

FitNesse

Kategorie: Framework für automatisierte Akzeptanztests auf Basis eines Wiki
Autor: Dirk O. Schweier

Tool:	FitNesse
Zweck:	Entwicklung und Durchführung von Akzeptanztests in Form von Wiki-Seiten. Die Testfalldarstellung setzt nicht zwingend Programmierkenntnisse voraus und ist damit auch für Tester aus Fachabteilungen von Interesse.
Hersteller:	Robert C. Martin, Micah D. Martin, Patrick Wilson-Welsh & FitNesse contributors
Webseite:	http://www.FitNesse.org
Kategorie:	Testausführungswerkzeug
Lizenz:	Open Source
Eigenschaften:	Test in Form von Wiki-Seiten, die hierarchisch gegliedert werden können. FitNesseApplikation kann über Web-Server betrieben werden und bringt auch eine Versionskontrolle mit.
Version:	Release v20140630

Inhaltsverzeichnis

1. Einleitung.....	3
1.1. Wer sollte das Dokument lesen?	3
1.2. Begrifflichkeiten in dem Dokument	3
2. Management Summary.....	3
3. Installation.....	5
4. Funktionsbeschreibung.....	5
4.1. Wiki-Seiten	6
4.1.1. Wiki-Seiten von Typ „Static“	7
4.1.2. Wiki-Seiten vom Typ „Suite“	8
4.1.3. Wiki-Seiten vom Typ „Test“	10
4.2. Testausführungsschicht in FitNesse	11
4.3. Fixtures.....	12
5. Quellen und Weiterführendes	13

Abbildungsverzeichnis

Abbildung 1: schematischer Aufbau von FitNesse.....	6
Abbildung 2: Beispiel für eine Testseite in FitNesse	7
Abbildung 3: Beispiel einer Wiki-Seite vom Typ "Suite"	8
Abbildung 4: Bearbeitungsansicht einer Wiki-Seite von Typ "Suite"	9
Abbildung 5: Beispiel einer Wiki-Seite nach Ausführung der Testfälle	10

1. Einleitung

1.1. Wer sollte das Dokument lesen?

Dieses Dokument soll einen Einblick in den Aufbau und die Funktionen von FitNesse, einem Akzeptanztestframework auf Basis von Wiki-Seiten, geben. Angesichts der Tatsache, dass auf den Webseiten von FitNesse und innerhalb der Standard-Installation sehr viele Informationen zur Benutzung zu finden sind, soll dieses Dokument nicht die deutschsprachige Hilfe von FitNesse sein.

Das Dokument richtet sich eher an Testmanager, Tester und Entwickler von Softwaresystemen, die sich mit dem Thema Testautomatisierung beschäftigen. Diesem Personenkreis soll das Dokument aufzeigen, in welchem Umfeld und mit welchem Nutzen sich das Testausführungswerkzeug FitNesse in ein Testprojekt integrieren lässt.

1.2. Begrifflichkeiten in dem Dokument

Die in diesem Dokument verwendeten Fachbegriffe orientieren sich an dem Glossar des ISTQB.

2. Management Summary

FitNesse ist ein Testframework, in dem Testfälle in Form von Wiki-Seiten abgelegt und automatisiert ausgeführt werden. Aus Sicht des Testdesigners wird die Software vollständig im Browser bedient. Der Start der Testdurchführung kann sowohl manuell im Browser als auch mittels Aufruf einer anderen Software erfolgen.

Aufgrund der relativ einfachen Sprache, in der Testfälle geschrieben werden, und der Nutzung des Browsers an Stelle einer Entwicklungsumgebung ist FitNesse auch von Testern gut nutzbar, die eher keine Programmierkenntnisse haben. Somit eignet sich FitNesse für automatisierte Regressionstests auf allen Teststufen.

Für einen produktiven Einsatz im Unternehmen wird empfohlen, dass FitNesse auf einem Server installiert wird.

Das Erstellen von Testfällen funktioniert wie das Schreiben einer Wiki-Seite. Damit hat der Editor nicht das typische Erscheinungsbild einer Entwicklungsumgebung und FitNesse ist somit auch für Tester geeignet, die keine oder wenige Kenntnisse aus dem Bereich der Softwareentwicklung haben.

Dabei soll aber nicht verschwiegen werden, dass neben den Wiki-Seiten mit den sogenannten Fixtures auch eine Komponente benötigt wird, die es für das jeweilige Testsystem zu entwickeln gilt. Die Fixtures sind Treiber, mit denen das SUT (System under test) stimuliert wird.

Der modulare Aufbau von FitNesse mag im ersten Augenblick kompliziert erscheinen, trägt aber im Alltag dazu bei, dass Tests arbeitsteilig und entsprechend der Qualifikation der Tester erstellt werden können. Testautomatisierungsexperten oder auch Softwareentwickler können

die Fixtures in der gewohnten Entwicklungsumgebung in der entsprechenden Programmiersprache erstellen und erzeugen damit Schlüsselwörter, die in den Wiki-Seiten genutzt werden können.

Tester mit guter Kenntnis der fachlichen Domäne können Wiki-Seiten erstellen, in den die Testfälle in Kapitel und mit Anmerkungen gegliedert werden können. Der Teil, der dann bei der Testdurchführung zur Ausführung gebracht wird, wird in Tabellenform in der Wiki-Seite erstellt und erlaubt sowohl eine datengetriebene als auch eine schlüsselwortgetriebene Testfalldarstellung.

Alle Testdurchführungen und Änderungen an Testfällen werden von FitNesse selbst versioniert und somit ist auch die erforderliche Nachvollziehbarkeit der Testarbeit gegeben.

