Testautomation in verteilten Entwicklungs- und Ausführungsumgebungen

Bernhard Moritz
CC GmbH, Wiesbaden
Bernhard.Moritz@caseconsult.com

Der Herstellungsprozeß für Software unterliegt schnellen Veränderungen und ist geprägt durch die Entwicklung neuer Methoden und Werkzeuge.

Seit etwa 10 Jahren befassen sich auch die Anwendungsentwickler großer Wirtschaftsunternehmen ernsthaft mit den Möglichkeiten, verteilte Anwendungen zu entwickeln.

Die Anwendungsentwicklung fokussierte sich bis dahin auf die Großrechnerwelt, für die "Frameworks" verfügbar waren, die tatsächlich auch von den beteiligten Entwicklern und Entscheidern verstanden wurden.

Die Neugestaltung der Anwendungsarchitektur orientierte sich häufig an den neuen technischen Möglichkeiten, die mit den neuen Programmiersprachen und Systemen einhergingen. Viele falsche Wege sind beschritten und wieder verlassen worden.

Kennzeichnend für alle Unternehmen, die sich für individuelle Software-Lösungen entschieden haben, ist fast immer, daß die Neuorientierung der Anwendungsentwicklung nicht auf der grünen Wiese beginnen durfte, sondern eine Integration der vorhandenen Systeme, mindestens aber deren schrittweise Ablösung vorsah.

Zusammenfassend läßt sich feststellen, daß Anwendungssoftware komplexer geworden ist und insgesamt gesehen mehr Funktionalität bietet, und sei es nur an der Benutzeroberfläche. Komplexer geworden ist ebenfalls der Herrstellungsprozeß für die Anwendungssysteme.

Der Herstellungsprozeß von Software wird typischerweise von Vorgehensmodellen begleitet, die für die beteiligten Menschen die Basis einer produktiven Zusammenarbeit beschreiben. Meist werden diese jedoch erst entwickelt, wenn bereits Erfahrungen mit den neuen Techniken vorliegen.

Software-Prozeßautomation befaßt sich insbesondere mit der Aufgabe, die vielen verschiedenen genutzten Werkzeugfunktionen zu einem sinnvollen Ganzen zu verbinden, die Übergänge von Werkzeug zu Werkzeug zu schaffen bzw. Klüfte zwischen zugrundeliegenden Modellen zu schließen und so die Arbeit der Entwickler (mittels Software) zu unterstützen.

Wenn der Herstellungsprozeß starken Veränderungen unterliegt, kommt der Software-Prozeßautomation zusätzlich die Aufgabe zu, den Lernprozeß der Entwickler zu begleiten, jederzeit eine homogene Sicht für Entwickler auf eine eigentlich heterogene Entwicklungsumgebung zu liefern.

Testprozeß-Automation

Unter Testern und Qualitätssicherern ist es längst ausgemachte Sache, daß Testen und qualitätssichernde Maßnahmen integrale Bestandteile des Software-Entwicklungsprozesses sind. Und doch finden sie sich in der Praxis immer wieder unvermittelt am Ende der Nahrungskette wieder.

De facto werden diejenigen Werkzeuge, die zum Testen benötigt werden, immer erst als letzte verfügbar. Entsprechend unausgereift erscheinen sie dann auch.

Oft widersetzen sich die eingesetzten Basissysteme und Entwicklungswerkzeuge der Testaufgabe in besonderem Maße. Fast so, als sei niemals über Testbarkeit diskutiert worden. Immer wieder erweisen sich die angewandten Methoden bez. der für die Software-Produkte erreichbaren Qualität als nicht so konstruktiv wie versprochen.

Bestimmte Aufgaben des Testens entziehen sich nach wie vor einer Automation. Bislang waren es vor allem die Aufgaben rund um die Testdurchführung, die leicht automatisiert werden konnten. Dazu zählte insbesondere auch die automatisierte Durchführung von Regressionstests.

Angesichts der Forderungen nach Wiederverwendbarkeit, Plattformunabhängigkeit oder Skalierbarkeit sollte angenommen werden dürfen, daß Regressionstests einer neuen Blüte entgegensehen, und daß insbesondere mit deren Automation ein nicht unerheblicher Nutzen verbunden werden kann.

