

SAP-AUTOMATISIERUNG: ERFAHRUNGEN MIT EINER FLEXIBLEN TESTSUITE BEI WECHSELNDEN TESTS



Dr. Dirk Hövener

(E-Mail: dirk.hoeverer@sogeti.de)

ist Senior Consultant bei der SOGETI Deutschland GmbH. Er berät und unterstützt Kunden bei der Einführung von Testautomatisierungslösungen und leitet den Brainpool TAKT, in dem Testautomatisierung in Anwendung und Entwicklung bearbeitet wird.

Dieser Artikel beschreibt eine komplett automatisierte Testsuite, in der ein Testprozessor in Zusammenarbeit mit einem generischen Framework und weiteren Hilfsprogrammen eine flexible Testausführung ermöglicht. Als Testwerkzeug wird „QuickTest Professional“ verwendet, die Testfälle sind im „HP Quality Center“ (beides Programme von HP) abgelegt. Diese Testsuite ist bei einem international tätigen Telekommunikationsunternehmen erfolgreich im Regressionstest im Einsatz. Eine gleichartig aufgebaute Testsuite verspricht auch in weiteren SAP-Projekten die flexible und schnelle Durchführung von Regressionstests.

Grundlagen der Automatisierungslösung

Die erfolgreiche Automatisierung verschiedenster Programme wird oft als sehr einfach dargestellt. Glaubt man den Werbebroschüren der Hersteller diverser Testautomatisierungswerkzeuge, reicht es aus, einfach einen Test aufzuzeichnen, um ihn später in der gleichen Weise wieder abspielen zu können. Schaut man sich aber die Realität an, ist es nicht so einfach. Viele Testdaten können nur einmal verwendet

werden, Programmteile reagieren zu verschiedenen Zeiten unterschiedlich und von den Business-Experten im Unternehmen werden die Prozesse manchmal schneller in Teilbereichen geändert, als man darüber informiert wird.

Gefordert ist also eine Lösung, die schnell und flexibel an wechselnde Testanforderungen angepasst werden kann, die nicht durch Abweichungen im Ablauf aus dem Takt gerät und die möglichst automatisch neue Testdaten erzeugt. Im

Folgenden wird eine solche Testautomatisierungslösung in Form einer Testsuite vorgestellt. Die Testsuite beruht in ihrer Ausführung auf dem bei uns entwickelten TAKT-Ansatz (*Testen, Automatisierung, Kenntnisse und Tools*).

TAKT (siehe Abb. 1) geht davon aus, dass ein einfacher Ansatz der Testautomatisierung (Record & Playback) nicht geeignet ist, um eine durchgehende, erfolgreiche Testautomatisierung über einen langen Zeitraum zu gewährleisten. Vielmehr sind hierfür fundiertes Wissen im strukturierten Testen, Vorgehensweisen im Aufbau einer Automatisierung (siehe auch Abb. 4), Kenntnisse zum Ablauf eines Automatisierungsprojekts und profundes Werkzeugwissen notwendig. Der Bereich „Tools“ in Abbildung 1 ist bewusst abgesetzt. Einerseits muss ein Automatisierungswerkzeug verwendet werden, andererseits ist nicht festgelegt, welches. Jedes hinreichend programmierbare Werkzeug kann – je nach den spezifischen Anforderungen – verwendet werden. Diese Auswahl ist Teil des Ablaufs eines Automatisierungsprojekts, hierfür existiert ein die verschiedenen Planungs- und Ausführungsphasen beschreibendes Phasenmodell.

Um erfolgreich automatisiert testen zu können, benötigt man eine intelligente Kombination aus aufgezeichneten Anteilen mit einer datengetriebenen Testausführung, einem generischen Framework und einem alle diese Bestandteile zusammenhaltenden Testprozessor (vgl. [Bro01]).

