

WHITE PAPER | APRIL 2016

# Test Data Management

Der vielleicht einzige Weg zu einer Continuous Delivery

Huw Price  
CA Technologies



## Inhaltsverzeichnis

---

|   |          |
|---|----------|
| <b>Abschnitt 1:</b>                                       | <b>3</b> |
| Einführung in die Continuous Delivery                     |          |
| <hr/>   |          |
| <b>Abschnitt 2:</b>                                       | <b>3</b> |
| Wie ein schwaches TDM eine Continuous Delivery verhindert |          |
| <hr/>   |          |
| <b>Abschnitt 3:</b>                                       | <b>6</b> |
| Ein anforderungsbasierter Ansatz für das TDM              |          |
| <hr/>   |          |
| <b>Abschnitt 4:</b>                                       | <b>8</b> |
| Zusammenfassung   |          |
| <hr/>   |          |
| <b>Abschnitt 5:</b>                                       | <b>9</b> |
| Literaturhinweise   |          |
| <hr/>   |          |
| <b>Abschnitt 6:</b>                                       | <b>9</b> |
| Informationen zum Autor                                   |          |

## Abschnitt 1

# Einführung in die Continuous Delivery

Im Bereich der Softwareentwicklung ist Continuous Delivery mittlerweile zu einer Art Schlagwort geworden.

Daher versprechen zahlreiche Anbieter ihre Verwirklichung und preisen ihre Tools als Allheilmittel gegen die typischen Ursachen von Projektverzögerungen und -fehlschlägen an. Sie behaupten, dass Unternehmen mit der Einführung ihrer Tools eine kontinuierliche Innovation und termin- und budgetgerechte Bereitstellung hochwertiger Lösungen erreichen.

Die Attraktivität der Continuous Delivery ist nachvollziehbar. In der modernen Application Economy verlassen sich Unternehmen auf Lösungen, um ihren Kunden einen Nutzen anzubieten. Daher sind die Anforderungen an Business und IT stark aneinander angeglichen, und die Marktposition eines Unternehmens hängt von seiner Fähigkeit ab, den Verbrauchern diesen Nutzen tagtäglich aufs Neue bereitzustellen zu können. IT-Teams müssen sich verändernden Markt- und Verbraucheranforderungen schnell anpassen können und Lösungen entwickeln, die auf wechselnde zentrale Anforderungen ausgerichtet sind und die Zeit und Kosten für das Testing reduzieren.

Obleich von den meisten Tools behauptet wird, dass sie die Continuous Delivery unterstützen, sind sie vorwiegend logistischer Natur und können erst in einer späten Phase des Entwicklungszyklus bereitgestellt werden. Aus dieser Perspektive wird DevOps nur in Hinblick auf Operations betrachtet, und Tools werden auf eine bereits designte Lösung angewendet, um Entwicklung und Regressions-Testing zu unterstützen. Die systemisch bedingten Probleme, die bereits früh in der Phase der Erfassung von Anforderungen entstehen, bleiben dann bestehen und bewirken kostspielige Engpässe und Projektverzögerungen, die eine erfolgreiche Implementierung von Continuous Delivery unmöglich machen.

Besonders häufig wird dabei die Bedeutung des Test Data Management (TDM) übersehen, bei dem es darum geht, für das Testing die richtigen Daten zur richtigen Zeit am richtigen Ort bereitzustellen. Ein schwaches Testfalldesign und die ineffiziente Provisionierung qualitativ unzureichender Daten bedeuten, dass Testteams nicht die Daten erhalten, die sie für das umfassende Testing eines Systems benötigen. Und so wird dann zugunsten der termin- und budgetgerechten Bereitstellung einer Lösung ihre Qualität beeinträchtigt.

---

## Abschnitt 2

# Wie ein schwaches TDM eine Continuous Delivery verhindert

Für die meisten Unternehmen und Anbieter beginnt und endet das TDM mit dem Kopieren, Maskieren und möglicherweise noch Unterteilen von Produktionsdaten. Diese TDM-Strategien sind rein logistischer Natur und konzentrieren sich lediglich auf das Verschieben von Daten, wobei die Daten bei der Definition der Projektanforderungen außen vor gelassen werden. Nach dem Kopieren der Produktionsdaten und ihrer Migration in eine Entwicklungs-, Testing- oder QA-Umgebung werden sie häufig als „Gold Copy“ bezeichnet, also als perfekter Testdatensatz, mit dem Tester alle für einen umfassenden Systemtest erforderlichen Daten haben.

