

RPA-gestützte Datenmigration in der Praxis

Bastian Engelking, viadee AG

Datenmigrationen sind meist einmalige Projekte und bedürfen daher individueller Strategien, um Daten zuverlässig und technisch sicher vom Quell- ins Zielsystem zu überführen. Abhängig von der jeweiligen Situation kann der Einsatz von RPA (Robotic Process Automation) und entsprechenden Tools ein Baustein in der Migrationsstrategie sein. In diesem Artikel erklären wir anhand eines beispielhaften Projekts aus der Energiebranche, wann der Einsatz sinnvoll ist und wie Daten mithilfe von RPA migriert werden können. In diesem Zuge werden ein in Java geschriebenes und auf Open-Source-Bibliotheken basierendes RPA-Tool und seine unterschiedlichen Automatisierungstreiber vorgestellt, sodass die Leser/innen ein vollumfassendes Bild über die Technologie erhalten.





Status Quo

RPA ist inzwischen eine etablierte Technologie, um repetitive und strukturierte Arbeitsschritte in Unternehmen zu automatisieren. Der Einsatz von RPA erfolgt häufig dort, wo Schnittstellen fehlen oder es sich nicht mehr lohnt, Systeme weiterzuentwickeln. Dabei ist der Einsatz von RPA zumeist keine langfristige Lösung. Allerdings lassen sich RPA-Lösungen verhältnismäßig schnell umsetzen und flexibel an eine sich ständig verändernde Systemumgebung anpassen. Damit hat RPA seine Daseinsberechtigung in der IT-Landschaft heutiger Unternehmen. Auch in Migrationsprojekten kann RPA eingesetzt werden, wie das im Folgenden vorgestellte Beispielprojekt verdeutlicht.

Migrationsszenario aus der Energiebranche

Als Datenmigration wird ein Vorhaben bezeichnet, das die Übertragung von Daten aus einem System in ein anderes zum Ziel hat. Es gibt meist viele Gründe für ein solches Vorhaben. Häufig führen Fusionen und Unternehmensübernahmen zur Konsolidierung mehrerer Datenbestände.

In dem hier gezeigten Beispielprojekt ist es genau andersherum. Aufgrund gesetzlicher Vorgaben im Zusammenhang mit der Fusion zweier Energieversorger musste ein Geschäftsbereich ausgegliedert werden. Der gesamte Datenbestand für mehrere hunderttausend Kunden musste daher in ein neues System überführt werden. Dies ist eine Größenordnung, für die RPA vermutlich nicht das geeignete Mittel der Wahl ist. Die Migration der Daten von Kunden und Vertragspartnern wurde auf Datenbankebene durchgeführt. Es wurden komplexe ETL-Prozesse (extract, transform, load) entwickelt, um große Datenmengen in kurzer Zeit zu migrieren. Ein solches Projekt bindet viele Ressourcen sowie Know-how und ist mit einigen Risiken verbunden.

Im Beispielprojekt war das Projektmanagement vorrausschauend und hat frühzeitig erkannt, dass es bei der Migration einer bestimmten Entität zu Engpässen kommen würde. Bei den Entitäten handelte es sich um die Datenbestände der Geschäftspartner die als Zahler, Zahlungs- oder Mitteilungsempfänger für andere Kunden agierten. Mengenmäßig handelte es sich hierbei um mehrere tausend Geschäftspartner und damit um eine hinreichend kleine Menge, die mit RPA in einem engen Zeitfenster migriert werden könnte. Das Ziel des Projektes war es, diese Geschäftspartner mit allen Stammdaten aus dem Quellsystem über die Benutzeroberfläche in das Zielsystem zu überführen und wieder den zuvor migrierten Kunden zuzuordnen. Was sich nach einem vielversprechenden Vorhaben anhört, musste jedoch wohl bedacht werden.

