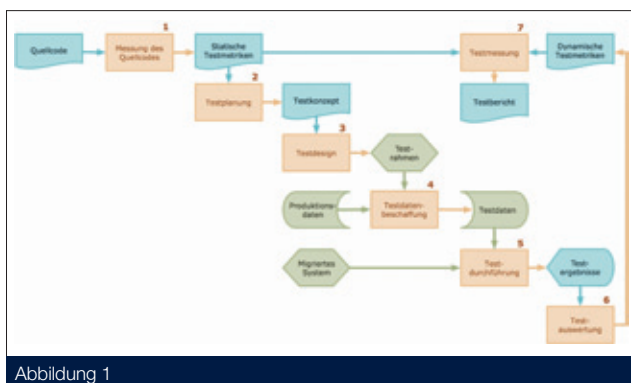


DER MIGRATIONSTEST-PROZESS Harry M. Sneed & Richard Seidl

Kurzbeschreibung: Ein Softwaremigrations-Projekt erfordert einen anderen Testansatz als ein Entwicklungsprojekt. Denn es existiert im Regelfall weder eine Anforderungsspezifikation noch ein Designmodell als Testgrundlage. Das entscheidende Testkriterium ist dann die vollständige funktionale Gleichheit von Alt- und migriertem System – und dafür müssen spezielle Methoden und Werkzeuge zum Einsatz kommen. Darüber hinaus sollte im Migrationstest schrittweise vorgegangen werden:

DIE MESSUNG DES CODES

„Man kann nicht planen, was man nicht messen kann“ [1]. Gemäß dieses Zitates von Tom deMarco muss im Rahmen der Planung eines Migrationstests das gemessen werden, was migriert werden soll: Der Quellcode. Denn in Migrationsprojekten ist dieser Code des Altsystems die entscheidende Grundlage für die Aufwandsschätzung [2]. Welche Aspekte einer Messung unterzogen werden, hängt von der gewünschten Information und Zielgröße ab, die man mit der Messprozedur gewinnen will. Die gewünschten Metriken können anhand statischer Analyse aus dem Quellcode, den Datenbank-Schemata, den Definitionen des Benutzerinterfaces sowie den Systemschnittstellen gewonnen werden [3].



Die Erfahrung vergangener Projekte zeigt, dass der Zeitaufwand für die Messung von Quellcode im Regelfall nicht mehr als 1 Woche beträgt; vorausgesetzt, man setzt dafür die geeigneten Messwerkzeuge ein. Im Laufe der Zeit sind Werkzeuge entstanden, die sich nicht nur für neue, sondern ebenso für alte Programmiersprachen eignen. Denn in den allermeisten Migrationsprojekten ist das Vorgängersystem in einer alten Programmiersprache wie PL/1, Cobol oder C geschrieben.

DIE PLANUNG DES MIGRATIONSTESTS

Zeit- und Aufwandsschätzung für die Testdurchführung sind Bestandteile der Testplanung. Der Testaufwand bemisst sich hier nach der Anzahl an zu exekutierenden Testfällen (Online-Transaktion, Batch-Prozess etc.) sowie jener an zu validierenden Testobjekten. Ein Testobjekt wiederum kann aus den



Produktionsdaten generiert werden: In Form eines Benutzer-Interfaces, einer Datei, einer Datenbank-Tabelle, einer Systemschnittstelle etc. So gelangt man zu einer Rohschätzung für das aktuelle Migrationstest-Projekt. Diese Schätzung kann z.B. mittels Produkttyp oder des Testwiederholungsfaktors verfeinert werden [4].

Das Ergebnis der Testplanung ist ein Testkonzept gemäß ANSI/IEEE-Standard 829. In diesem sollte besonderes Augenmerk auf die Testziele und die Testendkriterien sowie deren Messung gelegt werden. Denn nur wenn diesen beiden Aspekten entsprochen wird, kann der Kunde den Test am Ende abnehmen. Das kann z.B. mittels Kennzahlen über die durchgeführten Testfälle und über die validierten Testobjekte erfolgen. [5].

DAS DESIGN DES MIGRATIONSTESTS

Um einen Migrationstest zu designen, muss man berücksichtigen, was (Testfälle und Testobjekte) und wie (Testvorgehen) zu testen ist. Häufig ist mangels Verfügbarkeit und Aktualität einer Dokumentation des Altsystems die einzige Möglichkeit für die Erstellung eines Testdesigns, dass der Tester aufzeichnet, wie die Endbenutzern das System aktuell bedienen. Bei den Testobjekten ist die Anzahl an Datenelementen, die jedes Datenobjekt besitzt, von besonderer Bedeutung. Dazu zählen z.B. Felder und Spalten sowie deren Wertebereiche.

DIE GEWINNUNG VON TESTDATEN FÜR EIN MIGRATIONSPROJEKT

In einem Migrationsprojekt gewinnt man die Testdaten am besten direkt aus dem Produktionssystem, indem man einen Teil der Produktionsdaten abzieht (und anonymisiert). Im Testdesign sollte man bereits einzelne Transaktionen, Ereignisse und Batchläufe mit repräsentativem Charakter identifiziert haben.

Vor jedem Test müssen Abbildungen des Benutzer-Bedienfeldes, der Eingabedateien und des Datenbankinhaltes ar-

chiviert werden. In der gleichen Form müssen nach jedem Test die neuen Abbildungen derselben Objekte sowie zusätzlich gesendete Ausgabe-Nachrichten und ausgedruckte Reports festgehalten werden. All das führt zu einem großen Aufkommen an Testdaten.

