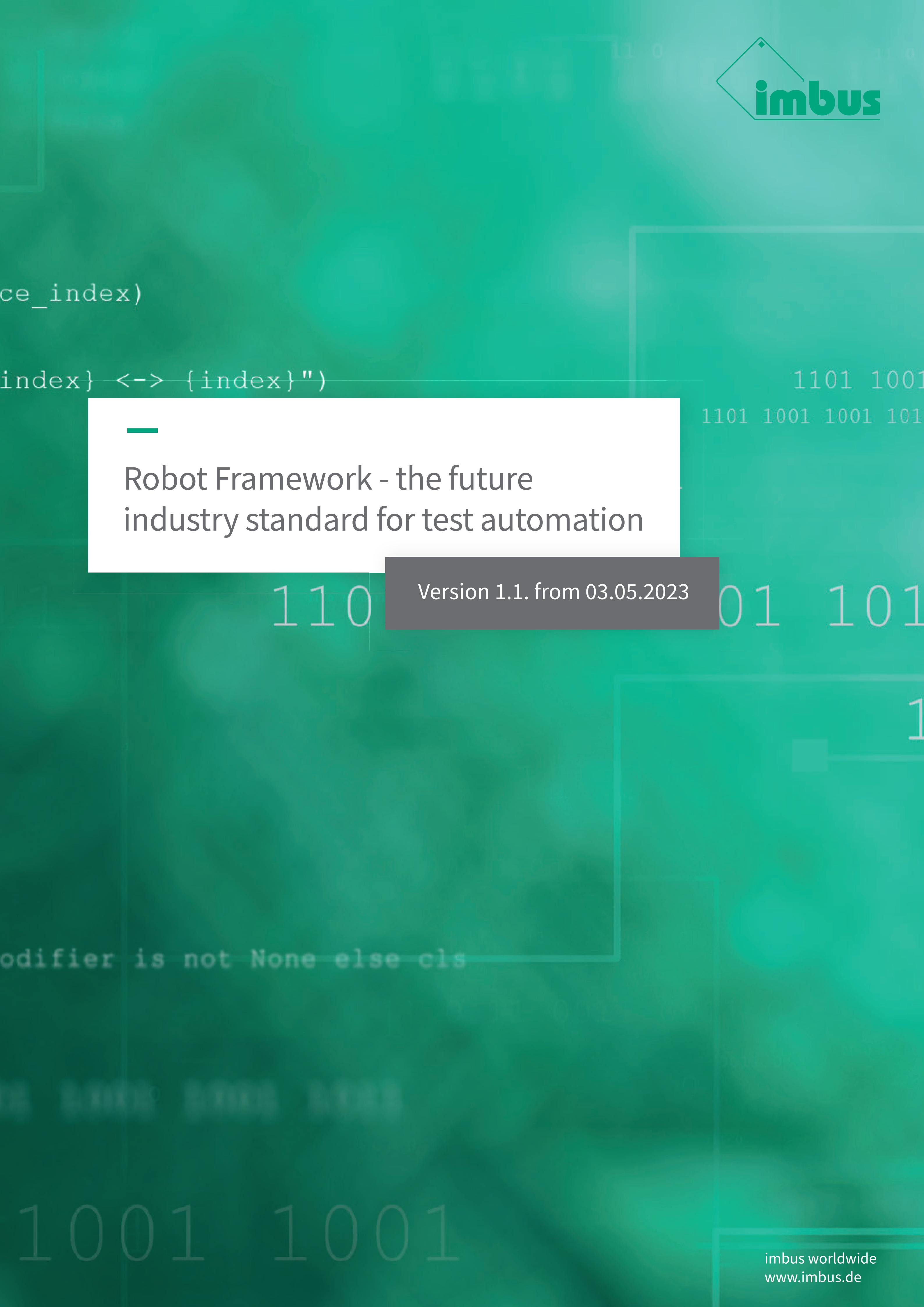


Robot Framework - the future industry standard for test automation

Version 1.1. from 03.05.2023

imbus worldwide
www.imbus.de

Robot Framework - the future industry standard for test automation

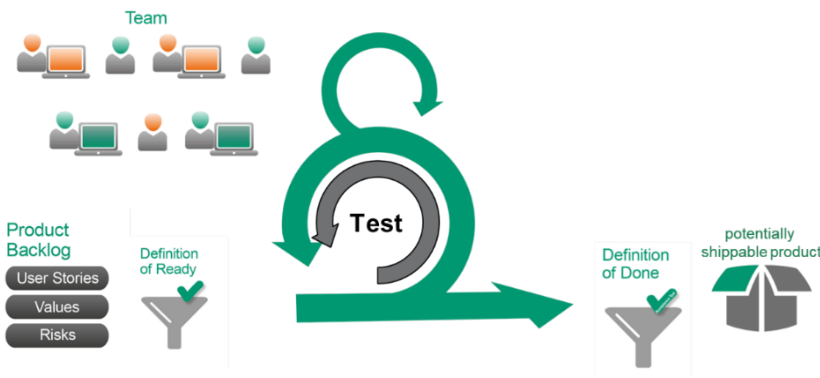
Version 1.1 from 03.05.2023

What is test automation and why is it becoming increasingly important?

The task of automated tests is to ensure the correctness of the existing core functionality of an application and to check it again and again in so-called regression tests even when changes are made to it, in other words: to find unintended cross-effects of changes before the new release goes live, if possible.

Whereas release cycles used to take many months (or even years) in the past, they have been reduced to typically 2 weeks with Scrum.

In Scrum, the so-called "potentially shippable product" at the end of a sprint is very important for the velocity of the entire team, as every context change (developers have to fix errors that they made many weeks ago) costs a lot of efficiency. Extensive regression tests during and at the end of the sprint should therefore ensure that all cross-effects built into a sprint are discovered promptly and rectified within the same sprint - then all team members have their "heads completely free for the next sprint" at the end of a sprint.