Eine flexible Testsuite besteht insgesamt aus vier Teilen:

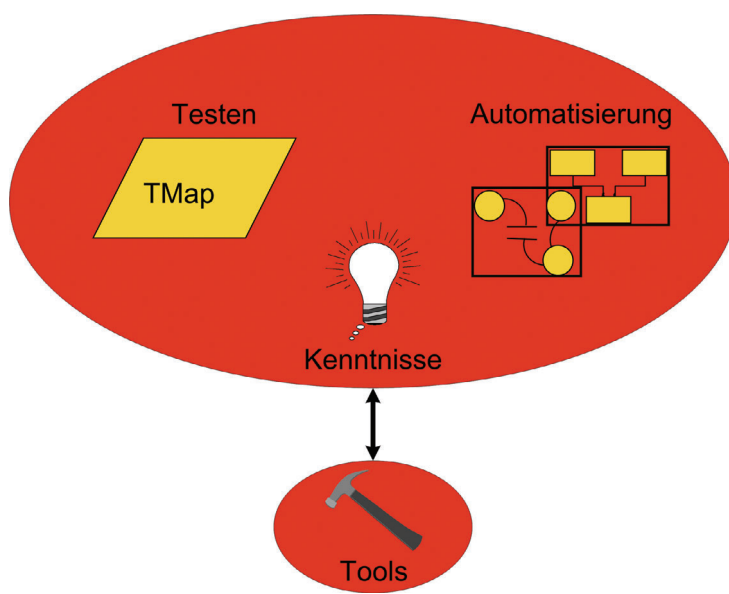


Abb. 1: Der Automatisierungsansatz TAKT.

- verschiedenen Eingabedateien für Test-szenarien und Testskripte
- einem Testprozessor für die Ablaufsteuerung
- einem Framework mit generischen Funktionen für verschiedene Aktionen
- diversen Ausgabedateien für Fortschritts- und Fehlerberichte

Eingabedateien können hierbei generell in vielen Formaten vorliegen oder sogar Datenbanken sein. Was hier möglich ist, hängt vom im konkreten Fall verwendeten Testwerkzeug ab. In dem in diesem Artikel beschriebenen Projekt werden Excel-Dateien verwendet.

Ähnliches gilt für die Ausgabedateien – auch hier sind verschiedene Formate vorstellbar. Um Ergebnisse einfach darzustellen, wird hier oft, wie auch in diesem Projekt, auf ein HTML-Format zurückgegriffen. Die Ergebnisse können dann in einem beliebigen Web-Browser dargestellt werden.

Der Testprozessor zur Steuerung des Testablaufs ist der in **Abbildung 4** rot dargestellte Teil und zeigt einen Aufbau in mehreren Ebenen, um verschiedene Aufgaben zu unterschiedlichen Zeiten ausführen zu können. Dieser Testprozessor wird – abhängig vom verwendeten Testwerkzeug – in der internen Programmiersprache erstellt, ist aber vom Aufbau her für die verschiedenen Implementierungen gleich. Somit ist gewährleistet, dass der Aufbau der Testautomatisierung in verschiedenen Testwerkzeugen zu einem gleichartigen Ablauf führt. Die Implementierung ist bereits in vielen Testwerkzeugen erfolgt und stellt eine Lösung aus der Praxis für die Praxis dar.

Das Framework als Teil der Testsuite stellt generische Funktionen zur Verfügung, die in den einzelnen Testaktionen verwendet werden. Durch die Verwendung von Framework-Funktionen anstelle von eingebauten Funktionen der Testwerkzeuge ist es möglich, sowohl eine über mehrere Umgebungen konsistente Namensgebung zu gestalten, als auch parallel zu der eigentlichen Funktion weitere Aktionen auszuführen, die nur einmal definiert werden müssen (z. B. Ausgabe in das Log, direkte Überprüfung von Eingaben usw.).

Das Projekt

Eine komplette Implementierung einer Testsuite wurde im Rahmen eines sehr

komplexen SAP-Einführungsprojekts verwendet. Für ein international tätiges Telekommunikationsunternehmen sollte im Rahmen der mehrjährigen Einführung von SAP im Bestellprozess verschiedener Ländergesellschaften der Regressionstest automatisiert werden. Die Automatisierung sollte an denjenigen Teilen erfolgen, die nicht andauernden Veränderungen unterworfen sind und die eine Automatisierung zulassen. Eine Automatisierung von sich häufig ändernden Testteilen wird durch den TAKT-Ansatz verworfen und ist auch nicht sinnvoll, da neue Tests initial manuell getestet werden. Die geplanten Regressionstests sind als Geschäftsprozesstests aufgebaut.

