dass dieses Verhalten nur zu beobachten ist, wenn mindestens eine Excel-Instanz aktiv ist. Dies ist hier mit der Excel-Datei, die das ERP-System simuliert, der Fall. Folgendermaßen mussten nach jeder Durchführung des Makros alle Excel-Dateien geschlossen werden.

Ohne aufwendiges programmieren lässt sich die Anzahl an zu bearbeiteten Daten nicht im Vorhinein ermitteln, wodurch nur so viele Lieferdaten verglichen werden können, wie initial aufgezeichnet wurde. Steigt also die Anzahl an Datensätzen, so werden diese nicht vollständig vom Makro verarbeitet.

Pullover's Macro Creator

Entwicklungszeit: 3 Stunden.

Vorgehensweise:

1. Für alle N Lieferdaten in der ERP-Datei:
 - a. Mit der *Image-Search*-Funktion nach dem Lieferdatum für Bauteil 1 suchen, kopieren und in einer Variablen zwischenspeichern.
 - b. Mit der *Image-Search*-Funktion nach dem Lieferdatum für Bauteil 2 suchen, kopieren und in einer Variablen zwischenspeichern.
 - c. Beide Variablen in eine Zahl umwandeln und miteinander vergleichen.
 - d. Größere Zahl in ein Datum umwandeln und in einer Variablen speichern.
 - e. Mit der *Image-Search*-Funktion nach der Spalte Reales Lieferdatum suchen von dieser Position n -mal die Höhe einer Zeile addieren.
 - f. In die Zelle klicken und die Variable einfügen.

Besonderheiten:

Die Aufzeichnungsfunktion wurde ab diesem Anwendungsfall nicht weiter genutzt. Stattdessen wurden ausschließlich mit den Funktionen der GUI-Elemente gearbeitet.

Zum Vergleich der Lieferdaten wird die *Wenn*-Funktion des Rekorders genutzt. Ein Datum-zu-Datum-Vergleich kann jedoch nicht direkt erfolgen und diese müssen zunächst in eine Zahl umgewandelt werden. Dazu liegt kein direkter Weg in der Anwendung vor, aber es könnte über die Schaltfläche *Functions* auf jegliche eingebauten Funktionen von AHK zugegriffen werden. Dies ermöglicht eine Umwandlung des Datums in eine Zahl, welche anschließend weiter bearbeitet werden kann.

Ist die Menge an zu vergleichenden Daten im Vorhinein bekannt, so kann die Anzahl an wiederholenden Prozessschritten vor dem Start des Makros festgelegt werden. Falls also mehr Datensätze hinzugefügt werden, kann der Roboter mit der neuen Anzahl aktualisiert werden.

Programmierung

Entwicklungszeit: 5 Stunden.

Vorgehensweise:

Die Vorgehensweise gleicht sehr dem Ansatz des *Pullover's Macro Creator*. So liegt der einzige Unterschied in der Wiederholung des Prozesses in Schritt 1: Anstelle die Anzahl an Lieferdaten vorab festzulegen, beginnt der Prozess in der Zeile der ersten zwei Lieferdaten, folgt der Vorgehensweise wie oben beschrieben bis Schritt 1.f und prüft anschließend, ob unterhalb der zuletzt kopierten Lieferdaten der ERP-Datei ein weiteres Datum gefunden werden kann. Ist dies der Fall, so findet eine Wiederholung statt; andernfalls stoppt der Roboter die Ausführung.

Besonderheiten:

Ein erheblicher Zeitaufwand wurde dem Verbessern der Lesbarkeit und Robustheit aufgewandt. Hierzu war das Erlernen neuer Programmierkonzepte, wie objektorientiertem Programmieren, erforderlich.

4.3 Umwandlung einer Preisliste

Die Mitarbeiter erhalten eine E-Mail vom Lieferanten mit den aktualisierten Preisen als PDF-Format. Diese wird im zugehörigen Lieferantenordner gespeichert und in eine Excel-Datei konvertiert. Hierfür werden die Werte einzeln aus dem PDF-Dokument kopiert und in die Excel-Datei eingefügt. Nachdem die Preisliste in Excel erstellt und gespeichert wurde, erfolgt eine Formatierung gemäß den internen Angaben. Die Datei wird schließlich als CSV-Format gespeichert, um vom ERP-System ausgelesen werden zu können.

