

TestFrame™ meets TestBench: State of the Art Testautomatisierung

Anton Schlatter
LogicaCMG

anton.schlatter@logiacmg.com
www.testframe.de

Die Produktivität in der SW-Entwicklung hat in den letzten Jahren deutlich zugenommen. Dies konnte erreicht werden durch moderne Entwurfsmethoden, eine konsequente Wiederverwendung der Artefakte aus der Entwicklung und durch zusehends integrierte Entwicklungsumgebungen. Demgegenüber erweist sich der Test zusehends als Flaschenhals. Die Dynamik der globalisierten Märkte erhöht den Druck auf die Time-to-Market sowie Quality-to-Market. Ein steigender Test-Workload ist in immer weniger Zeit, mit weniger Ressourcen und zu verminderten Kosten zu bewältigen. Der Testautomatisierung kommt im Rahmen der Effizienzsteigerung in diesem Kontext eine zentrale Rolle zu.

Testautomationsansätze

Die ursprünglichen Versprechen der Werkzeughersteller, mit einfachem Capture&Playback Verfahren schnell und kostengünstig die benötigten Tests automatisieren zu können, haben sich als nicht umsetzbar erwiesen. Die Ergebnisse dieses Vorgehens sind kaum wartbar sowie nur eingeschränkt wieder verwendbar. Neben der Wartungsfalle lauert die Abhängigkeit von technischen Spezialisten. In der Praxis hat man erkannt, dass eine erfolgreiche Test-automatisierung eine methodische Fundierung erfordert und hat sukzessive Konzepte aus der „normalen“ Softwareentwicklung, wie z.B. die Modularisierung, adaptiert. Zurückblickend lassen sich folgende Paradigmen unterscheiden:

- Capture&Playback
- Skriptprogrammierung
- Daten-getriebene Tests
- Keyword-getriebene Tests
- Framework-basierte Testautomation

Betrachtet man die aktuellen Produktgenerationen der großen Werkzeughersteller, dann lässt sich feststellen, dass versucht wird, den keyword-getriebenen Ansatz umzusetzen.

TestFrame™ und die Aktionsworte

Die Keywords der LogicaCMG sind die „Aktionsworte“. Bereits Anfang der 1990er Jahre hat LogicaCMG dieses Basiskonzept zur Testautomatisierung entwickelt und seither in einer Vielzahl von Kundenprojekten erfolgreich umgesetzt. Die „Aktions-

antwort“-Methode ist ein grundlegendes Konzept von TestFrame™, des generischen Testvorgehensmodells der LogicaCMG und wurde v.a. in den 1990er Jahren über verschiedene Kanäle veröffentlicht.

Im klassischen Vorgehen bilden die Aktionsworte das fachliche Vokabular der Tester zur Beschreibung der Testfälle für ein gegebenes System unter Test (SuT). Vor der eigentlichen Testspezifikation wird im Rahmen des Testdesigns die Liste der benötigten Aktionsworte identifiziert und dokumentiert. Die Aktionsworte werden parametrisiert. Die Spezifikation der Testfälle erfolgt durch Sequenzierung von Aktionsworten und Befüllung ihrer Parameter mit den dynamischen Testdaten bzw den erwarteten Ergebnisse. Parallel hierzu werden von den Testingenieuren die Aktionsworte implementiert. Nicht die Testfälle werden somit automatisiert sondern die modularen Bausteine, aus denen sie sich zusammensetzen. Der Initiaufwand für die Bereitstellung eines Testautomaten korreliert somit nicht mit der Anzahl der Testfälle sondern mit der Anzahl der Objekte auf der Benutzerschnittstelle, über die das SuT stimuliert sowie ein Soll-Ist Vergleich durchgeführt werden soll.

Cluster	C12 – Buchungen			
Version	1.1			
Autor	Anton Schlatter			
Testfall	Hier beginnt der erste Testfall			
	C.12.B1.01 Korrekte Berechnung, positiv			
	<i>Nachname</i>	<i>Vorname</i>	<i>KundenNr</i>	<i>Saldo</i>
Erfasse Kunde	Müller	Hans	12345678	10000
Erfasse Kunde	Maier	Klara	23456780	500
	<i>Von Nr</i>	<i>Nach Nr</i>		
Überweise	12345678	23456780	5000	
	<i>KundenNr</i>	<i>Saldo</i>		
Prüfe Saldo	12345678	5000		
Prüfe Saldo	23456780	5500		
Testfall	Hier endet der erste Testfall			
	C.12.B1.02 Korrekte Berechnung, negativ			
	<i>Nachname</i>	<i>Vorname</i>	<i>KundenNr</i>	<i>Saldo</i>
Erfasse Kunde	Muster	Max	35624567	100
Erfasse Kunde	Muster	Peter	45624567	10000
	<i>Von Nr</i>	<i>Nach Nr</i>		
Überweise	35624567	45624567	5000	
	<i>KundenNr</i>	<i>Saldo</i>		
Prüfe Saldo	35624567	-4900		
Prüfe Saldo	45624567	15000		

Abb. 1: Aktionswort-basierte Testspezifikation

Die Aktionswort-Methode bietet erfahrungsgemäß folgende Vorteile:

