



TECHNISCHES WHITEPAPER

# Wie Sie das Testumgebungs- und Testdatenchaos in den Griff bekommen



Die Anzahl und Komplexität der Testumgebungen, auf die für die verschiedenen Projekte eines Unternehmens zugegriffen werden muss, gerät zunehmend außer Kontrolle. Die Herausforderung, einen schnellen Zugriff auf die erforderlichen Testumgebungen mit den jeweils notwendigen Konfigurationen zu ermöglichen, lässt sich aus verschiedenen Gründen kaum bewerkstelligen.

Beim Testen von bestimmten Transaktionen, welche einer Vielzahl an Applikationen betreffen ist es unmöglich, die Interaktion mit abhängigen Anwendungen zu vermeiden, da sich diese häufig in der Entwicklung befinden, nicht verfügbar sind oder auf die für die Tests nur schwer zugegriffen werden kann.

In Unternehmen, die iterative oder agile Entwicklungsmethoden anwenden, existieren mehrere Projekte in denen abhängige Systeme mit unterschiedlichen Zeitplänen entwickelt werden. Die rasche Entwicklung dieser Systeme führt in verschiedenen Phasen zu einer erheblichen Komplexität an Staging-Umgebungen.

Auch wenn die Virtualisierung von Servern (Hardware) einige dieser Abhängigkeiten verringern kann, so bleibt doch ein erheblicher Zeit- und Kostenaufwand für die Anschaffung, Einrichtung, Verwaltung und ständige Neukonfiguration der Server für die verschiedenen Testanforderungen bestehen.

Einige abhängige Anwendungen, wie z. B. Anwendungen von Drittanbietern und Mainframes, lassen sich nicht virtualisieren oder bereitstellen. Die Teams können nur in eng begrenzten Zeiträumen auf sie zugreifen und haben nicht ausreichend Zeit oder die Freiheit, sie für die erforderlichen Tests zu konfigurieren.

Testpläne erfordern in der Regel, dass die Tests für eine Reihe verschiedener Szenarien (erwartete Antworten, unerwartete Antworten, Verzögerungen usw.) durchgeführt werden. Die genaue Bewertung der Funktionalität der zu prüfenden Anwendung (AUT, application under test) erfordert in der Regel Tests für unzählige Abwandlungen des Verhaltens abhängiger Komponenten.

Es ist eine gewaltige Aufgabe, den einfachen und gleichzeitigen Zugriff auf die verschiedenen abhängigen Komponenten in einer einzigen Testumgebung zu koordinieren. Das Bereitstellen all der verschiedenen Umgebungen, die in einem Team, einer Abteilung oder auch unternehmensübergreifend benötigt werden - jede dabei mit der erforderlichen Konfiguration - ist oft überwältigend. Und überwiegen die Kosten und der Aufwand für die Erstellung all dieser Testumgebungen den wahrgenommenen Nutzen der Testdurchführung, kommt das tatsächliche Testen zu kurz. Das Ergebnis sind Qualitätseinbußen und Termine, die nicht mehr eingehalten werden können.

## **DAS FEHLENDE STÜCK: SERVICE-VIRTUALISIERUNG**

Die Service-Virtualisierung lindert diese Probleme, indem sie eine Simulation der verschiedenen benötigten „realen Testumgebungen“ ermöglicht. Da Entwickler und Tester von nun an mit einer virtuellen Umgebung arbeiten, werden die Abhängigkeiten zu realen Systemen und deren Verfügbarkeit auf ein Minimum reduziert. Ebenso werden Abhängigkeiten zwischen den Teams beseitigt, da diese unabhängig voneinander auf die virtualisierten Services zugreifen können.

Sie setzt quasi dort an, wo die traditionelle Server-/Hardware-Virtualisierung aufhört. Wenn eine Servervirtualisierung nicht möglich ist, ermöglicht die Service-Virtualisierung die Emulation des Verhaltens von abhängigen Anwendungen. Im Gegensatz zu Stubs und Mocks sind virtuelle Assets einfach zu erstellen, repräsentieren ein breites Spektrum an realistischem Verhalten und sind leicht zu aktualisieren, wenn sich die abhängigen Anwendungen weiterentwickeln.

Selbst wenn eine Servervirtualisierung möglich ist, bedeutet sie im Kontext einer Testumgebung nicht immer den praktischsten Ansatz. In der Regel wird nur ein kleiner Prozentsatz der Funktionalität der abhängigen Anwendung in das Testszenario einbezogen. Allerdings reduziert diese niedrige Auslastung kaum die Kosten oder den Zeitaufwand für das Beschaffen dieser Anwendungen, ihre Konfiguration und die für jede gewünschte Testumgebung erforderlichen Anpassungen.