Entscheidung für eine RPA-gestützte Migration

Bei der Entscheidung für den Einsatz eines RPA-Tools zur Datenmigration wurden daher die Vor- und Nachteile gegeneinander abgewogen und mit einer klassischen Migration auf Datenbankebene verglichen.

RPA-Tools haben den Vorteil, dass sie einem Low-Code-Ansatz folgen, wodurch in der Regel weniger Programmieraufwand anfällt. Die gängigen Tools bringen out of the box vordefinierte und wiederverwendbare Funktionen mit, sodass Arbeitsabläufe effizient und in kurzer Zeit automatisiert werden können. Der Einsatz solcher Tools ist insbesondere dann lohnenswert, wenn RPA-Entwickler/

innen verfügbar sind und gleichzeitig das Personal im Datenbank- und Entwicklungsumfeld knapp ist. Sind die zu automatisierenden Abläufe erst mal konzeptionell entworfen und implementiert, kann mit der richtigen Infrastruktur und durch den Einsatz mehrerer Instanzen des RPA-Tools die Anzahl gleichzeitiger Prozessdurchläufe skaliert werden. Zudem können die Instanzen theoretisch rund um die Uhr ohne Unterbrechung betrieben werden, wenn dies die automatisierten Systeme zulassen und es beispielweise keine Tages- oder Nachtverarbeitungen gibt. Hierbei handelt es sich um ein Risiko, das durch die Geschwindigkeit bei einer Migration auf Datenbankebene vermieden werden könnte.

Beim Faktor Datenqualität können RPA-Tools Vorteile gegenüber einer Migration auf Datenbankebene aufweisen. Beispielweise kann bei der Migration auf die bestehenden Regeln und Prüfroutinen des Zielsystems zurückgegriffen werden, um diese zur Qualitätssicherung zu nutzen. Der Großteil der Validierungsregeln eines Systems ist in die Benutzeroberfläche und den Code integriert. Bei der Eingabe unvollständiger oder fehlerhafter Daten über die Benutzeroberfläche verweigert das System die Eingabe. Diese Daten können ausgesteuert und anschließend bereinigt werden. Bei einer ETL-basierten Migration müssten Validierungsregeln zunächst nachgebaut werden oder es würde in Kauf genommen, dass fehlerhafte Daten direkt in die Datenbanken geschrieben werden. Die Nutzung bestehender Workflows und Eingabemasken des Systems hat den weiteren Vorteil, dass auf der bestehenden Geschäftslogik aufgebaut werden kann. Dies führt auch dazu, dass weniger systemspezifisches Know-how benötigt wird und wiederum knappe Ressourcen geschont werden können.

Der Einsatz von RPA kann jedoch auch Nachteile mit sich bringen. Es müssen geeignete organisatorische Sicherheits- und Betriebskonzepte erarbeitet werden, um ein zuverlässiges Abarbeiten der automatisierten Abläufe zu gewährleisten. Häufig gibt es diese Konzepte bereits für die Datenbankentwicklung und entsprechende ETL-Tools, nicht aber für RPA-Tools. Des Weiteren besitzen RPA-Tools in der Regel keine ausgefeilten Datenanalysefunktionen wie ETL-Tools, wodurch Datenbereinigungsaktivitäten und die Nutzung temporärer Datenbanken für die Migration erschwert werden. Gute RPA-Tools sollten jedoch Datenbanktreiber mitbringen, was bei einer Toolauswahl berücksichtigt werden sollte. Das wichtigste Argument, das gegen den Einsatz von RPA-Tools spricht, ist vermutlich die Datenmenge. Bei großen Datenmengen, die über die Benutzeroberfläche migriert werden sollen, kommt RPA an seine Grenzen, da sehr viele Instanzen benötigt werden, die betrieben und überwacht werden müssen.

In dem vorgestellten Beispielprojekt haben die Vorteile von RPA zur Migration der genannten Entität überwogen. Die Verantwortlichen waren sich einig, dass RPA das Ziel einer fachlich richtigen, technisch sicheren und zuverlässigen Datenmigration optimal unterstützt. Somit wurde RPA ein Baustein in der Migrationsstrategie.