Anschließend wird mit dem Maskieren und Unterteilen begonnen, um die traditionellen Problempunkte in Bezug auf Infrastrukturkosten und Einhaltung von Vorschriften Stück für Stück anzugehen. Aber allein schon das Maskieren und Unterteilen einer Produktionsdatenbank beinhaltet viele der Probleme, die gerade mit der Verwendung von Produktionsdaten selbst zusammenhängen. Diese Probleme machen eine erfolgreiche Implementierung der Continuous Delivery unmöglich.

## Verschwendete Zeit

Vielen Unternehmen fehlt ein zentrales TDM-Team oder ein Provisionierungsservice, sodass die Daten von lokalen Teams zusammengesucht bzw. erstellt werden müssen. Tester können bis zu 50 % ihrer Arbeitszeit mit dem Suchen nach Daten und etwa 20 % des gesamten SDLC mit dem Warten darauf verbringen. Das führt zu Testing-Engpässen, die eine echte Agilität und Continuous Delivery verhindern. In einem Fall verwendete ein Team Grid-Tools (jetzt zu CA Technologies gehörend) und sollte ein Projekt in einem dreiwöchigen „Sprint“ abschließen, aber es benötigte vier Wochen Zeit für die Vorbereitung der Daten.

Das Maskieren und Unterteilen von Produktionsdaten beinhaltet nämlich auch, dass die Testteams die benötigten Daten zuerst einmal zusammentragen müssen. Dies ist eine zeitaufwendige und mühsame Aufgabe, die durch eine inkonsistente Datenspeicherung in unkontrollierten Kalkulationstabellen noch weiter erschwert wird. So könnten zum Beispiel personenbezogene Daten (Personally Identifiable Information, PII) in einer „Anmerkungen“-Spalte vermerkt sein, beispielsweise wenn es drei Spalten für Kreditkarten gibt, ein Kunde aber vier Kreditkarten hat. Darüber hinaus müssen die Daten referenziell intakt sein. Je komplexer diese Daten sind, desto schwieriger ist dies zu erreichen, und die von Testern für die Maskierung aufgewendete Arbeit und Zeit stehen häufig in keinem Verhältnis zu einer kompletten Neuerstellung der Daten.

Die manuelle Neuerstellung von Daten kann gleichermaßen zeitaufwendig sein. Außerdem sind die erstellten Daten für gewöhnlich auf spezifische Testfälle ausgerichtet, die durch die stetigen Veränderungen in der Realität schnell veraltet und damit nicht mehr relevant sind. Als typisches Beispiel dafür werden oft Handelsmuster und Wechselkurse für Währungen herangezogen, bei denen Daten praktisch jeden Tag veralten. Wenn dies geschieht, müssen die Daten aktualisiert werden – eine weitere zeitintensive Aufgabe – oder, was häufiger der Fall ist, sie werden einfach „verbrannt“. Das ist gleichermaßen der Fall, wenn Versionsänderungen stattfinden, wenn jemand eine Datenaktualisierung anfordert oder auch jedes Mal, wenn neue Daten erstellt werden müssen, weil sich die virtuelle Umgebung geändert hat.

Manuell erstellte Daten zwischen den Teams austauschen und wiederverwenden zu können, ist nur selten möglich. Die Unfähigkeit, zuvor geleistete Arbeit bei der Datenerstellung wiederverwenden zu können, erfordert erneute Arbeiten. Das stellt Unternehmen unmittelbar vor die unangenehme Wahl, entweder die Kosten zu senken und die Markteinführungszeit zu verkürzen oder aber hochwertige Softwareanwendungen mit allen benötigten Funktionalitäten bereitzustellen.

