Wolfram Kusterer

(E-Mail: wolfram.kusterer@parasoft.com)
ist als Geschäftsführer der Parasoft Deutschland GmbH verantwortlich für den Vertrieb von Software und Dienstleistungen in Deutschland, Österreich und der Schweiz. Er ist seit 20 Jahren in der IT und davon mehr als 10 Jahre in verschiedenen Funktionen im Consulting tätig.
Um zu erfahren, wie Parasoft die Virtualisierung des Anwendungsverhaltens durch Parasoft Virtualize ermöglicht, besuchen Sie www.parasoft.com/jsp/products/virtualize_splash.jsp.
Email: info-de@parasoft.com, Tel.: +49 731 880309-0, Fax: +49 731 880309-29

Das Testumgebungsmanagement der Zukunft: Virtualisierung des Anwendungsverhaltens

Im Bereich der Bereitstellung von realistischen Testumgebungen für komplexe Anwendungen schlummern enorme Einsparpotenziale. In diesem Artikel beleuchten wir, wie man durch die Verwendung von Virtual Assets, d. h. die Virtualisierung des Verhaltens von Anwendungen (z. B. Mainframes, CRM-Systeme oder Datenbanken) und externen Komponenten (z. B. Systeme von Geschäftspartnern), die Bereitstellung von neuen Testumgebungen beschleunigen, die Verwaltung der Testumgebung vereinfachen und somit das vorhandene Einsparpotenzial ausschöpfen kann.

Die Komplexität und externen Abhängigkeiten heutiger Systeme haben kritische Folgen für die parallele Entwicklung und für funktionale sowie nicht-funktionale Tests mit signifikanten Auswirkungen auf Produktivität, Qualität und Projektplanung. Zudem ist die Verfügbarkeit von realistischen Testumgebungen sowohl für Tester als auch für Entwickler erschwert. Dies ist bedingt durch

- Fehlende und instabile Komponenten
- Wachsende und sich verändernde Entwicklungsumgebungen
- Nicht verfügbare Systeme und Services von externen Partnern und Dienstleistern
- Zu hohe Systemkomplexität für den Betrieb in Testlaboren (z. B. Mainframes oder große CRM- und ERP-Systeme)
- Interne und externe Ressourcen, die mit anderen Parteien geteilt werden müssen.

Auch wenn Techniken zur Hardware- und Betriebssystem-Virtualisierung eine deutliche Verbesserung in Bezug auf Kosten und Verfügbarkeit von Infrastrukturen bieten, so bestehen dennoch signifikante Defizite in der Unterstützung der Entwicklung und der Tests. Ebenso ist es noch nicht möglich, effektiv eine Virtualisierung für eine hohe Anzahl großer Systeme, wie zum Beispiel Mainframes und ERP-Systeme bereitzustellen. Beispielsweise bedarf die Konfiguration und Pflege, sowohl der Umgebung als auch der benötigten Daten zur Unterstützung von Test- und Entwicklungsarbeiten, immer noch sehr hoher Aufwände. Als Folge dessen stellt die Synchronisierung von komplexen Testumgebungen mit sich ständig weiterentwickelnden agilen Projekten eine schier endlose Aufgabe dar.

Dieser Artikel stellt einen neuen Ansatz zur Virtualisierung des Anwendungsverhaltens vor. Hierbei handelt es sich um eine Möglichkeit, Entwicklern und Testern die Freiheit zu geben, ihre Anwendungen auch in unvollständigen, sich stetig ändernden und nicht ständig verfügbaren Umgebungen auszuführen. Im Gegensatz zur Virtualisierung von ganzen Anwendungen oder Datenbanken, fokussiert sich die Virtualisierung des Anwendungsverhaltens darauf, speziell diejenigen Teile zu virtualisieren, welche Entwickler und Tester für ihren speziellen Anwendungsfall benötigen. Neben der Virtualisierung von Infrastrukturen berücksichtigt dieser Ansatz auch alle Aspekte von zusammengesetzten Anwendungen. Hierzu gehören Dienste, Mainframes, Oberflächen von Webanwendungen und mobilen Endgeräten, ERPs, ESB/JMS, Altsysteme und vieles mehr.