Verwendete Automatisierungstreiber

Für die Migration wurde das in Java geschriebene und auf Open-Source-Bibliotheken basierende Tool „mateo rpa“ verwendet. Das Tool besitzt eine offene und erweiterbare Architektur, die es ermöglicht, verschiedene Automatisierungstreiber zu integrieren (siehe *Abbildung 1*). Angelehnt an die generische Testautomatisierungs-

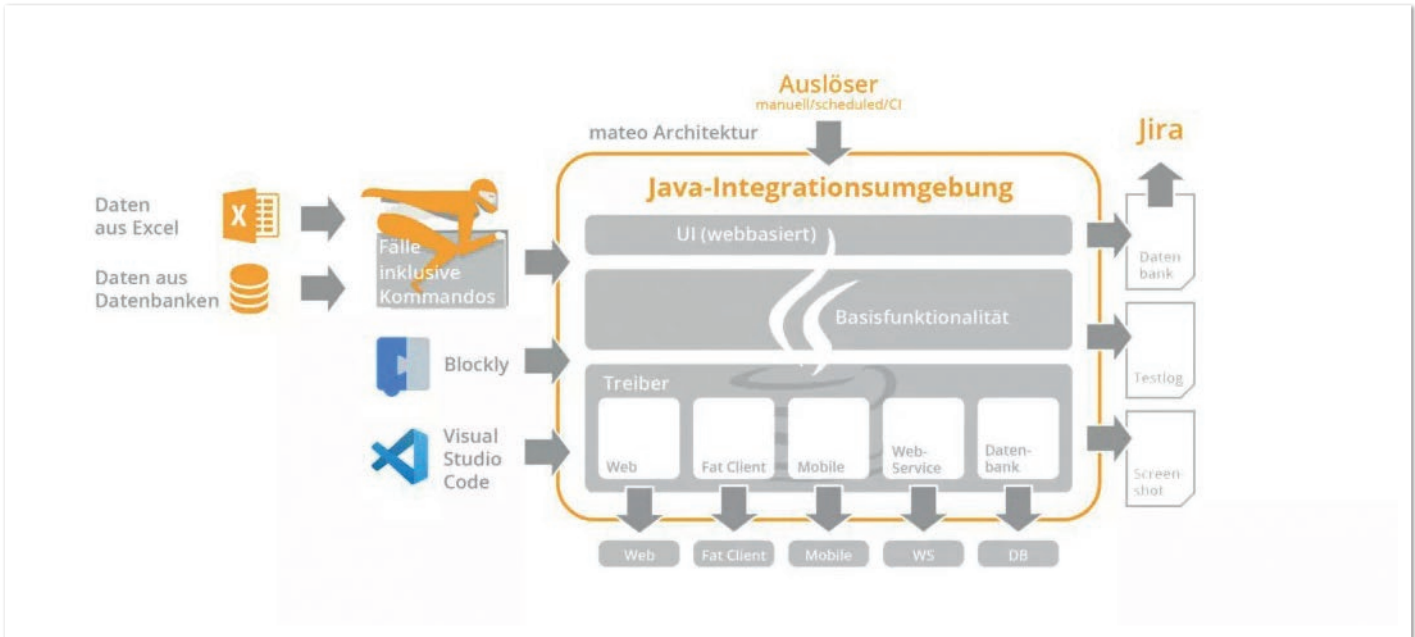


Abbildung 1: Architektur mateo rpa (© viadee Unternehmensberatung AG 2021)

Unter der Basisfunktionalität sind Funktionen zu verstehen, mit denen der Kontrollfluss der automatisierten Arbeitsabläufe abgebildet wird. Es können Bedingungen, Schleifen oder der Vergleich von Werten umgesetzt werden (siehe Listing 1).

```

1 while(MAX_ITERATIONS = "10"):
2     lessThanNum(LEFT = "{index}", RIGHT = "8")
3 do:
4     if():
5         lessThanNum(LEFT = "{index}", RIGHT = "4")
6     then:
7         success(MESSAGE = "Der Index ist kleiner 4.")
8     else:
9         success(MESSAGE = "Der Index ist größer gleich 4.")