FitNesse ist ein Testdurchführungswerkzeug, das keinerlei Installation auf den Rechnern der Testdesigner braucht, da es webbasiert ist und über die gängigen Browser bedient werden kann.

Im Gegensatz zu anderen Testdurchführungswerkzeugen fehlt FitNesse die Capture-and-Replay-Funktion. Dieses wird nicht als Nachteil angesehen, da die aufgenommen Skripte fast immer nachbearbeitet werden müssen, wozu auch fast immer Programmierkenntnisse erforderlich sind.

Da FitNesse von Softwareentwicklern spezifiziert und entwickelt wurde, kann man es sehr gut mit Unit-Test-Tools vergleichen. Während Unit-Test-Tools, wie z.B. junit, die Testfälle in Quellcode darstellen und in der Regel auf eine Programmiersprache begrenzt sind, ist die Darstellung als Wiki-Code unabhängig von einer speziellen Programmiersprache. Damit kann man bei Testern aus dem Fachbereich eher eine Akzeptanz erreichen.

Aufgrund der gut lesbaren Form mit Tabellen und erklärenden Texten kann FitNesse auch ein einfaches Testfallmanagement-Werkzeug für Testabteilungen sein, die Testfälle überwiegend in nicht spezialisierten und nicht zentralisierten Werkzeugen halten.

Für Unternehmen, die aktuell nur wenig oder keine Testautomatisierung einsetzen, kann FitNesse eine Lösung sein, um den Einstieg in das Thema Testautomatisierung auf allen Teststufen vom Komponententest bis zum Abnahmetest zu finden, da die Aufgaben entsprechend der Kenntnisse der beteiligten Personen verteilt werden können und damit auch bei den unterschiedlichen Testern eine gute Akzeptanz für das Werkzeug erzielt wird.

3. Installation

Auf der Website von FitNesse findet man unter dem Reiter „Download“ eine .jar-Datei mit dem Namen `fitnesse-standalone.jar`, die auf dem eigenen Rechner mit

```
java -jar fitnesse-standalone.jar
```

ausgeführt werden kann. Damit steht dann sofort eine lokale Installation von FitNesse zur Verfügung. FitNesse legt beim ersten Start automatisch eine Seitenstruktur an, die aus einer Startseite (und mehreren Hilfeseiten) besteht und im Browser über <http://127.0.0.1/> erreichbar ist.

Dieser einfache und schnelle Einstieg eignet sich in erster Linie zur Evaluierung der Software und ist auch ohne besondere Berechtigungen des Anwenders möglich.

FitNesse kann auch in einem Netzwerk bereitgestellt werden. Über entsprechende Kommandozeilenparameter lassen sich verschiedenste Parameter, wie z.B. der Port oder eine Liste von berechtigten Benutzern, übergeben. Damit ist FitNesse auch skalierbar und kann unabhängig von der Größe und geographischen Dislozierung des Testteams eingesetzt werden. Die Benutzerdokumentation von FitNesse hat ein eigenes Kapitel zur Administration, aus dem entnommen werden kann, wie FitNesse in eine bestehende Netzwerkumgebung eingebunden werden kann.

Die Tests können auch über Kommandozeile oder REST-Protokoll gestartet werden, was die Integration z.B. in Continuous-Integration-Lösungen möglich macht.

4. Funktionsbeschreibung

Der Anwender nimmt FitNesse als ein Wiki wahr, dass er über eine URL mit Hilfe eines Web-Browsers aufrufen kann. In dem Web-Browser kann der Anwender auch die Wiki-Seiten, in denen die Tests organisiert sind, bearbeiten und die Tests ausführen.

Die Teile, die ausführbaren Testcode repräsentieren, werden in Wiki-Tabellen organisiert, die durch ein Ausrufezeichen („!“) eingeleitet werden. Das Ausrufezeichen wird auch für bestimmte Schlüsselworte genutzt, die z.B. Überschriften formatieren. Alle anderen Teile werden als Text auf den Seiten dargestellt und bei der Ausführung des Tests ignoriert.

Damit lassen sich die Tests auf einer Wiki-Seite auch dokumentieren. Das Wiki kann somit sowohl zur Testausführung als auch zur Dokumentation genutzt werden.

In der Abbildung 1 ist die Architektur von FitNesse schematisch dargestellt. Eine Wiki-Seite, für die eine Testausführung erfolgen soll, wird zunächst auf darauf untersucht, welche Technik, FIT oder SLIM verwendet werden soll. Diese Techniken werden in Kapitel 4.1.1 genauer behandelt.

Von der verwendeten Technologie werden dann Fixtures aufgerufen, die dann mit dem System unter Test (SUT) kommunizieren. Die Ergebnisse werden dann direkt auf den Wiki-Seiten für den Anwender präsentiert.

