

Red Stack

Magazin

DOAG

SOUG
swiss oracle
user group

AOUG
AUSTRIAN ORACLE USER GROUP



Aus der Praxis

Cloud-Lösungen
in der Praxis

Im Interview

Prof. Dr. Thomas Rose,
Fraunhofer Blockchain-
Labor



Blockchain

Erfahrungen mit dem Oracle
Autonomous Blockchain
Cloud Service

Early Bird bis
zum 30. Sept.

2019
DOAG
Konferenz + Ausstellung
19. - 22. November in Nürnberg

PROGRAMM
ONLINE

2019.doag.org





Robert Szilinski
CEO, Vorstandsvorsitzender
esentri AG

Liebe Mitglieder, liebe Leserinnen und Leser,

Blockchain gehört dank einer breiten Berichterstattung und Kryptowährungen wie Bitcoin zu den Technologie-Superstars der Digitalisierung. Dabei wird das Potenzial der Blockchain oft entweder maßlos überschätzt oder völlig verkannt. Gerade im Enterprise-Umfeld ergeben sich bei genauem Hinschauen jedoch viele spannende Ansätze, um Geschäftsmodelle zu revolutionieren und bestehende Prozesse mit Smart Contracts und dezentralen Datenstrukturen völlig neu zu denken. Oracle erweitert derzeit seine Blockchain-Cloud-Services. Der große Durchbruch für Enterprise-Blockchains wird wahrscheinlich dann gelingen, wenn Firmen und Geschäftspartner auch kulturell den Mut aufbringen, neue Wege der Zusammenarbeit einzugehen.

Ich wünsche in diesem Sinne viele wertvolle Erkenntnisse beim Lesen dieser Ausgabe und darf mich als Vorstand für den Bereich Development nach vielen schönen Jahren bei der DOAG verabschieden. Mein Dank gilt allen langjährigen Mitstreitern für die tolle Unterstützung und insbesondere Niels de Bruijn, dem ich zu seiner Wahl als Vorstand herzlich gratuliere!

Ihr

MUNIQSOFT

TRAINING



Training

Training

Unter neuem Namen, aber in bewährter, hoher Schulungsqualität firmieren wir nun als

Munisoft Training GmbH

Wir haben unser Schulungsportfolio erweitert und würden uns freuen, Sie wieder bei uns zu begrüßen.

Telefon: 089 679090-40

Webseite: www.munisoft-training.de

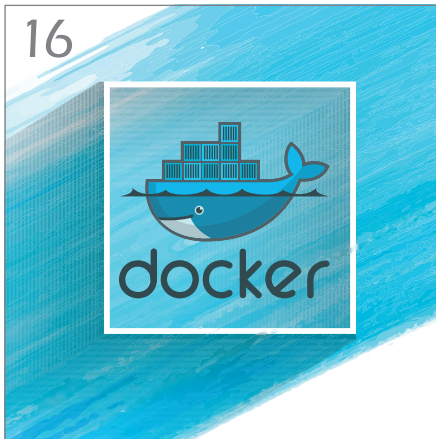
Tipps: www.munisoft-training.de/tipps



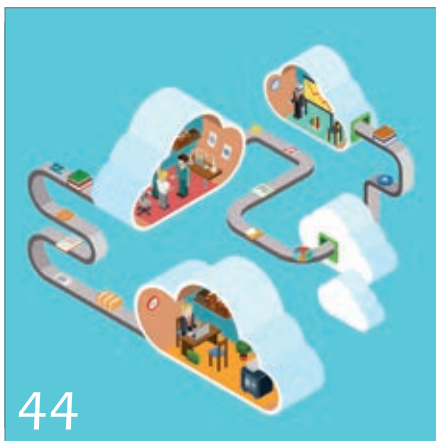
Unser Kundenservice:

- Öffentliche Datenbankschulungen:

z. B. Oracle APEX Fortschritt	16.-20.09.2019	1.990,-€ netto
• APEX im INTERNET NEU	26.-27.09.2019	1.290,-€ netto
• Neuerungen 12c/18c	07.-11.10.2019	1.990,-€ netto
• DBA Grundlagen	14.-18.10.2019	1.490,-€ netto
- Inhouse Schulungen individuell nach Ihren Wünschen, weitere Termine auf unserer Webseite.



Software-Qualitätssicherung und Integration mit Docker zum Test in virtualisierten Umgebungen



Vorstellung praxisbezogener Anwendungsszenarien und dafür passende Cloud-Lösungen



Im Vergleich: Implementierungen von Stateless Packages in PL/SQL

Einleitung

- 3 Editorial
- 5 Timeline
- 7 "Heute sind die ganz großen Anwendungsfelder die Produktnachverfolgung, das Identitätsmanagement und alles, was unter dem Begriff Machine to Machine Economy läuft."
Interview mit Prof. Dr. Thomas Rose

Blockchain

- 12 Erfahrungen mit dem Oracle Autonomous Blockchain Cloud Service für die Entwicklung eines dezentralen Rohstoffmonitors
Tim Rüb
- 16 Testautomatisierung mit Selenium in Docker
Davide Groppuso

Recht

- 20 Der Schutz von Geschäftsgeheimnissen und seine Neuerungen
Dr. Jan Bohnstedt

DevOps

- 22 Mit Jenkins in Richtung DevOps
Moritz Reinwald

APEX

- 30 REST-Schnittstellen und Application Express
Carsten Czarski
- 38 Agil unterwegs mit APEX in einem stark regulierten Umfeld
Hansjörg Grässlin, Christophe Girardey, Dr. Christian Wattering
- 55 Klick, Klick, Test – Automatisierte Tests mit APEX
Kai Donato

Cloud

- 44 Datenbank in der Wolke – Teil 3: Cloud-Lösungen in der Praxis
Borys Neselevskyi

Datenbank

- 15 Reaktion von Oracle auf die DOAG-Oracle-Support-Umfrage
Christian Trieb
- 50 Datentransfer mit Oracle Tools – Was ist möglich?
Christian Gohmann
- 66 Teile und herrsche: Partitionierung in der Oracle-Datenbank
Dierk Lenz

PL/SQL

- 61 Stateless Packages in PL/SQL und Hochverfügbarkeit von Applikationen
Athanasios Manolopoulos

JavaScript

- 69 Das Design von Formularen
Maximilian Liesegang

Intern

- 73 Termine
- 73 Neue Mitglieder
- 74 Impressum
- 74 Inserenten

✦ Timeline

3./4. Juni 2019

Auf der DOAG 2019 Datenbank in Düsseldorf liegt der Fokus im Wesentlichen auf der Datenbank selbst. Auf der zweitägigen Veranstaltung im Van der Valk Airporthotel erhalten die Teilnehmer in rund 60 Vorträgen und Projektberichten aktuelle Informationen und wertvollen Input für Ihre berufliche Praxis zu den Themen Datenbank und Exadata. Die Sessions gliedern sich unter den beiden Streams Datenbank und Engineered Systems.

7. Juni 2019

Im IMC-Webinar vom 7. Juni 2019 zeigt Michael Schulze von Opitz Consulting die Grundlagen von Docker sowie Architekturbeispiele ausgewählter Use Cases.

12. Juni 2019:

Am 12. Juni findet im Rahmen der Austrian Oracle User Group in Wien ein technisches Frühstück zum Thema Oracle und PostgreSQL – Zusammenspiel und Vergleich statt. Bei dieser sehr gutbesuchten Veranstaltung können sich die Teilnehmer darüber informieren, wie man Postgres- und Oracle-Datenbanken effizient miteinander verbinden kann. Im zweiten Teil der Veranstaltung werden dann typische Backup/Recovery-Konzepte von Oracle mit den Konzepten für Postgres-Datenbanken verglichen.

13. Juni 2019

Die Regionalgruppen Berlin/Brandenburg und Nürnberg finden sich zu ihren Regionaltreffen zusammen. In Berlin steht u.a. das Thema Oracle Support Escalation Handling/ Management Attention auf der Agenda. In Nürnberg zeigt Sabine Heimsath Tipps und Tricks mit dem SQL Developer.

19. bis 21. Juni 2019

Das DOAG-Team ist auf Mallorca zum Office-Workshop. Dieser wird von der synartiQ GmbH veranstaltet und widmet sich dem Thema Wissenstransfer und Wissensmanagement. Ziel des Workshops ist es zum einen, die Bedeutung und Notwendigkeit von Wissensmanagement deutlich zu machen und gleichzeitig die Kom-



plexität des Themas herunterzubrechen, damit es handhabbarer wird. Zum anderen können verschiedene Tools und Methoden direkt ausprobiert werden, sodass sie einen direkten Nutzen für die Praxis der KollegInnen schaffen. Auf einer Tour mit Rollern geht es am dritten Tag über kleine Landstraßen durch das hügelige Hinterland und durch ursprüngliche Dörfer.

26./27. Juni 2019

Das Berliner Expertenseminar zum Thema Forms & Reports findet in Berlin statt. Das Seminar von Jan-Peter Timmermann (Leitung Infrastruktur & Middleware Community) beschäftigt sich mit der Installation und der Konfiguration sowie dem Troubleshooting einer Oracle Forms & Reports-Umgebung. Dabei werden die Probleme der unterschiedlichen Betriebssysteme besprochen und die Erfahrungen der letzten drei Jahre mit dem Release 12c vorgestellt. Für jeden Teilnehmer wird eine eigene Oracle VBox-Umgebung auf Basis von Oracle Linux einschließlich aller genutzten Skripte bereitgestellt.

3. Juli 2019

Die jährliche ordentliche iJUG-Mitgliederversammlung findet in Stuttgart statt. Der Vorstandsvorsitzende Fried Saacke und seine beiden stellvertretenden Vorstände Tobias Frech und Stefan Koospal leiten die Veranstaltung.

5. Juli 2019

Die Programmsitzung DOAG 2019 findet in München im Hotel Hilton Munich Airport statt. Angesichts leicht gesunkener Bewerberzahlen für die Konferenz + Ausstellung und der Feststellung, dass die klassischen Oracle-Kernthemen nicht mehr so sehr im Fokus zu stehen scheinen, wird daher beschlossen, die DOAG auch für andere Themengebiete zu öffnen. Dennoch haben wir es wieder geschafft, ein tolles Programm mit über 400 Vorträgen für die diesjährige Konferenz + Ausstellung in Nürnberg zusammenzustellen.

11. Juli 2019

Der Call for Papers für die JavaLand 2020 beginnt. Diese findet vom 17. bis 19. März 2020 wieder im Phantasialand in Brühl statt.



12. Juli 2019

Das DOAG Datenbank Webinar mit Martin Bach mit dem Titel „Dank Swingbench die Oracle Datenbank besser verstehen“ steht auf dem Programm.



Dr. Dietmar Neugebauer
Ehemaliger DOAG-Vorstands-
vorsitzender

Aus der Ferne betrachtet: Nur wo DOAG drauf steht, ist auch DOAG drin ...

Das Ende des Oracle-Fiskaljahres im Mai ...

Wer da in Oracle-Datenbankprodukte investiert hat, der hat sicherlich vieles richtig gemacht. Laut Gartner ist die Oracle-Datenbank immer noch das weltweit führende Datenbankprodukt bezüglich der Technologie. Gerade auch, wenn es Richtung zukünftiger Weiterentwicklung geht.

Hier werden wichtige Entwicklungen immer weiter vorangetrieben. Die Maximum-Availability-Architektur garantiert höchste Verfügbarkeit mit dem Real Application Cluster, der Flashback-Funktionalität und den Möglichkeiten der Active-Standby-Lösung. Jedes neue Oracle Release hat Hunderte von neuen Features, die vor allem der Performance und der Automatisierung dienen, jedoch auch viele Anforderungen von Kunden umsetzen. Für alle diese Erweiterungen der Datenbank sind jetzt nicht einzelne neue Produkte entstanden, sondern es ist weiterhin die eine Datenbank, die man mit demselben Grundlagenwissen als Datenbankadministrator weiter betreiben kann. Dies gilt beispielsweise auch für die In-Memory-Lösung von Oracle, die sich dadurch entscheidend von Mitbewerbern unterscheidet.

Die Oracle-Datenbank profitiert dabei von ihrer unveränderten, ursprünglichen Basisarchitektur wie Transaktionssicher-

17. Juli 2019

Der DOAG User Day Hochverfügbarkeit öffnet für 40 Teilnehmer in Stuttgart seine Pforten. Die Tagesveranstaltung nimmt aktuelle Trendthemen und Dauerbrenner unter die Lupe. Von Oracle Data Guard über Oracle Real Application Cluster (RAC) bis Maximum Availability Architecture (MAA) – unsere Experten gehen auf unterschiedliche Aspekte der Hochverfügbarkeit ein. Natürlich darf dabei auch die Cloud nicht fehlen.

heit, Lesekonsistenz und keine Blockierung von Lese-/Schreiboperationen. Alles eigentlich selbstverständliche Eigenschaften, mit denen sich allerdings einige Datenbanken auf dem Markt dennoch schwertun.

Wieso tut sich Oracle nun gegenüber den Mitbewerbern auf dem Datenbank-Markt so schwer? Ist das Wissen um den technologischen Vorsprung von Oracle bei den Führungskräften in den Firmen nicht vorhanden? Was ist der Grund hierfür? Es ist sicherlich nicht immer einfach, sich gegenüber den Mitbewerbern zu positionieren. Aber Qualitätsmängel im Support oder nur schwer nachvollziehbare und schlecht kommunizierte Lizenzentscheidungen sind hier nicht vertrauensfördernd.

Auch Informatik-Studenten haben heute während des Studiums so gut wie keinen Kontakt zu Oracle-Datenbankprodukten. Im Hochschulprogramm hinkt hier Oracle weit hinter den Mitbewerbern hinterher. Obwohl Oracle hier mit Oracle XE und Oracle Standard Edition zwei gute Einsteiger-Produkte hat, kommen diese in der IT-Ausbildung eigentlich nicht vor. Hier könnte doch ein Zugang zur Oracle Cloud für Hochschulen sicherlich Verbesserung bringen, oder?

Wie gesagt, technologisch passt alles sehr gut – die autonome Datenbank übernimmt immer mehr Betriebsaufgaben, die Erweiterung Oracle Sharding mit einer horizontalen Partitionierung von Oracle-Datenbanken für große verteilte Datenmengen. Dies ist jedoch nur die eine Seite; die Agilität bei der Entwicklung von IT-Lösungen führt oft zu sehr kurzfristigen Entscheidungen, zunehmend auch zugunsten der Mitbewerber.

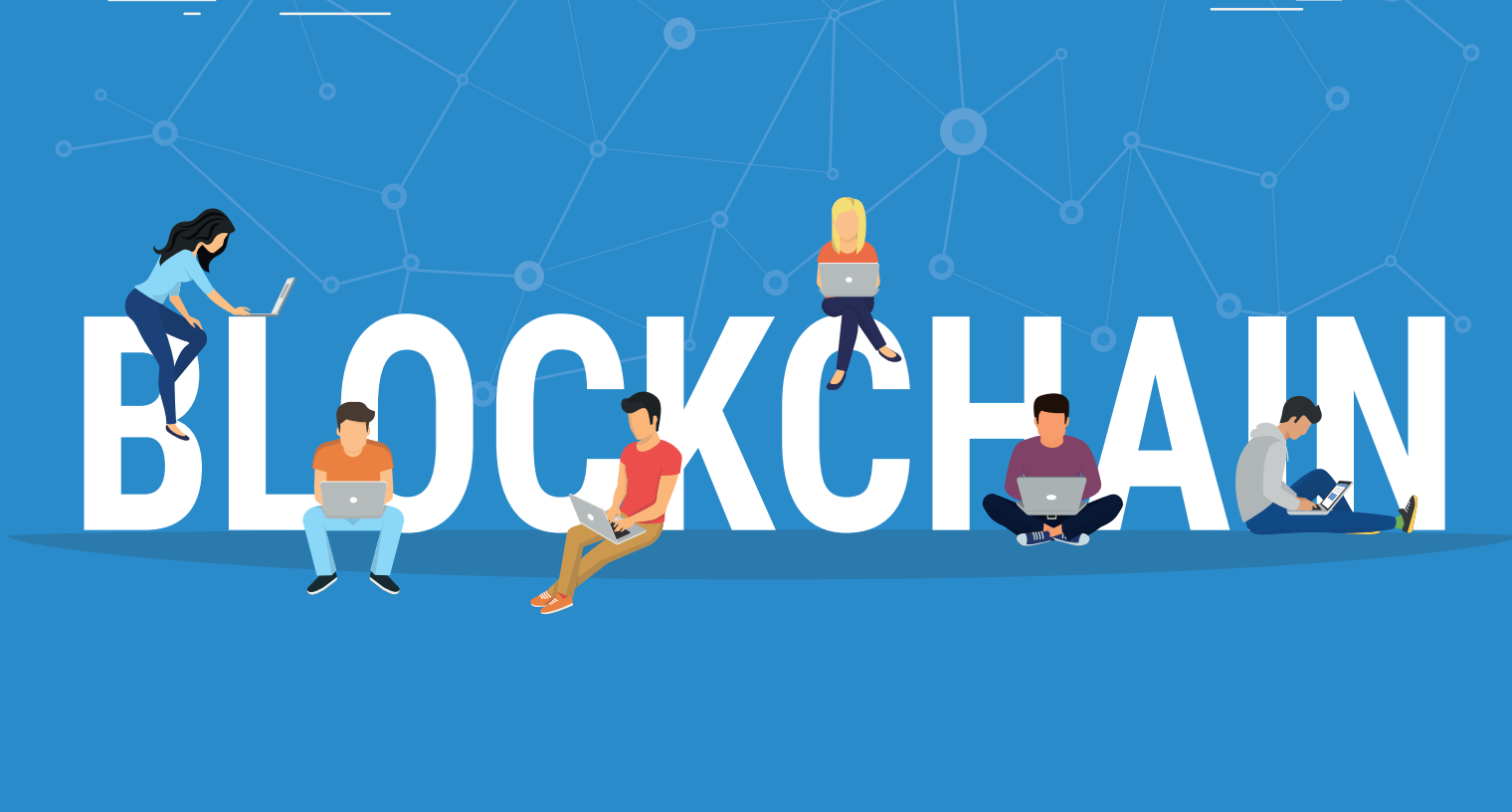
Die Oracle-Kunden von heute wollen ihre Investitionen in Oracle-Produkte gesichert wissen und zwar nicht nur auf Seiten der Technologie, sondern auch mit Kundenzufriedenheit und Vertrauen in langfristige und partnerschaftliche Geschäftsbeziehungen.

... aber wir stehen ja erst am Anfang des neuen Oracle-Geschäftsjahres

Richtigstellung

In der letzten Red Stack Ausgabe 3/2019 ist unserem Autor Dr. Dietmar Neugebauer in seinem Gastkommentar auf Seite 6 ein Fehler unterlaufen.

Die Firma Quest gehört nicht zu den Zukäufen von Oracle. Gemeint war hier der Zukauf von JD Edwards. Wir möchten dies hiermit richtigstellen und bitten unsere Leser um Entschuldigung.



„Heute sind die ganz großen Anwendungsfelder die Produktnachverfolgung, das Identitätsmanagement und alles, was unter dem Begriff Machine to Machine Economy läuft.“

Das Thema Blockchain und deren Anwendung in Unternehmen und Behörden steckt noch in einem frühen Anfangsstadium. Martin Meyer, Redaktionsleiter des Red Stack Magazin, sprach mit Prof. Dr. Rose vom Fraunhofer Blockchain-Labor und der RWTH Aachen über den aktuellen Stand sowie die zukünftig zu erwartenden Nutzungen und Entwicklungen der Blockchain.

Was bedeutet Blockchain eigentlich?

Blockchain ist eigentlich vor etwa 10 Jahren entstanden als Grundlage für eine Kryptowährung, mit der ich im Prinzip Finanztransaktionen an bestehenden Bankensystemen vorbei durchführen kann. Das war die ursprüngliche Anwendung der Blockchain. Die Technologie, die dort zur Anwendung kommt, ist als solches aber eigentlich nicht neu. Es sind nur bestehende Elemente geschickt kombiniert worden. Die Idee ist, dass man Transaktionen abwickeln kann, ohne über Vermittler (Intermediäre) wie hier zum Beispiel Banken zu gehen. Dies ist die Grundlage auch für viele andere Anwendungen im industriellen Bereich. Also im Prinzip ist eine Blockchain so etwas wie ein verteilter Transaktionsmanager, mit dem ich beliebige Transaktionen ausführen kann. Zu Beginn waren es die Finanztransaktio-

nen, sodass ich etwas von einem Teilnehmer zu einem anderen überweisen konnte. Heute sind die ganz großen Anwendungsfelder die Produktnachverfolgung, das Identitätsmanagement und alles, was unter dem Begriff Machine to Machine Economy läuft. Dies sind heute die drei großen Anwendungsklassen, die alle nichts mehr mit Kryptowährungen zu tun haben. Dahinter steht die Idee, dass man Transaktionen ohne einen Intermediär im Netz ausführen sowie dokumentieren kann und dass sie vor allem revisionssicher sind. Dies bedeutet, dass ich sie nachträglich nicht verändern kann.

Was sind die wichtigsten Grundlagen?

Grundlagen sind eigentlich, dass ich Transaktionen im Netz verteilt ausführen kann. Da die Informationen auf ganz vie-

len Knoten verteilt gespeichert werden, sind sie quasi angriffssicher. Ich muss nicht nur einen Knoten hacken, sondern ich müsste ganz viele Knoten hacken. Dazu kommt noch, dass ich nicht nur einen Block hacken müsste, sondern die komplette Kette, da die Blöcke innerhalb der Kette verknüpft sind. Das ist im Prinzip nicht möglich. Es gibt noch kein Beispiel dafür, dass eine Blockchain mal gehackt worden ist. Immer wenn mal etwas gehackt worden ist, dann waren nur die Umtauschbörsen betroffen. Durch die Verteilung aller Informationen ist diese Art von Transaktionsmanager angriffssicher und auch reaktionsschnell. Für einen Oracle-Datenbank-Anwender ist dies jedoch eine ziemlich dusselige Art der Datenspeicherung, da man praktisch ein Datum tausendfach repliziert speichert. Die zweite Grundlage, die viele Leute immer zu wenig sehen, ist die Tatsache, dass man mit einer Blockchain auch sichere Automationen durchführen kann. Das geht dann in Richtung Chain Codes und Smart Contracts. Ein sehr großer Teil der Anwendungen entfällt heute nicht nur auf das verteilte Transaktionsmanagement, also für die Datenverwaltung, sondern darauf, dass die Automation sicher ist. Man kann ganze Prozesse automatisch in einer Blockchain implementieren und dabei kann sich niemand in den Prozess einmischen und diesen verfälschen. Eine dritte wichtige Grundlage ist die Frage, wie die Knoten sich untereinander mitteilen, dass sie alle der gleichen Meinung sind. Dies ist die Konsensbildung, die es in der Datenbank-Welt so eigentlich nicht gibt. Dort geht man bei Transaktionen immer von einer Korrektheit aus. Eine Datenbank kann nie in einen inkonsistenten Zustand kommen. Da sie die Transaktionen aber beliebig verteilt und sie zum Beispiel Latenzen im Netz haben, müssen die Knoten sich darauf einigen, was sie für eine Meinung haben. Ich sage bewusst nicht, was korrekt ist, sondern sie einigen sich alle nur auf ein Meinungsbild. Das ist nicht Korrektheit. Dies hat man im Prinzip bei föderierten Datenbanken auch, da spricht man auch nicht mehr von Korrektheit. Diese Konsensbildung ist eigentlich auch schon ziemlich alt und stammt aus dem Kalten Krieg. Hier war es wichtig, dass sich Abschusssilos abstimmen mussten, dass alle zur gleichen Zeit schießen. Heute wird oft gesagt, dass die Konsensbildung einer Blockchain ein energverschwendendes Instrument sei. Es sei richtig, dass man diesen Proof of Work für die Konsensbildung nehme, also dass man kryptographische Rätsel löse, es gebe allerdings auch andere Möglichkeiten, Konsens zu erzielen. Man kann im Prinzip die Blockchain immer niederreden, aber man muss den Leuten einfach vermitteln, dass Blockchain nicht nur Bitcoin und Proof of Work ist. In der Bitcoin-Welt kann man keine großen Datenmengen verarbeiten, für Massenspeicherung wie bei einer Datenbank ist die Blockchain nicht geeignet.

Für welche Anwendungsfelder sehen Sie die größten Potenziale?

Im Moment zeichnet sich ab, dass im Finanzsektor noch viel experimentiert wird, über Proof of Concept hinaus gibt es da noch nicht viel. Viel tut sich in drei Bereichen. Der erste Bereich ist die Produktnachverfolgung, bei der es darum geht, dass man nachweisen kann, dass ein Produkt korrekt gebaut wurde, nur echte Bauteile beinhaltet oder auch dass man die Rechte hat, dieses Produkt überhaupt zu vertreiben. Hier sei als aktuelles Beispiel der Nachweis von Konfliktmineralien erwähnt. Bei der Produktion von Handys darf ein Hersteller nur Mineralien verbauen, so-

lange er nachweisen kann, dass er nur saubere Mineralien und keine sogenannten Blutmineralien einsetzt. Man sieht dies jedoch auch in vielen anderen Bereichen, wie beispielsweise beim Nachweis, dass in Kindernahrung keinerlei Verunreinigungen enthalten sein dürfen.

Der zweite große Bereich ist das Identitätsmanagement. Identitätsmanagement betrifft hier nicht nur Fragen zu Personalausweis oder beispielsweise zu Bordtickets, sondern auch, dass wir Qualifikationen haben. Nehmen wir zum Beispiel einen Wartungstechniker, der Fahrstühle warten darf. Diese einzusetzende Fähigkeit ist seine Identität. Sie haben als Mensch eigentlich ziemlich viele Identitäten und Fähigkeiten. Ich habe zum Beispiel die Identität, dass ich als Professor an der Universität Prüfungen abnehmen darf.

Dieses Gebiet der Identitäten ist sehr weit gefasst und ebenso haben natürlich auch Maschinen Identitäten. Ich darf mit einer bestimmten Maschine nur etwas Bestimmtes machen oder die Maschine braucht gewisse Wartungen, die durch gewisse Identitäten durchgeführt werden sollen. Identitäten sind das, was jemand im Allgemeinen tun darf. Identität heißt auch, dass im Cockpit eines Flugzeuges genug Expertise vorhanden ist. Dies kann man im Prinzip komplett auf Maschinen übertragen. Ein Auto hat eine Identität, die zum Beispiel durch die aufgespielte Software-Version festgelegt ist. Beim Einbau eines neuen Katalysators beispielsweise muss die Software aktualisiert werden.

Die dritte große Klasse umfasst die Machine to Machine Economy, also wenn Maschinen untereinander anfangen zu handeln. Wir haben zum Beispiel autonom fahrende Autos auf der letzten Hannover-Messe Industrie simuliert. Autos hatten Aufgaben, die sie auf einer Teststrecke zu erfüllen hatten, und wurden dafür entlohnt. Die Autos konnten untereinander handeln. So könnte in Zukunft ein selbstfahrendes Uber-Auto sich überlegen müssen, ob es sinnvoll ist unter Berücksichtigung von Faktoren wie einer zu zahlenden Straßenmaut oder Kosten, die für Autowäsche oder Parkgebühren anfallen, Passagiere zu befördern. Je nach Tageszeit und Verkehrslage könnte es dann selbstständig zu unterschiedlichen Entscheidungen kommen. Die Maschinen werden immer autonomer. Auch spiegelt sich dies in der Produktionstechnik wider. Elemente einer Produktionsstraße können flexibler eingesetzt werden, wobei diese auch in einen ökonomischen Wettbewerb gehen. Eine Maschine möchte nicht untätig herumstehen. Wir werden schon in Kürze immer häufiger erleben, dass Maschinen untereinander kommunizieren und handeln. Hier kommt dann auch die Revisionsicherheit der Blockchain zum Tragen. Wenn ich mit einer Maschine einen Auftrag entwickelt habe und diesen in der Blockchain ablege, ist er dort revisionsicher abgelegt und kann nicht verfälscht werden. Es besteht auch im Streitfall immer ein Nachweis, wer was gemacht hat. Wir brauchen keine dritten Parteien mehr, die uns als Vertrauensbasis dienen, um eine Korrektheit bei einer Transaktion sicherzustellen. Die Blockchain ist daher auch für sehr kleine Transaktionsvolumina geeignet

Was ist bei der Einführung von Blockchain-Technologien wichtig?

Man muss differenzieren. Zum einen stellt sich die Frage danach, was bei der Einführung von Blockchain-Anwendungen

wichtig ist und welche Eigenschaften von Blockchain noch benötigt werden. Man muss sich bei Anwendungen vorher anschauen, ob diese überhaupt blockchainaffin sind. Die meisten Dinge, die von Anwendern heute vorgeschlagen werden, lassen sich mit einer klassischen Datenbank viel besser und effizienter lösen. Blockchain ist grundsätzlich nur dann sinnvoll, wenn man bei einer vorhandenen Misstrauenssituation Vertrauen zwischen kooperierenden Parteien aufbauen will. Für uns am Blockchain-Labor ist die Blockchain eigentlich ein neues Kooperationswerkzeug, mit dem ich die Zusammenarbeit von Parteien unterstützen kann, die sich bisher noch nicht kennen. Wenn eine solche Netzwerkkonstellation nicht besteht, sollte eine Datenbank eingesetzt werden.

