

Test Data Management – Investieren oder ignorieren?

Klaus Haller

Die Jagdsaison ist eröffnet. Neue und etablierte Toolanbieter rüsten beim Thema Test Data Management auf. Und wenn Gartner dieses Feature (mit)bewertet, muss es wichtig sein. Ist also jetzt die Zeit reif, um mit Toolherstellern zu reden, das beste Tool zu finden und im eigenen Test Center einzuführen? Wer so vorgeht, riskiert eine Investitionsruine. Tools, die unterstützen, sind wichtig. Ihre Rolle beim Test Data Management wird aber oft überbewertet. Daher erklärt dieser Artikel, was Testorganisationen und IT-Abteilungen mit Test Data Management erreichen können und was dafür notwendig ist.

Test Data Management ist eng mit Informationssystemen, komplexen Applikationslandschaften oder ganz allgemein mit Datenbankapplikationen verbunden. Es geht um Applikationen, die auf eine oder mehrere Datenbanken zugreifen, egal ob die Datenbanken zur Applikation selbst gehören oder „stand-alone“ oder von einer anderen Applikation betrieben werden. Die Mehrheit der Businessapplikationen gehört zu dieser Kategorie, beispielsweise Bankapplikationen und ERP-Systeme. Dort ist Test Data Management auf allen Teststufen ein Thema, von den Modul- und Integrationstests bis hin zu User Acceptance Tests und System Integration Tests komplexer Applikationslandschaften.

Bei Datenbankapplikationen hilft, die Test Case-Definition gegenüber der aktuellen Lehrbuchmeinung zu verfeinern. In der Praxis werden heute bei Test Cases drei Elemente genau beschrieben. Erstens sind das die Inputparameter, die in GUI-Masken eingegeben werden müssen. Zweitens ist das die Bedienung der Applikation – welcher Button auf welcher Maske in welchem GUI wann bedient wird. Drittens wird der erwartete Zielzustand detailliert beschrieben. Für „Taschenrechner-Applikationen“ reichen diese drei Elemente. „3+4=“ muss bei jeder Eingabe „7“ liefern. Die drei Elemente reichen nicht für Datenbankapplikationen. Ein Test Case „Bankkunde hebt Geld ab“ mit den fixen Eingabeparametern „Kontonummer 53357“ und „Betrag EUR 4'000“ muss je nach Daten in der Datenbank ein anderes Resultat liefern. Das kann „Neuer Saldo: EUR 4'200“ sein, genauso „Abhebung zurückgewiesen, unzureichender Saldo“ oder „Kontonummer unbekannt“.

Dieses Beispiel illustriert zwei weitere wichtige Elemente bei Test Cases für Datenbankapplikationen. Beide gehören zur Precondition eines Test Cases. In der Praxis werden sie aber kaum beachtet. Einerseits geht es um das Objekt, mit dem der Test ausgeführt wird (kurz: DB(Datenbank)-Testing-Objekt). Das kann ein Kunde sein, eine Aktie oder in dem Beispiel das Konto mit der Kontonummer 53357. Die Kontonummer wird beim Testen über das GUI eingegeben. Doch

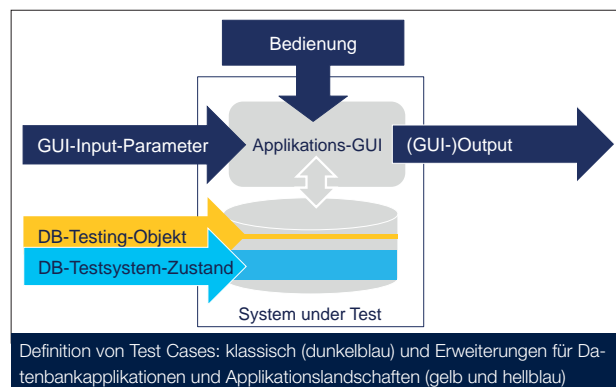


Klaus Haller

das Konto muss vor dem Start des Tests in der Datenbank verfügbar sein und einen passenden Saldo haben. Daneben müssen oft hunderte Tabellen der Bankapplikation mit konsistenten Daten befüllt sein, bevor der Prozess „Bankkunde hebt Geld ab“ ohne Absturz funktioniert. Das geht von Kunden-, Konti- und Buchhaltungstabellen bis hin zu Bankorganisati-