Da häufig nur der Zugriff auf einen Bruchteil der verfügbaren Funktionen notwendig ist, muss mithilfe der Service-Virtualisierung nicht das vollständige System simuliert werden.

Stattdessen emulieren "virtuelle Assets" die spezifischen Interaktionen der AUT's mit den abhängigen Anwendungen und treten dann in der Testumgebung an die Stelle der tatsächlichen Komponenten. Das Ergebnis sind nicht nur niedrigere Kosten, sondern auch ein schnellerer und flexiblerer Zugriff auf genau das Verhalten, welches das Team für die Tests benötigt.

Da virtuelle Assets die eingeschränkten Komponenten ersetzen, wird der Zugriff auf alle erforderlichen Systemkomponenten zu einem gewünschten Zeitpunkt erheblich vereinfacht.

## SCHNELLES ERSTELLEN, VERWALTEN UND BEREITSTELLEN VIRTUELLER ASSETS

Parasofts einzigartige Integration von Service-Virtualisierung mit der Verwaltung von Testumgebungen hilft Teams, den Prozess der Erstellung der verschiedenen und komplexen Umgebungen, die für gründliche Tests erforderlich sind zu beschleunigen als auch signifikant zu vereinfachen. Damit erhalten Entwicklungs-, QA- und Performance-Testingenieure einfachen Zugriff auf die spezifischen Kombinationen und Konfigurationen virtueller Ressourcen, die sie für verschiedene Testumgebungen benötigen.

Parasoft Virtualize unterstützt Entwicklungs- und QA-Teams beim Erstellen und dem Zugriff auf jede Umgebung, die für die Entwicklung oder den Test einer Anwendung benötigt wird. Durch die Aufzeichnung können Teams das notwendige Systemverhalten von abhängigen Anwendungen erfassen, indem sie entweder die Parasoft Proxy-Aufzeichnung nutzen oder Paket-Sniffing-Technologien wie Wireshark oder Fiddler verwenden. Diese Artefakte können dann in wiederverwendbare virtuelle Assets umgewandelt werden, um komplexe Testszenarien auszuführen. Diese leicht konfigurierbaren virtuellen Assets treten dann an die Stelle der eigentlichen abhängigen Anwendungen und geben den Teammitgliedern die Freiheit, ihre erwarteten Entwicklungs- und Testaufgaben durchzuführen, wann immer sie wollen und so umfangreich wie sie möchten.

Mit der Continuous Testing Platform (CTP) von Parasoft können Entwickler und Tester schnell die spezifischen Konfigurationen und Kombinationen von virtuellen Assets einrichten, die sie in einer bestimmten Testumgebung anwenden möchten. Ausgehend von einer grafischen Systemübersicht haben Anwender folgende Möglichkeiten:

- » Festlegen, ob jede der abhängigen Anwendungen durch ein virtuelles Asset ersetzt werden kann.
- » Bestimmen, welches Verhalten das virtuelle Asset im aktuellen Fall zeigen soll, wenn der obige Punkt bejaht wird.
- » Schnelles Definieren und Bereitstellen von Testumgebungen mit allen erforderlichen Abwandlungen des Verhaltens abhängiger Anwendungen und sofortiger Testbeginn.

### Virtuelle Assets beseitigen die mit abhängigen Anwendungen verbundenen Beschränkungen

Zur einfachen Aufzeichnung und Virtualisierung des Verhaltens abhängiger Anwendungen erfasst Parasoft Virtualize den Live-Verkehr der abhängigen Anwendungen, die mit der zu testenden Applikation verbunden sind. Nach der einfachen, einmaligen Einrichtung überwacht Parasoft Virtualize bei Bedarf das Verhalten an einer beliebigen Anzahl von Endpunkten und stellt dieses Verhalten dann in "virtuellen Assets" dar, die für die tatsächlich eingeschränkten Komponenten einspringen können. Sollte eine nachgelagerte Abhängigkeit unvollständig oder nicht verfügbar sein, kann eine Service-

Definition verwendet werden, um die virtuelle Version zu erzeugen. Neben Webservices erfasst Parasoft Virtualize auch das Verhalten von Datenbanken, Messaging-Systemen, Middleware, Mainframes, ESBs, Legacy-Systemen und mehr.

Bei der Arbeit in einer Umgebung, in der eingeschränkte Komponenten durch virtuelle Ressourcen ersetzt werden, haben Entwickler und Tester die Freiheit, die von ihnen erwarteten Aufgaben auszuführen, wann immer sie wollen und so umfangreich wie sie möchten. Teams, die mit Komponenten mit eingeschränktem Zugang arbeiten, müssen sich nicht mehr mit ungünstig

angesetzten Zugangszeiten herumschlagen (z. B. 2 Uhr nachts am Samstagmorgen). Sie müssen keine Zeit mehr damit verschwenden, die Testumgebung in den gewünschten Zustand zu versetzen, bisherige Zugangsgebühren fallen weg. Teams die parallel arbeiten möchten, sind nun keinen unnötigen Wartezeiten mehr ausgesetzt. Ebenso wird ein verspätetes Testen vermieden, in dem das Beheben von Fehlern exponentiell schwieriger als auch kostspieliger ist.

