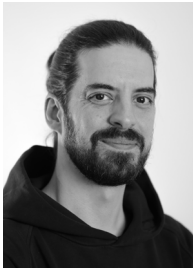
Matthias Daigl · René Rohner

Keyword-Driven Testing

Grundlage für effiziente Testspezifikation und Automatisierung

dpunkt.verlag

Matthias Daigl ist Product Owner bei der imbus AG. Er ist als Sprecher auf internationalen Konferenzen unterwegs, arbeitet in Arbeitsgruppen des German Testing Board, des ISTQB® und im Normungsausschuss von DIN und ISO mit, war Editor der Norm ISO/IEC/IEEE 29119-5 »Keyword-Driven Testing« und ist Autor des Buches »ISO 29119: Die Softwaretest-Normen verstehen und anwenden«.



René Rohner ist Product Owner des Value Streams Testautomatisierung sowie Senior Berater mit den Spezialgebieten Keyword-Driven Testing und Testautomatisierung bei der imbus AG. Er ist als Softwareentwickler, Trainer und Chairman of the Board der Robot Framework® Foundation international im Bereich des Keyword-Driven Testing tätig.

Matthias Daigl · René Rohner

Keyword-Driven Testing

Grundlage für effiziente Testspezifikation und Automatisierung

dpunkt.verlag

Matthias Daigl
KDT@daigl.de

René Rohner
buch@keyword-driven.de

Lektorat: Christa Preisendanz
Lektoratsassistenz: Julia Griebel
Copy-Editing: Ursula Zimpfer, Herrenberg
Satz: Matthias Daigl
Herstellung: Stefanie Weidner, Frank Heidt
Umschlaggestaltung: Helmut Kraus, *www.exclam.de*
Druck und Bindung: mediaprint solutions GmbH, 33100 Paderborn

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über *http://dnb.d-nb.de* abrufbar.

ISBN:
Print 978-3-86490-570-4
PDF 978-3-96088-482-8
ePub 978-3-96088-483-5
mobi 978-3-96088-484-2

Wieblinger Weg 17
69123 Heidelberg

Hinweis:
Dieses Buch wurde auf PEFC-zertifiziertem Papier aus nachhaltiger Waldwirtschaft gedruckt. Der Umwelt zuliebe verzichten wir zusätzlich auf die Einschweißfolie.

PEFC zertifiziert
Das Papier für dieses Buch stammt aus nachhaltig bewirtschafteten Wäldern und kontrollierten Quellen.
PEFC
PEFC/04-31-0810
www.pefc.de

Schreiben Sie uns:
Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen: *hallo@dpunkt.de*.

5 4 3 2 1 0

Vorwort

In Gesprächen mit anderen Softwaretesterinnen und -testern haben wir die Erfahrung gemacht, dass das Konzept »schlüsselwortbasierter Test« oder »Keyword-Driven Testing« vielen ein Begriff ist. Andererseits ist es uns öfter so gegangen, wenn jemand darüber sprach, dass uns der Gedanke kam – *ja, das ist es auch, aber nicht nur …*

Im Kern geht es darum, Tests besser zu formulieren, für Mensch und Maschine.

Hinter dem Begriff »Keyword-Driven Testing« verbirgt sich dabei mehr, als es auf den ersten Blick scheint. Auch macht es zunächst den Eindruck, dass es ganz einfach ist. Gleichzeitig hat es viele Facetten und bietet zahlreiche Möglichkeiten, die im Alltag beim Softwaretest helfen können, aber einfach nicht bekannt genug sind.

Vieles davon ließe sich auf die Formel bringen: »Nutzt bewährte Praktiken aus der Softwareentwicklung und nutzt imperative Programmierung, um Tester und Testerinnen sowie Testautomaten die Testanweisungen zu übermitteln. Haltet euch dabei an die allgemeingültigen Best Practices, was Lesbarkeit und Verständlichkeit angeht.«

Natürlich ist es nicht so einfach. Was wir über Keyword-Driven Testing wissen, haben wir über Jahrzehnte gelernt. Inzwischen haben wir das Thema in der Theorie vertreten, auf Ebene von ISO-Normen, und in der Praxis in einer Vielzahl von Beratungsprojekten Teams beim Einstieg in Keyword-Driven Testing unterstützt. Weitere Interessierte können wir jedoch nicht persönlich erreichen und gleichzeitig gibt es einfach noch zu wenig Literatur zum Thema.

Auftrag angenommen: Keyword-Driven Testing muss bekannter und gut aufbereitet zugänglicher werden.

Sie halten nun unseres Wissens das erste Buch in den Händen, das sich ausschließlich mit dem Thema »Keyword-Driven Testing« befasst. Wir hoffen, dass Sie darin viele Anregungen finden, Ihre Tests effizienter und wartbarer zu gestalten. Viel Erfolg damit!

Erlangen, im Februar 2022
Matthias Daigl

Königswinter, im Februar 2022
René Rohner

Dank

Unser Dank gilt an dieser Stelle den Kolleginnen und Kollegen sowie Personen aus dem Freundeskreis, die uns beim Schreiben dieses Buches zur Seite gestanden sind. In vielen Gesprächen haben sie sich Zeit für uns und unser Vorhaben genommen, ihre Erfahrungen und Ideen mit uns geteilt und uns Inspiration geschenkt.