Zu rechtlichen Fragen kann ich noch nichts sagen, da sich alles bisher noch im rechtsfreien Raum bewegt. Es gibt noch keine Streitfälle und daher auch noch keine Rechts-Entscheidungen eines Gerichtes zum Thema Blockchain.

Technologisch befinden wir uns bei der Blockchain noch ungefähr da, wo das Internet sich Mitte der 80er Jahre befand. Es ist noch eine extrem unausgereifte Technologie. Kommunikation zwischen Blockchains ist ein offenes Problem. Man stelle sich vor, man könnte Datenbanken nicht untereinander kommunizieren lassen, da bräche die ganze industrielle Wertschöpfungskette zusammen. Es gibt noch keine standardisierten Entwicklungsmethoden, mit denen ich bei einer Anwendung zu einer wirklichen Technologie-Plattform komme. Es gibt auch noch keine Referenz-Bibliotheken, die wesentlich bei der Prozess-Entwicklung sind. Alles, was das klassische Software-Engineering-Repertoire ausmacht, ist noch nicht so präsent oder nicht vorhanden. Dies hängt aber damit zusammen, dass diese Technologie sehr jung ist. Es fehlt die Methodik, wie ich Blockchain-Anwendungen entwickeln kann. Wir haben bei uns eine Methodik entwickelt, mit der man entscheiden kann, ob eine Anwendung überhaupt blockchaintauglich oder -affin ist. Wenn dies entschieden ist, ist die nächste Frage, wie man das Netzwerk kooperierender Parteien nachhaltig umsetzen kann. Welche Vorteile haben die Parteien dadurch?

Was sind Smart Contracts?

Ein Smart Contract ist nicht ein rechtsverbindlicher Vertrag zwischen Parteien, sondern so etwas wie eine Definition einer Kooperations-Logik. Wenn sich zum Beispiel zwei Parteien auf einen Autoverkauf einigen, dann können sie Ratenzahlung vereinbaren; wenn der Käufer zu dem Auto geht und dieses aufschließen will, dann öffnet sich das Auto, die Zündung funktioniert. Das kann ich alles über Smart Contracts realisieren. Sobald der Käufer nicht mehr seine Raten bezahlt hat, öffnet der Smart Contract das Auto nicht mehr und der Smart Contract lässt das Auto auch nicht starten. Das ist mit Kooperations-Logik gemeint und die kann ich automatisieren. Dies ist die Idee eines Smart Contracts. Man kann so die Zusammenarbeit zwischen Firmen spezifizieren. Der Smart Contract ist dadurch gesichert, dass er in der Blockchain ausgeführt wird und daher nicht verfälscht werden kann. Diese gesicherte Anwendung von Programmen ist die wichtigere Anwendung der Blockchain. Es kann allerdings auch ein Nachteil sein, da die Bedingungen nicht veränderbar sind. Sie müssten eigentlich gewährleisten,

dass der Smart Contract bei der Definition korrekt ist und auch dann, wenn sie ihn in die Blockchain einstellen.

Hier braucht man Methoden der Programmverifikation oder Plausibilitätsprüfungen mit Modell-Checking; wenn sie mit Smart Contracts arbeiten, kommen sie eigentlich an der formalen Korrektheitsprüfung von Programmen nicht vorbei. Prinzipiell ist das ein Mittel der Automation. Heute sieht man oft, dass es Kooperationen wie beispielsweise eine Supply Chain gibt, die mit Smart Contracts definiert werden. Das nennt man dann auch oftmals Distributed Autonomous Organisation. Damit können Sie die Zusammenarbeit mit Parteien spezifizieren. Die Stadt Düsseldorf – so glaube ich – hat das mit einer dauerspezifizierten Parkerlaubnis gemacht. Wenn Anwohner eine Parkerlaubnis beantragen, wird das komplett über eine Smart-Contract-Implementierung abgewickelt. Dies ist insbesondere auch mit Blick auf den demografischen Wandel eine sehr attraktive Sache. Damit können Sie Dienstleistungen ohne Personaleinsatz anbieten. Sie müssen nur kontrollieren, dass der Smart Contract korrekt funktioniert.

Gibt es einen Hype?

In drei Jahren werden wir nicht mehr von einem Blockchain-Hype reden, sondern wir werden bestimmt ganz neue Elemente der Kooperations-Unterstützung haben. Dies wird sicher bleiben. Wir haben ganz andere Arten der Abstimmung und der Konsensfindung zwischen Parteien. Ich glaube auch, dass dies viele Auswirkungen auf Hersteller-Zulieferer-Beziehungen haben wird. Die Korrektheit von Lieferungen kann über Blockchain abgesichert werden und damit eine ganz andere Art der Vertrauensbildung ermöglichen. Ein Zulieferer muss nicht sagen, wie er garantiert, dass seine Produkte bestimmte Spezifikationen erfüllen. Er hat diese erfüllt und als Nachweis zum Beispiel Messprotokolle in der Blockchain abgelegt. Diese Protokolle werden aber nicht als Rohdaten abgelegt, sondern in Form einer eindeutigen Kodierung (Hashverfahren zur Konvertierung von beliebigen Zeichenketten in Zahlenformate). Der Hersteller kann somit nur die Bestätigung sehen, dass der Prozess gemäß Vorgaben ausgeführt wurde. Im Falle von Reklamationen des Abnehmers kann der Lieferant seine Messwerte offenlegen und durch den Hash der Werte in der Blockchain nachweisen, dass die offengelegten Messwerte dem Original entsprechen. Somit können sich Parteien die Korrektheit ihrer Transaktionen bestätigen ohne Rohdaten offen zu legen. Im Streitfall sind die Beweise dann nachvollziehbar. Die Blockchain wird in Zukunft nicht die Massendatenspeicherung übernehmen, sondern die Funktionen von Audit-Zertifikaten erfüllen können. Der Hype um Blockchain ist eigentlich schon vorbei. Vertrauensbildung in der Kooperation und die Zertifizierung für korrektes Arbeiten wird als neues Element bleiben.

Welche Chancen und Risiken sind mit der Technologie verbunden?

Die vielen Chancen beruhen auf der Tatsache, dass viel mehr Parteien zusammen kooperieren können, da die Kooperation immer offener wird und man eine neue Form der Vertrauensbildung hat, die nicht über so etwas wie einen Konsortialpartner oder Intermediäre läuft. Hier sehe ich viele Chancen für flexiblere Kooperationen. Risiken sind natürlich für alle diejenigen

Parteien mit Blockchain verbunden, die als Konsortialführer oder Intermediär arbeiten und nun aufpassen müssen, dass sie weiterhin im Geschäft bleiben können. Transaktionen können ohne sie eigentlich billiger abgewickelt werden. Gleichzeitig wird es keine Welt mehr ohne Amazon geben. Die Geschäftsmodelle der Plattform-Provider werden sich wahrscheinlich jedoch ändern. Allerdings haben diese Anbieter auch jede Menge Know-how über die Geschäftsbeziehungen ihres Umfelds. Die hat eine Blockchain nicht, diese ist nur ein billiger Transaktionsmanager.

Was sind die Hindernisse und Treiber der Umsetzung?

Ein typischer Treiber ist der Wunsch, vieles schneller, billiger und sicher zu machen. Wichtiger und interessanter erscheinen mir aber die Anwendungen, die nun Leuten neue Möglichkeiten der Zusammenarbeit eröffnen. Ein einfaches Beispiel: Notärzte arbeiten draußen vor Ort und müssen sehr schnelle Entscheidungen mit einer beschränkten Diagnostik treffen. Nachdem die Patienten ins Krankenhaus gebracht worden sind, kommen dort möglicherweise andere Diagnosen zum Tragen. Bei Atemnot beispielsweise stellt sich die Frage: Ist es das Herz oder die Lunge? Je nachdem, wie die Notärzte entscheiden, wird dieses oder jenes Krankenhaus angefahren und daraus ergeben sich schon Therapie-Konsequenzen. Die Notärzte bekommen in der Regel jedoch keinerlei Feedback, ob ihre Erst-Entscheidung richtig war. Dafür sind auch Datenschutzgesetze verantwortlich. Wenn ich jetzt aber stattdessen in einer Blockchain eine neue Kooperationsform zwischen den beiden Parteien Krankenhaus und Notärzte aufbauen kann, dann könnte ich eine Rückmeldung geben. Dies ist ein typisches Beispiel für eine neue Kooperationsmöglichkeit. Es wird ein Informationsrückfluss darüber möglich, wie gut die Entscheidung am Unfallort war; dadurch wird dem einzelnen Notarzt die Möglichkeit gegeben, seine Arbeitsergebnisse zu verbessern. Hier müsste sich meiner Meinung nach noch vieles verbessern, denn viele versuchen nur, bereits bestehende Prozesse zu verbessern. Man kann einen Organtransplantations-Prozess mit einer Blockchain revisions-sicher abbilden – neue Möglichkeiten der Zusammenarbeit zu etablieren, wäre allerdings noch schöner. Als ein großes Hindernis bei der Umsetzung einer Blockchain hat sich trotz des Vorliegens ganz vieler schöner Proofs of Concept, die alle möglichen Parteien überzeugen, die Schwierigkeit herausgestellt, ein Netzwerk dann so zu betreiben, dass die Parteien zusammen eine Blockchain organisieren. Die Parteien müssen dabei ihre Zusammenarbeit regeln oder sie gehen in die Öffentlichkeit, was wiederum viele gar nicht wollen.

Welche Effekte wird die Technologie auf die Wirtschaft und öffentliche Verwaltung haben und wie können sich Unternehmen und Behörden – bei aller Ungewissheit – heute am besten vorbereiten?

Ich glaube, dass die Unternehmen und Behörden sich zunächst überlegen müssen, welche Prozesse in oder zwischen ihren Unternehmen und Behörden für eine Blockchain tauglich wären, um neue Formen der Kooperation zu unterstützen. Das ist das Erste. Der Effekt, der am stärksten zum Tragen kommen wird, ist die Automation von Zusammenarbeit. Ich spezifiziere für eine Blockchain, wie ich untereinander zusammenarbeite. Die Dokumentenabwicklung mit ihren Kommunikationsabläufen

in der Binnenlogistik beispielsweise kann man in einer Blockchain transparent und automatisiert organisieren. Die Automation der Kooperation wird ein sehr starker Faktor sein. Hierbei kommt auch immer die Digitalisierung mit ins Spiel.

Welche aktuellen erfolgreichen Projekte können Sie uns nennen?

Mit dem Bundesverkehrsministerium für Infrastruktur haben wir eine Evaluation gemacht, die Plattformen betrifft. Dabei sind auch Prototypen entwickelt worden wie zum Beispiel Platooning für LKW-Verkehr auf Autobahnen. Mit dem Bundesamt für Migration und Flüchtlinge (BAMF) haben wir einen Blockchain-Prototyp gebaut, bei dem es um die Rückverfolgung von Identitäten geht. Dies sind zwei Flaggschiff-Projekte im öffentlichen Bereich, die wir am Fraunhofer Blockchain-Labor haben. Wir haben auch Blockchain-Projekte für Education, wo es darum geht, Ausbildungsnachweise zu verwalten. Das kann man für Hochschulbildungen machen, da sind der TÜV Rheinland, der TÜV Süd und mehrere Hochschulen mit dabei. Im Bereich Produktion haben wir Projekte, bei denen man über eine Blockchain nachweisen kann, dass Sachen korrekt ausgeführt wurden, somit Prozesse auf einer Maschine korrekt liefen und damit ein Produkt korrekt gebaut worden ist. Diesen Produktions-Demonstrator haben wir auf der diesjährigen Hannover-Messe vorgestellt. Bildlich ausgesprochen: Wenn ein Auto quasi selbst einen Blockchain-Knoten hat und nachweisen kann, dass es sich immer korrekt verhalten



Zur Person: Prof. Dr. Thomas Rose

Prof. Dr. Thomas Rose ist Professor für Medienprozesse an der RWTH Aachen und Leiter des Forschungsbereichs Risikomanagement und Entscheidungsunterstützung des Fraunhofer-Instituts für Angewandte Informationstechnik FIT in Sankt Augustin.

Seine Forschungs- und Kooperationsinteressen sind: Geschäftsprozessmanagement, Geschäftsmodelle und Governance sowie Prozessunterstützung für Großschadenslagen und medizinische Behandlungen (Standard Operating Procedures). Er ist Mitbegründer des Blockchain Experience Lab des Fraunhofer FIT, das Technologieplattformen, Anwendungen und Geschäftsmodelle für die Blockchain-Technologie untersucht. Ein Schwerpunkt liegt in dem Engineering validierter Smart Contracts für die Prozessautomation.

hat, dann brauchen Sie keine Plaketten mehr, um nachzuweisen, wer wohin fahren darf. Das Auto dokumentiert, wie es sich verhalten hat, und dadurch kann man hinterher nachweisen, dass keine Regel-Verletzungen stattgefunden haben. Man kann sogar so weit gehen, dass man ein Auto gar nicht zu etwas gebrauchen kann, das es nicht machen soll. Da wird dann beispielsweise in einer Blockchain abgespeichert, dass man in der Innenstadt seine Abgaswerte reduzieren soll. Ein Auto ist im Grunde genommen auch nur ein großes IoT-Device.

Die CDU hat kürzlich ein Positionspapier zur Blockchain verabschiedet, in dem auch eine Bundes-Blockchain für Behörden mitbehandelt wird. Die Frage ist, wo kann ich Prozesse in der öffentlichen Verwaltung automatisieren und die Korrektheit nachher nachweisen wie zum Beispiel zum Thema Migration und Flüchtlinge.

Ich sehe insbesondere einen Riesenmarkt für Blockchain in der Produktionstechnik, bei der Nachweise für die korrekte Arbeit von Maschinen erwünscht sind.

Was sind Kriterien für einen Blockchain-Einsatz?

Eine Blockchain hat nur Sinn, wenn verschiedene Parteien kooperieren und sich gegenseitig eine Zusammenarbeit vertrauensvoll bestätigen müssen. Sie werden niemals operative Daten in einer Blockchain ablegen, sondern immer nur die Bestätigung einer korrekten Zusammenarbeit.

Informationen:

<https://www.fit.fraunhofer.de/de/fb/cscw/blockchain.html>

DOAG Projektpreis initiiert: DOAG ruft zur Bewerbung auf

Dr. Frank Schönthaler, Promatis software

Mit dem neuen Preis sollen digitale Projekte mit intensivem Einsatz von Oracle-Technologien ausgezeichnet werden. Bis zum 16. September 2019 können Anwenderunternehmen ihre Bewerbung in den Kategorien Applications und Technologie einreichen und so die Chance auf ein Preisgeld von 2.000 Euro und drei Freitickets für die DOAG 2019 Konferenz + Ausstellung erhalten.

Herausgestellt und prämiert werden Projekte aus dem deutschsprachigen Markt, die Anwenderunternehmen mit intensivem Einsatz von Oracle-Produkten durchgeführt haben. Interessant sind unter anderem Neuimplementierungen, Migrationen, Optimierung und Erweiterungen. Dabei darf der Projektabschluss maximal zwei Jahre zurückliegen.

Um möglichst viele Anwender zu erreichen, wurde der Aufwand für die Bewerbung auf ein Mindestmaß reduziert. Die Auswahl der Gewinner erfolgt durch eine Preisjury, der Vertreter des DOAG-Vorstands und der DOAG-Communities sowie unabhängige Experten angehören.

Die Preisverleihung erfolgt während der DOAG 2019 Konferenz + Ausstellung. Dabei sind die Mitglieder der siegreichen Projektteams dazu eingeladen, ihr Projekt während der Jahreskonferenz in Nürnberg vorzustellen. Der DOAG Projektpreis wird 2019 erstmalig ausgelobt.

Weitere Informationen zum DOAG Projektpreis, Ansprechpartnern, Auswahlkriterien und zu Teilnahmebedingungen erfahren Sie auf der Homepage zum DOAG Projektpreis 2019.

Informationen

<https://www.doag.org/index.php?id=1386>



Dr. Frank Schönthaler
frank.schoenthaler@promatis.de

Erfahrungen mit dem Oracle Autonomous Blockchain Cloud Service für die Entwicklung eines dezentralen Rohstoffmonitors

Tim Rüb, esentri

Blockchain und Smart Contracts sind gerade in aller Munde. Doch warum sollte man sie einsetzen und wie kann der Blockchain Cloud Service von Oracle dabei helfen? Ein Erfahrungsbericht über die Chaincode-Entwicklung und die Cloud.

Letztes Jahr stellte Oracle den Autonomous Blockchain Cloud Service (OABCS) auf Basis des Blockchain-Frameworks Hyperledger Fabric der Allgemeinheit zur Verfügung. Dabei handelt es sich um einen Blockchain-as-a-Service für sogenannte Smart Contracts – Intelligente Verträge. Diese werden entweder in Golang, JavaScript oder Java verfasst und können über Schnittstellen (sogenannte Oracles) auf externe Datenquellen zugreifen.

Hyperledger Fabric, eine Open-Source-Plattform für dezentrale Smart Contracts (bei Hyperledger Fabric Chaincode genannt), ist eine sogenannte Permissioned Blockchain. Eine Teilnahme in einem Hyperledger-Fabric-Netzwerk ist im Gegensatz zu Public Blockchains wie zum Beispiel Bitcoin nur auf Einladung möglich. Dadurch ist auch der Prozess des Mining, wie bei diversen Crypto-Währungen erforderlich, hinfällig. Bei Hyperledger Fabric wird ein sogenanntes Konsens-System verwendet, bei dem eine Mehrheit der am Netzwerk beteiligten Parteien einer Änderung der Blockchain, dem sogenannten Proposal (Vorschlag), zustimmen muss, damit diese tatsächlich abgespeichert wird. [1]

Was kann der OABCS?

Der Cloud Service wurde ins Leben gerufen, um die Einrichtung eines Hyperledger-Fabric-Netzwerks deutlich zu vereinfachen und den Fokus der Entwicklung

mehr auf den Use Case des jeweiligen Projekts legen zu können. Es ist möglich, einfach und schnell ein Netzwerk aufzusetzen oder Teil eines bereits bestehenden Netzwerks zu werden. Auch können zu Testzwecken Beispiel-Chaincodes deployt werden. Es ist anschließend möglich, einen neuen Chaincode im Netzwerk zu verteilen und Abfragen an diesen zu stellen. Dies geschieht mit der REST-Schnittstelle, die der Cloud-Service zur Verfügung stellt. Standardmäßig kann jeder Peer nur mithilfe des GRPC-Protokolls kommunizieren. Um die vorhandene Infrastruktur zu visualisieren, stellt der Service ein Dashboard bereit, über das auch die Konfiguration erfolgt.

Wofür kann der OABCS genutzt werden?

Für Smart Contracts gibt es viele potenzielle Anwendungsszenarien; viele manuelle Prozesse lassen sich durch die Blockchain-Technologie automatisieren. Als Proof of Concept haben wir bei der esentri AG zusammen mit Studierenden der Hochschule Karlsruhe im Fachbereich Technik und Wirtschaft einen dezentralen Rohstoffmonitor entworfen, der das Aushandeln von Produkteinkaufspreisen im Einzelhandel automatisieren soll. Der Einkaufspreis orientiert sich dabei anteilig an den börsenaktuellen Rohstoffpreisen, was den Prozess des Handelns für beide Parteien fair gestaltet. Der aktuelle

Rohstoffpreis wird dabei über ein sogenanntes Oracle, eine externe Schnittstelle außerhalb des Smart Contracts, zur Verfügung gestellt. Realisiert werden kann das in Go über das Abfragen eines REST-API, das vorab im Chaincode definiert wird. Anschließend ist es für den Nutzer möglich, den finalen Preis eines Produkts über eine Oberfläche, die auf das REST-API des OABCS zugreift, abzufragen und dieses auch zu bestellen. Eine Bestellung wird dann von allen beteiligten Parteien validiert und auf der Blockchain gespeichert. Eine zeit- und damit auch kostenintensive Preisverhandlung wird damit vermieden. Auch ist es möglich, Bestellungen für ein Datum in der Zukunft anzulegen, um auf sinkende Rohstoffpreise zu spekulieren oder bei Unterschreitung eines gewissen Preises automatisch eine Bestellung auszuführen (wie bei einer Limit-Order im Aktienhandel). Um Trends zu erkennen und Produkte zu vergleichen, können verschiedene Preisentwicklungen in Diagrammen verglichen und deren Zusammensetzungen angezeigt werden.

Entwickeln von Chaincode mit Go

Wie bereits beschrieben, kann Chaincode mithilfe von Go, JavaScript und Java erstellt werden, wobei Go die in der Vergangenheit am stärksten unterstützte Sprache darstellt. Auf der Hyperledger-Fabric-Website findet sich in der Doku-

mentation sowohl Go- als auch JavaScript-Beispiel-Chaincode, Java ist nur teilweise vertreten. Bei esentri haben wir uns für die Entwicklung mit Go entschieden, um die Vorteile einer stark typisierten Sprache zu nutzen.

Am Anfang jedes Smart Contract muss das Hyperledger Fabric „shim“- und das „peer“-Package importiert werden, um Smart-Contract-Funktionen bereitstellen zu können und Antworten an die Peers im Netzwerk zu senden. Jeder Aufruf eines Chaincodes landet anschließend in der Invoke-Funktion des Chaincodes. Von dort aus werden die Argumente, die dem Invoke-Aufruf mitgegeben wurden, nach dem Funktionsnamen gefiltert und die passende Funktion aufgerufen (z. B. „createOrder“). Diese wiederum verarbeitet die restlichen mitgegebenen Argumente, wie den Inhalt und die Stückzahl einer Bestellung, die mit dem Funktionsaufruf erstellt werden soll. Nachdem Preise kalkuliert und eine Bestellung auf der Blockchain angelegt wurde, gibt die Funktion über das zuvor importierte Shim mit shim.success() den Erfolg der Funktion zurück. Dabei kann auch ein Payload mitgegeben werden, beispielsweise der eindeutige Key der Bestellung. Falls das Anlegen einer neuen Bestellung nicht erfolgreich war, weil etwa das API der Rohstoffbörse ausfällt und eine Preiskalkulation so nicht möglich ist, wird eine Fehlermeldung mit shim.Error() zurückgegeben.

Um die mit der Zeit wachsende Komplexität eines Chaincodes zu beherrschen

und Fehler vor einem finalen Deployment zu vermeiden – Stichwort „Code is Law“ –, sollte ein Smart Contract vorher auch getestet werden. Dazu kann das „testing“-Package, das bereits in Go integriert ist, verwendet werden. Um alle Funktionalitäten des Chaincodes darzustellen, müssen sie vorher gemockt werden. Dazu kann wieder das Package „shim“ importiert werden, das Funktionen wie getState und Invoke zur Verfügung stellt, um Schreib- und Lesevorgänge auf der Blockchain zu ermöglichen.

Alle Tests können anschließend mit dem Befehl „go test“ ausgeführt werden. Auch eine Analyse der Testabdeckung ist möglich. Dafür kann der Befehl „go test -coverprofile=coverage.out“ und anschließend „go tool cover -html=coverage.out“ ausgeführt werden. Anschließend wird detailliert für jede Zeile die Testabdeckung angezeigt. Weitere Analysen der Code-Qualität sind zum Beispiel mit SonarQube möglich, formatiert wird der Code je nach verwendetem Editor automatisch mit Go-Tools wie „gofmt“ oder „goreturn“.

Bevor der Chaincode auf den OABCS geladen wird, ist es sinnvoll, diesen auch lokal zu testen, um etwaige Fehler frühzeitig zu erkennen. Dies kann durch lokales, manuelles Testen erfolgen. Um den Chaincode lokal zu installieren, kann die Build-your-first-network-Anleitung auf der Hyperledger-Fabric-Seite verwendet werden. Auf diesem lokal erzeugten Netzwerk wird die gerade entwickelte

Chaincode-Version installiert. Abfragen sind anschließend über den Hyperledger Fabric CLI Docker Container möglich. Alternativ kann dafür auch ein simples JavaScript-Tool geschrieben werden, wie es im Hyperledger Fabric Samples Repository im Fabcar-Beispiel verwendet wird. [2] In diesem Tool wird über das Hyperledger Fabric SDK for Node.js eine Chaincode-Aktion ausgeführt.

Gerade bei großen Projekten ist es jedoch oft unübersichtlich und schwer zu überblicken, welche verschiedenen Funktionen der Chaincode anbietet, genauer gesagt, welche Parameter diese Funktionen akzeptieren und benötigen. Da die Parameter als Kette von Argumenten an den Chaincode übergeben werden, ist es sinnvoll, dies direkt in der Funktion in Form eines Kommentars zu erläutern. Es erleichtert sowohl das lokale Testen als auch die Dokumentation, die für die REST-Schnittstelle des OABCS notwendig ist.

Vor- und Nachteile des Blockchain Cloud Service

Im Gegensatz zu einem Standalone-Hyperledger-Fabric-Netzwerk ist der OABCS deutlich leichter in bestehende Infrastruktur zu integrieren. Wie oben beschrieben, ist es über die REST-Schnittstelle direkt möglich, eine Query oder Invoke auf der Blockchain auszuführen. Bei kleineren Anwendungen kann ein Frontend direkt, ohne eine zusätzliche

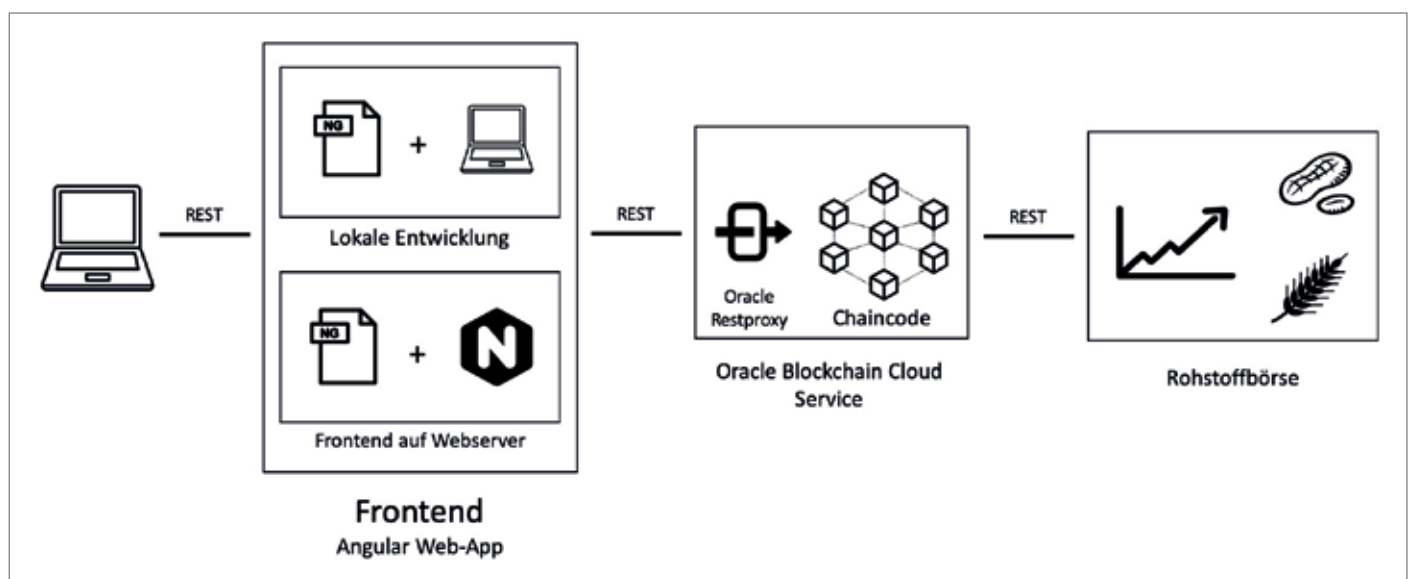


Abbildung 1: Beispielstack für den OABCS (Quelle: Tim Rüb)

Abbildung 2: Der Update Wizard (Quelle: Tim Rüb)

Middleware auf den Service zugreifen. Über das Dashboard lässt sich jederzeit der Zustand der einzelnen Peers kontrollieren, Logs auslesen und verwendete Chaincode-Versionen anzeigen, was sonst nur mühselig über Konsolenabfragen möglich ist.

Der OABCS bietet darüber hinaus einen einfachen Update Wizard, der eventuelle Änderungen an einem Smart Contract erleichtert. Dies ist zum Beispiel der Fall, wenn ein Bug vorhanden ist oder ein neues Feature hinzugefügt werden soll. In dem Service werden die Vorteile von Hyperledger Fabric mit den Vorteilen der Oracle Cloud gepaart, wie beispielsweise schnelle Verfügbarkeit, Performance und Sicherheit. Das ist vor allem durch die Anbindung an den Oracle Identity Cloud Service möglich, die die komplexe Zertifikats-Infrastruktur um Hyperledger Fabric erleichtert. Dabei kann für die Abfrage an die REST-Schnittstelle OAuth2 verwendet werden. Des Weiteren ist es möglich, Peers eines anderen Netzwerks über Zertifikate hinzuzufügen. So können auch Teilnehmer, die nicht die Oracle Cloud verwenden, dem Netzwerk hinzugefügt werden.

Während des Entwicklungsprozesses erwies sich die Dokumentation des Cloud Service teilweise als lückenhaft. So fehlten etwa eine detaillierte OAuth2-REST-Schnittstellen-Beispielabfrage des Cloud Service und eine Anleitung, in der die erforderlichen URLs und Secrets zu finden sind.