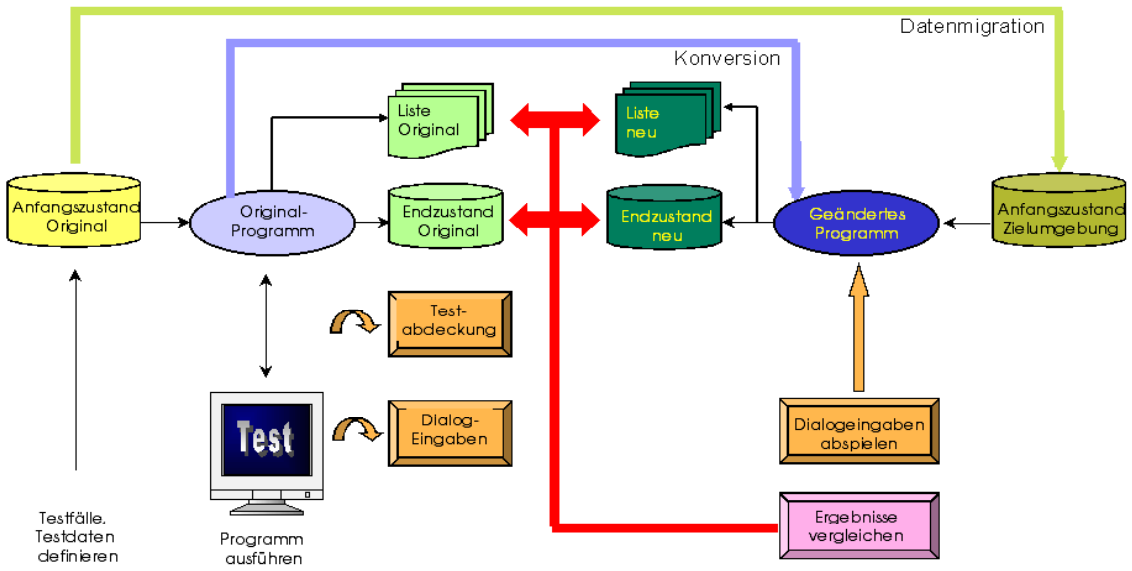
Testfallermittlung und Testdatenbeschaffung müssen für verteilte Anwendungen ebenfalls irgendwie mit der gesteigerten Komplexität umgehen. Dies stellt sich jedoch weitgehend als Herausforderung an die eingesetzten Methoden dar. Für jede einzelne Plattform sind durchaus geeignete Hilfsmittel verfügbar.

Die Testdurchführung sieht sich, sofern in der Konzeption der Software das Testen nicht mitbedacht wurde, jedoch zusätzlich mit der Vielfalt der eingesetzten Hardware und Basissysteme konfrontiert.

Im folgenden wird dargestellt, wie sich der Aufwand zur Automation der Testdurchführung für verteilte Anwendungen gegenüber den früheren hostbasierten Vorgehensweisen verändert hat.

Automatisierte Regressionstests

Das Prinzip von Regressionstests erscheint auf dem Papier als mehr oder weniger banal.



Für den Regressionstest lassen sich einige wenige automatisierbare Aufgaben identifizieren:

- Die Testdatenbestände werden benannt und in einem "Eingabearchiv" abgelegt.
- Die archivierten Testdaten werden in die Ausführungsumgebung gestellt.
- Die zu testende Software wird in der Ausführungsumgebung bereitgestellt
- Die Software wird mit den Testdaten in der Ausführungsumgebung ausgeführt. Dabei werden die Eingaben an der Benutzeroberfläche aufgezeichnet oder durch ein vorbereitetes Skript getrieben.
- Die Resultate der Ausführung werden in einem "Ausgabearchiv" abgelegt.
- Die Resultate der Testausführung werden ausgewertet – durch mehr oder weniger aufwendige Vergleichsoperatoren.

Auch in den "klassischen" Ausführungsumgebungen waren diese Aufgaben technisch nicht trivial, obwohl es sich im wesentlichen um Kopier- oder Ladeoperationen handelte, die zur Automation verwendet wurden.

Die Schwierigkeiten entstehen zunächst aus der Menge der zu bewegenden Datenbestände. Durch die eingesetzten Automaten war sicherzustellen, daß jede einzelne der Operationen erfolgreich durchgeführt worden ist, da andernfalls die Wiederholbarkeit nicht mehr gegeben war.