Die Mitarbeiter speichern die PDF-Datei im zugehörigen Ordner ab und starten den RPA-Roboter. Dieser öffnet daraufhin sowohl die PDF-Preisliste als auch Excel und beginnt die Preisinformationen in die Tabelle zu überführen. Nach Abschluss markiert der Roboter die Spaltenüberschriften und formatiert sie fett. Da der Standort des Lieferanten in den USA liegt und somit ein anderes Zahlentrennsystem als in Deutschland verwendet wird, muss die Spalte der Preise angepasst werden. Dabei werden alle Tausender-Trennzeichen entfernt und die Dezimaltrennzeichen von Punkt zu Komma vereinheitlicht. Anschließend wird die Spalte *Lead Time* angepasst, welche die Durchlaufzeit des Lieferanten angibt. Um die

tatsächliche Lieferzeit zu erlangen, muss die Zeit für den Lieferweg mit eingerechnet werden. Hierzu wird eine Lieferzeit von drei Tagen addiert. Die Spalte mit dem Namen *Vendor Name* wird in *Lieferanten ID* umbenannt und alle Zellen mit dem Lieferantennamen werden durch die interne Lieferanten-ID ersetzt. Das Ergebnis wird sowohl im XLSX- als auch im CSV-Format im Lieferantenordner gespeichert und alle Dateien werden geschlossen.

Macro Recorder v2.1+

Entwicklungszeit: 6 Stunden.

Vorgehensweise:

1. Mit der Tastenkombination *Windows + R* den *Ausführen-Dialog* starten.
 - a. Den Pfad zur Preisliste eingeben und den Dialog bestätigen.
2. Mit der Tastenkombination *Windows + R* den *Ausführen-Dialog* starten.
 - a. Den Pfad zu Excel eingeben und den Dialog bestätigen.
3. Eine leere Arbeitsmappe in Excel erstellen und zur Preisliste wechseln.
4. Für alle *N* Spalten der Preisliste:
 - a. Für alle *M* Zeilen der Preisliste:
 - i. Den Eintrag in der *m*-ten Zeile und *n*-ten Spalte kopieren und in die zugehörige Zelle in der Excel-Datei einfügen.
5. Die übertragene Tabelle kopieren, ein zweites Tabellenblatt erstellen und dort einfügen.
6. Spalte ‚Lead Time‘ formatieren:
 - a. Zugehörige Spalte markieren und mit *Suchen und Ersetzen* das Wort ‚Days‘ löschen.
7. Spalte ‚Preise‘ formatieren:
 - a. Zugehörige Spalte markieren und mit *Suchen und Ersetzen* alle Kommas löschen.
 - b. Weiter mit *Suchen und Ersetzen* alle Punkte durch Kommas ersetzen.
8. Spalte ‚Vendor Name‘ formatieren:
 - a. Zugehörige Spalte markieren und mit *Suchen und Ersetzen* ‚Vendor Name‘ durch ‚331645‘ ersetzen.
 - b. Kopfzeile in ‚Lieferanten ID‘ umbenennen.
9. Tatsächliche Lieferzeit berechnen:
 - a. In der Spalte rechts von ‚Lead Time‘ in die Zelle der zweiten Zeile klicken.
 - b. Mit der *SUMME*-Funktion den Zahlen aus der Spalte ‚Lead Time‘ die Zahl 3 addieren.
 - c. Die *SUMME*-Funktion auf die restlichen Zeilen von ‚Lead Time‘ durch Ziehen der unteren rechten Ecke anwenden.
 - d. Die so entstandene Spalte ohne Tabellenkopf kopieren und alle Zellen der Spalte ‚Lead Time‘ ersetzen.
10. Speichern der Tabelle für das ERP-System:
 - a. Die Reiter ‚Datei‘ → ‚Speichern unter‘ → ‚Durchsuchen‘ anklicken.
 - b. Den Pfad für den Lieferantenordner eintippen und als CSV-Format speichern.

11. Speichern der Tabelle ohne Formatierung:
 - a. Auf das erste Tabellenblatt wechseln.
 - b. Die Reiter ‚Datei‘ → ‚Speichern unter‘ → ‚Durchsuchen‘ anklicken.
 - c. Den Pfad für den Lieferanten Ordner eintippen und als XLSX-Format speichern.
12. Die Preisliste und Excel schließen.

Besonderheiten:

Durch einen Flüchtigkeitsfehler wurde hier die F1-Taste anstelle der F2-Taste zum Aufzeichnen genutzt. Infolgedessen wird nicht auf ein spezifisches Fenster gewartet, sondern bedingungslos Maus- und Tastatureingaben an den aufgezeichneten Stellen und Zeiten ausgeführt. Es ist also von besonderer Bedeutung, hier auf den Startkontext zu achten. Beim Abspielen sollten dieselben Anwendungen geöffnet sein, wie beim Aufzeichnen des Makros. Ist dies nicht der Fall, besteht die Möglichkeit, zu diesem Fenster zu wechseln und irreparable Änderungen durchzuführen.

Durch die wesentlich höhere Komplexität dieses Anwendungsfalls erhöht sich die Aufnahmezeit und damit auch die Wahrscheinlichkeit, Fehler zu begehen. Zusätzlich macht sich eine Einschränkung beim Kopieren der Preisliste deutlich: Zeilen müssen manuell kopiert werden. Es ist keine Möglichkeit, vorhanden diesen Prozess ohne Programmierung zu umgehen. Nach mehreren fehlerhaften Aufzeichnungen und der Erkenntnis, dass das Testen mit einer zunehmenden Anzahl an Zeilen länger dauert, wurde beschlossen, nur fünf Zeilen aus der Preisliste zu kopieren und zu verarbeiten.

