

ORDIX[®] news

einfach. gut. informiert.



Java EE 7 - Faces Flow

- 6 | Standardisierung der Ablaufsteuerung von Webanwendungen
- 11 | Softwareentwicklung im Wandel der Zeit: Objektorientierung - Quo vadis?
- 16 | Oracle Editionen: Muss es immer Enterprise sein?
- 32 | Ganzheitliches Datenqualitätsmanagement – Warum und wofür?
- 41 | Wechsel des JDK beim WebSphere Application Server 8: Neuer Antrieb für WebSphere

Es ist schön, wenn ein Administrator von seiner Datenbank sagen kann:

Die läuft und läuft und läuft ...

Aber in der Realität sieht es oft anders aus. Wie bei der Aufbereitung von alten Autos sind, auch bei der Optimierung von Datenbanken, Spezialisten gefragt. Hierbei gilt es an den richtigen Schrauben zu drehen um die optimale Performance aus der Datenbank zu holen. Um Sie dabei zu unterstützen haben wir unsere IT-Checks entwickelt.



Mit unserem **IT-Check** zum Performance Tuning behalten Sie den Überblick!

Sollte Ihre Datenbank einmal ins Stocken geraten oder Ihre Anwendung zu langsam reagieren, gilt es zunächst die Ursachen zu erforschen und anschließend die Optimierungspotentiale zu ermitteln. Im Anschluss an unseren IT-Check geben wir Ihnen konkrete Handlungsempfehlungen und kümmern uns auf Ihren Wunsch auch um die Umsetzung.

Unser Vertriebsteam steht Ihnen für alle Fragen zur Verfügung und erstellt gern ein individuelles Angebot. Sprechen Sie uns an! Wir zeigen Ihnen den Mehrwert für Ihr Unternehmen auf.

Kontaktieren Sie uns über: vertrieb@ordix.de

Eine Auswahl unserer IT-Checks finden Sie unter:
www.ordix.de/dienstleistung/ordix-it-checks.html

Wohin mit Deutschland?

Paderborn, März 2014

Nein, ich will nicht ein Preview über die Fußballweltmeisterschaft schreiben. Mir geht es mehr um den IT-Standort Deutschland und um den sieht es recht schlecht aus.

Vor 20 bis 30 Jahren gab es zu dieser Jahreszeit nur ein Thema: CeBit. „C was?“ fragen heute viele. Gut, Messen für nicht oder kaum greifbare Dinge haben sich im Zeitalter des Internets und der grenzenlosen, ständigen Kommunikation über alle möglichen Wege sicher etwas überholt, zumal der Besucher nichts zum Anfassen hat. Hardware (oder besser Smartphone-Neuheiten) konnte man bereits in Barcelona in die Hand nehmen.

Aber nicht nur der Niedergang der CeBit symbolisiert ein ständiges Schrumpfen des so wichtigen Marktes. IT-Produkte sind bei Deutschlands Einkäufern nur smart, wenn sie aus den USA kommen (von SAP vielleicht abgesehen, dafür verlagert SAP aber immer mehr die Entwicklung weg von Deutschland).

IT-Fachkräfte sind nur dann interessant, wenn sie billig sind und aus Indien, Russland oder anderen ehemaligen Staaten des Ostblocks stammen und für Stundensätze arbeiten, mit denen Sie hier noch nicht mal einen Mitarbeiter, der exakt den geplanten Mindestlohn bekommt, finanzieren könnten. Gerne hört man als mittelständisches Unternehmen: „Wir als globales Unternehmen müssen global denken. Sie müssen sich an weltweit tätigen Unternehmen messen lassen.“ Dass dabei meistens nur Kettengeschäfte entstehen, weil sich sogar Unternehmen in Indien oder sonst wo keine Mitarbeiter „auf Halde“ anschaffen, vergessen die meisten.

Jetzt aber hat das Frau Merkel erkannt¹⁾. Ich bin gespannt, was daraus wird bzw. ob und wie sich z.B. das Verhalten bei den vom Staat beherrschten Unternehmen ändern wird (wäre zumindest schon mal eine erste Maßnahme, die Frau Merkel direkt beeinflussen könnte). Aber die Europawahl ist ja bald vorbei, da werden dann sicher andere Themen wieder interessanter oder den Wählern wirk-samer verkauft.

Wir verkaufen Ihnen auch dieses Mal nichts, die ORDIX® news bleibt weiterhin kostenlos (für Sie) im Netz und auf Papier. Leider geizt aber auch unsere Ausgabe nicht mit Artikeln über Produkte, die nicht aus Deutschland stammen 😞.

Runner Up wie immer: Oracle. Mit Artikeln über Lizenzpolitik und neue Funktionalitäten in Oracle 12c. Bei Larrys Geburtstagskalender haben wir allerdings nicht das Smartphone von Larry Ellison geknackt, sondern wir zeigen Ihnen Lösungsansätze der Larry Ratlos Aufgabe aus der Ausgabe 3/2013.

Bleiben wir bei Oracle, aber nun beim „Ableger Java“. Da geht es um die neue Java Version 8 und auch um Erweiterungen bei JavaServer Faces. Mit WebSphere - auch in der Version 8 - liefern wir Ihnen dieses Mal Informationen aus den Laboren der IBM.

Die Artikel zu Softwarearchitektur und Qualitätsmanagement beschreiben Sachverhalte, die Sie auch in Deutschland „erwerben“ können. Auch bei unseren Artikeln im Projektmanagement-Umfeld stehen Leistungen aus Deutschland im Vordergrund: Es geht dabei fast wie immer um Organisation und Kommunikation.

Last but not Least noch mal was aus dem Hause Oracle: Solaris lebt! Und mit ihm neue Funktionalitäten rund um Systemdienste.

Sollten Sie jetzt in der Fastenzeit auf Alkohol verzichten wollen, hilft Ihnen mein letzter Tipp wenig: Probieren Sie doch mal wieder Krimsekt, zur Zeit sicher das Beste, was aus der Region kommt.

Viel Spaß beim Lesen wünscht Ihnen



Wolfgang Kögler

¹⁾ Focus online-Artikel: Europa soll Rückstand aufholen - Merkel fordert „absoluten Willen“ zur IT-Aufholjagd http://www.focus.de/finanzen/news/europa-soll-rueckstand-aufholen-merkel-fordert-absoluten-willen-zur-it-aufholjagd_id_3653529.html





Java EE 7 - Faces Flow

Datenbanken

- 16..... Oracle-Editionen:
Muss es immer Enterprise sein?
Ein wichtiges Kriterium für die Entscheidung, welche Datenbank im Unternehmen eingesetzt wird, sind die Lizenzkosten. Oracle hat, wie viele andere Hersteller, ein kaum zu durchschauendes Lizenzmodell. Wir erläutern Ihnen exemplarisch die Kosten der Standard und Enterprise Edition.
- 22 Neuerungen in der Oracle Database 12c (Teil III):
SQL-Neuerungen in Oracle 12c
Mit der neuen Version 12c bringt Oracle auch neue SQL- und weiterentwickelte Funktionen heraus. Unser Autor gibt Ihnen einen Überblick.
- 31..... Kalenderberechnung in Oracle-Datenbanken:
Larry und sein Geburtstagskalender
Nachdem Larry in der Ausgabe 3/2013 einige Probleme bei der Kalenderberechnung in Oracle-Datenbanken hatte, zeigen wir Ihnen hier 3 Lösungsansätze um das korrekte Ergebnis zu erhalten.

Web und Application Server

- 41..... Wechsel des JDK beim WebSphere
Application Server 8: Neuer Antrieb für WebSphere
Kritische Security Updates sind meist ein zeitintensives Unterfangen für viele Administratoren. Welche Möglichkeiten der WebSphere Application Server 8 Ihnen hierbei bietet, beschreiben wir im vorliegenden Artikel.



Oracle-Editionen: Muss es immer Enterprise sein?

Softwarearchitektur

- 11..... Softwareentwicklung im Wandel der Zeit:
Objektorientierung - Quo vadis?
Eine effiziente Softwareentwicklung ist ohne Objektorientierung nicht mehr denkbar. Wir werfen einen Blick in die Vergangenheit und in die Zukunft der objektorientierten Softwareentwicklung.
- 44 Qualitätssicherung in Softwareprojekten (Teil I):
QS - Was? Wann? Wer?
Das Testen von Softwareprojekten ist nicht der Garant für die Qualität der Software. Bei der Qualitätssicherung spielen viele Faktoren eine Rolle. Wir beleuchten im ersten Teil dieser neuen Reihe zunächst die wichtigsten Fragestellungen, die schon zu Beginn berücksichtigt werden müssen.

Java/JEE

- 6 Standardisierung der Ablaufsteuerung
von Webanwendungen:
Java EE 7 - Faces Flows
Die Erweiterung zu JavaServer Faces ermöglicht es, Webanwendungen aus Dialogsequenzen modular zusammenzustellen. Wir liefern Ihnen einen Überblick über die Möglichkeiten des neuen Standards.
- 38 Java 8 - Die neue Version (Teil I):
Ein erster Überblick
Mit diesem Artikel starten wir eine Reihe zu der neuen Version von Java. Wir geben Ihnen einen ersten Überblick über die neuen Funktionen, die wir dann in zukünftigen Artikeln vertiefen werden.



Ganzheitliches Datenqualitätsmanagement

Projekt-/IT-Management

28 Scrum - aber richtig! (Teil III):
Äußere Faktoren und Integration

Der Erfolg agiler Projektmanagement-Methoden hängt stark davon ab, wie sich die eingesetzten Methoden - in diesem Fall Scrum - in eine Organisation einbinden lassen und sich etablieren können.

32 Ganzheitliches Datenqualitätsmanagement –
Warum und wofür?

Kennen Sie das nicht auch? Stammdaten sind unvollständig, doppelt angelegt oder nicht mehr aktuell. Wie wichtig das Qualitätsmanagement gerade im Bereich der Stammdaten ist, erläutern wir in diesem Artikel.

48 Kommunikation im Projekt:
Mit Plan und Persönlichkeit

Kommuniziere ich richtig? Versteht mein Kollege im Projekt was ich meine? Diese Fragen beantwortet der vorliegende Artikel. Damit Sie sich zukünftig in Ihrem Projekt richtig verständigen.

Betriebssysteme und Monitoring

19 Neue Funktion svcbundle von Solaris 11:
Service bündeln leicht gemacht

Mit diesem Artikel stellen wir Ihnen das komfortable Werkzeug svcbundle vor, mit dem Sie Systemdienste einfach einbinden können.

Aktuell

26 Seminarübersicht: März bis Juni 2014

51 Larry Ratlos Rätsel



Kommunikation im Projekt

Impressum

Herausgeber: ORDIX AG Aktiengesellschaft für Softwareentwicklung, Beratung, Schulung und Systemintegration, Paderborn

Redaktion: Jens Pothmann, Evelyn Ernst

V.i.S.d.P.: Benedikt Georgi, Wolfgang Kögler

Anschrift der Redaktion: ORDIX AG | Westernmauer 12 - 16 | 33098 Paderborn
Tel.: 05251 1063-0 | Fax: 0180 1673490

Gestaltung/Layout: Jens Pothmann

Auflage: 7.200 Exemplare

Druck: Druckerei Bösmann, Detmold

Bildnachweis: © sxc.hu | stone statue against the sky | farangio
© piqs.de | Ithaca Falls | Adam Baker
© flickr.com | Volkswagen Beetle | BKM_BR
© flickr.com | The Wizard | Sean McGrath
© shutterstock | attractive man close up portrait on white background | Leonid and Anna Dedukh
© freepik.com | Infographic free vector

Autoren: Andre Dirr, Andreas Flügge, Matthias Heintl, Martin Hoermann, Dr. Stefan Koch, Wolfgang Kögler, Maik Krawinkel, Marcus Meisenberg, Roger Niemeyer, Klaus Reimers, Thomas Rohde, Michael Steinert, Sascha Slomka, Tobias Ummler, Irini Xanthopoulou

Copyright: Die ORDIX® news erscheint vierteljährlich. Alle Eigentums- und Nachdruckrechte, auch die der Übersetzung, der Vervielfältigung der Artikel oder von Teilen daraus, sind nur mit schriftlicher Zustimmung der ORDIX AG gestattet.

Warenzeichen: Einige der aufgeführten Bezeichnungen sind eingetragene Warenzeichen ihrer jeweiligen Inhaber. ORDIX® ist eine registrierte Marke der ORDIX AG.

Haftung: Eine Haftung für die Richtigkeit der Veröffentlichungen kann trotz sorgfältiger Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden.

Sie können die Zusendung der ORDIX® news jederzeit ohne Angabe von Gründen schriftlich (z.B. Brief, Fax, E-Mail) abbestellen.



Standardisierung der Ablaufsteuerung von Webanwendungen

Java EE 7 – Faces Flows

Faces Flows ist eine Erweiterung zu JavaServer Faces. Mit Faces Flows lassen sich Webanwendungen aus Flows (Dialogsequenzen) modular zusammenstellen. Die Steuerung dieser Dialogsequenzen ist konfigurierbar. Der vorliegende Artikel gibt einen exemplarischen Einstieg in den neuen Standard und einen Überblick über dessen Möglichkeiten.

Wieso noch mehr standardisieren?

Das Bestreben eines Standards ist es nach Ed Burns, „[...] die besten Ideen aus der aktuellen Praxis der Entwicklung aufzunehmen und sie sauber in den Rest der Spezifikation einfließen zu lassen.“ Seit Jahren gibt es Frameworks - Spring Web Flow, Apache MyFaces CODI, ADF Taskflows - für die Modularisierung von Webanwendungen. Besonders für Internetanwendungen, in denen eine Vielzahl von Dialogen zum Einsatz kommt, ist die Verwaltung der Dialoge eine separate Aufgabe. Dafür erscheint es sinnvoll, eine getrennte Komponente einzusetzen. Vorbild für diesen Standard ist das von Oracle entwickelte Oracle Application Development Framework (Oracle ADF).

Das Beispiel

In diesem Artikel wird Faces Flows eingesetzt, um einen Teil einer Webanwendung zu realisieren. Dabei geht es fachlich darum, Teilnehmerdaten für die Reservierung für

ein Seminar zu erfassen (siehe Abbildung 1). Zur Reservierung eines Seminars ist eine Einstiegsseite vorgesehen. Danach wird mit „Flow steuern“ ein Knoten angesprochen, der in Abhängigkeit von den verfügbaren Daten bestimmt, welche Dialoge der Anwender angezeigt bekommt.

In dem Beispiel dient die Steuerung nur der Ermittlung einer einzigen Größe: der Teilnehmer-ID. Ist diese noch nicht vorhanden, so sieht der Anwender den Dialog „Teilnehmer erfassen“. Dort gibt er entweder eine vorhandene Teilnehmer-ID ein oder registriert einen neuen Teilnehmer. Für die Registrierung geht die Steuerung in den Teilnehmer-Flow über. Dort werden Personendaten erfasst und eine Teilnehmer-ID ermittelt. Diese ID wird dem aufrufenden Flow „Reservieren“ zurückgegeben. Nach der Einstiegsseite wird dem Anwender der Dialog „Reservierung bestätigen“ angezeigt.

Die Webanwendung besteht aus den Flows „Reservieren“ und „Teilnehmer“. Daran lässt sich demonstrieren, wie die

Dialogsteuerung innerhalb eines Flow konfiguriert und ein anderer Flow mit der Übergabe von Daten aufgerufen werden kann.

Konfiguration durch Konvention

Faces Flows verfolgt das Prinzip der Konfiguration durch Konvention. Wenn Sie die folgenden Konventionen einhalten, können Sie komplett auf eine Konfiguration verzichten:

- Alle View-Definition-Language-Dateien (xhtml-Dateien) befinden sich in einem Verzeichnis direkt unterhalb von **WebContent**. Der Verzeichnisname stimmt mit dem Namen des Flow überein.
- Der Start des Flow wird durch den Dialog vorgegeben, der durch die Datei `<Flow-Name>.xhtml` beschrieben wird.
- Es gibt eine leere Konfigurationsdatei im Flow-Verzeichnis mit dem Namen `<Flow-Name>-flow.xml`.
- Durch den Dialog `<Flow-Name>-return.xhtml` wird der Flow beendet. Diese Datei liegt im Verzeichnis **WebContent**.

In diesem Fall erstellt Faces Flows automatisch eine voll funktionsfähige Standardkonfiguration. Damit können Sie den Flow von außen über den Flow-Namen aufrufen (siehe Abbildung 2). Solange der Anwender sich in dem Flow befindet, steht eine Managed Bean zur Verfügung, deren Lebenszeit sich auf den Flow bezieht. Sie wird erzeugt, wenn der Anwender den Flow betritt und freigegeben, sobald die Return-Seite aufgerufen wird.

Die Deklaration einer sogenannten Flow-Scoped Managed Bean ist in Abbildung 3 dargestellt. Durch diesen neuen Scope erspart man sich die Erzeugung von Conversation-Scoped Beans. Die eigentliche Navigation in dieser Standardkonfiguration erfolgt in gewohnter Weise: die Rückgabe (Outcome) der Action-Methode oder aber das Navigationsziel in der View definieren die Seite, die der Anwender als nächstes zu sehen bekommt.

Konfiguration der Flusssteuerung

Neben der Modularisierung eines Flow stellen die Steuerung des Ablaufs, die Übergabe von Parametern und der Aufruf anderer Flows wesentliche Eigenschaften von Faces Flows dar. Diese Eigenschaften werden im Flow `reservierenFlow` verwendet.

Dafür ist eine Konfigurationsdatei erforderlich (siehe Abbildung 4). Dort werden View-Knoten mit dem Element `view` für Dialoge, ein Switch- und ein Flow-Call-Knoten mit den gleichnamigen Elementen definiert. Die Knoten werden über ihren Identifier innerhalb der Konfiguration adressiert. Sie können aber auch als Navigationsziel aus den Dialogen des Flow angesprochen werden.

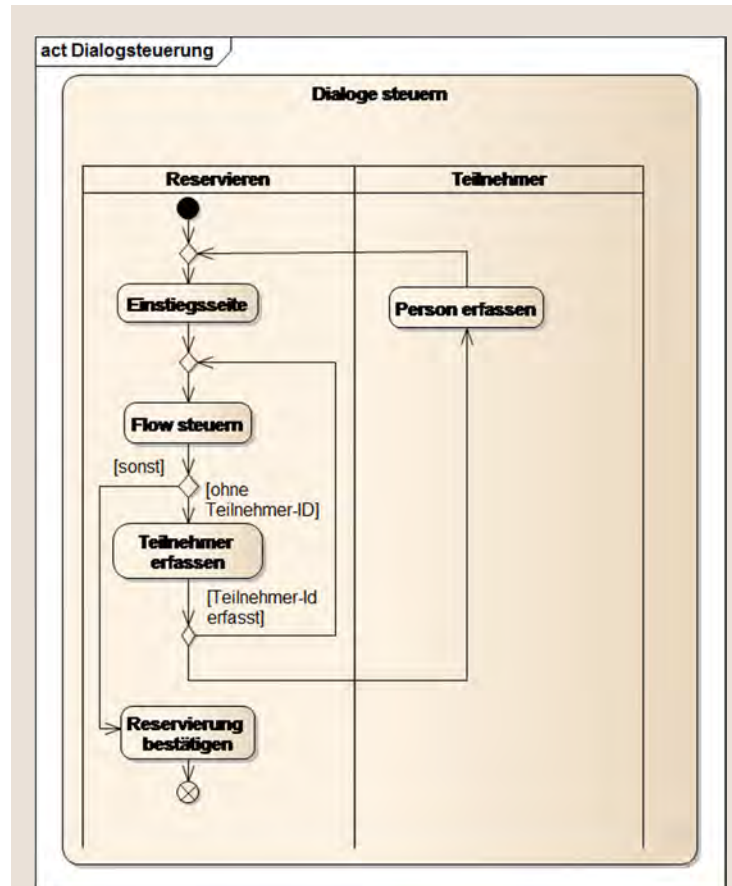


Abb. 1: Ablauf der Dialoge als Aktivitätsdiagramm.

```
<h:commandButton value="Seminar reservieren"
action="reservierenFlow"/>
```

Abb. 2: Aufruf des Flow `reservierenFlow`.

```
import javax.faces.flow.FlowScoped;
import javax.inject.Named;

@Named("reservierenBean")
@FlowScoped("reservierenFlow")
public class Reservieren implements Serializable{
```

Abb. 3: Deklaration einer Flow-Scoped Managed Bean.

Der Switch-Knoten bestimmt einen anzuzeigenden Dialog nach dem Prinzip einer Kaskade, in der von oben nach unten die Bedingungen geprüft werden. Die erste erfüllte Bedingung definiert das Navigationsziel, welches im Element `from-outcome` enthalten ist. Wenn die Teilnehmer-ID noch nicht festgelegt ist, so wird der Dialog „Teilnehmer erfassen“ angezeigt. Andernfalls sieht der Anwender den Dialog „Reservierung bestätigen“.

```

<?xml version='1.0' encoding='UTF-8'?>
<faces-config version="2.2" xmlns="http://xmlns.jcp.org/
xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-facesconfig_2_2.xsd">
<flow-definition id="reservierenFlow">
  <start-node>gutGeschult</start-node>
  <view id="gutGeschult">
    <vdl-document>
  /reservierenFlow/reservierenFlow.xhtml</vdl-document>
  </view>

  <view id="teilnehmerErfassen">
    <vdl-document>
  /reservierenFlow/teilnehmerErfassen.xhtml
  </vdl-document>
  </view>
  <view id="callSeminarFlow">
    <vdl-document>
  /reservierenFlow/callSeminarFlow.xhtml
  </vdl-document>
  </view>
  <view id="reservierungBestaetigen">
    <vdl-document>
  /reservierenFlow/reservierungBestaetigen.xhtml
  </vdl-document>
  </view>

  <switch id="flowSteuern">
    <case>
      <if>
        #{reservierenBean.teilnehmerId == 0}
      </if>
      <from-outcome>teilnehmerErfassen</from-outcome>
    </case>
    <default-outcome>
      reservierungBestaetigen
    </default-outcome>
  </switch>

  <flow-call id="callTeilnehmerFlow">
    <flow-reference>
      <flow-id>teilnehmerFlow</flow-id>
    </flow-reference>
    <outbound-parameter>
      <name>name</name>
      <value>#{reservierenBean.name}</value>
    </outbound-parameter>
    <outbound-parameter>
      <name>vorname</name>
      <value>#{reservierenBean.vorname}</value>
    </outbound-parameter>
  </flow-call>

  <inbound-parameter>
    <name>teilnehmerId</name>
    <value>#{reservierenBean.teilnehmerId}</value>
  </inbound-parameter>
</flow-definition>
</faces-config>

```

Abb. 4: Konfigurationsdatei zum Flow `reservierenFlow-flow.xml`.

Hat der Anwender keine Teilnehmer-ID, so kann er die Schaltfläche „Registrieren“ betätigen. Dadurch wird der Flow-Call-Knoten `callTeilnehmerFlow` angesprochen. Der Aufruf eines weiteren Flow ist wie der Aufruf einer Subroutine zu sehen: Die Managed Bean des aufrufenden Flow bleibt erhalten. Es wird davon ausgegangen, dass der Anwender zum aufrufenden Flow zurückkehrt. Dem aufgerufenen Flow können Parameter (Outbound-Parameter) mitgegeben werden. Diese Parameter werden vom aufgerufenen Flow durch die Konfiguration gleichnamiger Inbound-Parameter entgegengenommen.

Weitere Möglichkeiten

Neben der Konfiguration der Flusststeuerung durch einen Switch-Knoten stellt Faces Flows noch Navigation-Case- und Method-Call-Knoten zur Verfügung. Die Steuerung erfolgt analog zur `faces-config.xml` über Navigation-Case-Knoten. Anhand des Rückgabewerts von Action-Methoden werden unter Einbeziehung möglicher Bedingungen weitere Navigationsziele bestimmt.

Sind fachliche Kriterien für die weitere Navigation maßgeblich, bietet sich der Einsatz von Method-Call-Knoten an. Dabei wird eine Bean-Methode aufgerufen, deren Rückgabe das nächste Navigationsziel bestimmt.

Neben der Konfigurationsdatei ist es möglich, die Konfiguration auch klassenbasiert durchzuführen (siehe Abbildung 5). Die klassenbasierte Konfiguration wird in dem Setup mit WildFly 8 nur dann interpretiert, wenn sie mit der Annotation `@ApplicationScoped` versehen ist.

Zur Konfiguration wird in diesem Beispiel die Methode `defineFlow` verwendet. Nicht der Name, sondern vielmehr die Signatur der Methode und die Annotationen sind dabei entscheidend: Die Methode liefert ein Objekt vom Typ `Flow` zurück. Das Objekt enthält die Flow-Konfiguration.

Vergleich mit Alternativen

Faces Flows ist angetreten, die Steuerung von Dialogen zu vereinfachen. Dabei werden zwei Aspekte aufgegriffen: Navigation und Bean-Management. Diese Anforderung haben bereits alternative Frameworks wie beispielsweise Spring Web Flow [4, 5] implementiert.

Bei einem Vergleich fällt auf, dass die Übergabe von Daten beim Spring Web Flow eleganter ist: Hier hat der Subflow Zugriff auf die Bean des übergeordneten Flow. Daten können so nach Bedarf ausgetauscht werden. Eine Konfiguration ist nicht erforderlich. Auch Ajax-Requests werden von Spring Web Flow berücksichtigt, bei Faces Flows sind diese nicht vorgesehen.

Einen weiteren Aspekt stellt die Navigation mit Hilfe der Browser-History dar. Verwendet der Anwender die Schalt-

flächen des Web-Browser zum Blättern, so bekommt er ohne weitere Vorkehrungen die Seiten so zu sehen, wie er sie verlassen hat. Diese Art der Navigation lädt die Seiten aus dem Speicher des Browser. Der Server - und damit die Anwendung - erfährt nichts davon, dass der Anwender eine andere Seite lädt. Die Lösung für diesen Fall wird durch das Design Pattern Post Redirect Get (PRG) gelöst. Während Faces Flows die Implementierung dem Entwickler überlässt, kümmert sich Spring Web Flow um diesen Aspekt.

Eine weitere Alternative zu der Erweiterung Faces Flows stellt der Conversation Scope von Delta Spike [6], ehemals Seam, dar. Das Bean-Management kann in diesem Framework ausschließlich durch Annotations konfiguriert werden. Dieser Aspekt ist getrennt von den Möglichkeiten der Navigation.