Diese neue Form der Virtualisierung, die sich komplementär zu bestehenden Virtualisierungsstrategien bewegt, führt zu einer drastischen Reduzierung der Konfigurationsaufwände, des Hardware-Overheads und der Pflegeaufwände für Testdaten, die zur Bereitstellung einer realistischen und nachhaltigen Test- und Entwicklungsumgebung notwendig sind.

Virtualisierung reduziert Kosten, erhöht Verfügbarkeit, löst aber nicht alle Probleme

Angesichts verschiedenster Systemanforderungen haben viele Organisationen Hardware- und Betriebssystem-Virtuali-

sierung eingeführt, um ihren Teams ständigen Zugriff auf realistische Entwicklungs- und Testsysteme zu ermöglichen. Die Virtualisierung der Kernelemente, wie dem Betriebssystem, der Konfiguration und der Plattform selbst, haben zu drastischen Verbesserungen für das Management von Test- und Entwicklungsumgebungen geführt. Hierdurch wird eine beachtliche Unabhängigkeit vom Produktivsystem verbunden mit einer gleichzeitigen Reduktion der Infrastrukturkosten und einer höheren Verfügbarkeit spezieller Systemtypen erreicht. In Kombination mit einer Cloud-Infrastruktur bietet die Virtualisierung sogar eine nahezu unbegrenzte Bandbreite zur Skalierung der voneinander abhängigen Systeme. Nichtsdestotrotz bestehen speziell für Entwicklungs- und Testumgebungen weitergehende, noch nicht betrachtete Herausforderungen. Allen voran können mit dieser Strategie bestimmte Elemente mit dieser Strategie nicht virtualisiert werden. Beispielsweise ist eine Hardware- und Betriebssystem-Virtualisierung für große Mainframe-Anwendungen, Third Party-Anwendungen oder große ERP-Systeme oftmals gar nicht möglich.

Selbst wenn dies möglich wäre, so besteht immer noch die Herausforderung, jedes dieser Systeme zu konfigurieren und zusätzlich zur eigentlichen Virtualisierungsinfrastruktur zu verwalten. Die Pflege und Verwaltung der Konfiguration und der Datenintegrität für alle abhängigen Systeme bleibt somit eine signifikante und in vielen Fällen immer noch sehr zeitaufwändige Arbeit. Zudem handelt es sich hierbei um Aufgaben, die teilweise externe Unterstützung benötigen wie zum Beispiel von Administratoren für das Setup von Systemen.

Die Virtualisierung des Anwendungsverhaltens reduziert diesen Overhead für Konfiguration und Verwaltung, indem sie Entwickler und Tester in die Lage versetzt, mit wenig Aufwand die von ihnen benötigten Systemteile zu isolieren und zu virtualisieren, um speziell die von ihnen aktuell betrachteten Transaktionen vollständig durchführen zu können.

Im Gegensatz zur Virtualisierung des gesamten Systems werden nur diejenigen Ausschnitte des Systems virtualisiert, die für die aktuelle Test- oder Entwicklungsaufgabe notwendig sind. Die Einschränkung des zu virtualisierenden Teilsystems anhand der betrachteten Transaktionen reduziert gleichzeitig den Umfang dessen, was es zu virtualisieren gilt.

Was genau bedeutet nun die Virtualisierung des Anwendungsverhaltens?

Die Virtualisierung des Anwendungsverhaltens ist eine fokussierte und effektive Strategie zur Eliminierung von System- und Umgebungsabhängigkeiten, die ein Team beim Test einer heterogenen, komponentenbasierten Anwendung behindern.

Im Gegensatz zum Versuch, alle abhängigen Komponenten wie die gesamte Datenbank, alle externen Anwendungen, etc. vollständig zu virtualisieren, werden nur diejenigen Teilsysteme virtualisiert, die für das aktuell von den Entwicklern und Testern betrachtete Verhalten notwendig sind. Dies kann sich sowohl auf bestimmte Anwendungen, bestimmte Komponenten oder auch auf bestimmte Szenarien beziehen.

