

Praxisbeispiel: datengetriebene und schlüsselwortgetriebene Testfalldarstellung

Dirk O. Schweier – G. Muth Partners GmbH, Wiesbaden – Januar 2013

Wenn Testautomatisierung zum Einsatz kommt, stellt sich die Frage, wie die Testfälle zu organisieren und abzulegen sind. Dieses Dokument stellt zwei Möglichkeiten vor:

1. Datengetriebene Testfalldarstellung
2. Schlüsselwortgetriebene Testfalldarstellung

Beide Konzepte werden an einem konkreten Beispiel vorgestellt und diskutiert, das im Rahmen eines Kundenprojekts vom Autor realisiert wurde.

Projektsituation

In einem Unternehmen wird ein Dispositionssystem eingesetzt. Das System besteht zum einen aus einer Branchensoftware und speziellen Kundenanpassungen durch den Softwarelieferanten. Es handelt sich dabei hinsichtlich der Architektur um eine Client-Server-Applikation.

Die Einführung der Software im Unternehmen wird von einem Projektteam begleitet. Eine Aufgabe des Teams ist auch der Test neuer Versionen des Dispositionssystems bevor die neue Version produktiv gestellt wird.

Aufgrund der regelmäßigen Lieferungen des Herstellers und der Komplexität des Systems sind wiederkehrende und aufwändige Regressionstests erforderlich. Um den manuellen Aufwand der Regressionstests verringern zu können, wurde die Entscheidung getroffen, bestimmte Regressionstests zu automatisieren.

Die Testautomatisierung soll angewendet werden für eine Client-Anwendung, über die die Disponenten Aufträge verwalten und Ressourcen disponieren. Bei dieser Anwendung, die im Folgenden als SUT (System under Test) bezeichnet wird, handelt es sich um eine Java Web Start Applikation. Zur Testautomatisierung wird Software QF-Test von [Quality First Software GmbH](#) verwendet.

Verwaltung von Tests in QF-Test

Um die folgenden Beispiele und Screenshots verstehen zu können, werden hier die wesentlichen Funktionen in QF-Test hier erläutert.

Der Testfall in QF-Test ist der wichtigste Knoten, da er die kleinste Einheit ist, die später in der Auswertung mit einem Status versehen wird. In diesem Knoten sind Funktionalität und Prüfungen zu implementieren, mit denen eine oder mehrere Testbedingungen des SUT geprüft werden können. Für einen Testfall-Knoten können Variablen definiert werden und diese auch

mit einem Wert vorbelegt werden, der verwendet wird, wenn zur Laufzeit des Testfalls kein Wert für die Variable ermittelt werden konnte. Somit kann man auch sagen, dass ein Testfall-Knoten einem abstrakten Testfall entspricht.

Der Testaufruf in QF-Test ist eine Referenz auf einen Testfall. Auf diese Weise müssen Testfälle, wenn sie in unterschiedlichen Testsequenzen verwendet werden, nicht kopiert werden. Vielmehr können sie einmal implementiert und über die Nutzung des Testaufruf-Knoten in unterschiedlichen Testsequenzen verwendet werden.

Der Testfallsatz in QF-Test ist ein allgemeines Strukturelement, das sowohl Knoten vom Typ Testfall, Testaufruf und Testfallsatz beinhalten kann. Ein Einsatzzweck dieses Knotens ist die Gliederung von verschiedenen Tests in Sequenzen, die zusammen genommen einen fachlichen Geschäftsvorfall abbilden. Sie lassen sich aber auch einsetzen, um eine Menge von Testfällen in einer Weise zu gliedern, die es dem Anwender erleichtert, bestimmte Testfall-Knoten später schnell wiederzufinden.

(Weitere Informationen über die Software QF-Test finden sich im [Handbuch](#), das von Quality First Software GmbH online bereitgestellt wird.)

Datengetriebene Testfalldarstellung

Das Tool zur Testautomatisierung bietet die Möglichkeit, Testsequenzen aufzunehmen und diese abspielen zu können. Diese Technik wird „Capture and Replay“ genannt. Die Nutzung dieser Funktionalität erscheint sinnvoll zu sein, wenn man mit der Testautomatisierung beginnt.

Dennoch ist das Mitschneiden einer Testsequenz mit einem Tool sehr nützlich. Über eine Aufzeichnung von Aktionen mit Hilfe des Testausführungswerkzeuges lässt sich sehr effizient ein Gerüst für ein Testskript erstellen. So übernimmt z.B. QF-Test bei der Aufzeichnung eines Testfalls auch gleich eine Erkennung der GUI-Elemente, die später bei der Programmierung von Skripten hilfreich ist. Die aufgezeichneten Sequenzen lassen sich nachbearbeiten, indem die fixen Werte durch Parameter ersetzt werden. Auf diese Weise hat man ein Skript, das mit entsprechenden Daten versorgt und mehrfach ausgeführt werden kann.

Ein konkreter Geschäftsvorfall, der im Rahmen des Regressionstest immer wieder zu testen ist, ist das Einplanen von Ressourcen für einen Auftrag und die anschließende Datenerfassung von Arbeitszeiten und verbrauchtem Material, um somit die Grundlage zur Verrechnung zu erhalten.

Dieser Geschäftsvorfall wurde in QF-Test als Testfallsatz „EB – Prozess“ implementiert (siehe Abbildung 1). In der Abbildung erkennt man, dass im Testfallsatz „EB – Prozess“ der Testfallsatz „Testdurchlauf \$(iTestRun_ebp)“ enthalten ist, der wiederum weitere Knoten vom Typ Testfallsatz, Testfall und Testaufruf beinhaltet. Somit sind nur die Knoten zu sehen, in denen Eingaben und Prüfungen implementiert sind. Die eigentlichen Aktionen, wie ein Mausklick oder die Eingabe von Daten, sind in der Abbildung nicht zu sehen, um die Übersichtlichkeit zu bewahren.