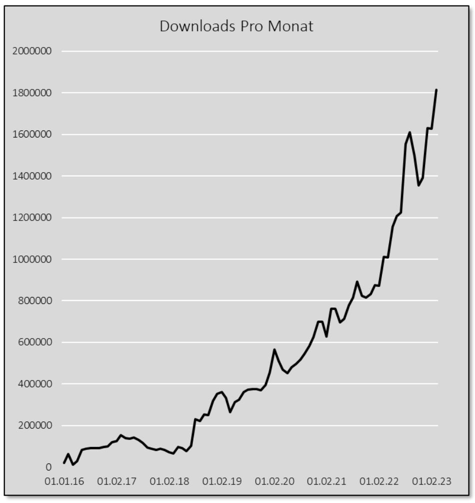
The necessary coverage and speed of the regression tests can only be achieved with comprehensive automation.

And 2-week release cycles are not enough - with the implementation of CD pipelines (Continuous Deploymemt) or so-called "value streams" in DevOps, the release cycles will be even shorter and the importance of test automation will therefore increase even further. This is also why Marc Hornbeek, for example, said[1] 2021: "Testing is ***THE*** major bottleneck for ***most*** DevOps value streams".

How widespread is the RFW?

And why should you bother with the Robot Framework (RFW) if you want to automate test cases?

RFW is now downloaded **more than 1.8 million times a month**!



The relevant manufacturers of test tools can only dream of such figures.

The first idea for this framework for the automation of functional regression tests came from Pekka Klärck, who was working for Nokia at the time, in 2005. He is still the chief architect behind the RFW today.

Thanks to the initiative of Frank Blechschmidt at imbus, imbus became the first company outside Finland to join the RFW Foundation in 2015. Today, most users are based in India (19%), followed by the USA (13%), while Germany is now ahead of Finland, Brazil, France and China (5% each) with 6%.