- Verlässliche Kalkulierbarkeit
- Unterstützung eines Rollenkonzepts
- Parallelisierung und Entkopplung
- Minimale Wartungsaufwände

Für die Umsetzung der Aktionswort-Methode bietet LogicaCMG seit Mitte der 1990er Jahre ein spezialisiertes On-Top Tool für die gängigen Testautomatisierungswerkzeuge an: die ConTest Engine. Die ConTest Engine der LogicaCMG fungiert innerhalb einer Testautomationsarchitektur als generischer Testtreiber, als Testexecution Engine. Vergleichbar einem Interpreter ist er in der Lage, aktionswort-basierte Testspezifikationen zu verarbeiten. Über eine API können SuT-spezifische Aktionsworte integriert werden. Diese Aktionsworte werden für eine GUI-Anwendung klassischerweise in der Skriptsprache des jeweils eingesetzten Testautomatisierungswerkzeugs, das als Interface fungiert, implementiert.

Multi-level Testautomationsarchitektur

Konzeptionell erfuhr die Aktionswort-Methode Ende der 1990er Jahre eine Erweiterung, die unter dem Begriff der „multi-level Testarchitektur“ der interessierten Community auf Fachkonferenzen bekannt gemacht wurde. Es wurden dabei Aktionsworte auf folgenden Ebenen unterschieden:

1. technisch, z.B. GUI-Controls
2. fachlich-funktional, z.B. GUI-Funktion
3. geschäftsprozess, z.B. Prozessschritt

Die weiterentwickelte ConTest Engine ermöglichte es, in den Testspezifikationen mit beliebig tief verschachtelten Aktionsworten zu arbeiten, die zum Testdurchführungszeitpunkt aufgelöst wurden.

Durch dieses Konzept konnte die Wiederverwendbarkeit der implementierten Aktionsworte über die Teststufen hinweg umgesetzt werden. Die im geschäftsprozess-orientierten Abnahmetest benötigten Aktionsworte lassen sich durch Aggregation der bereits vorhandenen technischen oder fachlich-funktionalen Aktionsworte aus dem vorgelagerten Systemtest bereitstellen. Dieses Vorgehen verringerte nochmals deutlich die Initial- und Wartungsaufwände bei einer Testautomation. Ebenfalls konnten die Rüstzeiten für einen Testautomaten nach der Bereitstellung des SuT deutlich reduziert werden, was zu positiven Effekten bei der Time-to-Market beitragen kann.

Framework-basierte Testautomation

Der nächste logische Schritt war die Bereitstellung eines Frameworks zur Testautomation auf Basis des oben beschriebenen Ebenenmodells. Das ConTest Framework der LogicaCMG bietet in der aktuell vorliegenden Version einen vorgefertigten Baukasten mit den atomaren Aktionen für nahezu alle standardmäßig benutzten Controls. Das Framework liefert somit out-of-the-box

die technische Ebene einer multi-level Testautomationsarchitektur. Anwendungsspezifische fachlich-funktionale Aktionsworte, die in den Testspezifikationen benutzt werden, können durch Aggregation dieser atomaren Aktionen beschrieben werden.

Damit konnte die Wiederverwendbarkeit um die Dimension der Anwendungs- und Plattformunabhängigkeit erweitert werden. In der aktuellen Version des Frameworks werden Java/J2EE, C/C++, .NET, PowerBuilder, HTML/Web unterstützt. Im Unterschied zu anderen Framework-orientierten Ansätzen werden hier keine Skripte generiert, sondern die Testspezifikationen werden aus bereits vorhandenen und erprobten Bausteinen zusammengesetzt.

Testspezifikation

Bei einem Einsatz des ConTest Framework können die Testfälle nach wie vor in Excel (siehe Abbildung 1) spezifiziert werden. Das Mapping der dort verwendeten fachlich-funktionalen Aktionsworte auf die technischen erfolgt ebenfalls in Excel über Tabellenvorlagen.

Als aktuelle Weiterentwicklung steht eine Integrationslösung mit der TestBench der Fa. imbus zur Verfügung.

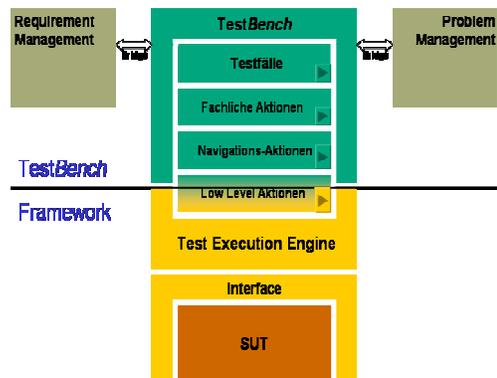


Abb. 2: Integration mit der TestBench

Fazit

Durch die Integration der imbus TestBench mit dem LogicaCMG ConTest Framework steht eine vollständig integrierte, moderne und offene Lösung zur Verfügung, die alle Phasen des Testprozesses optimal unterstützt, einen hohen Automatisierungsgrad gewährleistet und die Produktivität im Test deutlich verbessert.