Pullover's Macro Creator

Entwicklungszeit: 16 Stunden.

Vorgehensweise:

Das Vorgehen unterscheidet sich zu Beginn stark zu dem vorherigen Ansatz:

1. Preisliste aus dem Lieferantenordner öffnen.
2. Excel öffnen und leere Arbeitsmappe erstellen.
3. Für N Zeilen der Preisliste:
 - a. Für M Spalten der Preisliste:
 - i. Zur Preisliste wechseln und vor die m -te Spalte der n -ten Zeile klicken.
 - ii. Mit *Strg* + *Shift* + 2-mal *Rechte Pfeiltaste* den nächsten Eintrag auswählen und kopieren.
 - iii. Zu Excel wechseln und über die *Control*-Funktion des Rekorders:
 - (1) Das *Namenfeld* auswählen, die Koordinaten der Zelle in Zeile n und Spalte m eintippen und bestätigen.
 - (2) Die *Bearbeitungsleiste* auswählen, den kopierten Eintrag einfügen und bestätigen.

Von hieraus folgt der Prozess nach wie vor dem *Macro Recorder v2.1+* mit dem Unterschied, dass einzelne GUI-Elemente nicht durch die Aufzeichnung von Mauseaktionen, sondern durch die Nutzung der *Image-Search*- und *Control*-Funktionen ausgewählt werden. Wie zuvor werden Berechnungen mit Excel-Funktionen durchgeführt.

Besonderheiten:

Um Fehler durch ungenaue Mausaktionen zu verhindern, wurde die Preisliste nach dem ersten Klick an der richtigen Position ausschließlich mit Tastenkombinationen kopiert. Zusätzlich ermöglicht die *Control*-Funktion das Ansteuern von Kontroll-Elementen wie z. B. der Pfadeingabe jeglicher Dropdown-Elemente oder einer Suchleiste. Jedoch ist diese nicht immer anwendbar, weshalb weiterhin die *Image-Search*-Funktion als Alternative genutzt wird.

Zur Auswahl der korrekten Zellen in der Excel-Tabelle wurde hierzu das Namensfeld benutzt. Dieses erlaubt die Navigation zu beliebigen Zellen, indem die Koordinaten dieser eingegeben werden. Standardmäßig werden in Excel Zeilen als Zahl und Spalten als Buchstaben dargestellt. Dies kann allerdings in den Einstellungen der Anwendung angepasst werden, um beispielsweise die Zelle ‚4E‘ mit ‚Z4S5‘ anzusprechen. Wird die Dimension der Tabelle notiert, müssen die Spalten also nicht von Zahlen zu Buchstaben umgewandelt werden, was das Programmieren vereinfacht.

Programmierung

Entwicklungszeit: 60 Stunden.

Vorgehensweise:

1. Preisliste aus dem Lieferantenordner öffnen.
2. Gesamte Preisliste mit *STRG + A* auswählen und mit *STRG + C* kopieren.
3. Preisliste schließen.
4. Auf Programmiererebene Berechnungen durchführen und die kopierte Tabelle formatieren.
5. Über die COM-Schnittstelle von Excel die Dateien in den Lieferantenordner im XLSX- und CSV-Format speichern.

Besonderheiten:

Ein entscheidender Unterschied gegenüber den anderen Ansätzen ist hier die Verarbeitung der gesamten Preisliste auf Codeebene. Wird die gesamte Tabelle kopiert, so sind einzelne Einträge durch Zeilenumbrüche getrennt. Sowohl Spalten als auch Zeilen werden dadurch getrennt, wodurch nicht differenziert werden kann, welche Dimension die Preisliste aufweist. Folglich muss die Anzahl an Spalten im Vorhinein festgelegt werden.

Das Kopieren der Preisliste resultiert oft in fehlerhaften Ausgaben, da offenbar die Zeilenumbruch-Trennzeichen länger benötigen, um nach dem Öffnen geladen zu werden. So wird an der entsprechenden Stelle auf den möglichen Abschluss dieser Aktion eine feste Zeitspanne lang gewartet.

Das endgültige Speichern der Tabelle in den benötigten Formaten wurde über die COM-Schnittstelle durchgeführt. Hiermit konnte direkt mit Excel durch Code kommuniziert werden, wobei die zugehörige Dokumentation zu den Funktionen von Excel besonders schwer zu navigieren ist und an ausreichender Information mangelt. So wurde viel Zeit nur an dem letzten Prozessschritt aufgebracht.