## Einschränkungen durch Abhängigkeiten

Einen großen Teil seiner Arbeitszeit verbringt ein Tester damit, darauf zu warten, dass Daten provisioniert oder verfügbar gemacht werden. Häufig betrachten Unternehmen und Anbieter den SDLC als eine Abfolge linearer Phasen, bei dem ein Team etwas fertigstellt und es an ein anderes weiterleitet. Dabei müssen Teams warten, bis die Daten, die „für ihren Zweck geeignet sind“, von den vorgelagerten Teams verfügbar gemacht worden sind, und können die benötigten Daten dann aber womöglich gar nicht nutzen, weil ein anderes Team gerade mit ihnen arbeitet.

Das steht im völligen Gegensatz zu den Grundsätzen des Agile Parallel Development, die im Mittelpunkt der Continuous Delivery stehen. Hier wird versucht, aus jedem Moment der Arbeitszeit und der Innovationskraft eines Mitarbeiters den maximalen Nutzen zu ziehen. Hinzu kommt, dass die Änderungen, die ein Team an den Daten vornimmt, dazu führen können, dass die Anwendungstests mit diesen Daten durch ein anderes Team scheinbar grundlos fehlschlagen.

Wenn die Teams die Daten manuell erstellen oder Tage oder Wochen auf ihre Verfügbarkeit warten müssen, kann niemand von ihnen erwarten, dass sie sich veränderten Anforderungen schnell anpassen und eine umfassend getestete Lösung bereitstellen.

## Schlechte Qualität

Das eigentliche Problem bei der Verwendung von Produktionsdaten in Nicht-Produktionsumgebungen ist die Qualität. Produktionsdaten weisen für gewöhnlich nur 10–20 % funktionale Abdeckung auf, und deshalb ist es unwahrscheinlich, dass Stichprobenverfahren Daten bereitstellen, die alle zum Aufbau eines neuen Subsystems erforderlichen Testfälle abdecken. Der Großteil der Produktionsdaten ähnelt sich sehr stark, da die Daten aus gewöhnlichen bzw. „tagtäglichen“ Transaktionen herrühren und von sich aus bereits die „schlechten“ Daten ausschließen, die einem System Probleme bereiten könnten. Aus diesem Grund tendiert das Testing dazu, sich eher auf den „Happy Path“ und weniger auf nicht funktionales und Negativ-Testing zu konzentrieren.

Das Negativ-Testing sollte aber etwa 80 % des Testings ausmachen, weil es gerade die Ausreißer und Randszenarien sind, die ein System kollabieren lassen können. Solange das Testing den Schwerpunkt auf den Happy Path legt, werden unweigerlich Fehler ihren Weg bis in die Produktion finden, was zu Nacharbeiten, kritischen Verzögerungen, ausufernden Kosten und potenziellen Misserfolgen von Projekten führen kann. Einen Fehler nicht schon in der Anforderungsphase zu erkennen und zu beheben, sondern erst während des Testings, dauert Branchenuntersuchungen zufolge 50 Mal länger<sup>1</sup>. Mit solchen Verzögerungen und einem trägen Reaktionsvermögen auf Änderungen kann die Continuous Delivery unmöglich erreicht werden.

Die funktionale Abdeckung von Produktionsdaten kann zwar durch manuell erstellte Daten ergänzt werden, aber das ist eine ungenaue, zeitaufwendige und unwissenschaftliche Methode. Es gibt keine Möglichkeit, die maximale Abdeckung zu verifizieren oder sicherzugehen, dass die Daten referenziell intakt bleiben.

## Hohe Kosten

Neben den hohen Kosten von Nacharbeiten und verzögerter Bereitstellung stellt auch das Kopieren von Produktionsdaten ein nahezu unerträglich langsames und kostspieliges Verfahren dar. Berichten zufolge leisten sich einige Unternehmen bis zu 20 Kopien einer einzigen Datenbank, einschließlich der hohen Kosten für Hardware, Lizenzen und Support.

Außerdem kann ein hochvolumiger Daten-Storage extrem teuer sein. Der Studie eines Storage-Drittanbieters nach kann das Verschieben von Daten in kostengünstigere Umgebungen, einschließlich der Cloud, zwar vorübergehend die Storage-Kosten senken, aber es löst nicht die grundlegenden Probleme des Datenwachstums. Anders gesagt, jede TDM-Richtlinie, die auf Migration, Unterteilung und Maskierung hinausläuft, steht vor demselben Problem: Wie kann man die Daten, die in modernen und zunehmend komplexeren Composite-Anwendungen verarbeitet werden, kostengünstig speichern?

