

Testframework

Testen von Webseiten

Mit dem Testframework von Muth Partners steht eine Softwarelösung zur Verfügung, die es erlaubt, Webseiten automatisiert zu testen. Testen in Webprojekten wird für unsere Kunden transparent und effizienter.

Dirk O. Schweier

10.01.2011

Changes

Wer	Wann	Was
DS	21.12.2010	Ersterstellung
DS	10.01.2011	Überarbeitung des Dokuments

1. Inhaltsangabe

1. Inhaltsangabe	3
2. Motivation zur Testautomatisierung	4
3. Aufbau des Testframeworks	4
3.1. Softwarekomponenten	5
3.1.1. Java	5
3.1.2. Selenium	5
3.1.3. TestNG	5
3.2. Architektur des Testframeworks	6
3.3. Grenzen des Testframeworks	6
4. Das Testframework im Praxiseinsatz	7
4.1. Einsatz bei SinkaCom AG.....	7
4.1.1. Vorgehen bei Sinkacom	7
4.1.2. Vorteile für Sinkacom und ihre Kunden	7
4.2. Einsatz bei G. Muth Partners GmbH.....	8
5. Nutzen des Testframeworks	8
6. Informationen	10

2. Motivation zur Testautomatisierung

An ein Produkt werden von unterschiedlichen Interessensgruppen Anforderungen gestellt. Aus diesem Grund werden in allen Branchen die Produkte im Rahmen der Qualitätssicherung bestimmten Prüfungen unterzogen.

Im Umfeld der Softwareentwicklung ist Testen eine Maßnahme der Qualitätssicherung. Da Webseiten mit immer mehr Funktionalität erstellt werden, die über Skripte oder andere Programmieretechniken umgesetzt werden, können sie als komplexe Softwarelösungen angesehen werden, die intensiv getestet werden müssen.

In der Vergangenheit wurden Webseiten überwiegend manuell getestet. Diese Form der Tests ist zeitaufwändig und auch fehleranfällig. Die Qualität der Ergebnisse hängt nicht selten von der Motivation und Qualifikation des Testers ab. Hinzu kommt, dass die Arbeit vor allem dann monoton wird, wenn mehrere Testdurchläufe durchzuführen sind. In den späteren Testdurchläufen können dann leicht Fehler übersehen werden, da Aufmerksamkeit und Motivation des Testers nachlassen können.

Die **Testautomatisierung** bietet genau in diesem Umfeld Vorteile. Definierte Tests lassen sich beliebig oft wiederholen, ohne dass die menschlichen Effekte auftreten. Die Tests werden von einer Software durchgeführt, die mit gleichbleibender Genauigkeit Ist-Zustände mit Soll-Zuständen vergleicht und Abweichungen protokolliert. Der manuelle Aufwand wird auf die Auswertung der Protokollierung reduziert.

Die Testautomatisierung stellt Anforderungen an die Qualifikation einzelner Tester. Testautomatisierung bedeutet, dass Testwerkzeuge eingesetzt und Softwareentwicklung betrieben werden muss, was zusätzlichen Aufwand für die Entwicklung der Tests und Testskripte erfordert. Dieser Zusatzaufwand sollte nur dann getrieben werden, wenn der Aufwand der Testdurchführung mindestens um das Maß sinkt, dass im Testdesign zusätzlich anfällt.

Das vorliegende, von Muth Partners entwickelte Testframework spart Entwicklungszeit in der Testdurchführung ein, da die eigentliche Testdurchführung ohne manuelle Aktionen laufen kann und jeder einzelne Test weniger Zeit in Anspruch nimmt als eine manuelle Ausführung. Beim Design des Testframeworks wurde Wert darauf gelegt, dass es mit geringem Aufwand auf unterschiedliche Webseiten angewendet werden kann.

Mit dem Testframework wird die Qualität der Webseiten, die entwickelt werden, gesteigert. Gleichzeitig laufen die Tests reproduzierbar und automatisch ab, wodurch der Zeitaufwand für die Tests verringert wird, was insgesamt den Zeitraum zwischen Beauftragung einer Webpräsenz und Abschluss des Auftrags verkürzt. Mit dem Einsatz dieser Lösung geht eine Produktivitätssteigerung des Unternehmens einher.