Lediglich Beispiele für Invoke und Query-Abfragen sind über den Oracle Blockchain Platform API Catalog verfügbar. Auch wird bei einer fehlerhaften Abfrage oft ein 400 Bad Request zurückgegeben, eine genaue Fehlermeldung bleibt dem Nutzer jedoch vorenthalten. Manuelle Fehlermeldungen aus dem Chaincode werden vom Service jedoch zurückgegeben, ebenso wie Peer-Fehlermeldungen (z.B. im Falle eines nicht akzeptierten Proposals).

Im Hintergrund verwendet Hyperledger Fabric eine sogenannte State-Datenbank, die wahlweise auf LevelDB oder CouchDB basiert. Auf dieser wird der aktuelle Zustand der Blockchain gespeichert, um schnellere Transaktionsgeschwindigkeiten zu ermöglichen. CouchDB bietet gegenüber LevelDB den Vorteil, dass im Chaincode komplexe Querys abgefragt werden können, da es sich nicht um eine Key-Value-Datenbank handelt. Der OABCS verwendet intern eine Oracle BerkeleyDB als State-Datenbank, Abfragen sind jedoch über den Chaincode auch mithilfe von CouchDB-Syntax möglich. Hierbei kann es jedoch zu kleinen Kompatibilitätsproblemen kommen, Oracle selbst empfiehlt in der Dokumentation, die SQL-Rich-Query-Syntax zu verwenden, falls der Chaincode ausschließlich auf dem Cloud Service betrieben wird. Andere Hyperledger-Fabric-Instanzen, auch für die lokale Chaincode-Entwicklung, unterstützen die SQL-Rich-Query-Syntax nicht. [3]

Fazit

Zusammenfassend ist zu sagen, dass der OABCS die Arbeit mit Hyperledger Fabric deutlich vereinfacht. Einmal eingerichtet, bietet es deutlich mehr Komfort, sowohl für Chaincode-Entwickler als auch für den finalen Betrieb.

Weiterführende Literatur:

Nitin Gaur, Luc Desrosiers et al. (2018): Hands-On Blockchain with Hyperledger. Packt Publishing

Quellen:

- [1] Hyperledger-Fabric-Dokumentation <https://hyperledger-fabric.readthedocs.io/en/release-1.4/blockchain.html>
- [2] Fabcar-Beispiel auf GitHub <https://github.com/hyperledger/fabric-samples/tree/release-1.4/fabcar>
- [3] Oracle-Blockchain-Cloud-Dokumentation <https://docs.oracle.com/en/cloud/paas/blockchain-cloud/user/query-state-database.html>

Über den Autor:

Tim Rüb studiert Wirtschaftsinformatik an der Hochschule Karlsruhe – Technik und Wirtschaft und schreibt seine Bachelor-Thesis über Continuous Delivery mit Hyperledger Fabric bei der esentri AG.



Tim Rüb
tim.rueb@esentri.com

Reaktion von Oracle auf die DOAG-Oracle-Support-Umfrage

Martin Meyer, Redaktionsleitung

Bereits im November 2018 hatten wir über die Ergebnisse [1] unserer DOAG-Oracle-Support-Umfrage berichtet [2], bei der erstmalig auch Cloud-Lösungen im Fokus standen.

Nachdem die überwiegend negativen Ergebnisse bereits auf der DOAG 2018 Konferenz + Ausstellung mit Oracle-Vertretern diskutiert worden waren, reagierte das Unternehmen nun auf die Unzufriedenheit vieler Kunden: Drei Mitarbeiter von Oracle – Scott Jesse, Global Customer Support Senior Director, Bryan Vongray, Senior Principal Technical Support Engineer, und Bill Burton, Consulting Member of Technical Staff – bezogen nach einem von der DOAG organisierten Webinar Stellung.

Das Webinar [3] fand am 15. Mai 2019 statt. Themen waren die Optimierung der Verfügbarkeit, Skalierbarkeit und Leistung von Oracle-Datenbankinstallationen mithilfe der autonomen Cloud-Architektur sowie Best Practices für Support durch gezielte Diagnose, Effizienzsteigerungen und einfachere Upgrades. Im Anschluss an das englischsprachige Webinar stellten sich die Oracle-Vertreter in einem Q&A den Fragen der Teilnehmer (Webinar-Unterlagen siehe [4]). Oracle plant, im Laufe des Jahres weitere exklusive Webinare für DOAG-Mitglieder anzubieten.

Als Reaktion auf die Unzufriedenheit mit dem Oracle-Support versprochen die drei Oracle-Vertreter regelmäßige Telefonkonferenzen mit Support-Kunden der Prioritätsstufe 1, die besonders lange auf Oracle-Support-Dienste angewiesen sind. Oracle zeigte sich zudem engagiert, speziell DOAG-Mitgliedern weiterzuhelfen.

Christian Trieb, Leiter des DOAG Competence Center Support [5] versicherte, dass sich Oracle-Kunden mit einem bereits eskalierten Service Request der Pri-

oritätsstufe 1 mit längerer Laufzeit vertrauensvoll an ihn wenden können und in diesen Telefonkonferenzen besonders intensiv betreut werden, um das offene Problem schnell und gut zu lösen. Christian Trieb möchte dafür sorgen, dass der eingeleitete Prozess dann auch vorankommt. Außerdem haben die Regionalleiter die Kontaktdaten von Oracle erhalten. Auch bei den Regionaltreffen möchte Oracle präsenter sein.

Ein Oracle Customer Success Manager, der sich um das Thema Datenbank und Support kümmert, war auf der DOAG 2019 Datenbank in Düsseldorf und hat für die DOAG 2019 Konferenz + Ausstellung (vom 19. bis 22. November in Nürnberg) einen Vortrag zum Thema Support eingereicht.

Auf der DOAG 2019 Konferenz + Ausstellung soll zudem eine Session zum Thema „My Oracle Support Accreditation“ mit Schulung und anschließender Abschlussprüfung angeboten werden.

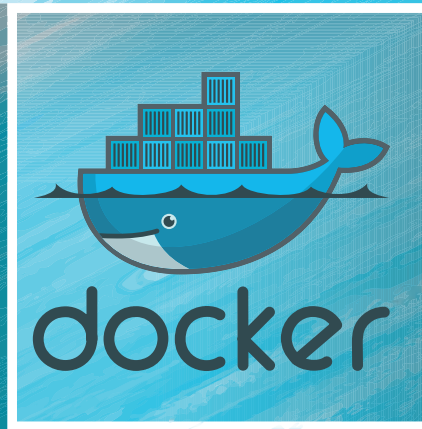
Grundsätzlich soll es also keine Änderungen in der Support-Politik geben, bei der Betreuung von Einzelfällen soll aber der Service verbessert werden.

Quellen

- [1] Ergebnisse der DOAG-Oracle-Support-Umfrage 2018: https://www.doag.org/formes/pub-files/10833356/docs/Presse/2018/Support-Umfrage_2018.pdf
- [2] Bericht zu Ergebnissen der Oracle-Support-Umfrage 2018: <https://www.doag.org/de/home/news/ergebnisse-der-oracle-support-umfrage-2018/detail>
- [3] Aufgezeichnetes Webinar vom 15.5.2019: Oracle Support Tools Bundle <https://www.doag.org/de/home/news/aufgezeichnet-oracle-support-tools-bundle/detail>
- [4] Unterlagen zum Webinar: https://www.doag.org/formes/pub-files/11308321/TFA_and_Uilities_v10.pdf
- [5] DOAG Competence Center Support: <https://www.doag.org/de/themen/competence-center/support>



Christian Trieb
christian.trieb@doag.org



Testautomatisierung mit Selenium in Docker

Davide Groppuso, MT AG

Testautomatisierung hat in den letzten Jahren eine immer größere Bedeutung in IT-Projekten bekommen und sich als fester Bestandteil im DevOps-Umfeld etabliert. In diesem Artikel wird der Einsatz von Selenium als Testautomatisierungs-Tool für die Software-Qualitätssicherung einer Webanwendung aufgezeigt. Des Weiteren erfolgt eine Beschreibung der Integration mit Docker zur Testausführung in virtualisierten Umgebungen.

Der Einsatz von Testautomatisierung in IT-Projekten basiert meist auf der Anforderung, agile Softwareprojekte im Unternehmen durchzuführen. Durch die häufigen Releases und kurzen Release-Zyklen in agilen Projekten wird die Anzahl von Deployments deutlich höher als bei Projekten, die nach klassischen Vorgehensweisen durchgeführt werden. Ist es nun der Anspruch, jedes Software-Release vollständig auf seine Funktionsfähigkeit zu testen, steigt der Zeit- und Kostenaufwand enorm an. Umfangreiche Tests sind dann kaum mehr zu stemmen, wodurch eine hohe Testabdeckung kaum noch möglich ist. Darüber hinaus ist das manuelle Reproduzieren von Testfällen als reine Fleißarbeit anzusehen, was früher oder später zu einer hohen Fehleranfälligkeit führen wird.

Die Automatisierung von Testfällen ist zwar ebenfalls mit Aufwand verbunden, da die Testautomatisierung meist ein eigenes Softwareprojekt ist, jedoch wird sich ihr Einsatz mittel- bis langfristig rentieren und somit eine Kostenersparnis gegenüber dem ständigen manuellen Testen mit sich bringen.

Dennoch sollte die Testautomatisierung stets durch manuelle Tests von Fachexperten ergänzt werden, da sie ausschließlich die im Vorhinein festgelegten Testkriterien abdeckt. Ein manueller Tester schaut meist auch rechts und links von den definierten Testfällen. Ist die Testautomatisierung einmal implementiert, lässt sie sich beliebig oft durchführen, beispielsweise nach jedem Deployment auf einer QA-Instanz. Der manuelle Aufwand beschränkt sich dann ausschließ-

lich auf die Auswertung der generierten Testberichte.

Selenium

Selenium ist ein Framework für die Erstellung von automatisierten Oberflächen-Tests von Webanwendungen.

Selenium ist primär ein Werkzeug, um die Benutzeroberfläche einer Anwendung zu testen. Durch die Überprüfung des Frontend können allerdings auch die Anwendungslogik und die anzuzeigenden Daten zumindest indirekt getestet werden. Für ausführliche Backend-Tests und Abfragen zur Datenqualität ist Selenium jedoch nicht geeignet.

Der Einsatz von Selenium als Testwerkzeug bietet die Möglichkeit, sowohl

Testsequenzen mit der Selenium IDE aufzuzeichnen und abzuspielen, als auch, diese programmatisch zu erstellen. Selenium unterstützt hierfür mehrere Programmiersprachen, unter anderem Java, C# und Python. Die Programmierung der Testfälle bietet hierbei einige Vorteile gegenüber der Testaufzeichnung.

So können im Code Validierungs- und Verifikationspunkte festgelegt und eingebaut werden, während die Testaufzeichnung in der Regel „nur durch die Oberfläche klickt und tippt“ und die Anwendung auf reine Funktionalität testet. Der Einbau von Validierungen ermöglicht eine bessere Überprüfung vorher definierter Akzeptanzkriterien. Wird beispielsweise eine Produkt-Liste in einem Webshop ausgewertet, kann eine Validierung sicherstellen, dass nicht zu viele oder zu wenige Produkte in der Liste angezeigt werden. Mit einer Testaufzeichnung ist diese Validierung nicht möglich. Das Definieren der Akzeptanzkriterien im Vorhinein verbessert zudem das gewünschte Ziel-Bild der Anwendung und sorgt dafür, dass es unter verschiedenen Stakeholdern dasselbe Verständnis darüber gibt, wie die Anwendung aussehen soll.

Weiterhin lassen sich codebasierte Testfälle besser strukturieren, wiederverwenden und warten. Durch die Modularisierung von Code-Bausteinen sind die Testfälle deutlich eigenständiger nutzbar. Gerade die Wiederverwendbarkeit von Code und die Anpassung an neue Anforderungen ist bei programmierten Tests von Vorteil. Weiterhin lässt sich im Code die Identifizierung der Elemente besser steuern. Insbesondere wenn Web-Elemente keine eindeutige ID haben, tendieren Aufzeichnungstools zur Nutzung von relativen oder absoluten Pfaden. Bei

```
java -Dtest.env={dev|local|qa} -jar .\target\selenium.jar
```

Listing 1: Anwendung mit Parametern starten

```
docker build . -t applicationtest
```

Listing 2: Docker Image bauen

```
docker run --rm --shm-size 2g -e TESTENV_PARAM="dev" --name applicationtest -it applicationtest
```

Listing 3: Docker-Tests starten

der Programmierung ist eine stückweise Ermittlung des Elements oftmals sinnvoller. Gerade bei sehr beweglicher Software sind die aufgezeichneten Tests später nicht mehr verwendbar. Ein weiterer erheblicher Nachteil von aufgezeichneten Tests ist die fehlende Möglichkeit, diese in einem Container auszuführen.

Für das Programmieren der Testfälle nutzt Selenium das WebDriver API. Unterstützt werden alle gängigen Browser wie Chrome, Firefox, Safari, Edge, IE und Opera. Bei der Erstellung von Cross-Browser-Tests sind insbesondere die Umgebungsvoraussetzungen für die jeweiligen Browser zu beachten. So benötigt Safari ein Apple-Betriebssystem, sofern man die neueste Safari-Version nutzen möchte.

Vor der Programmierung der Testfälle ist es wichtig, sich eine gute Vorgehensweise zu überlegen, wie die Tests sinnvoll gruppiert und zugeordnet werden können. Liegt ein gut konzipiertes Testmanagement vor, lassen sich die Testfälle im Anschluss besser warten und gegebenenfalls bei Änderungen anpassen. In diesem Zusammenhang sollte das Page Object Model (POM) genutzt werden. Das

POM kann als Java-Abbild einer HTML-Seite betrachtet werden. Im POM werden im Regelfall die Web-Elemente durch `findElement(By.id(„field1“))` ermittelt. Abgesehen von der ID eines Elements können Web-Elemente auch mittels `css`, `xpath`, `class` etc. gefunden werden. Ziel des POM sollte es sein, eine Java-Klasse zu haben, in der alle benötigten Elemente je einer HTML-Seite abgebildet werden.

Damit die Tests nach fertiger Entwicklung ausgeführt werden können, wird zunächst eine ausführbare Datei (.jar bei Java) benötigt. Diese kann mit einem Build-Tool wie Maven erstellt werden.

Das .jar-File kann an einem zentralen Ort, beispielsweise auf dem Repository-Manager Nexus abgelegt werden. Der Einsatz eines Nexus bietet den Vorteil, dass sowohl die Docker-Container (lediglich für Firefox und Chrome Tests) als auch weitere Clients auf die Datei zugreifen können.

Beim Starten der Tests wird dann das entsprechende .jar-File übergeben. Sofern die Anwendung parametrisiert ist, sollten diese beim Start gesetzt werden, wenn dies von den Default-Einstellungen abweicht.

Das Beispiel in Listing 1 startet die Anwendung mit dem Setzen des Parameters `test.env`. Weitere Parameter können dabei ergänzt werden. Bei Nutzung des Selenium WebDriver empfiehlt es sich, den `RemoteWebDriver` zu nutzen und den für den Testfall gewünschten Browser als Parameter zu übergeben. Auf dessen Basis wird dann im Programm der entsprechende browser-spezifische WebDriver erstellt. Bei der Erstellung der browser-spezifischen Driver ist darauf zu achten, dass die Browser-Profile unterschiedlich erstellt werden. Eine wichtige Einstellung

```
1 #some comment
2 @Important
3 Feature: Tests for RedStack
4
5 @ImportantTest1
6 Scenario: Suche bei Google nach Cucumber
7     Given Google Webseite ist geöffnet
8     When Cucumber eingegeben wird
9     And Enter gedrückt wird
10    Then Soll die Suche Ergebnisse anzeigen
```

Abbildung 1: Cucumber-Datei (Quelle: Davide Groppuso)

ist unter anderem das Erlauben der Or- tungsfunktion, sofern bei der zu testen- den Applikation eine Standortermittlung vorliegt. Neben der Erzeugung des Web- Driver sollten in einer Browser-Klasse zentrale Browser-Funktionen ausgelagert werden, die in mehreren Testfällen ge- nutzt werden.

In der folgenden Liste sehen Sie eini- ge Beispiele, die in keiner Browser-Klasse fehlen sollten:

- getTitle()
- getUrl()
- loadPage()
- openNewTab()
- getScreenshot()
- confirmAlertMessage()

Die Funktion getScreenshot() ist insbe- sondere im Fehlerfall äußerst wichtig, da es sich empfiehlt, die Reports um ent- sprechende Screenshots (siehe Abschnitt Cucumber) der Anwendung zu erweitern.

Docker

Bei Docker handelt es sich um eine Con- tainer-Technologie, die es ermöglicht, Li- nux-Container zu erstellen, zu betreiben und zu verwalten. Docker versteht sich als eine Art leichtgewichtige und modu- lare virtuelle Maschine. Docker-Container können flexibel skaliert und auf einem Zielsystem beliebig oft kopiert und ge- startet werden.

Für das Ausführen der Tests im Do- cker-Container müssen vorher die Docker Images (je eines für Chrome und Firefox) erstellt werden. Die Dockerfiles zur Erstel- lung der Images finden Sie in mehreren GitHub-Projekten. Falls Sie eine bestimm- te Version des Browsers testen möchten, ist darauf zu achten, dass Sie eine dazu kompatible Version des entsprechenden Drivers nutzen.

Sobald Sie das Dockerfile herunterge- laden haben, können Sie das Image, wie in Listing 2 beschrieben, bauen.

Zum Starten der Tests aus dem Do- cker-Container heraus verwenden Sie den Aufruf aus Listing 3.

Die Integration in Jenkins (oder in ein anderes CI-System) sorgt anschließend unter anderem dafür, dass die Tests nicht nur automatisiert ausführbar sind, son- dern dass sie beispielsweise auch auto-

```
Given("Google Webseite ist geöffnet", this::openGooglePage);  
When("Cucumber eingegeben wird", this::enterCucumber);  
And("Enter gedrückt wird", this::clickEnter);  
Then("Soll die Suche Ergebnisse anzeigen", this::showResults);
```

Abbildung 2: Steps-Datei (Quelle: Davide Groppuso)

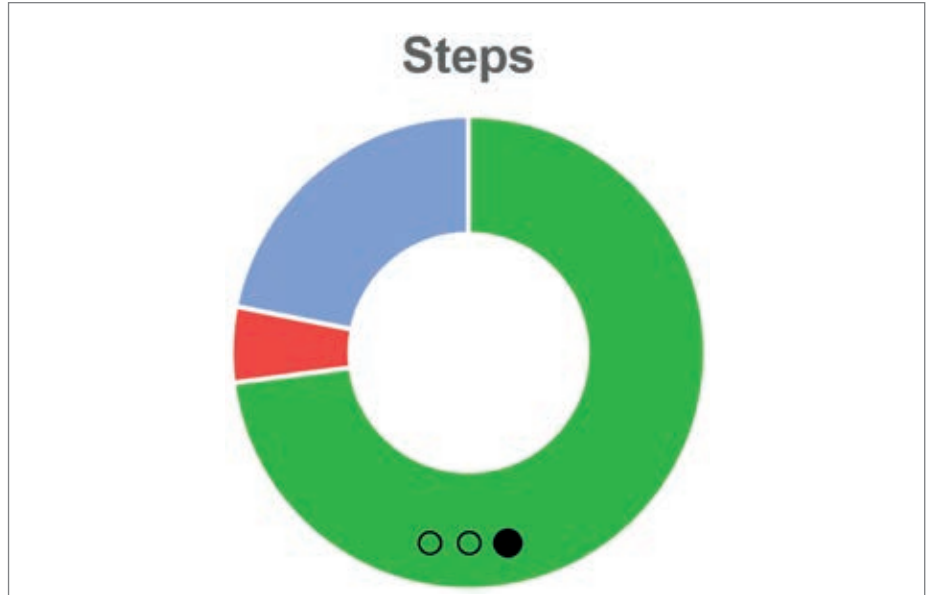


Abbildung 3: Übersicht der Testfälle als Donut-Chart (Quelle: Davide Groppuso)

matisiert getriggert werden. Neben dem zeitgesteuerten Build zu einem festen Zeitpunkt kann auch das zentrale Repo- sitory abgefragt werden. Die Tests bedür- fen somit keines manuellen Anstoßes.

Cucumber

Cucumber ist ein Tool, das das Behavior- Driven-Development (BDD) unterstützt und aus dem Umfeld des automatisierten Testens entstand. BDD basiert darauf, Software auf Basis von definierten Akzep- tanzkriterien zu entwickeln. Diese Akzep- tanzkriterien dienen als kommunikative Basis zwischen dem Entwicklerteam und den übrigen Stakeholdern. Akzeptanz- kriterien werden in natürlicher Sprache festgehalten, sodass sowohl Fachanwen- der als auch Informatiker und Program- mierer diese verstehen. Dies führt zudem zu einem gemeinsamen Verständnis aller Projektbeteiligten. Das Überführen der Anforderungen von natürlicher Spra- che nach Programmcode ist Aufgabe der Softwareentwickler.

Cucumber versteht sich selbst als eine Schnittstelle zwischen der IT und

der Fachabteilung. Mithilfe von Cucum- ber wird an genau der Stelle angesetzt, in der es darum geht, fachliche Anforderun- gen aus der natürlichen Sprache in Pro- grammcode zu überführen.

Cucumber führt die Anforderungsana- lyse und das automatisierte Testen durch die Gherkin-Notation zusammen. Der Entwickler programmiert anschließend die zerlegte Schritt-für-Schritt-Anforde- rung in die Programmiersprache seiner Wahl aus.

In Abbildung 1 ist eine für Cucumber erstellte Feature-Datei zu sehen. Ein Sze- nario repräsentiert dabei ein vorher defi- niertes Akzeptanzkriterium innerhalb der Produkthanforderungen. Das Akzeptanz- kriterium wird anschließend in mehrere Schritte (engl. Steps) unterteilt.

Hinter einem Step verbirgt sich dann letztendlich Programmcode in der jeweili- gen Programmiersprache, beispielsweise Java. Ein Beispiel für das „Umwandeln“ ei- nes Steps in eine Java-Funktion ist in Ab- bildung 2 zu sehen.

Nach der Testdurchführung wird in ei- nem generierten HTML-Bericht der Test farblich mit einem Status gekennzeich- net. Hierbei wird jeder einzelne Schritt

```
docker run --rm --shm-size 2g -e TESTENV_PARAM="dev" -e CUCUMBERTAG_PARAM="@tag" --name applicationtest -it applicationtest
```

Listing 4: Docker-Tests mit Tag „@tag“ starten

aus *Abbildung 1* markiert, sodass später klar erkennbar ist, welcher Testschritt zu einem Fehler geführt hat. Optional lassen sich die Berichte um Bilder, etwa Screenshots der Applikation, erweitern.

Durch die Nutzung von Cucumber als Bibliothek und dem Cucumber-Plug-in für Jenkins können die benötigten Reports bereits out of the box und ohne große Zusatzentwicklung erstellt werden. Neben dem Step-für-Step-Report bietet Cucumber auch direkt eine „managementgerechte“ Übersicht als Donut-Chart (*siehe Abbildung 3*) an oder auch eine Auflistung zum Beispiel nach Tags.

Die Nutzung von Tags in Cucumber bietet die Möglichkeit, die Tests besser zu gruppieren. Außerdem können die Tests

auch Tag-gesteuert ausgeführt werden (*siehe Listing 4*), es müssen also nicht zwingend immer alle Tests auf einmal durchgeführt werden.

Dies vereinfacht zum einen die Entwicklung, zum anderen ist es einfacher, die Tests über ein bestimmtes Tag auszuwerten und sich den Trend über einen spezifischen Bereich der Applikation anzuzeigen.

Über den Autor

Davide Groppuso arbeitet als IT-Berater mit Schwerpunkt APEX Development bei der MT AG in Ratingen. Neben der Erstellung von Datenmodellen und der Ent-

wicklung von Webapplikationen auf Basis APEX gehören das Reporting mit Jasper-Reports und MTdoxx zu seinen Schwerpunktthemen. Darüber hinaus beschäftigt er sich mit Continuous Integration und Delivery im Rahmen der Datenbankentwicklung sowie der Testautomatisierung von Webapplikationen.



Davide Groppuso
davide.groppuso@mt-ag.com



Alles in einem Pack!
Oracle Database Appliance
Oracle Database SE 2
Professionelle Konfigurierung
dbi DMK Management Kit
Erweiterungen möglich (DR/Perf)



Oracle ready-to-use!
Die sichere und sofort startbereite Oracle-Infrastruktur.

Consulting · Service Management (SLAs) · Lizenzmanagement · Workshops
Phone +41 32 422 96 00 · Basel · Bern · Nyon · Zürich · dbi-services.com



Der Schutz von Geschäftsgeheimnissen und seine Neuerungen

Dr. Jan Bohnstedt, Bartsch Rechtsanwälte PartG mbB

**TOP
SECRET**

Der Wandel zur „Informationsgesellschaft“ wird von Digitalisierung und Industrie-4.0-Projekten begleitet. Erfolg misst sich zunehmend an der Fähigkeit, nicht nur IP-Rechte wie Patente und Urheberrechte, sondern auch Wissen und Daten zu generieren und zu schützen. Der Schutz von Daten war lange problematisch. Am 21.03.2019 wurde nun das Geschäftsgeheimnisgesetz verabschiedet. Es setzt eine neue europäische Richtlinie um, die den bislang lückenhaften Schutz von Geheimnissen und damit auch Daten auf völlig neue Füße stellt. So wichtig, wie die DSGVO für die Daten von natürlichen Personen ist, so wichtig ist das Geschäftsgeheimnisgesetz für die Daten von Unternehmen.

Was ist neu?

1. Geheimhaltungsvereinbarungen bekommen neuen Biss

Die wenigsten Geheimhaltungsvereinbarungen (NDAs) enthalten Vertragsstrafen für den Fall der Nichtbeachtung. Das wäre jedoch eigentlich wünschenswert, weil es ohne sie nahezu unmöglich ist, den entstehenden Schaden bei einem Verlust der Geheimhaltung zu berechnen und zu beweisen. Das Gesetz zwingt allerdings den Inhaber des Geheimnisses nachzuweisen, welche finanzielle Einbuße er genau erlitten hat, weil das Geheimnis öffentlich wurde. Klar ist, dass wichtiges Know-how nun dem Wettbewerb zufließen kann. Aber wie hoch, gerechnet in Euros, ist der Schaden?

Das Geschäftsgeheimnisgesetz eröffnet nun die Möglichkeit, beim Bruch eines NDA eine Pauschale als Schadenersatz zu verlangen, ohne nachweisen zu müssen, dass ein Schaden entstanden ist oder in welcher Höhe. Damit ist faktisch jeder bereits geschlossene NDA nachträglich mit einer gesetzlichen Regelung versehen, die wie eine Vertragsstrafe wirkt (§ 10 Abs. 3 GeschGG). Die Gesetzesbegründung verweist auf die ähnliche Norm des § 97 Abs. 2 Urheberrechtsgesetz, wonach die Höhe nur der „Billigkeit“ entspricht, sich also in einem vernünftigen Rahmen bewegen muss und keine anderweitigen Beseitigungsalternativen vorhanden sein dürfen. Die Rechtsprechung orientiert sich bei der Entscheidung über die Höhe des immateriellen Schadens oft an sogenannten fiktiven Lizenzgebühren, die für die jeweilige Nutzung zu zahlen wären. Dies kann bei verloren gegangenen Know-how ein erheblicher Betrag sein.

Damit sind alle neuen Geheimhaltungsvereinbarungen an zwei Stellen zu ändern:

- Die Sorgfalt, mit der ein Geheimnis zu schützen ist, muss präziser als bislang definiert werden. Ein Hinweis auf das Grundschutzhandbuch des BSI und eine Zertifizierung nach ISO 27001 sollte ausreichend sein.
- Die Höhe des Schadensersatzes bei Verletzung einer Geheimhaltungsvereinbarung geregelt werden. Tut man dies nicht, gilt die gesetzliche Regelung. Danach muss ohne Nachweis eines tatsächlichen Schadens eine Pauschale gezahlt werden, deren Höhe beträchtlich sein kann. Leider kann damit bei jedem neuen NDA eine Verhandlung über eine Haftungsbegrenzung erforderlich sein.

2. Die Ökonomisierung von Daten ist möglich und eröffnet neue Geschäftsmodelle.

Daten sind das Öl des 21. Jahrhunderts. Künstliche Intelligenz benötigt viele Daten hoher Qualität, um leistungsfähig zu werden. Daten erlauben präzisere Vorhersagen, die einen Wettbewerbsvorteil bieten, also ökonomischen Wert haben. Ein Beispiel: Ein Betreiber von Windkraftanlagen kann nun mit dem Hersteller der Windanlage vereinbaren, dass die Wetterdaten, die von der Anlage erhoben werden, ein Geschäftsgeheimnis sind und nur der Anlagenbetreiber berechtigt ist, diese Daten zu verkaufen. Interessenten können Versicherungen sein, die das Risiko für Schadensversicherungen besser berechnen wollen, oder Energiehändler,

die den Energiebedarf in einer bestimmten Region besser vorhersagen wollen als der Wettbewerber.

- Achten Sie bei Ihrem bisherigen Geschäftsmodell darauf, ob Daten entstehen, die bisher ohne Schutz einem Dienstleister oder Zulieferer zufließen. Regeln Sie vertraglich, dass diese Daten ein Geschäftsgeheimnis Ihres Unternehmens darstellen und nur Ihnen zustehen.