Ganz herzlich danken wollen wir auch dem dpunkt.team und unserer Lektorin, Christa Preisendanz, für die schier unendliche Geduld mit uns. Wir haben es ihnen nicht leicht gemacht.

Am meisten wollen wir aber unseren Familien danken, die uns jederzeit unterstützt haben, oft in den Schreibphasen auf uns verzichten mussten und uns wieder aufgerichtet haben, wenn es nicht vorangehen wollte. Ohne Euch hätten wir das nicht geschafft!

Inhaltsverzeichnis

1 Einführung

Dieses Buch hat das Ziel, das Thema »Keyword-Driven Testing« möglichst umfassend zu beleuchten, unter anderem, damit Keyword-Driven Testing häufiger und intensiver eingesetzt wird. Aber was verstehen wir überhaupt unter Keyword-Driven Testing?

Keyword-Driven Testing ist eine Methode zur Spezifikation von Testfällen im Sinne einer Notation. Es macht Testen effizienter. Es entfaltet seinen Nutzen vor allen Dingen in Teams, hilft aber auch einzelnen Testerinnen und Testern. Und entgegen anderslautender Gerüchte ist »Keyword-Driven Testing« nicht nur für Testautomatisierung hilfreich – aber gerade auch dort.

Das erwartet Sie nun in den einzelnen Kapiteln:

Überblick über das Buch

Einführung – dieses Kapitel: Was »Keyword-Driven Testing« ist, warum man es nutzen sollte, Begriffe und die Einführung eines Beispiels, auf das sich das Buch im weiteren Verlauf bezieht. *Ergebnis: Sie wissen, worum es geht.*

Konzepte: Unterschiedliche (einfache und komplexe) Ansätze, mit denen »Keyword-Driven Testing« genutzt werden kann. *Ergebnis: Sie wissen, was Keyword-Driven Testing ist, und kennen Zusammenhänge.*

Umsetzung: Wie man Keywords auswählt, wie man sie strukturieren kann, was es aus sprachlicher Sicht zu beachten gilt und wie man Qualitätssicherung für Keywords betreiben kann. Und was das wirtschaftlich bedeutet. *Ergebnis: Sie wissen, wie man mit Keywords umgeht.*

Keywords und Normen: Was für Normen es im Testen und speziell zu Keywords gibt und was die Normen zu Keywords und Frameworks sagen. *Ergebnis: Sie wissen, wie Ihnen Normen zu Keyword-Driven Testing helfen können.*

Testautomatisierungsarchitektur: Was so alles gebraucht wird, wenn man Testautomatisierung und Keyword-Driven Testing betreiben möchten, und auch wie ISTQB®dazu steht. *Ergebnis: Sie kennen verschiedene Sichtweisen auf die Testautomatisierungsarchitektur und verfügen über die Grundlagen, sich Ihre Architektur im Hinblick auf Keyword-Driven Testing zu erarbeiten.*

Keyword-Driven Testing Frameworks: Praxisbeispiele, wie Keyword-Driven Testing in Projekten angewendet wird, einschließlich einer Bewertung der eingesetzten Frameworks. *Ergebnis: Sie haben einen Eindruck von unterschiedlichen Einsatzszenarien von Keyword-Driven Testing.*

Praxis mit Robot Framework: Anhand eines konkreten Frameworks zeigen wir, wie eine Umsetzung von Keyword-Driven Testing im Kontext aussehen kann. *Ergebnis: Sie können die Anwendung von Keyword-Driven Testing mit diesem Framework nachvollziehen.*

Brückenschlag: Verbindung von Keyword-Driven Testing mit verbreiteten Ansätzen im Softwaretest, die zwar nicht »Keyword-Driven Testing« im engeren Sinne sind, aber gut damit zusammenwirken. *Ergebnis: Sie können Keyword-Driven Testing im Hinblick auf diese Ansätze einordnen.*

Ausblick: Wie geht es weiter mit Keyword-Driven Testing. *Ergebnis: Sie haben das Buch geschafft und wissen nun, wie Sie mit Keyword-Driven Testing umgehen wollen!*

1.1 Wortwahl

Bevor wir mit dem Inhaltlichen beginnen, gibt es noch zwei Dinge bezüglich der Wortwahl, die uns Autoren am Herzen liegen und die wir daher hier kurz darlegen wollen:

Geschlechtsspezifische Begriffe: In diesem Buch werden wir häufig darüber reden, was eine Person – z.B. ein Tester oder eine Testerin – tut, denkt, tun kann ... Wir sind der festen Überzeugung, dass das Geschlecht einer Person grundsätzlich keine Rolle zu spielen hat. Dennoch leben wir im Bereich der Softwareentwicklung und des Softwaretests in einer stark männerdominierten Branche. Gerade deswegen ist uns eine Geschlechtergleichstellung auch besonders in sprachlicher Form wichtig.

Zugunsten einer besseren Lesbarkeit werden wir nicht jedes Mal beide sprachlichen Geschlechter aufzählen. Auch haben wir uns gegen eine Schreibweise wie »TesterIn« oder »Tester*in« entschieden.

Das generisches Maskulinum, also nur die männliche Form, zu verwenden, schied völlig aus.