Hinzu kommt das potenziell weitaus kostspieligere Risiko, Vorschriften nicht einzuhalten, das auch durch die Maskierung von Produktionsdaten nicht umgangen werden kann. Eine unabhängige Studie hat aufgezeigt, dass die Kosten durch Datenschutzverletzungen 2014 um durchschnittlich 15 % auf 3,5 Mio. USD pro Vorfall angestiegen sind. Gleichzeitig sieht die neue, 2016 in Kraft tretende EU-Datenschutzrichtlinie maximale Geldbußen für Datenschutzverletzungen von bis zu 100 Mio. EUR<sup>2</sup> oder 5 % des weltweiten Umsatzes vor – je nachdem, welcher Betrag höher ist.

Das Maskieren von Produktionsdaten garantiert nicht die Einhaltung von Vorschriften, weil die wirkliche Gefahr im menschlichen Fehlverhalten liegt, denn über 50 %<sup>3</sup> der Datenschutzverletzungen stehen im Zusammenhang mit Insider-Handlungen. Zudem muss bei der Maskierung die referenzielle Integrität der Daten gewährleistet bleiben, weil sonst das Testing von Anwendungen beeinträchtigt wird. Aber je komplexer die Daten sind, desto leichter sind sie zu kompromittieren, weil immer mehr einzelne Informationsteile korreliert werden können.

Der einzige Weg, diese hohen Infrastrukturkosten zu vermeiden und den Schutz vertraulicher Daten sicherzustellen, ist der Verzicht auf die Verwendung von Produktionsdaten. Auch wenn diese Ausgaben für Unternehmen, die ein Framework für die Continuous Delivery implementieren möchten, nicht bezeichnend sind, so stehen sie doch im Gegensatz zu deren Grundsätzen. Diese Kosten verhindern, dass Unternehmen hochwertige Lösungen, die sich schnell ändernden Unternehmensanforderungen anpassen können, termin- und budgetgerecht bereitstellen können.

### Abschnitt 3

## Ein anforderungsbasierter Ansatz für das TDM

Unternehmen, die eine Continuous Delivery implementieren möchten, müssen ihre Testing- und Entwicklungsprozesse überdenken, statt sie einfach nur neu anzuordnen. Sie müssen ihren Ansatz für das TDM auf den Prüfstand stellen. Ein umfassender und anforderungsgesteuerter End-to-End-Ansatz für das Test Data Management ermöglicht Unternehmen ein Testing nach dem „Shift Left“-Prinzip, die Reduzierung von Risiken, die Minimierung von Fehlern und in der Folge die zügigere und günstigere Bereitstellung hochwertiger Lösungen.

Anstatt Daten von Testfall zu Testfall zu betrachten, sollten Unternehmen dazu übergehen, Daten auf Basis von Designentscheidungen zu sehen – also hinsichtlich der Anforderungen selbst – und ihre Testfälle mit direkter Verknüpfung zu den Daten entwickeln. Die Ausrichtung von Testdaten auf Anforderungen stellt sicher, dass sie für ihren Zweck geeignet sind, und ihre schnelle Provisionierung für Testteams bedeutet, dass sie veränderten Unternehmensanforderungen zügig angepasst werden können.

Deshalb darf die Continuous Delivery nicht erst irgendwann innerhalb des SDLC beginnen: Sie muss gleichzeitig mit der Idee – mit der Definition der Anforderung selbst – beginnen, damit Test- und Entwicklungsteams auf veränderte Unternehmensanforderungen schnell reagieren und sofort zu Beginn der sich verändernden Idee mit ihrer Arbeit anfangen können. Teams müssen Anforderungen mühelos validieren und verifizieren können und daraus Testfälle ableiten, deren erwartete Ergebnisse und virtuelle Daten direkt mit ihnen verknüpft sind. Dieses modellbasierte Testing ist das, was eigentlich mit „Shift Left“ und kontinuierlicher Bereitstellung von Lösungen gemeint ist – die Verdichtung aller Arbeiten des Entwicklungszyklus in die Phase der Erfassung von Anforderungen, aus der sich dann alle nachfolgenden Arbeiten einfach weiterentwickeln, selbst wenn sich die Anforderungen ändern sollten.