In dem Testfallsatz „EB – Prozess“ ist als erster Knoten ein Knoten vom Typ Datentreiber zu sehen. Dieser Datentreiber beinhaltet einen Knoten, der als Referenz auf eine Excel-Tabelle zu verstehen ist. Die Logik von QF-Test liest die Excel-Tabelle ein und führt dann den Test-

fallsatz „Testdurchlauf $\$(iTestRun_ebp)$ “ so oft aus, wie Datenzeilen in der Excel-Tabelle gefunden werden. Aus einem Datensatz müssen die einzelnen Eingabewerte und die Sollwerte hervorgehen, die für das Testskript benötigt werden.

Ein Beispiel für den Aufbau der Excel-Datei ist in Abbildung 2 zu sehen.

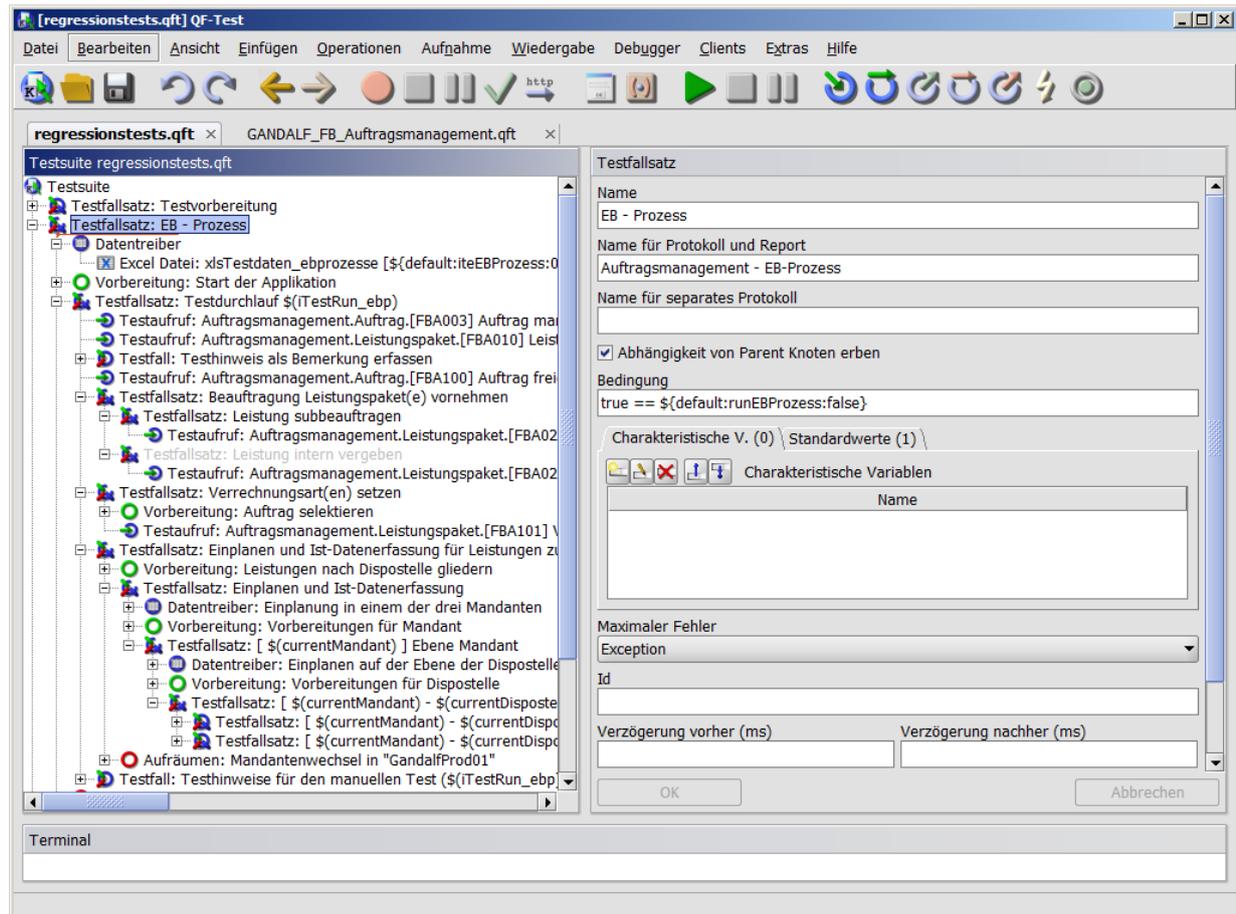


Abbildung 1 Testskript in der datengetriebenen Testfalldarstellung



Die datengetriebene Testfalldarstellung ist sehr gut geeignet, um Testskript und Testfälle arbeitsteilig zu entwickeln. In dem konkreten Projekt werden die Skripte von Testautomatisierern entwickelt, die QF-Test kennen und Kenntnisse in der Softwareentwicklung haben. Die konkreten Testfälle werden von Testdesignern mit den Testentwurfstechniken wie z.B.

- Äquivalenzklassenanalyse,
- Grenzwertanalyse und
- Entscheidungstabellen

entwickelt und sind dann in Excel zu hinterlegen.



Das einmal entwickelte Skript in Form des Testfallsatz ist hinsichtlich der Ausführung sehr starr und damit auch unflexibel. Wenn aus unterschiedlichen Gründen eine Änderung in der Ausführungsreihenfolge erforderlich wird, dann ist der Testautomatisierer

gefragt, der im QF-Test das Testskript ändern muss.

Angesichts der Tatsache, dass eine Änderung des Testfallsatz von einem Testdesigner identifiziert wird und ein Testautomatisierer die Anforderung umsetzen muss, ist dieses Konzept unflexibel.