4.4 Übergreifende Probleme und Lösungsansätze

Abschließend werden Probleme der einzelnen RPA-Werkzeuge beschrieben, die alle Anwendungsfälle umfassen. Ebenso werden Lösungsvorschläge erläutert, sofern diese zur Verfügung stehen.

Macro Recorder v2.1+

Ein häufiger Prozessschritt vieler Aufgaben ist die Wiederholung anderer Prozessschritte, was durch simples Aufzeichnen von Aktionen nicht implementiert werden kann. Wie im ersten Anwendungsfall kann dies durch das Unterteilen des Gesamtprozesses in mehrere Makros gelöst werden. Hierdurch entsteht zwar ein zusätzlicher Programmieraufwand, jedoch können dadurch eine zusätzliche Menge von Aufgaben gelöst werden.

Maus- und Tastatureingaben sind an den momentanen Systemkontext gebunden und sind dadurch stark fehleranfällig. So kann der Kontext durch beispielsweise

- Änderungen an der Benutzeroberfläche von Anwendungen,
- zu langsames Laden von Benutzeroberflächen,
- das Öffnen von Anwendungsfenstern an leicht unterschiedlichen Stellen,
- die Reihenfolge von Dateien und Ordnern im Datei-Browser,
- die Position von Desktop-Symbolen sowie
- den Namen von Anwendungsfenstern

verändert werden. Hierzu liegt keine direkte Lösung vor, da viele dieser Optionen außerhalb der Kontrolle der Nutzer liegen. Es sollte jedoch ein Ausführungskontext für jeden Prozess notiert werden und dieser vor dem Ausführen geprüft werden, um die Chance auf Fehler zu minimieren.

Letztlich steigt die Wahrscheinlichkeit auf Fehler während der Aufnahmezeit, je länger und/oder je komplizierter ein Prozess ist. So kann dies durch das Aufzeichnen mehrerer Makros angegangen werden: Ein Prozess wird in Teilprozesse zerlegt und getrennt aufgezeichnet. Anschließend können diese manuell zu einem Skript kombiniert werden.

Pullover's Macro Creator & Programmierung

Beide Werkzeuge weisen dasselbe Problem auf: Die genutzten OCR-Funktionen operieren nicht zuverlässig. So wird teilweise Text nicht erkannt, da eine andere Schriftfarbe als erwartet vorliegt oder die Schriftgröße zu klein ist. Diesem kann entgegengewirkt werden, indem das eingelesene Bild vor dem Umwandeln zu Text durch zusätzliche Bildbearbeitungsschritte manipuliert wird. Das Bild kann vergrößert, der Kontrast erhöht und Rauschen reduziert werden. Diese Schritte können oft einfach mit frei verfügbaren Programmierbibliotheken durchgeführt werden.

5 Auswertungen und Ergebnisse

In diesem Kapitel soll die ausgearbeitete Information ausgewertet werden.

Mit Blick auf die Einschränkungen von KMU (Lee et al., 2010, S. 296), die im Unterabschnitt 2.2 näher erläutert wurden, soll untersucht werden, welcher der drei Ansätze für KMU am besten geeignet ist.

Es wird davon ausgegangen, dass die Mitarbeiter kein Vorwissen zu den drei Ansätzen besitzen. Die Beurteilung der einzelnen Anwendungen erfolgt auf Basis ihrer internen Funktionen, da stets die Option besteht, durch Programmierung Unterstützung zu erhalten.

Mit einer Nutzwertanalyse in Tabelle 1 soll eine Entscheidung getroffen werden. Dafür wurden folgende fünf Kriterien gewichtet.

Wissensaneignung (30 %)

Wie einfach es ist, das spezifische Wissen zu erlernen.

Angesichts des Mangels an IT-Personal bzw. Programmierkenntnissen ist die Fähigkeit, grundlegendes Wissen schnell und effizient zu erlernen, von entscheidender Bedeutung (Lee et al., 2010, S. 296).

Robustheit (25 %)

Die Zuverlässigkeit der Anwendung.

Für KMU ist es wichtig, dass die Lösung stabil und zuverlässig ist, um Ausfallzeiten zu minimieren (Narula, 2004, S. 155). Robuste Systeme sind weniger anfällig für Fehler, was insbesondere für Benutzer ohne technischen Hintergrund von Vorteil ist.

Wartbarkeit / Modifizierbarkeit (20 %)

Wie leicht es ist, Änderungen vorzunehmen.

Auch wenn KMU keine IT-Experten haben, ist es wahrscheinlich, dass diese in Zukunft Änderungen oder Anpassungen an der Software vornehmen möchten. Eine wartbare und modifizierbare Lösung ermöglicht es, sich den ändernden Anforderungen anzupassen, ohne Fachpersonal anschaffen zu müssen.

Entwicklungszeit (15 %)

Wie schnell ein Unterfangen umgesetzt werden kann.