[1] CEO and Principal Consultant at Engineering DevOps, Ambassador of the DevOps Institute

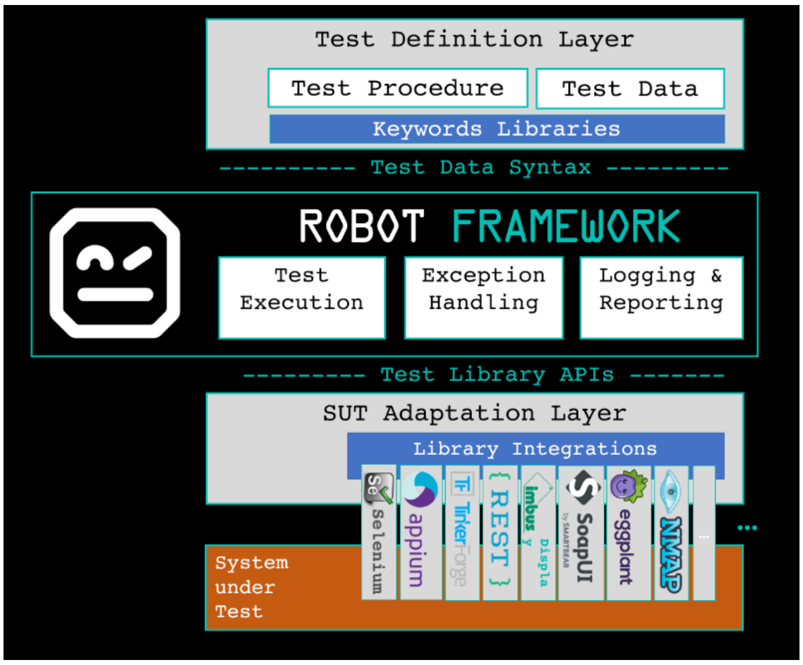
What distinguishes the RFW?

The most important thing is certainly its openness and expandability. The Robot Framework was published back in 2008 under the Apache License 2.0 and is therefore open source. In conjunction with the huge community, it is being developed very intensively: new libraries (library to access GUIs, APIs, ... of the "system under test") and extensions are constantly being created, existing libraries are being improved and the RFW itself is being given new features. And if you have any questions or problems, you can quickly get help from the community; for example, there are over 14,000 active RFW users in the corresponding Slack channel (as of April 2023).

There is almost nothing that cannot be automated with the RFW: As of mid-2022, there are already **more than 400 libraries**, a few examples of which are listed here: From Selenium and Playwright (web) to Postman (microservices), RoboSAPiens (SAP GUI, library by Dr. Marduk Bolaños Puchet / imbus) and Appium (mobile) to Eggplant (non-traditional UIs such as graphics, RFW library by Andre Mochinin / imbus). And should you ever want to automatically address an interface that, contrary to expectations, is not yet supported: The RFW is open and can be extended with additional libraries at any time and by anyone.

This is made possible by the RFW's orientation towards the so-called **generic test automation architecture,** as defined in ISO 29119 Part-5 Keyword Driven Testing and in the ISTQB® Advanced Level Test Automation Engineer. A distinction is made here between

- **Test Definition Layer**: Description of the test sequence, specification of the test case data, definition of the so-called keywords (reusable test steps) and the corresponding editor
- **Test execution layer**: reading and executing the **test specification**, error handling, logging of test execution and reporting
- **SuT Adaption Layer**: All automation technologies and the automation libraries so that the automation solution can access the test object; this is where the actual implementation of the automation and thus software development takes place.



A major advantage of the generic test automation architecture is that the layers are independent of each other: The description of the test case is independent of the underlying technology and a change in one layer has no effect on the other layers, a concrete example: The test automation engineer, who previously used the RFW in conjunction with Selenium, now works with Playwright instead of Selenium - this change or further development has no effect on the specification of the test cases at the level of the test definition layer.

The Test Definition Layer of the RFW supports both **BDT** (Behavior Driven Testing, keyword Cucumber / Gherkin) and **KDT** (Keyword Driven Testing, see e.g. Keyword-Driven Testing, published by dpunkt-Verlag). With the RFW, automated test steps can be reused for both types of test case notation.