Erste Erfahrungen

Das Beispiel wurde mit WildFly 8 (Version CR 1) durchgeführt. Dieser nutzt die Referenz-Implementierung Mojarra [3] für JavaServer Faces. Wird auf jegliche Konfiguration verzichtet, so können einzelne Flows problemlos modular aufgebaut werden.

Für Abweichungen von der Standardkonfiguration, alternativ kann diese auch im XML- oder Klassenformat erfolgen, ist ein nicht unerhebliches Maß an Frustrationstoleranz erforderlich: Fehler, mit denen insbesondere bei einer XML-Konfiguration zu rechnen ist, bleiben unprotokolliert - betroffene Knoten werden im Fehlerfall einfach nicht gebildet. Es ist davon auszugehen, dass sich neuere Versionen oder alternative Implementierungen von Faces Flows [2] solcher Probleme annehmen werden.

Die Erwartung, dass jede Art von Knoten gleichwertig als Navigationsziel verwendet werden kann, wird nicht erfüllt. Für das dargestellte Beispiel wäre es deutlich sinnvoller gewesen, den Switch-Knoten als Start-Knoten zu definieren. Dieses ist zwar möglich, doch steht dann die Flow-Scoped Bean beim Eintritt in den Flow nicht zur Verfügung.

Wenn Daten bei einem Flow-Call mitgegeben werden können, liegt die Erwartung nahe, dass auch Daten von einer Flow-Scoped Bean zurückgegeben werden können. Eine diesbezügliche Lösung wird aber nicht angeboten. Zurzeit finden sich nur wenige praktische Beispiele im Internet. Wie sinnvoll mit diesem Standard umzugehen ist, muss jedes Projekt für sich entscheiden.

Fazit

Die Aufnahme von Faces Flows in den Java-Enterprise-Standard ist ein gewichtiges Argument für den Einsatz dieses Framework. Es bietet eine Reihe von Vorzügen, wenn Sie eine Vielzahl von Dialogen logisch gruppieren und steuern müssen.

```
@ApplicationScoped
class FlowConfiguration implements Serializable {

    @Produces
    @FlowDefinition
    public Flow defineFlow(
        @FlowBuilderParameter FlowBuilder flowBuilder) {
        String flowId = "withMethodCall";
        flowBuilder.id("", flowId);
        flowBuilder
            .viewNode(flowId, "/" + flowId + "/" + flowId
                + ".xhtml")
            .markAsStartNode();
        flowBuilder.methodCallNode("navigationMethod")
            .expression(
                "#{withMethodCallBean.determineNavigation}");
        flowBuilder
            .returnNode("return")
            .fromOutcome("/" + flowId + "/ return.xhtml");
        return flowBuilder.getFlow();
    }
}
```

Abb. 5: Klassenbasierte Konfiguration des Flow withMethodCall.

Glossar

Flow

(engl. Fluss oder Strom) Im Zusammenhang mit Webanwendungen bezeichnet der Flow die Abfolge von Dialogen.

Java EE

Java Enterprise Edition

JSF

JavaServer Faces ist ein Framework für Webanwendungen, welches in den Standard der Java EE aufgenommen worden ist.

Managed Bean

Eine Managed Bean ist ein Java-Objekt, das die dynamischen Daten und Funktionen eines Dialoges auf der Server-Seite zur Verfügung stellt.

View-Definition-Language-Datei

Datei für die Gestaltung eines Webdialoges. Im aktuellen Standard handelt es sich um xhtml-Dateien.

Quelle

- ▶ Burns, Ed:
Wenn die „ADF Task Flows“ in JavaServer Faces einfließen, heißen sie „Faces Flows“ – Grundlagen eines neuen Features von JSF 2.2, [14.01.2014]
<http://www.doag.org/home/aktuelle-news/article/wenn-die-adf-task-flows-in-javaserver-faces-einfluessen-heissen-sie-faces-flows-gru.html>

Links

- ▶ [1] JSF 2.2-Spezifikation:
http://download.oracle.com/otndocs/jcp/jsf-2_2-fr-spec/index.html
[12.01.2014]
- ▶ [2] JSF 2.2-Implementierung:
myfaces.apache.org
- ▶ [3] JSF 2.2-Implementierung:
jaserverfaces.java.net/
- ▶ [4] Spring Web Flow, Homepage:
<http://projects.spring.io/spring-webflow/>
- ▶ [5] Conversation in DeltaSpike:
<http://deltaspike.apache.org/jsf.html#grouped-conversations>
- ▶ [6] Seminarempfehlung: Java EE 7 Neuheiten
<http://training.ordix.de/siteengine/action/load/kategorie/Java-JEE/nr/1637/index.html>

Bildnachweis

- ▶ © sxc.hu | stone statue against the sky | farangio
- ▶ © piqs.de | Ithaca Falls | Adam Baker

Es ist zu erwarten, dass der Umfang von Faces Flows in den folgenden Versionen entsprechend den Wünschen der Entwickler nachgerüstet wird. Zum jetzigen Zeitpunkt muss ein Projekt abwägen, ob dieser Standard bereits genügend Vorteile bietet oder eine individuelle Lösung bzw. ein alternatives Framework die bessere Wahl darstellt.

Gern führen wir Sie in unseren Workshops in dieses Thema ein, unterstützen Sie bei der Entscheidungsfindung und helfen Ihnen bei der Implementierung Ihrer Lösung.



*Dr. Stefan Koch
(info@ordix.de)*

Buchempfehlung: WildFly 8 - Das Handbuch für Entwickler und Admins

NEU



Mit WildFly 8 bringt JBoss einen frei verfügbaren Application Server für den Standard Java EE 7.0 heraus.

Das Handbuch zu WildFly 8 behandelt einerseits administrative Themen von der Installation bis hin zur Hochverfügbarkeitslösung. Andererseits lernt der Entwickler Grundlagen und praxisbezogene Lösungsansätze zu Java EE 7 kennen. Das Handbuch ist ideal für Ihr Java-Enterprise-Projekt.

WildFly 8 - Das Handbuch für Entwickler, Administratoren und Software-Architekten

Autor: Dr. Stefan Koch, ORDIX AG
Verlag: Galileo Computing/Galileo Press

Softwareentwicklung im Wandel der Zeit

Objektorientierung - Quo vadis?

Programmierparadigmen beschreiben Prinzipien, durch deren Anwendung sich softwaretechnische Probleme möglichst effizient lösen lassen sollen. Die Umsetzung dieser Prinzipien im Design einer Programmiersprache ist maßgeblich für deren Leistungsfähigkeit und bestimmt die Struktur und Architektur der Software. Dieser Artikel untersucht, wie es aktuell um das Paradigma der Objektorientierung bestellt ist und was sich in Zukunft verändern könnte.

Vergangenheit

Die Geschichte der Informatik ist noch nicht so alt wie die anderer klassischer Ingenieursdisziplinen. Die Ära der programmierbaren Rechner begann erst in den 40er Jahren des vorigen Jahrhunderts. Die ersten in Serie produzierten Mikroprozessoren wurden sogar erst zu Beginn der 1970er Jahre auf den Markt gebracht. Die Programmierung dieser Prozessoren erfolgte in einer jeweils spezifischen Maschinensprache.

Paradigmenwechsel im Bereich der Programmierung sind bis heute recht seltene Ereignisse. Auslöser dafür sind in der Regel eklatante Mängel oder Probleme, die mit den vorherrschenden Methoden nicht (oder zumindest nicht effizient) gelöst werden können.

Ein solcher Mangel war in der Anfangszeit der Mikroprozessoren die schwer verständliche und mühsame Programmierung in Maschinensprache. Zwar gab es für die einzelnen Prozessoren entwickelte symbolische Assembler-Sprachen, die den Vorteil hatten vom binären Code zu abstrahieren, aber eine einheitliche Sprache für verschiedene Prozessoren existierte nicht. Dies führte zur Entwicklung von Hochsprachen, die mittels eines Compiler in den entsprechenden Maschinencode übersetzt werden konnten.

Auch wenn es vereinzelte Exoten gab, so folgten die meisten in dieser Ära eingesetzten Sprachen wie C, FORTRAN, Basic, Cobol oder Pascal dem zu diesem Zeitpunkt vorherrschenden imperativen Programmierparadigma. Dieses beschreibt ein Programm als eine Abfolge von Befehlen (lat. imperare = befehlen), die der Reihe nach abgearbeitet werden. Im Fokus steht klar der Programmfluss, zu dessen Steuerung Mittel wie Schleifen und Verzweigungen zur Verfügung standen. Die Daten wurden größtenteils in unstrukturierten Variablen gehalten.

Innovationen

Spätestens zu Beginn der 1980er Jahre wurden Computer zu einem Massenphänomen. Über die Personal- und Heimcomputer wie z.B. den IBM PC oder den Commodore 64

gelangten leistungsfähige Rechner in Büros und private Haushalte. Die Spirale aus Anwenderwünschen und rasant fortschreitender Technologie, schien in allen Bereichen keine Grenzen zu kennen und drehte sich immer schneller.

Prozessortakt, Hauptspeicher, Festplattenkapazitäten und Bandbreiten wuchsen fast zeitgleich zu den Anforderungen. So waren Performance-Probleme oft nur temporär, da diesbezügliche Defizite in der Software häufig mit der nächsten Hardwaregeneration relativiert wurden.

Die deutlich leistungsfähigeren Rechner ermöglichten es natürlich auch, komplexere Software zu implementieren. Wie bereits erwähnt, bedarf es für einen Paradigmenwechsel einen Mangel an den bestehenden Vorgehensweisen. Die im Zuge der Technologieentwicklungen immer komplexer werdenden Probleme bereiteten in der Implementierung mit imperativen Programmiersprachen immer größere Schwierigkeiten und zeigten damit einen solchen eklatanten Mangel auf.

Insbesondere die fehlenden Strukturierungs- und Abstraktionsmöglichkeiten machten die Softwareentwicklung genauso aufwändig wie fehleranfällig und die fertige Software wenig wartungsfreundlich.

Objektorientierte Programmierung (OOP)

Obwohl objektorientierte Sprachen wie Algol, Simula 67 und vor allem Smalltalk bereits in den 60er und 70er Jahren des vorigen Jahrhunderts entwickelt wurden, setzte sich das objektorientierte Paradigma aus o.g. Gründen erst zu Beginn der 90er Jahre durch. Die zugrundeliegende Idee, die Objekte der Realität im Modell bzw. der Architektur der Software abzubilden, führte einige wichtige neue Konzepte ein:

- **Klasse/Objekt**
Jedes Objekt ist eine Instanz (konkrete Ausprägung) einer Klasse. Eine Klasse ist eine Art Vorlage oder Schablone, welche die grundlegende Datenstruktur und das Verhalten beschreibt (Daten und Methoden).

- **Kapselung**
Jede Klasse/Objekt verbirgt interne Details (Struktur/Implementierung) und stellt nach außen nur Informationen zur Verfügung (öffentliche Schnittstelle), die andere Objekte zur Kommunikation benötigen.
- **Messaging**
Objekte kommunizieren miteinander, indem sie Nachrichten austauschen und darauf mit Statusänderungen reagieren (ähnlich eines Funktions- oder Prozeduraufrufs).
- **Vererbung**
Ein Klasse kann von einer anderen Klasse (Oberklasse) abgeleitet werden. Sie erbt alle Eigenschaften (Daten und Methoden) der Oberklasse und kann diese wiederverwenden.
- **Polymorphie**
Die Fähigkeit verschiedener Objekte auf die gleiche Nachricht unterschiedlich zu reagieren. Die Entscheidung wie ein Objekt auf eine Nachricht reagiert, wird erst zur Laufzeit entschieden (späte Bindung).

Eine Folge der Einführung dieser Konzepte war die Verschiebung des Fokus von der reinen Ablaufkontrolle hin zu einer eher datenzentrierten Sicht. Die Steuerung des Programmlaufes verschwand in den Kommunikationsstrukturen der Objekte.

Die Prinzipien der objektorientierten Programmierung in Verbindung mit den dazu entwickelten Sprachen wie u.a. C++ und Java brachten die Softwareentwicklung ein großes

Stück voran. Auch wenn sich nicht alle Hoffnungen erfüllten, wie z.B. im Bereich der Wiederverwendung und Modularisierung, standen den Entwicklern nun ausreichend Mittel zur Verfügung, um auch größere Softwaresysteme strukturiert zu erstellen und wartbar zu halten. Hinzu kamen die etwa zur gleichen Zeit u.a. von Erich Gamma („Gang of Four“) veröffentlichten Arbeiten zum Thema Design-Patterns, die sich optimal ergänzten.

Physikalische Grenzen

Bis in die Mitte des vergangenen Jahrzehnts war der bereits beschriebene technologische Fortschritt intakt. Ab diesem Zeitpunkt verlangsamte sich jedoch die Innovationsgeschwindigkeit deutlich. Mit Erreichen der physikalischen Grenzen - die Größen der Strukturen bei Mikroprozessoren und magnetischen Speichersystemen hatten mittlerweile jene nahe der atomaren Struktur erreicht - setzte u.a. eine Stagnation bei den Taktfrequenzen der Prozessoren und den Kapazitäten magnetischer Speichermedien ein.

Da die Rechenleistung eines einzelnen Prozessors somit nicht mehr im gleichen Maße wie früher steigerbar war, setzten die Hersteller zunehmend auf Multicore-CPUs, deren einzelne Kerne echt parallel arbeiteten.

Hinzu kam ein stark verändertes Nutzerverhalten. Die Entwicklung weg von stationären Rechnern, hin zu kleinen, mobilen Geräten mit ständiger Online-Verbindung führten zu größeren Bandbreiten und damit zu einem explodierenden Datenaufkommen, das zunehmend auch einer verteilten Speicherung bedurfte.

Parallelität und Verteilung

Eine der größten Herausforderungen der Softwareentwicklung in der heutigen Zeit ist es deshalb, Programme möglichst effizient echt parallel und verteilt rechnen zu lassen. Die der Aufgabe innewohnende Problematik der Koordination paralleler Berechnungen ist schon lange bekannt.

Bereits 1967 formulierte Gene Amdahl das nach ihm benannte Gesetz, das in der Abbildung 1 illustriert wird. Demnach kann mit steigender Anzahl von Prozessoren nur dann effizient gerechnet werden, wenn der Anteil des echt parallelisierbaren Codes sehr hoch (mind. 90 %) ist.

Parallelität und Objektorientierung

Damit drängt sich die Frage auf, wie gut sich die Prinzipien des objektorientierten Paradigmas mit den Anforderungen nach massiver Parallelität vereinbaren lassen. Um diese Frage zu beantworten, ist insbesondere der Aspekt der Kapselung genauer zu untersuchen.

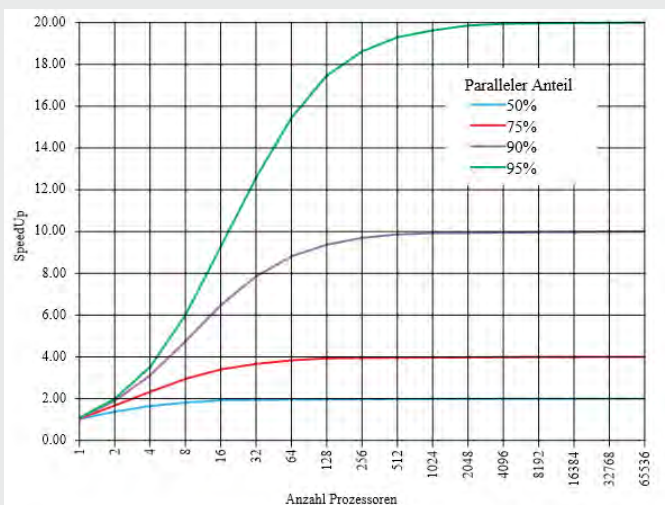


Abb. 1: Amdahlsches Gesetz.

Objekte sind eindeutige (mit einer ID versehene) Instanzen mit Daten und Verhalten. Jedes Objekt besitzt also zu einem bestimmten Zeitpunkt einen definierten Zustand (die internen Daten), mit dem es für andere Objekte zum Nachrichtenaustausch (Änderung des Zustands) bereitsteht. Dies bezeichnet man auch als „shared mutable state“. Kapselung (Information Hiding) soll Details über die Implementierung und Struktur von Objekten verbergen. Das bedeutet, wie weiter oben schon angedeutet, dass der Kontrollfluss und damit die Zustandsänderung ausschließlich dem Objekt selbst obliegt. Als Folge daraus kann kein Objekt Annahmen über das Ablaufverhalten und den internen Zustand eines anderen Objektes machen.

Die weitreichende Konsequenz ist, dass Objekte in echt parallel (multithreaded) arbeitender Software vor gleichzeitiger Verwendung mehrerer anderer Objekte unbedingt geschützt werden müssen, um unkoordinierte Zustandsänderungen durch parallele Zugriffe zu verhindern!

Multithreading und Synchronisation

Dieser Schutz wird durch eine Synchronisation erreicht, d.h. es wird (z.B. durch Mechanismen wie Semaphore und Locks) sichergestellt, dass bestimmte Code-Teile (z.B. die, die von einem anderen Objekt Gebrauch machen) niemals von mehr als einem einzelnen Ablaufaden (Thread) gleichzeitig durchlaufen werden können.

Diese Stellen sind also nicht parallelisierbar und werden automatisch zu einem Flaschenhals. So geartete Software erreicht aufgrund der schwer zu durchschauenden Kontrollflüsse, sehr schnell eine hohe Komplexität und birgt ständig Fehlersituationen wie Deadlocks und Race Conditions, die für parallele Programme typischen Gefahren darstellen.

Aktor-Modell

Diese Ausführungen zeigen, dass das OOP-Paradigma nicht optimal für die Anwendung in multithreaded-Umgebungen ist. Daraus folgt aber nicht zwingend, dass es völlig ungeeignet ist. Erweitert man die Objekt-Definition um einen Aspekt der Ablaufsteuerung, wird daraus ein Akteur.

Ein Akteur ist nichts anderes als ein Objekt, dem ein eigener definierter Ablaufpfad hinzugefügt wird. Sendet ein Objekt einem Akteur eine Nachricht, stellt der Akteur diese Nachricht in eine interne Warteschlange, die er nach dem FIFO-Prinzip (first in, first out) abarbeitet (siehe Abbildung 2). Die Abarbeitung findet unter der Ablaufkontrolle des Akteurs in einem eigenen Thread statt, so dass keine externen nebenläufigen Ereignisse (parallele Aufrufe anderer Objekte) den Zustand des Akteurs ändern können. Der Akteur kann die Nebenläufigkeit gezielt steuern.

Dies hat einige Konsequenzen, die es zu berücksichtigen gilt. Die Kommunikation mit einem Akteur ist stets asynchron. Es leuchtet ein, dass die ihm per Nachricht übergebenen

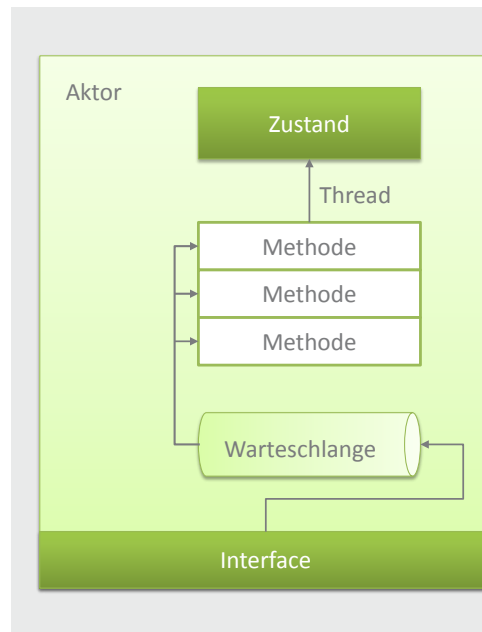


Abb. 2: Akteur-Modell.

Daten Kopien der Originalwerte sein müssen - Referenzen sind verboten. Damit werden auch gleichzeitig zwei Nachteile des Akteur-Modells deutlich: die geringere Performance durch den erhöhten internen Verwaltungsaufwand und der durch Wertekopien erhöhte Speicherverbrauch.

Prominente Vertreter für die Verwendung von Akteuren sind die Sprachen Scala und Erlang. Es gibt aber auch für fast jede andere Sprache Bibliotheken, die das Modell implementieren.

Funktionale Programmierung

Wie oben ausgeführt ist der Grund für das Synchronisationsdilemma der „shared mutable state“. Was läge also näher, als diesen komplett zu vermeiden. Es wird also ein Modell benötigt, in dem in Prozessen keine Zustandsänderungen stattfinden und jegliche Nebeneffekte vermieden werden. Damit wird eine Synchronisation in nebenläufigen Prozessen überflüssig.

Das klingt abstrakt und tatsächlich ist die Idee an den Funktionsbegriff der Mathematik angelehnt. Dort liefern Funktionen per Definition, egal wie oft und von wo sie aufgerufen werden, immer denselben Wert, sofern sie mit gleichen Eingangsparametern versorgt werden. Die Informatik kennt dieses Konzept unter dem Begriff „Seiteneffektfreiheit“ bereits seit den 1960er Jahren. Der prominenteste Vertreter einer rein funktionalen Sprache ist LISP (List Processing), die bereits 1958 spezifiziert wurde.

In diesem Zusammenhang erfährt der Begriff der Funktion außerdem eine deutliche Aufwertung. Funktionen werden

Glossar

Compiler

Ein Compiler ist ein Programm, das die Übersetzung einer Programmiersprache vornimmt. Dabei wird meist von einer höheren Programmiersprache, die im sg. Quellcode vorliegt, in eine für einen Rechner spezifische, direkt ausführbare Maschinensprache übersetzt.

Deadlock (Verklemmung)

Zwei gleichzeitig laufende Prozesse warten auf ein Ereignis, das aber nur der jeweils andere auslösen kann. Ein einfaches Beispiel dafür ist eine Kreuzung, an der in jeder Richtung jeweils ein Fahrzeug steht und die „Rechts vor Links“-Regel gilt.

DSL

Domänen-spezifische Sprachen haben einen Fokus auf eine eng eingegrenzte (fachliche oder technische) Problemstellung. Diese können als interne (in eine Wirtssprache eingebettet) oder als externe (mit eigener Syntax und Werkzeugunterstützung) DSL implementiert werden. Ein Beispiel für eine technische DSL ist SQL (Structured Query Language) für die Problem-domäne der Datenbankabfrage.

Multithreading

Multithreading - auch Nebenläufigkeit genannt - bezeichnet den echt parallelen Ablauf mehrerer Ausführungsstränge (Threads, Tasks) innerhalb eines Prozesses. Eine wesentliche Eigenschaft dieser Threads ist es, dass sie sich einen gemeinsamen Speicherbereich teilen, auf den sie gleichzeitig zugreifen können (im Gegensatz zu voneinander abgeschotteten Prozessen). Dadurch können Fehlersituationen entstehen, die durch eine Synchronisation der Zugriffe vermieden werden müssen.

Race Condition

Zwei gleichzeitig laufende Prozesse schreiben ihr Berechnungsergebnis in eine Variable. Abhängig von der Laufzeit der beiden Prozesse beinhaltet die Variable unterschiedliche, während der Laufzeit wechselnde und damit evtl. temporär falsche Werte. Es gibt diverse Variationen dieses einfachen Szenarios.

Smalltalk

Smalltalk gilt als die „Mutter der objektorientierten Sprachen“. Sie wurde u.a. von Alan Kay und Adele Goldberg in den 1970er Jahren am Xerox Palo Alto Research Center entwickelt. Smalltalk wurde bereits in einer virtuellen Maschine mit automatischer Speicherbereinigung ausgeführt und ist rein objektorientiert.

Viele Eigenschaften heutiger moderner Sprachen, wie beispielsweise Blöcke (Closures) waren in Smalltalk bereits implementiert. Smalltalk hat außerdem eine sehr einfache Syntax und sehr wenige Schlüsselwörter.

als „First Class Citizens“ eingeführt. Das bedeutet, dass eine einmal definierte Funktion auch Variablen zugewiesen werden kann. Funktionen besitzen die gleiche Wertigkeit wie andere Datenstrukturen, was es u.a. erlaubt sie auch in anderen Funktionen als Übergabe- oder Rückgabeparameter zu verwenden. Moderne Sprachvertreter des funktionalen Paradigmas sind beispielsweise F# (Microsoft), Clojure und Haskell.

Gegenwart

Durch die vorigen Ausführungen sind einige Gründe für die Entwicklungen der vergangenen Jahre klarer geworden. Insbesondere die Entstehung neuer Sprachen wie Scala, Closure u.a. sind zu großen Teilen dem Bedarf an Werkzeugen zur Handhabung von Nebenläufigkeiten geschuldet. Denn hier haben die funktionalen Prinzipien deutliche Vorteile.

Aber es wird noch etwas Anderes deutlich: Auf die lieb gewonnenen Errungenschaften der objektorientierten Programmierung wollen (und können) wir nicht verzichten. Schließlich haben sich die Problemstellungen nicht komplett verändert, sondern wurden um einzelne Aspekte (u.a. Nebenläufigkeit) erweitert. Das wird besonders an Sprachen wie Java und Scala deutlich.

Scala versucht bewusst aus der Vergangenheit von Java zu lernen. Vereinfacht formuliert bereinigt sie einige Dinge und ergänzt sinnvoll funktionale Aspekte. Daraus ergibt sich zwar eine nicht ganz einfach zu erlernende Sprache, jedoch bietet sie alle Mittel, um funktional oder objektorientiert auf aktuelle Probleme reagieren zu können.

Auch die Java Community ringt um funktionale Erweiterungen und versucht diese mit der kommenden Sprachversion (Java 8) u.a. in Form von Lambda-Ausdrücken in das vorhandene, bewährte Ökosystem zu integrieren.

Es gibt eine weitere Auffälligkeit, die erwähnt werden muss. Viele der neu entwickelten Sprachen haben eine gemeinsame Plattform: die Java Virtual Machine (JVM). Diese Plattform hat sich weiter geöffnet - in jüngerer Vergangenheit auch für dynamische typisierte Sprachen. Auch diesem Fakt ist der Sprachenhype ein wenig geschuldet, denn mit der JVM steht eine solide Sprachinfrastruktur mit vielen grundlegenden Funktionen wie z.B. Bytecode-Ausführung und automatischer Speicherverwaltung sowie ein riesiges Ökosystem an Bibliotheken zur Verfügung. Dies alles vereinfacht die Entwicklung neuer Sprachen erheblich.

Da sich die JVM-basierten Sprachen dank der gemeinsamen Plattform gut integrieren lassen, könnte sich demnächst der Fokus von Java als Sprache hin zu Java als Plattform verschieben. Je nach Schwerpunkt der Problemstellung könnte man das objektorientierte Java mit funktionalen Elementen oder aber Clojure als funktionale Sprache unter Einbeziehung bewährter Java Libraries z.B. zur Programmierung von Benutzeroberflächen verwenden.

