Robot Framework – der zukünftige Industriestandard für Testautomation

Version 1.1 vom 03.05.2023

imbus weltweit
www.imbus.de

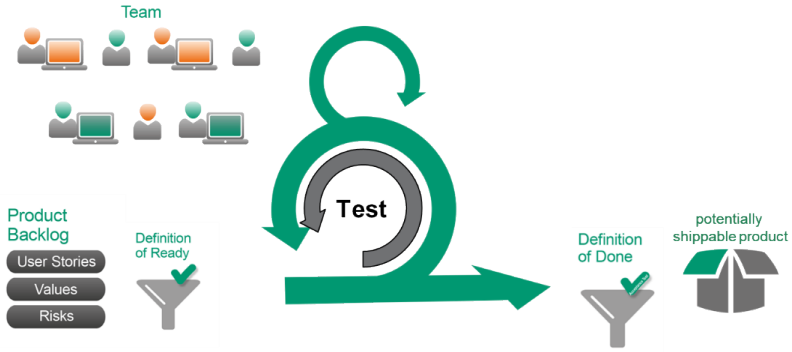
Das Robot Framework – der zukünftige Industriestandard für Testautomation

Version 1.1 vom 03.05.2023

Was ist Testautomation und warum wird diese immer wichtiger?

Die Aufgabe automatisierter Tests ist es, die Korrektheit bestehender Kernfunktionalität einer Anwendung sicherzustellen und auch bei Änderungen derselben immer wieder in sogenannten Regressionstests zu überprüfen, anders ausgedrückt: Unbeabsichtigte Quereffekte von Änderungen möglichst vor dem produktiven Einsatz des neuen Releases zu finden.
Betrugen Release-Zyklen in der Vergangenheit viele Monate (oder sogar Jahre), so haben sie sich bei Scrum auf typischerweise 2 Wochen verkürzt.
Bei Scrum ist das sog. „potentially shippable product" am Ende eines Sprints sehr wichtig für die Velocity des gesamten Teams, kostet doch jeder Kontext-Wechsel (Entwickler müssen Fehler beheben, die Ihnen vor vielen Wochen unterlaufen sind) viel Effizienz. Umfangreiche Regressionstests während und am Ende des Sprints sollen daher sicherstellen, dass alle in einem Sprint eingebauten Quereffekte zeitnah entdeckt und noch innerhalb desselben Sprints behoben werden – dann haben alle Teammitglieder am Ende eines Sprints den „Kopf vollständig für den nächsten Sprint frei".

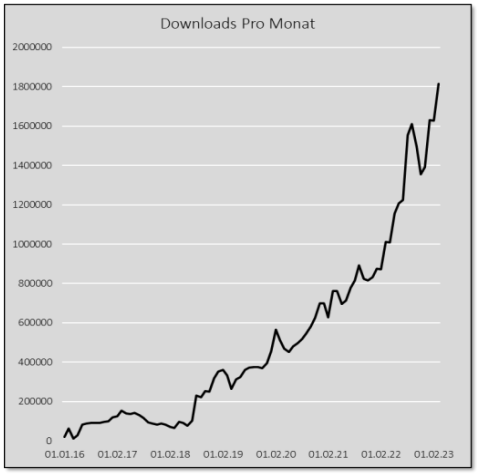
Die dafür notwendige Abdeckung und Geschwindigkeit der Durchführung der Regressionstests ist nur mit einer umfassenden Automatisierung erreichbar.



Und bei 2-wöchigen Release-Zyklen bleibt es nicht – mit der Umsetzung von CD-Pipelines (Continuous Deploymemt) bzw. sog. „Value Streams" bei DevOps werden die Release-Zyklen noch deutlicher kürzer sein und die Bedeutung von Testautomatisierung daher noch weiter zunehmen. Auch deswegen sagte z.B. Marc Hornbeek[1] 2021: „Testing is ***THE*** major bottleneck for ***most*** DevOps value streams".

Welche Verbreitung hat das RFW?

Und warum sollte man sich mit dem Robot Framework (RFW) beschäftigen, wenn man Testfälle automatisieren möchte?



Das RFW wird inzwischen **mehr als 1,8 Mio pro Monat** runtergeladen!