In einem Projekt dieser Größenordnung ist es essenziell, eine flexible, erweiterbare, skalierbare, in den Beschreibungen und Prozessen eindeutige und auch vom Funktional-Tester ohne Automatisierungserfahrung verwendbare Testsuite zu haben, um über den gesamten Testverlauf von mehreren Jahren von der Automatisierung profitieren zu können.

In dem Projekt waren die folgenden Rahmenbedingungen vorgegeben:

- „QuickTest Professional“ als Automatisierungswerkzeug
- „HP Quality Center“ als Testmanagement-Lösung

Daher wurde eine Implementierung für QuickTest Professional der generischen Testsuite mit folgenden Eigenschaften gewählt, wobei einzelne Teilkomponenten projektspezifisch erstellt wurden:

- Erstellung der Testszenarien und Testskripte im HP Quality Center mit automatischem Export dieser Daten in Excel-Dateien (internes Format der Testsuite)
- Lagerung aller Dateien (Funktionsbibliotheken, Objekt-Repositories, Analysedaten usw.) im HP Quality Center
- Erstellung von transaktionsbasierten Objektbibliotheken (eine separate Objektbibliothek pro Transaktion) und Nutzung nur der gerade zur Laufzeit benötigten Objektbibliothek
- Verwendung von spezialisierten Framework-Funktionen für SAP (z. B. zum Umgang mit Tabellen)
- Nutzung allgemein gültiger Testaktionen (z. B. Überprüfung des Titels einer

Seite) und spezialisierter Testaktionen (z. B. Überprüfung einer konkreten Tabelle)

- Zusammenfassung von logischen Einheiten zu Business-Objekten (für jede Bildschirmseite einer Transaktion ein Business-Objekt, bei Bildschirmseiten mit Reitern und sehr vielen Objekten können einzelne Reiter Business-Objekte werden)
- Benutzung einer eindeutigen Beschreibungssprache für Objekte auf der SAP-Oberfläche
- Automatische Erzeugung der Dokumentation aus den erstellten bzw. vorhandenen Funktionsbibliotheken

Testbeschreibung mit HP Quality Center

Als führendes System für den gesamten Test in diesem Projekt wurde das HP Quality Center verwendet. Alle Testszenarien und Testskripte – sowohl des manuellen als auch des automatisierten Tests – mussten hier erstellt und gepflegt werden. Für eine effiziente Automatisierung wurde ein Satz von Werkzeugen und Formatvorgaben verwendet, die unter anderem eine auch im manuellen Test benutzte eindeutige Testbeschreibung durch entsprechende Namenskonventionen ermöglichen.

Testbeschreibungen

Für eine erfolgreiche Testautomatisierung ist eine deutlich genauere und ausführlichere Beschreibung der durchzuführenden Programmfunktionen notwendig als beim manuellen Testen. Im Gegensatz zu einem menschlichen Tester, der – basierend auf seiner Erfahrung – viele Schritte in einer Kurzbeschreibung versteht, muss beim automatisierten Test jeder einzelne Zwischenschritt genau vorgegeben werden. Aus diesem Grund wurde das zu automatisierende SAP-System zuerst in logische Einheiten – so genannte *Business-Objekte* – unterteilt. Jedes Business-Objekt wird hierbei nach pragmatischen Gesichtspunkten erstellt. Da SAP transaktionsbasiert arbeitet, ist in der Testsuite jedes Business-Objekt maximal so groß wie eine Transaktion. Viele Transaktionen haben mehrere Bildschirmseiten – hier wird typischerweise aus jeder Bildschirmseite ein Business-Objekt erstellt. In manchen Fällen (mehrere Reiter auf einer Bildschirmseite und dadurch sehr viele Objekte) wurde sogar noch weiter herunter gebrochen.