3. Zuhören und Mitlesen ist verboten

Wer weiß oder zumindest erkennen könnte, dass sein Geschäftspartner Informationen in verbotener Weise erlangt hat, macht sich ebenfalls haftbar, wenn er sich die Geschäftsgeheimnisse ansieht, mitliest oder anhört. Wird in einer Präsentation erkennbar über Geschäftsgeheimnisse gesprochen, die der Präsentierende eigentlich nicht mitteilen dürfte, kann schon das Verbleiben im Raum zur zivilrechtlichen Haftung führen.

- Bei Geschäftsgeheimnissen fremder Unternehmen ist also Vorsicht geboten.



Dr. Jan Bohnstedt
Jan.Bohnstedt@bartsch.law

Mit Jenkins in Richtung DevOps

Moritz Reinwald, MT AG

Jenkins ist der weltweit am meisten genutzte Open-Source-Automatisierungsserver mit unzähligen Plugins für eine Vielzahl von Technologien und Einsatzszenarien rund um Continuous Integration und Continuous Delivery [1]. Dieser Artikel stellt verschiedene Möglichkeiten und Werkzeuge vor, mit denen der Jenkins-Server dazu genutzt werden kann, einen weiteren Schritt in Richtung DevOps zu gehen.

Jenkins Build Jobs

Ist ein Jenkins-Server eingerichtet und aktiv, können Aufgaben in sogenannten Jenkins Build Jobs definiert werden. Diese Jobs können einzeln über die Oberfläche konfiguriert und sowohl manuell als auch automatisiert, zum Beispiel zeitgesteuert oder bei Änderungen in der Versionsverwaltung, gestartet werden. Da die Jobs im Normalfall nach Aufgaben getrennt sind, pro Job also nur eine Aufgabe ausgeführt wird, werden pro Projekt mehrere Jobs benötigt. Über die Zeit kann so aus wenigen Jobs relativ schnell eine große und unübersichtliche Menge an Jobs heranwachsen, die sich nur schwer und mit viel Zeitaufwand pflegen lässt.

Soll nun eine Änderung in mehreren Jobs vorgenommen werden, wie beispielsweise das Versionierungssystem von Subversion auf Git umzustellen, muss jeder Jenkins-Job einzeln abgeändert werden. Da die Jobs von Jenkins intern als XML-Dateien gespeichert werden, ist es durchaus möglich, die Änderungen dort vorzunehmen, allerdings ist dies sehr kompliziert und oftmals fehlerbehaftet, da dort keinerlei Validierung der vorgenommenen Änderung stattfindet. Damit stellt die Änderung der XML-Dateien keinen sinnvollen Weg dar. Für die Bearbeitung von mehreren Jobs gibt es zudem Plug-ins aus der Jenkins-Community, die mehr oder weniger gut funktionieren. Das generelle Ziel sollte hier allerdings nicht sein, Werkzeuge zu finden, die einen bei der aufwendigen Änderungsmethode von Jenkins-Jobs unterstützen, sondern eine Methode zum Ausfindig machen, die es ermöglicht, Jenkins-Jobs einfach zu erstellen und später auch zu pflegen.

Ein weiterer Punkt, der gegen eine Vielzahl von Jenkins-Jobs spricht, ist die Dezentralisierung. Verteilt sich ein Prozess auf verschiedene Jobs, können diese zwar verkettet und somit automatisch hintereinander aufgerufen werden, allerdings kann dies auch Risiken mit sich bringen. Ein Nutzer kann eine Verkettung nicht direkt sehen, da er dazu in die Konfiguration der einzelnen Jobs schauen müsste. Führt dieser Nutzer zum falschen Zeitpunkt unwissentlich eine ganze Reihe von Jobs anstelle eines einzelnen Jobs aus, kann es schnell zu Problemen kommen. Zudem ist es schwierig, ein Reporting über mehrere Jobs zu realisieren, um den Fortschritt und die jeweiligen Ergebnisse aller Jobs auf einen Blick sehen zu können. Hier liegen die einzelnen Informationen alle verteilt in jedem einzelnen Jenkins-Job.

Jenkins stammt ursprünglich aus einer Zeit, in der Programme ausschließlich über die Benutzeroberfläche gesteuert wurden. Damals boten viele Programme

keinerlei Optionen, erstellte Neuerungen als Code zu speichern, um diesen dann zu versionieren. Auch die Jenkins Build Jobs verlassen sich daher zu sehr auf die Benutzeroberfläche, d.h. sie sind dazu designed, über die Oberfläche erstellt zu werden. Damit Jenkins Build Jobs versioniert werden können, müssen viele verschiedene XML-Dateien in das Versionierungssystem eingespielt werden. Zusätzlich zur Versionierung spielt hier auch die Dokumentation eine Rolle. Da die Jenkins Build Jobs über eine Oberfläche erstellt werden, lassen sich alle Änderungen und Konfigurationen nicht einfach dokumentieren, sondern es müssen alle in der UI vorgenommenen Änderungen detailgetreu aufgeschrieben oder Screenshots angefügt werden. Hier zeigt sich auch das Problem der Reproduzierbarkeit. Wird ein kleines Detail vergessen, kann es vorkommen, dass zwei fast identische Jenkins Build Jobs komplett unterschiedliche Ergebnisse erzielen.

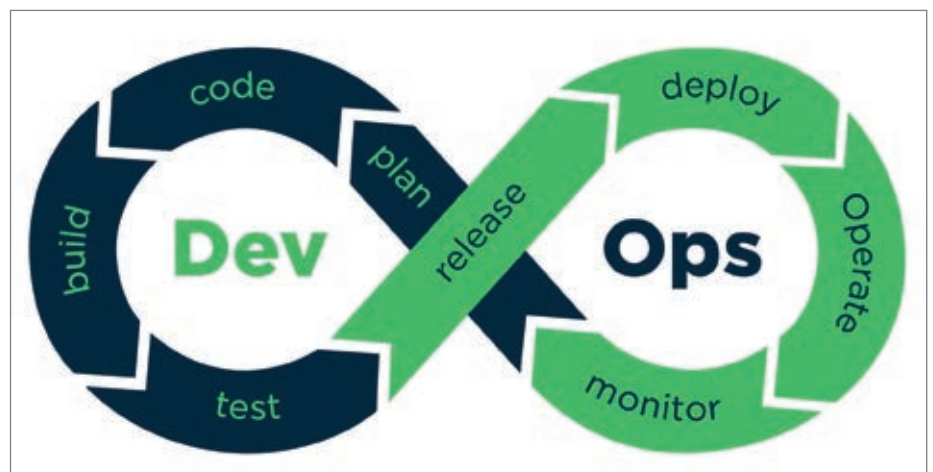


Abbildung 1: Der DevOps-Zyklus (Quelle siehe [2])

	Plan	Code	Build	Integrate	Test	Release	Deploy	Operate
Continuous Integration	Green	Green	Green	Green	Green	Red	Red	Red
Continuous Delivery	Green	Green	Green	Green	Green	Green	Red	Red
Continuous Deployment	Green	Green	Green	Green	Green	Green	Green	Red
DevOps	Green	Green	Green	Green	Green	Green	Green	Green

■ Automatisierte Ausführung
■ Manuelle Ausführung

Abbildung 2: Abdeckung des Software-Lebenszyklus durch Continuous Integration, Continuous Delivery, Continuous Deployment und DevOps (Quelle: Moritz Reinwald)

Zusammengefasst haben Jenkins Build Jobs mit umfangreicheren Projekten einige Probleme, sei es die reine Anzahl der einzelnen Jobs, die Unübersichtlichkeit und Dezentralisierung von Informationen, ungewollte Risiken oder die nicht triviale Versionierungsmöglichkeit. Eben diese Hürden gilt es zu überwinden, um weiter in Richtung DevOps zu gehen, und genau dafür gibt es seit Frühjahr 2016 eine offizielle Lösung mit Continuous Delivery Pipelines vom Jenkins-Entwicklerteam rund um Kohsuke Kawaguchi.

DevOps und CI/CD

Vielleicht gehen wir vorher aber nochmal einen Schritt zurück, um das große Ganze zu betrachten und zu verstehen. Wo soll es überhaupt hingehen? Was ist DevOps? Und wie helfen Continuous Delivery Pipelines dabei, mehr in Richtung DevOps zu kommen?

Ganz grob gesagt ist DevOps die Kombination aus Development und Operations. Wird versucht, DevOps genauer zu definieren, könnte es in etwa folgendermaßen klingen: „DevOps bezeichnet eine Reihe von Praktiken zur Automatisierung der Prozesse zwischen Softwareentwicklern und IT-Teams, durch die Software schneller und zuverlässiger entwickelt, getestet, freigegeben [3], betrieben und gewartet werden kann“. Eine genaue Definition ist allerdings schwierig, da DevOps kein konkretes Werkzeug beziehungsweise keine konkrete Methode, sondern eher eine Unternehmenskultur ist. Zusammengefasst kann gesagt werden: DevOps vereint alle notwendigen Schritte, sowohl die des Entwicklungszyklus:

- planen
- entwickeln (Code schreiben)
- bauen
- testen

Als auch die des Betriebszyklus:

- fertiges Release erstellen
- in der Kundenumgebung einspielen
- betreiben
- überwachen

Aus diesen beiden, ohne DevOps getrennten, Kreisläufen bildet sich somit ein einzelner Zyklus, der, solange das Pro-

dukt lebt, unendlich weiterläuft (siehe Abbildung 1).

Continuous Delivery ist, wie auch DevOps, eines der Schlagworte, die überall genannt werden, aber oftmals gar nicht richtig interpretiert werden können. Was ist zum Beispiel der Unterschied zwischen Continuous Integration, Continuous Delivery und Continuous Deployment? Und wie passt DevOps dazu? Die Unterschiede zeigen sich deutlich, wenn die einzelnen Schritte des Software-Lebenszyklus betrachtet werden (siehe Abbildung 2).

Nun ist auch ersichtlich, dass Continuous Delivery kein Synonym von Continuous Deployment ist, auch wenn beide häufig mit CD abgekürzt werden. Ein Release in eine Staging-Area, also Testumgebung, einzuspielen und danach manuell auszuliefern oder ein Release direkt in das Produktivsystem des Kunden einzuspielen, ist ein großer Unterschied.

Jenkins Pipelines

Nachdem nun definiert ist, wohin es gehen soll, müssen jetzt noch die technischen Mittel gefunden werden, um diesen Weg zu gehen. Seit 2016 gibt es vom offiziellen Jenkins-Entwicklerteam das

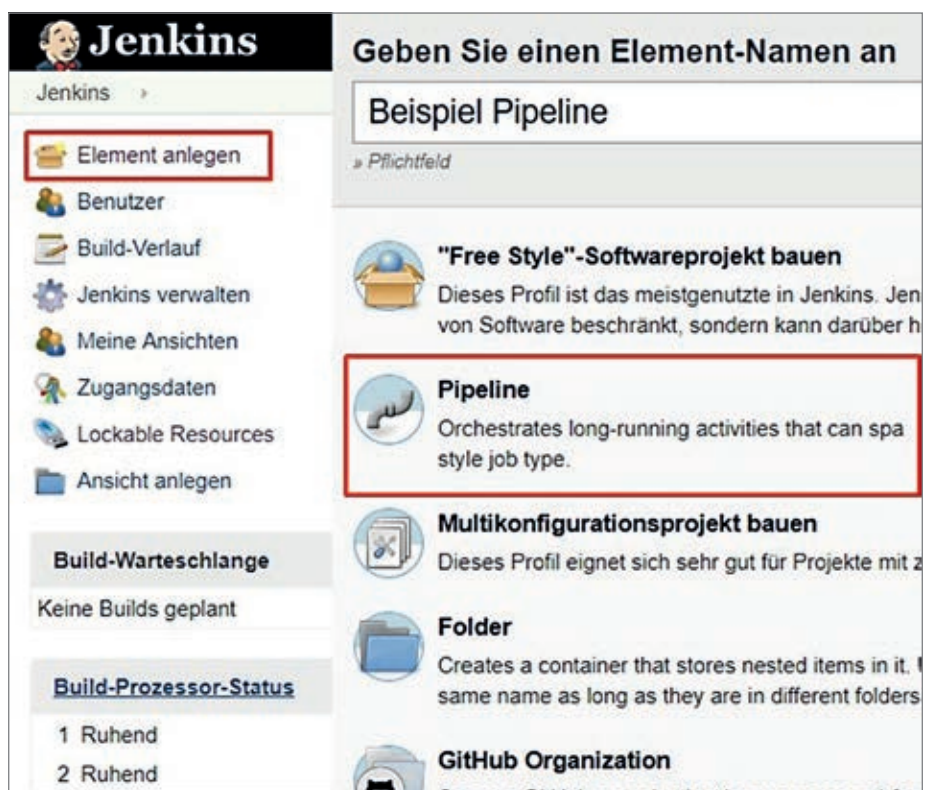


Abbildung 3: Erstellung einer neuen Jenkins Pipeline (Quelle: Moritz Reinwald)

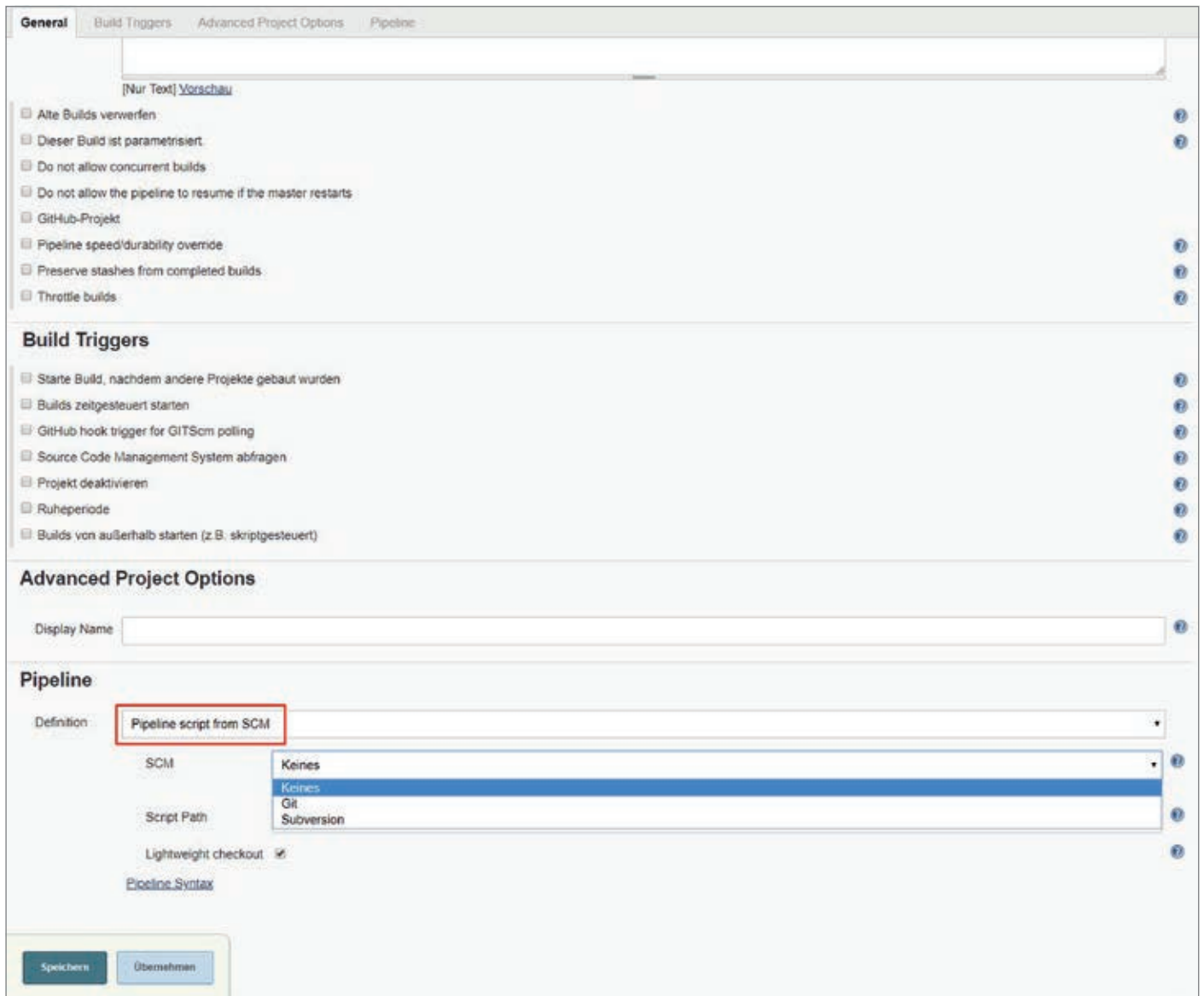


Abbildung 4: Beim Konfigurieren einer Jenkins Pipeline die Option „Pipeline script from SCM“ wählen (Quelle: Moritz Reinwald)

Pipeline-Plug-in [4] oder besser die Pipeline Plug-in Suite, da das Pipeline-Feature auf mehrere Plug-ins aufgeteilt ist. Diese Plug-ins erweitern den Jenkins-Server um die Funktionalität, Pipelines auszuführen. Pipelines sind Automatisierungsketten, die im Falle einer Continuous Delivery Pipeline beispielsweise alle Schritte zum Release einer Software abarbeiten. Nach einer einfachen Installation der Plug-ins auf dem Jenkins-Server kann bereits mit den Pipelines gearbeitet werden, in den meisten Fällen sogar ohne einen Neustart.

Im Gegensatz zu den Build Jobs wird bei einem Pipeline-Job die gesamte Logik in einem Skript, dem sogenannten Jenkinsfile, hinterlegt. Dieser Jenkinsfile sollte auch in das Versionierungssystem eingepflegt werden. Aktuell gibt es zwei Versionen von

Jenkins Pipelines. Die erste, initiale Variante heißt Scripted Pipeline. Hierbei wird der Jenkinsfile, auf dem die Pipeline aufbaut, noch größtenteils in Groovy, einer auf der Java-Plattform basierenden Programmier- und Skriptsprache, geschrieben. Zusätzlich gibt es die zweite und neuere Möglichkeit, die Pipelines deklarativ zu erstellen. Die sogenannten Declarative Pipelines basieren auf einer DSL, Domain-Specific-Language, die auf Groovy aufbaut. Wird eine Pipeline geschrieben, sollte der deklarative Ansatz gewählt werden, da die Scripted Pipeline in die deklarative Variante miteingebunden werden kann und somit obsolet ist.

Da der Jenkins-Server die Pipelines als weiteren Job-Typ definiert, können Pipelines auf die gleiche Weise wie die normalen Build Jobs angelegt werden. Nach einem Klick auf die Schaltfläche „Element anlegen“

kann nun als Job-Typ „Pipeline Job“ ausgewählt werden (siehe Abbildung 3). Die Oberfläche zur Konfiguration eines Pipeline-Jobs sieht hierbei ähnlich aus wie die der Build Jobs. Hier sollte allerdings nicht der Fehler gemacht werden, dass zu viel in der Oberfläche konfiguriert wird. Diese Konfigurationsoptionen stehen ebenfalls im Jenkinsfile zur Verfügung. Wenn mehrere Pipelines auf einem Jenkinsfile aufbauen, muss hier nicht jeder Pipeline-Job einzeln konfiguriert werden, sondern jeder Job zieht sich die notwendige Konfiguration aus dem Jenkinsfile. Einzig die Verbindung zum bevorzugten Versionierungssystem muss in jedem Pipeline-Job in der Oberfläche gesetzt werden.

Da ein Jenkinsfile genutzt wird, findet sich im Pipeline-Job selbst keinerlei Konfiguration wieder. Theoretisch besteht auch die Möglichkeit, die Konfiguration direkt


```

Pipeline {
  agent {
    label: 'linux'
  }
  ...
  stages {
    ...
  }
}

```

Listing 1: Einen bestimmten Agenten über ein Label selektieren

in der Oberfläche des Pipeline-Jobs zu erstellen und die Pipeline nicht in einen Jenkinsfile zu schreiben. Dies widerspricht allerdings der Idee der Jenkins Pipelines, da diese Pipeline-Jobs dann genau wie normale Build Jobs konfiguriert sind und alle Vorteile verloren gehen. Daher sollte ganz unten in der Konfiguration unbedingt die Option, den Jenkinsfile aus der Versionierung zu laden, gewählt werden (siehe Abbildung 4). Hier stehen alle unterstützten Versionierungssysteme von Jenkins zur Verfügung. Neben Git lassen sich weitere Systeme wie etwa Subversion und BitBucket über Plug-ins installieren.

Jenkinsfile

Der Jenkinsfile einer Declarative Pipeline besteht hauptsächlich aus Sektionen und

Direktiven. Der Anfang jeder Pipeline ist der pipeline-Block. Alles, was in der Pipeline passiert, muss innerhalb dieses Blockes stehen. Sind selbst geschriebene Groovy-Funktionen vorhanden, können diese auch außerhalb des Blockes definiert werden. Sinnvoller ist es in diesem Fall aber, eine sogenannte Shared Library [5] zu nutzen. Dies ist ein Feature, über das die Standard-DSL-Sprache um eigene Funktionen erweitert werden kann. Die grobe Struktur innerhalb des pipeline-Blockes ist durch sogenannte stages definiert. In jeder stage können verschiedene Befehle innerhalb einer steps-Sektion ausgeführt werden. Zudem kann in jeder stage eine post-Sektion eingefügt werden. In dieser können verschiedene Befehle aufgrund des Ergebnisses der steps-Sektion ausgeführt werden. Diese post-Sektion kann ebenfalls am Ende der gesamten stages-Sektion definiert werden.

Als Erstes innerhalb des pipeline-Blockes muss mithilfe der agent-Sektion ein Exekutor, auf dem die Pipeline oder einzelne Stages ausgeführt werden, gewählt werden. Hier bietet Jenkins verschiedene Möglichkeiten. Mit agent any bestimmt Jenkins den Exekutor, mit agent none wird global kein agent festgelegt, dieser muss dann innerhalb der jeweiligen stages definiert werden. Soll ein bestimmter agent gewählt werden, kann dies über ein label geschehen (siehe Listing 1). Weitere Möglichkeiten sind in der Jen-

kins-Pipeline-Dokumentation beschrieben [6].

Nachdem der Exekutor definiert wurde, können nun durch weitere Direktiven, noch vor den Stages, einige Optionen, Parameter, Trigger und Umgebungsvariablen gesetzt werden. Nachfolgend sind alle häufig genutzten Sektionen und Direktiven aufgelistet, ausführlichere Informationen zu diesen und weiteren Befehlen finden sich ebenfalls in der Dokumentation der Jenkins Pipelines [6].

Um die Pipeline nun mit Funktionalität zu versehen, müssen die gewünschten Befehle in die steps-Sektion geschrieben werden. Hierfür empfiehlt es sich, den durch die Pipeline-Plug-ins in den Jenkins integrierten Pipeline Snippet Generator zu nutzen. Dieser findet sich innerhalb eines Pipeline-Projektes im Menü auf der linken Bildschirmseite. Hier sind alle verfügbaren Befehle für eine Pipeline enthalten und können mithilfe der Oberfläche konfiguriert werden (siehe Abbildung 5). Als Ergebnis liefert der Generator ein Snippet, das nahtlos in die Pipeline integriert werden kann. Für Plug-ins, die Jenkins Pipelines unterstützen, sind die Befehle ebenfalls im Snippet Generator verfügbar. Zusätzlich ist auch ein Direktiven-Generator für Declarative Pipelines und viele Referenzen, beispielsweise für einzelne Steps oder globale Variablen, vorhanden.

Name	Benötigt	Verwendbar in	Beschreibung
pipeline	Ja	(1x)	Der Hauptblock
agent	Ja	pipeline, stage	Exekutor für Pipeline oder jede Stage
stages	Ja	pipeline (1x)	Hauptsektion für einzelne Stufen
stage	Ja mind. 1x	stages	Direktive für eine Stufe (Name muss angegeben werden)
steps	Ja	stage	Sektion für die einzelnen Befehle
post	Nein	pipeline, stage	Aktionen, je nach Ergebnis der Stage/Pipeline ausführen
environment	Nein	pipeline, stage	Definiert Umgebungsvariablen
options	Nein	pipeline (1x), stage (eingeschränkt)	Definiert Build-Optionen für die Pipeline (für eine Stage lassen sich nur bestimmte Optionen aktivieren)
parameters	Nein	pipeline (1x)	Definiert Parameter, die der User mitgeben soll
triggers	Nein	pipeline (1x)	Definiert automatisierte Starts der Pipeline
parallel	Nein	stage	Führt mehrere Stages innerhalb einer Stage parallel aus
script	Nein	steps	Führt Code einer Scripted Pipeline oder Groovy Code aus
when	Nein	stage	Bedingte Ausführung der Stage
catchError	Nein	steps	Fängt einen Fehler ab (ähnlich zu try/catch)

Tabelle 1: Auflistung einiger Sektionen und Direktiven

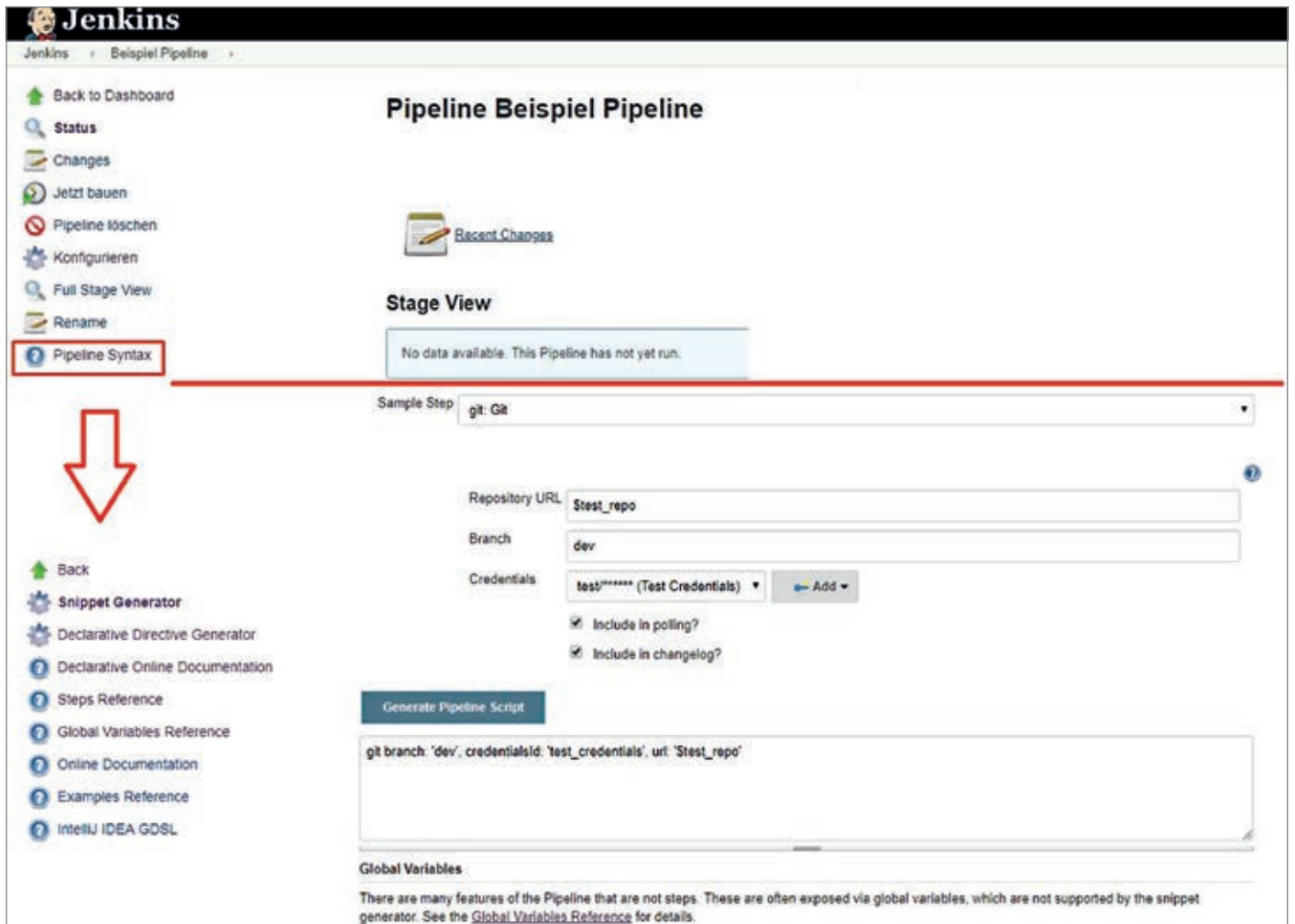


Abbildung 5: In Jenkins integrierter Snippet & Directive Generator (Quelle: Moritz Reinwald)

Zudem gibt es die Möglichkeit, den Jenkinsfile validieren zu lassen, bevor die Pipeline ausgeführt wird, indem der Jenkins Pipeline Linter genutzt wird. Dieser wird auf dem Jenkins-Server ausgeführt und der Jenkinsfile per HTTP-Post-Request übertragen. Je nach Konfiguration müssen hier noch Vorkehrungen getroffen werden, damit der Jenkins-Server die Anfrage erfolgreich erhält. Mehr dazu befindet sich in der Dokumentation [7].

Besonderheiten und Best Practices

Müssen innerhalb der Pipeline, also dem Jenkinsfile, Passwörter genutzt werden, sollte unbedingt das Jenkins Credentials Plug-in [8] genutzt werden, das normalerweise standardmäßig mit dem Jenkins installiert wird. Wie bei den Build Jobs ermöglicht das Plug-in auch in den Pipelines die Nutzung von IDs, um im Jenkins hinterlegte Benutzerdaten zur Laufzeit ab-

zurufen (siehe Listing 2). Somit stehen keine Benutzerdaten im Klartext in Skripten oder im Jenkinsfile.