Anstatt beispielsweise die komplette Datenbank eines Systems zu virtualisieren und die Testdaten in vollem Umfang verwalten zu müssen, wird einmalig beobachtet, wie die Anwendung innerhalb des untersuchten Testfalles oder einer ganzen Reihe von Testfällen mit der Datenbank interagiert und interagiert. Daraufhin können genau diese notwendigen Teile der Datenbank, beispielsweise SQL-Anfragen an die Datenbank beziehungsweise Ergebnisse die daraufhin zurückgeliefert werden, virtualisiert werden. Diese Systemteile können dann später abgefragt und gemäß unterschiedlicher Test- und Entwicklungsszenarien variiert werden.

Zu Beginn gilt es, die zu virtualisierenden Komponenten zu bestimmen. Dies sind in der Regel verbundene Systeme oder Randsysteme, welche zwar zur Durchführung der Tests notwendig, aber keine Kernkomponenten des zu testenden Systems (System Under Test) sind. Während der Ausführung wird das Verhalten der Randsysteme sowie der relevanten Transaktionen, Nachrichten, Services, etc. und der zugehörigen Daten in einem sogenannten „Virtual Asset“ erfasst. Dies ist in Abbildung 1 dargestellt.

Ein solches Virtual Asset kann dann später durch Parametrisierung seiner bedingten Ausführung, seiner Performanzeigen-

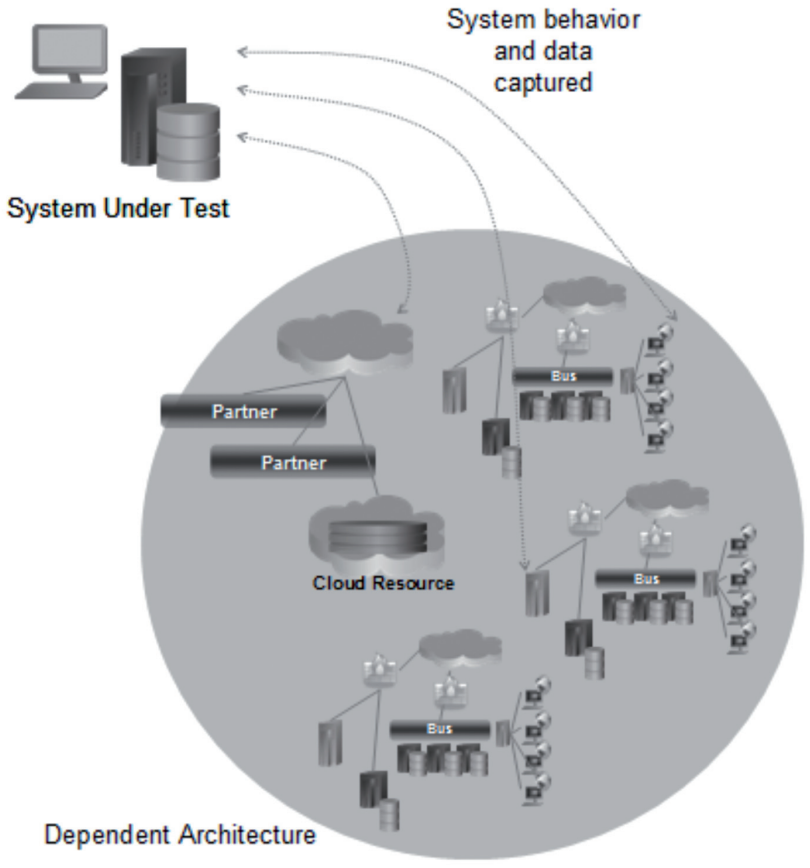

Abb. 1: Aufzeichnung des Verhaltens und der Daten von den zu virtualisierenden Systemkomponenten

schaften und seiner Testdaten konfiguriert werden. Ab diesem Zeitpunkt kann ein solches Virtual Asset das Verhalten eines verbundenen Systems emulieren und das tatsächliche System in der Test- und Entwicklungsumgebung ersetzen.

Darüber hinaus können Testdaten direkt mit diesen Virtual Assets verknüpft werden, was die Notwendigkeit einer zusätzlichen Datenbank und den damit verbundenen Verwaltungs- und Pflegeaufwand eliminiert. Zumal es gerade bei gemeinsam genutzten Datenbanken leicht zu veränderten oder sogar korrupten Testdaten kommen kann.