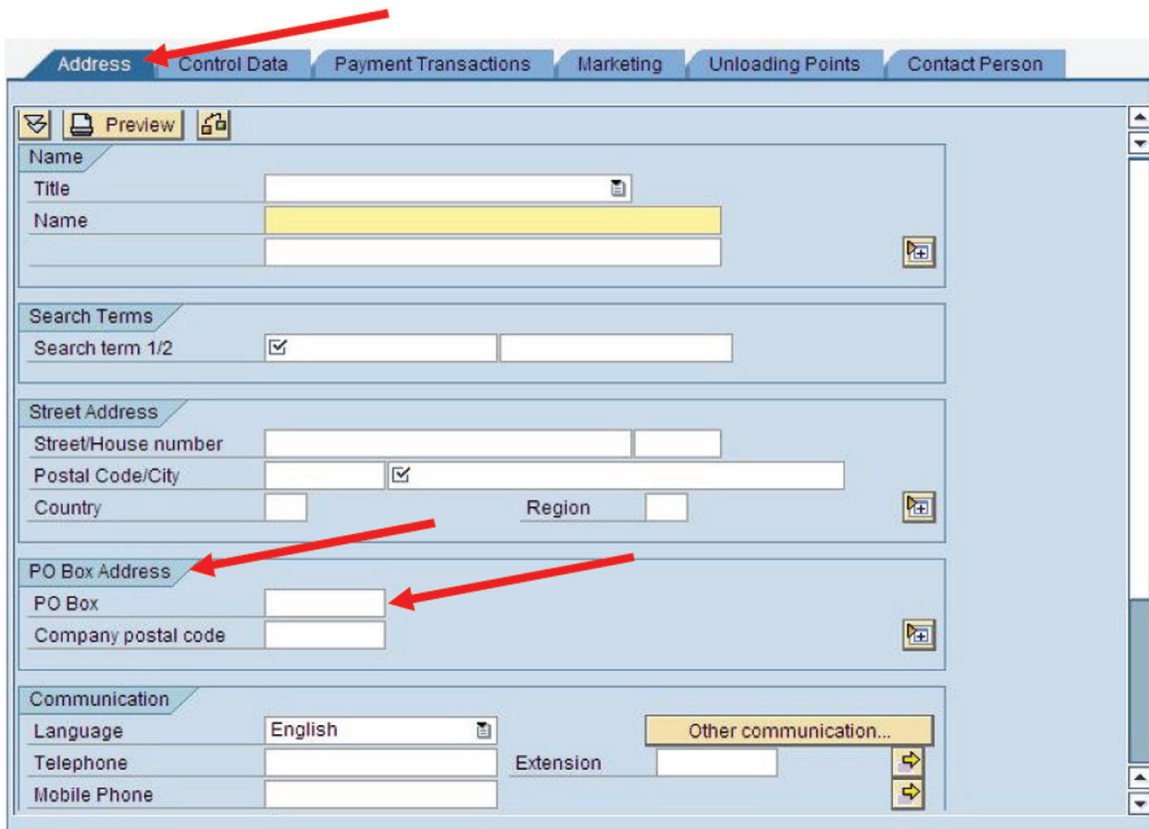


Abb. 2: Beschreibung eines Bildschirmobjekts.

Durch die Komplexität von SAP ergeben sich so bereits mehrere hundert Business-Objekte. Diese Zahl lässt sich jedoch aufgrund des verwendeten Namensschemas gut verwalten.

Alle Business-Objekte bestehen aus einzelnen zu bedienenden Objekten – und auch hier wird anhand einer Namenskonvention eine eindeutige Bezeichnung der auszuführenden Aktion verwendet. Zunächst wird definiert, was getan werden soll (z. B. insert für das Einfügen von Daten), dann wird der Objekttyp benannt (im Beispiel in **Abbildung 2** ein Edit-Feld) und schließlich wird ein Pfad zum Objekt beschrieben (das Beispiel führt zum Pfad Address_POBoxAddress_POBox). Dabei werden alle sichtbaren Gliederungen, wie Tabs oder grafische Boxen, mit einbezogen. Gliederungen, die nicht sichtbar sind und sich nur in der Programmstruktur niederschlagen, werden nicht berücksichtigt, da der Tester, der diese Aktion beschreibt, nur SAP selbst sieht, nicht aber die interne Struktur. Die gesamte Aktion würde also in

diesem Beispiel insert_Edit_Address_POBoxAddress_POBox heißen und in Verbindung mit dem Business-Objekt eine eindeutige Zuordnung ermöglichen.

Diese Beschreibung der auszuführenden Aktion ist so genau und eindeutig, dass auch die manuelle Testdurchführung von der Präzision der Testbeschreibung profitiert. Dadurch ist es leicht möglich, Tester, die diesen Teil in SAP noch nicht gesehen bzw. getestet haben, für einen effektiven Test einzusetzen.