Virtuelle Assets sind rund um die Uhr für automatisierte oder manuelle Unittests, funktionale Tests und Performancetests verfügbar. Sie können von jeder Testumgebung genutzt und mithilfe eines beliebigen CI-Build-Systems wie Jenkins, Bamboo, Microsoft Azure DevOps, TeamCity, GitHub, GitLab usw. bereitgestellt werden.

### Virtuelle Assets sind äußerst flexibel und lassen sich leicht für unterschiedliche Testanforderungen konfigurieren

In einer durch reale Systeme erschaffenen Testumgebung gibt es oft keinen Zugang zu der Vielzahl abhängiger Anwendungen, die mit einer AUT verbunden sind. Diese Einschränkung wird für Performance-Tester noch verschärft, da bei zusammengesetzten Testumgebungen traditionell die erforderlichen Rechenressourcen fehlen, um realistische Performance-Szenarien durchzuführen. In der Regel liegt nur ein begrenztes Maß an Kontrolle über das Verhalten der Komponenten vor. Ebenso kann nur ein begrenzter Bereich von Bedingungen in Ihrem Verhalten getestet werden (z. B. die aktuelle tatsächliche Antwortzeit, aber nicht die Spitzenantwortzeit oder Ausfälle). Dies führt zu unvollständigen Tests und birgt erhebliche Geschäftsrisiken.

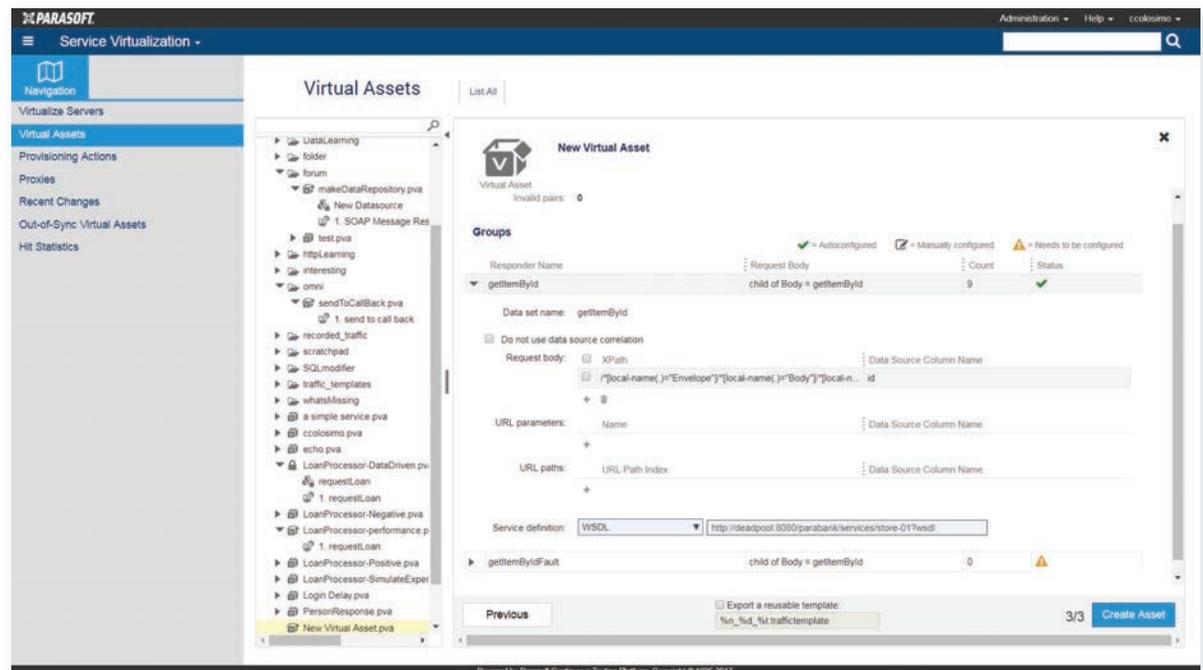


Abbildung 1:  
Erstellen, Konfigurieren  
und Verwalten virtueller  
Assets mit der Browser-  
Benutzeroberfläche der  
Service-Virtualisierung.