3. Aufbau des Testframeworks

Das Testframework ist eine Entwicklung von Muth Partners. In diesem Kapitel werden zunächst die Komponenten beschrieben, die in der Softwarelösung zum Einsatz kommen, bevor in einem zweiten Teil auf die Architektur der Lösung eingegangen wird.

3.1. Softwarekomponenten

3.1.1. Java

Das Testframework besteht zu einem wesentlichen Teil aus Java-Quellcode.

Java ist eine objektorientierte Programmiersprache und eingetragenes Warenzeichen der Firma Sun Microsystems, welche Ende Januar 2010 von Oracle übernommen wurde. Die Programmiersprache genießt sowohl im kommerziellen und nicht-kommerziellen Bereich eine hohe Akzeptanz und Verbreitung.

Java eignet sich für die vorliegende Aufgabenstellung aufgrund der guten Dokumentation der Sprache und des großen Sprachumfangs. Durch das Konzept der Bibliotheken lassen sich auch Fremdkomponenten in das Testframework integrieren, die Funktionalität implementieren, die im Standard nicht zur Verfügung gestellt wird.

3.1.2. Selenium

Bei **Selenium** handelt es sich um ein Testframework für Webanwendungen. Es wurde von einem Programmiererteam der Firma ThoughtWorks entwickelt und als Freie Software unter der Apache-2.0-Lizenz veröffentlicht.

Mit seiner Hilfe ist es möglich, Interaktionen mit einer Webanwendung aufnehmen zu lassen und diese Tests automatisiert beliebig oft zu wiederholen. Es kann vor allem Entwicklern von Webanwendungen sehr viel Tipparbeit abnehmen – beispielsweise beim Ausfüllen von Webformularen – und macht das Testen dadurch schneller, flexibler und verlässlicher.

Selenium basiert rein auf HTML und JavaScript und wird in dem Testframework genutzt, um die Webseiten, die unter Test sind, zu steuern oder Informationen abzufragen.

3.1.3. TestNG

TestNG steht für „Test Next Generation“. TestNG ist für den Unit-Test von Software entwickelt worden. Ziel war es, frühzeitig die Annotation-Features von Java 1.5 (Tiger) zu nutzen und ein umfassenderes und besseres Testframework zu schreiben, als es die JUnit-3.*-Familie bereitstellt. Entstanden ist ein Framework, das einige Aspekte mehr aufweist als JUnit 4 und vielleicht auch andere Projekte adressiert.

Cedric Beust startete TestNG 2003, da JUnit einige Mängel in Bezug auf Themen wie Gruppenbildung, Parametrisierbarkeit etc. aufwies und JUnit nicht besonders schnell auf Java Annotations reagierte. Ein Beispiel für den Mangel in JUnit sind die Methoden setUp() und tearDown(). Sie haben einen Klassenscope. Oftmals wünscht sich der Entwickler aber eine feinere (vor und nach jeder einzelnen Methode) oder größere Reichweite (vor jeder Testgruppe oder jeder Testsuite).

Das Tool wird in zwei Richtungen genutzt. Zum einen dient es der Gliederung und Steuerung der automatisierten Tests. Es ist dafür sehr gut geeignet, weil es zahlreiche Funktionen zum Abgleich von Informationen bietet und Exceptions mit parametrierbaren Fehlermeldungen erzeugt, wenn Bedingungen nicht erfüllt werden.

TestNG wird eingesetzt, um die Qualität des Testframeworks selbst mit Unit-Tests zu sichern.

Testfälle mit ganz unterschiedlichen Schwerpunkten und Testzielen können ohne Probleme im selben Projekt implementiert werden. Sie behindern einander nicht, da die Ausführung der Testfälle über eine Steuerungsdatei parametrisiert wird.

3.2. Architektur des Testframeworks

Die Architektur des Testframeworks basiert auf Erkenntnissen aus einem Pilotprojekt zur Testautomatisierung:

- Die Algorithmen der Testfälle sind sich häufig ähnlich.
- Die Testdaten sollen aus einer anderen Quelle als der Sourcecodedatei kommen.

Zu diesen Erkenntnissen kam zusätzlich auch die Anforderung des Kunden, dass er eine Softwarelösung haben möchte, die möglichst auf beliebige Webseiten angewendet werden kann.