Fazit

Ein echter Paradigmenwechsel, wie von der imperativen zur objektorientierten Programmierung, zeichnet sich momentan nicht ab. Die aufgezeigten Mängel am OOP-Paradigma lassen sich durch Hinzunahme einiger funktionaler Aspekte ausgleichen. Für die meisten Anforderungen stehen ausgereifte Plattformen und Sprachen zur Verfügung. Selbst sehr individuelle Anpassungen lassen sich heute mit guten Werkzeugen zur Erstellung von Domänen-spezifischen Sprachen (DSLs) relativ einfach realisieren.

Damit stehen uns für die aktuellen großen Herausforderungen wie z.B. das Mobile- und Cloud-Computing, bei denen die Aspekte der Parallelität und Verteilung eine besondere Rolle spielen, wichtige und geeignete Werkzeuge zur Verfügung. Das bedeutet jedoch keinesfalls, dass damit alle Probleme gelöst sind. Insbesondere bei der Entwicklung von intelligenten Compilern und Laufzeitumgebungen, die eigenständig Code parallelisieren können, herrscht noch Nachholbedarf. Echte Paradigmenwechsel in der Programmierung werden aber vermutlich auch in Zukunft eher seltene Ereignisse bleiben.



Andreas Flügge
(info@ordix.de)

Seminarempfehlung: Einführung in die objektorientierte Programmierung und UML

► Informationen/Online-Anmeldung: <http://training.ordix.de>

In diesem Seminar erwerben Sie Kenntnisse über die grundlegenden Konzepte und Begriffe der objektorientierten Softwareentwicklung sowie der Unified Modeling Language (UML). Das Seminar zeigt den Einsatz der UML für die Modellierung von objektorientierten Softwaresystemen in den frühen Prozessphasen Analyse und Design. Als Beispiel einer modernen OO-Programmiersprache dient Java. In praktischen Übungen erstellen Sie UML-Diagramme zur Visualisierung objektorientierter Grundlagen wie Klassen, Objekte, Abstraktion, Datenkapselung, Vererbung und Beziehungen bis hin zu Polymorphie.

Seminarinhalte

- Grundlagen der Objektorientierung:
 - Abgrenzung strukturierte / objektorientierte Programmierung
 - Klassen und Objekte
 - Beziehungsarten
 - Kapselung, Vererbung, abstrakte Klassen, Polymorphie
 - Nachrichtenaustausch
 - Entwurfsmuster
- UML-Notationen:
 - Klassendiagramme
 - Assoziation, Aggregation, Komposition
 - Sequenz- und Kollaborationsdiagramme
 - Anwendungsfälle (Use Cases)
 - Aktivitätsdiagramme
- Vorgehensmodell (Analyse und Design):
 - Anforderungsanalyse
 - Anwendungsarchitektur
 - Geschäftsklassenidentifizierung/-modellierung
 - Komponententwurf
- Vertiefung der Theorie durch praktische Übungen und Beispiele

Termine

17.03. - 19.03.2014 in Wiesbaden
 02.06. - 04.06.2014 in Wiesbaden
 08.09. - 10.09.2014 in Wiesbaden
 03.11. - 05.11.2014 in Wiesbaden

Seminar-ID: OO-01
Dauer: 3 Tage
Preis pro Teilnehmer:
 1.190,00 € (zzgl. MwSt.)
Frühbucherpreis:
 1.071,00 € (zzgl. MwSt.)



Buchen Sie gleich hier!



Oracle-Editionen

Muss es immer Enterprise sein?

Immer häufiger werden wir in unseren Kundenprojekten auf das Thema Lizenzkosten angesprochen. Neben anderen Herstellern hat auch Oracle ein für viele Kunden nicht leicht zu durchschauendes Lizenzmodell. Meist steht dabei die Frage im Mittelpunkt, welche Edition aus fachlicher und technischer Sicht die kostengünstigste ist. Dieser Artikel gibt einen kleinen Überblick über die verfügbaren Oracle-Editionen, ihren Funktionsumfang und die zu erwartenden Kosten.

Wie ist das Lizenzmodell strukturiert?

Die aktuelle Preisliste vom 13. Februar 2014 [2] weist grundsätzlich zwei unterschiedliche Lizenzierungsmodelle für Oracle-Datenbanken aus: die Standard Edition (SE) und die Enterprise Edition (EE). Diese beiden Editionen gibt es wiederum in unterschiedlichen Ausprägungen [2].

Die Standard Edition kann als SE One und als SE Variante lizenziert werden. Zusätzlich zur Enterprise Edition können wiederum Options und Packs in das kostenpflichtige Portfolio eingebunden werden.

Zusätzlich werden von Oracle noch die kostenfreie Express Edition und die Personal Edition angeboten. Diesen beiden Lizenzformen werden wir uns in dem vorliegenden Artikel allerdings nicht widmen.

Welche grundsätzlichen Begrenzungen gibt es?

Die Standard Edition betrachtet grundsätzlich nur Sockets. Die SE One kann auf Maschinen mit maximal 2 Sockets und die SE auf Maschinen mit bis zu 4 Sockets lizenziert werden.

Bei der Enterprise Edition sind grundsätzlich die Cores zu lizenzieren. Je Core gibt es einen Faktor. Die Anzahl der Cores multipliziert mit dem Faktor ergibt die zu lizenzierende CPU-Anzahl. Je nach CPU-Typ existieren unterschiedliche Faktoren [3].

Die Abbildung 1 stellt einen Auszug aus der Liste dar und gibt einen Überblick über die Faktoren für die gängigsten Prozessoren. Eine Lizenzierung ist dann je CPU oder je Benutzer (NUP) möglich. Nachfolgend betrachten wir die Lizenzierung nach CPU.

Vendor and Processor	Core Processor Licensing Factor
SPARC T3 Processor	0,25
AMD Opteron Models 13XX, 23XX, 24XX, 32XX, 41XX, 42XX, 43XX, 61XX, 62XX, 63XX, 83XX, 84XX or earlier Multicore chips	0,5
Intel Xeon Series 56XX, Series 65XX, Series 75XX, Series E7-28XX, Series E7-48XX, Series E7-88XX, Series E5-24XX, Series E5-26XX, Series E5-46XX, Series E5-16XX, Series E3-12XX or earlier Multicore chips	0,5
Sun UltraSPARC T2	0,75
IBM POWER7+	1,0

Abb. 1: Auszug aus der Core Factor Table.

Was kostet das denn?

Die Abbildung 2 zeigt einen Auszug aus der eingangs erwähnten Preisliste von Oracle. Eine Maschine mit 2 Sockets zu je 12 Cores (x86_64) müsste entsprechend, wie in Abbildung 3 dargestellt, lizenziert werden.

Anhand dieses Beispiels wird ersichtlich, dass es immense Kostenunterschiede gibt. Daher sollten sich die Kunden von Oracle die Frage stellen, was sie wirklich benötigen.

Bei einer Virtualisierung wird das Thema noch viel komplexer. Hier ist dann entscheidend, welche Form der Virtualisierung eingesetzt wird, z.B. ESX-Server, Oracle VM oder AIX lpar. Sollten Sie zu diesem Thema Fragen haben, können Sie uns gerne ansprechen.

Was beinhalten die einzelnen Editionen?

In der DOC_ID 1084132.1 im Oracle-Metalink (alter Name des Oracle Supports) [4] sind die Inhalte der einzelnen Editionen aufgelistet. Für jede Oracle-Version gibt es dazu ein dediziertes Dokument. Die wichtigsten Unterschiede am Beispiel der Version 11gR2 zeigt die Abbildung 4.

Unerlässlich ist daher vorab eine intensive Analyse der tatsächlichen Anforderungen des Kunden. In individuellen Beratungsgesprächen müssen die Anforderungen aufgenommen und anschließend die passende Lizenzierung ermittelt werden. Auch ein Downsizing von einer EE zu einer SE sollte betrachtet werden. Viele Kunden nutzen für ihre Datenbanken die EE, obwohl sie nur Funktionalitäten der SE in Anspruch nehmen.

Oftmals ist das Thema Hochverfügbarkeit ausschlaggebend für eine Lizenzierung der EE. In unseren Beratungsgesprächen weisen wir immer wieder darauf hin, dass RAC in der SE sehr wohl enthalten ist. Die Data-Guard-Funktionalität lässt sich mit Fremdprodukten (z.B. Dbvisit [1]) sehr gut nachstellen, auch wenn dann nicht der volle Funktionsumfang gegeben ist.

Database Product	Processor License (€)
Standard Edition One	4.578
Standard Edition	13.813
Enterprise Edition	37.492

Abb. 2: Auszug aus der Global Price List von Oracle [5].

Edition	Sockets	Cores	Faktor	CPUs	Preis (€)
SE One	2	24			9.156
SE	2	24			27.626
EE	2	24	0,5	12	449.904

Abb. 3: Rechenbeispiel.

Funktionalität	SE One	SE	EE
Online Rebuild			X
Data Guard			X
Active Data Guard			Option
RAC		X	Option
Parallel Query / DML / DDL			X
Online Table Redefinition			X
Flashback Query	X	X	X
Flashback Table / Database			X
Diagnostic Pack (AWR)			Pack

Abb. 4: Auszug aus „Difference Between Editions on Oracle 11.2!“.

Links

- ▶ [1] ORDIX® news Artikel 4/2013 „Dbvisit Standby vs. Oracle Data Guard - Wachablösung für Data Guard?": http://www.ordix.de/images/ordix/onevs_archiv/4_2013/ORDIX_news_4_2013_opf_files/WebSearch/page0035.html
- ▶ [2] Oracle Technology Global Price List/Lizenzmodell: <http://www.oracle.com/us/corporate/pricing/technology-price-list-070617.pdf>
- ▶ [3] Oracle Processor Core Factor Table: <http://www.oracle.com/us/corporate/contracts/processor-core-factor-table-070634.pdf>
- ▶ [4] Oracle Support: https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=6cm5ld42j_9&_afLoop=3705960881914

Glossar

EE

Enterprise Edition

SE / SE One

Standard Edition / Standard Edition ohne RAC

RAC

Real Application Cluster

Data Guard

Oracle Hochverfügbarkeitslösung

Dbvisit

kostengünstige Alternative zu Data Guard

NUP-Metrik

Named User Plus - Lizenzmetrik für Unternehmen bei der alle berechtigten Personen sowie alle nicht benutzerbedienten Geräte lizenziert werden, die auf die Oracle-Programme zugreifen.

HINWEIS:

Die genannten Beispielrechnungen sollen den Kostenunterschied illustrieren. Die Angaben sind ohne Gewähr.

Bildnachweis

► © flickr.com |The Wizard | Sean McGrath

Fazit

Das Thema Lizenzierung ist von stetig zunehmender Bedeutung. Bei vielen Kunden schlummern erhebliche Einsparungspotentiale, die nur erschlossen werden müssen. Daher beraten wir Sie gerne, hinsichtlich Ihrer individuellen Möglichkeiten.

Grundsätzlich gilt:

Es muss nicht immer die teure Enterprise Edition sein!



Klaus Reimers
(info@ordix.de)

Einladung zum Treffpunkt RheinMainIT:

„Wird Ihre Datenbank-Hochverfügbarkeit auch durch die Kosten bestimmt?“

ORDIX
Veranstaltung

Dieser Treffpunkt widmet sich den Themen Datenbank-Hochverfügbarkeit und Lizenzmodellen. In zwei Vorträgen möchten wir Ihnen einen Überblick über die angebotenen Oracle-Editionen, deren Leistungsumfang und mögliche Kostenmodelle geben.



Vortrag 1:
Datenbank-Lizenzen - muss es wirklich immer so teuer sein?
Klaus Reimers, ORDIX AG, gibt einen Überblick über die unterschiedlichen Lizenzierungsformen, stellt die Inhalte der einzelnen Editionen vor und erläutert mögliche Ansätze zur Kostenminimierung erläutert.



Vortrag 2:
Oracle Datenbank-Hochverfügbarkeit
Thilo Fleischhauer, ORDIX AG, widmet sich den Oracle-Hochverfügbarkeitslösungen. Zudem stellt er die Unterschiede zwischen einem RAC und Data Guard sowie Alternativen von Drittanbietern vor.

27. März 2014, 14.00 - 18.00 Uhr, ORDIX® Seminarzentrum

www.RheinMainIT.de

treff. RheinMainIT

Neue Funktion svcbundle von Solaris 11

Service bündeln leicht gemacht

In diesem und folgenden Artikeln möchten wir Ihnen die interessanten Neuheiten von Solaris 11 vorstellen. Der Fokus liegt dabei auf Themen, die sich seit Solaris 10 weiterentwickelt haben oder gänzlich durch andere Techniken ersetzt wurden. Das ist nicht zuletzt für Administratoren von Solaris 10 interessant, denen eine Migration auf Solaris 11 bevorsteht. Auch Systemadministratoren anderer Unix-Derivate finden hier sicher interessante Lösungsansätze, die ihnen das Betriebssystem aus dem Hause Oracle schmackhaft machen könnten.

Service Management Facility

Das mit Solaris 10 eingeführte Service Management Facility (SMF) stellt eine völlig neue Dienstverwaltung dar, welche u.a. auch den Servicebegriff an sich neu definiert und erweitert. Parallel werden weiterhin die klassischen RC-Skripte unterstützt. Allerdings können diese nicht die Vorteile des neuen Systems nutzen. Um von den neuen Funktionalitäten profitieren zu können, muss eine Applikation zu einem SMF-Service umgeschrieben werden.

Dies war bisher ein etwas mühsamer, individueller und dadurch nicht zuletzt fehleranfälliger Prozess. Mit Solaris 11.1 wird hierfür das Werkzeug `svcbundle` ausgeliefert, das dem Administrator einen Großteil der Arbeit abnimmt. Diese Funktion wird im vorliegenden Artikel vorgestellt und seine Arbeitsweise anhand eines Beispiels verdeutlicht.

Einführung in Solaris 10

Beginnen wir zunächst mit einer rückblickenden Zusammenfassung von SMF:

- Der Servicebegriff wird erweitert. Ein Dienst muss nicht unbedingt durch einen Daemon- oder Hintergrundprozess repräsentiert sein, er kann auch ein eingehängtes Dateisystem oder ein erfolgreich konfiguriertes Netzwerk sein.
- Eine systematische Service-Nomenklatur (FMRI), deren Aufbau nach Funktionalität strukturiert ist, wird eingeführt (siehe Abbildung 1).
- Die Dienste kennen ihre Abhängigkeiten untereinander. Damit ist implizit eine Startreihenfolge definiert und auch eine Parallelisierung des Systemstarts möglich.
- Die Dienste werden zur Laufzeit auf ihre Funktionalität hin überwacht und z.B. automatisch neu gestartet, falls sie einmal unvorhergesehen beendet worden sein sollten. Im Gegensatz dazu werden beim klassischen Runlevel-Konzept die Dienste, die das System anbieten soll, beim Boot-Prozess oder Runlevel-Wechsel per Shell-Skript gestartet und anschließend sich selbst überlassen.

- So hat man unter anderem zu jedem Zeitpunkt die Möglichkeit, den (ordnungsgemäßen) Zustand des Systems zu überblicken.
- Durch SMF werden einheitliche Administrations- und Konfigurationsschnittstellen übergreifend zur Verfügung gestellt (`svcs`, `svccfg`, `svcprop`, `svcadm`).
- Generalisiertes Logging: Alle Dienste protokollieren ihre Aktivitäten bzgl. Start- und Stopp-Vorgängen an zentraler Stelle (`/var/svc/log`) nach einem einheitlichen Verfahren.

Alle hier genannten Aspekte von SMF erweitern das klassische Runlevel-Konzept und beseitigen zugleich alte Schwächen.

Um in den Genuss dieses Konzeptes zu kommen, ist allerdings ein gewisser Formalismus notwendig, der im Folgenden skizziert wird:

Manifeste und Profile

Dienste werden über s.g. Manifeste definiert. Dies sind ASCII-Dateien im XML-Format und enthalten alle Informationen, die für den beschriebenen Service notwendig sind:

- Start- und Stop-Mechanismen (inkl. Timeouts)
- Abhängigkeiten zu anderen Diensten

FMRI	Dienstleistung
<code>svc:/system/filesystem/local:default</code>	Einhängen der lokalen FS
<code>svc:/network/inetd:default</code>	Startet und überwacht den inetd-Prozess
<code>svc:/application/pkg/server:default</code>	IPS-Paket-service

Abb. 1: Beispiele unterschiedlicher Services unter Solaris 10/11.

- evtl. Zugriffsberechtigungen (RBAC)
- und viele weitere, frei definierbare Eigenschaften, wie z.B. Dokumentationshinweise, Umgebungsvariablen, Firewall-Eigenschaften, Ressourcenzuweisungen, Instanznamen, Benutzer- und Gruppenzugehörigkeiten, etc.

Diese XML-Beschreibung von Hand zu erstellen stellt eine gewisse Herausforderung dar. Bei Solaris 10 gab es bisher vom Betriebssystem keine Unterstützung an dieser Stelle.

```
#!/bin/sh
# Einbinden des SMF-Skript-Frameworks:
. /lib/svc/share/smf_include.sh
# Verwendung möglicher ksh-Syntax,
# da unter Solaris 11.1 /bin/sh auf ksh93 verlinkt:
OUT=/tmp/${basename $0}.out
# Einmalige Startmeldung
echo "Start des Skriptes $0" >> $OUT
# Hintergrundprozess, der alle 10 Sekunden
# einen Zeitstempel # schreibt:
while true; do echo "# $(date)" >> $OUT; sleep 10; done &
```

Abb. 2: Das Shell-Skript `myapp.sh`.

```
rn@s11-1:~/SMF$ svcbundle \
-s service-name=site/myapp # Service-Name
-s start-method=~rn/SMF/myapp.sh
-s model=daemon # notwendig, da im Skript ein
# Hintergrundprozess gestartet
# wird
-i # sorgt für Import und
# Aktivierung des Services
```

Abb. 3: Generierung des SMF-Manifestes.

```
rn@s11-1:~/SMF$ svcs -lp myapp
fmri      svc:/site/myapp:default
enabled   true
state     online
next_state none
state_time 5. Januar 2014 04:45:19 Uhr CET
logfile   /var/svc/log/site-myapp:default.log
restarter svc:/system/svc/restarter:default
contract_id 295
manifest  /lib/svc/manifest/site/myapp.xml
dependency require_all/none svc:/milestone/multi-user
(online)
process   9985 /bin/sh /home/rn/SMF/myapp.sh
process   10046 sleep 10
```

Abb. 4: Der neue Dienst inklusive gestarteter Prozesse.

Daher ist man in der Regel so verfahren, dass man eine XML-Datei eines bereits laufenden Dienstes als Vorlage verwendet und diese dann von Hand an seine Bedürfnisse angepasst hat. Aber hier hilft nun `svcbundle`!

Hiermit können nun auf einfache Weise SMF-Manifeste und Profile erstellt werden, ohne sich tiefgehend mit dem XML-Format der zugrundeliegenden Dateien auseinandersetzen zu müssen. Dieser Zusammenhang wird nachfolgend an einem Beispiel demonstriert:

Einbinden eines Skriptes in SMF mit Hilfe von `svcbundle` (1M)

Im Rahmen dieses Beispiels soll ein Daemon-Prozess in SMF eingebunden werden. Dazu verwenden wir ein triviales Skript (`myapp.sh`), welches regelmäßig einen Zeitstempel in eine Datei schreibt.

Zunächst muss das Skript `myapp.sh` angelegt (siehe Abbildung 2) und mit Ausführungsrechten versehen werden.

Die nächsten Schritte sind:

- Erzeugen des Manifests
- Import in das Repository
- Aktivierung des neuen Dienstes

Im einfachsten Fall kann dies wie in Abbildung 3 geschehen. Die Option `-i` erledigt die drei genannten Punkte und bringt folgende Ausgabe:

```
Waiting for site/myapp to reach online state.
It is safe to interrupt.
```

Werden keine weiteren Angaben gemacht, verwendet `svcbundle` folgende Defaults:

- Master-Restarter ist `svc.startd`
- der neue Dienst ist abhängig vom multi-user-Meilenstein
- ein neuer Dienst wird aktiviert
- Service- und Bundle-Name sind gleich
- für alle `exec`-Methoden wird ein Timeout von 60 Sekunden gesetzt

Sollen Anpassungen an der XML-Datei vorgenommen werden, muss man auf die automatische Installation verzichten und dafür mit der Option `-o` eine Ausgabedatei angeben. Nachdem die Änderungen vorgenommen sind (z.B. Hinterlegung einer Beschreibung des Dienstes), kann der Dienst wie folgt aktiviert werden:

Im ersten Schritt wird der Dienst importiert. Hierzu muss das soeben generierte Manifest zunächst in das Standardverzeichnis kopiert werden:

```
rn@s11:~/SMF$ sudo cp myapp.xml /lib/svc/manifest/site/
```

Im zweiten Schritt muss der Manifest-Import-Dienst neu gestartet werden:

```
rn@s11:~/SMF$ sudo svcadm restart manifest-import
```

Dies führt dazu, dass der neue Service in das SMF-Repository eingebunden und aktiviert wird - er steht ab sofort unter der Kontrolle von SMF:

```
rn@s11:~/SMF$ svcs myapp
STATE      STIME      FMRI
online     8:11:20   svc:/site/myapp:default
```

Nun kann man sich den Dienst einschließlich der gestarteten Prozesse anzeigen lassen (siehe Abbildung 4). Damit wird **myapp** automatisch bei jedem Neustart des Systems aktiviert, sobald alle Abhängigkeiten (hier der multi-user-Meilenstein) erreicht sind. Das Ein- und Ausschalten erfolgt mit **svcadm**. Das sorgt dann auch dafür, dass die entsprechenden Prozesse beendet werden.

```
rn@s11:~/SMF$ svcadm disable myapp
```

Was bietet svcbundle noch?

Mit **svcbundle** lassen sich insbesondere SMF-Services auf Basis vorhandener RC-Skripte erstellen, Näheres dazu ist in den Handbuchseiten von **svcbundle (1M)** [2] nachzulesen. Außer Manifesten können auch Profile erstellt werden. Diese ähneln vom Aufbau her den Manifesten und werden verwendet, um Services anzupassen - z.B. um bestimmte Service-Konfigurationen zu setzen.

Fazit

Mit **svcbundle** hat der Administrator ein komfortables Werkzeug an der Hand, um den von Solaris beschrittenen Weg der Vereinheitlichung von Administration und Konfiguration von Systemdiensten auch auf die eigenen Services auszuweiten.



Roger Niemeyer
(info@ordix.de)

Glossar

Fault Management

Fehlermanagement - Hierunter versteht man das Erkennen, das Protokollieren, das Melden und Beheben von auftretenden Fehlerzuständen.

FMRI

Solaris Subsysteme wie z.B. Service Management Facility (SMF), der Fault Management Dämon (FMD) und das Image Packaging System (IPS) benutzen jeweils bestimmte Nomenklaturen, den s.g. Fault Management Resource Identifier (FMRI) um die Elemente, die sie handhaben eindeutig zu identifizieren.

IPS

Das Image Packaging System ist die neue Softwarepaketverwaltung unter Solaris 11.

RC-Skripte

RC-Skripte sind die im klassischen SVR4-Unix gebräuchliche Methodik, um Systemdienste zu starten und zu stoppen.

RBAC

Die Role Based Access Control ist eine rollenbasierte Berechtigungsverwaltung.

Links

- ▶ [1] ORDIX® news Artikel 1/2013
„Solaris 11 - Neue Funktionen (Teil III) - Ist Solaris 11 das bessere Linux?“:
http://www.ordix.de/images/ordix/onews_archiv/1_2013/ORDIX_news_1_2013_opf_files/WebSearch/page0010.html
- ▶ [2] Handbuchseiten zu svcbundle(1M):
http://docs.oracle.com/cd/E26502_01/html/E29031/svcbundle-1m.html#scrolltoc
- ▶ [3] Oracle Technetwork-Artikel zu svcbundle:
<http://www.oracle.com/technetwork/articles/servers-storage-admin/howto-svcbundle-manifest-profile-1866525.html>
- ▶ Seminarempfehlungen:
Solaris 11 Administration Neuheiten
<http://training.ordix.de/siteengine/action/load/kategorie/Betriebssysteme%20&%20Monitoring/nr/1111/index.html>
Solaris 11 Systemadministration Grundlagen
<http://training.ordix.de/siteengine/action/load/kategorie/Betriebssysteme%20&%20Monitoring/nr/1599/index.html>
Solaris 11 Systemadministration Aufbau
<http://training.ordix.de/siteengine/action/load/kategorie/Betriebssysteme%20&%20Monitoring/nr/1600/index.html>

SQL-Neuerungen in Oracle 12c

Nachdem im zweiten Teil dieser Reihe [2] die Multitenant-Architektur vorgestellt wurde, stellen wir Ihnen in diesem Artikel die Neuerungen rund um SQL vor. Wie erwartet, hat Oracle seiner Datenbankversion 12c sowohl eine Reihe neuer SQL-Funktionen hinzugefügt, als auch bestehende Funktionen weiterentwickelt. Als Schlagwörter seien hier bitemporale Abfragen, Limitierung von Ergebnismengen oder die Möglichkeit Tabellen- und View-Spalten für die Anwendung unsichtbar zu machen genannt.

Tabellenspalten: Sichtbar und unsichtbar

Ab Oracle 12c können Spalten sowohl in Tabellen als auch in Views unsichtbar gemacht werden. Die Besonderheit der unsichtbaren Spalten ist, dass diese bei Operationen wie **SELECT ***, dem Befehl **DESCRIBE** oder bei einem **INSERT**-Statement ohne explizite Spaltenangabe, nicht berücksichtigt werden. Sollen unsichtbare Spalten bei einem **SELECT**- oder **INSERT**-Statement berücksichtigt werden, so müssen diese explizit aufgeführt werden. Damit lassen sich Tabellen oder auch Views um neue unsichtbare Spalten erweitern, ohne dass diese Auswirkungen auf bestehende Anwendungen haben.

Bei der Verwendung von unsichtbaren Spalten gibt es allerdings auch einige Einschränkungen. So können sie bei External Tables, Cluster Tables oder Temporary Tables nicht verwendet werden. Die Attribute bei einem Objekttyp können ebenfalls nicht als unsichtbar definiert werden.

Unsichtbare Spalten sind dabei nicht mit den sogenannten Hidden Columns zu verwechseln. Eine unsichtbare Spalte kann nachträglich sichtbar gemacht werden - eine Hidden Column dagegen nicht.