Mit der einfachen Benutzeroberfläche von Parasoft Virtualize lassen sich virtuelle Assets leicht konfigurieren, um die spezifischen Bedingungen zu reproduzieren, die für das Ausführen von Entwicklungs-/Testaufgaben entscheidend sind. So können z. B. verschiedene Fehler-, Ausfall- und Leistungsbedingungen konfiguriert werden, die sich mit realen Systemen nur schwer reproduzieren oder nur sehr langsam konfigurieren lassen. Durch das Hinzufügen von Datenquellen und das Bereitstellen von Bedingungskriterien lässt sich das virtualisierte Asset so einstellen, dass es wie erwartet oder auch unerwartet (für negative Tests) funktioniert. Damit wird für die Teams ermöglicht, das gesamte Spektrum des Systemverhaltens zu validieren, einschließlich der Fähigkeit, in verschiedenen Ausnahmesituationen korrekt zu reagieren (oder zumindest einen ordnungsgemäßen Ausfall zu gewährleisten).

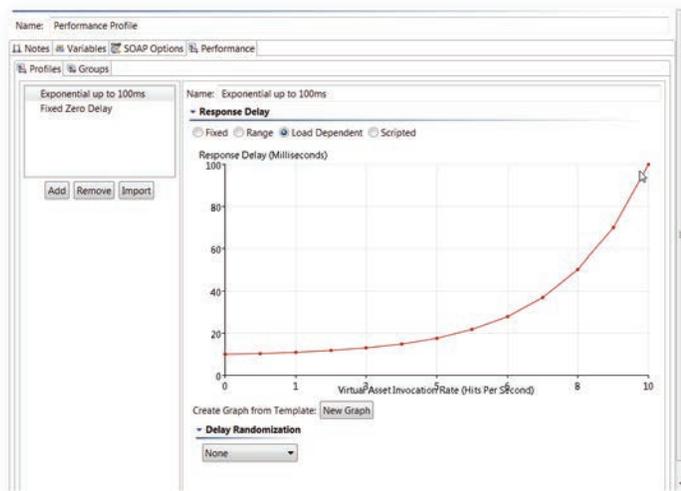


Abbildung 2:  
Zur Auswahl stehen unterschiedliche Profile, um die Leistung eines Dienstes unter Last zu simulieren.

Auf diese Weise erhält das Team die vollständige Kontrolle über ein virtuelles Asset, das eine abhängige Anwendung darstellt, über die in der realen Welt nur sehr wenig Kontrolle vorliegt. In dieser Umgebung kann das Verhalten, die Datenszenarien und das Leistungsprofil des abhängigen Systems gesteuert werden. Außerdem lassen sich mehrere Verhaltensprofile einrichten und mit einem einzigen Mausklick aktivieren oder deaktivieren.

### VIRTUELLE ASSETS LASSEN SICH IN EINER VOLLSTÄNDIGEN TESTUMGEBUNG LEICHT KONFIGURIEREN, BEREITSTELLEN UND ANPASSEN

Die Steuerung des Verhaltens eines einzelnen virtuellen Assets ist ein wichtiger Schritt, damit Entwickler und Tester die Freiheit erhalten, auf genau das abhängige Anwendungsverhalten zuzugreifen, das sie für den Test benötigen.

Parasoft geht noch einen Schritt weiter und ermöglicht die sofortige Definition und

Bereitstellung von Testumgebungen mit allen erforderlichen Permutationen des abhängigen Anwendungsverhaltens. Das bedeutet: Ein Anwender kann alle abhängigen Komponenten in einer Testumgebung sehen, jede Komponente grafisch in den gewünschten Zustand versetzen und dann auf eine Taste klicken, um die angegebene Kombination von Zuständen sofort für den Test verfügbar zu machen.

Eine vollständige Testumgebung, einschließlich virtueller Ressourcen, kann sofort bereitgestellt werden, um einen ganz bestimmten Bedarf zu erfüllen. Anschließend können sie schnell durch andere Umgebungsconfigurationen ersetzt werden, welche für gründliche Tests benötigt werden.

Werfen wir einen kurzen Blick darauf, wie dies mit dem Umgebungsmanager von Parasoft namens CTP (Continuous Testing Plattform) funktioniert.

## Konfigurieren von simulierten virtuellen Testumgebungen

Zunächst wird in einer Grundkonfiguration festgelegt, welche Elemente und Optionen für die Bereitstellung von Testumgebungen auf Abruf verfügbar sein sollen. Das passiert in folgenden Schritten:

- 1. Modellieren der Systemarchitektur.** Die Grundlage aller Testumgebungen ist die vollständige Systemabbildung, welche von einer mit dem System vertrauten Person modelliert wird. Dies bietet eine Übersicht über alle abhängigen, mit der AUT verbundenen Systeme sowie ein zentrales, leicht durchsuchbares Repository für die Konfigurationsdetails jeder Komponente.