on und Mitarbeitern (DB-Testsystemzustand). Sie sind die „Bühne“ für den Test. Sie treten während des Tests nicht in Erscheinung. Fehlen sie aber, stürzt die Applikation ab und kann nicht getestet werden.

In einem solchen Umfeld haben Testorganisationen zwei Gründe, um ins Test Data Management zu investieren. Der erste Grund ist die Optimierung des Testcenters für Datenbankapplikationen, der zweite der Datenschutz. Beim Datenschutz geht es um regulatorische Vorschriften und um Reputationsrisiken. Die Lufthansa-Bonusmeilen-Affäre ist ein Beispiel für Reputationsrisiken. Doch Kundendaten können nicht nur wie bei Lufthansa aus den produktiven Systemen in die Öffentlichkeit gelangen. Testsysteme sind genauso gefährdet. Sie enthalten meist Kopien der Daten der Produktion. Entwickler und Tester haben auf die Daten in Testumgebungen oft freien Zugang. Sie können mit SQL-Anfragen grosse Datenmengen entwenden. Neben dem Reputationsrisiko gibt es regulatorische Vorschriften wie Datenschutzgesetze und Bankkundenschutz. Sie müssen eingehalten werden, sonst greifen Aufsichtsbehörden und Justiz ein. Bei Near- und Offshoring-Initiativen stellt sich daher die Frage, wie und mit welchen Daten getestet werden



kann, wenn die Produktionsserver in Deutschland oder der Schweiz stehen, aber in Indien, Rumänien oder den USA getestet wird.

Ist die Motivation für das Test Data Management der Datenschutz, sind „Anonymisierung“ oder „synthetische Testdaten“ mögliche Ansätze. Sie verhindern, dass Tester (und Entwickler) sensible Daten sehen. Anonymisierung klingt einfach und unkompliziert. Das Wort suggeriert, dass aus sensiblen Daten auf Knopfdruck anonyme Daten werden, die man ins Ausland transferieren kann. So einfach ist es nicht. Dazu ein Beispiel aus der Bankenwelt. Hat ein Firmenkunde einen Kredit, speichert die Bank unter anderem Umsatz, Gewinn und Branche. Diese Daten sind wichtig für das Kreditrating des Kunden. Werden bei der „Anonymisierung“ Firmenname, Adresse und Name des CEO gelöscht, reichen trotzdem Umsatz, Gewinn und Branche in der Regel aus, um die Firma zu ermitteln. Ändert man auch diese Daten, wird es vielleicht unmöglich, die Firma zu ermitteln. Doch sicher werden die Daten inkonsistent. Das Testen wird erschwert oder unmöglich. „Anonymisierung“ ist daher eher ein „Verschleiern“. Es reduziert das Risiko, dass grosse Datenmengen entwendet werden. Ob es regulatorische Anforderungen oder das Entwenden der Daten einzelner (prominenter) Kunden verhindert, ist etwas anderes.

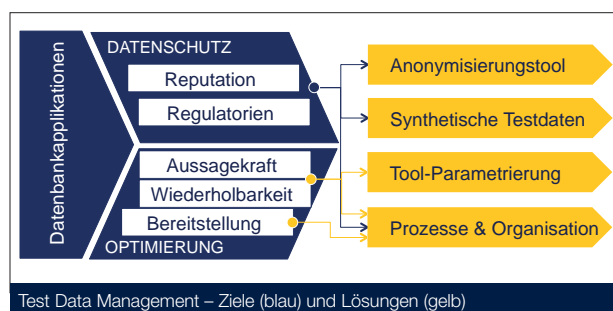
„Echte“ Anonymität gibt es fast nur mit synthetischen Testdaten. Dann werden beim Testen keine Kopien „realer“ Kunden und Konti verwendet, sondern künstliche generierte. Technisch kann das jede Testautomatisierungslösung, so zum Beispiel auch Quick Test Professional oder Selenium. Match-entscheidend ist eine saubere Konzeption – technisch und organisatorisch. Schliesslich kann man dann nur noch Test Cases testen, für die vorher Testdaten vollständig und korrekt generiert wurden.