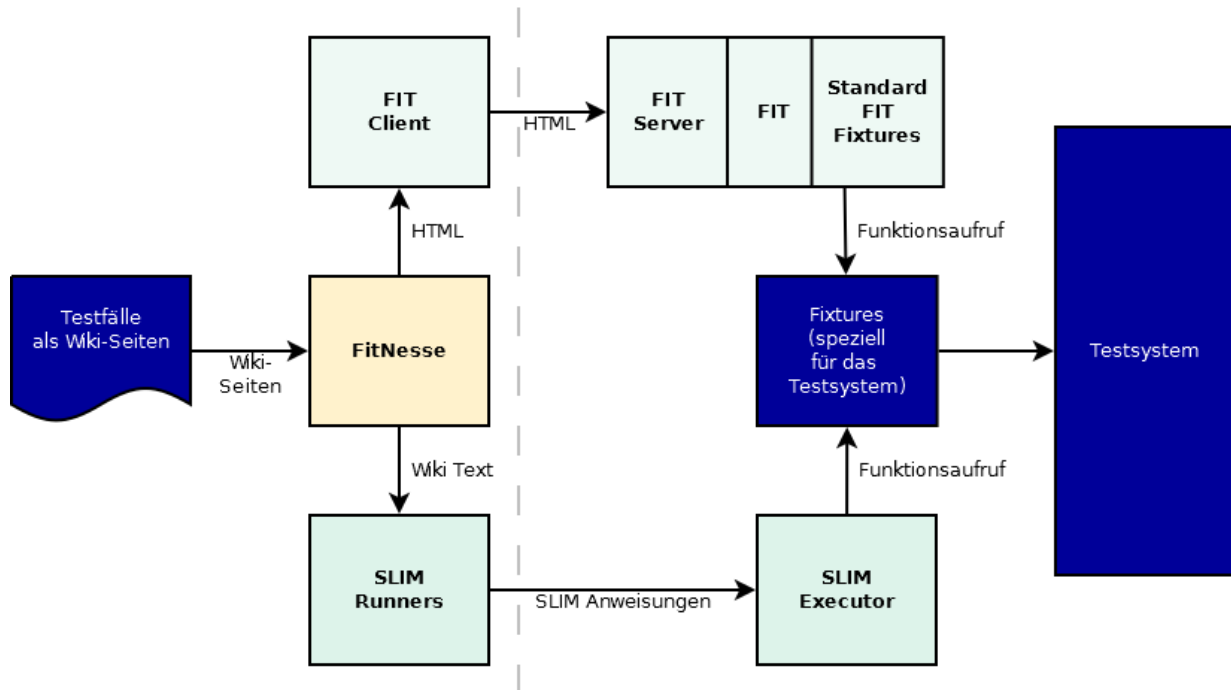


Abbildung 1: schematischer Aufbau von FitNesse

In den folgenden Kapiteln werden die einzelnen Bausteine, aus denen sich der automatisierte Akzeptanztest zusammensetzt, etwas genauer behandelt.

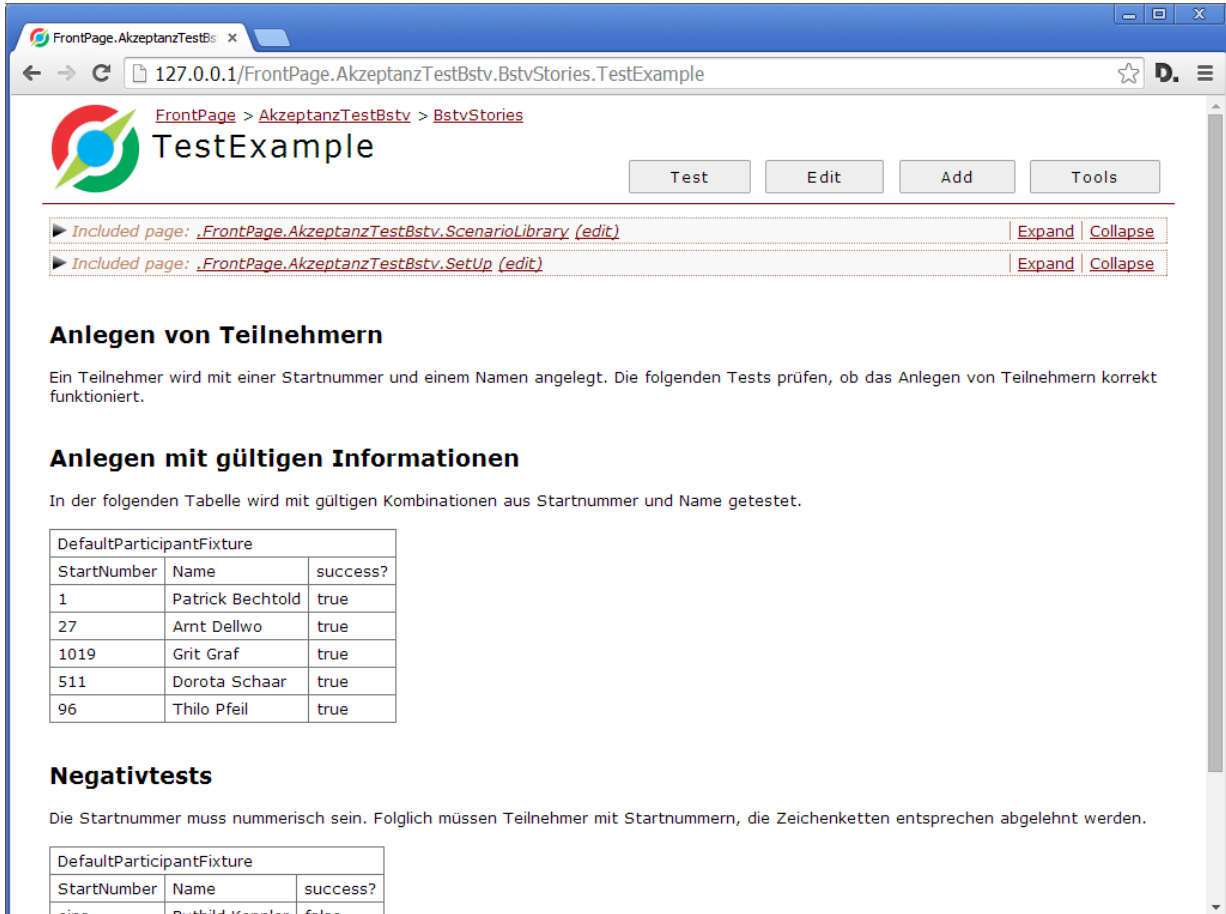
4.1. Wiki-Seiten

Innerhalb des Wiki von FitNesse kann der Anwender Wiki-Seiten anlegen, die hierarchisch organisiert sind. Jede Wiki-Seite muss einen Typ haben. In der vorliegenden Version gibt es die Typen „Static“, „Test“ und „Suite“. Ein Beispiel für eine Seite vom Typ „Test“ ist in Abbildung 2 zu sehen.