	G	H	I	J	K	L	M	N	O		
1	Auftrag.Typ	Auftrag.Titel	Auftrag.St art	Auftrag.En de	Auftrag.Sta rt	Auftrag.Sta rt.relative	Auftrag.Da uer	Auftrag.Auf tragsnumm er	Auftrag.Ver rechnungsn umber	Leistungspaket.Bezeichnung	Leistun g
2	EB	QFT: Prozess 2 - Testfall 1				-2:18:00	3:00		46170655	EB 2 Mann Team aktuell - Halber Tag	<body>eistung aktuell Tag</b chnung e></vei y>
3	EB	QFT: Prozess 2 - Testfall 2				-2:14:00	4:00		46170655	EB 2 Mann Team aktuell - Halber Tag	<body>eistung aktuell Tag</b chnung e></vei y>
4	EB	QFT: Prozess 2 - Testfall 3				-2:9:00	5:00		46170655	EB 2 Mann Team aktuell - Halber Tag	<body>eistung aktuell Tag</b chnung e></vei y>
5	EB	QFT: Prozess 2 - Testfall 4				-2:18:00	9:15		46170655	EB 2 Mann Team aktuell - Halber Tag	<body>eistung aktuell Tag</b chnung e></vei y>
6	EB	QFT: Prozess 2 - Testfall 5				-2:18:00	3:00		46170655	EB 2 Mann Team aktuell - Halber Tag	<body>eistung aktuell Tag</b chnung e></vei y>

Abbildung 2 Testdaten in der datengetriebenen Testfalldarstellung

Ein weiterer Nachteil kann die Wartbarkeit der Testskripte sein. Die Testsequenzen, die über das Testausführungswerkzeug aufgenommen werden, können schnell unübersichtlich werden. Wie bereits oben ausgeführt, sind in der Abbildung 1 lediglich die Strukturelemente Testfallsatz, Testfall und Testfallaufruf zu sehen. Unterhalb der Knoten Testfall, Testaufruf sind weitere Knoten zu finden, die teilweise wieder eine Anzahl von Knoten beinhalten.

Mangelnde Wartbarkeit der Testfälle kann bei sich ändernden Anforderungen schnell dazu führen, dass der Änderungsaufwand der Skripte groß wird. Der Wartungsaufwand und die Tatsache, dass nur Teile der bereits erstellten Skripte wiederverwendet werden können, wirken sich nachteilig auf die Effizienz der Testautomatisierung aus.

Bei steigender Anzahl an Daten, die für das Testskript benötigt werden, geht auch die Übersichtlichkeit der Datentabelle verloren. In dem konkreten Beispiel hat die Tabelle 30 Spalten.

In dem Projekt wurden aber auch Geschäftsvorfälle für die Testautomatisierung geplant, bei denen die Anzahl von 30 Spalten überschritten wurde.

Schlüsselwortgetriebene Testfalldarstellung

Ein Geschäftsprozess, der durch einen Testfall repräsentiert wird, lässt sich aus fachlicher Sicht in kleinere Schritte zerlegen, die im Folgenden als Schlüsselwörter bezeichnet werden. Ein Schlüsselwort kann z.B. der Start einer Applikation oder die Anmeldung an einem System sein.

Soll das Konzept der schlüsselwortgetriebenen Testfalldarstellung angewendet werden, dann ist im ersten Schritt zu analysieren, welche Schlüsselwörter für das SUT zu definieren sind. Bei dieser Analyse spielen fachliche und technische Aspekte eine Rolle.

Aus fachlicher Sicht ist zu überlegen, welche Aufgaben mit dem SUT im Rahmen des Tests auszuführen sind. Die Schlüsselwörter sind aus fachlicher Sicht möglichst so zu formulieren, dass die Schlüsselwörter in mehreren Testfällen genutzt werden können.