Diese Form der Virtualisierung des Anwendungsverhaltens ermöglicht es, die Abhängigkeit von Live-Systemen und -Architekturen aufzulösen und trotzdem den benötigten Zugriff auf deren Verhalten bereitzustellen. Dieser extrem fokussierte Ansatz ermöglicht eine deutliche Zeit- und Kostenersparnis bei der Bereitstellung und Verwaltung unterschiedlichster Umgebungen und komplexer Testdaten.

Was umfasst die Virtualisierung des Anwendungsverhaltens?

Die Virtualisierung des Anwendungsverhaltens kann in die folgenden drei Phasen untergliedert werden:

- Erfassung oder Modellierung des realen Verhaltens verbundener Systeme
- Konfiguration des Virtual Assets entsprechend der Anforderungen des betrachteten Szenarios
- Bereitstellung des Virtual Assets für den Zugriff durch berechtigte Teammitglieder und Partner

Phase 1: Erfassung

Das Verhalten des realen Systems wird aufgezeichnet und das Virtual Asset erstellt – mittels Monitoring zur Aufzeichnung realer Systemtransaktionen, durch die Analyse von Transaktionslogs oder durch die Modellierung des Verhaltens durch eine einfach zu bedienende, grafische Benutzerschnittstelle.

Das Ziel dieser Phase ist die Erfassung des Verhaltens und der Performanzeigenschaften der aufgerufenen Anwendung speziell für den Aufruf durch das zu testende System, um dieses Verhalten für spätere Test- und Entwicklungsarbeiten bereitzustellen. Die Erfassung kann auf drei verschiedene Arten geschehen:

- Wenn das reale System verfügbar ist, so kann man das Verhalten auf Basis des Datenverkehrs im System aufgezeichnen. Mittels eines Proxys, der die Kommunikation mit dem realen System beobachtet, können die relevanten Aufrufe und Antworten registriert und in einem Virtual Asset abgebildet werden. Hierbei kann man sowohl einfaches als auch komplexes Verhalten erfassen. Beispielsweise kann die Übertragung von Finanzanlagen an einem Endpunkt des Systems zu Kontoänderungen an einem anderen Endpunkt führen.
- Soll das Verhalten einer Anwendung auf Basis von Transaktionslogs erfasst werden, kann man Virtual Assets durch die Analyse dieser Logs erzeugen. Hierbei handelt es sich um einen eher passiven Ansatz zur Erfassung des Systemverhaltens.
- Handelt es sich bei dem zu untersuchenden System um eine noch nicht abgeschlossene Entwicklung, so ist es nötig, das Verhalten dieses noch nicht existierenden Systems mit der dafür vorgesehenen grafischen Benutzeroberfläche zu modellieren. Durch die umfangreiche Unterstützung von unterschiedlichen Protokollen ist es in kürzester Zeit möglich, Virtual Assets zu entwickeln, die nahezu jedes gewünschte Verhalten emulieren. Beispielsweise ist es möglich, unterschiedlichste Nachrichtenformate, beispielsweise XML oder JSON, und viele proprietäre Formate, wie zum Beispiel aus dem Finanzsektor, dem Gesundheitswesen oder anderen Domänen visuell zu modellieren.

Phase 2: Konfiguration

Die Virtual Assets werden hinsichtlich ihrer Performanzeigenschaften, der Nutzung von Datenquellen und ihres bedingten Antwortverhaltens weiter verfeinert und angepasst.

Nachdem ein Virtual Asset mittels einer der vorgestellten Methoden erzeugt wurde, kann dieses konfiguriert werden, um das von ihm emulierte Verhalten weiter zu verfeinern beziehungsweise zu erweitern. So ist es beispielsweise möglich Qualitätsmetriken anzuwenden um festzulegen, wie sich ein Virtual Asset in Bezug auf seine Performanz (Laufzeit, Latenz und Verzögerung) verhalten soll.

Ebenso können die Testdaten jedes Assets modifiziert oder erweitert werden, um kritische Konstellationen für spezifische Test und Entwicklungsfälle zu konstruieren. So kann man beispielsweise verschiedene Fehlerkonstellationen reproduzieren, die sonst nur sehr schwierig in realen Systemen nachgestellt werden könnten.