Sollen Informationen des aktuellen Builds des Pipeline-Jobs abgerufen werden, kann dies mithilfe der Eigenschaft currentBuild bewerkstelligt werden. Diese enthält Informationen über die ID bzw. Nummer,

den Display-Namen, die Beschreibung und das aktuelle Ergebnis, auch zur Laufzeit. Damit ist es möglich, das Ergebnis eines Build-Schrittes oder auch der gesamten Pipeline zu prüfen und manuell das Ergebnis zu setzen, sollte der Jenkins-Server dies einmal nicht automatisch richtig erkennen. Ist das Ergebnis allerdings fehlerhaft (FAI-

```
withCredentials([usernamePassword(
  credentialsId: 'test_credentials',
  passwordVariable: 'test_password',
  usernameVariable: 'test_user')]) {
  bat '''cd Projektordner
  connect_db %apex-connection% %test_user% %test_password%
  @scripts\myScript.sql'''
}
```

Listing 2: Aufruf des Credentials-Plug-ins aus dem Jenkinsfile

```
if (result == "UNSTABLE") {
  currentBuild.result = "SUCCESS"
}
```

Listing 3: Abrufen und Ändern des Build-Ergebnisses zur Laufzeit

```

pipeline {
  agent {
    label {
      label ""
      customWorkspace "path/to/workspace"
    }
  }
}

```

Listing 4: Setzen eines Custom Workspace innerhalb des Jenkinsfiles

LURE), lässt es sich nur noch auf „instabil“ (UNSTABLE) und nicht mehr auf „erfolgreich“ (SUCCESS) ändern (siehe Listing 3). Bricht ein Benutzer die Pipelineausführung ab, gibt es noch den Status ABORTED. Hier sollte also möglichst vor dem Setzen des Ergebnisses geprüft werden.

Soll ein Pipeline-Job nicht im Standard Jenkins Workspace laufen, kann ein sogenannter Custom Workspace angegeben werden (siehe Listing 4). Hier sollte darauf geachtet werden, dass der Jenkinsfile zuerst aus der Versionierung ausgecheckt und dann erst ausgeführt wird. Da der Custom Workspace

allerdings erst im Jenkinsfile definiert wird, wird der Jenkinsfile selbst immer in den Standard Workspace ausgecheckt. Hier sollte wirklich nur der Jenkinsfile und nicht etwa das ganze Repository ausgecheckt werden, um Duplikate zu verhindern und nicht unnötig Speicherplatz zu verschwenden.

Funktioniert eine Pipeline nicht wie gewünscht, gibt es in der Oberfläche des Pipeline-Jobs eine Replay-Funktion. Mit dieser ist es möglich, den Jenkinsfile temporär in der Oberfläche zu ändern, um dann einen erneuten Durchlauf zu starten (siehe Abbildung 6). Der Jenkinsfile im Hin-

tergrund bleibt dabei unberührt. Somit ist es einfacher, einen Fehler innerhalb des Jenkinsfiles zu finden, ohne diesen jedes Mal committen zu müssen.

Oberfläche

Wurde ein Pipeline-Job angelegt und mindestens einmal ausgeführt, erscheint in der Oberfläche eine Übersicht über die Ausführungsdauer sowie das Ergebnis der einzelnen Stages. Zudem lassen sich die Logs der jeweiligen Stages mit einem Klick auf die Stage abrufen. Ist die Pipeline erfolgreich durchgelaufen, erscheinen alle Stages grün. Gab es einen Fehler in einer Stage, sind diese und alle nachfolgenden, nicht ausgeführten Stages dunkelrot eingefärbt. In diesem Fall färben sich auch die zuvor erfolgreich durchgelaufenen Stages hellrot. Oben in der Übersicht stehen die zuvor vergebenen Namen der Stages und eine Anzeige dazu, wie lange die Ausführungsdauer dieser Stage durchschnittlich war. Auf der linken Seite werden die verschiedenen Builds, also Ausführungen, mit der jeweiligen Build-Nummer, dem Start-Datum und der Anzahl an Commits in die Versionierung seit der letzten Ausführung aufgelistet (siehe Abbildung 7).

Neben der Standardoberfläche gibt es per Plug-in [9] auch noch die extra komplett neu entwickelte Oberfläche namens Blue Ocean. Sie soll die User Experience von Jenkins deutlich steigern, indem sie folgende Hauptmerkmale mit sich bringt [10]:

- Unterstützung von anspruchsvollen Visualisierungen von Continuous Delivery Pipelines
- Intuitives Erstellen von Pipelines in einem visuellen Prozess durch den Pipeline Editor
- Anpassungen an die rollenbasierten Anforderungen jedes Teammitglieds
- Präzision bei der Lokalisierung von Problemen innerhalb einer Pipeline
- Native Integration von Branches und Pull Requests für GitHub und BitBucket

Wird Blue Ocean genutzt, kann auf einen Blick gesehen werden, wo die Pipeline fehlerhaft war und welche Stages eventuell trotzdem danach noch ausgeführt wurden. Zudem sieht die Oberfläche deutlich „frischer“ aus und ist von Grund auf für Pipelines entwickelt worden (siehe Abbildung 8).

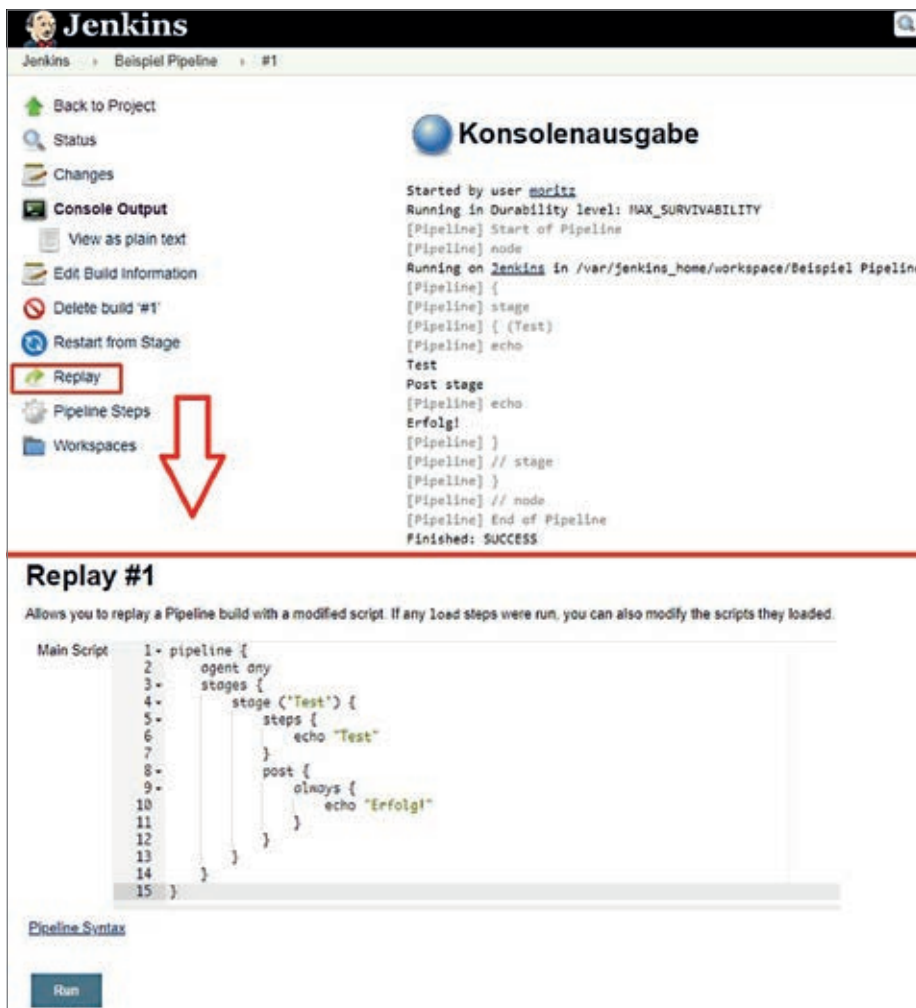


Abbildung 6: Replay-Funktion in der Oberfläche eines Builds (Quelle: Moritz Reinwald)

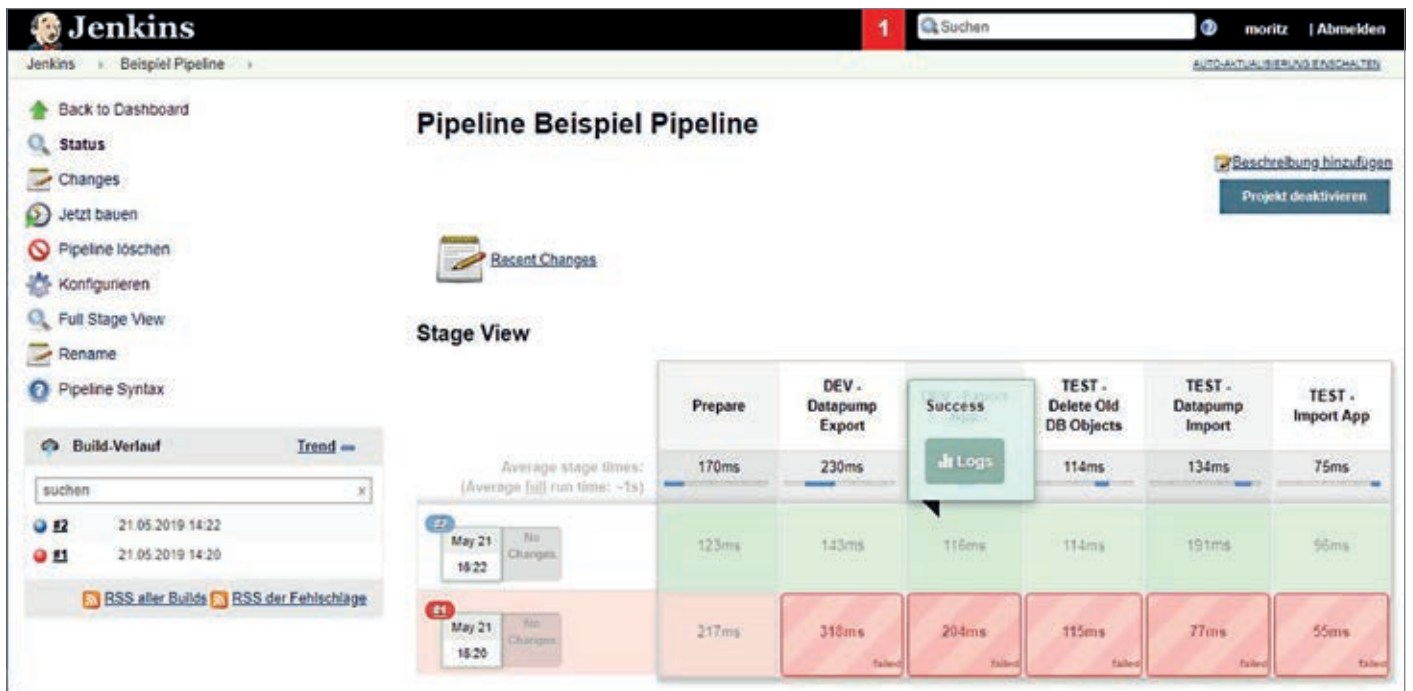


Abbildung 7: Standardoberfläche eines Jenkins-Pipeline-Jobs (Quelle: Moritz Reinwald)

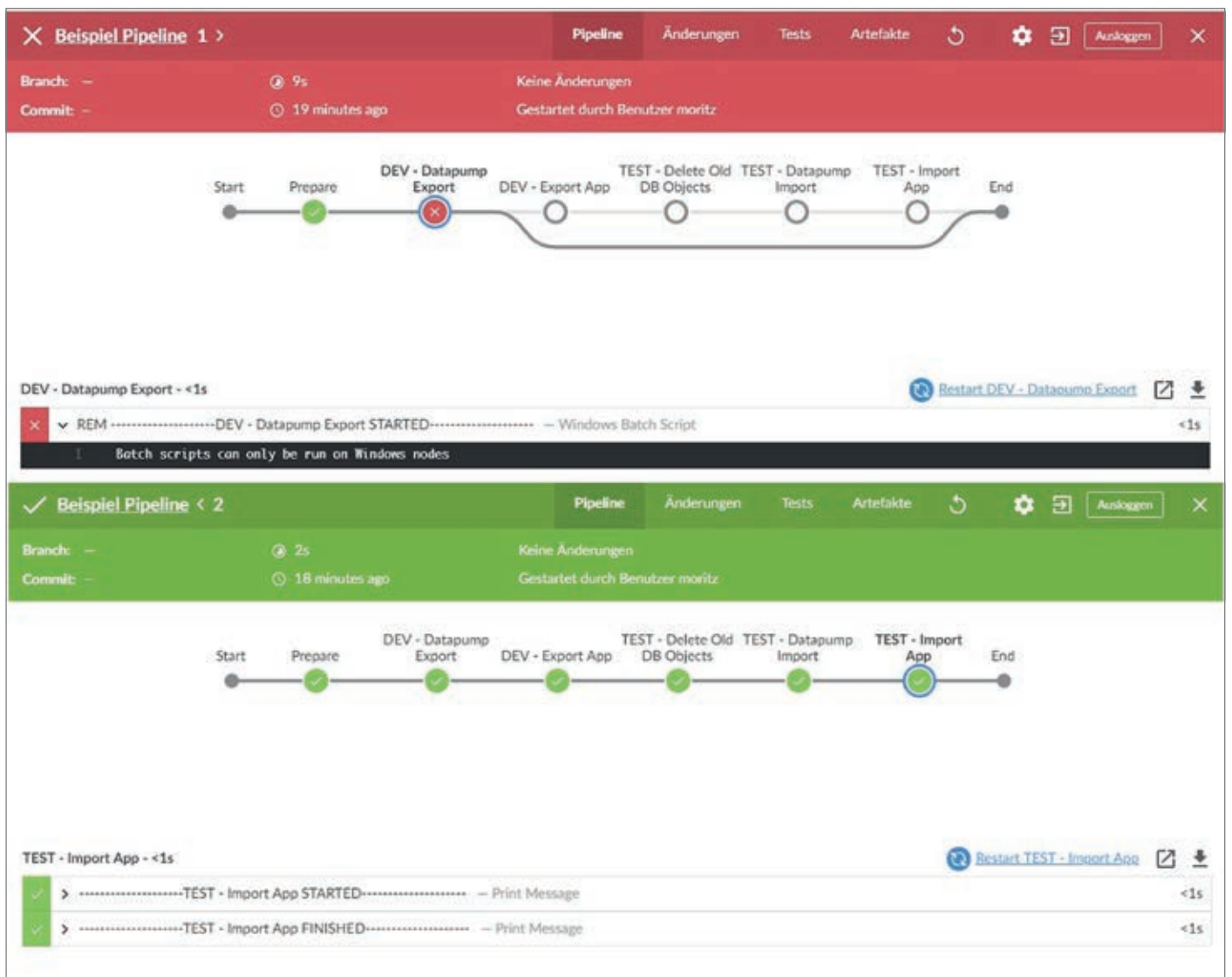


Abbildung 8: Blue-Ocean-Oberfläche für einen Fehlschlag und Erfolg des gleichen Jobs (Quelle: Moritz Reinwald)

Fazit

Jenkins bietet mit den Pipeline Plug-ins und der Blue-Ocean-Oberfläche eine gute Basis für Continuous Delivery Pipelines. Nach einer kurzen Eingewöhnung an den Aufbau des Jenkinsfiles geht die Erstellung von Pipelines, durch den deklarativen Ansatz, einfach von der Hand. Sollte die DSL einmal nicht ausreichen, existiert noch immer die Möglichkeit, eigene Groovy-Funktionen zu schreiben und einzubauen. Mit Blue Ocean bietet Jenkins auch an der Oberfläche viele Funktionen und eine gute Übersicht, somit lassen sich Pipelines gut verwalten und Probleme leichter beheben.

Quellen und weitere Informationen

- [1] Jenkins Homepage:
<https://jenkins.io>
- [2] DevOps-Zyklus:

- <https://medium.com/@neonrocket/devops-is-a-culture-not-a-role-be1bed149b0>
- [3] Atlassian Definition DevOps:
<https://de.atlassian.com/devops>
- [4] Jenkins Pipeline Plug-in:
<https://wiki.jenkins.io/display/JENKINS/Pipeline+Plugin>
- [5] Jenkins Pipeline Shared Libraries:
<https://jenkins.io/doc/book/pipeline/shared-libraries>
- [6] Jenkins Pipeline Syntax:
<https://jenkins.io/doc/book/pipeline/syntax>
- [7] Jenkins Pipeline Linter:
<https://jenkins.io/doc/book/pipeline/development/#linter>
- [8] Jenkins Credentials Plug-in:
<https://plugins.jenkins.io/credentials>
- [9] Jenkins Blue Ocean Plug-in:
<https://plugins.jenkins.io/blueocean>
- [10] Blue-Ocean-Dokumentation:
<https://jenkins.io/doc/book/blueocean>

Über den Autor

Moritz Reinwald arbeitet als Berater mit dem Schwerpunkt APEX Development und DevOps bei der MT AG in Ratingen. Neben der Erstellung von Webanwen-

dungen mithilfe von APEX liegen seine Schwerpunkte im Bereich CI/CD mit Jenkins und Testautomatisierung. Zusätzlich beschäftigt er sich mit Technologien rund um das Thema DevOps, insbesondere in Verbindung mit Datenbanken.



Moritz Reinwald
moritz.reinwald@mt-ag.com

Das E-3 Magazin heißt Sie herzlich WILLKOMMEN zum DSAG Jahreskongress



www.e-3.de

Information und Bildungsarbeit von und für die SAP®-Community



THEOBALD
SOFTWARE



DATAVARD





REST-Schnittstellen und Application Express

Carsten Czarski, ORACLE Deutschland

Wenn es um die Kopplung unterschiedlicher Systeme in einer IT-Landschaft geht, kommt heute vor allem REST zum Einsatz. Diese sehr einfachen, auf dem HTTP-Protokoll basierenden Schnittstellen machen es sehr leicht, ein System anzubinden: Es braucht lediglich einen HTTP-Client und (normalerweise) einen JSON-Parser – das ist nahezu in allen Umgebungen vorhanden.

Auch in APEX-Anwendungen lassen sich REST-Schnittstellen sehr leicht nutzen. Das kann mit PL/SQL-Code und dem Paket `APEX_WEB_SERVICE` oder, komplett deklarativ, mit den ab APEX 18.1 vorhandenen Web Source Modules geschehen.

Sobald eine REST-Schnittstelle vorhanden ist, spielt es keine Rolle mehr, was das angebundene System selbst tut. Egal ob es eine Blockchain oder ein ERP-System ist, ob es auf einer relationalen oder NoSQL-Datenbank basiert: Die Anbindung an eine APEX-Anwendung ist stets gleich.

REST Services

REST steht für Representational State Transfer und ist zunächst vor allem ein Architekturmodell für Anwendungen. Die Kommunikation zwischen verschiedenen Systemen wird in der REST-Architektur standardisiert.

- Ressourcen stellen Dienste bereit und sind eindeutig identifizierbar
- Methoden dienen dem Abrufen eines konkreten Dienstes

- Repräsentationen bestimmen, wie die Antworten zu interpretieren sind.

Abbildung 1 zeigt die Anwendung dieses „REST-Dreiecks“ auf HTTP-Webservices. Ressourcen werden mit URLs identifiziert; als Methoden kommen die des HTTP-Protokolls (GET, PUT, POST, DELETE und andere) zum Einsatz und anhand von Mime-Types können verschiedene Repräsentationen von Daten oder Dokumenten erkannt werden. Besonderes Merkmal ist, dass die Kommunikation in einer

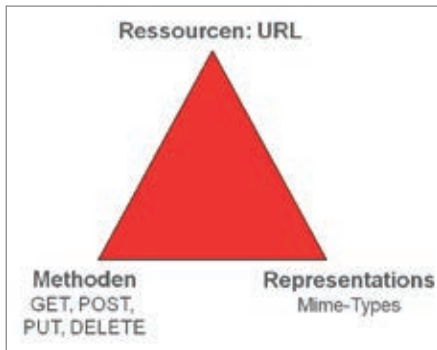


Abbildung 1: „REST-Dreieck“ (Quelle: Carsten Czarski)

REST-Architektur immer zustandlos ist – es gibt also niemals eine „Sitzung“ mit einem bestimmten Status.

Genau wegen dieser Tatsache bietet die REST-Architektur für Web Services einen enormen Vorteil: Da kein Session-Status verwaltet werden muss, lassen sich Ressourcen auf dem Server sehr effizient nutzen. Datenbankverbindungen werden beispielsweise nur für den kurzen Zeitpunkt benötigt, in dem eine Anfrage beantwortet wird. Unmittelbar danach kann sie in den Ressourcen-Pool zurückkehren und für andere Clients genutzt werden. REST Webservices können so typischerweise sehr gut skalieren. Allerdings ist REST nicht auf Webservices beschränkt – das grundsätzliche Architekturmodell kann auch in anderen Umgebungen genutzt werden.

REST Services 1: APEX_WEB_SERVICE

Das PL/SQL-Paket APEX_WEB_SERVICE wurde mit APEX 3 eingeführt und erlaubt, mit dem Unterprogramm MAKE_REST_REQUEST, das manuelle Programmieren eines Aufrufs einer REST-Schnittstelle. Möchte man beispielsweise das GitHub REST API (api.github.com) aus APEX heraus nutzen, um alle Repositories eines bestimmten GitHub-Users („oracle“) abzurufen, so funktioniert das mit APEX_WEB_SERVICE wie folgt: Zunächst werden HTTP-Request-Headers gesetzt (siehe Listing 1), dann wird mit MAKE_REST_REQUEST ein GET-Request auf das GitHub REST API gemacht (siehe Listing 2).

Das GitHub REST API liefert (wie die meisten REST-Schnittstellen) ein JSON-Dokument mit den Ergebnisdaten zurück.

In einer APEX- oder PL/SQL-Anwendung muss dieses im nächsten Schritt geparkt werden: Hierzu können die ab Oracle 12.1 vorhandenen SQL-/JSON-Funktionen sehr schön eingesetzt werden (siehe Listing 3).

POST, PUT, DELETE oder andere HTTP-Methoden lassen sich mit dem Paket

APEX_WEB_SERVICE analog nutzen. Für POST oder PUT Requests muss meist ein Request Body übergeben werden; hierfür bietet MAKE_REST_REQUEST die Parameter P_BODY und P_BODY_BLOB an.

APEX_WEB_SERVICE erlaubt es also, flexibel auf alle möglichen REST-Schnittstellen zuzugreifen – der durchzuführen

```
begin
  apex_web_service.g_request_headers( 1 ).name := 'User-Agent';
  apex_web_service.g_request_headers( 1 ).value := 'APEX';
end;
/
```

Listing 1: HTTP-Request-Header setzen ...

```
select apex_web_service.make_rest_request(
  p_url      => 'https://api.github.com/users/oracle/repos',
  p_http_method => 'GET' )
from dual
/

[{"id":33195015,"node_id":"MDEwOlJlcG9zaXRvcnkzMzE5NTAxNQ==", "name":"accelerators", "full_name":"oracle/accelerators", "private":false, "owner":{"login":"oracle", "id":4430336, "node_id":"MDEyOk9yZ2FuaXphdGlvbjQ0MzAzMzY=", "avatar_url":"https://avatars1.githubusercontent.com/u/4430336?v=4", "gravatar_i
```

Listing 2: GET Request auf das GitHub API durchführen.

```
with github as (
  select apex_web_service.make_rest_request(
    p_url      => 'https://api.github.com/users/oracle/repos',
    p_http_method => 'GET' ) as json
  from dual )
select name,
       language,
       forks_cnt,
       owner_id,
       owner_name
  from github, json_table(
    json,
    '$[*]'
    columns(
      name          varchar2(50) path '$.name',
      language      varchar2(50) path '$.language',
      forks_cnt     number       path '$.forks_count',
      owner_id     varchar2(15)  path '$.owner.id',
      owner_name   varchar2(30) path '$.owner.login' ) )
/
```

NAME	LANGUAGE	FORKS_CNT	OWNER_ID	OWNER_NA
accelerators	C#	54	4430336	oracle
accs-caching-java-sdk	Java	6	4430336	oracle
adf-samples	Java	18	4430336	oracle
analytical-sql-examples	PLSQL	66	4430336	oracle
apiplatform-mgmtscripts-sample	Python	2	4430336	oracle

Listing 3: JSON-Parsing

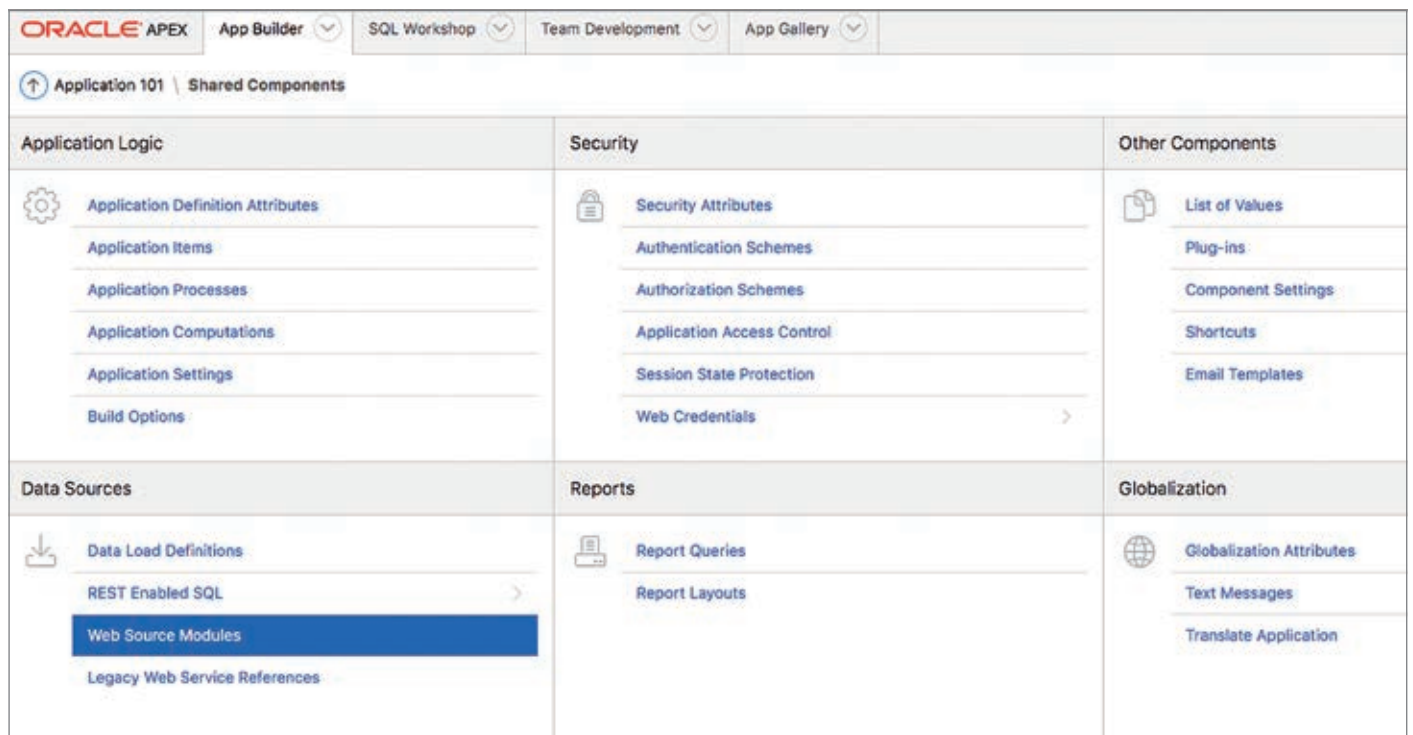


Abbildung 2: Shared Components einer APEX-Anwendung (Quelle: Carsten Czarski)

rende HTTP-Request kann frei programmiert, die Antwort der REST-Schnittstelle danach frei verarbeitet werden. So wäre nun ein PL/SQL-Paket GITHUB_REST_PKG denkbar, in dem alle Zugriffe auf GitHub gekapselt werden – so weit, so gut.

Allerdings ist auch festzustellen, dass recht viel „technischer“ und sich ständig wiederholender Code geschrieben werden muss. So funktioniert das JSON-Parsing stets gleich: Es findet ein SELECT statt und mit der JSON_TABLE-Funktion wird das JSON in eine Tabelle umgewandelt. Nur die angesprochenen Attribute und Spaltennamen ändern sich mit jeder REST-Schnittstelle.

Hätte man die Informationen zur URL, zu den HTTP-Headers und zur JSON-Struktur als Metadaten vorliegen, so könnten diese Aufrufe komplett automatisiert werden. Und an dieser Stelle wird der APEX-Entwickler hellhörig: Könnte APEX die Details oder Metadaten zu einer REST-Schnittstelle nicht speichern und dem Entwickler diese Aufgaben abnehmen?