### Erstellung besserer Anforderungen

Bereits die Anforderungen selbst müssen sämtliche qualitativen Informationen über das benötigte Testing-System enthalten, damit die Tester direkt aus dieser anfänglichen „Idee“ Anwendungs- und Testfälle, einschließlich ihrer Nachverfolgbarkeit untereinander, ableiten können. Diese Nachverfolgbarkeit ist notwendig, damit Tester ihre Tests schnell an veränderte Anforderungen anpassen können.

Über ein Modeling mit geeigneten Tools kann auf einfache Weise ein Flowchart erstellt werden, das alle qualitativen Informationen über das benötigte Testing-System abbildet. Mit CA Agile Requirements Designer (früher Agile Designer von Grid-Tools) können Sie ein Flowchart mit allen funktionalen Logiken erstellen, um daraus automatisch die für die maximale Abdeckung kleinste erforderliche Anzahl von Testfällen abzuleiten, ohne damit bei Unternehmensverantwortlichen für unnötige Verwirrung zu sorgen, wie es bei Ursache-Wirkung- oder paarweisen Ansätzen der Fall sein kann<sup>2</sup>.

Zudem erhöht ein Flowchart die Wahrscheinlichkeit, dass Anforderungen unzweideutig und vollständig erfasst werden. Es wird unnötig, Betrachtern ein nicht zusammenhängendes „Meer aus Wörtern“ und umständliche Diagramme vorzusetzen, in denen die Anforderungen für gewöhnlich in kleine, leicht verdauliche Stücke heruntergebrochen werden. Dieses Verfahren spiegelt die Logik von Ursache und Wirkung in einem System wider und stellt sie in einer Abfolge von Wenn-Dann-Aussagen dar.

Dies hilft nicht nur, die durch zweideutige Anforderungen verursachte Fehlerquote von 56 %<sup>4</sup> zu verringern, sondern es zwingt das Team auch dazu, beim Definieren von Anforderungen wie beim Modeling eines Systems vorzugehen – und Einschränkungen und Neben- und Randbedingungen zu berücksichtigen. Durch die in einem solchen Flowchart modellierten Anforderungen können dann alle möglichen Pfade durch ein System identifiziert werden. Anschließend können daraus Testfälle mit 100 % funktionaler Abdeckung abgeleitet werden, bei denen das Testing jeden möglichen Systempfad abdeckt, einschließlich negativer Pfade und unerwarteter Ergebnisse<sup>3</sup>.

Weiterhin können Testfälle mit einem aktiven Flowchart mit komplexen Messdaten, virtuellen Daten, Testdaten, Automatisierungs-Scripts, erwarteten Ergebnissen und Backlogs verknüpft werden, um im SDLC einen stärkeren Fokus auf die Phase der Erfassung von Anforderungen zu legen.

### Provisionierung „für ihren Zweck geeigneter“ Daten

Durch die Automatisierung der Prozesse für das Testfalldesign mit Tools wie CA Agile Requirements Designer können Tester genau die Testfälle erzeugen, die zur optimalen Erfüllung der Testanforderungen erforderlich sind. Dies führt uns zur zentralen Aussage dieses Berichts: Für die Continuous Delivery ist ein besseres TDM erforderlich. Tester brauchen zur richtigen Zeit und am richtigen Ort Zugriff auf „für ihren Zweck geeignete“ Daten, wenn ihre Testfälle eine Abdeckung von 100 % erreichen sollen.

### Qualität

Wie bereits beschrieben, kann mit Produktionsdaten nicht die benötigte Abdeckung für ein umfassendes Testing eines Systems erreicht werden. Demgegenüber können synthetisch erzeugte Testdaten kleine, aussagefähige Datensätze produzieren, die sämtliche möglichen Szenarien abdecken, selbst wenn ein Ereignis zuvor noch nie stattgefunden hat. Jedes Szenario aus der Praxis kann als weiterer Datenpunkt gedacht werden, und so können auch für neue und bevorstehende Szenarien Daten erzeugt werden. CA Test Data Manager (früher CA Data Finder bzw. Data Maker von Grid-Tools) nutzt intelligente Techniken zur Datenprofilerstellung, um ein präzises Datenmodell abzubilden und aussagekräftige, differenzierte Daten zu erzeugen, die 100 % der auf ihnen basierenden funktionalen Variationen abdecken.