Jede Wiki-Seite hat einen Kopf-Bereich, in dem man den Pfad von der Hauptseite zur derzeit angezeigten Seite findet. Die Schaltflächen in der Kopfzeile sind ebenfalls auf jeder Seite zu finden. Über diese kann der Anwender die Funktionen in dem Wiki aufrufen. Die meisten lassen sich auch über Kommandos erreichen, die der Anwender direkt in der Adresszeile des Browsers eingeben kann.

Unabhängig vom Typ der Wiki-Seite kann der Inhalt über entsprechende Steuerzeichen als Anweisung für FitNesse oder als reiner Text deklariert werden. Somit kann eine Wiki-Seite sowohl Test- als auch Dokumentationscharakter haben und der Testdesigner kann im Grunde alle Informationen zu in einer Wiki-Seite zusammenhalten.

Im Folgenden werden die einzelnen Typen von Wiki-Seiten etwas genauer beschrieben.



The screenshot shows a web browser window with the URL `127.0.0.1/FrontPage.AkzeptanzTestBstv.BstvStories.TestExample`. The page title is "TestExample" and it features a navigation menu with "Test", "Edit", "Add", and "Tools" buttons. Below the navigation, there are two included pages: `..FrontPage.AkzeptanzTestBstv.ScenarioLibrary (edit)` and `..FrontPage.AkzeptanzTestBstv.SetUp (edit)`, each with "Expand" and "Collapse" links.

Anlegen von Teilnehmern

Ein Teilnehmer wird mit einer Startnummer und einem Namen angelegt. Die folgenden Tests prüfen, ob das Anlegen von Teilnehmern korrekt funktioniert.

Anlegen mit gültigen Informationen

In der folgenden Tabelle wird mit gültigen Kombinationen aus Startnummer und Name getestet.

DefaultParticipantFixture		
StartNumber	Name	success?
1	Patrick Bechtold	true
27	Arnt Dellwo	true
1019	Grit Graf	true
511	Dorota Schaar	true
96	Thilo Pfeil	true

Negativtests

Die Startnummer muss numerisch sein. Folglich müssen Teilnehmer mit Startnummern, die Zeichenketten entsprechen abgelehnt werden.

DefaultParticipantFixture		
StartNumber	Name	success?
ein	Buthild Kennler	false

Abbildung 2: Beispiel für eine Testseite in FitNesse

4.1.1. Wiki-Seiten von Typ „Static“

Diese Seiten eignen sich für allgemeine Informationen, die der Dokumentation dienen. Dieser Typ wird auch für Seiten mit reservierten Schlüsselworten genutzt. Als Beispiel sind die beiden Seiten zu nennen, die in Abbildung 2 in den Rahmen unterhalb der Kopfzeile zu sehen sind:

- `FrontPage.AkzeptanzTestBstv.ScenarioLibrary`
- `FrontPage.AkzeptanzTestBstv.SetUp`

In FitNesse gibt es in der hier betrachteten Version acht Schlüsselwörter. Wie diese lauten und wofür diese gedacht sind, ist in der Hilfe von FitNesse so gut beschrieben, dass an dieser Stelle lediglich der Seitentyp „SetUp“ als Beispiel herausgestellt werden soll.

Seiten, die das Schlüsselwort „SetUp“ verwenden, werden in Abhängigkeit der hierarchischen Einordnung auf Seiten vom Typ „Test“ inkludiert. Wenn eine Seite eine SetUp-Seite inkludiert, dann wird diese Seite vor den Inhalten der Test-Seite ausgeführt. Somit können hier auch Inhalte zu finden sein, die im Kapitel 4.1.3 behandelt werden.

Aus Sicht des Anwenders macht die Verwendung des Typ „Static“ für alle Seiten Sinn, die keinen eigenständigen Testcode darstellen, denn diesen Seiten fehlt eine Schaltfläche, über die der Anwender eine oder mehrere Testseiten zur Ausführung bringen kann. In Abbildung 2 ist im Kopfbereich exemplarisch die Schaltfläche [Test] zu sehen.

4.1.2. Wiki-Seiten vom Typ „Suite“

Die Seiten vom Typ „Suite“ können als Elternknoten für verschiedene Seiten genutzt werden, die gemeinsam aus Sicht des Testens eine Einheit bilden. Diese Seite bietet dem Anwender mit der Schaltfläche [Suite] die Möglichkeit, alle Kindseiten zur Ausführung zu bringen.