Weiter war sicherzustellen, daß die benutzte Ausführungsumgebung während des gesamten Zyklus nicht durch Dritte, vielleicht unbeabsichtigt, gestört wurde.

Auswertungsoperatoren waren immer schon von bestimmten Eigenarten der zu testenden Software abhängig. Mit der unbedachten Verwendung von Datum und Uhrzeit, Benutzerkennungen, Sicherheitsmerkmalen, System-Ids etc. konnten automatische Auswertungen immer schon ad absurdum geführt werden.

Unkritisch war dagegen meist das Einstellen der Basissysteme oder der Umgang mit Capture- und Replay-Tools. Es war in der Regel auch durch statische Analysen möglich, die tatsächlich von einem Testobjekt benötigten Ressourcen zu ermitteln.

Insgesamt war die Anzahl der automatisierbaren Operatoren jedoch überschaubar. Es gab eine Hand voll verschiedener Datenhaltungsformen, die berücksichtigt werden mußten. Die verfügbaren Werkzeuge ließen sich in der Regel einfach in die Abläufe integrieren.

Verteilte Entwicklungs- und Ausführungsumgebungen

Moderne Entwicklungs- und Ausführungsumgebungen erfordern ein Vielfaches an Aufwand, um dieselben Aufgaben zu automatisieren:

Als wesentliche Änderungen gegenüber Automaten für "klassische" Systeme ergeben sich:

- Die Anzahl der Datenhaltungsformen hat zugenommen.
- Kopier- und Ladeoperatoren werden üblicherweise von Datentransfers über ein Netzwerk begleitet.
- Jeder der beteiligten Knoten muß in einen Anfangszustand versetzt werden (Starten von Services, Registry-Einstellungen etc.).
- Statische Analysen müssen wesentlich tiefer gehen, da Beziehungen zwischen Softwarekomponenten häufig nur noch aus den Parametern ermittelt werden können (wenn sie nicht in einer Datenbank hinterlegt werden).
- Viele Werkzeugfunktionen lassen sich nicht mehr "Seamless" in die Abläufe integrieren. Eine Parametrisierung von außen ist häufig gar nicht vorgesehen.
- Unter Umständen muß dieselbe Werkzeugfunktionalität für verschiedene Plattformen mehrfach vorgehalten werden.

- Insbesondere bei Systemen mit einer sehr breiten technischen Basis wird die Verwendung von Capture- und Replay-Tools sehr abenteuerlich.
- Häufig sind unterschiedliche Capture- und

Replay-Tools für die verschiedenen Zugänge, die die Software Benutzern heute anbietet, vorzusehen.

- Die Synchronisation und Kontrolle der Abfolgen der auszuführenden Operatoren ist schwieriger zu gewährleisten und zu dokumentieren.

Insgesamt ist die Aufgabe zur Automation, von den Grundoperationen her betrachtet, wesentlich aufwendiger geworden.

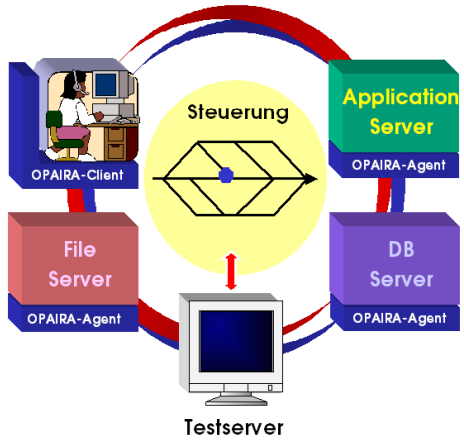
Am meisten wird jedoch ein "central point of control" vermißt, von dem aus alle Operatoren zentral ausgelöst und kontrolliert werden können.

Framework für Automationsaufgaben

Die Anforderungen an die Automation sind mit verteilten Entwicklungs- und Ausführungsumgebungen gewachsen.

Um Lösungen für alle Fragen der Automation bieten zu können, muß sich ein Framework zur Prozeßautomation denselben Techniken bedienen, die der zu bearbeitenden Software zugrunde liegen.