```

Listing 1: While-Schleife, Bedingungen und Wertevergleich in mateo rpa

architektur nach ISTQB [1] werden Automatisierungstreiber der Testadaptierungsschicht zugeordnet und trennen das zu steuernde System von der im RPA-Skript definierten Logik. Ein Automatisierungstreiber ist daher ein Software-Modul innerhalb des Tools, das es ermöglicht, die zu steuernde Anwendung anzusprechen und mit dieser zu interagieren. Für die Migration wurde neben der Basisfunktionalität des Tools auf Automatisierungstreiber für Web, SAP und Datenbanken zurückgegriffen.

Der verwendete Web-Treiber basiert auf Selenium [2], einem weitverbreiteten und quelloffenen Framework, das sich bei der Testautomatisierung von Webanwendungen bewährt hat, aber auch für RPA-Szenarien verwendet werden kann. Selenium kann mit verschiedenen Programmiersprachen (zum Beispiel Java, Python, C#) genutzt oder, wie im Fall von mateo, als Automatisierungstreiber in ein umfassendes RPA-Tool integriert werden. Diese Einbettung bringt verschiedene Vorteile für die Entwicklungszeit und Qualität der Skripte mit sich. Insbesondere vordefinierte Wrapper-Funktionen, die ansonsten initial entwickelt werden müssten, beschleunigen die Entwicklung. Als Beispiel kann hier die Funktionalität zum Warten auf die Klickbarkeit eines Elements vor dem tatsächlichen Klick auf das Element genannt werden. RPA-Tools besitzen hierfür vordefinierte Funktionen wie *clickWeb*, das einen impliziten Warte-schritt vor dem tatsächlichen Klick des Elements einschließt.

Bei dem zu automatisierenden SAP-System handelte es sich um eine Version, die über eine Windows-Oberfläche gesteuert werden musste. Hierzu bringt das RPA-Tool einen auf Autolt [3] basierenden SAP-Treiber mit. Autolt ist eine leistungsfähige, frei verfügbare Skriptsprache mit BASIC-ähnlicher Syntax. Sie ist in erster Linie für die Automatisierung von Operationen auf der Windows-GUI konzipiert. Autolt bietet eine COM-Schnittstelle, die es ermöglicht, das SAP-GUI-Scripting-API in Autolt zu verwenden. Mit diesem API können Benutzerinteraktionen mit SAP imitiert und somit ganze Arbeitsabläufe und Prozesse automatisiert werden. Analog zum Web-Treiber stellt das RPA-Tool mittels Autolt vordefinierte Funktionen bereit. Mit der Funktion *sapObjSelect* können beispielweise Objekte selektiert werden. Dass Autolt im Hintergrund zur Interaktion mit SAP genutzt wird, fällt nicht auf.

Um die exportierten Stammdaten sicher zwischenspeichern, wurde auf eine temporäre Datenbank gesetzt. Der Datenbank-Treiber des verwendeten Tools basiert auf JDBC (Java Database Connectivity), der Standardschnittstelle für datenbankunabhängige Verbindungen zwischen Java und einer Vielzahl von Datenbankverwaltungssystemen wie MySQL, Oracle und anderen. Damit lassen sich Datenbankverbindungen herstellen und SQL-Anweisungen oder -Abfragen ausführen. Das RPA-Tool hält hierzu Funktionen wie *querySql* bereit, um Datenbankabfragen auszuführen. Über einen