Wird eine unsichtbare Spalte sichtbar gemacht, so wird diese Spalte, unabhängig von der physikalischen, internen Reihenfolge innerhalb der Tabelle, immer an der letzten Stelle der Tabelle sichtbar gemacht. Dieses Verhalten wird über eine sogenannte externe und eine interne Spaltennummer realisiert. Dabei ist für die Darstellung der Tabelle nach außen, z.B. bei der Verwendung von einem **DESCRIBE**-Befehl, die externe Spaltennummer entscheidend.

Wenn Sie Romane oder andere große Texte speichern wollen, dann ist das mit Oracle 12c nun auch mit **VARCHAR2** möglich. Die Größe der Charakter-Datentypen **VARCHAR2**, **NVARCHAR2** und **RAW** wurde auf 32.767 Bytes erhöht. Um diese Funktion nutzen zu können, muss der Initialisierungsparameter **MAX_STRING_SIZE = EXTENDED** gesetzt werden. Ab 4.000 Bytes großen **VARCHAR2**-, **NVARCHAR2**-Spalten bzw. ab 2.000 Bytes großen **RAW**-Spalten spricht

man nicht mehr von Standard- sondern von Extended Datentypen. Diese unterliegen den gleichen Regeln und Restriktionen wie LOBs und werden, falls sie über 4 KB groß sind, out-of-line gespeichert.

Automatische Sequences und Identity Columns

Das Generieren von eindeutigen Werten für z.B. eine Schlüssel-Tabellenspalte, ist in Oracle 12c deutlich vereinfacht worden. So können dafür zum einen **DEFAULT**-Constraints mit Sequenzen oder die sogenannten Identity Columns verwendet werden.

Die Lösung mit einem **DEFAULT**-Constraint setzt eine explizit angelegte Sequence voraus. Diese wird anschließend beim Anlegen eines **DEFAULT**-Constraint auf einer Schlüssel-Tabellenspalte mit der altbewährten **nextval**-Funktion verwendet. Wird bei dieser Lösung der Spalte in einem **INSERT**-Statement der Wert **NULL** zugewiesen, so wird trotz des **DEFAULT**-Constraint auf dieser Spalte der Wert **NULL** gespeichert.

Soll statt **NULL** eine neue Nummer generiert werden, so kann bei der Definition des **DEFAULT**-Constraint der Zusatz **ON NULL** verwendet werden.

Die Identity Columns stellen eine weitere neue Möglichkeit zur Generierung von eindeutigen Werten in einer Spalte zur Verfügung. Die Besonderheit bei dieser Variante ist, dass dafür keine explizit angelegten Sequences notwendig sind. Die notwendige Sequence wird von Oracle implizit erzeugt. Die Identity Columns können dabei in drei verschiedenen Ausprägungen definiert werden:

- Die erste Ausprägung **GENERATED BY DEFAULT AS IDENTITY** zeichnet sich dadurch aus, dass bei der Zuweisung von dem Wert **NULL** zu einer Spalte mit dieser Ausprägung, der Fehler **ORA-01400** ausgegeben wird, da der Wert **NULL** bei dieser Variante nicht erlaubt ist.

- Bei der Ausprägung **GENERATED BY DEFAULT ON NULL AS IDENTITY** ist der Wert **NULL** zulässig. Wird bei dieser Variante einer Spalte der Wert **NULL** übergeben, so wird eine neue, eindeutige Nummer aus der implizit von Oracle angelegten Sequence gezogen.
- Die dritte Variante **GENERATED ALWAYS AS IDENTITY** ist die restriktivste Variante der Identity Columns. Einer Spalte mit dieser Ausprägung darf weder der Wert **NULL** noch ein beliebiger numerischer Wert zugewiesen werden (siehe Abbildung 1).

Eigenschaften wie Startwert, Schrittwert oder Höchstwert, wie man sie von einer Sequence kennt, können bei den Identity Columns ebenfalls angegeben werden.

Blättern im SQL-Ergebnis: LIMIT und OFFSET

Komplexe **SELECT**-Abfragen mit dem **rownum**-Attribut und großen **WHERE**-Bedingungen zum Abfragen einer Stichprobe aus der Mitte einer Datenmenge haben nun ein Ende. Dies ist natürlich nur eines von vielen Szenarien die von der neuen **ROW_LIMITING_CLAUSE** profitieren. Diese erlaubt die Limitierung der zurückgegebenen Zeilen eines **SELECT**-Befehls.

Mit dem Zauberwort **FETCH** lässt sich zudem die Größe der Rückgabe einschränken. Entweder durch Angabe einer Anzahl an Datensätzen oder durch Angabe eines Prozentsatzes.

Der zweite Zauber hört auf den Namen **OFFSET** und bestimmt, wie viele Zeilen zu Beginn der Rückgabe übersprungen werden sollen. Der Zusatz **WITH TIES** bei einem **FETCH** erlaubt zudem die Ausgabe aller Datensätze mit demselben Sortierschlüssel, auch wenn diese mehr als der angegebenen **FETCH**-Anzahl entsprechen. Abbildung 2 zeigt ein Beispiel-Statement, welches auf 10 Datensätzen (1, 2, 3, ... 10) aufbaut. Die ersten fünf Datensätze werden übersprungen (**OFFSET**) und die drei folgenden werden ausgegeben (**FETCH**).

Erweiterte Kompatibilität durch SQL-Übersetzungen

Viele Applikationen, die für einen Einsatz auf Nicht-Oracle-Datenbanken erstellt wurden, können oft nicht ohne signifikante Änderungen auf Oracle-Datenbanken portiert werden. Dies liegt oftmals an den verschiedenen SQL-Dialekten, die sich von Datenbank zu Datenbank unterscheiden.

Das SQL Translation Framework von Oracle 12c ermöglicht es, SQL-Befehle mit dem Dialekt fremder Datenbanken in Oracle SQL zu übersetzen (siehe Abbildung 3). Das Framework wurde für die Verwendung mit OPEN-API-Applikationen, wie JDBC oder ODBC, aber auch für konkrete SQL-Befehle entwickelt. Mit Stand Januar 2014 gibt es vorgefertigte Übersetzungsprofile für Sybase, SQL

```
CREATE TABLE example (
  nr NUMBER ...
);

```

Möglichkeiten:

```
> ...DEFAULT seq_example.NEXTVAL -- (externe Seq.)
> ...DEFAULT ON NULL seq_example.NEXTVAL -- (externe Seq.)
> ...GENERATED ALWAYS AS IDENTITY
> ...GENERATED BY DEFAULT AS IDENTITY
> ...GENERATED BY DEFAULT ON NULL AS IDENTITY
```

Abb. 1: Automatische Sequence-Generierung mit **DEFAULT**-Constraints und den Identity Columns.

```
SELECT zahl FROM
(SELECT level AS zahl
FROM dual
CONNECT BY level<11)
ORDER BY zahl
OFFSET 5 ROWS FETCH NEXT 3 ROWS ONLY;
```

Ausgabe:
ZAHL

6
7

Abb. 2: Beispiel mit **OFFSET** und **FETCH**.

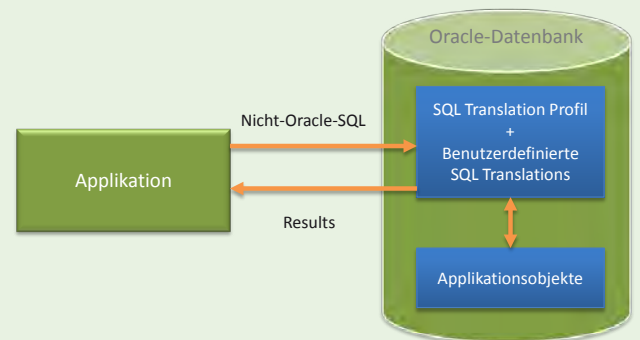


Abb. 3: Schema SQL Translator.

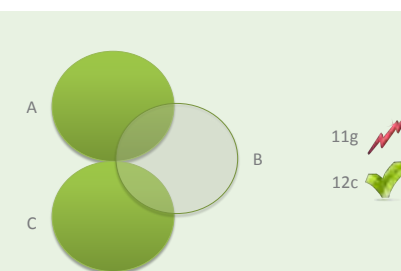


Abb. 4: **OUTER JOIN** bei nativer Oracle-SQL-Syntax in 11g und 12c.

```
SELECT a.a1, b.b1, c.c1
FROM a, b, c
WHERE a.a1 =b.b1(+)
AND c.c1 =b.b1(+);
```

Abb. 5: Erweiterte native OUTER JOIN Oracle-SQL-Syntax in 12c.

```
SELECT a.abteilungsnr,a.abteilungsname,m.mitarbeiter-
name
FROM abteilung a
OUTER APPLY(SELECT * FROM mitarbeiter m
WHERE m.abteilungsnr=a.abteilungsnr) m
ORDER BY a.abteilungsnr=m.mitarbeitername;
```

Abb. 6: Beispiel-SQL mit OUTER APPLY.

```
SELECT *
FROM mitarbeiter m,
(SELECT * FROM abteilung a
WHERE m.abteilungsnr = a.abteilungsnr);
ORA-00904: "M". "ABTEILUNGSNR":
ungültiger Bezeichner

SELECT *
FROM mitarbeiter m,
LATERAL(SELECT * FROM abteilung a
WHERE m.abteilungsnr = a.abteilungsnr);
Ab Oracle 12c ✓
```

Abb. 7: Beispiel-SQL für die Anwendung von LATERAL.

```
CREATE TABLE projekt(
projektnr NUMBER,
projektname VARCHAR(30),
projekt_start DATE,
projekt_ende DATE,
PERIOD FOR projekt_zeitraum(projekt_start,projekt_ende)
);
```

Abb. 8: Tabellendefinition mit Temporal Validity.

```
SELECT projektnr, projektname
FROM projekt VERSIONS PERIOD FOR projekt_zeitraum
BETWEEN to_date('01.01.2013','DD.MM.YYYY')
AND to_date('31.12.2013','DD.MM.YYYY')
ORDER BY projektnr;
```

Abb. 9: Beispiel mit der Abfrage aller gültigen Projekten in 2013.

Server und eingeschränkt für DB2. Selbstverständlich lässt sich auch ein eigenes Profil erstellen.

Verbesserte native OUTER-JOIN-Syntax

Bis zur Oracle-Datenbankversion 11g war es mit der Oracle-nativen SQL-Syntax nicht möglich, zwei oder mehr Tabellen mit einer anderen Tabelle über OUTER JOIN zu verknüpfen. Denn eine mit NULL-Werten generierte Tabelle konnte bei einem OUTER JOIN nur mit einer Tabelle verknüpft werden. Ansonsten lieferte Oracle die Fehlermeldung ORA-01417.

Ab Oracle 12c ist die Einschränkung bei der nativen OUTER JOIN SQL-Syntax aufgehoben (siehe Abbildungen 4 und 5).

ANSI-JOIN-Erweiterungen

Im Bereich der ANSI-Syntax gibt es ab Oracle 12c die folgenden drei neuen Varianten:

- CROSS APPLY
- OUTER APPLY
- LATERAL

Der CROSS APPLY entspricht im Grundsatz dem ANSI CROSS JOIN. Der OUTER APPLY ist hingegen ein Synonym für ANSI LEFT OUTER JOIN (siehe Abbildung 6). Mit LATERAL stellt Oracle eine weitere Variante der einfachen JOIN-Verknüpfung zur Verfügung.

Die Besonderheit der drei neuen Varianten ist die Möglichkeit der Verwendung von Left Correlation. Hiermit ist gemeint, dass z.B. innerhalb einer Inline View aus einer FROM-Klausel auf Werte von anderen Tabellen aus der FROM-Klausel Bezug genommen werden kann (siehe Abbildung 7).

Bei CROSS APPLY und OUTER APPLY besteht weiterhin die Möglichkeit in einer FROM-Klausel Collections ohne die TABLE-Funktion zu verwenden. Diese Collections lassen sich in einem weiteren Schritt mit Hilfe von Left Correlation und z.B. einer Stored Function mit Werten aus einer Tabellen aus der FROM-Klausel parametrisieren.

Temporal Validity

Mit Oracle 12c wird das Thema „Zeit“ auf zwei verschiedene Weisen betrachtet. Neben der bekannten Flashback-Funktionalität, die sich mit der Transaktionszeit auseinandersetzt, gibt es jetzt die Business-Zeit. Letztere wird durch die neue Temporal-Validity-Funktionalität realisiert. Diese erlaubt es, Datensätze mit einem Gültigkeitszeitraum zu verknüpfen. So kann zum Beispiel die begrenzte Gültigkeit eines Angebotes, einer Mitgliedschaft, Zertifikats oder Sonstiges auf einfache Weise realisiert werden.

Die Abbildung 8 zeigt die Erstellung einer Tabelle, die verschiedene Projekte einer Firma beinhalten soll. Den Tabellendaten wird mit der Spalte `projekt_start`, `projekt_ende` und dem Schlüsselwort `PERIOD` eine Gültigkeitsperiode zugeordnet.

SQL-Abfragen auf Tabellen mit Temporal-Validity-Daten können mit der Angabe einer Periode erweitert werden und somit z.B. gezielt nach gültigen Daten filtern. Dafür wird die `FLASHBACK_QUERY_CLAUSE` verwendet, deren Syntax erweitert wurde. So zeigt Abbildung 9 eine Abfrage, die alle Projekte auflistet, die im Jahr 2013 bearbeitet wurden.

Zudem kann festgelegt werden, welche Daten bei einer SQL-Abfrage ohne explizite `FLASHBACK_QUERY_CLAUSE` ausgegeben werden:

```
EXEC DBMS_FLASHBACK_ARCHIVE.enable_at_
valid_time('CURRENT');

SELECT * FROM projekt;
```

Der Befehl sorgt mit dem Stichwort `CURRENT` dafür, dass in der Session nur aktuell gültige/valide Daten ausgegeben werden – also Projekte bei denen das heutige Datum zwischen dem Projektstart und dem Projektende liegt. Die Default-Einstellung für eine Session ist `ALL`. Alternativ kann mit `AS OF` auch ein bestimmter Zeitpunkt angegeben werden.

Temporal Validity kann auch mit der Flashback-Funktion kombiniert werden. Dadurch sind bitemporale Abfragen möglich. Dies sind quasi Zeitreisen mit Oracle auf zwei Ebenen: Es ist möglich einen alten Datenbestand aus der Vergangenheit zu einem Zeitpunkt zu betrachten und gleichzeitig anhand der Validität (fachliche Gültigkeit) zu filtern, welche Projekte in einem Jahr aktuell sind.

Fazit

Insgesamt wurden einige interessante Neuerungen in der Version 12c veröffentlicht, die das Programmieren angenehmer gestalten. Vor allem mit Hilfe der vielen kleinen Verbesserungen können nun mit einfachen SQL-Befehlen Problemfälle gelöst werden, die in Vorgängerversionen aufwendige Alternativen benötigten. Darüber hinaus bietet Oracle viele neue Funktionen, die vor allem für mehr Kompatibilität sorgen.



Tobias Ummler
(info@ordix.de)

Glossar

ANSI SQL

SQL-Norm des American National Standards Institute

Hidden Column

Tabellenspalten, die meist für Oracle-interne Daten verwendet werden und nicht selektiert werden können

Identity Column

Tabellenspalte mit Werten, die von der Datenbank generiert sind und typischerweise für Primary Key Constraints verwendet werden

Inkrement

Schrittweise Erhöhung einer Größe oder Variable, i.d.R. um den Faktor 1

LOB

Ein Large Object wird verwendet, um große und unstrukturierte Daten, wie z.B. Videos, Fotos oder Audio-Dateien zu speichern.

rownum

Pseudospalte, welche die Reihenfolgenummer des aktuellen Datensatzes in der Anfrage wiedergibt

SCN

Die Oracle System Change Number ist ein logischer, interner Zeitstempel.

Links

- ▶ [1] ORDIX® news Artikel 3/2013
„Neuerungen in der Oracle Database 12 c (Teil I) - Eine für alle, alle in einer“:
http://www.ordix.de/images/ordix/onews_archiv/3_2013/ORDIX_news_3_2013_opf_files/WebSearch/page0034.html
- ▶ [2] ORDIX® news Artikel 4/2013
„Neuerungen in der Oracle Database 12 c (Teil II) - Multitenant-Architektur - eine Datenreise durch Raum und Zeit“:
http://www.ordix.de/images/ordix/onews_archiv/4_2013/ORDIX_news_4_2013_opf_files/WebSearch/page0031.html
- ▶ [3] Oracle Database New Features Guide 12c:
http://docs.oracle.com/cd/E16655_01/server.121/e17906/chapter1.htm#NEWFT001
- ▶ [4] Oracle Database SQL Language Reference 12c:
http://docs.oracle.com/cd/E16655_01/server.121/e17209/toc.htm
- ▶ [5] Seminarempfehlung: Oracle 12c Neuheiten
<http://training.ordix.de/siteengine/action/load/nr/1527/index.html>

Datenbanken

März - Juni 2014

DB-DB-03	Data Warehouse Grundlagen	3 Tage	1.290,00 €	28.04.2014
DB-ORA-01	Oracle SQL	5 Tage	1.890,00 €	31.03.2014 23.06.2014
DB-ORA-01A	Oracle SQL für Experten	3 Tage	1.290,00 €	24.03.2014
DB-ORA-02	Oracle Datenbankprogrammierung mit PL/SQL Grundlagen	5 Tage	1.890,00 €	07.04.2014 30.06.2014
DB-ORA-34	Oracle Datenbankprogrammierung mit PL/SQL Aufbau	5 Tage	1.890,00 €	26.05.2014
DB-ORA-42	Oracle PL/SQL Tuning	3 Tage	1.890,00 €	24.03.2014 16.06.2014
DB-ORA-03	Oracle Datenbankadministration Grundlagen	5 Tage	1.990,00 €	07.04.2014
DB-ORA-04	Oracle Datenbankadministration Aufbau	5 Tage	1.990,00 €	12.05.2014
DB-ORA-32	Oracle Backup und Recovery mit RMAN	5 Tage	1.990,00 €	19.05.2014
DB-ORA-07	Oracle Tuning und Monitoring	5 Tage	1.990,00 €	23.06.2014
DB-ORA-41	Oracle AWR und ASH Analyse und Interpretation	3 Tage	1.290,00 €	14.04.2014
DB-ORA-11	Oracle Troubleshooting Workshop	5 Tage	1.990,00 €	weitere Termine unter www.training.ordix.de
DB-ORA-08	Oracle 11gR2 RAC und Grid Infrastructure	5 Tage	2.050,00 €	05.05.2014
DB-ORA-33B	Oracle Security	5 Tage	1.890,00 €	23.06.2014
DB-ORA-31	Oracle Data Guard	5 Tage	1.990,00 €	19.05.2014
DB-ORA-35	Oracle Cloud Control	3 Tage	1.290,00 €	14.04.2014
DB-ORA-40	Oracle Capacity Planning	3 Tage	1.290,00 €	28.04.2014
DB-ORA-46	Oracle APEX Anwendungsentwicklung Grundlagen	3 Tage	1.290,00 €	26.05.2014
DB-ORA-47	Oracle APEX Anwendungsentwicklung für Fortgeschrittene	3 Tage	1.290,00 €	24.03.2014 10.06.2014
DB-ORA-48	Oracle Golden Gate	3 Tage	1.290,00 €	16.06.2014
DB-ORA-49	Oracle 12c Neuheiten	5 Tage	1.990,00 €	31.03.2014 30.06.2014
DB-INF-01	IBM Informix SQL	5 Tage	1.790,00 €	12.05.2014
DB-INF-02	IBM Informix Administration	5 Tage	1.990,00 €	19.05.2014
DB-INF-04	IBM Informix Backup und Recovery	3 Tage	1.290,00 €	26.05.2014
DB-INF-03	IBM Informix Tuning und Monitoring	5 Tage	1.990,00 €	weitere Termine unter www.training.ordix.de
DB-DB2-01	IBM DB2 für Linux/Unix/Windows SQL Grundlagen	5 Tage	1.890,00 €	31.03.2014
DB-DB2-02	IBM DB2 für Linux/Unix/Windows Administration	5 Tage	1.990,00 €	07.04.2014
DB-DB2-05	IBM DB2 für Linux/Unix/Windows Monitoring und Tuning	3 Tage	1.290,00 €	05.05.2014
DB-MY-01	MySQL Administration	3 Tage	1.190,00 €	12.05.2014
MS-SQL-01	Querying SQL Server 2012	5 Tage	1.890,00 €	05.05.2014
MS-SQL-02	Administering Microsoft SQL Server 2012	5 Tage	1.890,00 €	12.05.2014
MS-SQL-03	Schreiben von Abfragen mit MS SQL Server 2008 T-SQL	3 Tage	1.190,00 €	23.06.2014
MS-SQL-04	Verwalten einer MS SQL Server 2008 Datenbank	5 Tage	1.890,00 €	31.03.2014 30.06.2014

Entwicklung

März - Juni 2014

P-PHP-01	PHP Programmierung Grundlagen	5 Tage	1.690,00 €	19.05.2014
P-PERL-01	Perl Programmierung Grundlagen	5 Tage	1.690,00 €	23.06.2014
P-PERL-02	Perl Programmierung Aufbau	5 Tage	1.690,00 €	24.03.2014
P-UNIX-01	Shell, Awk und Sed	5 Tage	1.690,00 €	02.06.2014
P-XML-01	Einführung in XML	3 Tage	1.190,00 €	26.05.2014

Web- und Application-Server

März - Juni 2014

INT-04	Apache Web-Server Installation und Administration	3 Tage	1.190,00 €	16.06.2014
INT-07	Tomcat Konfiguration und Administration	3 Tage	1.190,00 €	26.05.2014
INT-08	WebSphere Application Server Installation und Administration	3 Tage	1.390,00 €	10.06.2014
INT-11_7	Administration und Konfiguration für JBoss 7	3 Tage	1.190,00 €	24.03.2014 30.06.2014
DB-ORA-50	Oracle Weblogic Administration Grundlagen	3 Tage	1.390,00 €	14.04.2014

Informationen und Anmeldung

Zentrale:
ORDIX AG
 Westernmauer 12 - 16
 33098 Paderborn
 Tel.: 05251 1063-0

Seminarzentrum:
ORDIX AG
 Kreuzberger Ring 13
 65205 Wiesbaden
 Tel.: 0611 77840-00

**Online-Anmeldung,
 aktuelle Seminarinhalte
 und Termine unter:**
<http://training.ordix.de>



Unser Seminarstandort ist Wiesbaden.
 Die hier angegebenen Termine entsprechen dem
 ersten Seminartag.

Die Preise gelten pro Seminar pro Teilnehmer in Euro
 zzgl. ges. MwSt., Inhouse-Preise auf Anfrage.

Betriebssysteme und Monitoring

März - Juni 2014

BS-01	Unix/Linux Grundlagen für Einsteiger	5 Tage	1.690,00 €	05.05.2014
BS-25	Unix Aufbau-seminar für Datenbank- und Applikationsbetreuer	5 Tage	1.790,00 €	07.04.2014
BS-02	Linux Systemadministration	5 Tage	1.690,00 €	19.05.2014
BS-09	Linux Hochverfügbarkeits-Cluster	5 Tage	1.890,00 €	12.05.2014
BS-03-11	Solaris 11 Systemadministration Grundlagen	5 Tage	1.990,00 €	31.03.2014
BS-04-11	Solaris 11 Systemadministration Aufbau	5 Tage	1.990,00 €	12.05.2014
BS-06-11	Solaris 11 für erfahrene Unix/Linux-Umsteiger	5 Tage	1.990,00 €	02.06.2014
BS-24	Solaris 11 Administration Neuheiten	3 Tage	1.290,00 €	14.04.2014 30.06.2014
BS-18	Solaris Virtualisierung mit ZFS und Container (Zonen)	5 Tage	1.990,00 €	07.04.2014
BS-23	Solaris Virtualisierung mit LDOM	3 Tage	1.290,00 €	10.06.2014
AIX-01	IBM AIX Systemadministration Grundlagen	5 Tage	1.990,00 €	02.06.2014
AIX-02	IBM AIX Installation, Backup und Recovery mit NIM	3 Tage	1.290,00 €	28.04.2014
SM-NAG-01	Systemüberwachung mit Nagios Grundlagen	3 Tage	1.190,00 €	19.05.2014
SM-NAG-02	Systemüberwachung mit Nagios Aufbau	2 Tage	890,00 €	22.05.2014

Projekt-/IT-Management

März - Juni 2014

PM-01	IT-Projektmanagement - Methoden und Techniken	5 Tage	1.990,00 €	05.05.2014
PM-06	Projekte souverän führen - Systemisches Projektmanagement	4 Tage	1.850,00 €	19.05.2014
PM-08	Agiles Projektmanagement mit SCRUM	2 Tage	1.100,00 €	24.03.2014 07.04.2014 16.06.2014
PM-08-Z	SCRUM Praxis und Zertifizierung	1 Tag	250,00 €	auf Anfrage
PM-10	IT-Controlling	3 Tage	1.690,00 €	02.06.2014
PM-07	Krisenmanagement in Projekten	2 Tage	1.100,00 €	07.04.2014
PM-11	Konfliktmanagement	2 Tage	1.100,00 €	09.04.2014
PM-12	Stresskompetenz für IT-Führungskräfte	1 Tag	550,00 €	07.05.2014
PM-13	Stresskompetenz für IT-Mitarbeiter	1 Tag	550,00 €	05.05.2014
MGM-02	IT-Architekturen	3 Tage	1.650,00 €	weitere Termine unter www.training.ordix.de
MGM-07	IT-Strategien effizient entwickeln	2 Tage	1.100,00 €	26.05.2014
MGM-03	IT-Management	5 Tage	1.990,00 €	23.06.2014
MGM-05	IT-Risikomanagement	3 Tage	1.650,00 €	12.05.2014
PM-05	IT-Projektcontrolling	3 Tage	1.290,00 €	07.04.2014 16.06.2014
MGM-04	Geschäftsprozessmanagement	3 Tage	1.650,00 €	10.06.2014

Java/JEE

März - Juni 2014

E-SWA-01	Softwarearchitekturen	5 Tage	1.890,00 €	24.03.2014 23.06.2014
OO-01	Einführung in die Objektorientierte Programmierung	3 Tage	1.190,00 €	02.06.2014
P-JAVA-01	Java Programmierung Grundlagen	5 Tage	1.690,00 €	07.04.2014
P-JAVA-03	Java Programmierung Aufbau	5 Tage	1.690,00 €	12.05.2014
P-JAVA-11	Java 8 Neuheiten	2 Tage	990,00 €	29.04.2014
P-JAVA-12	Java EE kompakt - Power Workshop	5 Tage	1.690,00 €	31.03.2014 02.06.2014
P-JEE-01	JEE für Entscheider	1 Tag	590,00 €	28.04.2014
P-JEE-03A	JSP und Servlet Programmierung	5 Tage	1.590,00 €	26.05.2014
P-JEE-05	Web-Anwendungen mit JavaServer Faces (JSF)	5 Tage	1.590,00 €	05.05.2014
INT-05	Java Web Services	3 Tage	1.190,00 €	10.06.2014
P-JEE-08	Java Performance Tuning	3 Tage	1.290,00 €	16.06.2014



Ihr Wunschseminar ist nicht dabei? Dann nehmen Sie Kontakt mit uns auf. Wir beraten Sie gerne individuell und kompetent. Fordern Sie unser **Seminarprogramm 2014** mit über 100 verschiedenen Seminaren und über 340 Terminen an. Kontaktieren Sie uns über www.ordix.de

Sie benötigen nur z.B. die Kompetenz-Seminare im Bereich Projekt-/IT-Management? Kein Problem. Wir haben für Sie diese Weiterbildungen in einer Spezialausgabe zusammengestellt. In Kürze werden weitere folgen!