Unser Weg ist es nun, für die in diesem Buch wichtigsten sechs Rollen uns jeweils zu gleichen Teilen für die weibliche und die männliche Form zu entscheiden und dann zugunsten der Lesbarkeit dabei zu bleiben – in der Hoffnung, dass jede und jeder sich so ausreichend angesprochen und wertgeschätzt fühlt.

Wir reden also von Testerinnen, Testmanagerinnen und Entwicklerinnen sowie von Testdesignern, Anwendern und Testautomatisierern.

Englisch – Deutsch: Dies ist ein deutschsprachiges Buch und im Allgemeinen streben wir an, Anglizismen zu vermeiden. Ausgerechnet für den Schlüsselbegriff dieses Buches – »Keyword-Driven Testing« – gibt es zwar mit »schlüsselwortbasiertem Test« eine deutsche Entsprechung, diese ist aber nach unserem Empfinden so wenig gebräuchlich, dass wir uns entschieden haben, innerhalb des Buches dennoch den englischen Begriff »Keyword-Driven Testing« zu bevorzugen. Gleiches gilt für die Verwendung von »Keyword« und »Schlüsselwort«. Die Germanisten mögen uns verzeihen.

1.2 Was ist Keyword-Driven Testing

Was verbirgt sich nun hinter dem Begriff »Keyword-Driven Testing«?

Es gibt dazu einige Analogien – ein Favorit: Testfälle werden aus Bausteinen zusammengesetzt. Wir alle erinnern uns gerne an unsere Kindheit. Die Vorstellung, Testfälle spielerisch zusammenzusetzen, ist daher bestimmt attraktiv.

Modularisierung

Große Dinge aus kleinen Bausteinen zusammenzusetzen ist aber jenseits vom spielerischen Aspekt eine generell bewährte Vorgehensweise, um Komplexität zu bewältigen. Wir sprechen hier von Modularisierung.

Während Modularisierung in der Softwareentwicklung im Allgemeinen selbstverständlich ist, und das vielleicht auch noch bei Testautomatisierung zutrifft, so ist sie im Zusammenhang mit Testspezifikation bei Weitem nicht so häufig anzutreffen.

Keyword-Driven Testing (KDT) ist eine Methode, um Modularisierung bei der Spezifikation von Testfällen zu erreichen.

Ein Keyword ist in diesem Sinne ein Baustein, aus dem Testfälle zusammengesetzt werden. Dabei steht das Keyword – typischerweise ein Verb zusammen mit einem Substantiv, beispielsweise `Erzeuge User` – für eine oder mehrere Aktivitäten, die bei der Testausführung später auszuführen sind.

Idealerweise stehen beim Erstellen der Testfälle alle benötigten Keywords fertig definiert zur Verfügung. Dann können alle Testfälle aus Keywords zusammengesetzt werden.

Keine Testtechnik

Unter »Testtechnik« versteht man meist »Testentwurfsverfahren«. Laut ISO/IEC 29119 [48] geht es dabei darum, auf einem definierten Weg Testfälle zu ermitteln (einem Prozess folgend).

Keyword-Driven Testing ist also keine Testtechnik in diesem Sinne. Unabhängig von Keyword-Driven Testing setzen wir für das Ableiten von Testfällen, also das Testdesign, voraus, dass eine Testtechnik (oder mehrere) eingesetzt werden.

Keyword-Driven Testing legt jedoch fest, wie die aus dem Testdesign entstandenen Testfälle aufgeschrieben oder dokumentiert werden.

1.3 Begriffe

Nur die beiden wichtigsten zentralen Begriffe – Keyword und Framework –, die in diesem Buch verwendet werden, sollen hier definiert und erläutert werden. Umfassendere Begriffssammlungen finden sich auf der Webseite von imbus [23] sowie auf der Seite des ISTQB® [32] (als Online-Anwendung), und von IEEE sind die Begriffe von IEEE, ISO, IEC und PMI veröffentlicht und als Norm ISO/IEC/IEEE 2765 [50] käuflich zu erwerben, aber auch im Web legal frei verfügbar [28].

1.3.1 Der Begriff »Keyword«

Der zentrale Begriff dieses Buches ist ohne Zweifel »Keyword« oder zu Deutsch »Schlüsselwort«.

Mehrdeutigkeit

Der Begriff »Keyword« selbst ist eigentlich ein Teekesselchen[1]: Man versteht darunter, je nach Zusammenhang, die Bezeichnung (im Beispiel oben »Erzeuge User«), die dahinter liegenden Aktivitäten (könnte hier das Auswählen eines Menüeintrags sein, dann das Ausfüllen der Felder eines Dialoges ...) oder ein vielleicht vorhandenes zugeordnetes Automatisierungsskript. Oder dies alles gemeinsam.

[1] Ein »Teekesselchen« – das Fachwort wäre Homonym – ist ein Begriff, der mehrere Bedeutungen hat. In einem Kinderspiel lässt man raten: »Mein Teekesselchen kann schwimmen oder man braucht es zum Zelten – was ist das?« – Antwort: ein Hering.