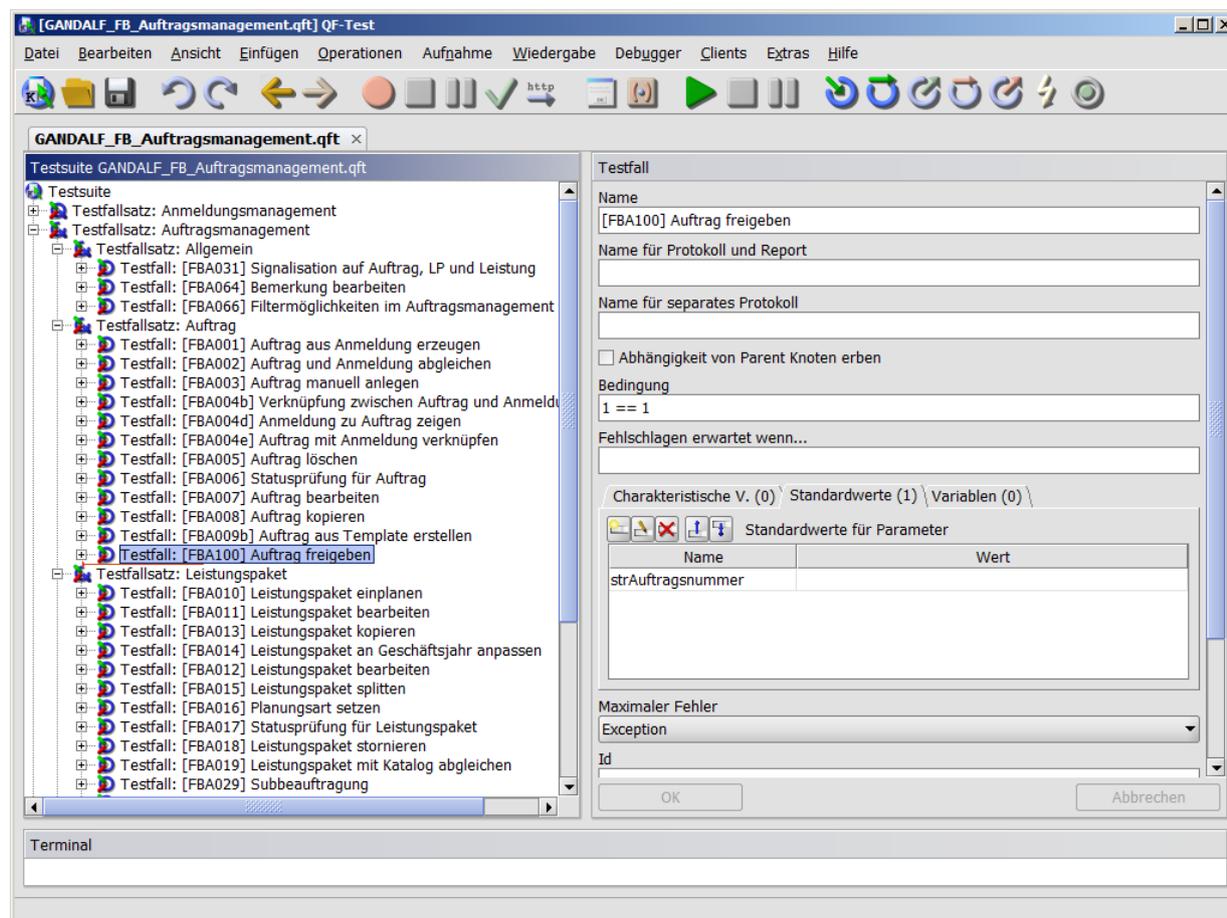
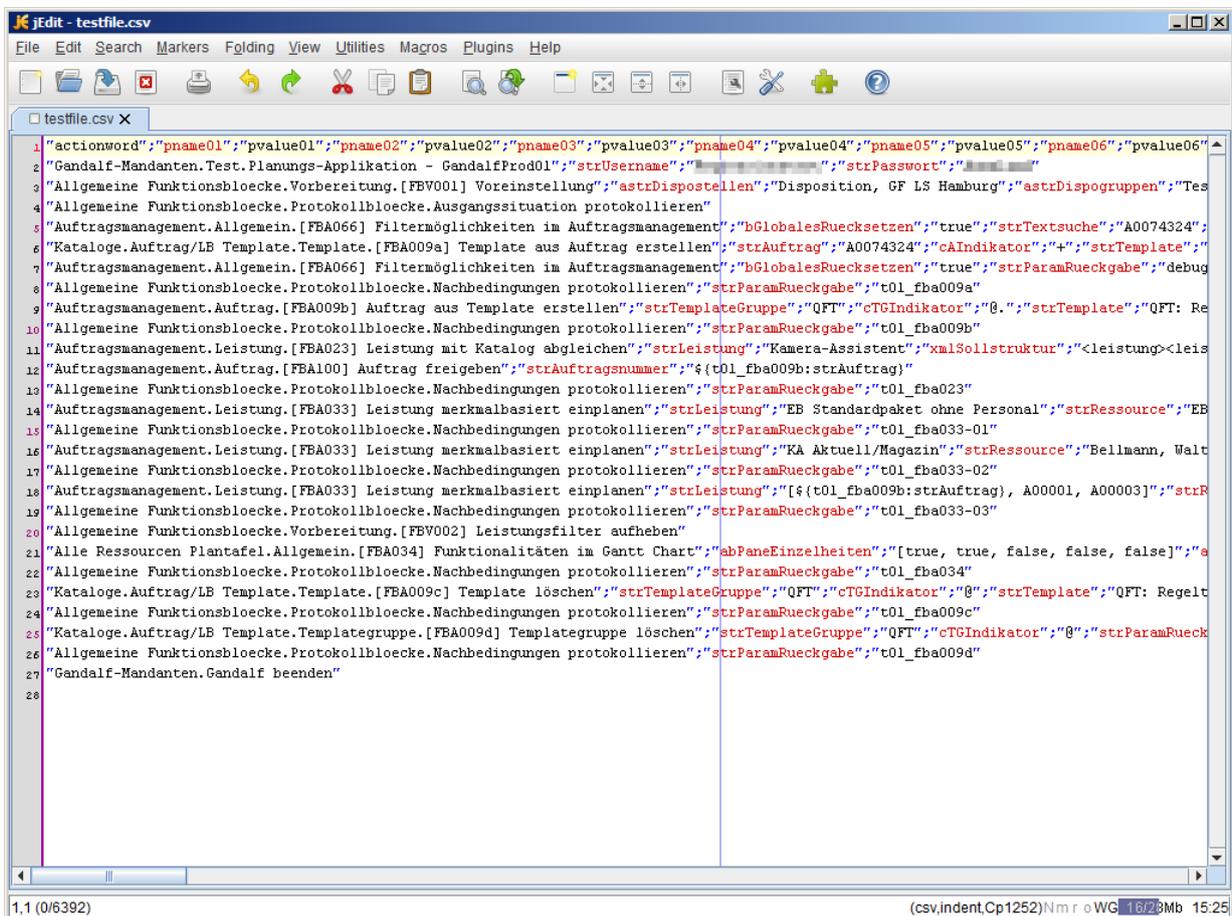


Abbildung 3 Bibliothek mit den bereitgestellten Schlüsselwörtern

Die Schlüsselwörter sollen aus technischer Sicht möglichst wenige Einzelaktionen – wie Eingaben oder Mausklicks – haben, um später ein sehr flexibles Testfalldesign zu ermöglichen. Ebenfalls ist aus technischer Sicht zu planen, welche Eingabeparameter dem Schlüsselwort mitzugeben sind und welche Ergebnisse im Rahmen der Verarbeitung als Rückgabewerte für nachfolgende Schlüsselwörter benötigt werden.

