

Konzeption und prototypische Implementierung eines Frameworks für den Einsatz von Robotic Process Automation zur Testautomatisierung

Jürgen Seisl BA MA und Dipl.-Ing. Dr. Igor Miladinovic, Ferdinand Porsche FernFH

Abstract: Da mit Robotic Process Automation (RPA) menschliche Interaktionen mit einem System nachgeahmt werden können, kann RPA für die Testautomatisierung im End-to-End Bereich eingesetzt werden. RPA ist aber kein Tool, welches für die Testautomatisierung erstellt wurde und somit fehlen RPA-Tools wichtige Funktionalitäten damit diese als primäre Testtools eingesetzt werden können. Aus dieser Problemstellung heraus resultiert meine Forschungsfrage:

„Wie muss ein Framework für den Einsatz von Robotic Process Automation Tools zur Testautomatisierung bei Regressionstests im End-to-End-Bereich konzipiert sein, damit die Steuerung der einzelnen Test-Cases in Form von RPA-Prozessen sowie ein Test-Reporting der Ergebnisse ermöglicht wird?“

Für die Beantwortung wurde die Methode des Design Science Research nach Österle et al. gewählt. In den vier iterativen Schritten, Analyse, Entwurf, Evaluierung und Diffusion, war es möglich, einen funktionalen Prototyp des Frameworks zu entwickeln. Bei der Evaluierung des Prototyps wurde festgestellt, dass die Umsetzung der Prozesse der Testautomatisierung mit dem Framework möglich sein müssen, dass eine individuelle Zusammenstellung des Testdurchlaufes möglich sein muss, es am Ende jedes Testdurchlaufes Details über die Fortschritte bei den einzelnen Testfällen sowie eine Übersicht über die Ergebnisse geben muss.

Keywords: Softwaretest, Robotic Process Automation, Softwareroboter, Testautomatisierung

1 EINLEITUNG

Die Komplexität von neuen Software-Systemen durch die Vielzahl an Kompetenzen nimmt zu und sorgt für einen enormen Ressourcenaufwand beim Testen [SnJu11]. Bei der Testautomatisierung werden manuelle durch Menschen durchgeführte Testaktivitäten durch eine Maschine durchgeführt. Das wiederum führt zu einer Zeit und Kostenersparnis. [AABK05]. Robotic Process Automation wird in den Unternehmen für die Automatisierung von Prozessen eingesetzt [Eise19]. RPA bietet eine Vielzahl an Werkzeugen und Schnittstellen, wodurch die Schritte der Menschlichen Interaktion mit einem System ersetzt werden können und so ganze Geschäftsprozesse mit RPA automatisiert werden können [LaTu20]. Es gibt bereits Ansätze RPA für Softwaretests einzusetzen. Gerade im End-to-End Bereich über User-Interfaces von Web-Applikationen hat sich gezeigt, dass RPA durchaus für Tests eingesetzt werden kann [CeSS20]. Es zeigte sich auch, dass RPA durchaus eine geeignete Alternative zu Selenium bei Tests von Web-Applikationen sein kann [Heis21]. Ein wesentlicher Vorteil beim Testen mit RPA ist die schnelle Erstellung von Robotern, da hierfür keine Programmierkenntnisse erforderlich sind. Die Erkenntnisse zeigten, dass es noch weitere Forschung und Anpassungen von RPA benötigt, damit RPA nicht nur auf einzelne Testszenarien, sondern als Testtool für umfangreiche Tests eingesetzt werden kann. RPA fehlt eine geeignete Testinfrastruktur, wie beispielsweise Test-Managing, Test-Reporting, Test-Durchführung, für die Testautomatisierung [CeSS20]. Daraus resultierte die Forschungsfrage:

„Wie muss ein Framework für den Einsatz von Robotic Process Automation Tools zur Testautomatisierung bei Regressionstests im End-to-End-Bereich konzipiert sein, damit die Steuerung der einzelnen Test-Cases in Form von RPA-Prozessen sowie ein Test-Reporting der Ergebnisse ermöglicht wird?“

2 METHODE

Für die Beantwortung der Forschungsfrage wurde die Methode des Design Science Research nach Österle et al. gewählt. Das ist eine Ableitung aus dem Modell von Hevner et al und beinhaltet einen vierstufigen Erkenntnisprozess [BeKS20]. Zudem ist es eine gestaltungsorientierte pragmatische Vorgehensweise bei der strukturiert in 4 Iterationsschritten, Analyse, Entwurf, Evaluation und Diffusion vorgegangen wird [PTRC07]. Da die Anforderungen an das Framework unbekannt sind und mit dieser Methode eine Vorgehensweise von der Analyse bis zur Diffusion stattfindet, erschien diese Methode als geeignet für die Beantwortung der Forschungsfrage. Zudem kann durch die jeweiligen Iterationen zwischen den einzelnen Schritten das Framework Schritt für Schritt erarbeitet, aber auch neue Erkenntnisse aus einem Folgeschritt wieder durch Rücksprünge in den vorigen Schritt miteingearbeitet werden. Dadurch war es möglich, in den vier iterativen Schritten einen funktionalen Prototyp des Frameworks zu entwickeln. In Abbildung 1 zeigt schematisch die Vorgehensweise mit den vier Iterationsschritten und den jeweiligen Ergebnissen aus den einzelnen Schritten.

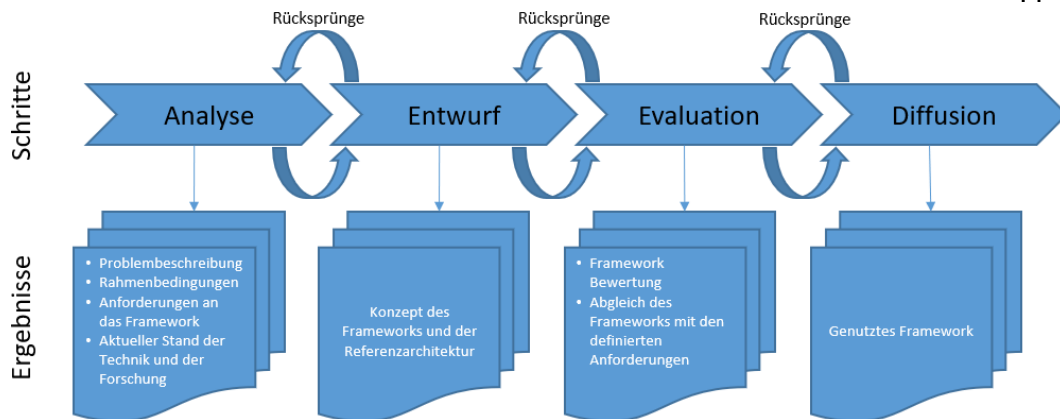


Abbildung 1: Iterationsschritte mit den Ergebnissen (eigene Darstellung in Anlehnung an [BeKS20])

3 FRAMEWORK

Ziel des Frameworks ist es fehlende Komponenten der RPA-Tools für die Testautomatisierung zu ergänzen und eine optimale Steuerung der Softwareroboter zu ermöglichen. Da bei den RPA-Tools die Entwicklung der Softwareroboter ohne umfangreiche Programmierkenntnisse erfolgt wird dieser Ansatz auch mit dem Framework beibehalten, so dass Anwender*Innen ihre Softwareroboter auch ohne Programmierkenntnisse in das Framework einbinden können. Abbildung 2 zeigt die schematische Darstellung der Architektur des Frameworks.