Schnelligkeit bei der Implementierung kann für KMU von Bedeutung sein, insbesondere wenn Ressourcen begrenzt sind (Lee et al., 2010, S. 292). Eine Lösung, die relativ schnell einsatzbereit ist, könnte einen höheren Stellenwert haben.

Konfiguration (10 %)

Wie leicht die Einstellung vonstattengeht.

Die Möglichkeit, die Lösung einfach zu konfigurieren ist wichtig um Kosten für spezielles IT-Personal zu sparen. Dies ermöglicht eine schnellere Implementierung mit dem eigenen Personal.

Tabelle 1

Nutzwertanalyse der drei Ansätze für KMU

Kriterien	Gewichtung	Ansatz 1: Macro Recorder v2.1+		Ansatz 2: Pullover's Macro Creator		Ansatz 3: Programmierung (Visual Studio Code)	
		Punkte	Gewichtete Punkte	Punkte	Gewichtete Punkte	Punkte	Gewichtete Punkte
Wissensaneignung	0,3	5	1,5	3	0,9	1	0,3
Robustheit	0,25	1	0,25	3	0,75	5	1,25
Wartbarkeit / Modifizierbarkeit	0,2	1	0,2	4	0,8	5	1
Entwicklungszeit	0,15	3	0,45	4	0,6	1	0,15
Konfiguration	0,1	5	0,5	5	0,5	3	0,3
Gesamt	1	15	2,9	19	3,55	15	3

Erläuterung: Punkte von 1 bis 5 mit 5 als besten Wert.

Anmerkung zur Entwicklungszeit: Bei der Bewertung wurde ein Widerspruch festgestellt. Die Entwicklungszeit mit dem *Macro Recorder v2.1+* war im ersten Anwendungsfall kurz, da die Aufgabe wenig Schritte beinhaltete. Im Gegensatz dazu war die Entwicklungszeit im dritten Anwendungsfall sehr lang, bedingt durch die Länge und Anzahl der Aufgaben. Dadurch würde der erste Ansatz im ersten Anwendungsfall fünf Punkte und im dritten Fall nur einen Punkt für die Entwicklungszeit erhalten. Aus diesem Grund wurden drei Punkte vergeben.

Die Ergebnisse der Nutzwertanalyse in Tabelle 1 verdeutlichen, dass der *Pullover's Macro Creator* mit 3,55 gewichteten Punkten für KMU als besonders geeignet bewertet wurde. Dieser Ansatz schnitt in allen Kategorien durchschnittlich oder besser ab. Im Vergleich dazu erreichte die Programmierung mit 3 gewichteten Punkten den zweiten Platz, obwohl diese die höchsten Punkte bei Robustheit und Wartbarkeit bzw. Modifizierbarkeit erzielte. Aufgrund der niedrigen Punktzahlen in den Kategorien Wissensaneignung und Entwicklungszeit wird sie jedoch als weniger geeignet angesehen. Der *Macro Recorder v2.1+* landet mit 2,9 gewichteten Punkten auf dem letzten Platz. Dieser könnte besser abschneiden, wenn die Punkteverteilung nur für kleinere Aufgaben in Betracht gezogen werden würde. Um genauer zu sein, wäre dieser mit 5 Punkten in der Entwicklungszeit bei 3,2 gewichteten Punkten. Dabei ist es wichtig zu beachten, dass die Gewichtung je nach Unternehmen und deren spezifische Situation vermutlich anders ausfällt.

Der *Pullover's Macro Creator* zeigt sich als vielseitig und eignet sich sowohl für kleine als auch große Aufgaben. Dieser zeichnet sich durch eine vergleichsweise einfache

Erlernbarkeit aus, gestützt durch ausreichende Online-Ressourcen, auch wenn das vollständige Verständnis der Zusammenhänge einige Zeit in Anspruch nehmen kann. Die Robustheit und Sicherheit der erstellten Abläufe erfordern wenig Konfigurationsaufwand. Die intuitive Änderungsumsetzung und die klare Benennung von Abschnitten erleichtern das Auffinden der gewünschten Abschnitte. Die Entwicklungszeit ist bei entsprechender Robustheit schnell, was diesen zu einer idealen Wahl für Projekte unterschiedlicher Größe macht.

Im Kontrast dazu empfiehlt sich der *Macro Recorder v2.1+* eher für kleinere Aufgaben. Die schnelle und unkomplizierte Erlernbarkeit spricht für diesen, jedoch ist die generelle Zuverlässigkeit nicht gewährleistet und Änderungen gestalten sich umständlich. Die Entwicklungszeit steigt mit zunehmender Datensatzanzahl, insbesondere ohne Option von generalisierten Wiederholungen.