Von solchen Zahlen können die einschlägigen Hersteller von Testtools nur träumen.

Die erste Idee zu diesem Framework für die Automatisierung funktionaler Regressionstests hatte im Jahre 2005 Pekka Klärck, der damals für Nokia arbeitete. Er ist auch heute noch der Chef-Architekt hinter dem RFW.

Der Initiative von Frank Blechschmidt bei imbus ist es zu verdanken, dass 2015 mit imbus das erste Unternehmen außerhalb Finnlands der RFW Foundation beitrat. Heute sitzen die meisten User in Indien (19%), gefolgt von den USA (13%), Deutschland ist mit 6% inzwischen vor Finnland, Brasilien, Frankreich und China (jeweils 5%).

[1] CEO und Principal Consultant bei Engineering DevOps, Ambassador des DevOps-Institutes

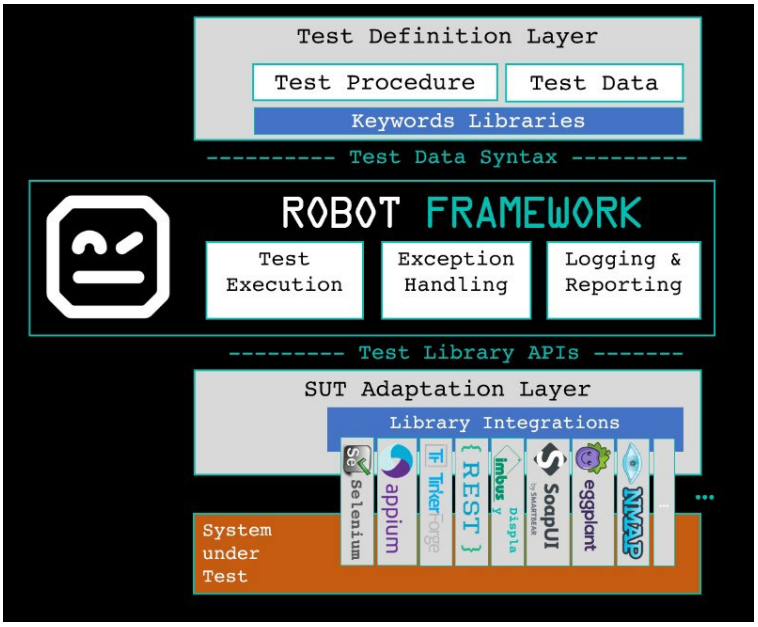
Was zeichnet das RFW aus?

Das Wichtigste ist sicherlich die Offenheit und Erweiterbarkeit. Das Robot Framework wurde bereits 2008 unter der Apache License 2.0 veröffentlicht und ist damit Open Source. In Verbindung mit der riesigen Community wird es sehr intensiv weiterentwickelt: es entstehen ständig neue Libraries (Bibliothek, um auf Oberflächen, Schnittstellen, ... des „System under Test“ zugreifen zu können) und Erweiterungen, vorhandene Libraries werden verbessert und das RFW selbst erhält neue Features. Und bei Fragen und Problemen erhält man schnell Hilfe von der Community, so sind z.B. in dem entsprechenden Slack-Channel über 14.000 RFW-User (Stand April 2023) aktiv.

Es gibt fast nichts, was man mit dem RFW nicht auch automatisieren kann: Stand Mitte 2022 sind es bereits **mehr als 400 Libraries**, ein paar Beispiele seien hier genannt: Von Selenium und Playwright (Web) über Postman (Microservices), RoboSAPiens (SAP GUI, Library von Dr. Marduk Bolaños Puchet / imbus) und Appium (Mobile) bis hin zu Eggplant (non-traditional UIs wie z.B. Graphiken, RFW library von Andre Mochinin / imbus). Und sollte man doch mal eine Schnittstelle automatisiert ansprechen wollen, die wider Erwarten noch nicht unterstützt wird: Das RFW ist offen und kann jederzeit und von jedermann um weitere Libraries erweitert werden.