The test definition layer also supports the definition of multiple levels of keywords. This allows the test automation engineer to provide the test designer with a "vocabulary" of keywords, based on that he can specify the test sequence using his domain knowledge. The test automation engineer maps these domain-specific keywords once to technical keywords, which he then automates, and the specified test cases can be executed automatically - without any media breaks or redundancies.

At the test execution layer level, the RFW can be very easily connected to CI/CD pipelines and advanced integrations are already available for the most widely used CI systems (e.g. Jenkins plugin). Connections are also already available for test management systems such as X-Ray or TestBench - even at the level of keywords, which can be combined into the test sequence in TestBench using drag & drop. This ensures traceability from the underlying user story or requirement through the test case to the test result and, if applicable, the error message.

ISO 29119 Part 5 Keyword Driven Testing includes a list of 32 properties that a framework should have in any case - these are referred to as "basic attributes" in the standard. In addition, there are 37 desirable properties, which are called "advanced attributes" in the standard. An evaluation of the RFW based on this standard can be found in chapter "6.4.2 Framework 2: Robot Framework" of the book "Keyword-Driven Testing", which dpunkt-Verlag has kindly made available: OpenBook Keyword-Driven Testing. In summary, it can be stated that the RFW fully fulfills 26 of the basic attributes, another 4 partially and only 2 not (manual test execution and error management - tasks that a connected test management tool can and should take over).

And who is "behind" the Robot Framework?

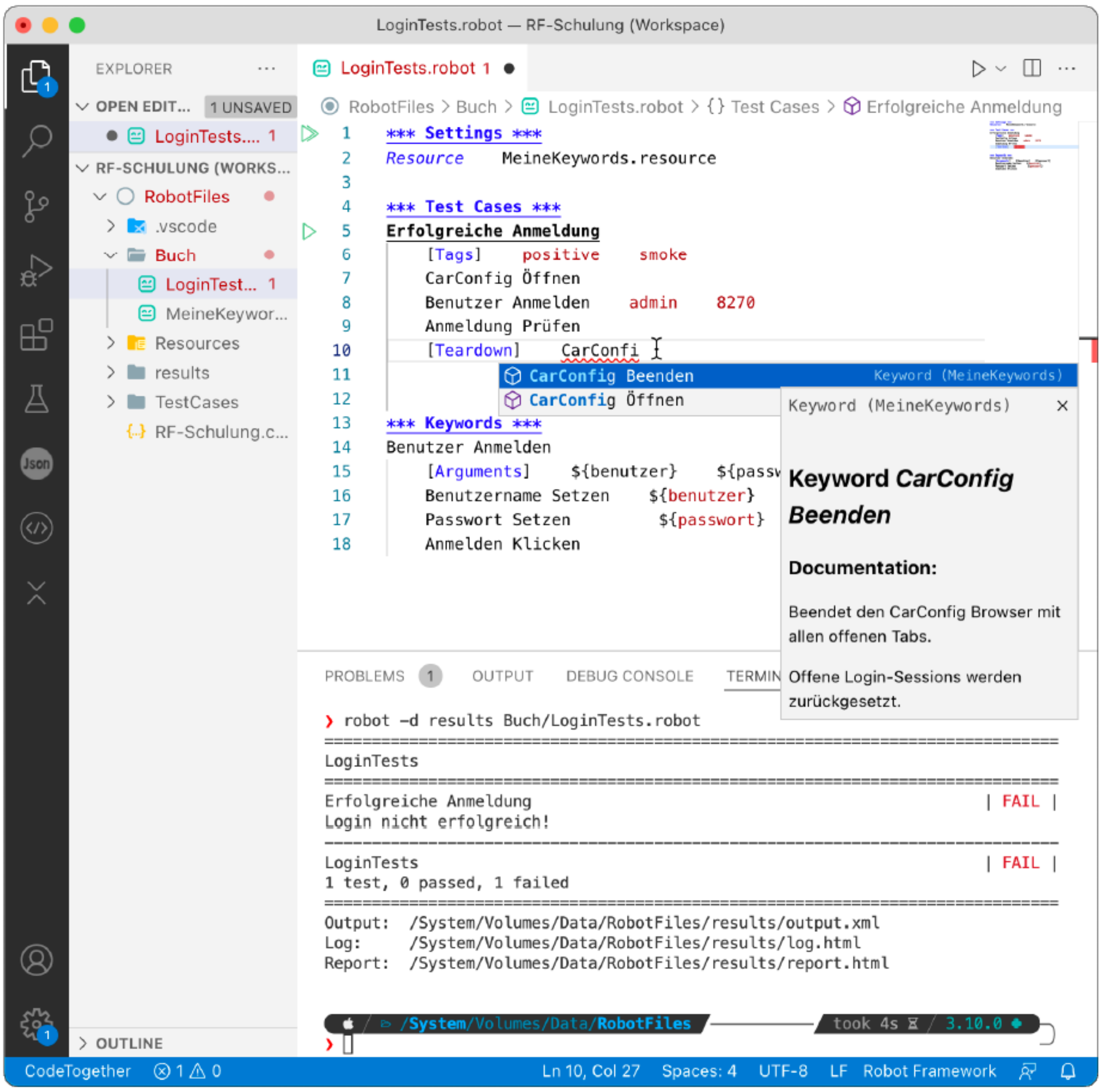
This includes the Robot Framework Foundation, which was established in 2015 and now has 56 companies as members (as of April 2023). The Foundation coordinates the further development of the RFW, organizes Robocon and has a budget of around €250 thousand (as of April 2023).