Bei diesem anforderungsbasierten Ansatz werden die erzeugten Daten einem Testfall zugeordnet und damit sichergestellt, dass sie allen Anforderungen jedes Testers entsprechen. Solch eine „Datenerzeugung für Testfälle“ erhöht die Wahrscheinlichkeit, dass Testteams bereits zu Beginn Fehler erkennen. Dadurch werden zeitaufwendige Nacharbeiten vermieden, die eine Continuous Delivery unmöglich machen.

### Zeit

Wie zuvor erwähnt, steht der Zeitaufwand für das manuelle Erstellen oder Bearbeiten von Daten oder das Warten darauf, dass sie verfügbar werden, im völligen Gegensatz zu den Grundsätzen einer kontinuierlichen Entwicklung.

Demgegenüber können Tools zur Automatisierung der Datenerzeugung wie CA Test Data Manager direkt mit RDBMs oder ERP API-Schichten zusammenarbeiten, damit Anwender so schnell Daten erzeugen können, wie es die Rechenleistung zulässt. Je nach Verarbeitungsgeschwindigkeit der Datenbankinfrastruktur kann das „Bulking“ (Zusammenfassen) von Scripts die Datenmenge in einem Unternehmen verdoppeln. Das stellt sicher, dass Tester die „für ihren Zweck geeigneten“ Daten nicht erst in Wochen, sondern innerhalb von Stunden bereitgestellt bekommen.

Zudem bedeutet eine leistungsstarke Testzuordnungsfunktion auch, dass bereits vorhandene Daten aus unterschiedlichen, nicht zusammenhängenden Quellen identifiziert, gefiltert und den entsprechenden Tests zugewiesen werden können. Bei dieser Testzuordnung hat sich gezeigt, dass der Zeitaufwand für das Suchen und Provisionieren der Daten gegenüber manuellen Prozessen um 95 % verringert werden kann. Zudem können Teams erkennen, welche Tests wegen Datenfehlern fehlschlagen werden, bevor sie sie durchführen.

Die Speicherung von Daten in einem zentralisierten Data Warehouse kann das Risiko von Engpässen noch weiter verringern, weil die Teams nicht mehr darauf warten müssen, bis die Daten verfügbar sind. Das On-Demand-Testdatenportal von CA Test Data Manager kann dynamisch gestaltet werden, sodass Anwender benötigte Datentypen auf Grundlage spezifischer Kriterien, wie z. B. Kreditkartentyp oder geografischer Standort, auswählen können. Bei diesem Verfahren werden die erforderlichen Daten entweder dem entsprechenden Testfall zugeordnet oder, sofern keine Daten existieren, neue Daten erzeugt. Dadurch erhalten Techniker mehr Zeit zur Behebung der beim Testing erkannten Fehler und müssen nicht die Hälfte ihrer Arbeitszeit auf die Provisionierung der Daten warten. Das heißt auch, dass Unternehmen die Datenbesitzrechte durch das IT-Security-Team zentralisieren können und vertrauliche Daten nur auf Anfrage durch autorisierte Mitarbeiter provisioniert werden.

### Kosten

Und schließlich kann eine bessere TDM-Richtlinie das Risiko von Kostenüberschreitungen reduzieren. Die frühere Fehlererkennung kann das Auftreten von Fehlern um bis zu 95 % verringern<sup>5</sup> und ermöglicht Einsparungen von über 50.000 USD pro Fehler<sup>5</sup>. Zudem können durch Verwendung kleinerer, aussagekräftigerer Datensätze die Infrastrukturkosten um bis zu 50.000 USD pro Datenbank reduziert werden<sup>5</sup>. Auch das Ausführen von weniger und hochwertigeren Tests kann den Zeit- und Kostenaufwand für das Testing deutlich verringern. Hinzu kommt, dass durch die Verwendung von synthetischen Daten das Risiko einer kostspieligen Nichteinhaltung von Vorschriften wegfällt, weil vertrauliche Daten die Produktionsumgebungen gar nicht erst verlassen.