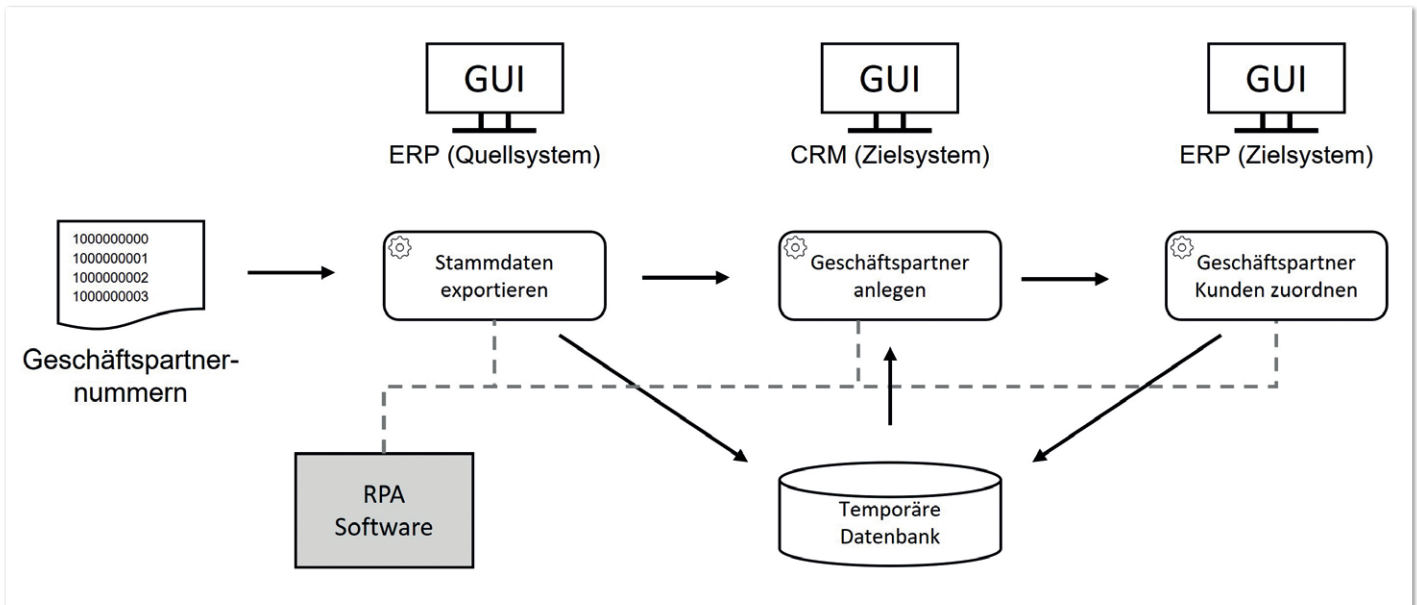


Abbildung 2: RPA-gestützter Migrationsablauf

Parameter wird die konkrete SQL-Abfrage an die Funktion übergeben. Die Rückgabewerte der Abfrage stehen im JSON-Format zur Verfügung. Das Parsen und die Weiterverarbeitung einzelner Werte lassen sich mit der Funktion *query/json* durchführen.

Es kann festgehalten werden, dass mit einem reifen RPA-Tool verschiedenste Anwendungen gesteuert werden können, indem Automatisierungstreiber geschickt integriert werden. Für die Funktionen der verschiedenen Treiber entstehen im RPA-Tool neue Wrapper-Funktionen und Funktionsnamen. Hierdurch entsteht eine neue anwendungsspezifische Sprache (Domain-specific Language, kurz DSL). Jedes RPA-Tool besitzt seine eigene DSL zur Automatisierung von Prozessen. Zum Schreiben der DSL gibt es jedoch unterschiedliche Ansätze. Einige Tools stellen einen Editor mit einer grafischen Benutzeroberfläche bereit, um Funktionen in Form von Symbolen via Drag-and-Drop in einem Prozessdiagramm zusammenzustellen. Andere Tools sind weniger grafisch und nutzen Texteditoren wie Visual Studio Code.