In den Testskripten können eigene Variablen erzeugt und auch zu einem späteren Zeitpunkt wiederverwendet werden, sodass die erstellten Daten im Rahmen eines Szenarios genutzt werden können, ohne diese hart kodieren zu müssen. Das ist insbesondere bei Tests in wechselnden Umgebungen von Vorteil.

Da bei SAP häufig Popup-Dialoge verwendet werden, die im Laufe einer Session nur einmal gezeigt werden und auch bei einer erneuten Verwendung der entspre-

chenden Transaktion nicht wieder auftauchen, sind hierfür Kontrollstrukturen wie If-Abfragen (**siehe Abb. 3**) oder Schleifen in den Testskripten verwendbar.

Die verwendeten Business-Objekte werden zusätzlich im HP Quality Center als Vorlagen sortiert nach Transaktionen gespeichert, um bei späteren Testskripten von den erprobten Abläufen in Zeit und Genauigkeit profitieren zu können. Bei komplizierten Abläufen und Zusammenhängen ist die Verwendung der Vorlagen in zusätzlichen Diagrammen zum Ablauf der Transaktion verdeutlicht; diese sind als Anhänge im Quality Center hinterlegt.

Testexport

Der verwendete Testprozessor nutzt als Eingabe Dateien in einem internen Format auf Excel-Basis. Diese Dateien werden durch verschiedene Werkzeuge direkt aus dem Eintrag im HP Quality Center erzeugt. Parallel dazu werden auch gleich statisti-

| Step Name | Description | Input Data |
|------------------------------|---|---|
| if | | <ClassType> = "<true>" |
| SAP_MM01_ClassType | This popup only appears | check_TitlePopup = "Class Type" activatecell_Table_ClassType = "Class type description Material Class <search> Class type description Material Class <activate>" |
| else | | |
| SAP_MM01_CreateMaterial_ZT01 | Insert the data for creation of an ZT02 | select_Button_Classification_Object_OtherClassType = "<true>" |
| SAP_MM01_ClassType | This popup only appears when | check_TitlePopup = "Class Type" activatecell_Table_ClassType = "Class type description Material Class <search> Class type description Material Class <activate>" |
| end if | | |
| SAP_MM01_CreateMaterial_ZT02 | Insert the data for | select_Button_Classification_NextScreen = "<true>" |

Abb. 3: Testbeschreibung im Quality Center.

sche Daten errechnet, die ein laufendes Reporting der Fortschritte in der Automatisierung ermöglichen. Hierzu wird die Statusführung des HP Quality Centers genutzt – die Status werden auf andere

Status abgebildet, die vom Testmanagement zur Berichterstattung vorgesehen sind. Das hat den Vorteil, dass jeder Projektmitarbeiter ohne weitere Nachfragen über den aktuellen Fortschritt des

entsprechenden Objekts informiert ist und dieses, falls es bereits automatisiert ist, zur Erzeugung neuer Testdaten für die weitere Analyse benutzen kann.

Ein Export der im HP Quality Center gespeicherten Dateien (z. B. Funktionsbibliotheken) ist aus einem weiteren Werkzeug ebenfalls möglich, sodass aus diesen Dateien mit Hilfe der bei der Programmierung in einem speziellen Format angelegten Kommentare eine komplette, konsistente und immer aktuelle Dokumentation der Business-Objekte und der jeweils möglichen Aktionen erstellt wird. Teil dieser Dokumentation sind natürlich auch alle Framework-Funktionen.

Die Bestandteile der Testsuite

Im Folgenden werden die einzelnen funktionalen Bestandteile der Testsuite näher beschrieben, um die Interaktionen zu verdeutlichen. Aufgrund von Einschränkungen durch den Auftraggeber des Projekts ist es nicht möglich, direkte Bestandteile des programmierten Testprozessors oder reale Prozesse aus dem Test abzubilden.

Testprozessor

Der Testprozessor stellt die zentrale Einheit der Testsuite dar und ist als eine mehrschichtige Funktionsbibliothek aufgebaut. Die einzelnen in **Abbildung 4** dargestellten Schichten haben hierbei unterschiedliche Aufgaben, wie z. B. Szenario-Steuerung, Skriptsteuerung und Ausführung der Business-Objekte, und implementieren jeweils unterschiedliche Möglichkeiten der Fehlerverarbeitung. Hierdurch ist es schnell und einfach möglich, auf Fehler von unterschiedlicher Schwere und Ursache flexibel und angemessen zu reagieren – das geht von der Wiederholung der Aktion über einen Neustart des Testskripts nach einem Neustart der Anwendung bis hin zum Gesamtabbruch.