Möglich macht das die Orientierung des RFW an der sog. **generischen Testautomatisierungs-Architektur**, wie Sie in der ISO 29119 Teil-5 Keyword Driven Testing und im ISTQB® Advanced Level Testautomation Engineer definiert ist. Hier wird unterschieden zwischen:

- **Test Definition Layer**: Beschreibung der Testsequenz, Spezifikation der Testfalldaten, Definition der sog Keywords (wiederverwendbare Testschritte) und dem dazugehörigen Editor
- **Test Execution Layer**: Einlesen und Ausführen der **Testspezifikation**, Error Handling, Protokollierung der Testdurchführung und Reporting
- **SuT Adaption Layer**: Alle Automatisierungstechnologien und die Automatisierungsbibliotheken, damit die Automatisierungslösung auf das Testobjekt zugreifen kann; hier findet die eigentliche Implementierung der Automatisierung und damit Softwareentwicklung statt.



Ein großer Vorteil der generischen Testautomatisierungs-Architektur ist, dass die Schichten voneinander unabhängig sind: Die Beschreibung des Testfalls ist unabhängig von der darunter liegenden Technik und eine Änderung in einer Schicht hat keine Auswirkungen auf die anderen Schichten, ein konkretes Beispiel: Der Testautomations-Engineer, welcher bisher das RFW in Verbindung mit Selenium genutzt hat, arbeitet ab sofort mit Playwright statt Selenium – diese Änderung bzw. Fortentwicklung hat keinerlei Auswirkung auf die Spezifikation der Testfälle auf der Ebene des Test Definition Layer.

Der Test Definition Layer des RFW unterstützt dabei sowohl **BDT** (Behavior Driven Testing, Stichwort Cucumber / Gherkin) als auch **KDT** (Keyword Driven Testing, siehe z.B. Keyword-Driven Testing, erschienen im dpunkt-Verlag). Mit dem RFW kann man automatisierte Testschritte für beide Arten der Testfallnotation wiederverwenden.

Zudem unterstützt der Test Definition Layer die Definition mehrerer Ebenen an Keywords. Damit kann der Testautomations-Engineer dem Testdesigner einen „Wortschatz“ an Keywords bereitstellen, auf dessen Basis er mit seinem Domänenwissen die Testsequenz spezifiziert. Der Testautomations- Engineer mappt diese fachlichen Keywords einmalig auf technische Keywords, die er dann automatisiert, und schon sind die spezifizierten Testfälle automatisch ausführbar – ganz ohne Medienbrüche und Redundanzen.

Auf der Ebene des Test Execution Layers kann das RFW sehr einfach an CI/CD-Pipelines angeschlossen werden und für die am weitesten verbreiteten CI-Systeme gibt es bereits fortgeschrittene Integrationen (z.B. Jenkins-Plugin). Auch für Testmanagement-Systeme wie z.B. X-Ray oder TestBench sind bereits Anbindungen verfügbar – hierfür sogar auf Ebene von Keywords, die sich in der TestBench per Drag&Drop zu der Testsequenz zusammenstellen lassen. Damit ist dann die Nachvollziehbarkeit („Traceability“) von der zugrunde liegenden User Story bzw. Anforderung über den Testfall bis hin zum Testergebnis und ggf. der Fehlermeldung gegeben.

Die ISO 29119 Teil-5 Keyword Driven Testing beinhaltet u.a. eine Liste an 32 Eigenschaften, über die ein Framework auf jeden Fall verfügen sollte – diese werden in der Norm mit »basic attributes« (Basisattribute) bezeichnet. Darüber hinaus gibt es noch 37 wünschenswerte Eigenschaften, welche in der Norm »advanced attributes« (fortgeschrittene Attribute) heißen. Eine Bewertung des RFW anhand dieser Norm ist in dem Kapitel „6.4.2 Framework 2: Robot Framework“ des Buches „Keyword-Driven Testing“ zu finden, welches der dpunkt-Verlag dankenswerterweise zur Verfügung gestellt hat: OpenBook Keyword-Driven Testing. Zusammenfassend kann man feststellen, dass das RFW 26 der Basisattribute vollständig erfüllt, weitere 4 teilweise und nur 2 nicht (manuelle Testausführung und Fehler-Management – Aufgaben, die ein angeschlossenes Testmanagement-Werkzeug übernehmen kann und sollte).

Und wer steht „hinter“ dem Robot Framework?