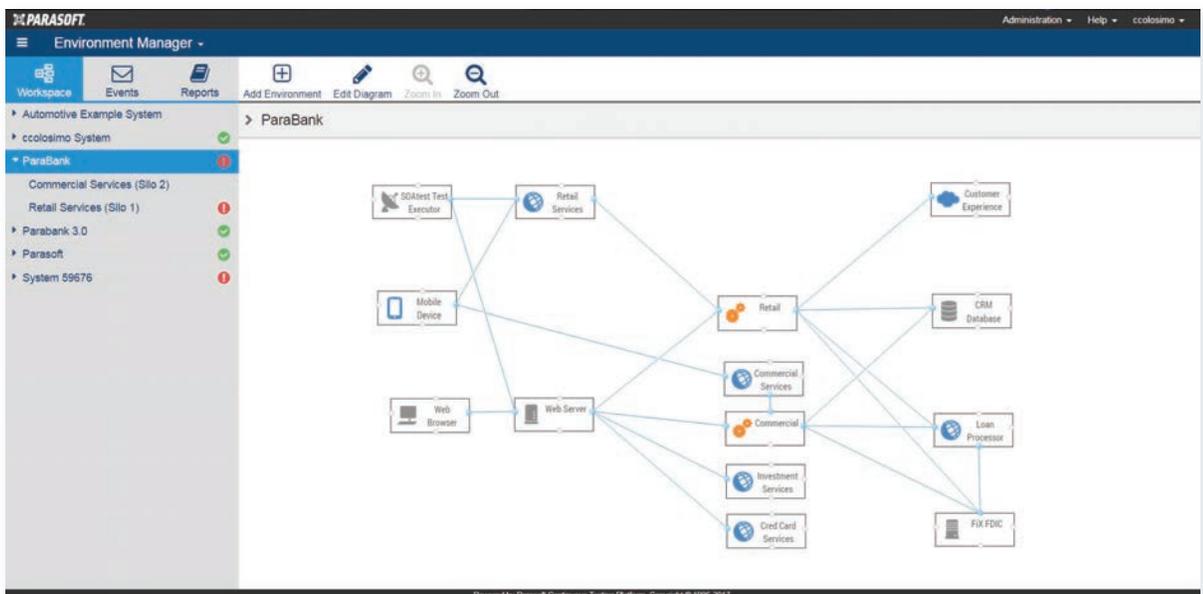
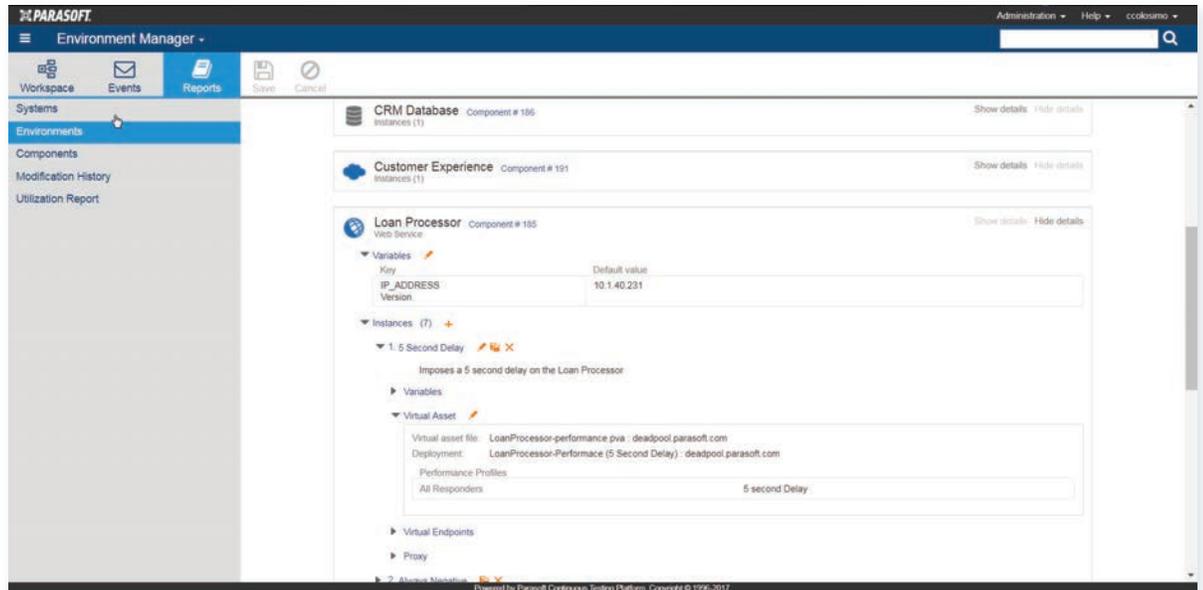


Abbildung 3:  
Ansicht des Umgebungsmodelldiagramms der verbundenen Dienste und ihrer Beziehungen sowie der Bereitstellung in verschiedenen Zuständen (live, virtuell usw.).