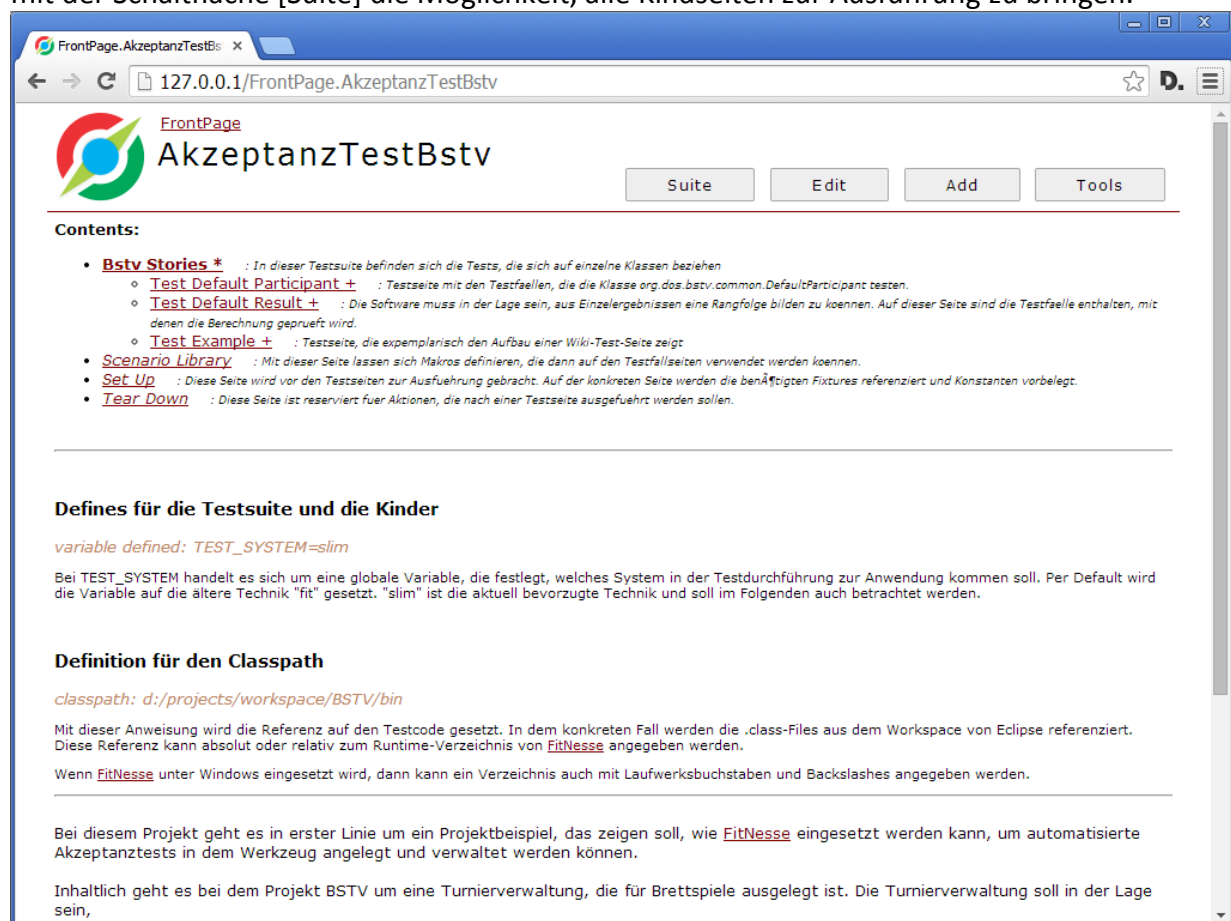


Abbildung 3: Beispiel einer Wiki-Seite vom Typ "Suite"

In der Abbildung 3 ist ein konkretes Beispiel für eine Testsuite zu erkennen. Unter der Kopfzeile ist zunächst ein Inhaltsverzeichnis zu sehen, in dem die Seiten gelistet sind, die unter der Seite „AkzeptanzTestBstv“ angeordnet sind.

Mit „Bstv Stories“ ist eine weitere Suite innerhalb der Suite.

Die anderen drei Seiten „ScenarioLibrary“, „Set Up“ und „Tear Down“ sind Seiten mit reservierten Namen für spezielle Funktionen, die bereits in Kapitel 4.1.1 an gesprochen wurden.

Ansonsten sind auf dieser Seite im Wesentlichen Definitionen enthalten, die für die Ausführung der Tests von Interesse sind. In den Kapiteln in unteren Bereich der Abbildung 3 erkennt man zwei Zeilen in hellbrauner Schrift. Der Rest dient der Erläuterung für den Leser.

Die Tatsache, dass Inhalte der Programmierung und Dokumentation so einfach nebeneinander stehen können, wird als großer Vorteil angesehen, da die Software (FitNesse?) damit für den Entwickler und den Tester gleichermaßen lesbar und verständlich gestaltet werden kann.

Dieses wird noch deutlicher, wenn man sich die Seite im Bearbeitungsmodus ansieht. In Abbildung 4 sieht man den „Quellcode“ der Suite und es fallen die Ausrufezeichen zu Beginn der Zeilen auf. In der ersten Zeile wird mit `!contents` festgelegt, wie das Inhaltsverzeichnis aussehen soll. Die genaue Bedeutung der einzelnen Parameter kann man im Benutzerhandbuch von FitNesse nachschlagen und soll an dieser Stelle nicht näher beschrieben werden.