In der Norm ISO 29119-5 [49] ist dieser Begriff folgendermaßen definiert:

Keyword	Schlüsselwort
one or more words used as a reference to a specific set of actions intended to be performed during the execution of one or more test cases	ein oder mehrere Wörter, die als Verweis auf eine festgelegte Menge von Aktivitäten verwendet werden und die während der Durchführung eines oder mehrerer Testfälle ausgeführt werden sollen
(Definition laut [49])	(Übersetzung der Autoren)

Ein Keyword besteht demnach aus ein oder mehreren Wörtern, die beschreiben, was an der Stelle, an der diese Wörter in einem Testfall vorkommen, getan werden soll. Ein Beispiel für ein einzelnes Wort könnte `Login` sein, eines für mehrere Wörter `Datensatz anlegen`. Meist wird ein Keyword ein Verb enthalten (so fordert es auch die ISO 29119-5), allein schon deswegen, weil ja durch das Keyword eine auszuführende Tätigkeit beschrieben ist.

Auch wenn es rein technisch gesehen nicht notwendig ist, Keywords mit einem Verb zu bilden, so hat es sich doch bewährt. Die Lesbarkeit profitiert davon, und daher ist es empfehlenswert, sich nach dieser Regel zu richten.

Um zu der Begriffsfrage zurückzukehren: Testautomatisierer verstehen unter einem »Keyword« gerne eine aufrufbare Funktion oder ein Skript, die bzw. das im Sourcecode in einem Testautomatisierungswerkzeug implementiert ist. Testerinnen denken dagegen vielleicht eher an eine Zusammenfassung mehrerer Testschritte zu einer fachlichen Tätigkeit.

Ein Keyword kann beides sein.

Ein Keyword kann eine automatisierte Funktion widerspiegeln oder manuelle Testschritte zusammenfassen. Und: Ein Keyword kann sogar eine Zusammenfassung anderer Keywords sein.

Häufig wird unter dem Begriff »Keyword« einfach der Name des Keywords (hier: `Benutzer anmelden`) verstanden – der ja im Alltag, beim Spezifizieren von Tests, auch am meisten gebraucht wird. Im Hinterkopf sollte man aber behalten, dass dazu einiges mehr gehört.

In Abschnitt 2.1 »Verschlagwortung« werden wir auf die sprachlichen Aspekte, die hier nur angerissen wurden, noch sehr viel genauer eingehen.

1.3.2 Der Begriff »Framework«

Framework

Den Begriff »Framework« benutzen wir in diesem Buch bezogen auf Keyword-Driven Testing. Alleine stehend ist er ein zu gängiges Wort der natürlichen englischen Sprache, als dass es einer ISO oder IEEE in den Sinn kommen würde, dafür eine Definition anzugeben. Im Zusammenhang mit Testautomatisierung bietet jedoch das ISTQB®eine Definition, dankenswerterweise sowohl auf englisch als auch auf deutsch:

Test automation framework	*Testautomatisierungsframework*
A tool that provides an environment for test automation. It usually includes a test harness and test libraries.	Ein Werkzeug, das eine Umgebung zur Testautomatisierung bereitstellt. Es beinhaltet üblicherweise einen Testrahmen und Testbibliotheken.
(Definition laut [32] – englisch)	(Definition laut [32] – deutsch)

Auch in diesem Buch steht der Begriff »Framework« im Zusammenhang mit Testautomatisierung. Allerdings ist der Kontext noch etwas spezifischer. Unter »Keyword-Driven Testing Framework« verstehen wir hier die Gesamtheit der Werkzeuge, Skripte und Bibliotheken, die als Grundlagen für eine bestimmte Variante von Keyword-Driven Testing – meist mit Fokus auf Automatisierung – benötigt und verwendet werden.

Mehr zum Thema Framework, was alles dazugehört und was man davon erwarten kann, findet sich in Abschnitt 4.3 (Sichtweise der Norm ISO 29119-5) und vor allem in Kapitel 6 »Keyword-Driven Testing Frameworks«.

1.4 Keywords unter der Lupe

Elementare Eigenschaften von Keywords

Um uns Keyword-Driven Testing weiter anzunähern, wollen wir uns in den folgenden Abschnitten Keywords genauer ansehen. Erst einmal geht es um die elementaren Eigenschaften – was ist ein Keyword, was gehört auf jeden Fall dazu? Und dann geht es um einen ganz praktischen, wichtigen Aspekt, nämlich die Struktur, die ein Keyword haben kann.

Verschlagwortung: Eines der Grundkonzepte hinter Keyword-Driven Testing ist, dass Tests nicht als Prosatext verfasst werden, sondern in einzelne, klar voneinander abgegrenzte, sehr kurze Testschritte gegliedert werden. Jeder dieser Testschritte soll keine komplette Beschreibung der Tätigkeit sein, sondern nur ein kurzer Befehl, der

die entsprechende Tätigkeit zusammenfasst. Daher rührt der Begriff »Keyword«. Meist reicht es nicht aus, sich auf ein einzelnes Wort als Benennung für ein Keyword zu beschränken. Mehrere Wörter sind also in Ordnung. Als Faustformel für die Länge der Keywords kann man von 1-6 Wörtern ausgehen.

1 Keyword ≈ 1-6 Wörter

Kurze Anweisungen von bis zu sechs Wörtern können von einem Leser schnell erfasst und verstanden werden. Oft reicht einer Testerin ein einziger Blick auf eine solche Anweisung, um zu wissen, was zu tun ist.

Einige Beispiele für Keywords:

```
Webbrowser starten
Benutzerverwaltung öffnen
Benutzer an Windows anmelden
Device über WLAN verbinden
Mobilgerät entsperren
```

Beschreibung: Keyword-Driven Testing würde für manuelles Testen nicht funktionieren, wenn man sich ausschließlich auf die sprechenden Namen verlassen würde. Bestenfalls könnte man dann erwarten, dass mit den Keywords sehr vertraute Testerinnen sie mühelos und zuverlässig ausführen.