- 2. Definieren der verfügbaren Komponentenzustände.** Als nächstes werden die unterschiedlichen Zustände bestimmt, auf die jede Komponente in einer Testumgebung eingestellt werden kann. So lassen sich beispielsweise Optionen konfigurieren, um den Datenverkehr an die tatsächliche Komponente oder an eine Reihe verschiedener virtueller Assets zu leiten, die jeweils unterschiedliche Antworten, Leistungsprofile, Datensätze usw. liefern können. Möglich ist auch, eine beliebige Anzahl und einen beliebigen Umfang von Testfällen zu konfigurieren, die bei der Bereitstellung der Umgebung ausgeführt werden sollen. Dies bietet einen einfachen Ansatz für die Testautomatisierung.

Abbildung 4:  
Verknüpfen der  
virtuellen Assets und  
Proxy-Zustände mit  
Umgebungsinstanzen, um  
Konfigurationen für die  
Bereitstellung zu erstellen.

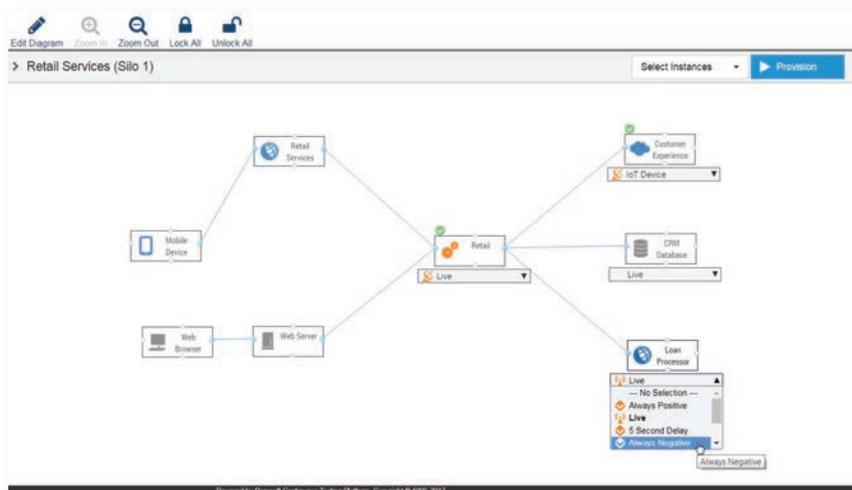


## Selbständige Bereitstellung von Umgebungen für den Zugang

Ausgehend von den auf dieser Grundlage erstellten Entwürfen können Entwickler und Tester dann die verschiedenen Umgebungen, auf die sie zugreifen müssen, selbst bereitstellen.

- 1. Spezifizieren einer vollständigen Umgebungsinstanz.** Anhand der Systemdefinition als Vorlage wählen Entwickler und Tester die Teilmenge der für eine bestimmte Testumgebung wichtigen Komponenten aus. Sie verwenden dann grafische Steuerelemente, um jede der relevanten Komponenten in den Zustand zu versetzen, der für eine bestimmte Testumgebung, mit der sie arbeiten müssen, erforderlich ist.

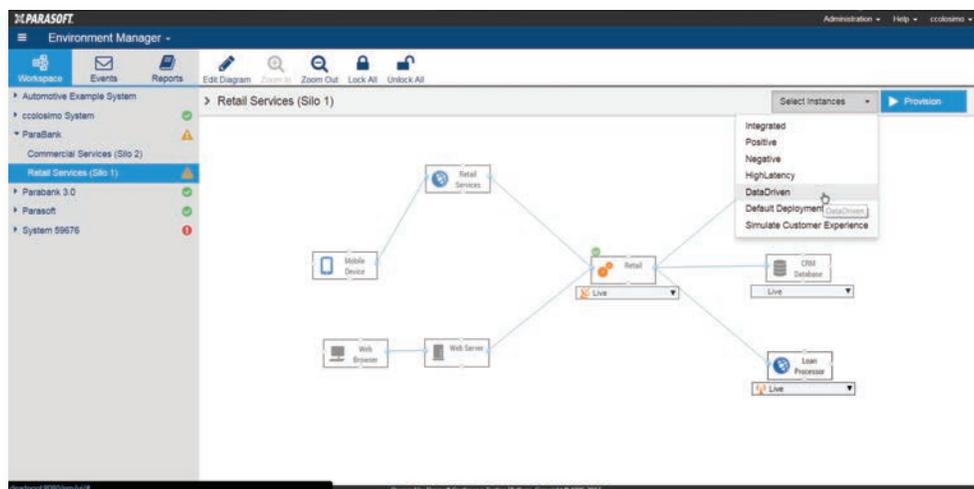
Abbildung 5:  
Konfigurieren einer  
Mischung aus realen und  
virtuellen Komponenten  
und Bereitstellen der  
resultierenden Umgebung,  
um die Fortsetzung der  
Tests ohne Verzögerungen  
zu ermöglichen.