REST Services 2: Web Source Modules

Mit dem im April 2018 erschienenen APEX Release 18.1 wurde die Unterstützung für externe REST Services auf eine völlig neue

Grundlage gestellt. Die neu eingeführten Web Source Modules erlauben es, Metadaten zu einer externen REST-Schnittstelle

zu speichern. Ein solches Web Source Module kann in der APEX-Anwendung dann wie eine Tabelle verwendet werden:

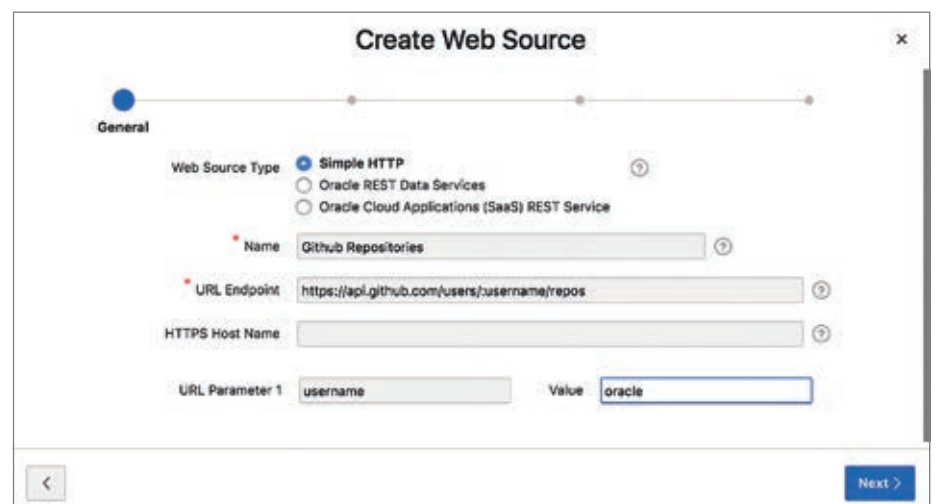


Abbildung 3: Ein neues Web Source Module erstellen (Quelle: Carsten Czarski)



Abbildung 4: Authentifizierung für ein Web Source Module (Quelle: Carsten Czarski)

Web Source Discovery

Preview

Data					Data Profile			
Id ↑	Url	Fork	Name	Size_	Forks	Id2	Url2	Type
102033881	https://api.github.com/repos/oracle/bosh-oracle-cpi	false	bosh-oracle-cpi	2045	5	4430336	https://api.github.com/users/oracle	Organization
102035088	https://api.github.com/repos/oracle/bosh-oracle-cpi-release	false	bosh-oracle-cpi-release	2670	5	4430336	https://api.github.com/users/oracle	Organization
102132349	https://api.github.com/repos/oracle/adf-samples	false	adf-samples	52283	18	4430336	https://api.github.com/users/oracle	Organization
110888512	https://api.github.com/repos/oracle/bosh-agent	true	bosh-agent	38241	1	4430336	https://api.github.com/users/oracle	Organization
110888537	https://api.github.com/repos/oracle/bosh-linux-stemcell-builder	true	bosh-linux-stemcell-builder	20733	2	4430336	https://api.github.com/users/oracle	Organization
112513560	https://api.github.com/repos/oracle/cluster-manager	false	cluster-manager	53339	13	4430336	https://api.github.com/users/oracle	Organization

More Detail Create Web Source

Abbildung 5: Die Antwort der REST-Schnittstelle kann als Tabelle aufgefasst werden. (Quelle: Carsten Czarski)

Der Entwickler fügt seiner Anwendung eine neue Komponente hinzu und wählt das Web Source Module als Datenquelle aus. In den Shared Components ist daher ab Version 18.1 der Bereich Web Source Modules enthalten (siehe Abbildung 2).

Es gilt nun also, ein Web Source Module für das GitHub REST API bereitzustellen. Der Wizard zum Erstellen eines neuen Web Source Module beginnt mit Angabe der Schnittstellen-URL. Mit der Doppelpunkt-

Syntax können dynamische Teile der URL schon jetzt gekennzeichnet werden: Das „oracle“ in der URL steht für den GitHub-User „oracle“. Daher wird es durch :username ersetzt – und diese Variable weiter unten mit dem Default von „oracle“ belegt. APEX erfährt so Details zum Aufbau der URL für das GitHub REST API (siehe Abbildung 3).

APEX erlaubt auch das Hinterlegen von Details zur Authentifizierung. Passwörter oder Client Secrets werden dabei

verschlüsselt gespeichert und sind nur für die APEX-Engine zu sehen. APEX unterstützt Basic Authentication und den OAuth2 „Client Credentials Flow“ aus dem Stand. Das GitHub REST API ist jedoch öffentlich und erwartet keine Authentifizierung (siehe Abbildung 4).

Das Erstellen eines Web Source Module wird normalerweise mit der Schaltfläche Discover abgeschlossen (mit der Schaltfläche Advanced können noch wei-

Web Source Module

Delete Apply Changes

Show All | Web Source Module | Subscription | Data Profile | Operations | Authentication | Module Parameters | Advanced

Web Source Module

Name:

Web Source Type:

Remote Server:

Base URL:

URL Path Prefix:

Subscription

This is the "master" copy of this web source module.

No web source modules subscribe to this module.

Data Profile Edit Data Profile

JSON	Table	95	95	0
Response Format	Returns	Columns	Visible	Derived

Abbildung 6: Das Web Source Module speichert Metadaten zur REST-Schnittstelle (Quelle: Carsten Czarski)

tere Angaben zu Parametern, HTTP-Headern oder zum JSON- bzw. XML-Parsing gemacht werden).

Nun ruft APEX das REST API zum ersten Mal auf. Die JSON- oder XML-Antwort wird untersucht, um Aufschlüsse über die Struktur zu erhalten. APEX versucht dann, das JSON als Tabelle aufzufassen, und zeigt die Daten bei Erfolg als Vorschau an (siehe Abbildung 5).

Wenn die Vorschau den Erwartungen entspricht, sollte das Web Source Module gespeichert werden. Danach liegen die

Metadaten für diese REST-Schnittstelle in der APEX-Anwendung vor (siehe Abbildungen 6, 7 und 8).

Der zu Beginn deklarierte dynamische Teil der URL ist als Web Source Parameter berücksichtigt: Es gibt die URL-Pattern-Variable :username mit dem Default von oracle.

Das Data Profile enthält alle Details zur JSON-Struktur – mit diesen Angaben kann APEX die JSON_TABLE-Funktion vollautomatisch einsetzen und die Daten sowie eine Tabelle bereitstellen.

Wenn das Web Source Module gespeichert ist, kann es für die Erstellung neuer APEX-Komponenten genutzt werden. Abbildung 9 zeigt, wie eine neue APEX-Seite mit einem Bericht (Report) auf Basis des Web Source Module erzeugt wird.

Schaut man sich die neue Seite im Page Designer an, so sieht man, dass APEX nicht etwa PL/SQL-Code mit APEX_WEB_SERVICE oder JSON_TABLE generiert – die Tatsache, dass der Bericht das Web Source Module als Datenquelle hat, wird vielmehr deklarativ, als Metadaten der Anwendung, gespeichert.

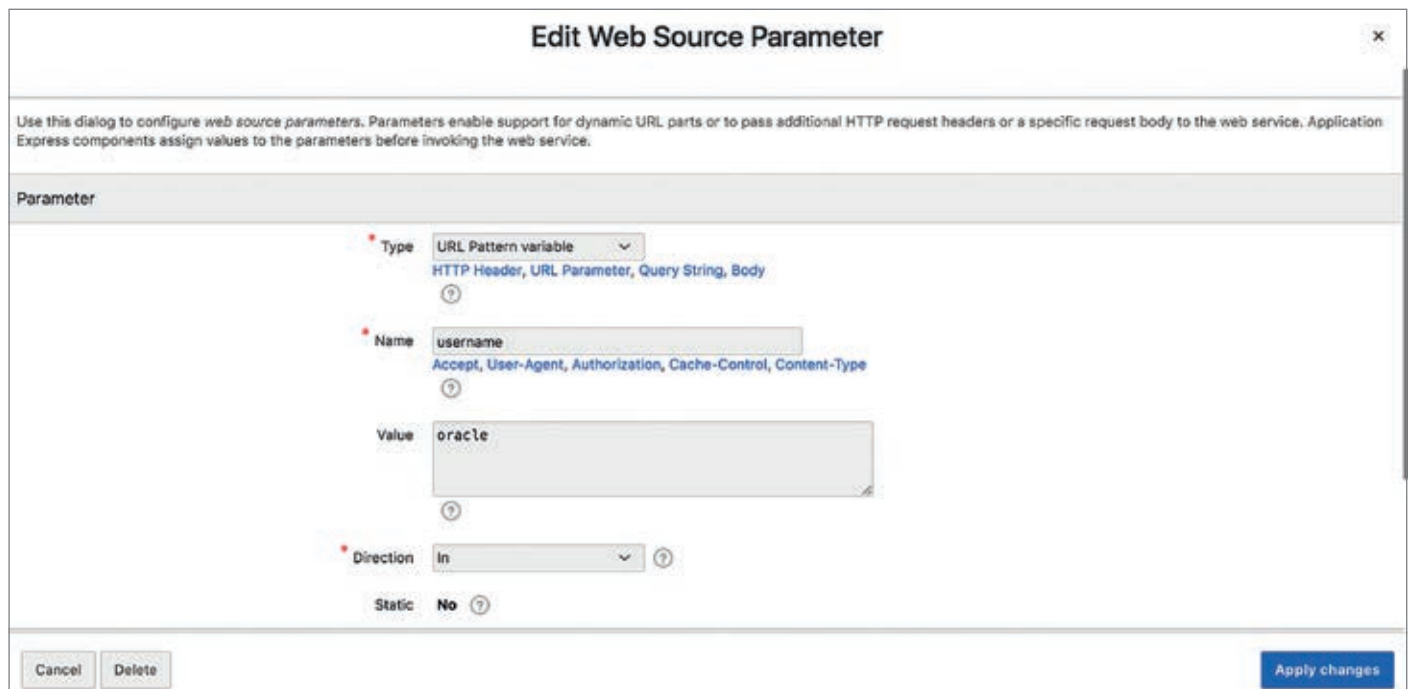


Abbildung 7: Ein Teil der URL („:username“) ist dynamisch. (Quelle: Carsten Czarski)

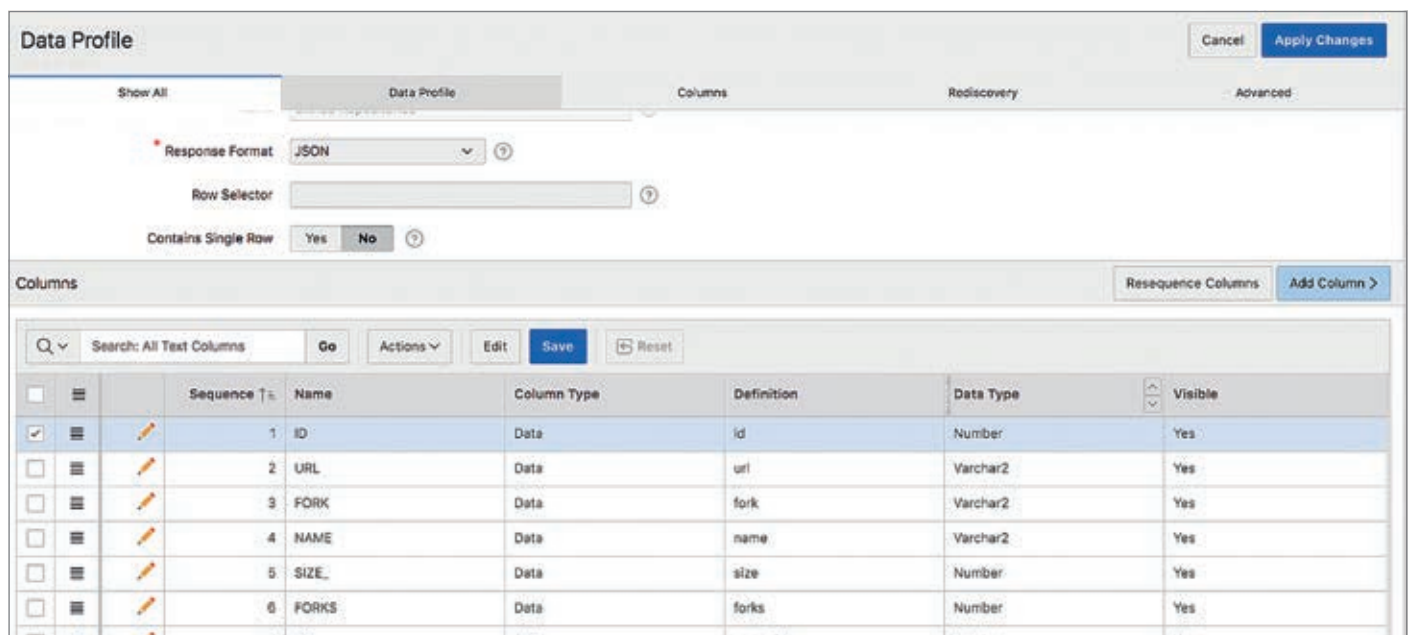


Abbildung 8: Das Data Profile enthält Angaben zur Struktur der JSON-Antwort (Quelle: Carsten Czarski)

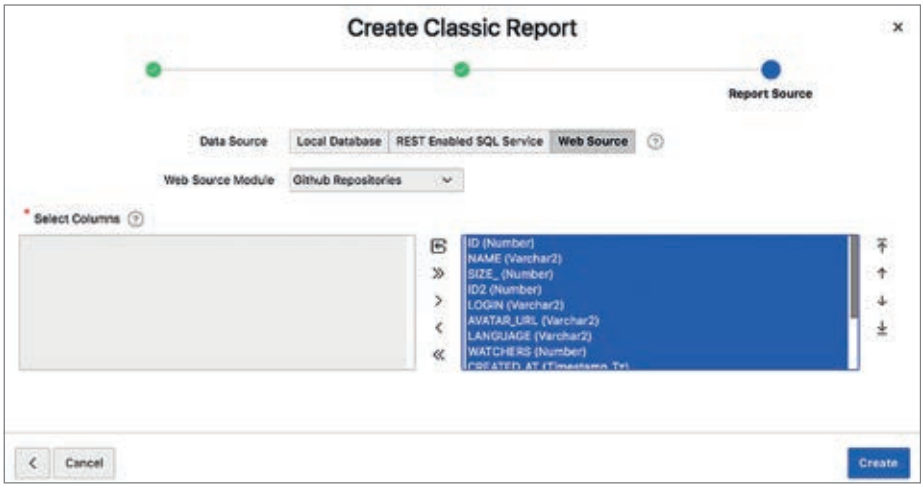


Abbildung 9: Einen Classic Report auf Basis des Web Source Module erstellen (Quelle: Carsten Czarski)

Der Classic Report funktioniert so, wie er funktionieren soll - man erkennt kaum Unterschiede zu einem Classic Report auf Basis einer Tabelle oder SQL-Abfrage. Auch Berichtsfeatures wie Formatmasken oder HTML-Ausdrücke können normal eingesetzt werden (siehe Abbildung 11).

Und auch der dynamische Teil der Schnittstellen-URL findet sich im Page Designer wieder. Der Classic Report hat einen neuen Knoten Parameters. Hier kann der Web Source Module Parameter username mit einem neuen Wert belegt oder mit einem APEX-Seitenelement (P1_ITEM_NAME) verknüpft werden (siehe Abbildung 12).

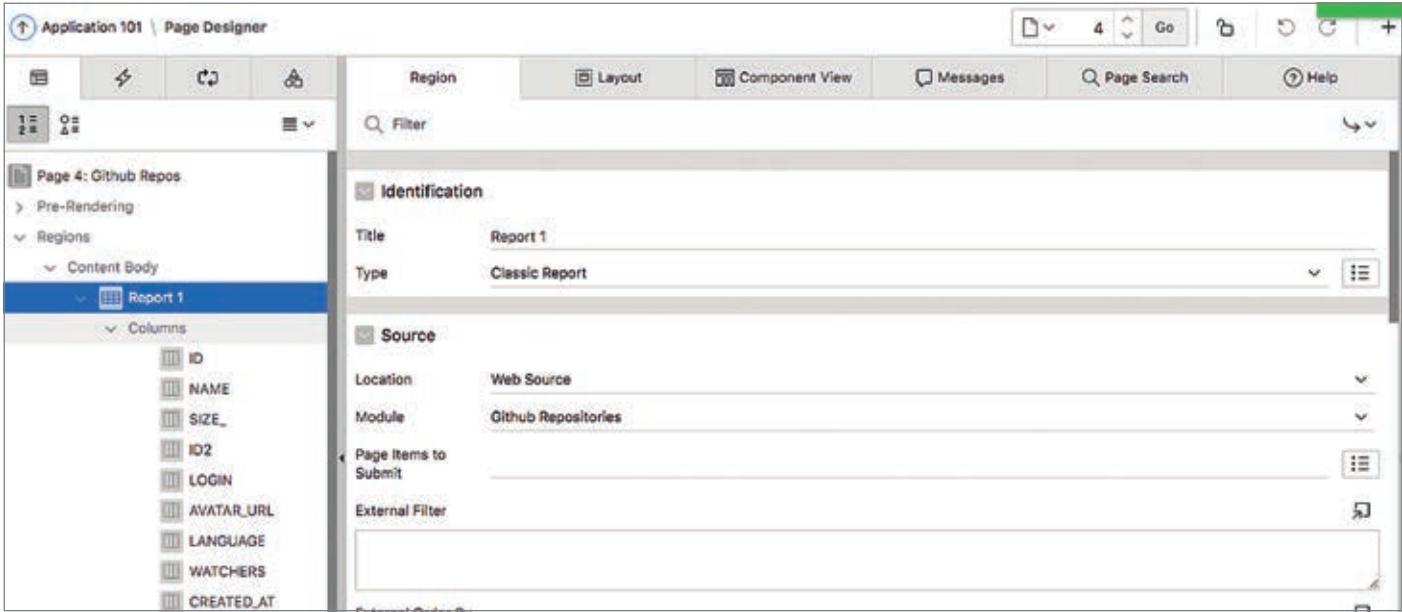


Abbildung 10: Classic Report auf Basis eines Web Source Module im Page Designer (Quelle: Carsten Czarski)

Github Repos											
Name	Size	Login		Language	Watchers	Created At	Forks Count	Open Issues	Watchers Count	Stargazers Count	Open Issues Count
analytical-sql-examples	599	oracle	ORACLE	PLSQL	105	5 years ago	66	0	105	105	0
accelerators	41KB	oracle	ORACLE	C#	26	4.2 years ago	54	1	26	26	1
big-data-lite	87KB	oracle	ORACLE	Java	36	3.8 years ago	31	0	36	36	0
compute-cloud-service-demos	34	oracle	ORACLE	Python	8	3.6 years ago	5	0	8	8	0
atg-build-tools	479	oracle	ORACLE	Java	11	3.5 years ago	11	1	11	11	1

Abbildung 11: Der Classic Report in Aktion (Quelle: Carsten Czarski)

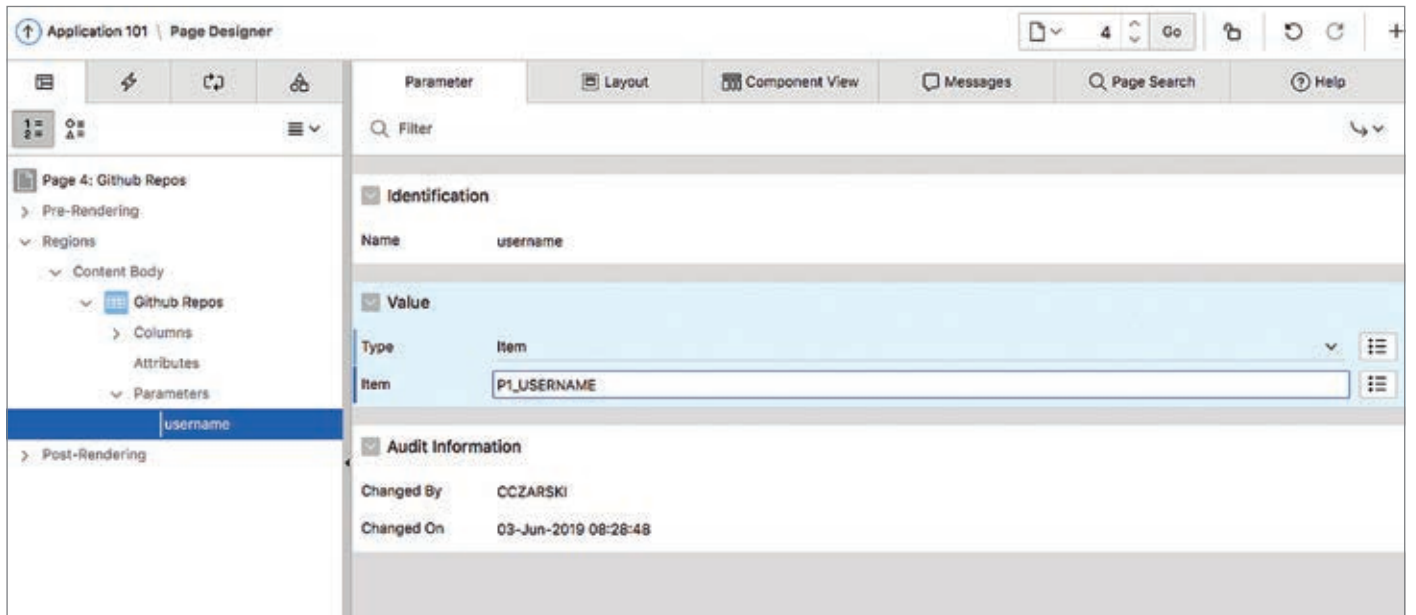


Abbildung 12: Ein APEX-Seitenelement wird mit dem Web Source Module verknüpft. (Quelle: Carsten Czarski)

Insbesondere dieser Parameter für das Web Source Module ist sehr interessant, denn auf der APEX-Seite wird er nur noch als Parameter für das Web Source Module verwendet. Ob dieser dann, beim konkreten Aufruf, als Teil der URL, als HTTP-Header oder Cookie an den REST Service gesendet wird, spielt für die Nutzung auf der APEX-Seite keine Rolle mehr. Diese Details werden komplett von der APEX-Seite abstrahiert und sind im Web Source Module gekapselt.

In APEX 18.1 und 18.2 können Web Source Modules für alle „Read Only“-Komponenten verwendet werden; hierunter fallen Classic oder Interactive Reports, Charts, Tree, CSS Calendar, Reflow Report, Toggle Column Report.

In APEX 19.1 wurde die Unterstützung für Formulare hinzugefügt. Dabei wird natürlich vorausgesetzt, dass die REST-Schnittstelle Methoden zum Einfügen („POST“), Ändern („PUT“) oder Löschen („DELETE“) eines Datensatzes anbietet und dass diese im Web Source Module hinterlegt sind. Für das kommende Release APEX 19.2 ist die Unterstützung von Web Source Modules im Interactive Grid und für gemeinsam verwendete (Shared Components) List of Values geplant.

REST Services 3: Web Source Modules und PL/SQL

Web Source Modules sind jedoch nicht auf die deklarative Nutzung mit APEX-

```

declare
  l_context apex_exec.t_context;
  l_columns apex_exec.t_columns;
begin
  -- Zum Zugriff auf das Web Source Module ist eine APEX-Session nötig.
  apex_session.create_session( 101, 1, 'CCZARSKI' );

  apex_exec.add_column( l_columns, 'ID' );
  apex_exec.add_column( l_columns, 'NAME' );
  apex_exec.add_column( l_columns, 'SIZE' );
  apex_exec.add_column( l_columns, 'LANGUAGE' );

  l_context := apex_exec.open_web_source_query(
    p_module_static_id => 'Github_Repositories',
    p_columns           => l_columns );

  while apex_exec.next_row( p_context => l_context ) loop
    dbms_output.put_line(
      '- ' || apex_exec.get_varchar2( l_context, 'NAME' )
      || ' ( ID '
      || apex_exec.get_varchar2( l_context, 'ID' ) || ' ) );
    dbms_output.put_line(
      ' ' || apex_exec.get_varchar2( l_context, 'LANGUAGE' ) );
    dbms_output.put_line(
      ' ' || apex_exec.get_varchar2( l_context, 'SIZE' ) );
    dbms_output.put_line( ' ' );
  end loop;

  apex_exec.close( l_context );
exception when others then
  apex_exec.close( l_context );
  raise;
end;

```

Listing 4: Zugriff auf das Web Source Module mit PL/SQL

Komponenten beschränkt. Das ab APEX 18.1 vorhandene PL/SQL-Paket APEX_EXEC macht sie auch für den PL/SQL-Entwickler verfügbar.

Doch wozu ist das gut? Schließlich kann man als PL/SQL-Entwickler doch gleich mit APEX_WEB_SERVICE arbeiten

und die REST-Schnittstelle so direkt ansprechen.

Der Vorteil des Web Source Module ist, dass alle technischen Details der REST-Schnittstelle dort gekapselt sind. Ein Web Source Module hat einen Namen, es hat Parameter und es gibt Zeilen und Spalten zurück. Wie der konkrete HTTP-Request aussieht und wie die JSON- oder XML-Antwort in Zeilen und Spalten übersetzt wird, ist für den Nutzer eines Web Source Module nicht mehr relevant. Ein Web Source Module dient hervorragend der Abstraktion der Geschäftslogik von den technischen Details der REST-Schnittstelle.

Das folgende *Listing 4* zeigt, wie das Web Source Module GitHub in einer PL/SQL Stored Procedure verwendet wird.

Zusammenfassung

Mit dem APEX Release 18.1 wurde das Integrieren fremder Systeme in Applica-

tion-Express-Anwendungen massiv vereinfacht. Sobald eine REST-Schnittstelle vorliegt, kann diese als Web Source Module in die APEX-Anwendung aufgenommen werden.

APEX speichert dann Metadaten zur REST-Schnittstelle, sodass der APEX-Entwickler sie deklarativ in APEX-Komponenten nutzen kann. Technische Aufgaben wie das Ausführen des HTTP-Request oder das Parsen der JSON- oder XML-Antwort übernimmt APEX. Der APEX-Entwickler kann sich voll und ganz auf die eigentliche Anwendung konzentrieren.

Mit APEX 18.1 werden zunächst Read-Only-Komponenten unterstützt; APEX 19.1 bringt Unterstützung für Formulare und das geplante APEX 19.2 wird die Unterstützung mit dem Interactive Grid sowie den Shared Component LOVs komplett machen.

Mit APEX ist man also mitnichten auf die „lokale“ Datenbank begrenzt – APEX-Anwendungen können alle möglichen Daten-

quellen ansprechen: REST-Services sind genauso Datenquelle wie eine lokale Tabelle.

Weitere Informationen

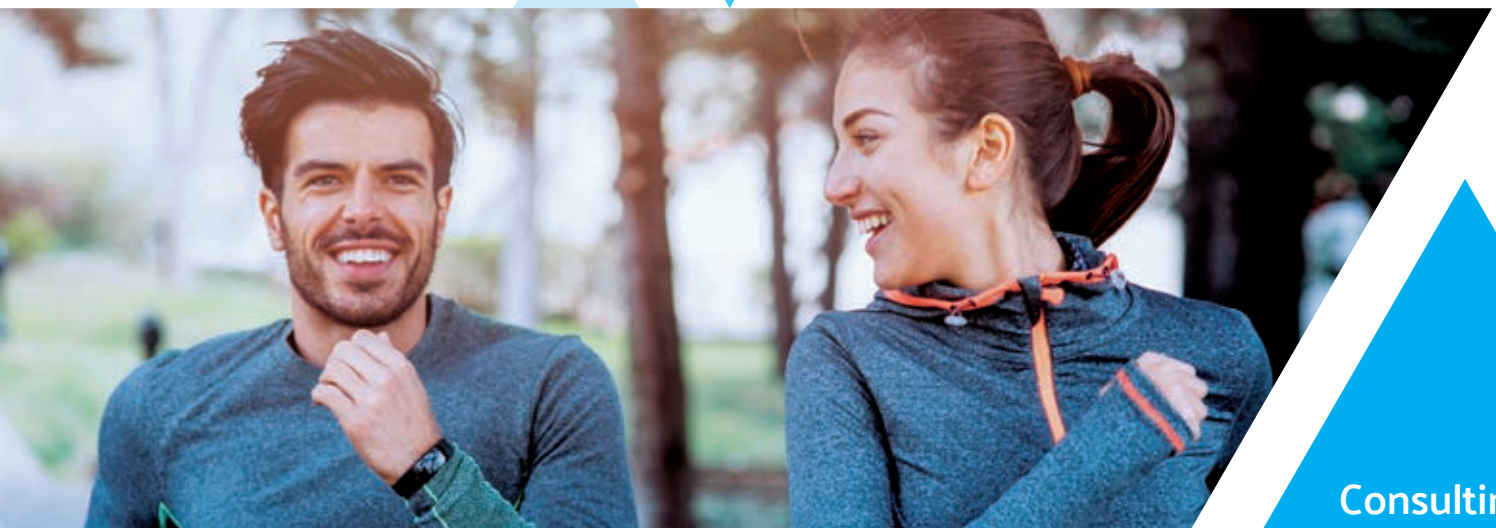
- Informationen zu Application Express und zur Demo-Umgebung <http://apex.oracle.com>
- Oracle Application Express Blog – mit vielen Postings zu REST Services in APEX <http://blogs.oracle.com/apex>



Carsten Czarski
carsten.czarski@oracle.com



MUNIQSOFT
— CONSULTING —



Consulting

Performance-Tuning mit IQ

Mehr Power für Ihre Oracle Lösungen!