Der erste Schritt hin zum Testframework war Abstraktion. Durch eine sehr abstrakte Beschreibung eines allgemeinen logischen Testfalls ist es gelungen, einen Algorithmus mit zugehörigen Datenstrukturen zu implementieren. Die Grundidee basiert darauf, dass eine Webseite, die geprüft werden soll, bestimmte Merkmale hat, die für Tests von Relevanz sind. Diese Merkmale werden als Objekte in einer Struktur gehalten. Auf dieser Struktur werden dann verschiedene Aktionen ausgeführt, die sowohl Bedienung oder Prüfung sein können.

Bei der Architektur von allgemeinem Testfall, Seitenstruktur und Aktion wurde auch der Aspekt der Erweiterbarkeit berücksichtigt, damit das Testframework sich auch in Zukunft neuen Anforderungen stellen kann.

Desweiteren wurde eine Trennung von Testlogik und Testdaten erreicht. Dazu wurde eine Schnittstelle geschaffen, über die die Testdaten aus Quellen eingelesen werden können. In der aktuellen Version sind Schnittstellen zu Dateien mit den Formaten

- Microsoft Excel und
- CSV implementiert worden.

Diese Schnittstellen lassen sich ebenfalls erweitern, so dass vom Prinzip her auch Datenbanken und andere Dateiformate vorstellbar sind.

3.3. Grenzen des Testframeworks

Das Testframework setzt auf Selenium 1.0.1 auf. Damit ist aktuell keine Unterstützung von Flash gegeben. Es existiert jedoch eine Erweiterung, mit der es auch möglich ist, Flash-Applikationen zu testen. Diese Variante ist noch zu evaluieren.

Grundsätzlich sind Tests nur sehr eingeschränkt möglich, wenn auf den Webseiten Re-Captchas verwendet werden. Diese Funktion soll dazu dienen, dass Webformulare nicht durch Roboter ausgefüllt werden können, um z.B. eine Spam-Attacke durchzuführen. Hier ist genau zu prüfen, welche Art von Tests gefahren werden kann und wie diese durch manuelle Tests zu ergänzen sind.

4. Das Testframework im Praxiseinsatz

4.1. Einsatz bei SinkaCom AG

Die SinkaCom AG ist eine Webagentur aus Wiesbaden, die als Typo3 Spezialist und Online Marketing Agentur für Kunden in ganz Deutschland arbeitet.

Webseiten im Kundenauftrag werden von SinkaCom häufig unter Nutzung des Content Management System Typo3 erstellt. Dieses System erlaubt die Nutzung von Modulen, die von Typo3 bereitgestellt werden. Darüber hinaus lassen sich auch Module durch Entwickler erstellen und modifizieren, die dann in Kundenprojekten eingesetzt werden können.

Um Kunden Webseiten mit einem hohen Qualitätsstandard anbieten zu können, hat SinkaCom eine Art „Baukasten“ entwickelt, aus dem dann Kundenprojekte abgeleitet werden sollen. Das Qualitätsniveau der Kundenseite steht damit in direkter Beziehung zum Qualitätsniveau des Baukastens.

4.1.1. Vorgehen bei Sinkacom

Mit dem Einsatz des Testframeworks verfolgt SinkaCom zwei wesentliche Ziele:

- **Automatisiertes Testen des Baukastens**

Die Einzelmodule des Baukastens sind die „Bausteine“ aus denen dann Kundenseiten erstellt werden sollen. Nur wenn diese Bausteine stabil und gut ausgetestet sind, lassen sich Webseiten erstellen, die den Qualitätsanforderungen von SinkaCom und seinen Kunden genügen.

Der mehrstufige Testprozess der SinkaCom setzt in der ersten Stufe auf den Einzelmodulen, wie z.B. einem Eingabefeld für eine E-Mail-Adresse, auf.

Aus diesen getesteten Modulen werden dann Seiten entwickelt. Auf den Seiten wird in der zweiten Teststufe das Zusammenspiel von Modulen getestet mit dem Ziel, Seiten oder Formulare zu haben, die dann nur durch Konfiguration an Bedürfnisse eines konkreten Kunden angepasst werden können.