Die Programmierung hingegen eignet sich für größere Projekte, erfordert jedoch einen gewissen Konfigurationsaufwand. Die erstellten Abläufe zeichnen sich durch ihre Robustheit, Wartbarkeit und Anpassbarkeit aus. Trotz moderater Programmiererfahrung bleibt die Entwicklungszeit dieses Ansatzes entsprechend hoch. Es ist wichtig zu betonen, dass eine gewisse Fachkenntnis in der Programmierung unerlässlich ist, damit die Programmierung als robuste und leicht wartbare Option für KMU in Betracht gezogen werden kann.

Auf den ersten Blick mögen die Entwicklungszeiten mit den jeweiligen Ansätzen vergleichsweise hoch erscheinen, jedoch verringern sich diese mit zunehmender Erfahrung, da sowohl die Zeiten für Wissensaneignung als auch für die Implementierung abnehmen.

6 Zusammenfassung und Ausblick

Die vorliegende Bachelorarbeit widmet sich der Untersuchung von Potenzialen der Open Source Attended RPA für KMU in der Beschaffungslogistik und den spezifischen Herausforderungen, denen KMU gegenüberstehen. Das Hauptziel besteht darin, eine fundierte Entscheidungsgrundlage für die Auswahl von RPA-Tools zu schaffen, indem die Vor- und Nachteile von E-Procurement, RPA, Open Source und AHK eingehend beleuchtet.

Ein zentraler Aspekt der Arbeit ist die Entwicklung eines Ansatzes zur Identifizierung potenzieller RPA-Prozesse für KMU. Bedauerlicherweise hat sich dieser Ansatz als weniger hilfreich erwiesen, als zunächst angenommen. Statt bei der Identifikation von Prozessen zu unterstützen, fungiert dieser eher als Filtermechanismus auf Basis von Operatoren, was möglicherweise zu einer schnelleren, aber nicht unbedingt zielführenden Entscheidungsfindung führt. Die Entwicklungszeit und die unvollständige Überprüfung dieses Ansatzes erschweren die genaue Bestimmung der konkreten Vorteile.

Im Rahmen der Machbarkeitsprüfung mit AHK wurden drei Anwendungsfälle für KMU in der Beschaffungslogistik entwickelt und erfolgreich umgesetzt. Dabei wurde besonderes Augenmerk darauf gelegt, die spezifischen Herausforderungen von KMU angemessen zu berücksichtigen. Dennoch wurden während der Entwicklung wichtige Faktoren wie Kosten,

Skalierbarkeit und Langlebigkeit außer Acht gelassen, was ein vielversprechendes Gebiet für zukünftige Forschung darstellt.

Die Arbeit präsentiert drei unterschiedliche Umsetzungsansätze und führt eine Nutzwertanalyse durch, um den optimalen Ansatz für KMU zu ermitteln. Die Analyse integriert die spezifischen Herausforderungen von KMU sowie die limitierenden Faktoren der verschiedenen Ansätze, die wiederum die Bewertungskriterien in der Nutzwertanalyse beeinflussten. Die Ergebnisse der Nutzwertanalyse deuten darauf hin, dass der *Pullover's Macro Creator* die optimale Wahl für KMU darstellt, insbesondere wenn es um die Bewältigung vielfältiger Aufgaben in unterschiedlicher Größe geht. Hingegen zeigt der *Macro Recorder v2.1+* starke Limitationen und eignet sich besser für kleinere Aufgaben. Die Programmierung erreicht zwar den zweiten Platz, in der Nutzwertanalyse wird jedoch KMU nicht empfohlen, vor allem wenn die erforderlichen Ressourcen wie qualifiziertes Fachpersonal nicht verfügbar sind.

Im Ausblick auf künftige Entwicklungen und Forschungsbemühungen könnte eine Weiterführung der vorliegenden Arbeit verschiedene Dimensionen erkunden. Zunächst wäre es sinnvoll, den entwickelten Ansatz zur Identifizierung von RPA-Prozessen für KMU weiter zu verfeinern und die Anwendbarkeit unter realen Bedingungen zu validieren.

Ein weiterer Punkt wäre die Weiterentwicklung von AHK für die Bedürfnisse von KMU. Dies könnte die Integration bisher unbeachteter Faktoren wie Kosten, Skalierbarkeit und Langlebigkeit umfassen, um eine ganzheitlichere Bewertung der Anwendung im Unternehmenskontext zu ermöglichen.

Abschließend könnte die Erweiterung der Nutzwertanalyse um weitere Kriterien, die speziell auf die Anforderungen von KMU zugeschnitten sind, dazu beitragen, die Auswahl des optimalen RPA-Tools noch präziser zu gestalten. Hierbei wären eine kontinuierliche Evaluierung und Anpassung der Bewertungskriterien an die sich verändernden Bedürfnisse und Technologien von KMU von entscheidender Bedeutung.