Die Schlüsselwörter lassen sich in QF-Test in Form einer Bibliothek anlegen. Die Abbildung 3 zeigt eine solche Bibliothek. Die Schlüsselwörter sind Testfall-Knoten; die Testfallsätze wurde verwendet, um die einzelnen Schlüsselwörter fachlich zu sortieren. Dadurch soll die Übersichtlichkeit der Bibliotheken auch bei großer Anzahl von Schlüsselwörtern gewährleistet bleiben.



```

1 "actionword";"pname01";"pvalue01";"pname02";"pvalue02";"pname03";"pvalue03";"pname04";"pvalue04";"pname05";"pvalue05";"pname06";"pvalue06"
2 "Gandalf-Mandanten.Test. Planungs-Applikation - GandalfProd01";"strUsername";"
3 "Allgemeine Funktionsblöcke.Vorbereitung. [FBV001] Voreinstellung";"astrDispostellen";"Disposition, GF LS Hamburg";"astrDispogruppen";"Tes
4 "Allgemeine Funktionsblöcke. Protokollblöcke. Ausgangssituation protokollieren"
5 "Auftragsmanagement. Allgemein. [FBA066] Filtermöglichkeiten im Auftragsmanagement";"bGlobalesRuecksetzen";"true";"strTextsuche";"A0074324";
6 "Kataloge. Auftrag/LB Template. Template. [FBA009a] Template aus Auftrag erstellen";"strAuftrag";"A0074324";"cAIndikator";"@"; "strTemplate";"
7 "Auftragsmanagement. Allgemein. [FBA066] Filtermöglichkeiten im Auftragsmanagement";"bGlobalesRuecksetzen";"true";"strParamRueckgabe";"debug
8 "Allgemeine Funktionsblöcke. Protokollblöcke. Nachbedingungen protokollieren";"strParamRueckgabe";"t01_fba009a"
9 "Auftragsmanagement. Auftrag. [FBA009b] Auftrag aus Template erstellen";"strTemplateGruppe";"QFT";"cTGIndikator";"@"; "strTemplate";"QFT: Re
10 "Allgemeine Funktionsblöcke. Protokollblöcke. Nachbedingungen protokollieren";"strParamRueckgabe";"t01_fba009b"
11 "Auftragsmanagement. Leistung. [FBA023] Leistung mit Katalog abgleichen";"strLeistung";"Kamera-Assistent";"xmlSollstruktur";"<leistung><leis
12 "Auftragsmanagement. Auftrag. [FBA100] Auftrag freigeben";"strAuftragsnummer";"${t01_fba009b: strAuftrag}"
13 "Allgemeine Funktionsblöcke. Protokollblöcke. Nachbedingungen protokollieren";"strParamRueckgabe";"t01_fba023"
14 "Auftragsmanagement. Leistung. [FBA033] Leistung merkmalsbasiert einplanen";"strLeistung";"EB Standardpaket ohne Personal";"strRessource";"EB
15 "Allgemeine Funktionsblöcke. Protokollblöcke. Nachbedingungen protokollieren";"strParamRueckgabe";"t01_fba033-01"
16 "Auftragsmanagement. Leistung. [FBA033] Leistung merkmalsbasiert einplanen";"strLeistung";"KA Aktuell/Magazin";"strRessource";"Bellmann, Walt
17 "Allgemeine Funktionsblöcke. Protokollblöcke. Nachbedingungen protokollieren";"strParamRueckgabe";"t01_fba033-02"
18 "Auftragsmanagement. Leistung. [FBA033] Leistung merkmalsbasiert einplanen";"strLeistung";"[${t01_fba009b: strAuftrag}, A00001, A00003]";"strR
19 "Allgemeine Funktionsblöcke. Protokollblöcke. Nachbedingungen protokollieren";"strParamRueckgabe";"t01_fba033-03"
20 "Allgemeine Funktionsblöcke.Vorbereitung. [FBV002] Leistungsfilter aufheben"
21 "Alle Ressourcen Plantafel. Allgemein. [FBA034] Funktionalitäten im Gantt Chart";"abPaneEinzelheiten";"[true, true, false, false, false]";"a
22 "Allgemeine Funktionsblöcke. Protokollblöcke. Nachbedingungen protokollieren";"strParamRueckgabe";"t01_fba034"
23 "Kataloge. Auftrag/LB Template. Template. [FBA009c] Template löschen";"strTemplateGruppe";"QFT";"cTGIndikator";"@"; "strTemplate";"QFT: Regelt
24 "Allgemeine Funktionsblöcke. Protokollblöcke. Nachbedingungen protokollieren";"strParamRueckgabe";"t01_fba009c"
25 "Kataloge. Auftrag/LB Template. Templategruppe. [FBA009d] Templategruppe löschen";"strTemplateGruppe";"QFT";"cTGIndikator";"@"; "strParamRueck
26 "Allgemeine Funktionsblöcke. Protokollblöcke. Nachbedingungen protokollieren";"strParamRueckgabe";"t01_fba009d"
27 "Gandalf-Mandanten. Gandalf beenden"
28

```

Abbildung 4 Testfall in der schlüsselwortgetriebenen Testfalldarstellung

Im konkreten Projekt werden in dem Dispositionssystem Produktionsaufträge verwaltet. Die Disponenten legen fest, welche Leistungen benötigt werden, und weisen dann den Leistungen konkretes Personal und Material zu.

Bezogen auf den Test dieser Zuweisung wird ein Auftrag benötigt, der den Status „freigegeben“ hat.

In der Bibliothek ist zu erkennen, dass es das Schlüsselwort „[FBA100] Auftrag freigeben“ gibt. Dieses Schlüsselwort legt aber keinen Auftrag an, sondern gibt nur einen Auftrag frei, der vorher angelegt worden sein muss.

Man hat sich bewusst dagegen entschlossen, Anlegen und Freigeben in einem Schlüsselwort zu implementieren, da Aufträge auf unterschiedliche Weise im System angelegt werden können. Folgende Schlüsselwörter führen dazu, dass ein neuer Auftrag im System angelegt wird:

- [FBA001] Auftrag aus Anmeldung erzeugen
- [FBA003] Auftrag manuell anlegen
- [FBA008] Auftrag kopieren
- [FBA009b] Auftrag aus Template erstellen

Der Testdesigner ist mit mehreren Schlüsselwörtern also flexibler in der Gestaltung seiner Testfälle und kann z.B. Schlüsselwörter zwischen dem Anlegen und Freigeben eines Auftrages einfügen. Dieses wäre nicht möglich gewesen, wenn der Testautomatisierer genau ein Schlüsselwort bereitgestellt hätte.