For all users of the RFW, the existence of the RFW Foundation is a guarantee that **the RFW will continue to exist for a very long time**.

And what does the RFW actually look like?

The test sequences and test case data of the RFW are stored in text files. This makes the RFW compatible with all versioning/SCM (Source Code Management) systems. RFW-compatible test suites can either be generated by a test management tool such as TestBench, or you can use a supported IDE of your choice.

This is explained below using Microsoft's Visual Studio Code IDE, or "VS Code" for short, as an example. In conjunction with Robot Code (the plug-in for VS Code), convenient functions such as Code Completion (IntelliSense) are also supported for the RFW:



```
*** Settings ***
Resource      MeineKeywords.resource

*** Test Cases ***
Erfolgreiche Anmeldung
    [Tags]    positive    smoke
    CarConfig Öffnen
    Benutzer Anmelden    admin    8270
    Anmeldung Prüfen
    [Teardown]    CarConfi
                       CarConfig Beenden                Keyword (MeineKeywords)
                       CarConfig Öffnen

*** Keywords ***
Benutzer Anmelden
    [Arguments]    ${benutzer}    ${passw
    Benutzername Setzen    ${benutzer}
    Passwort Setzen    ${passwort}
    Anmelden Klicken
```

Keyword (MeineKeywords)

Keyword *CarConfig Beenden*

Documentation:

Beendet den CarConfig Browser mit allen offenen Tabs.

Offene Login-Sessions werden zurückgesetzt.