Durch einzelne, den in der Abbildung erkennbaren Ebenen parallele Routinen werden die Ein- und Ausgabedateien verarbeitet. Hierdurch wird eine große Flexibilität in den Skripten ermöglicht, um die durch das SAP-System vorgegebene Variabilität einfach abbilden zu können. Dies wird durch eine integrierte Ablaufsteuerung mit If-Abfragen und For-Schleife bis zu einem dedizierten Abbruchkriterium erreicht.

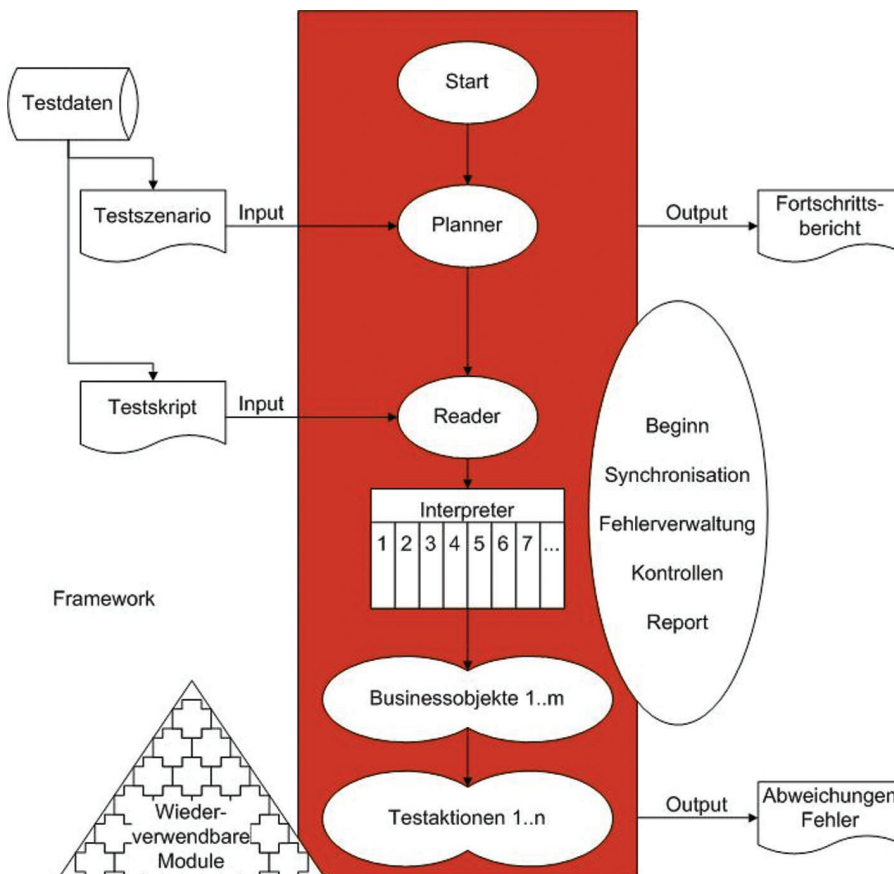


Abb. 4: Aufbau der Testsuite mit Testprozessor.



Dateien

Die Eingabedateien sind Excel-Dateien in einem internen Format. Dieses Format ist eher unwichtig, da diese Dateien automatisch erzeugt werden. Allerdings handelt es sich um ein im Klartext lesbares Format, sodass bei kleineren Skriptfehlern Änderungen sehr einfach direkt durchgeführt werden können.

Die Ausgabe erfolgt zum einen in einer hierarchisch sortierten Log-Datei im HTML-Format (somit in jedem beliebigen Web-Browser lesbar), zum anderen werden Daten des Testlaufs zur Rückführung in das HP Quality Center in einem internen Excel-Format gespeichert. Hierdurch kann im Anschluss an den Testlauf das Ergebnis in das HP Quality Center importiert werden. Der manuelle Aufwand und mögliche Übertragungsfehler werden minimiert, da nur noch Abweichungen separat betrachtet werden müssen. Nur hier ist es dann notwendig, den Testablauf nach Erfassung aller notwendigen Informationen manuell zu vervollständigen und einen Fehlerbericht im HP Quality Center zu erstellen.