- **Automatisiertes Testen der Kundenseiten**

Nur mit einem Abnahmetest des Kunden kann dargestellt werden, dass das fertige Produkt die Anforderungen des Kunden erfüllt und die geforderte Funktionalität umfasst.

Die Testskripte und Testdaten aus dem Automatisierten Test des Baukastens können Grundlage für den Test auf Kundenseite sein.

Durch die Trennung von Testdaten und Testlogik kann der Kunde auf einfache Weise Daten zur Verfügung stellen, die dann für ihn automatisiert vor Auslieferung getestet werden.

4.1.2. Vorteile für Sinkacom und ihre Kunden

Mit der Einführung des Testframeworks bei SinkaCom ist es gelungen, dem Kunden durch Reportings aufzuzeigen, mit welchen Daten und in welchem Umfang die Tests durchgeführt wurden. Der Kunde kann an den Tests beteiligt werden, in dem er Daten liefert, die er dann im Testbericht wiederfindet.

SinkaCom ist davon überzeugt, dass diese Mehrstufigkeit des Testens die Voraussetzung ist, um Fehler möglichst früh zu finden und auf einer stabilen Grundlage qualitativ hochwertige Kundenseiten entwickeln zu können. Durch die Tests der Standardvorlage für Kundenprojekte und den Test der fertigen Kundenseiten werden mögliche Fehler schnell entdeckt, die auf eine nicht korrekte Verwendung der Vorlage zurückzuführen sind. Der Testprozess für die Kundenseiten kann ebenfalls effizienter ablaufen, da mit Testdaten gearbeitet werden kann, die auch für die Tests der Standardvorlage genutzt werden.

SinkaCom hält nur durch diese automatisierten Tests der Kosten und Aufwand der Qualitätssicherung in einem vertretbaren Rahmen.

4.2. Einsatz bei G. Muth Partners GmbH

Die G. Muth Partners GmbH ist eine Unternehmensberatung, die Softwaretest und Qualitätsmanagement als einen Schwerpunkt der Beratungstätigkeit anbietet.

Mit dem Testframework steht den Beratern von Muth Partners eine Lösung für Projekte zur Verfügung, die einen Fokus auf Testautomatisierung legen.

Grundsätzlich lassen sich Architektur und wesentliche Mechanismen des Testframeworks auch auf Softwarelösungen übertragen, bei denen über eine Benutzeroberfläche Tests automatisiert ausgeführt werden sollen.

Die Berater von Muth Partners sind damit kompetenter Ansprechpartner in Fragen zum Thema Testautomatisierung und Testdesign. Der Nutzen, den die Kunden daraus ziehen können, wurde im vorangegangenen Kapitel am Beispiel der SinkaCom AG aufgezeigt.

5. Nutzen des Testframeworks

- **Das Testframework reduziert Testaufwand.**

Die Testautomatisierung ist ganz allgemein ein Beitrag zur Prozessverbesserung im Unternehmen.

Zwar kommt durch Testautomatisierung der Zusatzaufwand der Entwicklung von Testskripten in der Phase Testdesign hinzu, jedoch lässt der sich in aller Regel in den Phasen Testdurchführung und Testauswertung einsparen.

Die automatische Testdurchführung ist in aller Regel durch einen Tester zu starten. Die Ausführung selbst bedarf keines personellen Aufwands, da die Testskripte automatisch und ohne Benutzerinteraktion ablaufen. In dieser Phase gibt es dann auch keine Vermischung zwischen Testdurchführung und Testauswertung, wie es bei manuellen Tests häufig vorkommt, wenn Tester unmittelbar nach Erkennen einer Abweichung die Abweichungsmeldung abfassen.

- **Das Testframework ist unbestechlich.**

Ein weiterer Vorteil der Automatisierung steckt in dem Grad der Genauigkeit der Prüfungen. Bei manuellen Testern stellt man häufig fest, dass Tester mit fortschreitender Dauer der Testphase nicht mehr so exakt arbeiten, wie zu Anfang. Dieser Effekt tritt vor allem dann auf wenn viele Nachtests mit häufigen Wiederholungen durchzuführen sind. Hier garantiert ein Testskript eine konstante Genauigkeit in der Testdurchführung.