Durch das Hinzufügen weiterer Datenquellen und Antwortverhalten können Virtual Assets sowohl ein erwartetes als auch unerwartetes Verhalten reproduzieren, so dass man sowohl positive als auch negative Testfälle durchspielen kann.

Phase 3: Bereitstellung und Test

Die Ausführungsumgebung wird für einen sicheren Zugriff sowohl teamübergreifend als auch für Partner bereitgestellt. Die Virtual Assets können hierdurch effektiv zur Ausführung von Tests genutzt werden.

Sobald ein Virtual Asset erstellt wurde, kann dieses für einen unkomplizierten Zugriff durch verschiedene Teams und sogar durch Geschäftspartner sowohl lokal als auch global zur Verfügung gestellt werden. Letzteres kann entweder durch einen global erreichbaren Server oder auch über eine Cloudumgebung ermöglicht werden. Auf diesem Weg sind die Virtual Assets sowohl in Unit-Tests, als auch in funktionalen und Performanztests einsetzbar. Dank der nativen Unterstützung unterschiedlichster Zugriffsprotokolle ist es möglich, die Virtual Assets sowohl für manuelle als auch für automatisierte Tests mittels Test Suites und Frameworks wie Parasoft Test, HP Quality Center, IBM Rational Quality Management, Oracle ATS und anderen zu nutzen. Ebenso leicht ist eine Skalierung der Virtual Assets möglich, um Performanztests in hoch skalierenden und unter hoher Last stehenden Systemen zu realisieren.

Selbst nach der initialen Bereitstellung dieser Virtual Assets können sie leicht angepasst und wiederverwendet werden, um unterschiedlichste Test- und Entwicklungsszenarien zu unterstützen. Wird beispielsweise im Rahmen eines Testszenarios ein bestimmtes Virtual Asset mit einer spezifischen Konfiguration verwendet, so kann hiervon direkt ein weiteres Virtual Asset inklusive der Konfiguration abgeleitet werden. Dieses Asset kann nun für ein ähnliches Testszenario entsprechend angepasst und direkt wiederverwendet werden.

Wie die Virtualisierung des Anwendungsverhaltens das Testen beschleunigt und Kosten reduziert: 3 typische Anwendungsfälle

Zum Abschluss betrachten wir, wie Organisationen bereits die Virtualisierung des Anwendungsverhaltens erfolgreich einsetzen, um die Herausforderungen des Managements von Entwicklungs- und Testumgebungen in den folgenden drei Kontexten zu bewältigen:

- Umgebungen mit Performanz- und Kapazitätseinschränkungen
- Komplexe und schlecht verfügbare Systeme (Mainframes, große ERP-Systeme, externe Systeme)
- Parallele Entwicklungsarbeit (agile oder andere iterative Prozesse)

Umgebungen mit Performanz- und Kapazitätseinschränkungen

Testumgebungen bieten oftmals keine ausreichende Leistungsfähigkeit für realistische Performanztests. Die Bereitstellung mehrerer virtualisierter Systeme auf einer Hardware erhöht zwar die Verfügbarkeit der Testsysteme, verringert aber oftmals auch die verfügbare Leistung. Auch wenn Tests durch die erhöhte Verfügbarkeit oft überhaupt erst möglich werden, stellen die virtualisierten Systeme in der Regel kein realistisches Systemverhalten zur Verfügung. Dies mindert deutlich den Wert der Testaufwände.

Die Virtualisierung des Anwendungsverhaltens ermöglicht hingegen eine Skalierung realistischer Performanzeigenschaften unabhängig vom emulierten Live-System. Wenn ein Virtual Asset, das die realen Performanzeigenschaften beschreibt, einmal erzeugt wurde, so kann dieses beliebig konfiguriert werden um realistische Performanzeigenschaften des Systems zu simulieren. Hierdurch können Performanztests gegen das Virtual Asset mit realistischen Performanzeigenschaften entsprechend vereinbarter Qualitätszusicherungen durchgeführt werden, anstatt mit Testsystemen, die nur eingeschränkte Performanzeigenschaften bieten, arbeiten zu müssen.