Die Grundidee zur Automation von Testaufgaben besteht also darin, einen gesonderten "Testserver" in die Ausführungsumgebung für die geplanten Tests einzuklinken.



Testserver

Dieser Server ist für die zentrale Steuerung und Kontrolle der Operatoren zuständig. Auf jedem beteiligten Knoten der Umgebung muß Funktionalität für Testaufgaben bereitgestellt werden (Agent).

Alle Informationen, die durch Ausführung von Operatoren auf den verschiedenen Knoten entstehen, werden dem Server zurückgemeldet, so daß der Status einer Testaufgabe zentral aufbereitet und zur Verfügung gestellt werden kann.

Über die Mechanismen lassen sich sämtliche lokal auf den beteiligten Knoten verfügbaren Werkzeugfunktionen in Abläufe integrieren.

Am Arbeitsplatz des Testers wird eine Benutzerschnittstelle zur Verfügung gestellt, um am automatisierten Testprozeß teilzunehmen (Client).

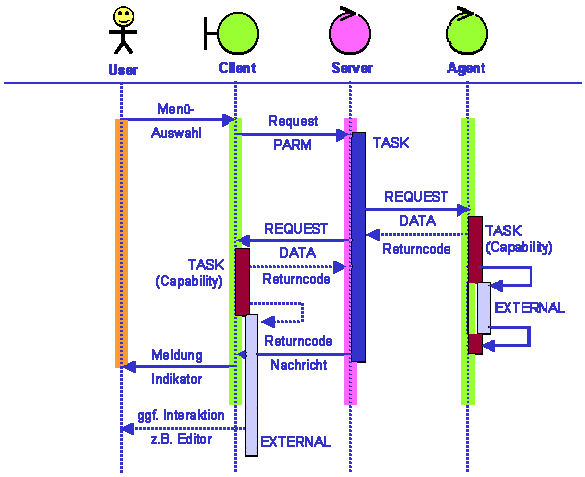
Der Testserver enthält eine Definition des Vorgehens. An den definierten Prozessen können die Benutzer über die speziell für diesen Prozeß generierten Benutzeroberflächen teilnehmen. Die Benutzer "treiben" den Prozeß voran, indem sie die ihnen angebotenen Menüpunkte auswählen.

Dem Benutzer stellt sich ein Testprozeß als eine Abfolge von auszuführenden Aufgaben dar.

Die Prozeßdefinition für ein Testverfahren auf dem Testserver ist vergleichsweise abstrakt und läßt sich leicht wiederverwenden.

Die Implementierung der Prozeßprogramme auf den Knoten der Ausführungsumgebung unterscheidet sich notwendig, je nach verwendeter Plattform.

Insgesamt kann sich aber eine recht aufwendige Koordination der Prozeßprogramme ergeben, z.B.:



Über die verfügbaren Kommunikationsmethoden zwischen Testserver und Agenten wird es ermöglicht, daß ein Tester durch die Auswahl eines Menüpunkts alle Vorbereitungsaufgaben für die Testumgebung auslöst. Dies umfaßt z.B. sowohl das Laden von Daten auf dem Host, das Starten von Services auf zu nutzenden Application-Servern als auch die Registry am Arbeitsplatz. Alle ausgelösten Prozeßprogramme melden den Erfolg oder Mißerfolg an den Testserver zurück, so daß die Bedingungen geprüft werden können, die zum Zeitpunkt des Tests geherrscht haben.

Ausblick

In der Praxis der Testautomation hat sich der dargestellte Rahmen für die Automation von Testprozessen bereits bewährt.

Parallel zur Automation von Testprozessen liegen bereits ähnlich positive Verfahren zum Problem-Tracking und zum Konfigurationsmanagement vor.

Für eine bessere Skalierbarkeit wird jedoch die verfügbare Funktionalität noch erweitert. In der nächsten Ausbaustufe werden Hilfsmittel zur Server-Server-Kommunikation implementiert. Dies dient in erster Linie dem Zweck, auch Projekte mit mehreren Standorten zu unterstützen.

Weiter wird eine bedarfsgetriebene Verteilung von Prozeßprogrammen an die beteiligten Knoten angestrebt. Damit wird in erster Linie eine Vereinfachung der Administration von Tests und den Prozeßprogrammen für dynamisch konfigurierbare Prozesse erreicht.