Die Phase Testauswertung bleibt grundsätzlich eine manuelle Tätigkeit, die jedoch effizienter durchgeführt werden kann, da ihr das Testprotokoll des Testframework zu Grunde liegt, in dem die Abweichungen dokumentiert sind.

- **Das Testframework ist auf Webseiten mit beliebiger Struktur vorbereitet.**

Durch die Abstraktion von Testfällen zu einem allgemeinen logischen Testfall ist eine Logik implementiert worden, durch die sich beliebige Webseiten mit dem Framework testen lassen. Der Aufwand zum Einrichten der Tests weiterer Webseiten besteht im Wesentlichen nur in Konfiguration des Testfalls, wenn man voraussetzt, dass die im Testframework vorliegende Funktionalität ausreichend ist.

- **Das Testframework kann aufgrund der Architektur jederzeit erweitert werden, indem**

- neue prüfungsrelevante Strukturelemente implementiert werden können

Mit prüfungsrelevanten Strukturelementen sind Teile der Webseite gemeint, die zur Bedienung der Seite oder für Prüfungen relevant sind. Das können z.B. Eingabefelder oder Schaltflächen sein. Wird ein Element in der vorliegenden Version nicht unterstützt, kann es nachträglich implementiert und zur Verfügung gestellt werden.

- neue Prüfungen und Aktionen auf Webseiten entwickelt werden können

Nach einer erfolgreichen Einführung der Testautomatisierung wird eine Organisation in dem Thema lernen und dadurch neue Möglichkeiten der Testautomatisierung identifizieren und umsetzen wollen. Aus diesem Grund wurde bei der Architektur besonderer Wert darauf gelegt, dass zusätzliche Aktionen nachträglich hinzu gefügt werden können. Auf diese Weise kann das Testframework im Laufe der Zeit „mitwachsen“ und ist damit auch zukunftsorientiert.

- **Das Testframework erlaubt eine Trennung von Testdaten und Testlogik.**

Die Logik des Testframework ist in der Programmiersprache Java implementiert.

Testdaten werden jedoch nicht zwingend von jemandem entworfen, der über Kenntnissen in einer Programmiersprache verfügt. Folglich wäre es eine hohe Einstiegshürde, wenn der Testdesigner die Testdaten im Quellcode hinterlegen soll.

Um diese Hürde abzubauen und eine Arbeitsteilung der Rollen Testautomatisierer und Testdesigner zu erreichen, wurden in dem Testframework eine Schnittstelle geschaffen, über die Testdaten aus CSV-Dateien oder Excel-Dateien zur Laufzeit der Testapplikation geladen werden können.

Mit dieser Trennung ist es, wie im Kapitel 4.1 Einsatz bei SinkaCom AG beschrieben, möglich, den Kunden in die Testphase einzubeziehen, in dem er Testdaten liefern kann, die dann für ihn dokumentiert getestet werden.

Mit dieser Form der Datenhaltung können die konkreten Testfälle rein durch Bearbeiten der Datenquelle erweitert oder reduziert werden, ohne dass Programmierkenntnisse erforderlich sind.

Durch die Trennung von Testdaten und Testlogik kann zusätzlich auch über den Einsatz von Software zur Testdatengenerierung nachgedacht werden, um auf diese Weise Massendaten zu erzeugen.

6. Informationen

Wir freuen uns über Ihre Fragen und Anregungen zu dem beschriebenen Testframework, das wir Ihnen auch gerne live vorführen.

Ansprechpartner

Dirk O. Schweier
Adresse: G. Muth Partners GmbH
Borsigstraße 32
65205 Wiesbaden
Telefonnummer: +49 6122 5981 - 0
EMail-Adresse: dirk.schweier@muthpartners.de

Links

- Muth Partners GmbH (<http://www.muthpartners.de>)
 - diesen und weitere Artikel zum Thema Testen (<http://www.muthpartners.de/Fachwissen>)
- Java (<http://www.java.com/de/>)
- Java-Bibliotheken
 - Java Excel API (<http://jexcelapi.sourceforge.net/>)
 - OpenCSV (<http://sourceforge.net/projects/opencsv/>)
 - ReportNG (<http://reportng.uncommons.org/>)
- Selenium (<http://seleniumhq.org/>)
- TestNG (<http://testng.org/doc/index.html>)