Äußere Faktoren und Integration

Der Erfolg bei der Anwendung von Scrum hängt in großem Maße davon ab, wie nahtlos sich das agile Framework in die Organisation integrieren lässt und sich dort etablieren kann. Im letzten Teil unserer Artikelreihe behandeln wir daher einige wichtige Aspekte, um einen nachhaltigen Erfolg beim Einsatz von Scrum im Unternehmen sicherzustellen.

Die erste Herausforderung: Integration in die Governance

In größeren Organisationen gibt die Unternehmensführung die wesentlichen Richtlinien zu Projektvorgehensweisen, Qualitätsmanagement und IT-Reifegradmodellen vor. Existieren bereits etablierte Vorgaben zu klassischen Vorgehensmodellen, stellt die Einbettung agiler Projekte in ein klassisches Projektumfeld eine besondere Herausforderung dar. Wie und wann sollen klassische Quality Gates eingehalten werden, wenn man in der Praxis mit Scrum primär in kurzen Sprint-Zyklen fortschreitet?

Die Lösung: Skalierte Quality Gates

Klassische Vorgehensmodelle orientieren sich bevorzugt am Wasserfallmodell. In diesem wird jede der relevanten Projektphasen Initialisierung, Analyse, Design, Realisierung, Test und Einführung jeweils nur einmal durchlaufen. Bei agilen Vorgehensweisen geht man hingegen iterativ und inkrementell vor. Die Phasen Analyse, Design, Realisierung, Test und Einführung werden pro Iteration meist komplett in einem Sprint durchlaufen, um ein potentiell auslieferbares Teilprodukt zu erstellen.

Streng betrachtet müssten nun zur Einhaltung der Richtlinien in klassischen Vorgehensmodellen der Abschluss aller Phasen erst nach der kompletten Abarbeitung aller Sprints gemeldet werden. Dies wiederum verzerrt den Fortschrittstatus durch formal nicht abgeschlossene externe Quality Gates trotz erreichter Sprint-Ziele. Im Unternehmen selbst kann man nicht erkennen, wie weit und in welcher Qualität ein agiles Projekt tatsächlich fortgeschritten ist.

Zur Schaffung von mehr Transparenz empfiehlt es sich daher, skalierte Quality Gates einzuführen. Pro Phase wird ermittelt, mit welchem prozentualen Anteil der Fortschritt in Bezug auf das zu erwartende Ergebnis durchlaufen wurde. Für eine optimale Transparenz bietet sich die Nutzung einer auf den Gesamtanteil harmonisierten 10er-Skala an, die pro Phase angewendet wird. Anhand der Skalen wird damit von außen klar sichtbar, wie hoch der jeweilige Fortschritt im Bezug auf die zu berichtenden Quality Gates ist.

In der Abbildung 1 wird die Darstellung der Skaleninhalte beispielhaft illustriert. So signalisieren die Skalenwerte 5 in einer Realisierungsphase und 8 in einer Testphase einen bereits zur Hälfte erreichten Gesamtfortschritt in der Entwicklung mit bereits zu 80 % getesteten Funktionalitäten über alle bereits durchlaufenen Sprints.

Aus externer Controllingsicht gilt es hierbei, weniger auf die Komplettierung einzelner Quality Gates, sondern vielmehr auf die relativen Ergebnisfortschritte gegenüber den bereits durchlaufenen Sprints zu achten. Dies hat zudem den Vorteil, dass der anteilige Gesamtfortschritt in den betrachteten Projekten viel transparenter wird.

Die zweite Herausforderung: Interaktion mit der Linienorganisation

Der Erfolg von Scrum ist sehr abhängig von einem stark ausgeprägten Team-Gedanken mit einer möglichst ausgewogenen Besetzung an Experten. Ist ein Team unterbesetzt oder sind entscheidende Schlüsselpositionen nicht ausreichend verfügbar, leidet unmittelbar die Velocity und damit zwangsläufig der Gesamtfortschritt des Vorhabens.

Derartige Situationen ergeben sich häufig dann, wenn man im Projekt von externen Zulieferungen abhängig ist. Hierzu bedarf es bereits im Vorfeld einer gezielten Planung von externen Ressourcen aus der Linienorganisation. Wie soll die Planung jedoch erfolgen, wenn man bevorzugt agil vorgehen möchte und sich ein Bedarf erst in den jeweiligen Sprints abzeichnet?

Die Lösung: Kontingentreservierung

In der Kapazitätsplanung sind Schlüsselressourcen der Linienorganisationen in der Regel zu nahezu 100 % ausgelastet. Dies erschwert leider eine kurzfristig angeforderte Verfügbarkeit, um gezielt in Sprints eingesetzt zu werden. Der Schlüssel zu einer funktionierenden Lösung liegt in

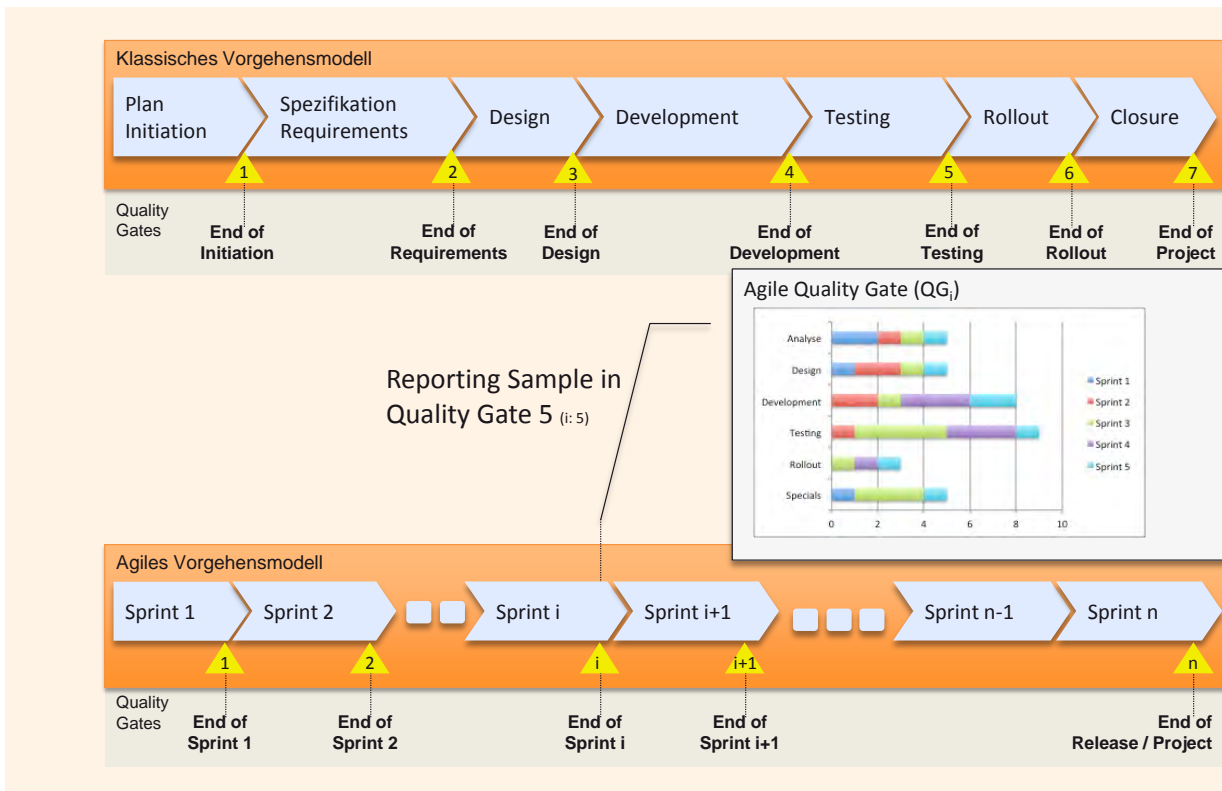


Abb. 1: Skalierte Quality Gates in agilen Projekten.

einer übergreifenden Kapazitätsplanung sämtlicher in Unternehmen bekannten Schlüsselpositionen.

Hierzu wird im Vorfeld aus allen geplanten Vorhaben proaktiv ein übergreifender Bedarf ermittelt und ein Kontingent aus Schlüsselressourcen in einem gemeinsam genutzten Ressourcen-Pool bereitgestellt. Um den kapazitiven Gesamtbedarf zu ermitteln, orientiert man sich am besten an den vorliegenden Release Plannings der jeweils agilen Projekte.

Diese Lösung setzt zwei wesentliche Rahmenbedingungen voraus. Einerseits müssen die verfügbaren Schlüsselressourcen aus den gemeinsam genutzten Pools als „Springer“ in den anfordernden Sprints agieren. Mit anderen Worten werden diese Ressourcen alternierend für jeweils nur wenige Sprints in verschiedenen Projekten eingesetzt. Andererseits bedarf es einer übergeordneten Einheit, welche die zentrale Kapazitätsplanung übernimmt.

Die dritte Herausforderung: Controlling und Kennzahlen

Für die Erfolgssteuerung agiler Vorgehensweisen sind präzise Kennzahlen unabdingbar, insbesondere um den aktuellen Status für alle Stakeholder möglichst transparent zu halten. Doch welche Kennzahlen sind von außen betrachtet sinnvoll und relevant? Was sind die wesentlichen Informa-

<input type="checkbox"/>	Tailoring der Quality Gates an agile Vorgehensweise in Abstimmung mit übergreifendem Qualitätsmanagement
<input type="checkbox"/>	Erstellung von skalierten Quality Gates
<input type="checkbox"/>	Übergreifende und zentrale Kapazitätsplanung von Schlüsselpositionen
<input type="checkbox"/>	Etablierung eines Shared Ressourcen-Pools
<input type="checkbox"/>	Erwägung des Einsatzes von Themenparks zum übergreifenden Reporting an externe Stakeholder
<input type="checkbox"/>	Einsatz von agilen Dashboards pro Projekt

Abb. 2: Checkliste „Äußere Faktoren und Integration“.

tionen in agilen Projekten, die unbedingt nach außen berichtet werden sollen, um ein übergreifendes Controlling in der gesamten Organisation zu erzielen?

Die Lösung: Themenparks und agile Dashboards

Insgesamt schafft die Berichterstattung in Scrum bereits ein sehr hohes Maß an Transparenz. Am Ende jeden Sprints wird der Release-Plan aktualisiert. Dort werden sämtliche Aufwandsänderungen im Product Backlog ebenso wie die Velocity des Teams aktualisiert und berichtet. Damit werden Projektverzögerungen ebenso schnell offengelegt,

Glossar

Dashboard

Ein Dashboard dient der Visualisierung von Informationen in verdichteter, meist grafischer Form.

FDD

Feature-Driven Development ist eine Sammlung von Arbeitstechniken, Strukturen, Rollen und Methoden für das Projektmanagement im Rahmen der agilen Softwareentwicklung. Dabei werden ein Prozess- und ein Rollenmodell definiert, die gut mit existierenden klassischen Projektstrukturen harmonisieren.

Product Backlog

Das Product Backlog ist eine priorisierte Liste, die alles enthält, was im zu entwickelnden Produkt enthalten sein sollte.

Product Owner

Der Product Owner ist verantwortlich für die Anforderungen und deren Priorisierung in Zusammenarbeit mit dem Auftraggeber und Know-how-Träger.

Quality Gates

Dies sind Punkte im Ablauf eines Entwicklungsprojekts, bei denen anhand von im Voraus eindeutig bestimmten Qualitätskriterien über die Freigabe des nächsten Projektschrittes entschieden wird.

Release Planning

Ein Release Planning beinhaltet die Planungen von Veröffentlichungen und Erweiterungen vorhandener Versionen, die meistens neue Module, Prozesse oder Funktionsblöcke enthalten. Ein Release kann auch durch verschiedene kleinere Erweiterungen entstehen.

Scrum

Scrum ist eine Methode für das agile Projektmanagement. Der Ansatz von Scrum ist empirisch, inkrementell und iterativ.

Sprint

Iteration fester Länge zur Umsetzung von Anforderungen

Velocity

Metrik für die Entwicklungsgeschwindigkeit innerhalb eines Sprints

Links

- ▶ [1] ORDIX® news Artikel 2/2013
„Scrum - aber richtig! (Teil I) - Optimale Abläufe und innere Faktoren“:
http://www.ordix.de/images/ordix/onews_archiv/2_2013/ORDIX_news_2_2013_opf_files/WebSearch/page0047.html
- ▶ [2] ORDIX® news Artikel 3/2013
„Scrum - aber richtig! (Teil II) - Kunde und Product Owner“:
http://www.ordix.de/images/ordix/onews_archiv/3_2013/ORDIX_news_3_2013_opf_files/WebSearch/page0010.html
- ▶ [3] Scrum Alliance:
<http://www.scrumalliance.org/>
- ▶ [4] Agiles Manifest:
<http://agilemanifesto.org/iso/de/>
- ▶ [5] Webseite der coniatos AG:
www.coniatos.de
- ▶ [6] Projektmanagement-Seminare der coniatos AG/ORDIX AG:
<http://training.ordix.de/siteengine/action/load/kategorie/Projekt-und%20IT-Management/index.html>

wie Probleme und Risiken früh sichtbar gemacht. Darüber hinaus bietet sich für ein transparentes Reporting des Fertigstellungsgrades der sogenannte Themenpark an. Dieser ist eine Berichterstattungsform, die ursprünglich aus dem Feature-Driven Development (FDD) stammt.

In Themenparks werden inhaltlich zusammengehörige Anforderungen zu Themen gruppiert. Die Basis für die Bildung von Themen stellt funktional zusammenhängende Stories aus dem Product Backlog dar. Das Reporting in Form von Themenparks ist besonders für den Product Owner von großem Nutzen, um die Abnahmefähigkeit von Funktions-Clustern besser zu erkennen.

Um die Transparenz auch in Richtung Auftraggeber und Top-Management zu erhöhen, empfiehlt es sich, die folgenden Kennzahlen in ein agiles Dashboard aufzunehmen:

- **Project Progress Schedule**
Verbrauchte Zeit ggf. Gesamtzeit in Prozent
- **Business Value**
Gelieferte Business Value Points der DONE User Stories in Prozent
- **Team Performance**
Erreichte Team Velocity in relativen Tendenzwerten
- **Top-5-Risiken und -Aufgaben**
aus dem Sprint-Reporting

Fazit

Die Stärken von Scrum liegen bei der Umsetzung von komplexen Projekten eindeutig in der schnell gewonnenen Produktivität durch die adaptive Vorgehensweise. Dennoch sollte man nicht unterschätzen, wie wichtig ein transparentes Reporting zur äußeren Managementsicht ist. Für die Auftraggeber, die Liniorganisation sowie das Top-Management ist eine inhaltlich qualitative Korrelation zu den Ergebnissen der „inneren Sprint-Sichten“ unverzichtbar. Nur so lässt sich der Projektfortschritt und der inhaltliche Ergebnisstand von außen erkennen und bildet eine tragfähige Basis für weitere Entscheidungen.

In unserer Checkliste aus Abbildung 2 finden Sie zusammengefasst die wesentlichen Faktoren, um die Transparenz für agile Projekte in Ihrer Organisation zu erhöhen. Für alle äußeren Beteiligten und besonders für den Auftraggeber sollte der Blick auf ein agiles Projekt so unverstellt und ergebnisorientiert wie möglich sein.



Matthias Heint
(info@coniatos.de)

Kalenderberechnung in Oracle-Datenbanken - Lösungsansätze

Larry und sein Geburtstagskalender



In der Ausgabe 3/2013 der ORDIX® news hatten wir nach einem Algorithmus gesucht, der alle Personen anzeigt, die heute oder in den nächsten sechs Tagen Geburtstag haben. Leider hat Larry nur eine Einsendung erhalten. In diesem Artikel stellen wir Ihnen daher drei unterschiedliche Verfahren zur Lösung des Problems vor (die Abbildungen finden Sie im Internet unter www.ordix.de/ordixnews/larry).

Die Aufgabe

Gegeben ist eine Liste von Geburtstagen (Abbildung 1). Gesucht sind alle Personen, die heute oder in den nächsten 6 Tagen Geburtstag haben. Zugegebenermaßen war Larry's Spezifikation nicht ganz vollständig. Wir wollen deswegen weiterhin annehmen, dass die am 29. Februar Geborenen in Nicht-Schaltjahren am 28.02. ihren Geburtstag feiern. Zur Erinnerung sei kurz erwähnt, dass alle Jahre die durch 4, nicht durch 100 aber durch 400 teilbar sind, Schaltjahre sind. So ist das Jahr 1900 kein Schaltjahr (auch wenn Excel etwas anderes behauptet), das Jahr 2000 dagegen schon.

Zwei grundlegende Probleme sind bei der Aufgabe zu lösen. Zum einen muss der Jahreswechsel gelingen, das heißt am 28.12. müssen auch die Geburtstage bis zum 3. Januar angezeigt werden. In Nicht-Schaltjahren müssen die Geburtstage vom 29.02. am 28.02. eingeordnet werden. In Schaltjahren muss der 29.02. als „normaler“ Tag betrachtet werden.

Lösung 1 - Der nächste Geburtstag

In der ersten Lösung suchen wir den nächsten Geburtstag zu einem Datum. Hierzu ermitteln wir, ob der Geburtstag ohne Jahr (MMDD) kleiner ist als der aktuelle Tag (MMDD). Ist dies der Fall, wird der Geburtstag auf das nächste Jahr festgelegt. Andernfalls ist er in diesem Jahr. Versucht man allerdings, den 29.02 für ein Nicht-Schaltjahr zu generieren, so gibt Oracle die Fehlermeldung ORA-01839 aus. Mit Hilfe eines Exception Handler wird dann aus dem 29.02. der 28.02. gemacht (Abbildung 2).

Im letzten Schritt wird einfach das aktuelle Datum vom generierten Datum abgezogen und das Ergebnis ist die Anzahl Tage bis zum nächsten Geburtstag. Diese Lösung hat den Vorteil, dass sie nicht auf die 6 Tage beschränkt ist.

Lösung 2 - Die Tagesliste

Die zweite Lösung verfolgt einen ganz anderen Ansatz. Für heute und die nächsten sechs Tage werden die Tage in der Form MMDD ermittelt. Wenn kein Schaltjahr ist, wird der

29.02. hinzugefügt, wenn der 28.02. in der Liste enthalten ist. Mit Hilfe der `IN`-Klausel wird geprüft, ob der Geburtstag (MMDD) in dieser Liste enthalten ist (Abbildung 3).

Diese Lösung hat den Vorteil, dass sie relativ einfach auf beliebige n-Tage und auf m-vergangene Tage erweitert werden kann.

Lösung 3 - Für Kalenderexperten

Die dritte Lösung ist für Freunde der Kalenderlogik. Im ersten Schritt werden die Schalttage zwischen dem Tag der Geburt und dem folgendem Geburtstag ermittelt. Die Schalttage bis zu einem Jahr (Abbildung 4) lassen sich mit dem folgenden Ausdruck ermitteln:

```
floor((pi_year) / 4) - floor((pi_year) / 100) + floor((pi_year) / 400)
```

Im nächsten Schritt wird die Anzahl Tage zwischen dem Tag der Geburt und dem aktuellen Datum ermittelt. Davon werden dann die Schalttage abgezogen. Anschließend wird mit der Funktion Modulo 365 die Anzahl der Tage bis zum nächsten Geburtstag ermittelt. Historisch ist dieser Ansatz nicht geeignet, da im Jahre 1582 aufgrund der Umstellung vom julianischen auf den gregorianischen Kalender die Tage zwischen dem 04.10. und dem 15.10.1582 fehlen.

Fazit

Wie immer führen viele Wege nach Rom. Mit Hilfe der hier vorgestellten Verfahren wurden drei Wege aufgezeigt, wie bestimmt kein Geburtstag mehr auf dem Kalender fehlt. Gratulieren müssen Sie natürlich noch selbst und hoffentlich auch ganz persönlich.



Martin Hoermann
(info@ordix.de)



Ganzheitliches Datenqualitätsmanagement – Warum und wofür?

„Qualität ist nicht alles, aber ohne Qualität ist alles nichts.“ Diese Zitat von Prof. Dr. Walter Masing, Ehrenvorsitzender der Deutschen Gesellschaft für Qualität e.V., beschreibt in Kurzform worum es bei der Einhaltung von Qualitätsstandards geht. Der Begriff „Datenqualität“ definiert die Eignung von Daten, welche die Realität beschreiben. Hierbei beschreibt der Begriff im Besonderen, wie verlässlich Daten für die Grundlage weiterer Vorgehensweisen und als Entscheidungshilfe genutzt werden können.

Motivation

Eine gesunkene Datenqualität kann mehrere Ursachen haben. So hat sich zum einen das zu verarbeitende Datenvolumen in den letzten Jahren deutlich erhöht und zum anderen sind die Daten durch die Zusammenführung unterschiedlicher Quellen, z.B. in einem Data Warehouse, noch komplexer geworden. Darüberhinaus werden in operativen Systemen zwar unterschiedliche Mechanismen eingesetzt, welche die Datenqualität über Plausibilisierungsregeln prüfen. Meist verhindern diese aber nicht z.B. eine doppelte Eingabe von Datensätzen mit gleichem Inhalt. Um dem negativen Effekt einer kontinuierlichen Verunreinigung der Datenbestände entgegenzuwirken, sollten sich Unternehmen daher frühzeitig Gedanken über ein ausreichendes Datenqualitätsmanagement machen.

Ohne ein präventives Datenqualitätsmanagement wird meist erst bei Überschreiten eines kritischen Punktes gehandelt und versucht, durch Add-Hoc-Lösungen eine Verbesserung zu erzielen. Dieser Eingriff stellt aber nur eine punktuelle Verbesserung dar und wird beim nächsten Zyklus durch negative Auswirkungen wieder überlagert.

Solche kurzfristigen Maßnahmen sind zudem schlecht planbar und binden das Personal der Fachabteilungen. Um nicht nur auf Probleme zu reagieren, sondern schon im Vorfeld mögliche Auswirkungen zu unterbinden, ist es ratsam sich auf eine Vorgehensweise zur Erhaltung der Datenqualität zu einigen. Die Nutzung von Daten mit schlechter Qualität zieht besonders durch Folgeaufwände laufend unnötige Kosten nach sich (siehe Abbildung 1).

Die Datenqualität ist ein wesentlicher Faktor für die erfolgreiche Entscheidungspolitik eines Unternehmens. Mangelnde, ungenaue oder fehlerhafte Daten führen zu falschen Entscheidungen und damit verbunden zu negativen Auswirkungen für ein Unternehmen.

Von zentraler Bedeutung ist die Datenqualität, wenn es um die Einhaltung der gesteigerten gesetzlichen oder regulatorischen Vorschriften wie z.B. SOX, Solvency oder Basel geht. Sie ist weiterhin ein wesentlicher Erfolgsfaktor in automatisierten und digitalisierten Arbeitsabläufen.

Die häufigsten Ursachen für eine mangelhafte Datenqualität sind oft fehlende Verantwortlichkeiten, menschliche Fehler, regionale Interpretationsunterschiede oder redundant gespeicherte Kopien von Daten (Dubletten).

Die nachfolgenden Beispiele illustrieren die oben genannte Problematik:

- Stammdatenduplikate führen zu einer doppelten Lagerhaltung und einer fehlerträchtigen Einkaufssteuerung.
- Kunden werden mehrfach angeschrieben oder nicht erreicht, weil der Personenstamm Duplikate aufweist oder eine fehlerhafte Adresse vorliegt.
- Die eigene Position gegenüber einem Geschäftspartner oder Kunden wird durch fehlerhafte Daten falsch ermittelt und führt zu falschen Entscheidungen.

Vorgehensweise

Der erste Schritt in einem Projekt zur Steigerung der Datenqualität sollte immer sein, Klarheit über den aktuellen Qualitätsstand der Daten und die damit verbundenen Ziele zu erhalten. Dies bedingt die Definition der Kriterien, anhand derer die Qualität bewertet werden soll. Nicht immer sind alle denkbaren Dimensionen der Datenqualität relevant.

Bei der allgemeinen Definition der Datenqualität sind Kriterien wie Zugangsqualität, Darstellungsqualität, etc. zu bewerten. Diese allgemeinen Kriterien decken alle Punkte der Informationsverarbeitung sowohl in Datenbanken als auch im Dokumentenmanagement ab. Der Focus dieses Artikels bezieht sich primär auf (semi-) strukturierte Daten.

Dimensionen der Datenqualität

Die Bewertung der Datenqualität kann hinsichtlich verschiedener Aspekte betrachtet werden. Eine Einteilung in Dimensionen hilft dabei zur Einordnung und Messung (siehe Abbildung 2). Die zu unterscheidenden Aspekte sind allerdings nicht immer trennscharf. Die dargestellte Unterteilung bietet jedoch einen nützlichen Ausgangspunkt zur Analyse.

Im Rahmen der Analyse der Datenqualität werden sogenannte Qualitätsregeln verwendet. Eine Regel kann dabei verschiedene Aspekte abdecken. Ein einfaches Beispiel ist eine Existenzprüfung zur Bestimmung der Füllgrade. Kom-

plexere Regeln können ganze Geschäftsregeln abbilden. So kann ein Feld in einer bestimmten Datenkonstellation (abhängig von anderen Attributen) zu einem Pflichtfeld werden. Ein Beispiel: Für die Personenstammdaten einer Bank ist es erheblich, ob eine Person als potentieller Interessent angesprochen wird oder ob diese Person ein Kunde ist. Diese Unterscheidung ist relevant, da für Kunden im Gegensatz zu Interessenten auch ein Ausweisdokument erfasst werden muss.