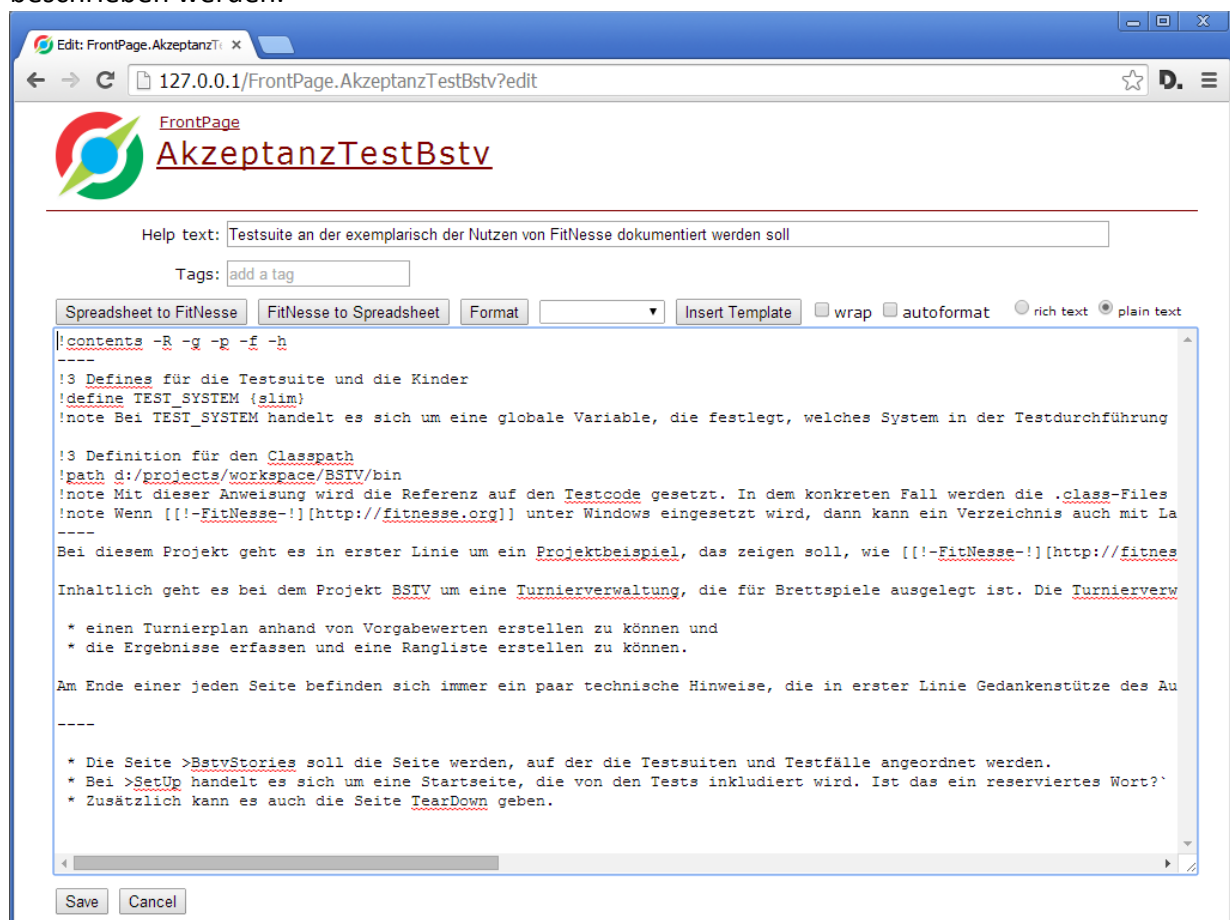


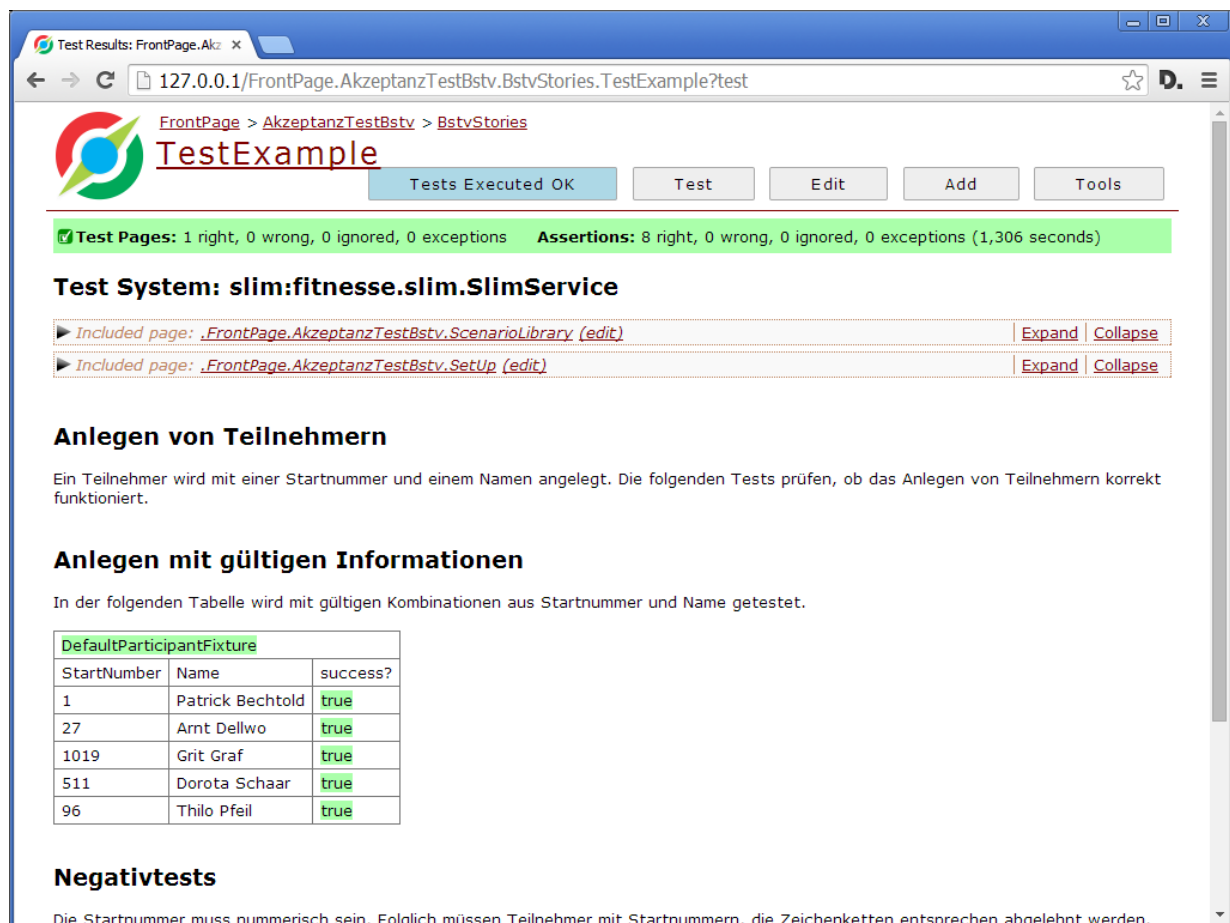
Abbildung 4: Bearbeitungsansicht einer Wiki-Seite von Typ "Suite"