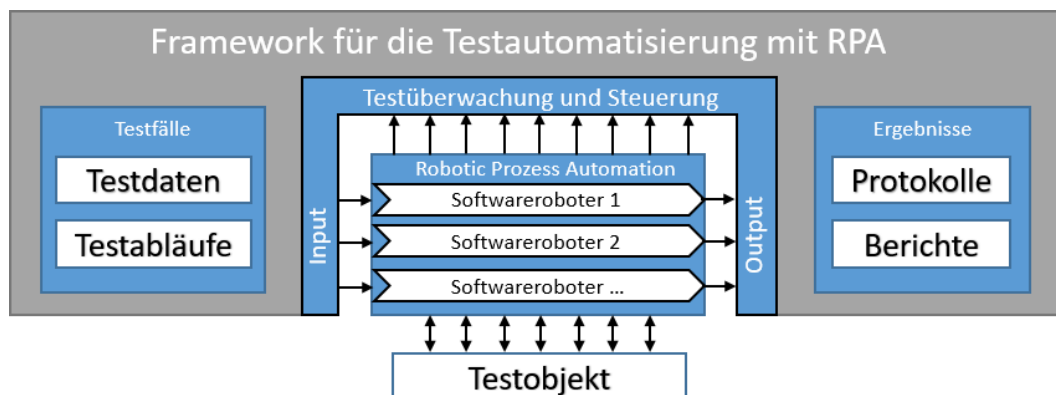


Abbildung 2: Architektur des Frameworks für die Testautomatisierung mit RPA (eigene Darstellung)

Das Framework verwaltet die Testfälle. Diese bestehen aus den Testdaten und aus den Testabläufen. Testabläufe können einerseits aus den Skripten der einzelnen Softwareroboter bestehen und andererseits bestehen diese aus der Beschreibung, worin beschrieben wird, was bei diesem Test passieren soll.

Im Zentrum des Frameworks ist die RPA-Software mit den Softwarerobotern eingebettet. Das Framework steuert und überwacht die Softwareroboter und bereitet die Ergebnisse der Softwareroboter auf. Ausschließlich die Softwareroboter interagieren mit dem Testobjekt. Bei einem Testdurchlauf werden die einzelnen Softwareroboter sequenziell

vom Framework gestartet. Die Softwareroboter holen sich die entsprechenden Daten und arbeiten den Testfall ab. Dabei protokollieren sie ihren Fortschritt und die jeweiligen Testergebnisse. Das Framework wartet bis der Softwareroboter seine Arbeit beendet hat und startet im Anschluss den nächsten. Damit eine RPA-Software mit dem Framework verwendet werden kann gibt es zwei Voraussetzungen. Einerseits muss der Softwareroboter über die Kommandozeile gestartet werden können und andererseits müssen die Funktionen der RPA-Software mittels Dynamic Link Library (DLL) erweiterbar sein. Diese Funktionen ermöglichen es, dass der Softwareroboter die Testergebnisse und Fortschrittsinformationen speichern kann und zusätzlich ist es möglich bei einem Fehler des Softwareroboters das er den Testfall auf den Status Error setzen und sich selbst beenden kann.

Die protokollierten Fortschritte und Ergebnisse des Soll- und Ist-Vergleiches werden am Ende aufbereitet und zu einem Bericht zusammengefasst. Im Bericht wird der Start- und Endzeitpunkt wie auch die Dauer des Testdurchlaufes angezeigt. Jeder Testfall wird aufgelistet und mit dem jeweiligen Ergebnis wie OK, FAIL oder Error angezeigt. Da bei einem Testfall mehrere Ergebnisse möglich sind, wird in der Übersicht immer das schlechteste ausgegeben. Error kommt vor FAIL und FAIL vor OK. Diese Ergebnisse werden zur besseren Übersicht auch in einem Balkendiagramm dargestellt. Zusätzlich können zu einem Testfall die Detailergebnisse mit allen Testergebnissen und Fortschrittsinformationen angezeigt werden. Dadurch ist es möglich, dass die Testergebnisse aus dem Soll- und Ist-Vergleich schneller nachvollzogen werden können und durch die Fortschrittsprotokollierung kann bei einem auftretenden Fehler beim Softwareroboter schnell die Ursache nachvollzogen werden.