Ein Testfall kann jetzt vom Testdesigner implementiert werden, indem er die zur Verfügung gestellten Schlüsselwörter in die Reihenfolge bringt, in der er sie ausgeführt haben möchte, um einen bestimmten Testfall zu implementieren. Das Ergebnis ist eine Liste aus Schlüsselwörtern, die z.B. als .csv-Datei abgelegt werden kann.

In Abbildung 4 ist ein Beispiel für eine solche Datei zu sehen. Die erste Zeile enthält die Spaltenüberschriften. Danach ist in jeder Zeile zuerst das Schlüsselwort zu übergeben. Dahinter folgen dann jeweils Paare aus Parametername („pname01“) und Parameterwert („pvalue01“). Die konkrete Anzahl der Paare ist abhängig vom jeweiligen Schlüsselwort.

In Zeile 12 wird das Schlüsselwort „Auftragsmanagement.Auftrag.[FBA100] Auftrag freigeben“ mit dem Parameter „strAuftragsnummer“ und dem Parameterwert „\${t01_fba009b:strAuftrag}“ aufgerufen. Die Notwendigkeit der Parameterübergabe ergibt sich aus der Tatsache, dass dem Schlüsselwort eine Auftragsnummer zu übergeben ist, damit aus der Menge der vorhandenen Aufträge der selektiert werden kann, der freigegeben werden soll.

Man könnte auf die Idee kommen, dass die Übergabe der Parameternamen nicht erforderlich und nur die Parameterwerte übergeben werden und die Reihenfolge sich aus dem Schlüsselwort ergibt. Der Grund für die Verwendung von Paaren ist in der Architektur von QF-Test zu finden und soll hier nicht näher ausgeführt werden, da dieses zu weit führt.

Ein anderer Aspekt, der an dieser Zeile deutlich wird, ist die Verwendung von Rückgabewerten. Die Auftragsnummer wird vom Dispositionssystem vergeben, wenn ein neuer Auftrag angelegt wird. Somit kann zum Zeitpunkt der Erstellung des Testfalls nicht gesagt werden, welche Auftragsnummer der Auftrag bekommt, der über das Schlüsselwort in Zeile 9 angelegt wird. Der Wert „\${t01_fba009b:strAuftrag}“ steht für die Referenz auf eine Variable. Somit wird erst zur Laufzeit dieser Wert in eine konkrete Auftragsnummer umgewandelt.

Nachdem jetzt die Bibliothek und Testfälle vorliegen, fehlt noch das geeignete Testskript, mit dem die entsprechenden Schlüsselwörter aufgerufen werden und variable Angaben in konkrete Werte umgesetzt werden. In Abbildung 5 ist der Testfallsatz dargestellt, der diese Aufgabe übernimmt.

In dem Testfallsatz „Keyword-Driven-Testing“ sind zwei weitere Testfallsatz-Knoten und ein Testaufruf-Knoten ineinander geschachtelt.