4.1.3. Wiki-Seiten vom Typ „Test“

Die Seitentypen, die in den vorangegangenen Kapiteln besprochen wurden, dienen im Wesentlichen der Strukturierung von Inhalten. Was noch nicht vorgestellt wurde, ist die Darstellung von Testfällen, die zur Ausführung gebracht werden können. Die eigentlichen Testfälle werden auf den Seiten vom Typ „Test“ definiert. In der Abbildung 2 ist ein Beispiel für eine Testseite zu sehen.

Aus technischer Sicht ist es vorstellbar, dass ein Testframework mit einer einzigen Testseite auskommt, auf der alle Testfälle implementiert sind. Aus fachlicher Sicht sollte eine Testseite nur so viele Testfälle enthalten, dass die Seite für den Tester übersichtlich bleibt.

Im konkreten Fall wurde das Thema „Anlegen von Teilnehmern“ als eine Einheit angesehen, das auf einer Testseite behandelt werden soll.



The screenshot shows a web browser window with the URL `127.0.0.1/FrontPage.AkzeptanzTestBstv.BstvStories.TestExample?test`. The page title is "TestExample" and the breadcrumb is "FrontPage > AkzeptanzTestBstv > BstvStories". The page contains a navigation bar with buttons for "Tests Executed OK", "Test", "Edit", "Add", and "Tools". A green status bar reports: "Test Pages: 1 right, 0 wrong, 0 ignored, 0 exceptions" and "Assertions: 8 right, 0 wrong, 0 ignored, 0 exceptions (1,306 seconds)". Below this, the test system is identified as "slim:fitness.slim.SlimService". Two included pages are listed: ".FrontPage.AkzeptanzTestBstv.ScenarioLibrary" and ".FrontPage.AkzeptanzTestBstv.SetUp", each with "Expand" and "Collapse" links. The main content area is titled "Anlegen von Teilnehmern" and contains a description: "Ein Teilnehmer wird mit einer Startnummer und einem Namen angelegt. Die folgenden Tests prüfen, ob das Anlegen von Teilnehmern korrekt funktioniert." Below this is a section "Anlegen mit gültigen Informationen" with a note: "In der folgenden Tabelle wird mit gültigen Kombinationen aus Startnummer und Name getestet." A table follows:

StartNumber	Name	success?
1	Patrick Bechtold	true
27	Arnt Dellwo	true
1019	Grit Graf	true
511	Dorota Schaar	true
96	Thilo Pfeil	true

The page concludes with a section "Negativtests" and a note: "Die Startnummer muss numerisch sein. Folglich müssen Teilnehmer mit Startnummern, die Zeichenketten entsprechen abgelehnt werden."

Abbildung 5: Beispiel einer Wiki-Seite nach Ausführung der Testfälle

Über die Schaltfläche [Test] lassen sich die Tests, die auf der aktuellen Testseite zu sehen sind, ausführen. Nach einer Testausführung ändert sich die Darstellung, wie in Abbildung 5 zu sehen ist.

Unterhalb der Kopfzeile ist eine Zusammenfassung des Testergebnisses zu sehen. Im konkreten Beispiel wurden 8 Testfälle ausgeführt, von denen 8 als erfolgreich ausgeführt bewertet wurden. Die Testfälle werden in FitNesse in Tabellen organisiert, wie z.B. in der Tabelle „DefaultParticipantFixture“. In dieser Tabelle definiert die Überschrift, welche Klasse instanziiert werden soll. Die Spalten stehen für den Aufruf von Methoden der Klasse DefaultParticipantFixture. Die Methoden der Klassen müssen nicht exakt angegeben werden, da FitNesse automatisch dem Spaltennamen auch Präfixe, wie z.B. „set“ voranstellt. Somit kann der Entwickler wie gewohnt eine Methode `.setName()` implementieren und der Testdesigner kann mit der Überschrift „Name“ die Setter-Methode aufrufen.

Eine Besonderheit stellt die Spalte „success?“ dar. Diese Spalte steht auch für einen Methodenaufruf und das Fragezeichen zeigt an, dass hier ein Test erfolgen soll. Der Rückgabewert der Methode wird mit dem jeweiligen Wert in der Zelle verglichen.

Jede Zeile entspricht in diesem Fall einem Testfall, der ausgeführt werden soll. Und in der Tabelle ist jeweils der Anteil farblich hinterlegt, der zu einer Prüfung gehört.