Für erfahrene RPA-Entwickler/innen, die regelmäßig Prozesse automatisieren, ist letztere Variante in der Regel das Mittel der Wahl, da die Entwicklungsgeschwindigkeit hiermit deutlich erhöht werden kann. Die Nutzung eines RPA-Tools hat den Vorteil, dass nur eine Tool-spezifische Sprache geschrieben werden muss, anstatt dass mehrere kleinteilige Prozesse in unterschiedlichen Sprachen und Bibliotheken umgesetzt werden müssten.

RPA-gestützter Prozess

Der RPA-gestützte Migrationsablauf bestand im Wesentlichen aus drei Prozessschritten: Stammdaten exportieren, Geschäftspartner im neuen System anlegen und Geschäftspartner den Kunden zurechnen.

Für jeden der Prozessschritte gab es ein Skript. Zusätzlich war es notwendig, Skripte für Zwischenschritte zu entwickeln, beispielsweise um Daten zusammenzuführen und Teilmengen dieser auf die unterschiedlichen RPA-Instanzen zu verteilen. Zum Datenexport wurde der SAP-Treiber verwendet. Es wurden verschiedene Stammdaten-

Tabellen heruntergeladen, in eine temporäre Datenbank importiert, bereinigt und zu einem neuen Datenmodell zusammengefügt. Das Herz der Datenbank war eine Steuerungstabelle (siehe Tabelle 1), in der die Geschäftspartner-Kunden-Zuordnung vorgehalten und über die der Bearbeitungsstatus eines jeden Datensatzes nachvollzogen werden konnte.

Die Anlage der Geschäftspartner im neuen System erfolgte über das webbasierte Kundenmanagement-System. Das Skript zur Anlage von Kunden war so aufgebaut, dass pro Ausführung genau ein Kunde angelegt wurde. Das RPA-Tool hat in einer großen Schleife das gleiche Skript mehrere Tausend Mal verteilt über sechs Instanzen ausgeführt. Gleiches galt für den letzten Prozessschritt. Die Zuordnung von Geschäftspartnern zu Kunden erfolgte, nachdem alle Geschäftspartner angelegt worden waren. Dies war zugleich der Prozessschritt mit den meisten Ausführungen pro Skript, da ein Geschäftspartner mehreren Kunden zugeordnet war und dabei unterschiedliche Rollen einnehmen konnte. In der Steuerungstabelle waren die verschiedenen Kombinationsmöglichkeiten ersichtlich. Ein Datensatz entsprach der einmaligen Ausführung eines Skriptes. In der Steuerungstabelle wurde die Spalte *Status* als Kontrollmerkmal angelegt. Mithilfe der Spalte wurden die doppelte Verarbeitung einzelner Datensätze vermieden, die Rückverfolgbarkeit fehlerhafter Ausführungen sichergestellt und der Bearbeitungsfortschritt überwacht.

Herausforderung Datenqualität

Das Thema Datenqualität hatte einen großen Stellenwert bei der Migration. Bereits während der RPA-Entwicklung auf den Testsystemen wurde festgestellt, dass einige Datensätze nicht mehr den aktuellen Anforderungen entsprachen und in ihrer derzeitigen Form gar nicht migriert werden konnten. Für diese Fälle wurde vorab definiert, wie im Zuge der Migration mit diesen umzugehen sei.

Beispiel: Bei der Anlage eines neuen Kunden im Kundenmanagement-System gab es ein Dropdown-Menü zur Selektion der korrekten Anredeform. Die Auswahl der zur Verfügung stehenden Anredeformen hatte sich jedoch im Laufe der Zeit verändert. Im

ID	Geschäftspartner	Kunde	Rolle	Status
1	1000000000	3000000000	Zahlungsempfänger	Zugeordnet
2	1000000000	3000000000	Zahler	Zugeordnet
3	1000000000	3000000000	Mitteilungsempfänger	Error
4	1000000000	4000000000	Mitteilungsempfänger	Zugeordnet
5	1000000000	4000000000	Zahlungsempfänger	Bearbeitung
6	2000000000	5000000000	Zahlungsempfänger	Offen

Tabelle 1: Steuerungstabelle

Kundenmanagement-System wurde die Anrede „Lebensgemeinschaft“ nicht mehr unterstützt. Im SAP-System wurde diese Anredeform jedoch weiterhin geführt. In diesem Fall wurde definiert, dass aus der Anrede „Lebensgemeinschaft“ die Anrede „Herrn/Frau“ werden sollte.

