

b4AWhite Paper

Test Automation –

Verbesserungen der Stabilität mit automatischen Tests

best-blu consulting with energy GmbH

Ihr Ansprechpartner:
Andreas Büsching

E-Mail: a.buesching@best-blu.de
Telefon: +49 421 491 811 80
Datum: 27. Mai 2025

Inhalt

Motivation.....	3
Einleitung	3
Spezifikation von Testdefinitionen	3
Ausführung von Tests	5
Fazit	7

Motivation

In den heutigen IT-Landschaften steigen insgesamt die Anforderungen und somit sind Standardisierungen in der Technik und den Prozessen dringend erforderlich. Dazu gehört auch der Bereich der Automatisierung. Dieser hat sich in vielen Umgebungen mit ihren Techniken und auch Prozessen von den anderen Teilen der IT-Landschaft unterschieden. Heute wird die Automatisierung in den Unternehmen immer wichtiger und somit ist auch der Bedarf für den Angleich an den Rest immer wichtiger. Außer Prozessen und Techniken aus dem Bereich des Deployment gehören auch andere Themen dazu. Das Gebiet, um das es in diesem Dokument gehen soll, ist die Testautomatisierung. Der Inhalt basiert auf einem Beitrag von Sebastian Weha, der Projekte in diesem Gebiet bereits bei mehreren Kunden betreut hat.

Einleitung

Im Bereich der Softwareentwicklung ist der Begriff *Software Development Lifecycle* (SDLC) weitverbreitet. Dieser beschreibt die Stationen von der Anforderungsanalyse bis zur fertigen Software und deren Evaluierung. Um den SDLC umzusetzen, gibt es viele Methoden – und viele Diskussionen darum, welche davon die beste ist. Darunter sind die bekanntesten Agile Softwareentwicklung, das Wasserfallprinzip und auch DevOps.

Eines haben alle diese Methoden gemeinsam. Sie legen großen Wert auf das Testen der Software. Dies lässt sich auch auf die Entwicklung in der Automatic Automation Engine abbilden – zumindest in der Theorie. In der Praxis wird das Testen hier aber eher stiefmütterlich behandelt, wie sich in vielen Projekten herausgestellt hat. Aus dieser Erfahrung heraus wurde ein Konzept entwickelt das Testen zu automatisieren und in best4Automatic (b4A) eine Lösung dafür bereitzustellen.

Die Lösung basiert auf b4A Packages, die sich an die Struktur der Automatic Action Packs anlehnen. In der Standard-Konfiguration wird die Struktur der Action Packs berücksichtigt, so können Kunden, die bereits Action Packs nutzen, sofort starten. Als Vorbereitung zur die Testautomatisierung ist es notwendig zu definieren wie die Tests durchgeführt und ausgewertet werden sollen. Dafür dienen Testdefinitionen. Da in der Softwareentwicklung für solche Fragestellungen bereits Lösungen existieren wird auf diese zurückgegriffen.