Die zweite Motivation für Testdatenmanagement-Projekte ist die Optimierung für das Testen von Datenbankapplikationen. Optimiert wird, was bei einer Bankapplikation anders ist als bei einer Taschenrechner-Applikation. Themen sind die Aussagekraft der Tests, ihre Wiederholbarkeit und die Datenbereitstellung.

Die Aussagekraft von Tests meint, dass ein Testfall das testet, wofür er erstellt wurde. Soll der Aktienkauf eines amerikanischen Staatsbürgers mit Wohnsitz in Österreich getestet werden und führt man den Test mit einem deutschen Staatsbürger aus, war der Test umsonst. Daher sind DB-Testing-Objekt und DB-Testsystemzustand beim Test Case-Design mit zu erfassen. Entsprechende (Pflicht-)Felder helfen, diese Informationen strukturiert zu verwalten. Eine clevere Parametrierung von Standardtools wie Quality Center oder JIRA ist notwendig. Über den Erfolg entscheidet weiter, wie gut Test Engineers und Test Analysten geschult und motiviert werden.

Wer in die Aussagekraft von Tests investiert, erreicht auch die Wiederholbarkeit von Tests. Gerade bei Outsourcing und Offshoring verhindern Test Cases mit DB-Testing-Objekt und DB-Testsystemzustand eine zu starke Abhängigkeit vom aktuellen Outsourcing-Partner und fördern die Qualität der Tests. Finden Tester einen Fehler, ist nach dem Bugfix ein identischer Retest notwendig. Ist ein Test Case Bestandteil der Regressionstests, müssen alle Parameter dokumentiert sein. Das umfasst auch den DB-Testsystemzustand und das DB-Testing-Objekt.

Beschäftigt sich eine Testorganisation mit der Frage nach der effizienten Testdatenbereitstellung, hat sie bereits einen hohen Maturitätsgrad. Die Frage ist besonders wichtig, wenn Daten anonymisiert oder synthetisch generiert werden. Dann können Test Engineers und Test Manager Testdaten nicht mehr selbst suchen oder erstellen. Fragen nach einer zentralen Testdatenorganisation oder nach einer Einbettung der Testdatenbestellungen in den Testprozess gewinnen an Bedeutung, ebenso wer Datenbanken anonymisiert und kopiert. Das sind Organisations- und Prozessfragen.



Test Data Management – Ziele (blau) und Lösungen (gelb)

Zum Abschluss steht die Antwort auf die Frage aus der Überschrift aus – ob man in das Test Data Management investieren oder das Thema ignorieren soll. Test Data Management mit einer Tool-Evaluation und Tool-Einführung aufzurollen sollte man sich sehr genau überlegen. Hingegen versprechen folgende Massnahmen in einer komplexen Applikationslandschaft und bei Datenbankapplikationen einen echten Mehrwert:

- Test Data Management aus der Perspektive Prozesse und Organisation angehen.
- Bei Bedarf ein Test Data Management-Team etablieren.
- Das Tooling durch eine angepasste Parametrierung verbessern.

XING Diskutieren Sie mit uns zu diesem Thema in der ASQF-XING-Gruppe unter www.xing.com/net/asqf !

Der Autor

Dr. Klaus Haller arbeitet als Expert im Test-Consulting bei Swisscom IT Services Finance in Zürich mit den Schwerpunkten Organisation und Prozesse, Testdatenmanagement und Testen von Informationssystemen.