Entwickler und Tester können dann die Einstellungen für eine andere Umgebungsconfiguration anpassen (z. B. durch Ändern des Zustands einiger Komponenten), eine zweite Umgebungsinstanz erstellen, zusätzliche Anpassungen vornehmen, eine weitere Instanz anlegen, und so weiter. So lassen sich die verschiedenen Umgebungen, auf die sie zugreifen müssen, schnell definieren.

2. **Bereitstellen einer Umgebungsinstanz, um sie zu aktivieren.** Für den Entwickler oder Tester ist es möglich, eine definierte "Umgebungsinstanz" sofort bereitzustellen oder sie für eine spätere Verwendung zu speichern. Mit einem einzigen Klick wendet Parasoft Virtualize alle verschiedenen Konfigurationsmaßnahmen an, die für das Einrichten der abhängigen Komponenten in dieser Testumgebung erforderlich sind. So kann das Testen sofort losgehen.

Abbildung 6:  
Auswahl einer  
vordefinierten  
Konfiguration aus einer  
Reihe von Instanzen zur  
schnellen Bereitstellung  
der Testumgebung.



3. **Bereitstellen und Zerstören von Umgebungen.** Nachdem Speichern der verschiedenen Umgebungsinstanzen in der Plattform können diese Konfigurationen bei Bedarf oder als Teil des Build-Systems auf entfernten Virtualisierungsservern bereitgestellt werden. Die Umgebungsconfiguration ist eine Vorlage, die alle Verbindungen, virtuellen Assets, Testfälle und Daten enthält. Parasoft CTP kann das Build-System aufrufen und diese Konfiguration dynamisch bereitstellen. So lassen sich je nach Bedarf verschiedene Versionen der erforderlichen Testumgebungen erstellen, um die Tests zu unterstützen. Sind die Tests abgeschlossen, können sie wieder abgebaut werden, um wertvolle Hardwareressourcen zu sparen.

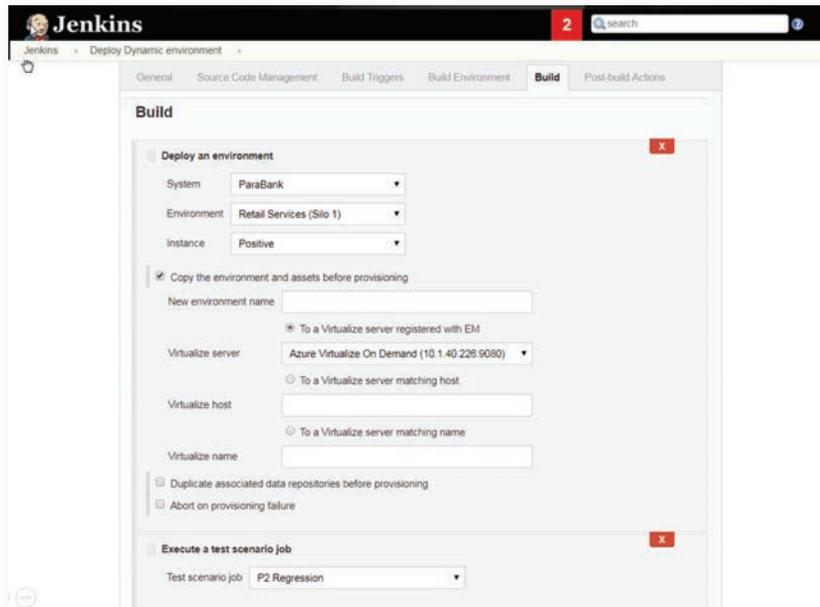


Abbildung 7:  
Die Bereitstellung der  
Testumgebung in die CI/  
CD-Pipeline integrieren  
unter Verwendung des  
bevorzugten Build-  
Systems.

## ZUSAMMENFASSUNG: NEUE (EINWEG-)TESTUMGEBUNGEN AUF ABRUF

Parasoft ermöglicht es Entwicklern und Testern, auf einfache Weise eine Testumgebung aus vorhandenen Komponenten zu konfigurieren, um eine bestimmte Testanforderung zu erfüllen. Eine Testumgebung kann auf diese Weise jederzeit verworfen werden. Sobald ein Team seine Arbeit mit einer Testumgebung beendet hat und sich Testanforderungen verändern, kann durch ein paar Anpassungen in der Konfiguration innerhalb von wenigen Minuten eine neue Testumgebung erschaffen werden. Während die Zeit für das Bereitstellen von Testumgebungen auf ein Minimum reduziert wird, wird der Fokus auf das tatsächliche Testen ermöglicht.