Um die Performanz von Virtual Assets zu definieren, bedarf es lediglich der Anpassung von Timing, Latenz und Verzögerungskonfigurationen. Neben der Simulation von realistischem Anwendungsverhalten ist es auch möglich, direkt Performanzverhalten zu simulieren, die ansonsten kaum zu reproduzieren wären. Beispielsweise wird es möglich, aufgerufene Komponenten mit einer sehr schlechten Performanz, bis hin zu Fällen, in denen gar keine Antwort zurück geliefert wird, zu simulieren. Dadurch kann man testen, wie das System mit solchen Engpässen umgeht. Auch wenn alle für die Tests notwendigen Systeme, welche ein realistisches funktionales Verhalten bieten, bereitgestellt werden können, so ist es dennoch nur selten möglich, verschiedenste Komponenten mit einer Last zu versorgen, die für repräsentative Last- und Stresstests erforderlich ist. Beispielsweise könnte es nötig sein zu testen, wie die eigene Anwendung auf ein extrem hohes Transfervolumen reagiert, um Spitzenbelastungen zu simulieren. Doch wie kann dies vertretbar realisiert werden, wenn die betrachtete Transaktion einen externen Service aufrufen muss, für den in Abhängigkeit vom Transaktionsaufkommen Zugriffskosten entstehen?

Wenn ein Performanztest eine Komponente verwendet, die man nicht unter extrem hohen Lastbedingungen aufrufen kann oder möchte, so ist es durch die Virtualisierung des Anwendungsverhaltens möglich ihr Verhalten unter geringer Last, beispielsweise mit nur einer Benutzertransaktion, zu erfassen und ihr aufgezeichnetes Performanzverhalten nach den eigenen Anforderungen im Virtual Asset anzupassen. Die Lasttests können dann gegen das Virtual Asset anstatt gegen die eigentliche Komponente ausgeführt werden.

Falls die betrachtete Komponente nicht verfügbar ist und ihr Verhalten nicht automatisiert erfasst werden kann, so ist es möglich ein entsprechendes Virtual Asset von Grund auf neu zu erstellen. Hierzu steht eine grafische Benutzeroberfläche zur Modellierung zur Verfügung, mit der das erwartete Verhalten der zu virtualisierenden Komponente spezifiziert werden kann.

Komplexe und schlecht verfügbare Systeme

Bei der Entwicklung und dem Testen großer und komplexer Systeme verschiedener Plattformen kommt es oft vor, dass sich parallel arbeitende Teams begrenzte Ressourcen in einer gemeinsamen Testumgebung teilen müssen. Hinzu kommt, dass die meisten dieser großen Systeme sogar zu komplex sind, um sie in einem Testlabor oder in einer realen Testumgebung nachzubilden, geschweige denn redundant für alle Teammitglieder vorzuhalten. Außerdem müssen sich die testenden Teams und Softwareentwickler möglichst gut miteinander abstimmen, wenn es darum geht Anwendungsfälle zu testen, die den Aufruf kostenpflichtiger, externer Angebote beinhalten, um die entstehenden Kosten niedrig zu halten. Dies hat oft zur Folge, dass Tests verschoben werden müssen und Teams nicht wie gewünscht in voller Breite und Tiefe testen können. Insbesondere in iterativen Entwicklungsprozessen besteht ein erhöhter Bedarf an regelmäßigen und zeitnahen Tests, was die Problematik solcher Verzögerungen und Kosten exponentiell steigen lässt.

Auch wenn Unternehmen in der Lage sind, solche komplexen Systeme zu virtualisieren, so verlangt die Anpassung dieser Systeme für die individuellen Testszenarien der einzelnen Teams sehr aufwändige Konfigurationsarbeit. Wenn diese Hürde erst einmal bewältigt wurde, wartet bereits die nächste: Die Entwicklung und das Management der benötigten Testdaten.