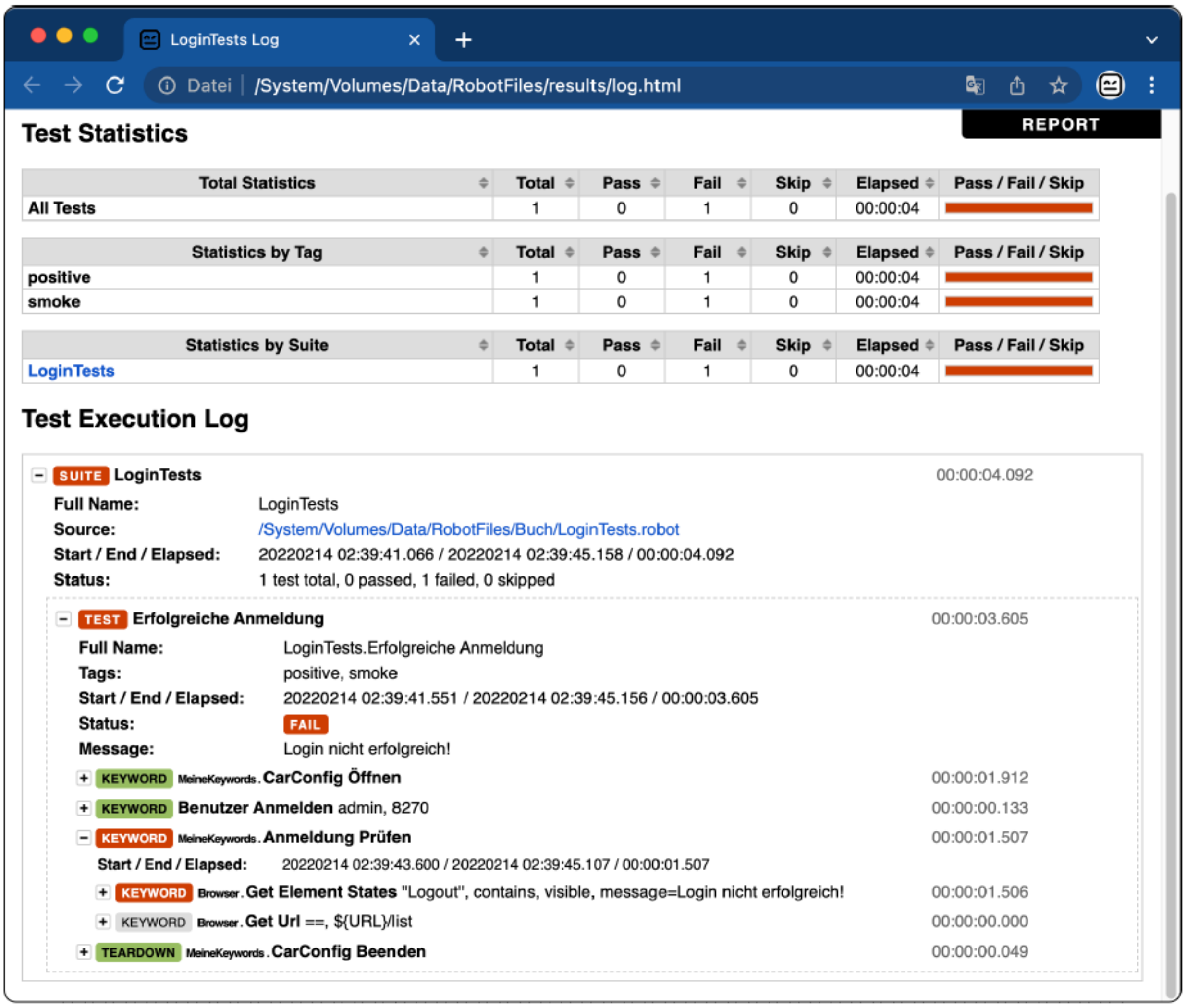
```
❯ robot -d results Buch/LoginTests.robot
==============================================================================
LoginTests
==============================================================================
Erfolgreiche Anmeldung                                                | FAIL |
Login nicht erfolgreich!
------------------------------------------------------------------------------
LoginTests                                                            | FAIL |
1 test, 0 passed, 1 failed
==============================================================================
Output:  /System/Volumes/Data/RobotFiles/results/output.xml
Log:     /System/Volumes/Data/RobotFiles/results/log.html
Report:  /System/Volumes/Data/RobotFiles/results/report.html

 /System/Volumes/Data/RobotFiles ──────────── took 4s ⧗ / 3.10.0
❯
```

Source: Book "Keyword-Driven Testing" by Matthias Daigl / René Rohner

The test designer has entered `CarConfi` and the two existing keywords `CarConfig Exit` and `CarConfig Open` which match it are suggested. The definition of the keyword `User login` is visible below the `Successful login` test case. In the lower area of the GUI, a command line and the resulting output of the RFW test case that has run and failed can be seen.