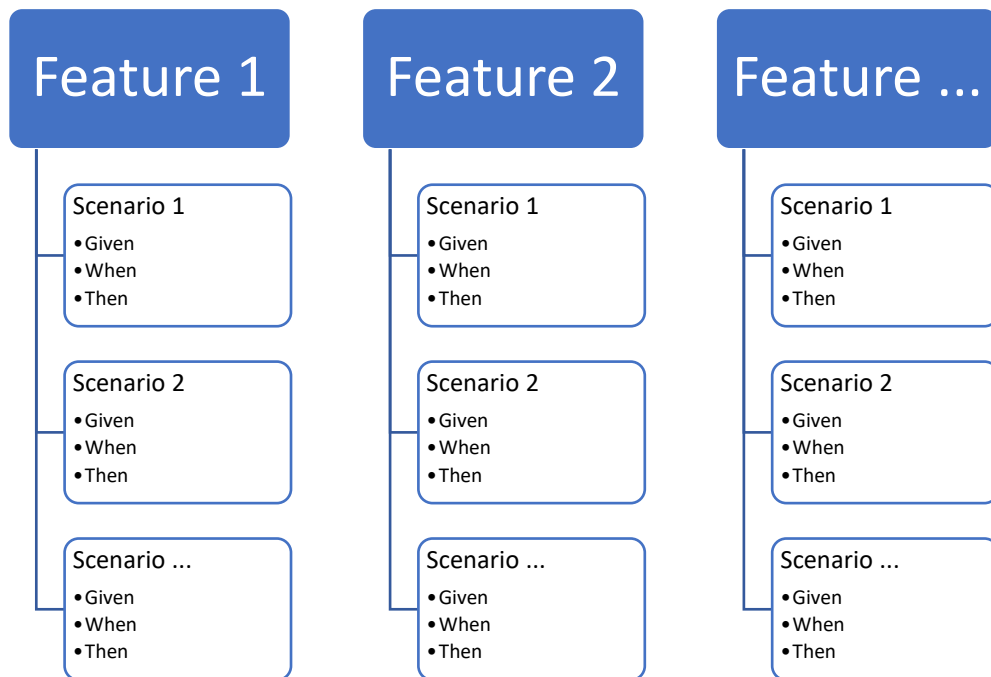
Spezifikation von Testdefinitionen

Die Testdefinitionen werden in der [Gherkin](#) Beschreibungssprache geschrieben. Diese Syntax ist weitverbreitet und wird von vielen sogenannten *Behavior-Driven-Development-Tools* genutzt. Dabei ist die Idee, die Tests in einer „natürlichen“ Schriftsprache zu notieren. Durch bestimmte Schlüsselwörter können diese Testdefinitionen automatisiert abgearbeitet werden.

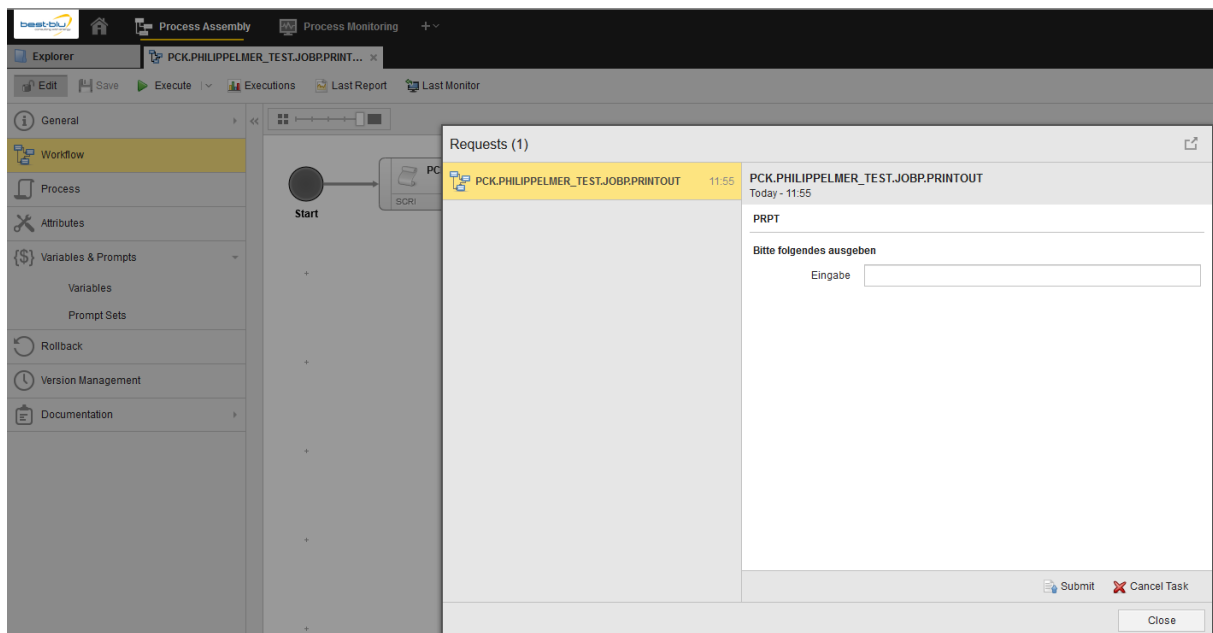
Grundsätzlich sind die Tests folgendermaßen aufgebaut:

- **Feature:** Beschreibt ein Set an Testfällen (Szenarien), die thematisch zusammenhängen
- **Scenario:** Beschreibt einen einzelnen Testfall
- **Given:** Beschreibt die Schritte, die vor der eigentlichen Testausführung überprüft werden (Vorbedingungen).
- **When:** Beschreibt die eigentliche Testausführung

- **Then:** Beschreibt die Schritte, die nach der Testausführung überprüft werden (Nachbedingungen).