Nutzen Sie unseren proaktiven Datenbank-Healthcheck als Startschuss für die Optimierung Ihrer Oracle Datenbanken.

Ungebremst ans Ziel mit der Muniqsoft Consulting GmbH
www.muniqsoft-consulting.de

ORACLE® Gold Partner

Specialized
Oracle Database



Jetzt Beratungstermin vereinbaren:
+49 89 62286789-39



Agil unterwegs mit APEX in einem stark regulierten Umfeld

Hansjörg Grässlin, Christophe Girardey, Dr. Christian Wattering, wega Informatik, Basel, Schweiz

Viele Branchen haben strenge Vorschriften (Autoindustrie, Pharma, Flugsicherung, Banken usw.), was die Umsetzung von IT-Projekten betrifft, und es gibt vielerorts eine Tendenz zu mehr und strengeren Regulierungen. APEX mit seiner Ausrichtung auf agile und schnelle Entwicklungsansätze (z.B. Low Code, Rapid Application Development) scheint auf den ersten Blick nicht per se für Lösungen in einem starren, regulierten Umfeld geeignet zu sein. Am Beispiel einer APEX-Implementierung in einer streng validierten Umgebung in der pharmazeutischen Industrie im Bereich der klinischen Entwicklung (Clinical Development) zeigen wir, wie diese inhärenten Probleme erfolgreich überwunden werden können.

In diesem Artikel möchten wir am Beispiel eines erfolgreichen Projekts, das wir in einem Großkonzern in der Pharma-Branche durchgeführt haben, darstellen, wie man die auf den ersten Blick widersprüchlichen Konzepte „agil“ und „streng reglementiert“ mit APEX zusammenbringen kann.

Dazu werden zuerst die beiden gegenüberstehenden Konzepte beschrieben, bevor der Lösungsansatz dargestellt wird.

Validierung ist...

...gemäß der amerikanischen Food and Drug Administration (FDA):

„Die Etablierung dokumentierter Nachweise, die ein hohes Maß an Sicherheit dafür bieten, dass ein bestimmter Prozess ein Produkt herstellt, das seinen vorgegebenen Spezifikationen und Qualitätsmerkmalen entspricht.“

In der Pharma-Industrie werden die regulatorischen Anforderungen als Validierung bezeichnet und in den verschiedenen Ländern von den entsprechenden Ämtern vorgegeben. In *Tabelle 1* sind einige relevante Beispiele aufgeführt:

In diesem Zusammenhang unterscheidet man zwischen System und Software. Dies ist wichtig, denn eine Validierung bezieht sich immer auf das ganze System,

Land	Institution	Regulatorien
USA	FDA	GLP: 21 CFR 58 GMP: 21 CFR 210 + 211 Electronic Records and Signatures: 21 CFR 11
EU		EudraLex - Volume 4: GMP Guidelines Annex 11: Computerised Systems
OECD		Principles of Good Laboratory Practice, Nr. 10 on Computerized Systems
Deutschland	Bundesamt für Arzneimittel und Medizinprodukte	GMP: Arzneimittel- und Wirkstoffherstellungsverordnung – AMWHV
Schweiz	Swissmedic	Bundesgesetz über Arzneimittel und Medizinprodukte (Heilmittelgesetz, HMG)

Tabelle 1: CSV Gesetzliche Grundlagen

in dem die Software nur ein – wenn auch wichtiger – Teilbereich ist. Das System beinhaltet unter anderem aber auch die Prozeduren und Personen, die involviert sind (siehe Abbildung 1).

Zentraler Leitfaden der Validierung ist GAMP 5 [2] (Good Automated Manufacturing Practice), der einen risikobasierten Ansatz beschreibt, um Systeme zu validieren. GAMP teilt Software in die in Tabelle 2 aufgeführten Kategorien auf, wobei der Validierungsaufwand mit der Kategorie ansteigt.

Im Pharma-Bereich ist es in folgenden Fällen angezeigt und vorgeschrieben, ein IT-System zu validieren, falls das System eines der folgenden Themen betrifft:

- Sicherheit von Personen, insbesondere Patienten
- Produktqualität
- Daten-Integrität und damit letztlich das Vertrauen in die Daten (wichtig zum Beispiel bei klinischen Studien für die Zulassung eines Medikaments)

In allen diesen Fällen soll eine Validierung sicherstellen, dass Fehlerquellen minimiert werden und die Zuverlässigkeit der Prozesse maximiert wird. Des Weiteren müssen auftretende Prozessabweichungen klar erkennbar und nachvollziehbar sein.

Im validierten Umfeld der Pharma-Industrie werden Software-Entwicklungen meist nach dem V-Modell durchgeführt. Dieses ist in Abbildung 2 dargestellt: Beginnend oben links wird der folgende sequenzielle Prozess durchlaufen:

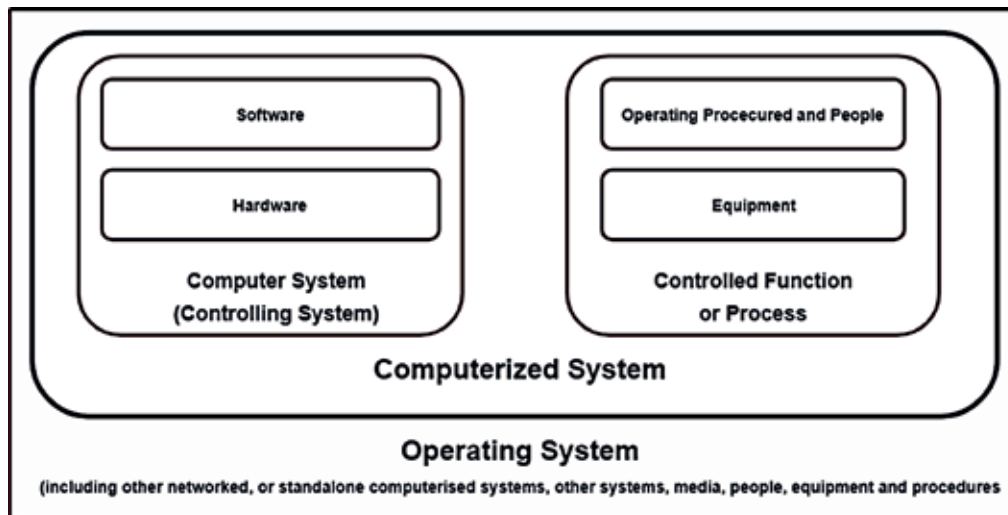


Abbildung 1: Die Software ist nur ein Teil des zu validierenden Gesamtsystems. (Quelle: PIC/S (2007). [1])

SW Kat.	Name	Beispiele
1	Infrastruktur-Software	Betriebssystem, Middleware, Datenbank, Monitoring Tools, Anti-Virus Programme etc.
2	Nicht mehr verwendet in GAMP 5	N/A
4	Konfigurierte Software	LIMS, gekaufte Datenbankanwendungen, Ressourcenplanung etc. (die meisten Programme fallen darunter).
5	Selbstentwickelte Software	Datenbank-Programme, Web-Anwendungen etc.

Tabelle 2: SW-Kategorien nach GAMP 5 [2]

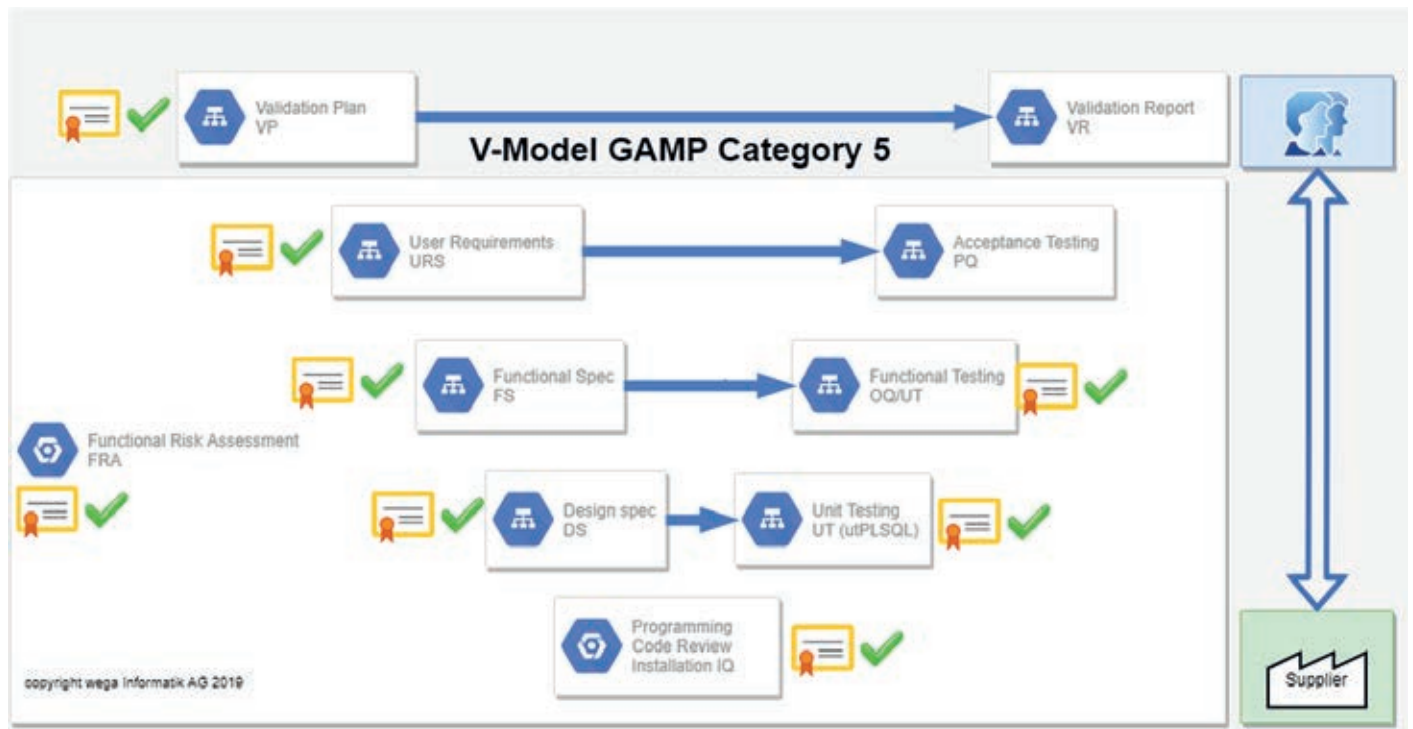


Abbildung 2: Validierung im V-Modell (Quelle: wega Informatik AG).

- Erstellen und Abnahme eines Validation-Plans (VP), der beschreibt, wie das System validiert werden soll.
- Erstellen und Abnahme der User Requirements Specification (URS), die beschreibt, welche Nutzeranforderungen an das System gestellt werden.
- Erstellen und Abnahme der Functional Specification (FS), die erläutert, wie die URS funktional umgesetzt wird.
- Erstellen und Abnahme der Design Specification (DS). Hier wird genau dargestellt, wie die Funktionalitäten aus der FS technisch implementiert werden.
- Dann – ganz unten im V-Modell – wird die Software programmiert. Am Ende der Implementierung erfolgt ein Code Review, um die Programmierung durch das 4-Augen-Prinzip einem Qualitätscheck zu unterwerfen. Hieraus wird eine zu testende Installationsanleitung (IQ) erstellt und abgenommen.
- Auf der rechten Seite, einen Schritt hoch, erfolgt dann die Ausführung der Unit-Tests, um die akkurate Umsetzung gemäß DS zu überprüfen. Test-Abweichungen müssen formell behandelt und am Ende müssen alle Test-Resultate abgenommen werden.
- Erst dann geht es wieder einen Schritt hoch, wo die funktionalen Tests (OQ) die korrekte Umsetzung der Funktionalität gemäß FS überprüfen. Wieder-

um gilt hier wie oben: Test-Abweichungen müssen formell behandelt und am Ende alle Test-Resultate abgenommen werden.

- Erst dann wird der Abnahme-Test (PQ) durch die Benutzer durchgeführt!
- Ganz am Ende wird ein Validation-Report (VR) erstellt, der darstellt, wie alle geplanten Prüfungs- und Abnahmeschritte erfolgreich durchlaufen wurden; er beweist so die nachvollziehbare Qualität des Resultats – also der fertigen Software. Die formale Freigabe des Validation-Reports führt zur Freigabe des Systems für die produktive Verwendung.

Zentrales Werkzeug des gesamten Prozesses ist die funktionale Risiko-Analyse (FRA). Hierbei werden die verschiedenen möglichen Risiken analysiert und bewertet. Auf Basis dieser Bewertungen werden Maßnahmen festgelegt, die dazu dienen, das Risiko zu minimieren. Dies kann zu zusätzlichen Testaktivitäten, prozeduralen Lösungen, zusätzlichem Training oder gar zu Designänderungen führen.

Es wird unmittelbar klar, dass dies ein sehr dokumentationslastiger, unflexibler Prozess ist. Der Validierungsaufwand ist oft viel höher als die effektive Programmierung und Anpassungen können mit unverhältnismäßig großem Aufwand verbunden sein, vor allem, wenn sie spät

erkannt werden (z.B. erst im Abnahmetest durch die Endbenutzervertreter – ein durchaus typischer Fall). Dabei sind auch die vielen Unterschriften für die Abnahmen zu berücksichtigen; das Sammeln aller nötigen Unterschriften kann oft lange dauern, wenn Verantwortliche nicht verfügbar sind. Dies ist insbesondere darum kritisch, da man im V-Modell nicht mit dem nächsten Schritt beginnen darf, bevor nicht alle vorhergehenden Schritte durch Unterschriften abgenommen sind.

Dies widerspricht in vielfältiger Weise dem modernen agilen Ansatz – zum Beispiel von Scrum –, der sich in der Software-Entwicklung wachsender Beliebtheit erfreut, und ist somit fast das genaue Gegenteil von dem flexiblen und iterativen Vorgehen, das wir in der modernen Softwareentwicklung eher gewohnt sind.

Scrum

Scrum ist unterdessen in vielen Organisationen die Methodologie der Wahl für die agile Software-Entwicklung.

Hier wollen wir diese Methodologie daher nur kurz umreißen, um die Gegensätzlichkeit zum Konzept der Validierung herauszustrichen.

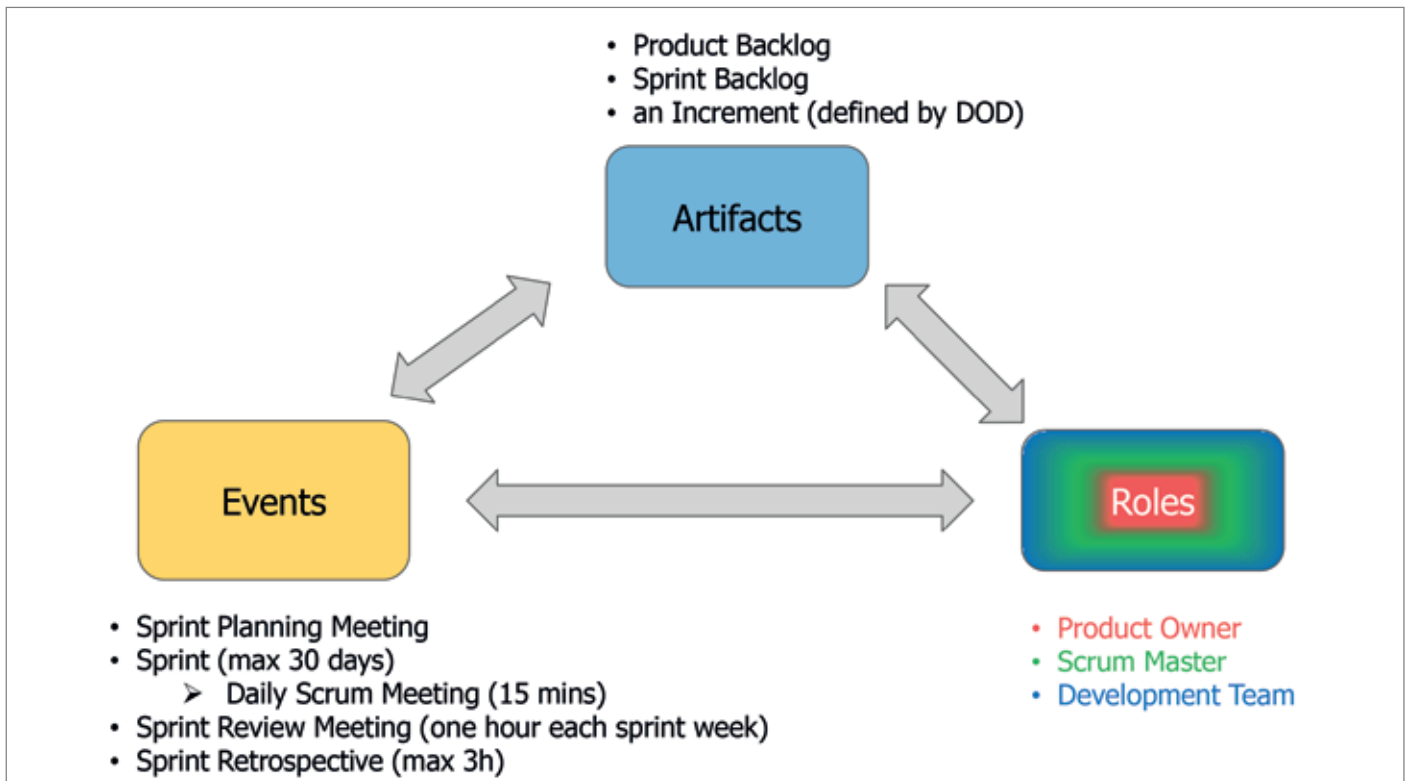


Abbildung 3: Elemente des Scrum-Frameworks (Quelle: wega Informatik AG).

Die Elemente des Scrum-Frameworks sind in *Abbildung 3* dargestellt:

Scrum ist ein iterativer Prozess, in dem es darum geht, dass das Team in jeweils gleich langen Arbeitsblöcken – „Sprints“ – ein sogenanntes Backlog abarbeitet. Dabei wird mehr Wert auf funktionierende Software als auf extensive Dokumentation gelegt.

Regelmäßige Retrospektiven und tägliche kurze „Daily Scrum Meetings“, in denen jeweils alle Probleme direkt und zeitnah angesprochen werden, dienen der Verbesserung der Teamarbeit und somit der Erhöhung der Umsetzungsgeschwindigkeit („Velocity“) des Teams, also der Kapazität, während eines Sprints mehr Arbeit erfolgreich zu erledigen.

Drei Grundpfeiler von Scrum sind daher:

1. Offenes und direktes Ansprechen von Hürden – „Transparency“
2. Selbsterprüfung des Teams und der Prozesse – „Inspection“
3. Anpassungen und Verbesserung des Teams – „Adaption“

Die Herausforderungen der Validierung mit Oracle APEX

In der Validierung sind sich ändernde Anforderungen eine große Herausforderung.

Hier kollidiert die agile Methodologie mit dem V-Modell. Im agilen Umfeld sind Anforderungen nicht festgelegt, sie können sich während des Projekts laufend ändern. Dies ist eine Stärke des agilen Ansatzes.

Umgekehrt basiert die Wasserfall-Methode (V-Modell) auf starren Anforderungen.

Wie also kann man das regulierte V-Modell mit einer agilen Methodologie umsetzen?

Eine zweite Herausforderung in Zusammenhang mit APEX ist die starke Betonung der Nachvollziehbarkeit von Anpassungen im validierten Umfeld. APEX bietet kein eingebautes Audit Trail, woraus direkt ersichtlich ist, wer, wann, was aus welchem Grund geändert hat. Versionskontrolle – zum Beispiel mit Git oder Subversion – ist nicht in der APEX-Kernphilosophie enthalten.

Wie kann dieser Umstand effizient adressiert werden, sodass die für die Validierung kritische Nachvollziehbarkeit gewährleistet werden kann?

Diese Punkte zeigen wir im Folgenden anhand eines Kundenprojekts auf.

Ein validiertes Kundenprojekt mit Oracle APEX

Der Kunde ist eine der zehn größten Pharma-Firmen der Welt. Das Projekt

war in der klinischen Entwicklung angesiedelt.

Die unterstützten Prozesse sind stark reglementiert, da die Software innerhalb des Zulassungsverfahrens eingesetzt wurde. Das gewonnene Vertrauen in die erhobenen Studiendaten hat eine direkte Auswirkung auf die Zulassung des Medikaments. Außerdem könnte ein Software-Fehler während einer Studie Patienten gefährden – mit allen sich daraus ergebenden Konsequenzen.

Die vom Kunden vorgegebenen zu erbringenden Validierungs-Services beinhalten das Erstellen der vom V-Modell vorgeschriebenen Dokumente in HP ALM (ein Application Lifecycle Management System), das Durchführen von Code Reviews (4-Augen-Prinzip), die Durchführung der Testzyklen (Unit-Tests – mit utPLSQL, OQ) und die Installation (gemäß IQ).

Der pragmatische Approach

In dieser Konstellation haben wir uns für eine pragmatische Herangehensweise entschieden und diese während des Projekts laufend verfeinert.

Für das gemeinsame Arbeiten an Dokumenten verwendeten wir die Kollaborationstools von Microsoft (MS Teams) sowie

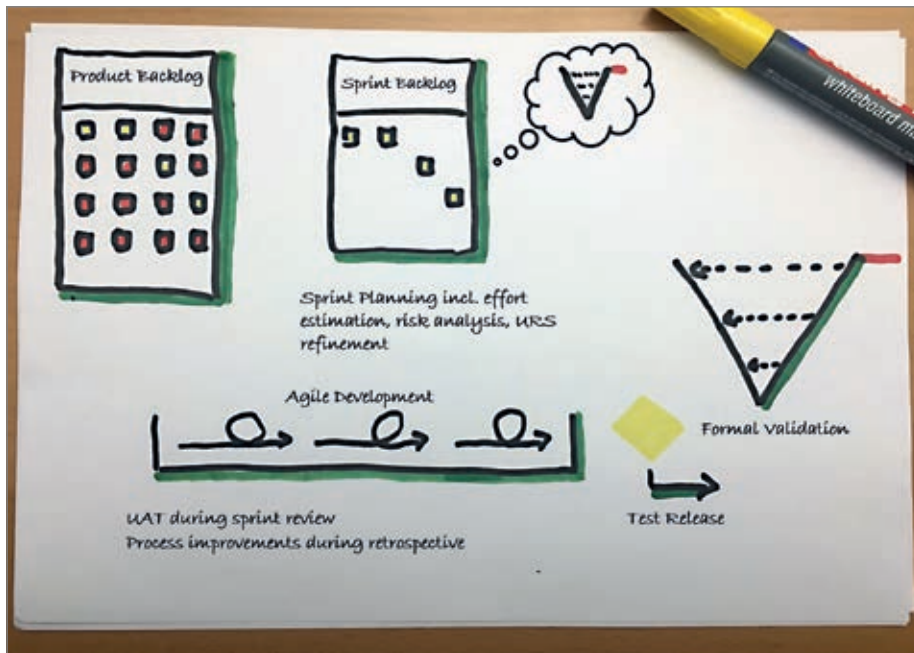


Abbildung 4: Agile Entwicklung in mehreren Sprints, gefolgt durch eine formale Validierung in einem Validierungs-Sprint (Quelle: wega Informatik AG).

Atlassian Jira für das agile Tracking der Arbeitsblöcke und zum Planen der Sprints.

Wir haben das Projekt in Zeitintervalle eingeteilt, analog Sprints, und in

diesen Intervallen jeweils agil gearbeitet. Am Ende eines Intervalls haben wir eine Baseline definiert und darauf die Wasserfall-artige Validierung gemäß V-

Modell (FS-DS-UT-IQ-OQ-PQ) in einem Validierungs-Sprint angewendet, um die formale Freigabe zu erreichen (siehe Abbildung 4).

Die festgestellten Abweichungen und Anpassungen aus Unit-Tests, System-Tests (OQ) und Benutzer-Akzeptanz-Tests (PQ) haben wir dann in einem Change-Management-System erfasst und anschließend die damit verbundenen Risiken durch das Functional Risk Assessment (FRA) bewertet.

Bei der Umsetzung haben wir starken Wert darauf gelegt, den APEX-Teil einfach zu halten. Das heißt unter anderem:

1. den Source Code möglichst gering zu halten durch Ausnutzen des Low-Code-Ansatzes von APEX,
2. PL/SQL-Code in APEX wurde konsequent auf DB-Packages ausgelagert (z.B. pipelined functions in APEX Interactive Reports (IR), Datenbankfunktionsaufrufe in APEX-Prozessen, Berechtigungssteuerung in Datenbankfunktions-Aufrufen),
3. alle Datenbankobjekte (Packages, Functions, Views etc.) wurden konsequent der Versionskontrolle in Git unterwor-

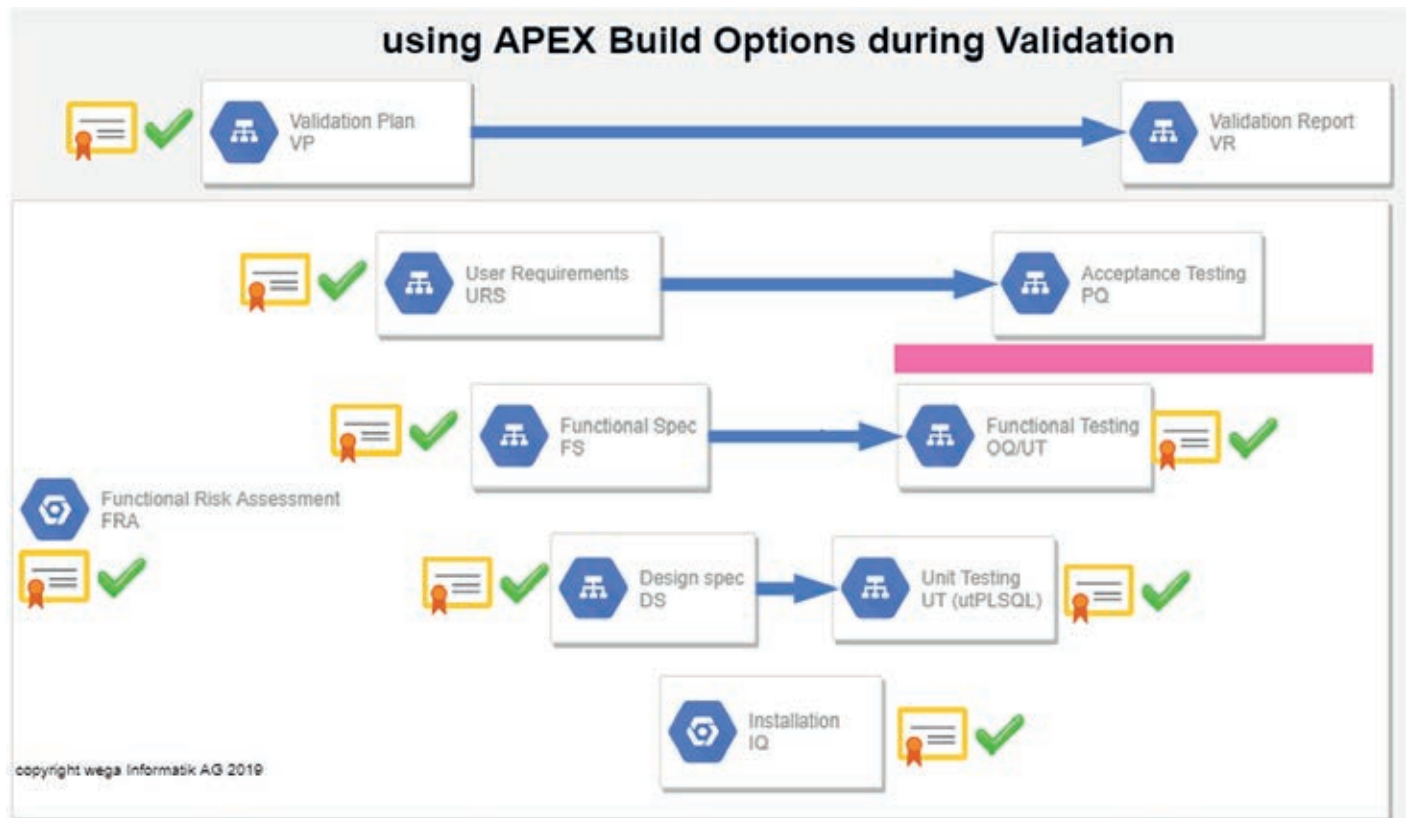


Abbildung 5: Benutzung von APEX Build Options erlaubt das Einbauen und Vortesten (bis zur roten Linie im Diagramm) von neuen Funktionen ohne Bereitstellung an den Endbenutzer (Quelle: wega Informatik AG).

fen. Git wurde dabei als single source of truth definiert.

Coding Guidelines sind sehr wichtig, schon zu Beginn des Projekts. Diese Richtlinien hatten wir nicht starr fixiert, sondern laufend in den Sprint Retrospectives weiterentwickelt und verfeinert. Diese Guidelines dienten dann als Basis für den formellen Code-Review-Prozess.

Im Laufe des Projekts hatten wir auch Gelegenheit, aus Fehlern zu lernen: Das Error Handling hat sich als sehr wichtig herausgestellt. Das Persistieren von Fehlern, Warnungen und Informationen stellte sich für das Debugging als unverzichtbar heraus. Hier ist aus unserer Sicht die Open-Source-Software Logger von OraOpenSource [3] sehr empfehlenswert, da es das gesamte Error Handling standardisiert. Des Weiteren hat es sich als sehr nützlich erwiesen, eine APEX-Seite für das Information- und Error-Log einzubauen, die mit Admin-Berechtigung sichtbar ist, um die Transparenz zu erhöhen.