Literaturverzeichnis

- "Descolada OCR" [GitHub]. (n. d.). Verfügbar unter: <https://github.com/Descolada/OCR>
- Amtsblatt der Europäischen Union, L124 vom 25. Mai 2003. (2003). Amtsblatt, L 124, 36–41.
- Angeles, R. & Nath, R. (2007). Business-to-business e-procurement: success factors and challenges to implementation. *Supply Chain Management: An International Journal*, 12(2), 104–115. <https://doi.org/10.1108/13598540710737299>
- AutoHotkey Foundation LLC. (n. d.a). Verfügbar unter: <https://www.autohotkey.com/>
- AutoHotkey Foundation LLC. (n. d.b). *Changelog for Years 2003-2004*. Verfügbar unter: <https://web.archive.org/web/20111104142142/http://www.autohotkey.com/changelog/2004.htm>
- AutoHotkey Foundation LLC. (n. d.c). *Macro Recorder v2.1+*. Verfügbar unter: https://www.autohotkey.com/boards/viewtopic.php?t=34184#post_content158990
- AutoHotkey Foundation LLC. (n. d.d). *Software License*. Verfügbar unter: <https://www.autohotkey.com/docs/v2/license.htm>
- Batista, R. U. (n. d.). *Pullover's Macro Creator*. Verfügbar unter: <https://www.macrocreator.com/>
- Bohli, J.-M., Kaspar, F. & Westhoff, D. (Hrsg.). (2014). *Software-technologien und -prozesse. IT-sicherheit & privatheit in zeiten von big data tagungsband proceedings zur 4. konferenz STeP 2014, hochschule furtwangen, 20. mai 2014*. München Deutschland: De Gruyter Oldenbourg.
- Erdmann, S. & Sandkuhl, K. (2023). Robotic Process Automation in Small Enterprises: An Investigation into Application Potential. *Complex Systems Informatics and Modeling Quarterly*, 0(34), 84–105. <https://doi.org/10.7250/csimq.2023-34.04>
- Fersht, P. & Slaby, J. R. (2012). Robotic automation emerges as a threat to traditional low-cost outsourcing. *Horses for Sources, London*, 1, 18.
- Fung, H. P. (2014). Criteria, use cases and effects of information technology process automation (ITPA). *Advances in Robotics & Automation*, 3.
- Gupta, H. & Barua, M. K. (2018). Modelling cause and effect relationship among enablers of innovation in SMEs. *Benchmarking: An International Journal*, 25(5), 1597–1622. <https://doi.org/10.1108/BIJ-03-2017-0050>
- Hausladen, I. (2020). Prozesse und Anwendungen einer IT-gestützten Logistik. In I. Hausladen (Hrsg.), *IT-gestützte Logistik* (S. 95–250). Wiesbaden: Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-31260-2_4
- Hausman, A. (2005). Innovativeness among small businesses: Theory and propositions for future research. *Industrial Marketing Management*, 34(8), 773–782. <https://doi.org/10.1016/j.indmarman.2004.12.009>

- Hüller, L., Jenß, K. E., Seph, S., Woelki, D., Völker, M. & Weske, M. *Ark Automate – an Open-Source Platform for Robotic Process Automation*. Vortrag anlässlich Proceedings of the Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Track at BPM 2021. Verfügbar unter: https://www.researchgate.net/profile/maximilian-voelker-2/publication/371539287_ark_automate_-_an_open-source_platform_for_robotic_process_automation
- Hwang, M.-H., Na, U.-K., Lee, S. [Seungjun], Cho, B., Kim, Y., Lee, D. et al. (2020). MIORPA: Middleware System for Open-Source Robotic Process Automation. *Journal of Computing Science and Engineering*, 14(1), 19–25. <https://doi.org/10.5626/JCSE.2020.14.1.19>
- Institut für Mittelstandsforschung. (2016). *KMU-Definition des IfM Bonn*. Verfügbar unter: <https://www.ifm-bonn.org/definitionen-/kmu-definition-des-ifm-bonn>
- Koch, C. & Fedtke, S. (2020). *Robotic Process Automation. Ein Leitfaden für Führungskräfte zur erfolgreichen Einführung und Betrieb von Software-Robots im Unternehmen*. Berlin, Heidelberg: Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-662-61178-4>
- Kollmann, T. (2020). *Handbuch Digitale Wirtschaft*. Wiesbaden: Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-17291-6>
- Krakau, J., Feldmann, C. & Kaupe, V. (Hrsg.). (2021). *Robotic process automation in logistics: Implementation model and factors of success* (Proceedings of the Hamburg International Conference of Logistics (HICL), Bd. 32, 14. Auflage). Berlin: epubli GmbH.
- Lacity, M., Willcocks, L. P. & Craig, A. (2015). *Robotic process automation at Telefónica O2*. LSE Research Online Documents on Economics. London School of Economics and Political Science, LSE Library. Verfügbar unter: <https://EconPapers.repec.org/RePEc:ehl:lserod:64516>
- Langmann, C. & Turi, D. (2022). *Robotic Process Automation (RPA) - Digitization and Automation of Processes. Prerequisites, functionality and implementation using accounting as an example*. Wiesbaden: Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-38692-4>
- Lee, S. [Sungjoo], Park, G., Yoon, B. & Park, J. (2010). Open innovation in SMEs—An intermediated network model. *Research Policy*, 39(2), 290–300. <https://doi.org/10.1016/j.respol.2009.12.009>
- Lhuer, X. (2016). The next acronym you need to know about: RPA (robotic process automation).
- Mostaquim, M. E. (2022). *Study on open source tool for robotic process automation (RPA) with respect to commercial software for small and medium sized enterprises (SMEs) in logistics field*. Master's Thesis. Technische Hochschule Ingolstadt. Verfügbar unter: <http://nbn-resolving.de/urn:nbn:de:bvb:573-35787>