Neue Kolleginnen oder Testerinnen, die die Testspezifikation nur auszuführen haben, aber die Schritte (in Form von Keywords) vorher nicht kennen, könnten nicht wissen, was der Testdesigner von ihnen genau erwartet, und die Tests demnach auch nicht präzise ausführen. Zu der Definition eines Keywords gehört daher obligatorisch auch ein beschreibender Text.

Auch im Hinblick auf Automatisierung ist man sehr gut beraten, eine Dokumentation für das Keyword zu erfassen. Schließlich stellen Keywords hier die Schnittstelle zwischen fachlicher Sicht und technischer Sicht dar. Testautomatisierer müssen genau wissen, was sie implementieren sollen.

Bei der Keyword-Dokumentation geht es nicht nur darum, zu beschreiben, was das Keyword tun soll, sondern auch, wie es zu verwenden ist.

Parameter: Eine Interaktion zwischen einer Testerin und dem Testobjekt ist oft von Daten getrieben oder wird anhand von Daten verändert. Daher können Keywords in ihrer Verwendung ebenfalls über Testdaten variiert werden.

Die Benutzeranmeldung an einem System wird beispielsweise immer auf die gleiche Art und Weise durchgeführt. Je nach Benutzerdaten, die für die Anmeldung verwendet werden, unterscheidet sich das Verhalten des Testobjektes.

Diese Daten eines Keywords werden üblicherweise als Parameter oder Argumente bezeichnet. Parameter sind typischerweise im Keyword-Driven Testing dem eigentlichen Keyword angehängt.

Ob dieses Anhängen bzw. Trennen vom Keyword selbst wie im folgenden Beispiel mit zwei oder mehr Leerstellen oder in Tabellen mit mehreren Spalten oder wie in der Programmierung üblich in Klammern erfolgt, ist je nach Tool unterschiedlich und spielt keine Rolle.

Im Folgenden sehen wir ein Beispiel mit einigen Parametern:

```
Webbrowser starten        www.keyword-driven.de
Benutzername eingeben     admin
Passwort eingeben         123456
Anmelden klicken
Benutzerverwaltung öffnen
Device mit WLAN verbinden     Phone2     GuestWifi
Mobilgerät entsperren
```

Mehr zu diesem Thema findet sich in Abschnitt 2.3 »Data-Driven Testing«.

Keyword-Struktur: Ein wichtiger Aspekt bei Keyword-Driven Testing ist, dass Keywords »strukturiert« sein können. Wir meinen damit, dass Keywords aus kleineren Keywords zusammengebaut sein können – oder andersherum: Keywords können zu größeren Keywords zusammengefügt werden. Diese Eigenschaft von Keywords macht man sich zunutze, um Wiederverwendbarkeit und Wartbarkeit zu steigern.

Ein Beispiel: Die Anmeldung eines Benutzers am System besteht aus der Eingabe von Benutzername und Passwort und dem Klicken des »Anmelden«-Knopfes. Nehmen wir also an, dass `Benutzer anmelden` ein strukturiertes Keyword ist und aus drei Einzelschritten besteht:

```
Benutzer anmelden     Benutzer     Passwort
    Benutzername eingeben     Benutzer
    Passwort eingeben         Passwort
    Anmelden klicken
```

Ein Testdesigner braucht nun bei einem Test, der die Anmeldung benötigt, nicht immer wieder die drei Einzelschritte zu nutzen, sondern erfordert nur das eine strukturierte Keyword.

Das Thema der Struktur von Keywords greifen wir in Abschnitt 2.2 im Detail auf.

1.5 Evolution der Testautomatisierung

In einem Vortrag über Testautomatisierungsarchitekturen im Rahmen der Veranstaltung »Trends in Testing« im Jahr 2010 [13] wurde eine Evolution von Architekturen zur Testautomatisierung beschrieben.

In dieser »Evolution« hat auch Keyword-Driven Testing seinen Platz.

Vorsicht Glatteis ...

Keyword-Driven Testing wird von manchen als Königsweg der Testautomatisierung betrachtet.

An dieser Stelle über Testautomatisierungsarchitekturen zu sprechen könnte den Eindruck erwecken, dass es in diesem Buch auch so gesehen wird. Daher zur Klarstellung:

- Keyword-Driven Testing ist wunderbar geeignet für Testautomatisierung, aber:
- Keyword-Driven Testing hat auch viele Vorteile bei manuellem Test!

Eine Beschränkung des Keyword-Driven Testing auf Testautomatisierung wäre Verschwendung.

Abbildung 1-1 bringt zum Ausdruck, wie mit der Entwicklung der Testautomatisierungsarchitekturen, von Capture & Replay über datengetriebenen Test und Keyword-Driven Testing bis hin zu generierten Tests, ein Reifegrad verbunden wird.