Um dies praxisnäher zu erklären, wird im Folgenden ein einfacher Workflow erstellt. Dieser beinhaltet ein PromptSet zur Texteingabe und ein Skript, das die Eingabe wieder ausgibt.



Ein einfacher Workflow mit PromptSet

Die Funktion dieses Workflows wird mit der folgenden Testspezifikation geprüft.

```

white-paper.feature U
tests > white-paper.feature
1 @WHITE_PAPERS
2 Feature: Teste den Workflow für die White Paper Demo!
3
4 @PRINT_OUT @TEST0001
5 Scenario: Test Skript und überprüfe ob die Eingabe im Report ausgegeben wird.
6   Given Object PCK.WHITEPAPER_TEST.JOBP.PRINTOUT exists
7   And Object PCK.WHITEPAPER_TEST.SCRI.PRINTOUT exists
8   And Object PCK.WHITEPAPER_TEST.PRPT.PRINTOUT exists
9
10  When Execute PCK.WHITEPAPER_TEST.JOBP.PRINTOUT
11    | Key      | Value      |
12    | INPUT#    | Testeingabe |
13
14  Then Task status of PCK.WHITEPAPER_TEST.JOBP.PRINTOUT is
15    | Task                                | status |
16    | PCK.WHITEPAPER_TEST.SCRI.PRINTOUT(1) | ENDED_OK |
17
18  And Check report ACT of PCK.WHITEPAPER_TEST.SCRI.PRINTOUT
19    from PCK.WHITEPAPER_TEST.JOBP.PRINTOUT for ".*Testeingabe.*"

```

Die Spezifikation enthält ein **Feature** mit einem sehr einfachen **Szenario**.

Zuerst wird in der **GIVEN**-Phase geprüft, ob alle drei Objekte im Mandanten vorhanden sind. Ist eins nicht vorhanden, bricht der Test mit einem Fehler ab.

Die **WHEN**-Phase beschreibt die Ausführung des Workflows und übergibt die PromptSet-Parameter. In der Tabelle stehen die PromptSet-Parameter, die übergeben werden (Key: INPUT#, Value: Testeingabe).

Die **THEN**-Phase überprüft, ob der Task im Workflow den Status ENDED_OK hat und ob der Report den eingegebenen Text enthält.

Ausführung von Tests

Die *Gherkin*-Sprache führt Tests über das *Cucumber*-Framework aus und ruft von dort die Ergebnisse ab. Dieser Schritt ist nicht ganz trivial, da ein Interpreter benötigt wird, der Automatic in die erwarteten Ergebnisse und in eine für Cucumber verständliche Sprache übersetzt.

Diese Funktionalität ist Teil der best4Automatic Solution (Modul *ta.Execute*). Mit dem Modul lassen sich die in *Gherkin* spezifizierten Testfälle ausführen. Sie können entweder die Testspezifikation in einer Textdatei übergeben oder die Testspezifikation in einem DOCU-Objekt hinterlegen. In dem Beispiel wurde ein DOCU-Objekt genutzt. In diesem wird auch der Bericht zur Testausführung von dem Modul gespeichert (Objekt für Bericht). Dieser enthält sämtliche Details zu der Ausführung in einer XML-Struktur.

Konfiguration

Zugangsdaten Optionen Ausgabe Erweitert

Quelle ?

Objekt ▾

Cucumber-Definition (Objekt) ?

B4A.WHITE-PAPERS.DOCU.TESTS

Tags ?

Abbildung 1 - Festlegen der Quelle für die Testspezifikationen

Konfiguration

Zugangsdaten Optionen Ausgabe Erweitert

Objekt für strukturierte Dokumentation ?

B4A.WHITE-PAPERS.DOCU.TESTS

strukturierte Dokumentation zuvor leeren ☐ ?

Verbindung ?