Business-Objekte

Die einzelnen Business-Objekte implementieren jeweils eine Routine, um die zum Business-Objekt gehörenden Testaktionen und weitere allgemeingültige Aktionen auszuführen. In der Logik des Testprozessors sind sie direkt oberhalb der Testaktionen angeordnet. Das ist auch die Stelle, an der möglicherweise gleichlautende Aktionen unterschiedlicher Business-Objekte unterschieden werden – so ist die Eingabe eines Wertes in SAP beispielsweise an vielen Stellen möglich. Für die Business-Objekte existiert eine verbindliche Vorlage, um einen gleichartigen Ablauf zu gewährleisten. Da mögliche Testaktionen separat implementiert werden und das Business-Objekt eine beliebige Abfolge der möglichen Testaktionen erlaubt, wird hier eine maximal mögliche Flexibilität in der Abfolge der zum Business-Objekt gehörenden Testaktionen gewährleistet.

Testaktionen

Die Testaktionen sind im Allgemeinen als Einzeiler aufgebaut. In der jeweiligen Testaktion wird mit den konkreten Daten des betroffenen Objekts eine Funktion im Framework zum Ausführen der Aktion aufgerufen.

| | Manueller Test | Automatisierter Test |
|--------------------|----------------|----------------------|
| Testfälle (Anzahl) | 1.200 | 120 |
| Tester (Anzahl) | 16 | 1 |
| Testzeitraum (h) | 80 | 3 |
| Testfälle (1/h) | ~1 | 40 |

Tabella 1: Manueller versus automatisierter Test.

Teilweise werden durch die SAP-eigene Programmierung SAP-Objekte in anderem Zusammenhang genutzt als dem, in dem das ursprüngliche Objekt erstellt wurde. So gibt es beispielsweise Edit-Felder, die als Label-Felder verwendet werden – sie sind also nicht änderbar und sehen grafisch wie Label-Felder aus. Hier werden die Testaktionen mit diesen Objekten so benannt, wie die Objekte genutzt werden, und nicht so, wie die Objekte programmiert wurden. Im Beispiel aus **Abbildung 2** würde also eine Überprüfung des Textes vor dem PO-Box-Objekt als check_Label_ABC und nicht als check_Edit_ABC benannt werden, da das programmierte Edit-Feld als Label genutzt wird und auch so aussieht.

Framework

In Framework-Bibliotheken werden für die einzelnen Objekttypen generische Funktionen zur Verfügung gestellt, die das jeweilige Objekt steuern können. Das kann sehr einfach sein, denn die Änderung einer Checkbox erfordert nur einen einzigen Befehl. Aber auch sehr komplexe Strukturen können zur Steuerung notwendig sein, denn die Steuerung einer Tabelle, insbesondere einer aus Textelementen erstellten Tabelle, bedingt viele Zeilen Programmcode. Ein großer Vorteil bei der Verwendung von Framework-Funktionen liegt darin, dass weitergehende Aktionen bei der Bearbeitung eines Objekttyps nur einmal programmiert werden müssen und bei einer notwendigen Änderung zentral geändert werden können. Beispiele hierfür sind ein automatisches Eintragen der durchgeführten Aktion im Log oder auch eine sofort durchgeführte Überprüfung des Erfolgs einer Aktion.

Projektergebnisse

Die Anwendung der Testautomatisierungslösung im Rahmen des SAP-Projekts

bedingte einen höheren Personaleinsatz als zuerst erhofft. Der Grund hierfür waren nicht die notwendigen Anpassungen in der Testsuite – diese Aufgaben waren durch wenige Personen in kurzer Zeit durchzuführen. Hingegen nahm die Aufzeichnung der Objektbibliotheken viel Zeit in Anspruch, denn ein Objekt kann nur während der manuellen Ausführung eines Tests aufgezeichnet werden (SAP ist sehr restriktiv im Workflow, wenn Pflichtfelder nicht ausgefüllt sind). Um diese Aufzeichnung möglichst effektiv zu gestalten, wurde eine Person nur dafür eingesetzt, die Testscenarien manuell durchzuführen, dabei die Objekte aufzuzeichnen und die Namen der Objekte an die Namenskonvention anzupassen. Hierzu ist es natürlich notwendig, dass die Testfall-Beschreibungen hinreichend genau und aktuell sind.