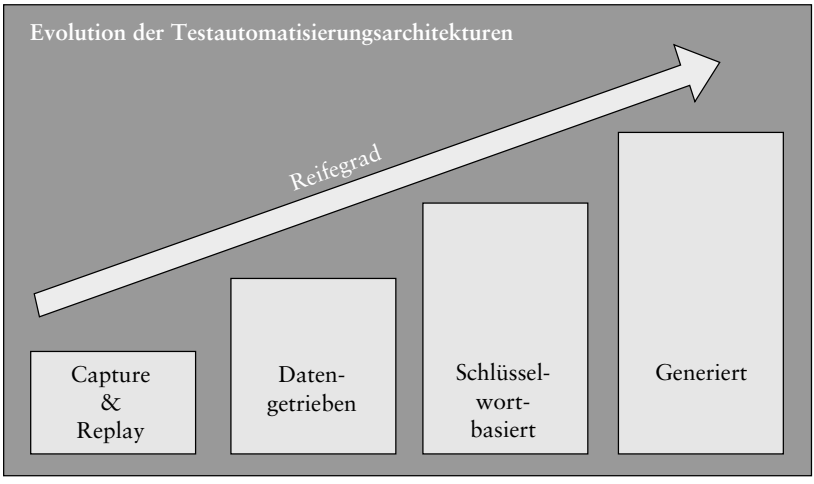


Abb. 1-1 *Evolution der Test-automatisierungs-architekturen*

Nach der Einführung von Testautomatisierung in einer Organisation entwickeln sich durch die sukzessive Gewinnung von Erfahrungen der Testerinnen sowohl die Ansprüche an die Umsetzung der Testautomatisierung als auch die Leistungsfähigkeit der Architektur typischerweise in folgenden Phasen – immer vorausgesetzt, externe Einflüsse sorgen nicht dafür, dass diese Phasen direkt übersprungen werden:

Wartungsproblem

1. Capture & Replay: Automatisierte Testfälle werden als Skripte aufgezeichnet, dafür wird eine entsprechende »Rekorder«-Funktion des Werkzeugs genutzt. Das Resultat sind automatisierte Testskripte, die einfach strukturiert sind, in der Regel mit der aktuellen Version des Testobjektes funktionieren (aber schon bei kleinen Änderungen am Test Interface Probleme bekommen können), die untereinander redundant sind (gleiche Aktionen werden in jedem Skript erneut erfasst) und, wie sich meist schnell herausstellt, einfach ein Albtraum im Hinblick auf Wartbarkeit darstellen.

   Abbildung 1-2 veranschaulicht einen solchen Fall: Eine Testspezifikation mag vorhanden sein und enthält sowohl Testlogik als auch Daten. Die Verknüpfung zur Testautomatisierung ist nur lose. Ein Testskript liegt individuell für jeden Testfall vor und wird aus einem kaum trennbaren Mix aus Code, Logik und Daten gebildet. Eine Änderung, egal aus welchem Grund, betrifft immer alles, und Fehler passieren leicht.

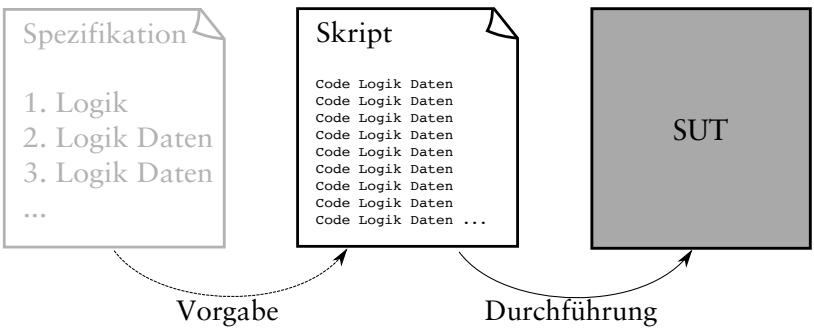


Abb. 1-2 *Monolithisches Testskript*

   Also sucht man schnell nach einem besseren Weg.

2. Datengetriebener Test: Während man durch Abkehr vom reinen Aufzeichnen der automatisierten Testskripte schon erste Vorteile in Richtung besserer Wartbarkeit erzielt, kommt die Erkenntnis, dass die Mühe, die man mit der Wartung hat, auch zwischen Testfällen geteilt werden kann. Mehrere strukturell identische Testfälle werden durch ein einziges, verallgemeinertes Skript implementiert und die Unterschiede zwischen den Testfällen, die in den Daten liegen, in

Trennung von Code und Daten

von den Testskripten getrennten Datentabellen ausgelagert (mehr dazu in Abschnitt 2.3).