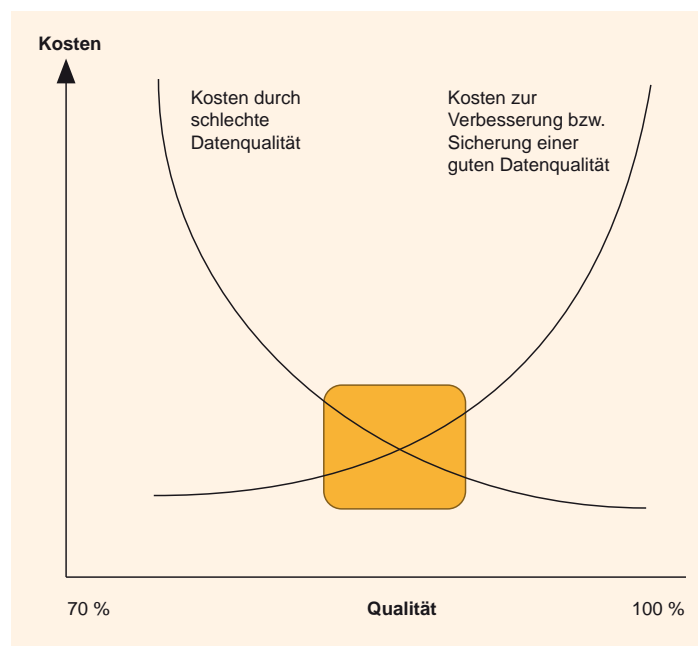


Abb. 1: Beispiel eines Kosten-Nutzen-Vergleichs.

Dimension	Beschreibung
Vollständigkeit	Füllgrade, leere Felder
Korrektheit	Stimmen die gespeicherten Daten mit der Realität überein?
Konsistenz	Schlüsseleindeutigkeit, referentielle Integrität
Konformität/Einheitlichkeit	Einheitliches Format, Typ, Genauigkeit
Aktualität	Sind die Werte veraltet, müssen sie ggf. gelöscht werden?
Redundanzfreiheit	Normalisierung, Verwendung von Referenz und Stammdaten
Transparenz	Gibt es ein geeignetes Metadatenmanagement?
Zugänglichkeit	Verfügbarkeit der Daten
Interpretierbarkeit	Auswertbarkeit der Datenausprägungen

Abb. 2: Dimensionen der Datenqualität.



Abb. 3: Bausteine der Datenqualitätsanalyse.

Datenqualitätsmetriken

Eine Metrik beruht auf der Anwendung von Regeln auf einen Datenbestand. Für die Analyse muss weiterhin beachtet werden, dass die Relevanz der Datenqualität wesentlich von der Verwendung bzw. dem Zweck der untersuchten Daten beeinflusst wird. So macht es beispielsweise einen Unterschied, wenn die verwendeten Daten melderechtlich relevanten Reports zugrundeliegen oder für Marketingmaßnahmen verwendet werden.

Datenqualitätsanalyse

Diese Analyse steht am Anfang einer Datenqualitätsinitiative. Dazu lässt sich das sogenannte Daten-Profiling grob in verschiedene Bausteine unterteilen (siehe Abbildung 3):

Domänenanalyse

Die Domänenanalyse beschreibt die Klassifikation der Datenbestände, zum Beispiel welcher Domäne gehören die Tabellen/Felder an. Dabei werden Spaltenanalysen ausgeführt, welche die Datenformate (z.B. 99-XXX-9999) und die jeweiligen Häufigkeiten ermitteln.

Vollständigkeit und Gültigkeit

Zur Vollständigkeitsanalyse werden die Felder hinsichtlich der Füllgrade untersucht. Dabei werden sowohl NULL- als auch Default-Werte ermittelt und unterschieden.

Wenn die Domänenanalyse z.B. ergibt, dass ein Freitextfeld für ein Datum verwendet wurde, so ist es prinzipiell möglich, dass ungültige Werte wie der 30. Februar verwendet wurden.

Strukturelle Integrität

Hiermit ist die Überprüfung gemeint, ob ein Primärschlüssel definiert wurde bzw. die Erschließung zusammengesetzter Schlüssel. Fremdschlüsselbeziehungen können automatisch erschlossen werden, dabei erkennt eine Cross-Domain-Analyse redundant gehaltene Daten in unterschiedlichen Tabellen.

Semantische Analyse

Durch die semantische Analyse werden Freitexte mit komplexer Mustererkennung analysiert und klassifiziert, dabei erfolgt eine Textstandardisierung und Datenverifikation mit Referenzbeständen (z.B. Adressvalidierung).

Geschäftsregelanalyse

Die Datenbestände werden auf ihre Konformität mit komplexeren Geschäftsregeln untersucht. Hier werden entsprechend des Zustandes eines Objekts (Beispiel: Ist der Kunde volljährig?) abhängige Objekte abgeglichen (Ist ein gesetzlicher Vertreter erfasst?). Generell ist für die Interpretation der Ergebnisse zumeist ein fachlicher Hintergrund zu den betrachteten Ergebnissen gefragt, was in der Regel die aktive Mitarbeit durch den Fachbereich bedingt.

Reifegradmodell für das Datenqualitätsmanagement

Datenqualität ist ein abstrakter Begriff und somit nur schwer greifbar. Die Verwendung eines Reifegradmodells [1] gibt eine Orientierung sowohl hinsichtlich der Einordnung zum aktuellen Stand als auch zur Zielsetzung im Datenqualitätsmanagement. Die unterschiedlichen Stufen zeigt die Abbildung 4:



Abb. 4: Reifegradstufen im Datenqualitätsmanagement.

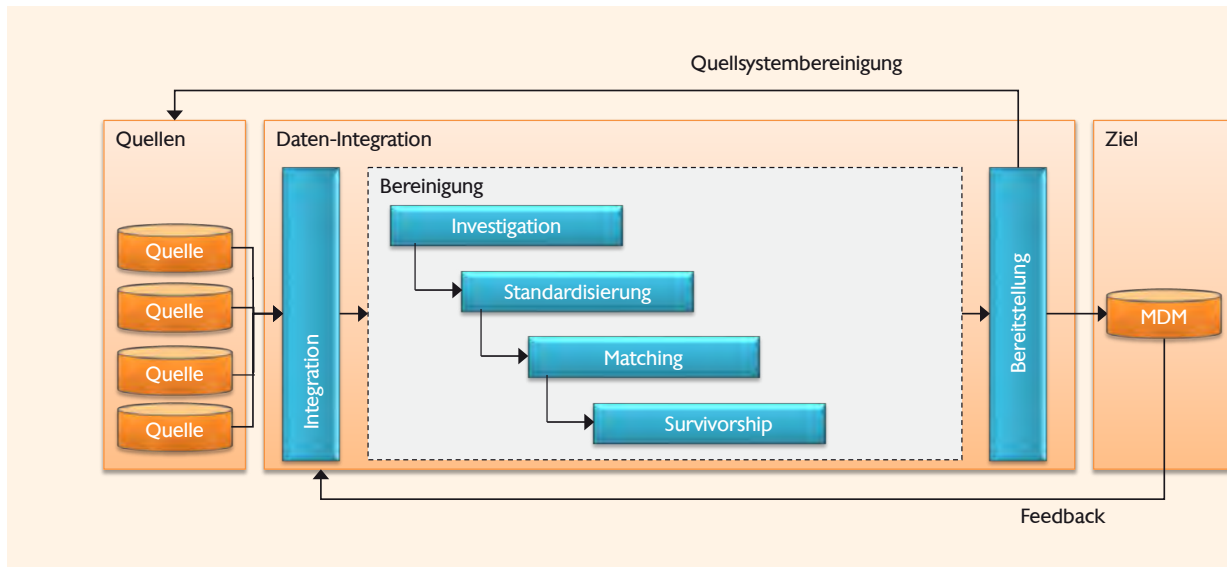


Abb. 5: Daten-Deduplizierung.

Initial

Es besteht nur ein rudimentäres Verständnis von Datenqualität und es sind keine definierten Verantwortlichkeiten festgelegt. Probleme in der Datenqualität werden rein reaktiv behoben, wobei die Datenmodellierung durch die physische Implementierung definiert wird.

Wiederholbar

Die Unternehmensführung erkennt den Wert von Datenqualität, unternimmt jedoch noch keine organisatorischen Maßnahmen um entsprechende Prozesse einzuführen. Die Datenqualität wird mit taktischen, häufig manuellen Maßnahmen adressiert. Dabei wird versucht, über die Datenmodellierung die unterschiedlichen Modelle konsistent zu halten.

Wohldefiniert

Das Datenqualitätsbewusstsein wird unternehmensweit geschaffen und Datenqualitätswerkzeuge werden eingeführt. Gleichzeitig werden Richtlinien für eine Datenqualitätsanalyse und eine Bereinigung der Daten definiert.

Gesteuert

Der Wert eines Datenqualitätsmanagements ist erkannt. Die Datenqualität wird mit klar definierten Verantwortlichkeiten und Richtlinien unterstützt. Mögliche Probleme werden durch Analysen frühzeitig entdeckt. Ein Metadatenmanagement und Glossare werden zusätzlich eingesetzt.

Optimiert

Als Ergebnis wird die optimierte Datenqualität zu einer kontinuierlich verfolgten strategischen Initiative. Prozesse und Metriken werden definiert und etabliert. Diese sollen sicherstellen, dass die Datenqualitätsziele eingehalten werden. Die Informationsarchitektur wird mit unternehmensweit einheitlichen Standards und Richtlinien gelebt.

Verbesserung der Datenqualität

Die Datenqualität lässt sich manuell oder automatisiert verbessern. Dabei lässt sich die automatisierte Bereinigung (Data Cleaning) von Datenbeständen in drei Arten unterscheiden:

- **Hinzufügen von fehlenden Werten**
Das Behandeln von fehlenden Werten lässt sich häufig nicht vollautomatisch bewerkstelligen, da die Datenpflege einer menschlichen Interaktion bedarf. Es gibt jedoch auch Beispiele für eine automatische Bereinigung, z.B. bei Stammdaten mit der Verwendung von extern verfügbaren Referenzdaten wie Adresskatalogen, Spezifikationen, etc.
- **Löschen von Werten**
Der Auslöser zum Löschen von Daten ist das Vorhandensein von redundant gehaltenen Informationen (insbesondere Duplikate) oder veralteten Daten.
- **Löschen von Duplikaten**
Bei transaktionalen Daten genügt es in der Regel, wenn Duplikate vorliegen, nur den letzten gültigen Stand zu behalten, ältere Versionen können gelöscht werden.

Löschung von veralteten Werten

Die Notwendigkeit veraltete Daten zu löschen kann aus rechtlichen Anforderungen resultieren, zum Beispiel aus den gesetzlichen Regelungen zum Datenschutz. So verpflichten sich 450 deutsche Versicherer in den „Verhaltensregeln für den Umgang mit personenbezogenen Daten durch die deutsche Versicherungswirtschaft“ [2], Daten zu löschen sobald „[...] die Kenntnis der Daten für die verantwortliche Stelle zur Erfüllung des Zwecks der Verarbeitung oder Nutzung nicht mehr erforderlich ist.“ Ein zu löschendes Personenstammdatum wäre hier dadurch zu erkennen, dass keine referenzierende Vertragsbeziehung mehr existiert.

Konsolidierung von redundanten Informationen

Bei der Behandlung von redundant gehaltenen Informationen gibt es grundsätzlich die Optionen einen Wert zu löschen oder zu konsolidieren. Eine Konsolidierung bietet sich insbesondere bei Duplikaten von Stammdaten an. Hier ist es häufig der Fall, dass nicht der denkbar beste Datensatz sondern die Kombination der Attribute gefordert ist.

Das Entdecken und Bereinigen von Dubletten ist ein mehrstufiger Prozess, der in mehreren Iterationen durchgeführt wird (siehe Abbildung 5). Hierbei wird der Datenbestand lexikalisch untersucht. Die in der Eingabe enthaltenen Muster werden erschlossen und ihre Häufigkeiten ermittelt. Die ermittelten Muster dienen anschließend zur Auswahl

und Anpassung eines geeigneten Regelwerks zur Standardisierung.

In der Standardisierung werden die Daten in eine einheitliche Struktur überführt. Will man z.B. Namensbestände bereinigen, so teilt man die Namensbestandteile in dedizierte Felder auf, die üblicherweise in Freitexten gemischt vorkommen. Ein Beispiel: Es kann vorkommen, dass in den Quellen Namensbestandteile wie „Dr.“ nicht in eigenen Feldern gehalten sondern teils führend am Anfang, teils mit Komma getrennt am Ende des Namens abgelegt werden. Damit man erkennt, dass sich hinter Herrn „Dr. Schmidt“ und „Schmidt, Dr.“ die gleiche Person verbirgt, muss man somit zunächst den Namensbestandteil vom Nachnamen trennen. Bei Spezialfällen wie Namenszusätzen (Junior/Senior), Adelstiteln, Adressierungshinweisen (2. Stock, z.Hd.,...) fehlt in der Regel jegliche Systematik, wie diese gespeichert werden sollen.

In der Matching-Phase werden die standardisierten Sätze hinsichtlich Duplikate abgeglichen. Für den Umgang mit Tippfehlern oder Abweichungen der Erfassungsmethodik (z.B. Mueller vs. Müller) sollte eine probabilistische Matching Engine verwendet werden. Hierbei wird ein Ähnlichkeitsmaß vorgegeben, um ein Datensatzpaar als Match zu erkennen - es muss also keine hundertprozentige Übereinstimmung vorliegen. Bei der Berechnung von Wahrscheinlichkeiten wird die Häufigkeit mit herangezogen. Zwei Michael Meier müssen nicht unbedingt die gleiche Person sein, Übereinstimmungen bei selteneren Namen sind ein geeigneteres Indiz für ein Duplikat. Am Ende der Matching-Phase ergeben sich Gruppen von Datensätzen, die mit ausreichend hoher Genauigkeit die gleiche Entität darstellen.

In der Konsolidierungsphase (Survivorship) wird aus den Gruppen ein „Golden Record“ gebildet. Dabei werden die Attribute der Gruppenmitglieder zum Resultat kombiniert. Der Prozess der Standardisierung und Deduplizierung erfolgt immer iterativ. Die verwendeten Regelwerke werden sukzessive solange verbessert, bis der gewünschte Gütegrad erreicht ist.

Manuelle Bereinigung

Die manuelle Bereinigung erfolgt in der Regel über die jeweiligen Frontends der betroffenen Systeme, auch kleiner SQL-Batches (SPUFI) können eine Option darstellen. Die Entscheidung, ob manuell oder automatisiert bereinigt wird, hängt von verschiedenen Faktoren ab:

- Sind zukünftig weitere Verunreinigungen ausgeschlossen, da die Ursache bereits behoben wurde oder das Quellsystem im Zuge einer Migration abgeschaltet wird?
- Das Verhältnis des Aufwands bei einer manuellen Bereinigung zu den Aufwänden die Daten während des Migrationsprozesses zu bereinigen
- Dem Zeitpunkt, wann das Datenqualitätsproblem entdeckt wird

Glossar

Basel

Basel umfasst die Gesamtheit der Eigenkapitalvorschriften, die vom Basler Ausschuss für Bankenaufsicht in den letzten Jahren vorgeschlagen wurden. Die Regeln müssen gemäß den EU-Richtlinien seit dem 1. Januar 2007 in den Mitgliedsstaaten der Europäischen Union für alle Kreditinstitute und Finanzdienstleistungsinstitute (= Institute) angewendet werden.

Daten-Profilung

Das Daten Profiling bezeichnet den weitgehend automatisierten Prozess zur Analyse vorhandener Datenbestände (z.B. in einer Datenbank) durch unterschiedliche Analysetechniken.

Metadatenmanagement (MDM)

Die adäquate Bereitstellung der Metadaten auf strategischer, taktischer und operativer Ebene

Solvency

Ein Projekt der EU-Kommission zu einer grundlegenden Reform des Versicherungsaufsichtsrechts in Europa, vor allem der Solvabilitätsvorschriften für die Eigenmittelausstattung von Versicherungsunternehmen

SOX

Diese Abkürzung steht für Sarbanes-Oxley Act, ein amerikanisches Bundesgesetz. Ziel des Gesetzes ist es, das Vertrauen der Anleger in die Richtigkeit und Verlässlichkeit der veröffentlichten Finanzdaten von Unternehmen wiederherzustellen.

Auch wenn die Bereinigung an sich manuell erfolgt, sollte in jedem Fall werkzeuggestützt überwacht werden, ob die Maßnahme erfolgreich war. Dazu wird idealerweise das initiale Daten-Profiling in eine kontinuierliche Überwachung überführt.

Fazit

In den kommenden Jahren werden sich die Datenmengen um ein Vielfaches erhöhen. Gleichzeitig wird die Anzahl an IT-Fachkräften - wenn überhaupt - nur unterproportional steigen. Dies bedeutet einen noch größeren Aufwand für Add-Hoc-Lösungen und erhöht damit die Notwendigkeit für ein durchdachtes Datenqualitätsmanagement. Denn wenn man mit Add-Hoc-Lösungen in „irgendeiner Art und Weise“ eine Verbesserung der Datenqualität erzielt, erhält man im eigenen Data Warehouse auch nur „irgendeine“ Datenqualität.

Der kombinierte Ansatz der vorgestellten Maßnahmen zur Analyse und Bereinigung mit einem strukturierten Vorgehensmodell führt zu einem integrierten Datenqualitätsmanagement. Mit Werkzeugen wie z.B. dem IBM InfoSphere Information Server gibt es Möglichkeiten, um die Qualität der Daten nachhaltig zu verbessern. IBM bietet mit dem Datenqualitäts-Assessment-Workshop zudem die Möglichkeit für einen klar definierten Scope den Status Ihrer Datenqualität zu erfassen und Handlungsoptionen aufzuzeigen.

Gerne beraten wir Sie diesbezüglich, sprechen Sie uns einfach an!



Michael Steinert
(info@ordix.de)



Sascha Slomka
(sascha.slomka@de.ibm.com)

Links

- ▶ [1] IBM-Publikation „The IBM Data Governance unified Process“:
<http://public.dhe.ibm.com/common/ssi/ecm/en/imm14074usen/IM-M14074USEN.PDF>
- ▶ [2] Verhaltensregeln für den Umgang mit personenbezogenen Daten durch die deutsche Versicherungswirtschaft:
http://www.gdv.de/wp-content/uploads/2013/03/GDV_Code-of-Conduct_Datenschutz_2012.pdf

Bildnachweis

- ▶ © shutterstock | attractive man close up portrait on white background | Leonid and Anna Dedukh

Ein erster Überblick

Nachdem wir mit dem „Plan B“ von Java 7 im Juli 2011 auf einige Sprachfunktionen verzichten müssen, kommt mit Java 8 nun wieder richtig Bewegung in die Syntax: Lambda Expressions, Functional Interfaces und Method References sind hier die Platzhirsche. Die lang erwarteten Lambda Expressions bieten neue Möglichkeiten für besseren Java-Code und die damit verbundenen Collections-Funktionen eröffnen uns neue Wege. Diese und viele weitere Neuerungen stellen wir Ihnen in der neuen Reihe zum Thema Java 8 vor.

GA - General Availability

Pünktlich zum Erscheinungstermin unserer ORDIX® news wird nun auch die finale Version von Java 8 (GA) veröffentlicht.

Die Funktionsliste ist mit 55 JEPs recht üppig. Doch glücklicherweise haben nicht alle JEPs direkte Auswirkungen auf die Sprache. Wir konzentrieren uns daher auf die wichtigsten Neuerungen und werden Ihnen diese vorstellen.

Hatte der Vorgänger Java 7 noch den Code-Namen „Dolphin“, so ist mit Java 8 eins auf jeden Fall anders: Es gibt keinen Code-Namen.

Die essentiellen Sprachfunktionen

Die folgenden Erweiterungen von Java 8 sind für die Entwicklung von besonderem Interesse:

- Interface-Erweiterungen: statische Methoden und Default-Methoden, `@FunctionalInterface`
- Lambda Expressions: neue Syntax für schlankeren Code
- Methoden-Referenzen: Nutzung von Methoden ohne Aufruf
- Lambda Expressions und erweiterte Collections-API
- Neue Date- & Time-API
- Erweiterungen im Bereich IO/NIO
- Neue Streaming-API

Darüber hinaus stellt der Wegfall der Permanent Generation der Hotspot VM für den Betrieb eine Veränderung im Hinblick auf das Memory-Sizing und Tuning dar.

Lediglich das Projekt Jigsaw (Modularisierung in Java), das schon für Java 7 geplant war, wird auf Java 9 verschoben.

Interface-Erweiterungen

Die Interface-Erweiterungen sind zum Teil aus Kompatibilitätsgründen bei der Einführung der Lambda Expressions entstanden. Bisher konnten in Interfaces Methoden nur deklariert, aber nicht implementiert werden.

Jetzt können Interfaces sogenannte Default-Methoden und statische Methoden beinhalten. Die Default-Methoden können von der implementierenden Klasse überschrieben werden (müssen sie aber nicht). Die Abbildung 1 zeigt eine solche Default-Implementierung. In der statischen Methode `getLogger()` ist ebenfalls zu sehen, dass das Interface `Logger` anonym implementiert wird und die Methode `log(...)` nicht überschrieben werden muss.

Warum die Lambda Expressions der Auslöser für die Default-Methoden waren, schauen wir uns später an.

Lambda Expressions

Die Lambda Expressions sind ein wesentlicher Bestandteil von Java 8. Sie ermöglichen die einfache Verwendung eines Ein-Methoden-Interface ohne konkrete Implementierung (z.B. mittels anonymer Implementierung oder innerer Klasse).

Schauen wir uns beispielsweise das bekannte Interface `java.lang.Runnable` an. Es ermöglicht die Ausführung der Methode `run()` in einem separaten Thread. Eine einfache und bekannte Implementierung könnte aussehen wie in Abbildung 2 dargestellt.

Mit einer Lambda Expression kann auf die anonyme Implementierung verzichtet werden und man kann sich auf die eigentliche Logik konzentrieren - nämlich auf das, was im `run()` geschehen soll. Die Abbildung 3 zeigt, wie die gesamte anonyme Implementierung eines `Runnable` entfallen kann. Da es sich hierbei um ein Ein-Methoden-Interface handelt, weiß der Compiler, welche Methode überschrie-

ben werden soll und kann die Übersetzung in den Bytecode durchführen.

Ein solches Ein-Methoden-Interface wird als Functional Interface bezeichnet. Um ein Interface als solches zu kennzeichnen wurde die Annotation `@FunctionalInterface` eingeführt. Versucht man dem Interface nun eine weitere Methode hinzuzufügen, gibt der Compiler eine Fehlermeldung aus.

Methoden-Referenzen

Wir haben aufgezeigt, wie man mit Lambda Expressions anonyme Methoden implementieren kann. Sehr häufig ist der Funktionsumfang in diesen anonymen Methoden auf den Aufruf einer weiteren Methode beschränkt. In unserem Beispiel war das die Methode `sayHello()` der Klasse `Runner`. Statt die Methode direkt aufzurufen, kann man sie aber auch einfach als Methoden-Referenz angeben:

```
new Thread(Runner::sayHello)
```

Da hier die Methoden-Signatur von `Runnable#run()` und `Runner#sayHello()` gleich ist, kann der Compiler dies direkt auflösen (siehe Abbildung 4). Der Thread ruft zunächst `run()` und darin anschließend `sayHello()` auf. Zur Laufzeit ist dies dann dasselbe, als hätte man es manuell implementiert.

Lambda Expressions und Collections

Nachdem Sie nun einen ersten Einblick in die Lambda Expressions bekommen haben, schauen wir uns an, wie diese im Zusammenhang mit Collections effizient genutzt werden können.

Die Abbildung 5 zeigt dazu ein kleines Beispiel. Zunächst wird eine einfache Liste mit Namen erstellt. Anschließend soll diese Liste gefiltert werden. Es sollen nur Namen berücksichtigt werden, die mit einem „R“ beginnen. Anschließend werden die gefilterten Elemente auf der Konsole ausgegeben.

Maßgeblich für diese Funktionalität ist das Interface `java.util.stream.Stream`. Dieses stellt u.a. die hier verwendeten Methoden `filter(...)` und `forEach(...)` zur Verfügung. Dadurch wird über eine Schleife aus der üblicherweise verwendeten externen eine interne Iteration. Dies hat den Vorteil, dass man sich als Entwickler nicht um die Iteration kümmern muss und die Verantwortung abgibt. Dadurch sind auch sehr einfach Parallelisierungen (wie z.B. beim Fork-/Join-Framework) möglich.

Spätestens jetzt wird klar, warum Default-Methoden in Interfaces eingeführt worden sind. Die `stream()`-Methode ist beispielsweise neu in das `Collection`-Interface aufgenommen worden. Wäre das nicht über den Default-Mechanismus geschehen, hätten alle Implementierungen

```
public interface Logger {
    default void log(String value) {
        System.out.println(value);
    };
    static Logger getLogger() {
        return new Logger() {
            // nichts weiter
        };
    }
}
```

Abb. 1: Interface mit einer Default-Implementierung und einer statischen Methode.

```
public class Runner {
    private static void noLambda() {
        Thread t = new Thread(new Runnable() {
            @Override
            public void run() {
                sayHello();
            }
        });
        t.start();
    }
    public static void sayHello() {
        System.out.println("Hello Java 8!");
    }
    ...
}
```

Abb. 2: Anonyme Implementierung von `Runnable`.

```
public class Runner {
    private static void withLambda() {
        Thread t = new Thread(
            () -> sayHello()
        );
        t.start();
    }
    public static void sayHello() {
        System.out.println("Hello Java 8!");
    }
    ...
}
```

Abb. 3: Implementierung von `Runnable` mittels einer Lambda Expression.

```
public class Runner {
    private static void withMethodReference() {
        Thread t = new Thread(Runner::sayHello);
        t.start();
    }
    public static void sayHello() {
        System.out.println("Hello Java 8!");
    }
    ...
}
```

Abb. 4: Beispiel für eine Methoden-Referenz.

von `Collection` diese Methode implementieren müssen. Das gilt selbstverständlich auch für Klassen, die nicht zur Java-API gehören. Mit diesem Mechanismus wird somit die Abwärtskompatibilität gewährleistet.

Neue Date- & Time-API

Wer schon einmal intensive Zeit- und Datumsberechnungen durchführen musste, der weiß, dass das Thema zunächst einfacher klingt als es ist. Schaltjahre, Zeitzonen sowie unterschiedliche Kalender und Darstellungen können einem das Leben ganz schön schwer machen. Um dafür eine bessere API zu schaffen, wurde das Paket `java.time` in die API von Java 8 aufgenommen.