Tests, die erfolgreich ausgeführt wurden, werden grün hinterlegt und solche, die fehlgeschlagen sind, werden rot hinterlegt. Gelbe Farbe wird immer dann verwendet, wenn zum Beispiel Klassen nicht gefunden wurden oder andere Warnungen auszugeben sind.

Jede Testausführung wird protokolliert und kann auch später noch nachvollzogen werden.

4.2. Testausführungsschicht in FitNesse

Die Testfälle, die in Form von Wiki-Seiten vorliegen, werden mit einer der beiden Engines ausgeführt. Der Anwender kann durch entsprechende Direktiven entscheiden, ob die Wiki-Seiten mit der FIT- oder der SLIM-Engine verarbeitet werden sollen.

FIT steht für „Framework for Integrated Testing“ und ist historisch gesehen die erste Engine von FitNesse gewesen. Man kann auch sagen, dass FitNesse im Grunde das Front-End für FIT ist. FIT interpretiert die Wiki-Seiten und bringt die Tabellen, die Testfälle repräsentieren, zur Ausführung.

SLIM steht für „Simple List Invocation Method“. In dieser Engine sind die Präsentationsschicht und die Testausführungsschicht stärker voneinander getrennt. Wenn man sich für SLIM entscheidet, dann sind die Testfälle auch in Tabellenform anzulegen. In diesem Fall werden aber nicht die HTML-Seiten verarbeitet sondern es gibt zwischen FitNesse und SLIM ein Protokoll, über das Testfälle und Ergebnisse transportiert werden.

Die SLIM-Engine ist dadurch, dass sie sich rein um die Testausführung kümmert und die Ergebnisse zurückmeldet, deutlich leichtgewichtiger und performanter.

Damit wurde die Möglichkeit eröffnet, dass der Standard SLIM-Server, der Fixtures in Java erwartet, durch andere Komponenten ausgetauscht werden kann. Auf der Website von

FitNesse finden sich im Bereich Plug-ins Komponenten, die das SLIM Protokoll für Ruby, .net, C++ und andere Programmiersprachen implementiert haben.

4.3. Fixtures

Bei Fixtures handelt es sich um Programmteile, die in der Regel vom Anwender von FitNesse selbst zu implementieren sind.

Wie in Abbildung 1 zu erkennen ist, rufen beide Techniken aus FitNesse heraus Funktionen aus den Fixtures auf, die dann das eigentliche SUT im Zugriff haben.

Auf den ersten Blick mag man sich fragen, warum FitNesse nicht direkt auf das SUT zugreift? Zunächst ist zu sagen, dass dieses technisch grundsätzlich möglich ist, jedoch muss in diesem Fall der Testdesigner eine genaue Kenntnis der Klassen des SUT haben, um diese direkt aufrufen zu können.

Ferner versucht FitNesse mit einem sehr raffinierten Algorithmus aus einzelnen Worten wieder Methodennamen für den Aufruf von internen Modulen und den Testmethoden zu generieren. Damit kann der Anwender einfacher in einem formalen Sprachstil als in einer Programmiersprache entwickeln.

Der Zweck der Fixtures lässt sich auch über das Strukturmuster Fassade begründen, wie es in dem Buch Entwurfsmuster der Gang of Four (GOF) definiert wurde. Danach kann es sein, dass der Test eines Geschäftsvorfalles, wie z.B. das „Anlegen eines Kunden“ aus vielen Aktionen bestehen kann, die der Tester in der richtigen Reihenfolge im Testfall beschreiben muss. Diese Komplexität der Aufgabe kann hinter einer Fassade verborgen werden, die dann z.B. eine einzelne Methode anbietet, in der die kleinen Einzelschritte implementiert sind. Der Testfall innerhalb der Wiki-Seite wird damit kürzer und ist für den fachlichen Tester leichter zu verstehen.

Darüber hinaus können Fixtures auch einen Beitrag dazu leisten, dass die Testfälle nicht bei jedem Release geändert werden müssen. Wenn z.B. strukturelle Änderungen in der Architektur vorgenommen wurden, dann können auch diese Änderungen über das Fixture in Richtung FitNesse verborgen werden und machen keine Entwicklung oder Modifikation von Testfällen erforderlich.

Eine weiterer Anwendungsfall für Fixtures können umfangreiche Vor- oder Nachbereitungsschritte im Zusammenhang mit dem Test sein, die dann entweder mit wenigen Schlüsselwörtern vom Tester aus dem Testfall aufgerufen werden oder aber die Schlüsselwörter kennen selbst die Abhängigkeiten und führen die Schritte nach einer vorherigen Prüfung automatisch aus, ohne dass der Tester explizit ein Kommando dafür schreiben muss.

Damit tragen Fixtures zu einer Flexibilität bei, die das Framework alleine nicht gewährleisten könnte, und erlauben darüber hinaus einen Einsatz von FitNesse vom Komponententest über alle Teststufen bis zum Akzeptanztest.

5. Quellen und Weiterführendes

FitNesse Website

<http://fitnesse.org/FrontPage>

Benutzerhandbuch für FitNesse

<http://fitnesse.org/FitNesse.UserGuide>

Informationen zum Entwurfsmuster „Fassade“

[http://de.wikipedia.org/wiki/Fassade_\(Entwurfsmuster\)](http://de.wikipedia.org/wiki/Fassade_(Entwurfsmuster))

ISTQB® /GTB Standardglossar der Testbegriffe

<http://www.german-testing->

board.info/fileadmin/gtb_repository/downloads/pdf/glossar/CT_Glossar_DE_EN_V23.pdf