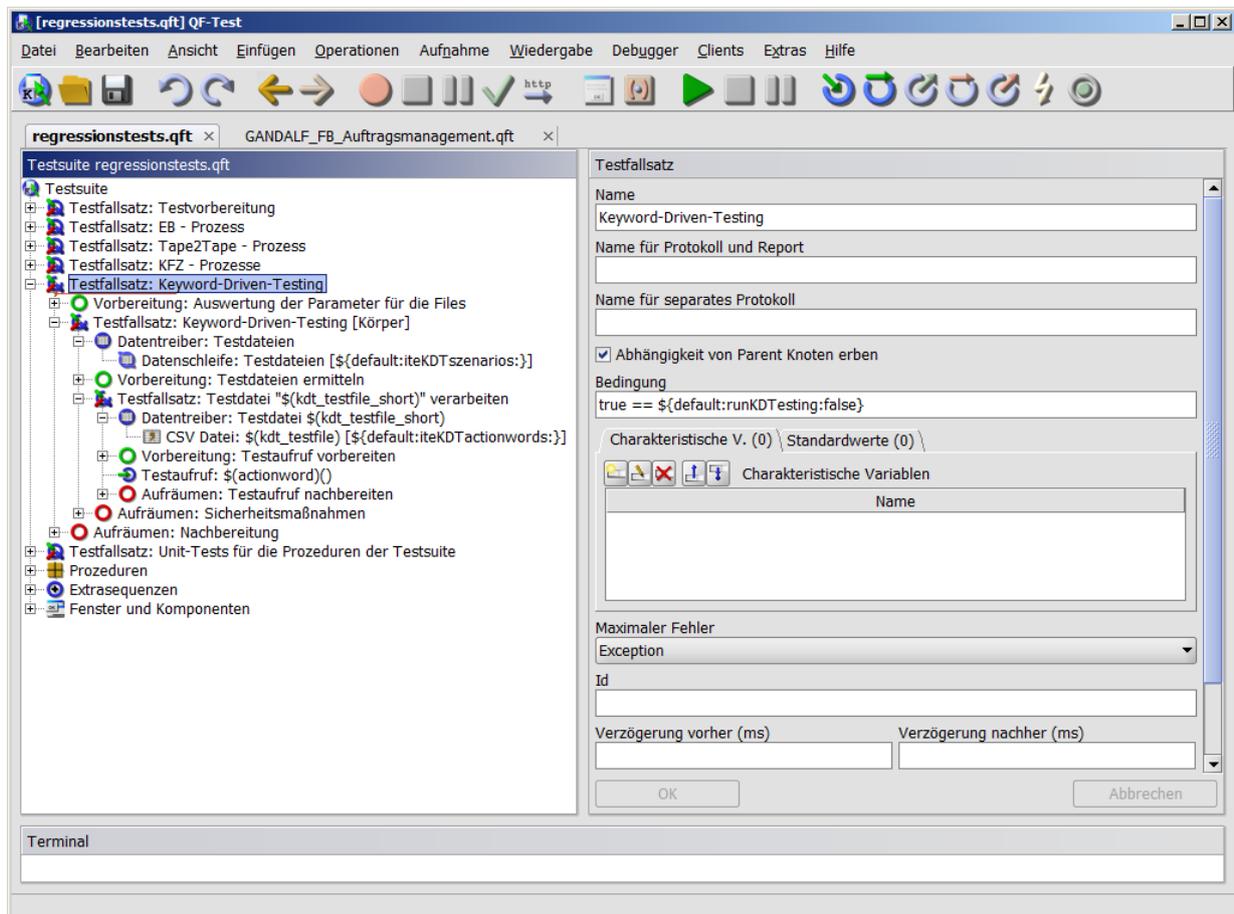


Abbildung 5 Testskript zur Verarbeitung der schlüsselwortgetriebenen Testfälle

Das „innerste Element“ ist der Testaufruf-Knoten „\$(actionword)()“ und repräsentiert den Aufruf eines Testfall-Knoten, der ein Schlüsselwort repräsentiert. Der konkrete Wert, der Variable „actionword“ ergibt sich über den Datentreiber „Testdatei \$(kdt_testfile_short)“. In den .csv-Dateien, die den Testfällen entsprechen trägt die erste Spalte die Überschrift „actionword“ (siehe Zeile 1 in Abbildung 4). Die Versorgung des Schlüsselwortes mit den Parameterwerten, die in der Datei übergeben werden, findet in diesem Beispiel in dem Knoten „Testaufruf vorbereiten“ statt.

Für den Test des SUT wird der Testdesigner mehrere Dateien mit Listen von Schlüsselwörtern anlegen und ausführen wollen. Aus diesem Grund gibt es die Testfall-Knoten „Testdatei “\$(kdt_testfile_short)“ verarbeiten“ und „Keyword-Driven-Testing [Körper]“.

Der Testfallknoten „Keyword-Driven-Testing [Körper]“ beinhaltet den Datentreiber „Testdateien“. Dieser Datentreiber greift nicht auf eine Tabelle zu, sondern ist eine Schleife, bei der ein Index hochgezählt wird. Das Maximum für den Index wurde zuvor in dem Knoten „Auswertung der Parameter für die Files“ berechnet.

Für jede Datei, die sich aus den Aufrufparametern ergibt, wird der Testfallsatz „Testdatei “\$(kdt_testfile_short)” verarbeiten“ ausgeführt.



Ein wesentlicher Vorteil der schlüsselwortgetriebenen Testfalldarstellung ist der, dass Testfalldesigner und Testautomatisierer arbeitsteilig arbeiten können und nicht immer aufeinander angewiesen sind.

Der Testdesigner kann mit der fachlichen Kenntnis der Applikation die Testfälle leicht erstellen, pflegen und modifizieren.

Der Testautomatisierer kann sich darauf beschränken, die Schlüsselwörter gemäß den Anforderungen des Testdesigners zu implementieren. Er benötigt dazu in erster Linie die Fachkenntnisse zur Bedienung der Software und auf dem Gebiet der Softwareentwicklung.

Ein weiterer Vorteil ist in der Modularisierung zu sehen. Die Definition von Schlüsselwörtern führt dazu, dass die Skripte, die als Testfälle entwickelt und bereitgestellt werden, einen hohen Grad an Wiederverwendung haben, wodurch auch einmalige Routineaufgaben z.B. im Rahmen einer Datenmigration mit den Mitteln der Testautomatisierung effizient ausgeführt werden können.



Die Definition der Schlüsselwörter, die wesentlicher Bestandteil dieses Konzepts ist, erfordert neben guten Fachkenntnissen über das Testsystem auch sehr gute Kenntnisse der Softwareentwicklung und teilweise auch der Softwarearchitektur, da zu wenige Schlüsselwörter die Flexibilität des Testdesigner einschränken. Zu viele Schlüsselwörter führen dazu, dass die Listen zu lang werden und damit unübersichtlich werden können.

Das Testframework ist ebenfalls etwas, das selbst entwickelt werden muss, da in den seltensten Fällen ein Werkzeughersteller etwas Fertiges anbietet. Bei dem Testframework sind neben der Syntax auch Überlegungen anzustellen, wie mit unerwarteten Ereignissen während der Testausführung umzugehen ist. Wenn ein Schlüsselwort nicht abgearbeitet werden kann, dann können unter Umständen andere Schlüsselwörter ebenfalls nicht bearbeitet werden, weil es zum einen aus fachlicher Sicht und zum andern aus technischer Sicht nicht möglich ist. Hier sind dann Überlegungen anzustellen, wie mit solchen Ausnahmesituationen umgegangen werden soll. Auch diese Lösungen können vom Testobjekt abhängig sein und sind im Einzelfall zu entscheiden.

Welches Konzept sollte in der Praxis Anwendung finden?

Nachdem die beiden Konzepte der datengetriebenen und schlüsselwortgetriebenen Testfalldarstellung in den vorangegangenen Kapiteln erläutert und jedes Konzept für sich bewertet wurde, stellt sich die Frage, welches Konzept in einem Projekt verwendet werden sollte?

In dem konkreten Fall des Dispositionssystems sollte im Test die Verrechnung von Leistungen geprüft werden. Dazu wurden insgesamt 94 Testfälle definiert, in denen Aufträge angelegt, Leistungen disponiert und Zeiten erfasst werden mussten.

Das Konzept der datengetriebene Testfalldarstellung erlaubt es, die 94 Testfälle in einer Tabelle zu erfassen und zu pflegen.

Unter Nutzung des Konzepts der schlüsselwortgetriebenen Testfalldarstellung sind 94 Dateien zu erstellen. Die Inhalte der Dateien unterscheiden sich lediglich in den konkreten Parame-

terwerten, während die Anzahl und Reihenfolge der verwendeten Schlüsselworte in den Dateien identisch ist.