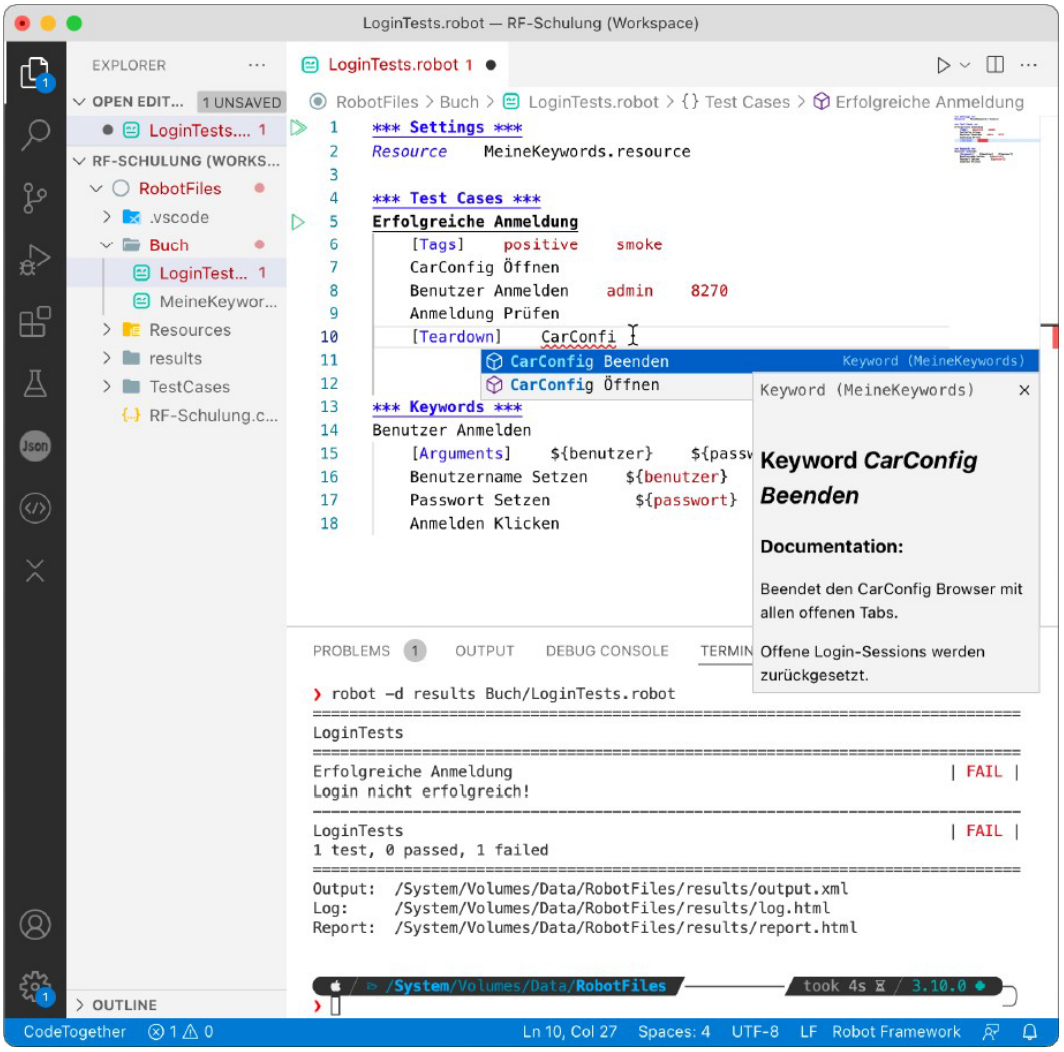
Hier ist die Robot Framework Foundation zu nennen, welche 2015 gegründet wurde und der inzwischen 56 Firmen (Stand April 2023) angehören. Die Foundation koordiniert die Fortentwicklung des RFW, organisiert die Robocon und verfügt über ein Budget von ca. 250 T€ (Stand April 2023).

Für alle Nutzer des RFW ist die Existenz der RFW Foundation ein Garant dafür, dass es **das RFW noch sehr lange geben wird**.

Und wie sieht das RFW nun konkret aus?