Inspiziert durch das Projekt „Joda-Time“ sollen Datumsoperationen nun einfacher, intuitiver, threadsafe und auch performanter sein. Wenn Sie also schon Joda-Time verwendet haben, werden Sie sich recht schnell an die neue Date- & Time-API gewöhnen. In einem späteren Artikel dieser Reihe werden wir Ihnen weitere Details dazu vorstellen.

```
List<String> names = Arrays.asList("Rohde", "Grimm",
    "Vogt",
    "Zeller", "Schäfers", "Löhr", "Röber");

names.stream()
    .filter((name) -> name.startsWith("R"))
    .forEach(System.out::println);
```

Abb. 5: Lambda Expressions in Verbindung mit den neuen Collection-Funktionen.

Glossar

JDK Enhancement Proposal (JEP)

Der JDK Enhancement Proposal ist ein Mechanismus von Oracle zur Sammlung von Verbesserungs- und Erweiterungsvorschlägen für das Java Development Kit und Open JDK.

Joda Time

Joda-Time ist eine Erweiterung für Java, um Datums- und Zeitoperationen durchzuführen.

Links

- ▶ [1] OpenJDK Webseite „JDK 8“:
<http://openjdk.java.net/projects/jdk8/>
- ▶ [2] Joda-Time Webseite:
<http://www.joda.org/joda-time/>
- ▶ [3] Mark Reinhold's Blog „Re-thinking JDK 7“:
<http://mreinhold.org/blog/rethinking-jdk7>
- ▶ [4] OpenJDK Webseite „Project Jigsaw“:
<http://openjdk.java.net/projects/jigsaw/>

Erweiterungen im Bereich IO/NIO

Nach der Einführung des Fork-/Join-Framework mit Java 7 und der neuen Filesystem-API, wird die IO/NIO-API nun um ein paar Methoden erweitert:

- `BufferedReader.lines()`
- `Files.list(Path)`
- `Files.walk(Path, int, FileVisitOption...)`
- `Files.walk(Path, FileVisitOption...)`
- `Files.find(Path, int, BiPredicate, FileVisitOption...)`
- `Files.lines(Path, Charset)`
- `DirectoryStream.stream()`

Damit hat man nun einen direkten Zugriff auf einen `java.util.stream.Stream` und kann direkt die beschriebenen Funktionen der Lambda Expressions und Collections nutzen.

Ebenfalls neu ist die `UncheckedIOException`. Diese wird allerdings nur an wenigen Stellen genutzt. Im Wesentlichen immer dann, wenn es sich beim Rückgabewert um einen Stream handelt.

Neue Streaming-API

Die Verwendung eines Stream haben Sie bereits im Zusammenhang mit den Lambdas und Collections gesehen. In einem weiteren Artikel dieser Reihe zeigen wir Ihnen, wie man diese effizient verwendet und für eine parallele Datenverarbeitung nutzen kann.

Fazit

Java 8 hat einiges zu bieten und viele der neuen Konstrukte, insbesondere die Syntax rund um die Lambdas, müssen erst durch die intensive Verwendung zur Routine werden. Aber wie auch schon bei der Einführung von Generics mit Java 5, werden wir viele der neuen Funktionen bald nicht mehr missen wollen.



Thomas Rohde
(info@ordix.de)

Wechsel des JDK beim WebSphere Application Server 8

Neuer Antrieb für WebSphere

Jeder Administrator kennt das Problem: In regelmäßigen Abständen erscheinen kritische Security Updates, die auf diversen Maschinen installiert werden müssen. Aber auch Entwickler möchten ihre erstellten Programme unter Umständen mit der neusten Version der verwendeten Programmiersprache testen. Dieser Artikel beschreibt die Überprüfung des aktuell eingesetzten Java Development Kit (JDK) und den Austausch bzw. Wechsel durch eine neuere Version beim WebSphere Application Server 8.

Ausgangslage

Ist bereits ein WebSphere Application Server (WAS) auf dem System installiert, stellt sich die Frage nach der aktuell verwendeten Java-Version. Während bei der Version 8.0 des WAS ausschließlich die Java-Version 6 unterstützt wurde, so besteht ab dem WAS 8.5, der standardmäßig mit der Version 6 des SDK daher kommt, die Möglichkeit des späteren Austausches durch SDK 7. Dabei verwendet IBM eine eigene Version des Original-JDK von Oracle.

Bestandsaufnahme

Aber welche Versionen des SDKs sind auf dem verwendeten Server installiert und welche werden verwendet? Diese Information erhält man über mehrere Wege:

Der einfachste Weg ist der über die Administrationskonsole. In der Baumstruktur auf der linken Seite wählt man über folgende Vorgehensweise einen Server aus:

Servers ► Server Types ► WebSphere application servers und gelangt dort über

Configuration ► Server Infrastructure ► Java SDKs

auf die Übersichtsseite aller für den Server verfügbaren Java-Versionen (siehe Abbildung 1). Sind mehrere Versionen installiert, so ist eine davon als Default definiert und es besteht dort die Möglichkeit auf eine andere Version zu wechseln. Der Server und alle darauf installierten Anwendungen werden nach einem Neustart mit dem neu gewählten JDK betrieben. Die Übersicht zeigt nicht nur die Versionsnummern, sondern auch die verwendete Architektur (32 oder 64 Bit).

Wer es lieber auf der Kommandozeilenebene mag, der ist mit dem Kommando `managesdk` gut bedient. Dieses befindet sich im Verzeichnis `bin` des Application Server. Eine Übersicht über die Mächtigkeit dieses kleinen aber feinen Werkzeugs erhält man mittels `managesdk -help`.

Ein sinnvoller Aufruf wäre zum Beispiel `managesdk.bat -listEnabledProfile -profileName AppSrv01`.



Abb. 1: Übersicht der verfügbaren SDKs.

Dieser zeigt alle im Profil **AppSrv01** verwendeten JDKs an. Eine Ausgabe ist in Abbildung 2 dargestellt.

Java 6 ist installiert im Verzeichnis **java**, Java 7 ist unter **java_1.7** zu finden - jeweils im **Home**-Verzeichnis des WAS.

Welche Build/Patch-Version genau installiert ist, lässt sich über **java -version** im jeweiligen Verzeichnis **java/bin** ermitteln (siehe Abbildung 3).

```
C:\Program Files\IBM\WebSphere\AppServer\bin>managesdk.bat
-listEnabledProfile -profileName AppSrv01
CWSDK1004I: Profile AppSrv01 :
CWSDK1006I: PROFILE_COMMAND_SDK = 1.6_32
CWSDK1008I: Node win7vmNode02 SDK name: 1.6_32
CWSDK1009I: Server server1 SDK name: 1.6_32
CWSDK1001I: Successfully performed the requested managesdk
task.
```

Abb. 2: Alle im Profil AppSrv01 verwendeten JDKs.

```
C:\Program Files\IBM\WebSphere\AppServer\java\bin>java.exe
-version
java version "1.6.0"
Java(TM) SE Runtime Environment
(build pwi3260_26sr2ifix-20120419_02 (SR2+IV19661))
IBM J9 VM
(build 2.6, JRE 1.6.0 Windows 7 x86-32
20120322_106210 (JIT enabled, AOT enabled)
J9VM - R26_Java626_SR2_20120322_1722_B106210
JIT - r11_20120322_22976
GC - R26_Java626_SR2_20120322_1722_B106210
J9CL - 20120322_106210)
JCL - 20120316_01
```

Abb. 3: Detaillierte Ausgabe zur installierten Java-Version.

```
/IBM/WebSphere/AppServer/bin # ./managesdk.sh -listAvailable -verbose
CWSDK1003I: Verfügbare SDKs:
CWSDK1005I: SDK-Name: 1.7_32
- com.ibm.websphere.sdk.version.1.7_32=1.7
- com.ibm.websphere.sdk.bits.1.7_32=32
- com.ibm.websphere.sdk.location.1.7_32=${WAS_INSTALL_ROOT}/java_1.7_32
- com.ibm.websphere.sdk.platform.1.7_32=linux
- com.ibm.websphere.sdk.architecture.1.7_32=x86_32
- com.ibm.websphere.sdk.nativeLibPath.1.7_32=${WAS_INSTALL_ROOT}/lib/native/linux/x86_32/
CWSDK1005I: SDK-Name: 1.6_32
- com.ibm.websphere.sdk.version.1.6_32=1.6
- com.ibm.websphere.sdk.bits.1.6_32=32
- com.ibm.websphere.sdk.location.1.6_32=${WAS_INSTALL_ROOT}/java
- com.ibm.websphere.sdk.platform.1.6_32=linux
- com.ibm.websphere.sdk.architecture.1.6_32=x86_32
- com.ibm.websphere.sdk.nativeLibPath.1.6_32=${WAS_INSTALL_ROOT}/lib/native/linux/x86_32/
CWSDK1001I: Die angeforderte managesdk-Task wurde erfolgreich ausgeführt.
```

Abb. 4: Ausgabe des Befehls **managesdk**.

Update einer bestehenden Version

Wie kommt man nun an eine neuere Java-Version? Bis zur Version 7 des WAS standen Updates für das JDK noch einzeln bei IBM zum Download bereit.

Ab Version 8 des Application Server werden die Updates für den Server und Updates für das SDK zusammen ausgeliefert. Wird der Server mit einem Fixpack aktualisiert, so wird gleichzeitig das JDK gepatched.

Kritische Security Fixes für Java liefert IBM aber auch einzeln in sogenannten Interim Fixes. Es lohnt sich also von Zeit zu Zeit einen Blick auf die Update-Seiten von IBM zu werfen.

Installation des JDK 7

Möchte man als Entwickler nun in den Genuss des JDK 7 kommen, so muss diese Version nachinstalliert werden. Dies erfolgt in allen Fällen über den IBM Installation Manager (IM). Dabei gibt es drei Möglichkeiten:

- vorhandenes SDK auf den beim Kauf mitgelieferten Medien nutzen
- Online Repository im IM einbinden
- Installationspaket von der IBM-Webseite beziehen

Die Installation läuft dann, wie bei allen anderen Installationen, über den Installation Manager ab.

Anschließend kann über den Befehl **managesdk** überprüft werden, ob das neu hinzugekommene JDK für den Application Server zur Verfügung steht. Die Ausgabe auf der Kommandozeile sollte wie in Abbildung 4 dargestellt aussehen. Dort ist auch zu sehen, wo sich das Installationsverzeichnis von Java befindet.

Der Wechsel

Für den Wechsel des SDK für ein spezielles Profil wird wieder der Befehl **managesdk** verwendet. Beispielsweise wird für den Application Server mit dem Namen **AppSrv01** mit folgendem Befehl auf Java 7 gewechselt:

```
managesdk -enableProfile -sdkname 1.7_32
-profileName AppSrv01
```

Das ist auch schon alles. Der korrekte Wechsel lässt sich anschließend wieder über **managesdk** oder mittels der Administrationskonsole überprüfen.

Wird die Network Deployment Version (ND) des WAS genutzt, so muss der Deployment Manager aktiv sein und das Kommando von dort abgesetzt werden. Dadurch wird die Master-Konfiguration des ND aktualisiert und anschließend per Synchronisation über die Node Agents auf die einzelnen Knoten verteilt.

Noch mehr Stoff

Mit dem Befehl `managesdk` lassen sich noch viele weitere Einstellungen bzw. Änderungen durchführen. Diese alle hier aufzuführen würde aber den Umfang des Artikels sprengen. Daher werden hier nur zwei nützliche Aktionen kurz vorgestellt.

Neue Profile, die mit dem Profilmanager angelegt werden, laufen unter dem default-mäßig eingestellten SDK. Wie eingangs erwähnt ist das die Version 6. Mit folgendem Befehl werden alle neu angelegten Profile zukünftig mit Java 7 gestartet:

```
managesdk.sh -setNewProfileDefault -sdkname 1.7_32
```

Möchte man alle angelegten Profile mit einer bestimmten SDK-Version betreiben und nicht jedes Profil einzeln umstellen, dann erledigt dies der folgende Befehl auf einen Schlag:

```
managesdk.sh -enableProfileAll -sdkname 1.7_32 -enableServers
```

Anschließend verwenden alle Profile Java in der Version 7. Schneller geht es wirklich nicht.

Fazit

Wie wir gesehen haben, ist es denkbar einfach, Informationen über die aktuell genutzte Java-Version eines Application Server zu erhalten. Auch die Installation und der Wechsel auf ein Update lassen sich schnell durchführen.

Glossar

JDK

Das Java Development Kit ist eine Laufzeitumgebung und eine Sammlung von Entwicklungswerkzeugen. Anders ausgedrückt ist es ein SDK für die Programmiersprache Java.

SDK

Das Software Development Kit ist eine allgemeine Bezeichnung für eine Ansammlung von Werkzeugen für die Erstellung von Software.

WAS

Der WebSphere Application Server ist eine Laufzeitumgebung und ein Verwaltungswerkzeug für Java-Enterprise-Anwendungen von IBM.

So sind die gehosteten Anwendungen sicherer, weniger fehleranfällig und laufen stabiler. Entwickler haben hingegen die Möglichkeit Anwendungen mit dem aktuellen JDK 7 zu entwickeln und zu testen.



Andre Dirr
(info@ordix.de)

Seminarempfehlung: WebSphere Application Server Installation und Administration

► Informationen/Online-Anmeldung: <http://training.ordix.de>

In diesem Seminar werden Sie mit der Installation und Administration des WebSphere Application Server vertraut gemacht. Außerdem werden die Themen Hochverfügbarkeit, Lastverteilung und automatische Softwareverteilung intensiv in Theorie und Praxis behandelt.

Seminarinhalte

- Überblick über die Java-EE-Architektur
- Architektur des WebSphere Application Server
- Installation des WebSphere Application Server
- Grundlagen der Administration mit der Administrationskonsole und CLI-Werkzeugen
- Installation einer Anwendung
- Verwaltung und Installation von Data Sources
- Fehlersuche
- Absicherung der Administrationskonsole
- Erstellung eines Cluster zur Lastverteilung und Hochverfügbarkeit
- Konfiguration von Java Messaging Service (JMS)
- Konfiguration und Deployment von Message Driven Beans
- Vertiefung der Theorie durch praktische Übungen und Beispiele

Termine

10.03. - 12.03.2014 in Wiesbaden
10.06. - 12.06.2014 in Wiesbaden
01.09. - 03.09.2014 in Wiesbaden
24.11. - 26.11.2014 in Wiesbaden

Seminar-ID: INT-08

Dauer: 3 Tage

Preis pro Teilnehmer:
1.390,00 € (zzgl. MwSt.)

Frühbucherpreis:
1.251,00 € (zzgl. MwSt.)



Buchen Sie gleich hier!

QS - Was? Wann? Wer?

Wer die Begriffe „Qualitätssicherung“ und „Softwareprojekte“ hört, denkt unmittelbar an das Testen von ausführbaren Programmen. Das ist zwar grundsätzlich nicht falsch, aber es ist eine sehr eingeschränkte Betrachtungsweise. Ein fehlerfreies Programm ist noch lange nicht qualitativ hochwertig. Demnach umfasst die Qualitätssicherung auch mehr als nur das reine Testen von Software.

Statisches Testen

Am Anfang eines Softwareprojektes steht meist ein Dokument in Form einer Anforderungsspezifikation (Lastenheft). Darin werden jene Wünsche aufgeführt, die der Auftraggeber an das zukünftige Softwareprodukt hat. Wird ein Lastenheft als Instrument für eine Ausschreibung verwendet, so werden die potentiellen Dienstleister mit einem Pflichtenheft antworten.

Softwarearchitekten, Projektleiter und Entwickler sind jene Rollen, die in der Regel mit der Analyse des Lastenhefts und der Erstellung eines Pflichtenhefts assoziiert werden. Aber schon hier sollte bereits die Qualitätssicherung in Form des statischen Testens einsetzen. Denn Qualität kann nicht erst nach der Fertigstellung der Software „hineingetestet“ werden.

Welche Fragestellungen sollten berücksichtigt werden?

Welche Technik setzen wir ein? Was für Ressourcen brauchen wir? Welche Werkzeuge sollen wir nutzen? Welche Entwicklungsmethode(n) wenden wir an? Das sind typische Fragen, die sich Softwarearchitekten und -entwickler stellen und beantworten müssen. Der Qualitätsmanager untersucht das vorliegende Dokument hingegen nach anderen Schwerpunkten. Er stellt sich Fragen wie: Gibt es Widersprüche? Sind die Anforderungen eindeutig spezifiziert? Bleiben Fragen unbeantwortet? Haben alle Beteiligten (Kunde, Entwicklung und Test) dasselbe Verständnis – wird dieselbe „Sprache“ gesprochen?

Insbesondere die Interpretierbarkeit von Aussagen und damit verbundene Fehlannahmen können im späteren Verlauf des Projektes sehr teuer werden. Aus rein wirtschaftlichen Gründen ist es daher sehr zu empfehlen, noch weit vor dem ersten produzierten Quellcode mit der Qualitätssicherung zu beginnen. Was hier exemplarisch am Lastenheft beschrieben wird, setzt sich im späteren Projektverlauf konsequent fort. Mit jedem Change Request sollte erneut statisch getestet werden.

Review - das Instrument für statische Tests

Im IEEE1028-Standard wird zwischen zwei Review-Gruppen unterschieden. Reviews auf die Produkte und Teilprodukte (Gruppe 1) sowie Reviews auf den Projektablauf oder Entwicklungsprozess (Gruppe 2). Wir betrachten hier nur die erste Gruppe mit folgenden Review-Arten:

- Technischer Review
- Walkthrough
- Inspection
- Informeller Review

Ohne im Einzelnen auf die unterschiedlichen Review-Arten einzugehen, können diese prinzipiell auf alle Arbeitsergebnisse im Softwareentwicklungsprozess (z.B. Anforderungsspezifikationen, Designspezifikationen, Testspezifikationen oder Softwaredokumentation) angewendet werden und bieten damit die Möglichkeit, sehr früh in dem Softwareentwicklungsprozess qualitätssichernde Maßnahmen durchzuführen.

Wer ist beteiligt und wie geht man vor?

Teilnehmer eines solchen Review sind mindestens einer der Verfasser (oft Kundenvertreter), ein Gutachter (z.B. Projektleiter), ein Protokollant (z.B. Testmanager) und ein Moderator (z.B. Softwarearchitekt).

Sinnvollerweise kommt dabei eine standardisierte Checkliste oder Fragenliste zum Einsatz. Die Punkte dieser Liste hat der Moderator zur Vorbereitung von den Teilnehmern im Vorfeld eingefordert. Mit Hilfe eines vollständigen Review können bereits vorab 60 - 90 % der potentiellen Fehler gefunden werden, die sich unmittelbar auf den Erfolg des Projektes auswirken können.

Dynamisches Testen

Beim dynamischen Testen wird zwischen White-Box- und Black-Box-Tests unterschieden. Zunächst soll an dieser Stelle der Black-Box-Test näher betrachtet werden. In die-

sem wird das laufende (Teil-)Programm/System getestet. Black-Box-Tests werden zu verschiedenen Zeitpunkten und mit unterschiedlichem Fokus bzw. Aufgaben im Projektverlauf ausgeführt. Dieser Zusammenhang wird im klassischen V-Modell in der Abbildung 1 gut illustriert.

Anforderungsdefinition - das Maß aller Dinge

Im Rahmen der Anforderungsdefinition, welche zu Beginn des Projektes formuliert wird (u.a. im Lastenheft), werden die Leistungen und Anforderungen des Kunden an das zukünftige Softwaresystem aufgeführt. Diese Definition steht am Ende des Projektes dem Abnahmetest gegenüber. Der Abnahmetest ist somit ein Test gegen den Vertrag, der zwischen dem Auftraggeber und dem Softwarelieferanten besteht. Hier wird final festgestellt, ob das ausgelieferte Softwareprodukt den Erwartungen des Kunden entspricht. Tests dieser Art werden entweder durch Kundenvertreter oder vom Testteam unter der Regie des Kunden durchgeführt.

Funktionaler Systementwurf

Im funktionalen Systementwurf werden die Anforderungen des Systems mit genauerem Detaillierungsgrad in Funktionen und Dialogabläufen abgebildet. Dem gegenüber steht vor der Abnahme der Systemtest. Sämtliche Teilsysteme des Gesamtproduktes werden jetzt verbunden und als Gesamtheit getestet. Dies ist gewissermaßen die Generalprobe des Softwarelieferanten vor dem Abnahmetest.

Technischer Systementwurf

Im technischen Systementwurf wird die technische Realisierung des Systems vorbereitet. Schnittstellen werden definiert und das Gesamtsystem in sinnvolle Teilsysteme untergliedert. Diese werden idealerweise getrennt entwickelt und getestet. Dem technischen Systementwurf steht der Integrationstest gegenüber. Auf dieser Teststufe werden demnach einzelne Softwaremodule zu autarken Teilsystemen verbunden, welche unabhängig voneinander getestet werden.

Komponententest

Die Heimat der Entwickler sind die sogenannten Komponententests (oft auch als Unit-Tests oder Modul-Tests bezeichnet). Hier werden die kleinsten Softwareeinheiten (Funktionen/Prozeduren/Methoden) getestet und hinsichtlich eines korrekten Verhaltens geprüft. Auf der Entwurfsseite steht diesen Tests die Komponentenspezifikation gegenüber. Darin werden im Detail die einzelnen Softwaremodule definiert (Schnittstellen, Ein- und Ausgabe, etc.). Obwohl der Entwickler Zugriff auf den Code hat, handelt es sich hierbei um einen Black-Box-Test, da lediglich das Verhalten der Software zur Laufzeit geprüft wird.

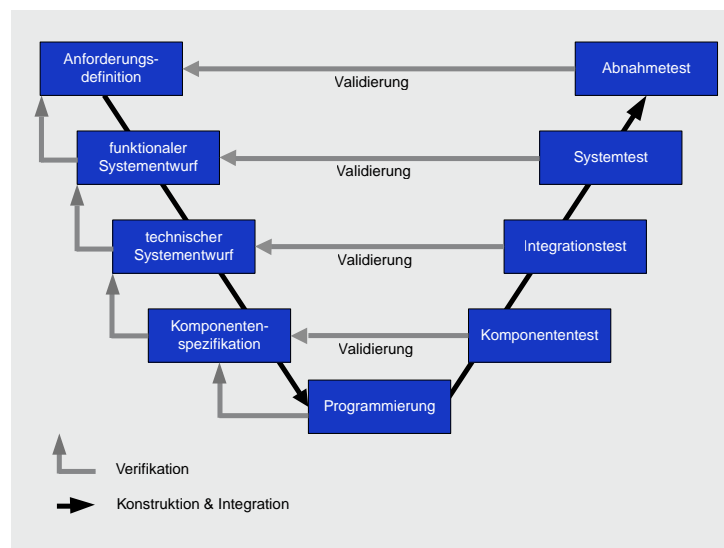


Abb. 1: Allgemeines V-Modell, Quelle [2].

Code-Analyse mit dem White-Box-Test

Dem Black-Box-Test gegenüber steht der sogenannte White-Box-Test. Darin wird der Quellcode als solcher analysiert und nicht etwa ausgeführt. In der sogenannten statischen Code-Analyse werden werkzeunterstützt Metriken erstellt, mit deren Hilfe eine Aussage über die Qualität des Quellcodes gemacht werden kann. Hinzu kommen erfahrungsbasierte Betrachtungen. So kann u.a. geschätzt werden, wieviele Fehler im Verhältnis zu den vorhandenen Zeilen Code zu erwarten sind. Mit Hilfe der werkzeunterstützten Code-Analyse sind u.a. folgende Prüfungen durchführbar:

- Anweisungsüberdeckung
- Zweigüberdeckung
- Test der Bedingungen
- Pfadüberdeckung

Die statische Code-Analyse kann in Entwicklungsumgebungen während des Entwicklungsprozesses eingesetzt werden. Sie sollte aber auf jeden Fall Bestandteil einer Continuous-Integration-Umgebung sein.

Interessanterweise sagt die Literatur nicht viel darüber aus, zu welchem Zeitpunkt und von welchem Personenkreis White-Box-Tests durchgeführt werden sollten. Da es sich jedoch um Tätigkeiten der Code-Analyse handelt, liegt es nahe, diese begleitend zur Programmierung durchzuführen. Idealerweise liegt die Verantwortung bei einem erfahrenen Entwickler, Softwarearchitekten oder auch technikaffinen Tester.

Wer macht was?

Bisher wurden nur die unterschiedlichen Teststufen betrachtet. Lediglich bei den Komponententests und der

Glossar

Anweisungsüberdeckung

Anweisungsüberdeckungstests, auch C0-Test genannt, testen jede Anweisung mindestens ein Mal. Damit stellen sie sicher, dass kein toter Code (Anweisungen die niemals durchlaufen werden) im Programm existiert. Wurde jede Anweisung in einem Programm mindestens einmal ausgeführt, spricht man von völliger Anweisungsüberdeckung.

Black-Box-Tests

Ein Black-Box-Test bezeichnet eine Methode des Softwaretests, bei der die Tests ohne Kenntnisse über die innere Funktionsweise des zu testenden Systems entwickelt werden. Er beschränkt sich auf funktionsorientiertes Testen, d.h. für die Ermittlung der Testfälle werden nur die Anforderungen, aber nicht die Implementierung des Testobjekts herangezogen.

Change Requests (CR)

Change Requests sind Anforderungen, die sich nach Abschluss der Anforderungsdefinition für eine Software, also während der Implementierungsphase, ergeben.

Continous-Integration-Umgebung

Umgebung in der fortlaufend Komponenten zu einer Anwendung hinzugefügt werden. Das Ziel der kontinuierlichen Integration ist die Steigerung der Softwarequalität.

Informeller Review

Ein informeller Review entspricht inhaltlich dem technischen Review, es soll ihm gegenüber aber Zeit gespart werden und daher wird es als nicht formaler Prozess durchgeführt. Der informelle Review ist nicht im IEEE-Standard für Software-Reviews enthalten.

Inspection

Eine Inspection ist die formalste Review-Technik mit einem dokumentierten Vorgehen nach IEEE 610 und IEEE 1028. Zweck: Sichtüberprüfung von Dokumenten, um Mängel zu finden (z.B. Nichteinhaltung von Entwicklungsstandards, Nicht-Konformität gegenüber Spezifikationen, usw.).

Pfadüberdeckung

Beim Pfadüberdeckungstest (auch C2-Test bzw. englisch path coverage) werden im Kontrollflussgraphen die möglichen Pfade vom Startknoten bis zum Endknoten betrachtet.