Even a test case that has not found an error has its value: it proves that the SuT (system under test) is functioning correctly. However, if a test case fails, the RFW provides support with traceability and root cause analysis through appropriate logging.

And the RFW enables the rapid analysis of extensive automated regression test runs thanks to the included reporting:



Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	1	0	1	0	00:00:04	🟥

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
positive	1	0	1	0	00:00:04	🟥
smoke	1	0	1	0	00:00:04	🟥

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
LoginTests	1	0	1	0	00:00:04	🟥

Test Execution Log

```
− SUITE LoginTests                                                          00:00:04.092
    Full Name:              LoginTests
    Source:                 /System/Volumes/Data/RobotFiles/Buch/LoginTests.robot
    Start / End / Elapsed:  20220214 02:39:41.066 / 20220214 02:39:45.158 / 00:00:04.092
    Status:                 1 test total, 0 passed, 1 failed, 0 skipped
  − TEST Erfolgreiche Anmeldung                                             00:00:03.605
      Full Name:              LoginTests.Erfolgreiche Anmeldung
      Tags:                   positive, smoke
      Start / End / Elapsed:  20220214 02:39:41.551 / 20220214 02:39:45.156 / 00:00:03.605
      Status:                 FAIL
      Message:                Login nicht erfolgreich!
    + KEYWORD MeineKeywords.CarConfig Öffnen                                00:00:01.912
    + KEYWORD Benutzer Anmelden admin, 8270                                 00:00:00.133
    − KEYWORD MeineKeywords.Anmeldung Prüfen                                00:00:01.507
        Start / End / Elapsed:  20220214 02:39:43.600 / 20220214 02:39:45.107 / 00:00:01.507
      + KEYWORD Browser.Get Element States "Logout", contains, visible, message=Login nicht erfolgreich!   00:00:01.506
      + KEYWORD Browser.Get Url ==, ${URL}/list                             00:00:00.000
    + TEARDOWN MeineKeywords.CarConfig Beenden                              00:00:00.049
```

Source: Book "Keyword-Driven Testing" by Matthias Daigl / René Rohner

And how do I start working with the RFW?

The wide and monthly increasing distribution, interfaces to all common software systems, openness and expandability - **this is the future industry standard for the automation of functional regression tests!**

If you would now like to start automating test cases with the RFW, we recommend reading the chapter "7 Practice with Robot Framework" in the book "Keyword-Driven Testing"[2] (Note: This is not an affiliate link[3]) by Matthias Daigl and René Rohner (both employees at imbus, René has been Chairman of the RFW Foundation since 2022).

The following sources should enable the interested reader to get started quickly when working with the RFW:

- 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] The autor of this technical article has no advantage in purchasing this book

- Robot Framework Website: https://robotframework.org
- Robot Framework Language Server: https://marketplace.visualstudio.com/items?itemName=robocorp.robotframework-lsp
- Slack channel for the 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
- ISTQB® Certified Tester Advanced Level Test Automation Engineer course at imbus: https://www.imbus.de/akademie/istqb-advanced-test-automation-engineer
- Test automation with Robot Framework at 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 Part 5 "Keyword Driven Testing" in the Beuth Verlag web shop: https://www.beuth.de/de/norm/iso-iec-ieee-29119-5/266302684
- Book "Keyword-Driven Testing" by Matthias Daigl / René Rohner, available from dpunkt Verlag https://dpunkt.de/produkt/keyword-driven-testing/; the chapter "6.4.2 Framework 2: Robot Framework" kindly provided by dpunkt-Verlag can be found at: OpenBook Keyword-Driven Testing
- RoboSAPiens: Flyer Integration of SAP automation in complex systems

There is no substitute for experience except for more experience! If the reader does not have the time and leisure to gain all this experience themselves, they can exchange ideas with over 600 participants at the annual Robocon, ask questions to one of the over 14,000 members in the community chat (Slack) and receive training and advice from imbus on the introduction of the RFW (www.imbus.de , 09131, 7518 - 0, info@imbus.de).