Die Testsequenzen und Testfalldaten des RFW werden in Textdateien abgelegt. Damit ist das RFW mit allen Versionierungs-/SCM-Systemen (Source Code Management) kompatibel. RFW-kompatible Test Suiten können entweder durch ein Testmanagement-Werkzeug wie z.B. die TestBench, generiert werden, oder man nutzt eine unterstützte IDE seiner Wahl.

Im Folgenden wird das am Beispiel der IDE Visual Studio Code, kurz „VS Code“, von Microsoft erläutert. In Verbindung mit Robot Code (dem PlugIn für VS Code), werden dann auch so komfortable Funktionen wie Code Completion (IntelliSense) für das RFW unterstützt:

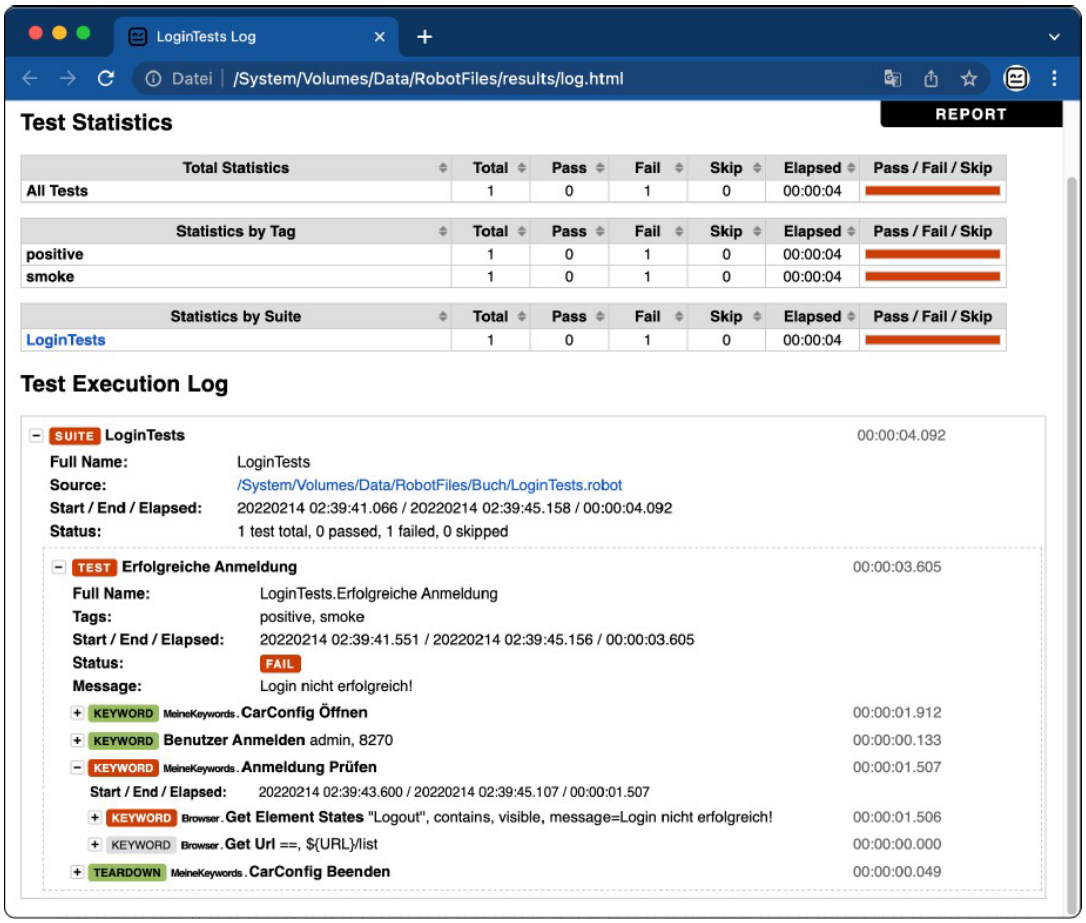


Quelle: Buch „Keyword-Driven Testing“ von Matthias Daigl / René Rohner

Der Testdesigner hat `CarConfi` eingegeben und es werden ihm die zwei vorhandenen Keywords `CarConfig Beenden` und `CarConfig Öffnen` vorgeschlagen, die dazu passen. Unterhalb des Testfalls `ErfolgreicheAnmeldung` ist die Definition des Keywords `Benutzer Anmelden` sichtbar. Im unteren Bereich der GUI ist eine Kommandozeile und die daraus resultierende Ausgabe des gelaufenen und fehlgeschlagenen RFW-Testfalls zu sehen.