---

## Abschnitt 4

### Zusammenfassung

Das Kopieren von Produktionsdaten als „zentrale Datenquelle“ in Nicht-Produktionsumgebungen verhindert eine erfolgreiche Implementierung der Continuous Delivery. Wenn Testteams die Lösung dann umfassend testen möchten, stehen sie vermutlich ohne die „für ihren Zweck geeigneten“ Daten da. Daher werden sie auch nicht in der Lage sein, schnell auf veränderte Unternehmensanforderungen reagieren zu können. Tools für das Unterteilen und Maskieren können zwar einige Problempunkte beseitigen, aber Produktionsdaten fehlt die erforderliche Qualität, um Fehler erkennen und beheben zu können.

Die Einführung eines anforderungsbasierten End-to-End-Ansatzes in der Lösungsentwicklung und die Anerkennung von Testdaten als wertvolle und wiederverwendbare Assets ist der beste Weg zur erfolgreichen Implementierung einer Continuous Delivery. Durch die Mitarbeit an der „Idee“ können Teams auf veränderte Unternehmensanforderungen schnell reagieren und erhalten zur richtigen Zeit und am richtigen Ort Zugriff auf „für ihren Zweck geeignete“ Daten, um Lösungen mit der geringstmöglichen Anzahl von Testläufen umfassend zu testen.

Dieser Ansatz ermöglicht es Unternehmen, die Effizienz ihrer Testdatenprovisionierung zu erhöhen und dabei gleichzeitig das Risiko von Verzögerungen, Nacharbeiten und ausufernden Kosten zu minimieren, das eine Continuous Delivery unmöglich macht. Er ebnet den Weg zu einer kontinuierlichen Bereitstellung hochwertiger Lösungen, die zentrale Unternehmensanforderungen termin- und kostengerecht erfüllen.



## Abschnitt 5

### Literaturhinweise

- 1 <http://www.softwaretestingclass.com/why-testing-should-start-early-in-software-development-life-cycle/>
  - 2 <http://www.agile-designer.com/resources/test-case-generation-bloor-market-report/>
  - 3 Siehe: „Primer of A Critique of Testing for a conceptual, mathematical treatment of the advantages of flowchart modelling“ von Llyr Wyn Jones, erhältlich unter <http://www.agile-designer.com/resources/critique-testing-primer/>
- 

## Abschnitt 6

### Informationen zum Autor

#### Mitbegründer von Grid-Tools und VP bei CA Technologies



Huw Price wurde von QA Guild zum „IT-Leiter des Jahres 2010“ gewählt. Er war der leitende technische Architekt bei mehreren Softwareunternehmen in den USA und in Europa. Als Experte für Tools zur Automatisierung von Tests war er für die Einführung zahlreicher innovativer Produkte verantwortlich, die das in der Softwarebranche verwendete Testing-Modell stark verändert haben.



Kontaktieren Sie CA Technologies unter [ca.com/de](http://ca.com/de).



CA Technologies (NASDAQ: CA) entwickelt Software, die Unternehmen bei der Umstellung auf die Application Economy unterstützt. Software steht in allen Branchen und in allen Unternehmen im Mittelpunkt. Ob Planung, Entwicklung, Management oder Security – CA Technologies arbeitet weltweit mit Unternehmen zusammen, um die Art, wie wir leben, Transaktionen abwickeln und kommunizieren, in mobilen, privaten und öffentlichen Cloud-Umgebungen oder in verteilten Systemen und Mainframe-Umgebungen neu zu gestalten. Weitere Informationen zu unseren Customer Success-Programmen finden Sie unter [ca.com/customer-success](http://ca.com/customer-success). Weitere Informationen finden Sie unter [ca.com/de](http://ca.com/de).

1 Ponemon Institute, 2014, „Cost of Data Breach“

2 Vorgeschlagene EU-Datenschutzrichtlinie

3 Verizon, 2013: „Data Breach Investigations Report“

4 Bender Report: „Requirements Based Testing“

5 Die Messdaten wurden im Verlauf von Implementierungen durch Grid-Tools erfasst.