4 CONCLUSIO

Die Evaluierung des Prototyps hat gezeigt, dass ein Framework so gestaltet sein muss, dass die Prozesse der Testautomatisierung auch mit dem Framework umgesetzt werden können. Bei diesem Framework werden die fehlenden Funktionalitäten wie Testdatenverwaltung, Test-Reporting und Test-Durchführung bei RPA-Tools ergänzt und somit kann RPA als primäres Testtool eingesetzt werden. Eine Erweiterung der Funktionen der RPA-Tools ist erforderlich, damit beim Start jedem Softwareroboter entsprechende Informationen über den Testfall übergeben werden können, das Framework den Fortschritt des Softwareroboters überwachen und der Softwareroboter entsprechende Fortschrittsinformationen und Testergebnisse an das Framework zurück liefern kann. Das Framework übernimmt die strukturierte Verwaltung der Testfälle mit den Testdaten. Die Testdurchläufe können individuell zusammengestellt und angepasst werden. Am Ende jedes Testdurchlaufes wird eine Übersicht erstellt welche Informationen über die Ergebnisse der Tests anhand eines Soll- und Ist-Vergleiches, sowie Details über die Fortschritte bei den einzelnen Testfällen beinhaltet.

5 REFERENZEN

- [AABK05] AMIRY, SAMIR ; ARMBRUST, OVE ; BERGER, JULIA ; KLINCK, JANINE ; LUTTENBERGER, KONSTANTIN: Research Lab Rheinland-Pfalz Testen und Testautomatisierung: Anforderungen an Testwerkzeuge und Marktstudie. In: *IESE-Report Nr Bd. 131* (2005)
- [BeKS20] BENNER-WICKNER, MARIAN ; KNEUPER, RALF ; SCHLÖMER, INGA: *Leitfaden für die Nutzung von Design Science Research in Abschlussarbeiten* : IUBH Discussion Papers-IT & Engineering, 2020
- [CeSS20] CERNAT, MARINA ; STAIUCU, ADELINA-NICOLETA ; STEFANESCU, ALIN: Improving UI Test Automation using Robotic Process Automation. In: *ICSOFIT, 2020*, S. 260–267
- [Eise19] EISELE, OLAF: ROBOTIC PROCESS AUTOMATION (RPA) Mensch-Roboter-Kollaboration in indirekten Bereichen. In: *ifaa — Institut für angewandte Arbeitswissenschaft e. V.* (2019), S. 1–6
- [Heis21] HEISKANEN, ANDREAS: Robotic Process Automation in automated GUI testing of Web Applications.
- [LaTu20] LANGMANN, CHRISTIAN ; TURI, DANIEL: *Robotic Process Automation (RPA) - Digitalisierung und Automatisierung von Prozessen: Voraussetzungen, Funktionsweise und Implementierung am Beispiel des Controllings und Rechnungswesens* : Springer-Verlag, 2020. — Google-Books-ID: QrPRDwAAQBAJ — ISBN 978-3-658-28299-8
- [PTRC07] PEFFERS, KEN ; TUUNANEN, TUURE ; ROTHENBERGER, MARCUS A. ; CHATTERJEE, SAMIR: A design science research methodology for information systems research. In: *Journal of management information systems* Bd. 24, Taylor & Francis (2007), Nr. 3, S. 45–77
- [SnJu11] SNEED, HARRY M. ; JUNGMAHR, STEFAN: Mehr Testwirtschaftlichkeit durch Value-Driven-Testing. In: *Informatik-Spektrum* Bd. 34 (2011), Nr. 2, S. 192–209. — Company: SpringerDistributor: SpringerInstitution: SpringerLabel: Springernumber: 2publisher: Springer-Verlag