Auch ein Testfall, der keinen Fehler gefunden hat, hat seinen Wert: Er weist die korrekte Funktionsweise des SuT (System under Test) nach. Schlägt ein Testfall jedoch fehl, dann unterstützt das RFW bei der Nachvollziehbarkeit und Ursachenanalyse durch eine entsprechende Protokollierung.

Und die schnelle Analyse umfangreicher automatisierter Regressionstest-Läufe ermöglicht das RFW durch das enthaltene Reporting:



Quelle: Buch „Keyword-Driven Testing“ von Matthias Daigl / René Rohner

Und wie fange ich nun mit der Arbeit mit dem RFW an?

Die große und monatlich weiter steigende Verbreitung, Schnittstellen zu allen üblichen Software-Systemen, Offenheit und Erweiterbarkeit – **das ist der zukünftige Industriestandard für die Automation funktionaler Regressionstests!**

Wer nun anfangen möchte, mit dem RFW Testfälle zu automatisieren, dem sei die Lektüre des Kapitels „7 Praxis mit Robot Framework“ des Buches „Keyword-Driven Testing“ (Hinweis: Dies ist kein Affiliate-Link[2]) von Matthias Daigl und René Rohner (beide Mitarbeiter bei imbus, René seit 2022 Vorsitzender der RFW-Foundation) empfohlen.

Die nachfolgenden Quellen sollen dem/der interessierten Leser:in einen zügigen Start bei der Arbeit mit dem RFW ermöglichen:

- RobotCode (by Daniel Biehl/imbus): https://github.com/d-biehl/robotcode
- Robot Framework Foundation: https://robotframework.org/foundation/
- Robot Framework User Guide: https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html
- Robot Framework Project: https://github.com/robotframework/robotframework

[2] Der Autor dieses Fachartikels hat keinerlei Vorteil beim Kauf dieses Buches

- Robot Framework Website: https://robotframework.org
- Robot Framework Language Server: https://marketplace.visualstudio.com/items?itemName=robocorp.robotframework-lsp
- Slack-Channel für das RFW: https://robotframework.slack.com/
- ISTQB® Certified Tester Advanced Level Test Automation Engineer Syllabus: https://www.istqb.org/downloads/category/48-advanced-level-test-automation-engineerdocuments
- Kurs ISTQB® Certified Tester Advanced Level Test Automation Engineer bei imbus: https://www.imbus.de/akademie/istqb-advanced-test-automation-engineer
- Testautomatisierung mit Robot Framework bei imbus: https://www.imbus.de/softwaretest/testautomatisierung-mit-robot-framework
- Robot Framework Browser Library: https://robotframework-browser.org
- Microsoft Playwright: https://playwright.dev/
- Microsoft Visual Studio Code: https://code.visualstudio.com
- ISO 29119 Teil 5 „Keyword Driven Testing“ im Webshop des Beuth Verlags: https://www.beuth.de/de/norm/iso-iec-ieee-29119-5/266302684
- Buch „Keyword-Driven Testing“ von Matthias Daigl / René Rohner, erhältlich im dpunkt Verlag https://dpunkt.de/produkt/keyword-driven-testing/; das vom dpunkt-Verlag dankenswerterweise bereit gestellte Kapitel „6.4.2 Framework 2: Robot Framework“ ist zu finden unter: OpenBook Keyword-Driven Testing
- RoboSAPiens: Flyer Integration von SAP Automatisierung in komplexe Systeme

Erfahrung ist durch nichts zu ersetzen außer durch noch mehr Erfahrung! Sollte der/die geneigte Leser:in nicht die Zeit und Muße haben, alle diese Erfahrungen selber zu machen, dann kann er/sie sich auf der jährlichen Robocon mit über 600 Teilnehmern austauschen, sich mit Fragen an einen der über 14.000 Mitglieder in dem Community Chat (Slack) wenden und sich von imbus bei der Einführung des RFW schulen und beraten lassen (www.imbus.de, 09131, 7518 – 0, info@imbus.de).