Wenn sich in dem Ablauf der Testausführung etwas ändert, dann ist im einen Fall das Testskript zu ändern und im anderen Fall sind 94 Dateien anzupassen. Ungeachtet vom benötigten Aufwand kann man wohl davon ausgehen, dass eine Änderung in den Dateien in der Regel fehleranfälliger sein dürfte als eine zentrale Änderung im Testskript.

Folglich ist vorstellbar, dass in dem konkreten Fall die datengetriebene Testfalldarstellung besser geeignet ist.

In einem anderen Fall soll im Rahmen der Regressionstest geprüft werden, ob die Stornierung von Aufträgen korrekt funktioniert. Dazu werden Aufträge benötigt, die unterschiedliche Status haben und auch der Zeitpunkt der Leistungserbringung kann eine Rolle spielen. In diesem Fall ist bezogen auf vorstellbare Testfälle nur die Aktion des Stornierens ein gemeinsame. Die Vorarbeiten zum Anlegen des Auftrags und Schaffen der Ausgangsbedingungen unterscheiden sich. Diese Tests lassen sich nur schwer in der datengetriebenen Testfalldarstellung abbilden.

Der Vorteil der Flexibilität, der im Konzept der schlüsselwortgetriebene Testfalldarstellung beschrieben wurde, überwiegt jetzt deutlich, so dass dieses Konzept zu bevorzugen ist.

Neben dem Anwendungszweck können auch organisatorische Gründe ausschlaggebend für die Auswahl des Konzepts sein. Hierbei ist zu prüfen, ob die Rollen Testdesigner und Testautomatisierer im Unternehmen durch unterschiedliche Personen wahrgenommen werden. Wenn dieses der Fall ist und unterstellt werden kann, dass der Testautomatisierer mit den fachlichen Anforderungen nicht so vertraut ist, dann erscheint ebenfalls das Konzept der schlüsselwortgetriebenen Testfalldarstellung besser geeignet zu sein.

Ist jedoch der Testdesigner mit der Erstellung der Tests in der geforderten formalen Darstellungsform überfordert, dann ist zu überlegen, ob man nicht mit der datengetriebenen Testfalldarstellung eine höhere Akzeptanz erreichen kann.

Die Akzeptanz für das gewünschte Konzept ist ein Erfolgsfaktor, der zwar selten in Zahlen ausgedrückt werden kann, sollte aber nicht unterschätzt werden, da mangelnde Akzeptanz bezüglich der Testautomatisierung dazu führen kann, dass diese nur in geringen Umfang eingesetzt wird und somit der deutliche höhere Aufwand zu Beginn des Projekts sich über die Zeit sich im Ergebnis nicht rechnet.

Aus dem Blickwinkel der Softwareentwicklung überwiegen die Vorteile der Modularisierung, Wartbarkeit und Flexibilität so deutlich, dass man sich für die Definition und Implementierung von Schlüsselwörtern entscheiden wird. Die Schlüsselwörter erlauben dann auf jeden Fall die schlüsselwortgetriebene Testfalldarstellung und die Testskripte für eine datengetriebene Testfalldarstellung lassen sich unter Verwendung der Schlüsselwörter ebenfalls effizient erstellen.

Wenn in der Testautomatisierung ausreichend Know-How bezüglich Softwareentwicklung und Softwarearchitektur vorhanden ist, wird man fast automatisch modularisieren und auf diese Weise Schlüsselwörter implementieren.

Die Wahl der Testfalldarstellung hängt dann eher von der konkreten Aufgabenstellung aus fachlicher Sicht und den Fähigkeiten des Testdesigners ab.

Natürlich ist auch vorstellbar, dass der Inhalt einer .csv-Datei in einem Werkzeug für das Testfallmanagement gehalten wird und das Werkzeug zur Testautomatisierung die Testfälle aus dem Testfallmanagement entnimmt und die Ergebnisse an dieses zurückmeldet.

Die Firma Quality First Software GmbH liefert mit QF-Test ein Beispiel aus, in dem eine Schnittstelle zu dem Werkzeug Testlink, das unter den Bestimmungen der GPL bereitgestellt wird, implementiert ist.

Fazit

Welches Konzept der Testfalldarstellung zu nutzen ist, lässt sich nicht pauschal beantworten. Wie im vorangegangenen Kapitel gezeigt, lassen sich Beispiele finden, die klar für das eine oder andere Konzept sprechen. Von daher kommt es darauf an, dass der Testmanager bei der Planung der Testautomatisierung die fachlichen und organisatorischen Aspekte sorgfältig gegeneinander abwägt.

Im konkreten Projekt wurde anfangs die datengetriebene Testfalldarstellung bevorzugt, da sich das Projektteam rein auf die Fachlichkeit im Testdesign fokussieren wollte. Nachdem erste Erfahrungen mit der Testautomatisierung gesammelt waren, kam die Anfrage, ob es möglich wäre, in bestimmten Bereichen des SUT die schlüsselwortgetriebene Testfalldarstellung nutzen zu können. Aufgrund der Tatsache, dass die bereits erstellten Teile stark modularisiert waren, konnten die Schlüsselwörter und das Testskript zur Verarbeitung der Listen innerhalb von wenigen Personentagen bereitgestellt werden.

In den Regressionstests für das Dispositionssystem finden beide Konzepte Anwendung. Die neu entwickelten Testfälle, die den Umfang der Regressionstests erweitern sollen, werden überwiegend in der schlüsselwortgetriebenen Testfallerstellung formuliert.

Die Unterstützung des Projektteams durch die Testautomatisierer bezog sich anfangs auf die Entwicklung der Testskripte und Templates für die Datentabellen. Nach dem Konzeptwechsel lag der Fokus auf der Analyse von Logfiles, die von Testfällen stammten, die durch Fehler in den Testfall-Knoten oder in den Daten fehlgeschlagen waren.