In diesem speziellen Projekt wurden die Testfälle durch eine weitere externe Firma erstellt, die auch für das *Customizing* der SAP-Umgebung verantwortlich zeichnete. Da Menschen dazu neigen, Testfälle nur so weit zu beschreiben, wie sie den eigenen Kenntnissen und Fähigkeiten entsprechen, entstammten diese häufig einem anderen Erfahrungshorizont. Das führte speziell in diesem Projekt dazu, dass ein Großteil der Arbeit am Regressionstest in die Analyse der bestehenden Testfälle und deren genauer Beschreibung im HP Quality Center in der Beschreibungssprache für die Testsuite bestand. Eine an den Erfahrungshorizont der Tester angepasste Testfall-Beschreibung hätte viel Zeit und Geld sparen können. Um hier einen möglichst schnellen und effektiven Ablauf zu erreichen, wurden zwei Teilteams gebildet: eines zur Analyse der Testfälle und zur Erstellung der Testscenarien im Quality Center, ein zweites zur technischen Umsetzung der Automatisierung. Wegen der Qualität der gelieferten Testscenarien und -fälle war das

Analyseteam größer als das Team zur technischen Umsetzung. Da jedoch ein entsprechender Wissensstand vorhanden war, konnten Ressourcen je nach Notwendigkeit zwischen den Teilteams verschoben werden.

Nachdem die einzelnen Testskripte und -szenarien erstellt waren, haben wir in einem ersten Schritt die Objekt-Bibliotheken aufgezeichnet und die noch nicht fertigen Testaktionen der Business-Objekte erstellt und getestet.

Hier zeigte sich dann schnell, dass die Übernahme der Testszenarien in eine weitere Testumgebung leicht möglich war und nach kurzer Zeit (ca. drei Stunden) ein effektiver Test ausgeführt werden konnte. In **Tabelle 1** sind Vergleichszahlen für einen in einem frühen Ausbaustadium durchgeführten Regressionstest aufgeführt. Die Zahlen zeigen eindrucksvoll, dass ein automatisierter Test eine deutliche Beschleunigung der Testausführung bei einer gleichzeitig höheren Qualität in der Ausführung zur Folge hat, da mehr Überprüfungen pro-

grammiert werden als im ursprünglichen manuellen Test vorgesehen waren. Die Erstellung der Testszenarien und -fälle ist hierbei nicht berücksichtigt – es handelt sich ausschließlich um Zahlen aus der Testausführung.

Bei weiteren entsprechend komplexeren Testszenarien ist mit einem ähnlich hohen Einsparpotenzial zu rechnen, denn neben der schnellen Durchführung und effektiveren Überprüfung entstehen auch Synergieeffekte durch eine Wiederverwendung von Teilen schon erstellter Testszenarien.

Insgesamt ist die Automatisierung auch im SAP-Umfeld eine effektive Möglichkeit, um die Testqualität zu steigern, die Durchlaufzeiten zu reduzieren und einen ausführlichen Regressionstest bei restriktiver Zeitplanung durchzuführen. Dazu ist es unbedingt notwendig, einen strukturierten Testansatz zu nutzen und die Automatisierung so flexibel zu erstellen, wie es nur ein Testansatz wie TAKT kann.

Wegen die Komplexität des SAP-Umfeldes und der sehr kreativen Ver-

wendung verschiedenster Objekte ist es erforderlich, erfahrene Testautomatisierungsexperten einzusetzen, da häufig für spezielle Fragestellungen und Sonderobjekte entsprechende Funktionen erstellt und dem Framework hinzugefügt werden müssen. Die beschriebene Testsuite eignet sich wegen der großen Flexibilität des Ansatzes nicht nur zur Automatisierung von Geschäftsprozess-Tests. Vielmehr sind bei der Anwendung unterschiedlicher Spezifikationstechniken auch Tests möglich, die die Einzelheiten der verschiedenen Transaktionen stärker testen. Die Automatisierung eines Tests unter Verwendung einer derart leistungsfähigen Testsuite macht darüber hinaus Spaß und motiviert die Mitarbeiter. ■

Literatur

[Bro01] B. Broekman, C. Hoos, M. Paap, Automatisierung van de testuitvoering, Academic Service, 2001