Wenn Teams in einem solchen Kontext jedoch Virtual Assets nutzen, dann genügt es, einen Ausschnitt des Systems lange genug zu beobachten, um das Verhalten derjenigen Komponenten und Transaktionen zu erfassen, die für ihre Arbeit relevant sind. Indem dieses Verhalten in Virtual Assets gespeichert und vorgehalten wird, ist es ihnen nun möglich, vollständige Transaktionen auszuführen wann immer sie wollen (ohne eine bestimmte Zeitplanung), so oft sie wollen (ohne wiederkehrende und teure Zugriffsgebühren) und ohne zusätzliche Konfigurationsaufwände für einzelne Testläufe.

Parallele Entwicklung

Für Teams, welche parallel entwickeln (in agilen oder anderen iterativen Prozessen), kann die dauerhafte Bereitstellung einer realistischen Testumgebung selbst für kleine Anwendungen zu einer großen Herausforderung werden. Viele verschiedene Teammitglieder, wie Entwickler, Tester und sogar Business Analysten, benötigen einen einfachen Zugang zu einer Entwicklungs- und Testumgebung, die stets synchron mit der sich weiterentwickelnden Anwendung ist. Entscheidet sich hier das Team für einen klassischen Virtualisierungsansatz, so muss es nicht nur den anfänglichen Setup-

Aufwand leisten, sondern läuft auch Gefahr, sich in der ständigen Aktualisierung des Systems zu verzetteln, um dieses an den aktuellen Entwicklungsstand anzugleichen. Wird das Team außerdem ständig blockiert, weil es auf den Zugriff auf benötigte Funktionalität wartet, dann verhindert dies jegliche Agilität seiner Arbeit. Virtual Assets reduzieren diese Einschränkungen und die damit verbundenen Wartezeiten, in dem sie Entwicklern und Testern die Möglichkeit geben, das benötigte Verhalten selbst zu emulieren statt darauf warten zu müssen, dass andere die benötigten Randsysteme aktualisieren, konfigurieren und bereitstellen. Selbst wenn die erwarteten Dienste und Komponenten noch nicht implementiert wurden, so kann ihr Verhalten modelliert und den Teammitgliedern zur Verfügung gestellt werden, um die von ihnen benötigten Transaktionen ohne Verzug ausführen zu können. Auch wenn sich das Verhalten bestimmter Funktionen ändert, so können die entsprechenden Virtual Assets leicht angepasst werden, indem das Verhalten entweder neu aufgezeichnet oder die Konfiguration der Virtual Assets angepasst wird. Letzteres kann einfach über eine grafische Oberfläche vorgenommen werden – ohne Programmierkenntnisse oder Skriptaufwand.

Beispielsweise gibt es viele Unternehmen, die mobile Anwendungen entwickeln, welche typischerweise von speziellen Entwicklungsteams erstellt werden. Diese mobilen Anwendungen greifen oftmals auf Kernanwendungen zu, die in der Regel von anderen Teams entwickelt werden. Diese Abhängigkeit kann leicht zu Verzögerungen seitens der Entwicklungsteams für die mobilen Anwendungen führen, da sie auf die Bereitstellung der Kernanwendungen warten müssen. Die Virtualisierung des Verhaltens dieser Kernanwendungen ermöglicht es nun, die Verzögerungen zu eliminieren, auch wenn die entsprechenden Anwendungen noch nicht oder nur teilweise fertig gestellt wurden oder während einer parallelen Entwicklung nur schwer zugänglich sind.

Die wichtigsten Erkenntnisse

Durch die Verwendung von Virtual Assets werden Teams in die Lage versetzt, die Komplexität und Kosten für das Management verschiedener Umgebungen zu reduzieren und gleichzeitig realistische Systeme für den Zugriff im Rahmen ihrer Entwicklungs- und Testarbeit bereitzustellen. Die Virtualisierung des Anwendungsverhaltens ermöglicht:

- Die Infrastrukturkosten zu reduzieren
- Die Bereitstellung und Pflege von Testumgebungen zu verbessern
- Die Testabdeckung zu erhöhen
- Die Anzahl von Fehlern zu reduzieren
- Die Vorhersagbarkeit und Kontrolle der Softwarelaufzeiten zu verbessern
- Die Entwicklungsproduktivität zu steigern
- Die Zugriffsgebühren für externe Systeme zu reduzieren

Dieser Artikel erschien erstmals in der Ausgabe 2011 des OBJEKTspektrum Online Themenspecials Testing.