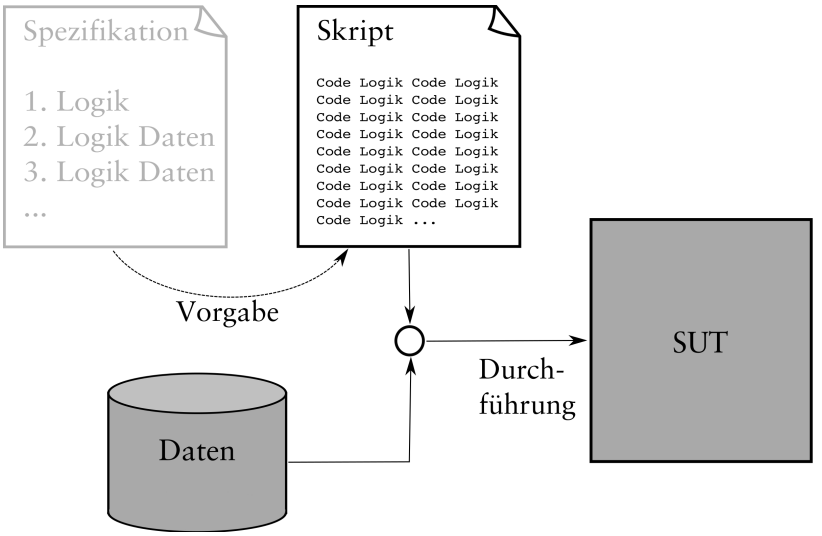


Abb. 1-3
Trennung von Skript und Daten

In Abbildung 1-3 sind strukturelle Vorteile dieses Ansatzes gegenüber den monolithischen Skripten aus Abbildung 1-2 zu erkennen: Die Kopplung zur Spezifikation ist noch lose (Anmerkung: Daten könnten an dieser Stelle bereits aus der Spezifikation ausgelagert sein), das Chaos in der Testimplementierung ist aber schon geringer, denn es sind nur noch Code und Logik im Skript miteinander vermischt. Änderungen, die nur die Daten betreffen, können unabhängig vom Rest durchgeführt werden, und das kann in vielen Fällen die Änderung sein, die benötigt wird.

Mehr noch: Es muss für eine Anzahl von fünf, zehn oder hundert Testfällen nur noch ein einziges automatisiertes Skript gewartet werden – ein großer Schritt in Richtung eines besseren Kosten-Nutzen-Verhältnisses.

3. Keyword-Driven Testing: Das, was mit datengetriebenem Test begonnen wurde, nämlich die Trennung von Code und Daten, wird beim Keyword-Driven Testing konsequent weitergeführt: Die bisher im Code verankerte Testlogik (der abstrakte Testfall) wird nun getrennt vom (Automatisierungs-)Code abgelegt.

Trennung von Logik und Code

Abbildung 1-4 zeigt, dass sich damit massive Änderungen (im Sinne von Verbesserungen) ergeben: Die Spezifikation ist ins Zentrum gerückt und enthält die reine Testlogik. Zur Spezifikation werden Keywords genutzt, denen in einem Repository entsprechende Skripte zugeordnet sind. Hier liegt also der Code. Die drei Aspekte Logik, Code und Daten werden zur Durchführung zusammengeführt.

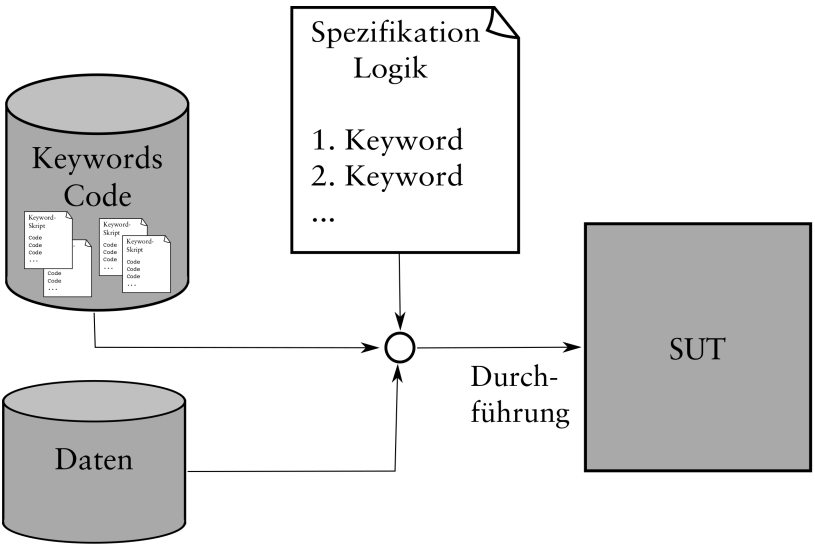


Abb. 1-4 Trennung von Skript, Daten und Logik

Jegliche Wartungstätigkeiten können jetzt unabhängig voneinander an Logik, Code und Daten erfolgen. Braucht man eines davon nicht zu ändern, dann muss man es auch nicht anfassen.

> Was hier passiert, die Trennung von Code und Daten, ist ein uraltes Prinzip der Informatik[2] und in der Softwareentwicklung generell ein alter Hut.
>
> Aber zu der Erkenntnis, dass die Entwicklung von Testautomatisierung die gleichen Prinzipien wie Softwareentwicklung erfordert, muss man ja erst gelangen – zunächst sieht alles so einfach aus.

4. Generierte Testfälle: Diese Entwicklungsstufe wird nach heutigem Stand nicht allerorts erreicht und muss auch nicht das Ziel sein. Hier geht es um Model-Based Testing (modellbasierten Test), also darum, Testfälle aus formal beschriebenen Modellen mechanisch abzuleiten. Keywords können dafür eine Grundlage sein.

Modellbasierter Test

Was mit diesem Ansatz vorangetrieben wird, ist der Grad der Automatisierung, die nicht mehr nur die Durchführung der Tests,

[2] Und nicht nur dort: In der Rechnerarchitektur ist dieses Prinzip unter dem Begriff »Harvard-Architektur« schon seit dem Mark II bekannt – wenn auch hier eher aus Gründen der Performanz als aus Gründen der Strukturierung und Wiederverwendung.

sondern auch die Erstellung der Testfälle beinhaltet. Die Wartung verlagert sich weg von der Automatisierung und hin zu den Modellen.