Mithilfe von virtuellen Assets können „Was wäre wenn“ Testszenarios jederzeit in nur wenigen Minuten durchgeführt werden. Da es sich bei den Endpunkten um eine Ansammlung von simuliertem abhängigem Systemverhalten handelt, birgt das Ausführen von Tests in dieser Umgebung keine externen Geschäftsrisiken.

Dieser Ansatz beseitigt auch das allgegenwärtige Problem, dass Teams die Arbeit der anderen überschreiben. Wenn mehrere Teams in einer herkömmlichen Testumgebung arbeiten, verbringt ein Team nur allzu oft viel Zeit damit, die Umgebung so zu konfigurieren, dass sie seinen spezifischen Anforderungen entspricht. Ein anderes Team setzt sich dann über diese Einrichtung hinweg, wenn es versucht, die Umgebung an seine eigenen Anforderungen anzupassen. Wenn jedes Team auf einfache Weise seine eigenen "Einweg"-Testumgebungen erstellen kann, die auf seine jeweiligen Bedürfnisse zugeschnitten sind, wird das Potenzial für solche Konflikte ausgeschaltet. Die Möglichkeit parallel zu testen stellt sicher, dass die Tests ungehindert fortgesetzt werden können - dies ermöglicht eine frühere Fertigstellung als auch Rückmeldung.

Zusammengefasst lassen sich folgende Herausforderungen mithilfe von Service-Virtualisierung lösen:

Herausforderungen	Wie die Service-Virtualisierung hilft
Der Zugriff auf Daten, die in einer separaten Infrastruktur gespeichert sind, kann sehr teuer sein.	Kein Zugriff auf ein reales System notwendig, Arbeiten mit einer zuvor simulierten virtuelle Umgebung.
Oft erfolgt das Testen spät im Zyklus, weil Systeme oder Umgebungen nicht zugänglich sind.	Frühzeitigeres Testen möglich, da Abhängigkeiten zu realen Systemen mit einer Simulation überbrückt werden.
Das späte Auffinden von Fehlern im Prozess kann sehr teuer werden und die Produktionszeitpläne beeinträchtigen.	Fehler früher finden und beheben durch das Vorverlegen von Tests ('Shift Left') innerhalb des CI/CD-Workflows.
Entwicklungsteams können in Verzug geraten, weil der rechtzeitige Zugriff auf wichtige Systeme nicht möglich ist.	Teams können ihre Entwicklung fortsetzen, indem sie mit einer Simulation von wichtigen Systemen arbeiten.
Unerwartete Verzögerungen bei vorgelagerten Teams führen zu weiteren Verzögerungen bei nachgelagerten Teams, die auf Testdaten warten.	Minimierung von Verzögerungen, weil die Teams mithilfe von virtuellen Services unabhängig von realen Systemen agieren können.
Externen Partnern den Zugang zu Testumgebungen zu ermöglichen, kann sehr teuer und langsam bei der Bereitstellung und Verwaltung sein.	Virtuelle Sandboxes erleichtern das Bereitstellen und Kontrollieren von Testumgebungen auf Abruf und ermöglichen dadurch schnellere Innovationen

## MACHEN SIE DEN NÄCHSTEN SCHRITT

Erfahren Sie mehr über das Erstellen, Bereitstellen und Verwalten virtueller Testumgebungen bei Bedarf - jederzeit und überall. [Sprechen Sie noch heute mit einem unserer Experten.](#)

### ÜBER PARASOFT

Parasoft unterstützt Unternehmen bei der kontinuierlichen Bereitstellung von Qualitätssoftware mit seiner marktbewährten, integrierten Suite von automatisierten Softwaretest-Tools. Die Technologien von Parasoft unterstützen die Märkte für eingebettete Systeme, Unternehmen und IoT. Sie reduzieren den Zeitaufwand und die Kosten für die Bereitstellung sicherer, zuverlässiger und konformer Software, indem sie alles von tiefgreifender Code-Analyse und Unit-Tests bis hin zu Web-UI- und API-Tests sowie Service-Virtualisierung und vollständiger Code-Abdeckung in die Lieferpipeline integrieren. Das preisgekrönte Reporting- und Analyse-Dashboard von Parasoft bietet einen zentralen Überblick über die Qualität und ermöglicht es Unternehmen, mit Zuversicht zu liefern und in den strategisch wichtigsten Ökosystemen und Entwicklungsinitiativen von heute erfolgreich zu sein - Sicherheit, sicherheitskritisch, agil, DevOps und kontinuierliche Tests.