▾ b4a b4a b4a

Name der strukturierten Dokumentation ?

XML-Variable ?

B4A.WHITE-PAPERS.VARA@XML.TESTS

Key ?

WhitePapers

Quellverbindung für die Variable ?

▾ b4a b4a b4a

b4A best for Atomic **Dokumentation** ▾

Abbildung 2 - Ausgabe des Berichts in eine XML-Variable und/oder eine strukturierte Dokumentation

Nach der Ausführung der Tests wird ein Bericht mit den Ergebnissen angezeigt.

Fortschritt	Bericht
Test Case Test Skript und überprüfe ob die Eingabe im Report ausgegeben wird.	
Ergebnis von Given Object B4A.WHITE-PAPERS.JOBP.PRINTOUT exists	
2021-10-26 12:59:58 Rückgabewert:Success 2021-10-26 12:59:58 Details: Objekt B4A.WHITE-PAPERS.JOBP.PRINTOUT existiert	
Ergebnis von And Object B4A.WHITE-PAPERS.SCRI.PRINTOUT exists	
2021-10-26 12:59:58 Rückgabewert:Success 2021-10-26 12:59:58 Details: Objekt B4A.WHITE-PAPERS.SCRI.PRINTOUT existiert	
Ergebnis von And Object B4A.WHITE-PAPERS.PRPT.PRINTOUT exists	
2021-10-26 12:59:58 Rückgabewert:Success 2021-10-26 12:59:58 Details: Objekt B4A.WHITE-PAPERS.PRPT.PRINTOUT existiert	
Ergebnis von When Execute B4A.WHITE-PAPERS.JOBP.PRINTOUT	
2021-10-26 12:59:59 Rückgabewert:Success 2021-10-26 12:59:59 Details: Das Objekt B4A.WHITE-PAPERS.JOBP.PRINTOUT wurde erfolgreich mit der Run-Id 1812699 aktiviert	
Ergebnis von Then Task status of B4A.WHITE-PAPERS.JOBP.PRINTOUT is	
2021-10-26 12:59:59 Rückgabewert:Success 2021-10-26 12:59:59 Details: Status-Code 1900 der Aufgabe B4A.WHITE-PAPERS.SCRI.PRINTOUT(2) passt zum Status ENDED_OK	
Ergebnis von And Check report ACT of B4A.WHITE-PAPERS.SCRI.PRINTOUT from B4A.WHITE-PAPERS.JOBP.PRINTOUT for ".*Testeingabe.*"	
2021-10-26 12:59:59 Rückgabewert:Success 2021-10-26 12:59:59 Details: Das Muster '.*Testeingabe.*' wurde im Report ACT gefunden (Seite 1) 2021-10-26 12:59:59 Test Case Ausführung ist erfolgreich beendet 2021-10-26 13:00:00 Verbindung zu AE123A(0011) geschlossen 2021-10-26 13:00:00 erfolgreich: 6, fehlgeschlagen: 0, ignoriert: 0	


[Dokumentation](#)

Konfiguration ...
Beenden

Der Test wurde erfolgreich ausgeführt und im Aktivierungsreport des SCRI wurde der Text „Testeingabe“ gefunden.

Der hier gezeigten Testbericht befindet sich als XML-Dokument auch in dem Objekt, das per Option festgelegt wurde. Dadurch können die Testergebnisse mit Automic-Skriptmitteln analysiert und weiterverarbeitet werden. Beispielsweise können die Testergebnisse als E-Mail weiterleitet oder in eine Testmanagementsoftware übertragen werden.

Die best4Automic Test Automation unterstützt mittlerweile um die 40 verschiedenen Testschritte in den drei Phasen, von denen viele in unterschiedlichen Ausprägungen genutzt werden können. Zusätzlich können die Testspezifikationen parametrisiert werden, so dass diese auch flexible genutzt werden können.

Fazit

Mit der best4Automic Test Automation ist es möglich Testspezifikationen in einer weitverbreiteten Sprache zu schreiben und diese automatisiert durchzuführen und auszuwerten. Somit haben automatische Tests ausschließlich einen initialen Aufwand für die Erstellung der Spezifikationen und können dann automatisiert durchgeführt werden, um so die Stabilität der Automation deutlich zu verbessern.