Auch wenn Keyword-Driven Testing nach diesem Modell nicht die Spitze der Evolution darstellt – es sieht doch so aus, als wäre im Zuge der Entwicklung von Praktiken zur Testautomatisierung KDT unausweichlich. Aber mehr noch: KDT passt auch ausgezeichnet zu Model-Based Testing; das greifen wir später in Abschnitt 2.6 wieder auf.

1.6 Vorteile des Keyword-Driven Testing

Mit dem bisher Gesagten haben wir eine Vorstellung davon vermittelt, was Keyword-Driven Testing ist, aber uns nicht weiter damit beschäftigt, warum es eingesetzt werden sollte. Immerhin liegt es auf der Hand, dass Keyword-Driven Testing mit einem gewissen Aufwand verbunden ist – zumindest müssen Keywords ja definiert und dokumentiert werden. Das Konzept und den damit verbundenen Aufwand können wir nicht alleine mit einem angeblich höheren Reifegrad rechtfertigen.

Im Folgenden werden wir also beleuchten, was die tatsächlichen Vorteile beim Einsatz von Keyword-Driven Testing sind.

1.6.1 Klarheit

Das Formulieren von Testfällen aus vorgefertigten Bausteinen führt auf Dauer zu mehr Klarheit, und zwar bei der Interpretation und dem Verständnis dessen, was später bei der Ausführung der Testfälle zu tun ist.

Warum ist das so?

Jede Testerin muss beim Lesen einer Testspezifikation entscheiden, was mit einer bestimmten Formulierung gemeint ist (wir unterstellen, dass sie die Testspezifikation nicht selbst geschrieben hat). Wenn die Testspezifikation herkömmlich, also in Prosa verfasst ist, so sind die Formulierungen nicht einheitlich. Derselbe Sachverhalt kann mit verschiedenen Worten beschrieben sein. Ein Beispiel: Es wird einmal der Begriff »Anmelden« und das nächste Mal der Begriff »Einloggen« verwendet. Durchführende Testerinnen müssen an dieser Stelle entscheiden, ob diese Begriffe dasselbe bedeuten - wenn ja, was, und wenn nein: Was ist der Unterschied?

Keywords verhindern Ambivalenz.

Dieselbe Interpretationsleistung muss auch bei Verwendung von Keywords erbracht werden, aber eben nur einmal. Durch die Vereinheitlichung der Begriffe und das Bausteinprinzip gibt es nur ein Keyword für diese Aktivität. Am Anfang muss man einmal verstehen, was damit

Lerneffekt

gemeint ist – aber hier findet ein Lernprozess statt und bald kennt man die Bedeutung.

Dieses Beispiel für Vorteile des Keyword-Driven Testing greift insbesondere beim manuellen Test. Also Schluss mit dem Mythos, dass Keyword-Driven Testing nur ein Thema für Testautomatisierer ist!

Klar ist: Das hier beschriebene Mehr an Klarheit wird insbesondere dann seinen Nutzen entfalten, wenn in Teams gearbeitet wird. Aber eine verbesserte Lesbarkeit ist mit Sicherheit ein Gewinn.

1.6.2 Wiederverwendbarkeit

Mehr oder weniger selbsterklärend ist die Tatsache, dass Keywords wiederverwendet werden können. Das ist natürlich bei fast allen Testautomaten in Bezug auf die automatisierten Testschritte der Fall. Jedoch können wir diese Funktionalität bei kaum einem Werkzeug für manuelles Testen oder Testmanagement finden. Hier wird in aller Regel mit rein textuellen nicht wiederverwendbaren Testschritten gearbeitet.

Die Wiederverwendbarkeit steht in direktem Zusammenhang mit dem nächsten Abschnitt über Wartbarkeit. Aber es geht nicht nur darum, sondern auch um den ersparten Aufwand bei der Erstellung von Tests und um den Vorteil bezüglich der Klarheit und Lesbarkeit.

Aufwand sparen

Ein einmalig erstelltes Keyword, das beschreibt oder implementiert, wie unser Testobjekt zu starten ist, spart bei jeder Nutzung Zeit, verglichen mit einem immer wieder erneuten Beschreiben dieser Tätigkeit.

Wiederverwendbarkeit reduziert repetitive Arbeiten also auf ein Minimum und beschleunigt das Arbeiten entsprechend.

1.6.3 Wartbarkeit

Eine verbesserte Wartbarkeit ist wahrscheinlich der am häufigsten beschworene Vorteil von Keyword-Driven Testing.

Er ergibt sich auch bei manuellem Test, ist aber besonders im Zusammenhang mit Testautomatisierung wichtig, da der Nutzen von Testautomatisierung[3] von hohen Wartungsaufwänden zunichtegemacht werden kann.

Wartbarkeit ist ein Qualitätsmerkmal, das beschreibt, mit wie kleinem oder großem Aufwand Änderungen durchgeführt werden können. Ob Änderungen an unseren Testfällen notwendig sind, können wir uns

Auslöser für Wartung

[3] Der wirtschaftliche Nutzen wird dadurch festgestellt, dass die Kostenersparnis auf Dauer die Erstellungs- und Wartungskosten aufwiegt. Es gibt noch andere Nutzenaspekte, wie Motivation, Geschwindigkeit etc.