Hier können Testwerkzeuge und Capture-Replay-Tools als nützliche Helfer zum Einsatz kommen [6].

DIE DURCHFÜHRUNG DES MIGRATIONSTESTS

Etwa 60% bis 80% des Gesamtaufwandes werden in Migrationsprojekten dem Test gewidmet. Sofern der Migrationstest also nicht automatisiert durchgeführt wird, ist es unmöglich, ihn innerhalb eines vernünftigen Zeitrahmens zu absolvieren. Damit man im Nachhinein feststellen kann, welcher Teil des migrierten Programmcodes getestet wurde, sollten (z.B. mittels Automatisierungswerkzeugen) Messfühler vor Teststart in z.B. jedem Zweig oder zumindest in jeder Methode platziert werden.

Nach jeder Transaktion oder jedem Batchprozess werden die Inhalte der Ausgabefelder, der Ausgabe-Nachrichten, das neue Erscheinungsbild der betroffenen Datenbank und Anomalien für eine spätere Evaluierung aufgezeichnet [7].

DIE EVALUIERUNG DES MIGRATIONSTESTS

Im sechsten Schritt werden die Ergebnisse des Migrationstests evaluiert, um festzustellen, ob das migrierte System die Erwartungen erfüllt: Nämlich dieselbe Leistung (inkl. Performance), wie das Altsystem zu erbringen.

Zu Validierungszwecken werden die Bildschirminhalte, Schnittstellendaten und Datenbank-Tabellen der Tests aus beiden Systemen automatisch miteinander verglichen. Weichen die Formate voneinander ab, muss zugunsten einer Vergleichbarkeit eine Konvertierung in ein einheitliches Format stattfinden.

Ebenso werden die Inhalte der alten Datenbanken und jene der neuen, relationalen Datenbanken heruntergeladen und dabei zu Vergleichszwecken in komma-separierte (CSV-) Dateien umgewandelt (Abbildung 2).

Im Falle des Vergleichs von Millionen von Datenausgaben ist der automatisierte Vergleich die beste Art und Weise, Fehler im migrierten Code aufzuspüren [8].

DIE MESSUNG IM MIGRATIONSTEST

Die vor der Testausführung automatisch im Code eingepflanzten Messfühler oder Verfolgungspunkte werden während der Testausführung in Log-Dateien aufgezeichnet. Nach dem Testdurchlauf werden sie ausgewertet. Messreports können unterschiedliche Informationen dokumentieren: Den Testabdeckungsgrad, den Pfad jeder einzelnen Testtransaktion etc.

Beim Test eines komplett neu entwickelten Systems treten z.B. folgende Fehler auf: Es werden Funktionen ausgespart, Ergebnisse falsch kalkuliert usw. Beim Test eines migrierten

Systems sind die (Transformations-) Fehler allerdings subtiler Natur: Nachkommastellen gehen verloren, Datensätze werden zerstört, Daten haben sich an einen anderen Ort verlagert usw.

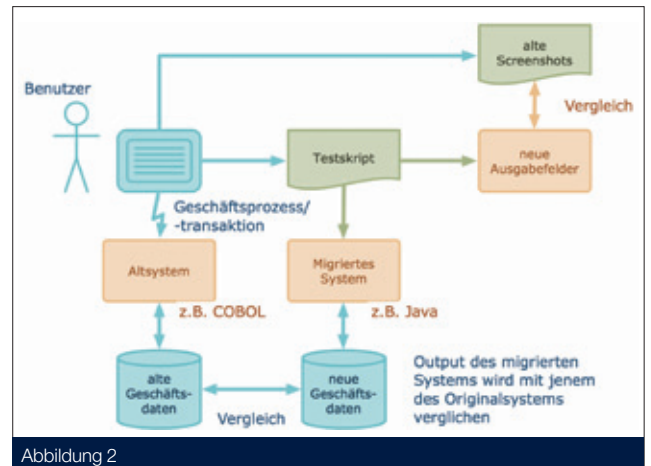


Abbildung 2

Die einzige Möglichkeit zur Aufdeckung solcher Fehler ist es, alle Daten zu vergleichen; kombiniert mit einer Abdeckung des kompletten Programmcodes. Jede Anweisung muss getestet werden – das bedeutet, dass eine wiederholte Durchführung des Tests notwendig ist [9].

Neben der Messung der Testabdeckung ist es erforderlich, die Richtigkeit der Ausgabedaten und den Prozentsatz an tatsächlich aufgetretenen Fehlern relativ zu jenem der vorausgesagten Fehler zu messen. Mit diesen Informationen kann man feststellen, wann der Test beendet werden kann (Abbildung 3).

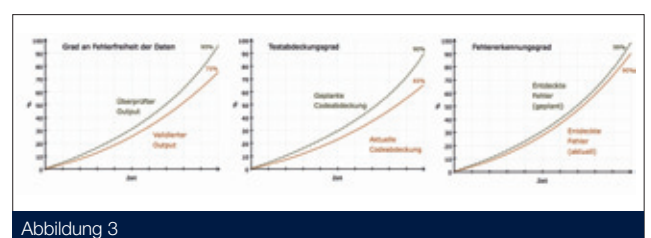


Abbildung 3

Eine ausführliche Version dieses Fachbeitrages sowie das vollständige Literaturverzeichnis finden Sie unter www.asqf.de/publikationen

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

Die Autoren

Richard Seidl ist als Testspezialist und Testmanager bei der ANECON GmbH tätig. Er ist regionaler Fachgruppenleiter der FG Softwaretest Sachsen.

Harry Sneed ist Autor zahlreicher Artikel und Bücher. Er ist Dozent und für die ANECON GmbH tätig.