- Narula, R. (2004). R&D collaboration by SMEs: new opportunities and limitations in the face of globalisation. *Technovation*, 24(2), 153–161. [https://doi.org/10.1016/S0166-4972\(02\)00045-7](https://doi.org/10.1016/S0166-4972(02)00045-7)
- Open Source Initiative. (2023, 22. Februar). Verfügbar unter: <https://opensource.org/osd/>
- Ramkumar, R., Satsish, R., Chetan, S., Patnana, V. K. & Gulve, R. (2021). OPEN-SOURCE ROBOTIC PROCESS AUTOMATION–A CASE STUDY WITH ROBOCORP, 6(1).
- Rik Donckels & Erwin Fröhlich. (1991). Are Family Businesses Really Different? European Experiences from STRATOS. *Family Business Review*, 4(2), 149–160. <https://doi.org/10.1111/j.1741-6248.1991.00149.x>
- Sánchez-Rodríguez, C., Martínez-Lorente, A. R. & Hemsworth, D. (2020). E-procurement in small and medium sized enterprises; facilitators, obstacles and effect on performance. *Benchmarking: An International Journal*, 27(2), 839–866. <https://doi.org/10.1108/BIJ-12-2018-0413>
- Scheer, A. W. (2017). Performancesteigerung durch Automatisierung von Geschäftsprozessen. Saarbrücken: AWSi. https://www.aws-institut.de/wp-content/uploads/2017/11/031117_GPPPerformance_44seiten_final_300dpi_2Aufl_einzel.pdf.
- Shin, J., Lee, S. [Sungho], Seo, H., Im, J.-S., Cho, H. & Hwang, M.-H. (2023). EnterRPA: Open-Source Robotic Process Automation for Enterprise. *The Transactions of The Korean Institute of Electrical Engineers*, 72(3), 419–427. <https://doi.org/10.5370/KIEE.2023.72.3.419>
- Slonneger, K. & Kurtz, B. L. (1995). *Formal syntax and semantics of programming languages. A laboratory based approach*. Reading Mass.: Addison-Wesley Pub. Co.
- Spath, D., Schimpf, S. & Kugler, A. (2007). Webbasierte Open Source-Kollaborationsplattformen. *Studie der Fraunhofer-Gesellschaft, Institut für Arbeitswirtschaft und Organisation, Stuttgart*.
- Statista. (2022a). *Estimated number of large companies (250+ employees) worldwide from 2000 to 2021*. Verfügbar unter: <https://www.statista.com/statistics/1261035/large-global-companies/>
- Statista. (2022b). *Estimated number of small and medium sized enterprises (SMEs) worldwide from 2000 to 2021*. Verfügbar unter: <https://www.statista.com/statistics/1261592/global-smes/>
- Statista. (2023). *Umfrage zu Automatisierungslösungen in Unternehmen in Deutschland 2023*. Verfügbar unter: <https://de.statista.com/statistik/daten/studie/1373604/umfrage/automatisierung-prozesse-in-unternehmen/>
- Sutherland, C. (2013). Framing a constitution for robotistan. *Hfs Research, ottobre*.

- Taulli, T. (2020). *The Robotic Process Automation Handbook*. Berkeley, CA: Apress. <https://doi.org/10.1007/978-1-4842-5729-6>
- Teunissen, T. (2019). *Success factors for RPA application in small and medium sized enterprises* (S. 1–10). Verfügbar unter: <http://essay.utwente.nl/77592/>
- Turban, E., Outland, J., King, D., Lee, J. K., Liang, T.-P. & Turban, D. C. (2018). *Electronic Commerce 2018. A Managerial and Social Networks Perspective*. Cham: Springer International Publishing. <https://doi.org/10.1007/978-3-319-58715-8>
- Willcocks, L. P., Lacity, M. & Craig, A. (2015). The IT function and robotic process automation, 38.

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit eigenständig und ohne fremde Hilfe angefertigt habe. Textpassagen, die wörtlich oder dem Sinn nach auf Publikationen oder Vorträgen anderer Autoren beruhen, sind als solche kenntlich gemacht. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Merseburg, 18.12.2023

Viktor Neumann