Technischer Review

Ein technischer Review ist eine fachliche Prüfung eines wesentlichen Dokumentes (z.B. Architekturentwurf) auf Übereinstimmung mit Spezifikation. Zweck: Diskussion, Entscheidungen treffen, Alternativen bewerten, Fehler finden, technische Probleme lösen.

Test der Bedingungen

Das Problem der bisherigen Überdeckungstests (C1-Test, C2-Test) ist, dass zusammengesetzte, hierarchische Bedingungen nicht ausreichend getestet werden. Dies wird mit Hilfe von Bedingungsüberdeckungstest geprüft.

Walkthrough

Ein Walkthrough ist eine Diskussion von Szenarien, Probelaufen und Alternativen im Kreis gleichgestellter Mitarbeiter mit möglichst niedrig gehaltenem Aufwand. Zweck: Lernen, Verständnis erzielen und Fehler finden

Code-Analyse wurde erwähnt, wer diese durchführt. Doch wie verhält es sich mit den anderen Teststufen? Bei der Qualitätssicherung werden grundsätzlich fünf verschiedene Testrollen unterschieden:

- **Testmanager**
Der Testmanager plant und steuert den Test, hat Verantwortung für sein Testteam und berichtet den Teststatus an das Projektmanagement.
- **Testdesigner**
Der Testdesigner entwirft die Testfälle. Ebenso ist er ein Experte für Testmethoden und die Testspezifikation im Softwaretest.
- **Testautomatisierer**
Der Testautomatisierer ist ein Experte für die Automatisierung von Testfällen. Das heißt, er entwickelt Testfälle, die automatisch und werkzeunterstützt ablaufen können.
- **Testadministrator**
Ein Experte für die Installation und den Betrieb der Testumgebung ist der Testadministrator. Seine Kenntnisse sind vergleichbar mit denen eines Systemadministrators.
- **Tester**
Der Tester ist für die Testdurchführung zuständig ist, erstellt Fehlerberichte und meldet Probleme.

In der Praxis kommt es regelmäßig vor, dass einige der oben genannten Rollen in Personalunion wahrgenommen werden. Dies ist auch legitim, da eine scharfe Trennung meist gar nicht möglich oder sinnvoll ist. Mit Ausnahme des Komponententests (hier testen die Entwickler) sind die oben genannten Rollen in allen Teststufen anzutreffen.

Warum es nicht sinnvoll ist, dass Entwickler im System-, Integrations- oder Abnahmetest mitwirken, erläutern wir übrigens im zweiten Teil dieser Reihe.

Begrifflichkeiten

Wie bereits beschrieben, werden White-Box-Tests wie auch Black-Box-Tests dem dynamischen Testen zugeordnet. So ist es zumindest im ISTQB-Standard definiert, nach welchem auch hierzulande zertifiziert wird. Schlägt man allerdings unter dem Stichwort „Statisches Software-Testverfahren“ in Wikipedia nach, so wird man feststellen, dass dort die Tätigkeit der Code-Analyse, da sie nicht zur Laufzeit durchgeführt wird, als statisches Testen angesehen wird. An dieser Stelle existieren somit verschiedene Betrachtungsweisen.

Das in Abbildung 1 abgebildete allgemeine V-Modell ist ebenso in verschiedenen Ausprägungen zu finden. Im Rahmen dieses Artikels wurde der von ISTQB gelehrte Standard zugrundegelegt.

Nicht selten kann auch das Verständnis der beschriebenen Begrifflichkeiten im Kundenumfeld unterschiedlich sein. So wird beispielsweise der Integrationstest gerne als Systemtest verstanden und der Systemtest als solcher gar nicht mehr erwähnt. Um Missverständnissen vorzubeugen, sollte man sich daher zuvor genauestens mit der Testkultur des Auftraggebers vertraut machen.

Unter anderem in Telekommunikationsunternehmen findet man häufig den Begriff des End-to-End-Tests. Im Prinzip ist damit ein produktionsnaher Systemtest gemeint, bei dem von einem Ende (z.B. Anwender) bis zum anderen Ende (z.B. Datenbank) geprüft wird, ob sich das System im Produktionsumfeld korrekt verhalten würde.

End-to-End-Tests werden auch gerne als Ersatz für Abnahmetests genutzt. Erwartungsgemäß sollten dabei kaum noch Fehler auftreten, da diese bereits in den vorangestellten Teststufen entdeckt worden sein sollten. Doch die Praxis beweist auch hier regelmäßig etwas anderes.

Eine ebenfalls gängige Tätigkeit in der Qualitätssicherung ist der sogenannte Regressionstest. Damit ist nach einer Softwaremodifikation das wiederholte Testen von Bestandsfunktionalitäten gemeint.

Im Regressionstest wird somit sichergestellt, dass Softwareänderungen oder -erweiterungen keine Bestandsfunktionalität beeinträchtigen. Regressionstests werden mit jedem neuen Software-Release (Wartungsphase) durchgeführt und haben die Eigenschaft quantitativ zuzunehmen. Hier muss also mit jedem neuen Release eine Priorisierung bzw. Gewichtung stattfinden.

Fazit

In diesem ersten Artikel der neuen Reihe wurde verstärkt auf die Testobjekte „Dokumente“ und „Software“ geschaut. Doch auch die permanente Bewertung des Entwicklungs- und Qualitätssicherungsprozesses selbst ist eine qualitätsfördernde Maßnahme.

Da jede Kundensituation individuell ist, muss Bewährtes aus Projekt A nicht auch das Richtige für Projekt B sein. Hier gilt es aufmerksam zu hinterfragen und anzupassen. Qualität erzielt man nicht allein durch das Ausführen von Testfällen. Einen echten Sinn macht die Qualitätssicherung nur dann, wenn sie als ein ganzheitlicher Prozess betrachtet und dieser an die sich ständig ändernden Bedingungen anpasst wird.

Um die gebräuchlichen Begrifflichkeiten im Testumfeld vorzustellen, wurde hier das allgemeine V-Modell als Vorgangsmethode gewählt. Heutzutage wird aber zunehmend agil entwickelt (z.B. nach Scrum). In agilen Softwareprojekten muss man sich jedoch anderes in Sachen Qualitätssicherung aufstellen als in klassischen Projekten. Auf diesen Aspekt werden wir in folgenden Artikeln näher eingehen.

Glossar (Fortsetzung)

White-Box-Tests

Der Begriff White-Box-Test (auch Glass-Box-Test) bezeichnet eine Methode des Softwaretests, bei der die Tests mit Kenntnissen über die innere Funktionsweise des zu testenden Systems entwickelt werden. Im Gegensatz zum Black-Box-Test ist für diesen Test also ein Blick in den Quellcode gestattet, d.h. es wird der Code geprüft.

Zweigüberdeckung

Der Zweigüberdeckungstest (C1-Test) umfasst den Anweisungsüberdeckungstest vollständig. Für den C1-Test müssen strengere Kriterien erfüllt werden als beim Anweisungsüberdeckungstest. Mit Hilfe des Zweigüberdeckungstests lassen sich nicht ausführbare Programmzweige aufspüren. Anhand dessen kann man dann Softwareteile, die oft durchlaufen werden, gezielt optimieren.

Links

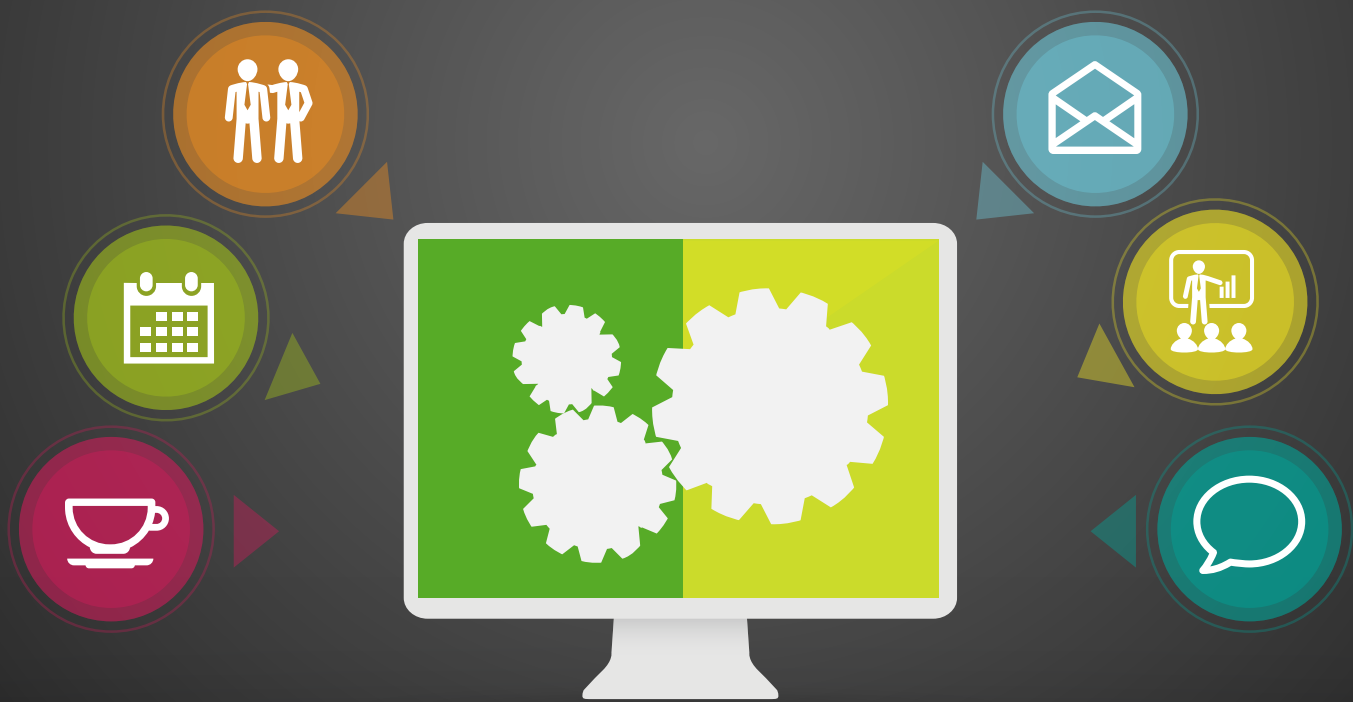
- ▶ [1] ISTQB International Software Testing Qualification Board: <http://www.istqb.org/>
- ▶ [2] Wikipedia-Definition „Statisches Software-Testverfahren“: http://de.wikipedia.org/wiki/Statisches_Software-Testverfahren
- ▶ [3] Wikipedia-Definition „V-Modell“: <http://de.wikipedia.org/wiki/V-Modell>
- ▶ [4] Internetseite der Object Systems GmbH: <http://www.objectsystems.de/>

Quellen

- ▶ [1] Spillner, Linz (2004): Basiswissen Softwaretest, 2. überarbeitete Auflage, dpunkt.verlag, ISBN 3-89864-358-1
- ▶ [2] Abbildung 1 in Anlehnung an die Abbildung V-Modell aus dem Buch Spillner, Linz (2004): Basiswissen Softwaretest, 2. überarbeitete Auflage, dpunkt.verlag, ISBN 3-89864-358-1



Marcus Meisenberg
(info@objectsystems.de)



Kommunikation im Projekt

Mit Plan und Persönlichkeit

Kommunizieren, *communicare* (lat.), *to communicate* (engl.), sich mitteilen, sprechen, reden. Wir alle machen es tagtäglich - fast in jeder Minute. Stets treten wir in Interaktion mit anderen Menschen. Dabei kommunizieren wir mit Worten, Blicken und manchmal auch mit Händen und Füßen. In Projekten übernimmt die Kommunikation eine Schlüsselrolle. Zum einen werden relevante Informationen über den Fortschritt des Projektes ausgetauscht, zum anderen kann das Projekt mit Informationen gezielt beeinflusst werden. Wenn Projekte scheitern, dann sind Kommunikationsprobleme fast immer eine der Hauptursachen. Aus diesem Grund zeigen wir in diesem Artikel, wie der Projektmanager die Kommunikation im Projekt schon vorab gut planen kann und welche Kommunikationswege sich anbieten.

Die Auswahl ist groß

Das Ziel der Kommunikationsplanung im Projekt ist es, den Kommunikationsbedarf der beteiligten Personen zu erfassen, zu kategorisieren und in einer geeigneten Form zum richtigen Zeitpunkt zufrieden zu stellen. Die beteiligten und/oder betroffenen Personen im Projekt werden im Projektmanagement als Stakeholder bezeichnet. Sie haben unterschiedliche Motive, Hintergründe und Interessen. Das Hauptziel des Stakeholder Managements ist es, die Ziele der Stakeholder weitestgehend mit den Projektzielen in Einklang zu bringen. Dabei werden die Interessen und Bedürfnisse der Interessensgruppen zusammengefasst und bei der Realisierung des Projektes weitestgehend berücksichtigt.

Um ein erfolgreiches Stakeholder Management zu betreiben ist es unerlässlich, dass die Projektleitung schon zu Beginn des Projektes einen vollständigen Projektkommunikationsplan erstellt. In diesem sollten folgende Fragen beantwortet werden:

- Welche Stakeholder sind vorhanden und welchen Einfluss haben sie auf das Projekt?
- Welche Motivation und Erwartungen haben diese Stakeholder für/an das Projekt?
- Welche Informationen sind für die Erfüllung dieser Erwartungen notwendig?

Wer	Erwartung	Was	Wie		
Vorstand	Unternehmensziele erreichen	Projektantrag	Präsentationen	E-Mail	Meetings
Lenkungsausschuss	Innovationen vorantreiben	Planungen	Telefonkonferenzen	Protokolle	Meetings
Leiter der Sparte	Umsatz generieren	Statusberichte	Persönliche Gespräche	„Feierabendbier“	Events
Geldgeber/Sponsoren	Profilierung der Abteilung	Entscheidungen	Flyer	Protokolle	Telefon
Projektleiter anderer Projekte	Synergieeffekte nutzen	Ergebnisse	„Lunch Dates“	Instant Messaging	Presseartikel
Projektbeteiligte	Freude, Projektarbeit	Feedback	Kickoff Event	Daily Scrum	Virtuelle Meetings
Betriebsrat	Erhaltung von Arbeitsplätzen	Erfolgsmeldungen	Projektmanagement-Portal	Weblog	Intranet
Führungskräfte	Personalentwicklung	Lessons Learned	Pressekonferenz	Anzeigen	Mitarbeiterzeitschrift
Mitarbeiter	Informationen erhalten	Projektabschluss	Wikis	Feedback-Gespräche	Intranet

Abb. 1: Beispiel eines Kommunikationsplans.

- Welches sind die geeigneten Kommunikationswege um die jeweiligen Zielgruppen im Sinne des Projektzieles zu informieren und/oder zu beeinflussen?
- Wann und wie häufig sollen die jeweiligen Stakeholder informiert werden?

Die Abbildung 1 zeigt beispielhaft eine Auswahl an Stakeholdern, deren Erwartungen an das Projekt, deren Informationsbedarfe sowie die möglichen Kommunikationswege. Wie die Übersicht zu lesen ist, illustriert folgendes Beispiel:

Der Vorstand eines Unternehmens möchte mit einem Projekt die Unternehmensziele erreichen. Den Projektantrag kann der Vorstand sinnvollerweise in Form von Präsentationen, E-Mails und Meetings erhalten.

Die Kommunikationswege

Im Folgenden wird eine Auswahl an Kommunikationswegen näher betrachtet und deren Vor- und Nachteile skizziert.

Persönliche Gespräche

Das wichtigste Kommunikationsmedium ist und bleibt das persönliche Gespräch. Hier können alle relevanten Informationen direkt ausgetauscht werden. Die Teilnehmer können anhand von Gestik, Mimik und dem Tonfall schnell erkennen, wie die erhaltenen Nachrichten angekommen sind. Der persönliche Kontakt ist für die weitere Zusammenarbeit sehr wichtig. Wenn dem Namen ein Gesicht zugeordnet

werden kann, fällt es leichter miteinander zu kommunizieren und zu arbeiten. Insofern sind persönliche Gespräche gerade zum Beginn eines Projektes wichtig, damit sich die Projektbeteiligten frühzeitig persönlich kennenlernen.

Protokolle

Das Protokoll übernimmt in der Projektarbeit eine wichtige und oft unterschätzte Aufgabe: Die Verteilung relevanter Informationen, Entscheidungen und Maßnahmen als Ergebnis eines Projektmeetings an einen definierten Verteilerkreis ist für ein gemeinsames Verständnis im Projekt und für dessen weiteren Verlauf wichtig. Deshalb sollte das Protokoll von einem erfahrenen und vollständig im Projekt integrierten Protokollant verfasst werden.

Nur so kann sichergestellt werden, dass alle Informationen, Beschlüsse und Aufgaben genau erfasst werden. Der Verteilerkreis des Protokolls hat wiederum die Aufgabe, das Protokoll zu lesen, zu prüfen und ggf. Fehler korrigieren zu lassen. Des Weiteren sind Protokolle ein wichtiger Bestandteil der Projektdokumentation, die allen zugänglich sein sollte.

Daily Scrum

Im Rahmen des agilen Projektmanagements ist ein Daily Scrum ein tägliches Meeting des Projektteams, das nicht länger als 15 Minuten dauern sollte. Das Besondere ist, dass das Meeting in der Regel im Stehen durchgeführt wird. Die Teammitglieder tauschen sich kurz und knapp zu ihren aktuellen Aufgaben aus. Zur Visualisierung der Aufgaben kann ein Flipchart verwendet werden. An diesem

Glossar

Stakeholder

Stakeholder eines Projektes sind alle Personen, die ein Interesse am Projekt haben oder von ihm in irgendeiner Weise betroffen sind.

Quellen

- ▶ [1] Seminarunterlagen der ORDIX AG: IT-Projektmanagement und -Methoden, 2013
- ▶ [2] Fachartikel: Kommunikation in virtuellen Teams, Dr. Tomas Bohinc, Ausgabe 13/2011
- ▶ [3] Stakeholdermanagement – Ein Best-Practice-Ansatz, 9:pm Projektmanagement GmbH, 2004
- ▶ [4] http://scrum-master.de/Scrum-Meetings/Daily_Scrum_Meeting, abgerufen am 24.01.2014
- ▶ [5] Kommunikationsplanung nach Laswell, Günter Drews
- ▶ [6] Professionelles Projektmanagement in der Praxis, Dr. Harald Wehnes, Universität Würzburg, FB Informatik, Prof. Dr. P. Tran-Gia, 2008

Bildnachweis

- ▶ © freepik.com | Infographic free vector

können die Teammitglieder die Arbeitspakete mit Hilfe von z.B. Notizzetteln in die Kategorien „geplant“, „in Arbeit“ oder „abgeschlossen“ sortieren.

Die gemeinsame Betrachtung des Projektfortschritts bewirkt dabei ein „Wir-Gefühl“ im Team. Damit steigt wiederum die Motivation im Team das Projekt weiter voranzutreiben. Diese Meeting-Form erzeugt bewusst weiteren Gesprächs- und Informationsbedarf, auf den später in weiteren Gesprächen eingegangen werden kann. Der Scrum Master hat im Daily Scrum eine moderierende Funktion.

Instant Messaging (Chats)

Immer mehr Unternehmen haben zusätzlich zum E-Mail-Standardprogramm (z.B. Microsoft Outlook) ein Instant Messaging Tool (z.B. Microsoft Lync) im professionellen Umfeld im Einsatz. Der gemeinsame Chat ermöglicht Kollegen an unterschiedlichen Standorten einen direkten Austausch von Nachrichten. Durch die direkte Kommunikation entsteht ein Gefühl von räumlicher Nähe. Diese Art der Kommunikation eignet sich besonders für kurze, schnelle und informelle Anfragen. Bevor der Mitarbeiter mit E-Mails „bombardiert“ wird, ist es einfacher den Messenger zu benutzen. Der Empfänger kann direkt antworten, ohne extra eine E-Mail öffnen zu müssen. Für lange Texte und offizielle Berichte eignet sich dieser Kommunikationskanal allerdings nicht.

Virtuelle Meetings

In vielen Projekten sitzen die Mitarbeiter an verschiedenen Standorten, in verschiedenen Ländern oder sogar Kontinenten. Um trotzdem gemeinsame Meetings abhalten zu können, sind Videokonferenzen über das Internet ein hilfreicher Kommunikationsweg. Hier gelten die gleichen Regeln wie für ein Präsenzmeeting. Durch die Übertragung von Bild und Ton erscheint es den Konferenzteilnehmern als würden sie gemeinsam an einem Ort zusammensitzen. Trotzdem können sprachliche Barrieren, Bildverzerrungen und Tonausfall während der Übertragung das Verständnis für die Inhalte einschränken. Deshalb ist es insbesondere im Anschluss an ein virtuelles Meeting wichtig, ein Protokoll mit den relevanten Informationen und Beschlüssen an die Teilnehmer zu versenden.

„Lunch Dates“ / „Feierabendbier“

Die Einladung für ein „Lunch Date“ oder ein „Feierabendbier“ ist ein wichtiger Baustein zur Förderung einer guten Zusammenarbeit im Projekt. Die Projektmitarbeiter lernen sich fernab von Konferenz- und Büroräumen kennen. Sie erfahren etwas über den anderen Menschen und nicht nur über die Aufgabe, mit der er oder sie im Projekt betraut ist. Der Aufbau einer zwischenmenschlichen Beziehung fördert die gemeinsame Zusammenarbeit und schafft Vertrauen. Wenn das Vertrauensverhältnis zu der Person noch in den Startlöchern steht, sollte allerdings mit vertraulichen Informationen vorsichtig umgegangen werden.

Fazit

Am Ende ist es wie im richtigen Leben: Die richtige Mischung macht's! Eine ausgewogene Mischung der unterschiedlichen Kommunikationskanäle führt am ehesten zum Erfolg.

Die Schlüsselfaktoren für eine erfolgreiche Durchführung eines Projektes sind Motivation, Vertrauen und Kommunikation. Die Analyse und das Erkennen der Motivatoren der Stakeholder schafft die Grundlage für den Kommunikationsplan. Mit der Vermittlung des richtigen Inhalts an die Stakeholder wird Vertrauen und ein gemeinsames Verständnis geschaffen.

Die Auswahl der geeigneten Kommunikationswege ist eine sehr anspruchsvolle Aufgabe der Projektleitung. Eine gute Vorbereitung und der Mut neue Wege zu gehen ermöglichen es ihr, eine förderliche Mischung an Kommunikationswegen auszuwählen. Die Erstellung sowie die konsequente Umsetzung eines Kommunikationsplans schafft die Basis für eine erfolgreiche Projektarbeit.



*Irini Xanthopoulou
(info@ordix.de)*



Rätsel

Larry und die Bestellproblematik



Larry arbeitet im Moment an einem Webshop für einen Fahrradladen. Hierbei ist er für die Programmierung des Bestellvorgangs verantwortlich. Allerdings kämpft er mit einem Problem bei der Verarbeitung der Bestellungen. Kennen Sie sich mit CDI Events aus und können Larry helfen?

Die Bestellung erzeugt keine Meldung

Im ersten Schritt wird eine Bestellung erfasst, was auch schon funktioniert. Anschließend soll das Warenlager über die Bestellung informiert werden, damit das Fahrrad an den Kunden versandt werden kann.

Doch momentan wird das Warenlager nicht informiert. Dies führt dazu, dass das Fahrrad nicht versendet werden kann. Larry hat sich das Problem bereits angeschaut und schon einiges herausgefunden. Das Problem selber konnte er aber bisher noch nicht beheben.

Es besteht ein Problem mit CDI Events

In der Klasse `BestellungAufnehmen` wird ein CDI Event erzeugt und gefeuert. Dieses Event soll in einer weiteren Klasse `WarenLager` aufgenommen und verarbeitet werden.

Können Sie Larry helfen?

Leider kommt das Event nicht in der Klasse `WarenLager` an. Die Methode `bearbeiteBestellungEvent` (siehe Abbildung 1) wird also nie aufgerufen.

Können Sie Larry helfen das Problem zu lösen?

Larry freut sich auf Ihren Lösungsvorschlag! Senden Sie Ihre Antwort bis zum **11. April 2014** an kniffel@ordix.de.

Das Rätsel aus der letzten Ausgabe wurde gelöst

Larry bedankt sich mit einer kleinen Aufmerksamkeit für die richtige Lösung bei Arndt Stajta, Nicolai Seuss und Carlo Schoutz..

```
#####BestellungAufnehmen Klasse#####
import javax.enterprise.event.Event;
import javax.inject.Inject;
public class BestellungAufnehmen {
    @Inject
    Event<Bestellung> bestellungEvent;
    public void bestellungAbsenden() {
        Produkt fahrrad = new MountainBike();
        // [...]
        Bestellung b1 = new Bestellung(fahrrad);
        bestellungEvent.fire(b1);
    }
}
#####WarenLager Klasse#####
import javax.enterprise.event.Observe;
public class WarenLager {
    @Observe
    public void bearbeiteBestellungEvent(Bestellung b) {
        // hier möchte Larry das Event verarbeiten
    }
}
```

Abb. 1: Quellcode der Klassen Warenlager und BestellungAufnehmen.


Lösung Larry Rätsel 4/2013:

Die Prozedur, die zur Lösung des letzten Larry-Rätsels führt, finden Sie im Internet unter:

<http://www.ordix.de/ordixnews/larry.html>

Die Teilnahmebedingungen für unser Larry Rätsel finden Sie unter: <http://www.ordix.de/ordixnews/larry/teilnahme.html>

Bild: © istockphoto.com | Computer Geek: Two Thumbs Up | sdominick



Soft Skills sind der „emotionale IQ“ eines Menschen.

Fachwissen alleine ist heute nicht mehr alles. Vor allem in Berufen, in denen viel kommuniziert wird, in denen man gut im Team zusammenarbeiten muss oder in denen man permanent Kontakt zum Kunden hat, sind ausgeprägte Soft Skills unerlässlich.

Soft Skills wie Teamfähigkeit, Kommunikationsstärke und motivierender Umgang mit Kollegen sind wichtig und notwendig. Zusätzlich müssen Führungskräfte wie Mitarbeiter aber auch den kom-

petenten Umgang mit Konflikt-, Krisen- und Stresssituationen erlernen. Die Referenten unserer Seminare zeigen Ihnen, wie Sie Krisen erfolgreich bewältigen, Konflikte lösen und Stress auslösende Reize erkennen und vermeiden können.

Weitere Informationen zu unserem Seminarangebot im Bereich soziale Kompetenzen finden Sie in unserer Spezial-Broschüre „Projekt-/IT-Management“ oder auf training.ordix.de.