Trotz umfänglicher Tests und Datenanalysen im Vorfeld der Migration zeigte sich auch während der Migration, dass die zu migrierenden Daten an einigen Stellen nicht immer der Erwartungshaltung entsprachen.

Beispiel: Die Migration der Kundenstammdaten beinhaltete unter anderem die Übertragung der E-Mail-Adresse des Kunden. Ausgehend von den Anforderungen wurde erwartet, dass jedem Kunden genau eine E-Mail-Adresse zugeordnet ist. Im Kundenmanagement-System gab es ein entsprechendes Textfeld, in das diese eingetragen werden sollte. Nachdem alle zu migrierenden Daten ausgelesen worden waren, zeigte sich, dass in etwa 0,1 Prozent der Fälle zwei E-Mail-Adressen zu einem Kunden hinterlegt worden waren. Welche E-Mail-Adresse sollte nun migriert werden?

Da diese Möglichkeit während der Anforderungsanalyse und der Entwicklung der Skripte nicht bedacht worden war, wurden die entsprechenden Datensätze aussortiert und an die Fachabteilung weitergeleitet. Im Nachhinein stellte sich heraus, dass es in der Vergangenheit Überlegungen zur Hinterlegung einer zweiten E-Mail-Adresse gegeben hatte und dass die in diesem Kontext teilweise modifizierten Stammdaten nicht wieder bereinigt worden waren. Diese simplen Beispiele zeigen, dass auch eine RPA-gestützte Migration viel Vorbereitung bedarf und dennoch mit Risiken verbunden ist, auf die flexibel reagiert werden muss.

Fazit

An dem vorgestellten Beispielprojekt zeigt sich, dass sich RPA als Technologie zur Datenmigration in bestimmten Fällen eignet und zu einer erfolgreichen Migration beitragen kann. Es können komplexe, anwendungsübergreifende Prozesse durch die Integration verschiedenster Automatisierungstreiber realisiert werden, wobei nur eine Tool-spezifische Sprache geschrieben werden muss. Gleichzeitig darf der Realisierungsaufwand nicht unterschätzt werden. Die Entscheidung für oder gegen die Aufnahme von RPA in die Migrationsstrategie wird wohl immer vom Einzelfall sowie von den verfügbaren Ressourcen und vorhandenen Tools abhängen.

Letztlich birgt eine RPA-gestützte Datenmigration die gleichen Herausforderungen wie eine klassische Migration. Selbst, wenn auf die

Prüfroutinen des Zielsystems zur Qualitätssicherung gebaut wird, müssen die Datenbestände vorab analysiert und Regeln definiert werden, wie mit Abweichungen umzugehen ist. Anforderungsanalysen und Migrationskonzepte sind hierfür die Grundlage.

Quellen

- [1] German Testing Board (2019): *Certified Tester Advanced Level Syllabus Testautomatisierungsentwickler*.
- [2] <https://selenium.dev>
- [3] <https://autoitscript.com>



Bastian Engelking

viadee Unternehmensberatung AG
bastian.engelking@viadee.de

Bastian Engelking ist seit 2019 als IT-Berater für die viadee Unternehmensberatung AG in den Branchen Energiewirtschaft, Versicherungen und Banken unterwegs. Seine Einsatzschwerpunkte liegen in der Business- und Anforderungsanalyse mit Fokus auf Migrationsprojekte. Seine Begeisterung für Automatisierungstechnologien führen ihn immer wieder zur Entwicklung mit RPA- und Testautomatisierungs-lösungen zurück.