Wir setzten die sogenannten „Build Options“ von APEX ein, um flexibel Funktionalitäten einbauen zu können (siehe Abbildung 5). Durch einen einfachen Configuration Change konnten diese „versteckten“ Funktionalitäten für den Endbenutzer einfach ein- und aus-

geschaltet werden (Pages, Menü Items etc.). Dies diente zur Verifizierung der neuen Funktionalitäten ohne deren Bereitstellung für die Endbenutzer; so konnte das Einführen von Funktionalität vom starren Gerüst des V-Modells entkoppelt werden.

Fazit

Durch diese pragmatischen Maßnahmen gelang es uns, die agilen Prinzipien erfolgreich zum Vorteil für das Projekt zu nutzen und gleichzeitig den strengen regulatorischen Anforderungen zu genügen.

Dies hat entscheidend dazu beigetragen, unser Projekt erfolgreich und zur Zufriedenheit des Kunden umzusetzen.

Die gezogenen Lehren können auch für ähnlich gelagerte APEX-Projekte in streng regulierten Umgebungen hilfreich sein.

Quellen

- [1] Components of a validated computerized system, PIC/S Quelle: PIC/S (2007)
- [2] GAMP5, A Risk based Approach to Compliant GxP Computerized Systems, ISPE (2008). Tampa, Fla. ISPE Headquarters [u.a].
- [3] Logger, OraOpenSource, <https://github.com/OraOpenSource/Logger>



Hansjörg Grässlin
Hansjoerg.Graesslin@wega-it.com



Christophe Girardey
christophe.girardey@wega-it.com



Dr. Christian Wattering
christian.wattering@wega-it.com

WECHSEL VON MY ORACLE SUPPORT ZU THE IDEA LABS FÜR ERP UND SCM CLOUD

Ab 1. August 2019 sollen The Idea Labs für Verbesserungsprozesse bei Oracle ERP und SCM Cloud zuständig sein.

Bei The Idea Labs handelt es sich um Feedback-Foren, die Oracle-Kunden an Verbesserungsprozessen (Enhancement Request Processes) teilhaben lassen. Die Foren sind eingebettet im Cloud Customer Connect. Kunden können dort Ideen einreichen, sich an Lösungsansätzen beteiligen, über Vorschläge abstimmen und die anschließende Implementierung beobachten.

Wenn eine Serviceanforderung (Service Request) als Produkterweiterung

identifiziert wird, kann diese als Idee im passenden Forum festgehalten werden. Anders als bisher bei My Oracle Support wird die neue Forum-Umgebung dazu führen, dass Kunden mit anderen ERP- und SCM-Cloud-Kunden, -Partnern und -Produktentwicklern ins Gespräch kommen. Das ERP- und SCM-Cloud-Entwicklerteam soll zudem Kunden regelmäßig über Updates der am häufigsten abgestimmten Ideen informieren.

Dr. Frank Schönthaler, Vorstand und Leiter der DOAG Business Solutions Community, bewertet die Entscheidung positiv: „Das ist eine gute Sache, weil einfach mehr Transparenz in den Verbesserungsprozess kommt. Es steht zudem zu erwarten, dass so – bei aller Wichtigkeit – nicht Bug Fixing im Vordergrund stehen wird, sondern zukunftsgerichtete Innovationen. Die DOAG Business Solutions Community wird die Entwicklung im Auge behalten.“



Datenbank in der Wolke – Teil 3: Cloud-Lösungen in der Praxis

Borys Neselovskyi, OPITZ CONSULTING Deutschland GmbH

Die Cloud-Technologie ist heute nicht mehr wegzudenken. Viele Unternehmen nutzen bereits Public Cloud Services oder betreiben einige Teile ihrer Infrastruktur in der Cloud. Auch der Datenbankbetrieb wird immer häufiger von eigenen Rechenzentren in die Cloud verlagert. Die Vorteile der Cloud sind dabei vielfältig. Sie reichen von konkreten Kosteneinsparungen über die Entlastung von Fachkräften bis hin zu einem Mehr an Übersichtlichkeit und Flexibilität. In den letzten beiden Ausgaben erschienen die Teile 1 und 2 dieser Serie. In Teil 1 wurden die Vorteile der Cloud diskutiert sowie Abrechnungsmodelle und verschiedene Dienste von Oracle im Detail betrachtet. Der zweite Teil dieser Serie beschäftigte sich mit Infrastrukturen in der Cloud. Dieser dritte und letzte Teil der Serie stellt einige praxisbezogene Anwendungsszenarien und dafür passende Cloud-Lösungen vor.

Szenario 1: Kleine Test- bzw. Entwicklungsdatenbank

(siehe auch Tabelle 3)

Anforderungen:

- Verwendung: Test- bzw. Entwicklungsdatenbank
- Betriebszeiten: Betrieb an Werktagen von 8:00 bis 17:00 Uhr (berechnet werden 22 Tage mit jeweils 9 Stunden)

- Hochverfügbarkeitsanforderungen: keine
- Datenbankressourcen:
 - CPU: 1-4
 - RAM: 5-10 Gigabyte
 - Datenbank-Storage: bis 1 Terabyte

Lösung 1: Autonome Datenbank für OLTP (Online Transaction Processing)

Bei der autonomen Datenbank für OLTP von Oracle handelt es sich um eine kom-

plett administrierte Container-Datenbank in der Cloud, auch Pluggable Datenbank oder kurz PDB genannt, die automatisch und unter Verwendung des maschinellen Lernens verwaltet wird. Administrative Tätigkeiten wie Patching, Aktualisierung, Sicherung und sogar Performance Tuning wurden komplett automatisiert. Der Storage für die Datenbank-Backups ist bereits inkludiert. Da die autonome Datenbank auf der Exadata läuft, profitieren die Nutzer von besonde-

ren Eigenschaften wie Storage-Indexierung. Oracle garantiert für die autonome Datenbank 99,995% Verfügbarkeit, sowohl für geplante als auch für ungeplante Ausfälle. Darüber hinaus stehen fast alle Eigenschaften der Enterprise Edition inklusive der Optionen den Nutzern zur Verfügung.

Nachteile: Der Endkunde hat keinen Einfluss auf die Auswahl der Datenbankversion und den Patch-Stand, was die Flexibilität der Lösung einschränkt. Momentan wird die autonome Datenbank in der Version 18.4 angeboten. Darüber hinaus gibt es keinen Zugriff auf die Betriebssystemebene, sodass keine benutzerspezifischen Einstellungen möglich sind.

In diesem Beispiel wird eine autonome OLTP-Datenbank vorgestellt, die mit einer CPU, acht Gigabyte Memory für SGA (Shared Global Area), drei Gigabyte Memory für PGA (Program Global Area) und einem Terabyte Datenbank-Storage (Oracle Exadata) ausgestattet ist.

Da die Datenbank nur an Werktagen genutzt wird, bietet sich das Finanzierungsmodell „Pay as You Go“ (PAYG) an, bei dem die tatsächlich genutzten Ressourcen auf Stundenbasis abgerechnet werden.

Die Preise für die autonome OLTP-Datenbank [1] gliedern sich folgendermaßen auf:

- Oracle Autonomous Transaction Processing, pro Stunde und OCPU: 2,188 €
- Oracle Autonomous Database Exadata Storage, ein Terabyte pro Monat: 192,7404 €

Die Berechnung der monatlichen Kosten bei der Nutzung der autonomen OLTP-Datenbank für eine CPU und ein Terabyte Storage setzt sich folgendermaßen zusammen (siehe auch Tabelle 1):

- Oracle Autonomous Transaction Processing: $2,188 \text{ €} \times 1 \text{ OCPU} \times 22 \text{ Tage} \times 9 \text{ Stunden} = 433,224 \text{ €}$
- Oracle Autonomous Database Exadata Storage (1 TB pro Monat): 192,7404 €
- Monatliche Gesamtkosten: 625,9644 €

Lösung 2: Compute Classic Shape OC4

Bei dieser Lösung handelt sich um eine virtuelle Maschine mit folgenden Parametern:

- OCPU: 2
- RAM: 15 GB
- Datenbank-Storage (Block): 1 TB

Die Preise für Compute-Einheiten und Block Storage sind nachfolgend aufgelistet [2]:

- Compute Classic (pro Stunde und OCPU): 0,0738 €
- Block Storage Classic (1 GB pro Monat): 0,0369 €

Kosten beinhalten die Ausgaben für Compute-Einheiten, Block Storage und für die Datenbank-Edition. Anbei die Preise für die Datenbank-Editionen in der Cloud [3]:

- Standard Edition (pro Stunde und OCPU): 0,3501 €
- Enterprise Edition (pro Stunde und OCPU): 0,7001 €

Für die Datenbanksicherungen berechnen wir ein Terabyte Object Storage [4]:

- Object Storage – Storage (1 GB pro Monat): 0,0221 €
- Object Storage – Requests (10000 Requests pro Monat): 0,0369 €

Die Berechnung von monatlichen Kosten bei der Nutzung des OC4 Shape mit zwei OCPUs und einem Terabyte Storage setzt sich folgendermaßen zusammen (siehe auch Tabelle 2):

- Variante mit dem Standard Package:
 - Compute: $0,0738 \text{ €} \times 2 \text{ OCPU} \times 22 \text{ Tage} \times 9 \text{ Stunden} = 29,2248 \text{ €}$
 - Storage: $0,0369 \text{ €} \times 1.000 = 36,90 \text{ €}$
 - Standard Package: $0,3501 \text{ €} \times 2 \text{ OCPU} \times 22 \text{ Tage} \times 9 \text{ Stunden} = 138,6396 \text{ €}$
 - Backup Storage: $0,0221 \text{ €} \times 1.000 + 0,0369 \text{ €} = 22,1369 \text{ €}$
 - Monatliche Gesamtkosten: 226,9013 €
- Variante mit dem Enterprise Package:
 - Compute: $0,0738 \text{ €} \times 2 \text{ OCPU} \times 22 \text{ Tage} \times 9 \text{ Stunden} = 29,2248 \text{ €}$
 - Storage: $0,0369 \text{ €} \times 1.000 = 36,90 \text{ €}$
 - Enterprise Package: $0,7001 \text{ €} \times 2 \text{ OCPU} \times 22 \text{ Tage} \times 9 \text{ Stunde} = 277,2396 \text{ €}$
 - Backup Storage: $0,0221 \text{ €} \times 1.000 + 0,0369 \text{ €} = 22,1369 \text{ €}$
 - Monatliche Gesamtkosten: 365,5013 €

Konfiguration	Anzahl OCPU	PGA/SGA (GB)	DB Storage (GB)	Monatliche Kosten
Autonome OLTP	1	8/3	1024	625,9644 €

Tabelle 1: Lösung mit der Oracle Autonomous OLTP-Datenbank

Shape	Anzahl CPU	RAM (GB)	DB Storage (TB)	DB Package	Monatliche Kosten
OC4	2	15	1	Standard	226,9013 €
OC4	2	15	1	Enterprise	365,5013 €

Tabelle 2: Lösung mit dem Database Cloud Classic Service: Shape OC4

	Exadata Express X500	OCI Classic OC4
DB-Versionen	18: +	11.2, 12.1, 12.2, 18: +++
Verfügbarkeit	+++ (99,995%)	Data Guard möglich: +
DB-Optionen	Alle: +++	Standard/Enterprise: +
Hardware-Ausstattung	++ (mehr OCPU)	+
Storage-Intelligenz	Smart Scans, Storage-Indexierung: +++	-
DB-Performance	++	+
Storage-Größe	+	+
Verwaltungsaufwand	+++	+
Kosten	+	+++

Tabelle 3: Direkter Vergleich der Lösungen für Szenario 1

Fazit

Lösung 1 in diesem Szenario eignet sich für Kunden, die

- über kein oder geringes Know-how im Bereich Datenbankadministration verfügen,
- in der Testumgebung Exadata-Eigenschaften nutzen wollen,
- für eine Testumgebung hohe Verfügbarkeitsanforderungen stellen,
- die neusten Datenbankversionen einsetzen.

Lösung 2 kann für Nutzer interessant werden, die

- Erfahrung in der Datenbank- und Linux-Administration mitbringen,
- die Datenbankversion bestimmen wollen,
- eine preiswertere Variante bevorzugen.

Szenario 2: OLTP bzw. Data-Warehouse-Datenbanken (mittlere Größe)

(siehe auch Tabelle 4 und 5)
Anforderungen:

- Verwendung: Mittlere OLTP oder Data-Warehouse-Datenbank, Produktion, Integration, Wartungsumgebung, Test und Entwicklung
- Betriebszeiten: 24x7-Betrieb
- Hochverfügbarkeitsanforderungen, mittlere Stufe: Kein Backup in Object Storage, RAC und Data Guard notwendig
- Datenbankausstattung: Standard Edition oder Enterprise Edition
- Datenbankressourcen:
 - CPU: 4-8
 - RAM: mind. 50 GB
 - Datenbank-Storage: 5 bis 12 TB

Lösung 1: OCI Virtual Image VM.Standard2.4
Bei dieser Lösung handelt sich um eine virtuelle Maschine mit folgenden Parametern:

- OCPU: 4
- RAM: 60 GB
- Datenbank-Storage (Block): 10 TB

Die Preise für Compute-Einheiten und Block Storage sind nachfolgend aufgelistet [4], [5]:

- Compute (pro Stunde und OCPU): 0,0554 €
- Block Storage (1 GB pro Monat): 0,0369 €

Die Kosten beinhalten die Ausgaben für Compute-Einheiten, Block Storage und

für die Datenbank-Edition. Anbei die Preise für Datenbank-Editionen in der Cloud:

- Standard Edition (pro Stunde und OCPU): 0,3501 €
- Enterprise Edition (pro Stunde und OCPU): 0,7001 €

Für die Datenbanksicherungen berechnen wir 10 Terabyte Object Storage [5]:

- Object Storage – Storage (1 GB pro Monat): 0,0221 €
- Object Storage – Requests (10.000 Requests pro Monat): 0,0369 €

Die Berechnung von monatlichen Kosten bei der Nutzung des VM.Standard2.4 Shape mit 4 OCPUs und 10 Terabyte Storage setzt sich folgendermaßen zusammen:

- Variante mit dem Standard Package:
 - Compute: 0,0554 € x 4 CPU x 31 Tage x 24 Stunden = 164,8704 €
 - Storage: 0,0369 € x 10.000 GB = 369 €
 - Standard Package: 0,3501 € x 4 OCPU x 31 Tage x 24 Stunden = 1041,8976 €
 - Backup Storage: 0,0221 € x 10.000 GB + 0,0369 € = 221,0369 €
 - Monatliche Gesamtkosten: 1.796,8049 €
- Variante mit dem Enterprise Package:
 - Compute: 0,0554 € x 4 OCPU x 31 Tage x 24 Stunden = 164,8704 €
 - Storage: 0,0369 € x 10.000 GB = 369 €
 - Enterprise Package: 0,7001 € x 4 OCPU x 31 Tage x 24 Stunden = 2083,4976 €
 - Backup Storage: 0,0221 € x 10.000 GB + 0,0369 € = 221,0369 €
 - Monatliche Kosten: 2.838,4049 €

Lösung 2: OCI Bare Metal X7 Variante BM.DenseIO1.36

Diese Variante bietet eine dedizierte Maschine mit der folgenden Ausstattung:

- OCPU: 36
- Speicher: 512 GB
- Lokaler Datenträger: 28,8 TB NVMe SSD

Wir berechnen die Maschine mit vier aktivierten OCPUs.

Der Preis setzt sich aus den folgenden Positionen zusammen:

Shape	Anzahl CPU	RAM (GB)	DB Storage (TB)	DB Package	Monatliche Kosten (PAYG)
VM.Standard2.4	4	60	10	Standard	1.796,8049 €
VM.Standard2.4	4	60	10	Enterprise	2.838,4049 €
BM.DenseIO1.36	4	512	10	Standard	7.683,2825 €
BM.DenseIO1.36	4	512	10	Enterprise	8.724,9569 €

Tabelle 4: Szenario 2 – Lösungen im Überblick

	VM.Standard.2.4	BM.DenseIO1.36
DB-Versionen	11.2, 12.1, 12.2, 18: +++	11.2, 12.1, 12.2, 18: +++
Verfügbarkeit	Data Guard möglich: +	Data Guard möglich: +
DB-Optionen	Standard/Enterprise: +++	Standard/Enterprise: +++
Hardware-Ausstattung	+	+++
Storage-Intelligenz	-	-
DB-Performance	+	+++
Storage-Größe	+	+
Verwaltungsaufwand	+	+
Kosten	+++	+

Tabelle 5: Direkter Vergleich der Lösungen für Szenario 2

- Der Preis für die BM.DenseIO1.36 beträgt in der PAYG-Variante 9,3297 € pro Stunde für die Standard Edition und 10,0298 € für die Enterprise Edition [3]. Dabei sind standardmäßig zwei OCPUs bereits im Preis inklusive.
- Der Preis für eine zusätzliche OCPU beträgt pro Stunde für die Standard Edition 0,3501 € und für die Enterprise Edition 0,7001 €.

Für die Datenbanksicherungen nehmen wir 10 Terabyte Object Storage als Grundlage [5]:

- Object Storage – Storage (1 GB pro Monat): 0,0221 €
- Object Storage – Requests (10.000 Requests pro Monat): 0,0369 €

Die Berechnung von monatlichen Kosten bei der Nutzung des BM.DenseIO1.36 Shape mit 4 OCPUs setzt sich folgendermaßen zusammen:

- Variante mit dem Standard Package:
 - Server BM.DenseIO1.36 mit zwei OCPUs: $9,3297 \text{ €} \times 31 \text{ Tage} \times 24 \text{ Stunden} = 6.941,2968 \text{ €}$
 - Zwei zusätzliche OCPUs: $0,3501 \text{ €} \times 2 \text{ OCPUs} \times 31 \text{ Tage} \times 24 \text{ Stunden} = 520,9488 \text{ €}$
 - Backup Storage: $0,0221 \text{ €} \times 10000 \text{ GB} + 0,0369 \text{ €} = 221,0369 \text{ €}$
 - Monatliche Kosten: 7.683,2825 €
- Variante mit dem Enterprise Package:
 - Server BM.DenseIO1.36 mit 2 OCPUs: $10,0298 \text{ €} \times 31 \text{ Tage} \times 24 \text{ Stunden} = 7462,1712 \text{ €}$
 - Zwei zusätzliche OCPUs: $0,7001 \text{ €} \times 2 \text{ OCPUs} \times 31 \text{ Tage} \times 24 \text{ Stunden} = 1041,7488 \text{ €}$
 - Backup Storage: $0,0221 \text{ €} \times 10.000 \text{ GB} + 0,0369 \text{ €} = 221,0369 \text{ €}$
 - Monatliche Kosten: 8.724,9569 €

Hinweis: Die Absicherung der Datenbank mittels Oracle Data Guard ist in diesem Beispiel nicht berücksichtigt. Das Aktivieren des Data Guard kann bei der Enterprise Edition durch Aktivierung der zweiten Instanz zu erheblichen Kostensteigerungen führen.

Fazit

Lösung 1 ist aus Kostengründen sehr attraktiv. Lösung 2 bringt jedoch einige Vor-

teile, die für die Auswahl unter Umständen entscheidend sein können, wie

- mehr RAM für speicherintensive Datenbankanwendungen,
- sehr performanter NVME Storage, der mehr als 200.000 Transaktionen pro Sekunde abarbeitet (Quelle [6]),
- Reservierung der gesamten Leistung auf dem dedizierten Server für einen Anwender. (Bei Lösung 1 werden die physikalischen Ressourcen mit anderen virtuellen Maschinen geteilt, was zu Performance-Einbußen führen kann.)

Szenario 3: Unternehmenskritische Datenbanken

(siehe auch Tabelle 6 und 7)

Anforderungen:

- Verwendung: Unternehmenskritische produktive OLTP/DWH-Datenbanken mit hohen IO-Anforderungen
- Betriebszeiten: 24x7-Betrieb
- Hochverfügbarkeitsanforderungen: hoch
- Datenbankausstattung: extreme Performance (RAC, Active Data Guard, In-Memory)
- Datenbankressourcen:
 - CPU: ab 22
 - RAM: ab 512 GB
 - Datenbank-Storage: ab 40 TB

Lösung 1: 2-Knoten-RAC-System mit dem DB Shape VM.Standard.2.24.

Bei dieser Lösung handelt es sich um zwei VMs mit folgender Ausstattung:

- OCPUs: 2 bis 24
- RAM: 320 GB
- Storage (Block): bis zu 40 TB

In unserem Beispiel werden beide VMs mit zehn aktivierten OCPUs berechnet. Als Datenbank-Edition wird die Enterprise Edition Extreme Performance Package ausgewählt, die die Konfiguration einer RAC-Umgebung ermöglicht. Als Datenbank-Storage werden 40 Terabyte Block Storage konfiguriert.

Die Preise für die in der Lösung benötigten Positionen berechnen sich folgendermaßen [4], [5]:

- Compute (pro Stunde und OCPU): 0,0554 €
- Block Storage Classic High I/O (1 GB pro Monat): 0,0738 €
- Database Enterprise Edition High Performance (pro Stunde und OCPU): 2,188 €

Für die Datenbanksicherungen gehen wir von 40 Terabyte Object Storage aus [5]:

- Object Storage – Storage (1 GB pro Monat): 0,0221 €
- Object Storage – Requests (10000 Requests pro Monat): 0,0369 €

Die monatlichen Kosten werden folgendermaßen berechnet:

- Compute-Einheit: $0,0554 \text{ €} \times 10 \text{ OCPU} \times 31 \text{ Tage} \times 24 \text{ Stunden} = 412,176 \text{ €}$ (Preis für 2 VMs = 824,352 €)
- Database Enterprise Edition High Performance: $2,188 \text{ €} \times 10 \text{ OCPU} \times 31 \text{ Tage} \times 24 \text{ Stunden} = 16.278,72 \text{ €}$
- Preis für 2 VMs = 32.557,44 €
- Block Storage (High IO): 40 TB = 3.023 €
- Backup Storage: $0,0221 \text{ €} \times 40.000 \text{ GB} + 0,0369 \text{ €} = 884,0369 \text{ €}$
- Monatliche Gesamtkosten: 37.288,8289 €

Lösung 2: Exadata Cloud Service Quarter Rack X7

Bei dieser Lösung handelt es sich um eine Exadata X7 Quarter Rack Machine in der Cloud, die folgende Eigenschaften aufweist:

- 2 DB Server, 3 Storage Server
- OCPU: 20 aktiviert (10 pro Datenbank-server)
- RAM: 720 GB pro DB Server
- Storage: 360 TB Raw-Speicher oder 106 TB nutzbarer Speicher

Die Preise für die in der Lösung benötigten Positionen werden folgendermaßen berechnet [7]:

- Oracle Cloud Infrastructure, Database Exadata Infrastructure, Quarter Rack, X7: 35,008 € pro Stunde
- Exadata Cloud Service, Additional OCPUs: 2,188 € pro Stunde und OCPU

Für die Datenbanksicherungen berechnen wir 40 Terabyte Object Storage [5]:

Cloud Service	Server	OCPUs	RAM	Storage row (TB)	Monatliche Kosten (PAYG)
VM.Standard.2.24	2 DB	20	320 x 2	40	37.288,83 €
Exadata Quarter Rack X7	2 DB / 3 Storages	20	720 x 2	360	59.487,04 €

Tabelle 6: Szenario 3 – Lösungen im Überblick

- Object Storage – Storage (1 GB pro Monat): 0,0221 €
- Object Storage – Requests (10.000 Requests pro Monat): 0,0369 €

Die monatlichen Kosten werden folgendermaßen berechnet:

- Oracle Cloud Infrastructure, Database Exadata Infrastructure, Quarter Rack X7: 35,008 € x 31 Tage x 24 Stunden = 26.046 €
- Exadata Cloud Service, Additional OCPUs: 2,188 € x 20 OCPUs x 31 Tage x 24 Stunden = 32.557,44 €
- Backup Storage: 0,0221 € x 40.000 GB + 0,0369 € = 884,0369 €
- Monatliche Gesamtkosten: 59.487,0369 €

Hinweis: Die Absicherung der Datenbank mit Oracle Data Guard ist in diesem Beispiel nicht berücksichtigt. Das Aktivieren des Data Guard durch Aktivierung der zweiten Instanz kann zu erheblichen Kostensteigerungen führen.

Fazit

Exadata Cloud Service kommt zwangsläufig bei Datenbanken zum Einsatz, wenn folgende Anforderungen im Raum stehen:

- Datenbankgröße von mehr als 40 TB
- Sehr hohes I/O-Aufkommen bzw. -Durchsatz
- Die Datenbank profitiert von den Eigenschaften der Exadata
- Mehrere Datenbanken sollen auf einer Cloud-Plattform konsolidiert werden

Resümee

In diesem Artikel haben wir einige Optionen und Szenarios beschrieben und rechnerisch durchexerziert. Dabei handelt es sich nur um einzelne Fälle mit ausgewählten Optionen. Die Aufzählung sämtlicher Möglichkeiten hätte den Rahmen

	VM.Standard.2.24	Exadata Quarter Rack X6
DB-Versionen	11.2, 12.1, 12.2, 18: +++	11.2, 12.1, 12.2, 18: +++
Verfügbarkeit	RAC: ++	RAC: ++
DB-Optionen	Extreme Edition: +++	Extreme Edition: +++
Hardware-Ausstattung	+	+++
Storage-Intelligenz	-	Smart Scans, Storage-Indexierung: +++
DB-Performance	+	+++
Storage-Größe	+	+++
Verwaltungsaufwand	++	+
Kosten	++	+

Tabelle 7: Direkter Vergleich der Lösungen für Szenario 3

des Artikels gesprengt. Entscheidend bei der Auswahl einer Umgebung sind erfahrungsgemäß Fragen wie: Wie viel Storage steht für Daten nach Abzug der Redundanz tatsächlich zur Verfügung? Sollen die Datenbanksicherungen vielleicht zuerst lokal gespeichert werden? Welche Rolle spielt bei meiner Planung hinsichtlich des Verfügbarkeitslevels die geographische Absicherung mittels Data Guard zwischen unterschiedlichen Verfügbarkeitszonen? Sollte ich eventuell in der Oracle Cloud RAC-Datenbanken via Data Guard replizieren? Und last but not least sollten Sie für die Kostenoptimierung das Dauerauftragsmodell „Monthly Flex“ prüfen und klären, ob Oracle eventuell Rabatte oder Aktionspreise für einen bestimmten Dienst anbietet.

Sind diese Fragen geklärt, fällt die Auswahl einer passenden Datenbankumgebung in der Cloud meist sehr leicht. Wie sind Ihre Erfahrungen? Hilft Ihnen dieser Artikel weiter? Schreiben Sie mir gerne. Ich freue mich auf Ihr Feedback!

[6] <https://cloud.oracle.com/database/faq>
 [7] <https://cloud.oracle.com/database/exadata/pricing>

Quellen

[1] https://cloud.oracle.com/de_DE/atp
 [2] https://cloud.oracle.com/de_DE/compute-classic/pricing
 [3] https://cloud.oracle.com/de_DE/database/pricing
 [4] https://cloud.oracle.com/de_DE/compute/pricing
 [5] <https://cloud.oracle.com/storage/pricing>



Borys Neselovskyi
 borys.neselovskyi@opitz-consulting.com

Finden Sie die passende Schulung im Oracle-Umfeld auf

university.doag.org

- ▶ Oracle-Technologien
- ▶ IT-Methoden
- ▶ IT-Management



Erhalten Sie als
DOAG-Mitglied einen
exklusiven Rabatt auf
den regulären
Kurspreis.



Datentransfer mit Oracle Tools – Was ist möglich?

Christian Gohmann, Trivadis GmbH

Zu den Aufgabengebieten eines DBA gehört neben der klassischen Wartung einer Datenbank auch der Datenaustausch zwischen unterschiedlichen, zum Teil heterogenen Systemen. Die Redewendung „Viele Wege führen nach Rom“ trifft es hier sehr gut. Für das Laden von externen Daten, wie zum Beispiel CSV-Dateien, bietet Oracle mehrere Alternativen an. Die zur Verfügung stehenden Werkzeuge werden entweder mit der Oracle-Datenbank-Installation mitgeliefert oder stehen kostenfrei auf den Oracle-Webseiten zum Download bereit. In diesem Artikel schauen wir uns an, welche Werkzeuge Oracle uns für diese Aufgabe an die Hand gibt.

Zeichenkodierung

Der Zeichensatz einer Datenbank legt fest, welche Zeichen gespeichert werden können. Deswegen findet in der heutigen Zeit immer mehr der Unicode-Zeichensatz Anwendung. Damit die Zeichen zwischen Client und der Datenbank korrekt umgewandelt und somit korrekt in der Datenbank gespeichert werden, muss Oracle wissen, welchen Zeichensatz der

Client verwendet. Dies wird über die Umgebungsvariable beziehungsweise bei Windows alternativ über den Registry-Eintrag `NLS_LANG` realisiert.

Bei der Verarbeitung externer Dateien muss darauf geachtet werden, mit welchem Zeichensatz die Eingabedateien kodiert wurden. Weicht der dort verwendete Zeichensatz von dem des Clients ab, so muss die `NLS_LANG` entsprechend angepasst werden. In diesem Fall muss

die Variable so gesetzt werden, dass der Zeichensatz mit der Kodierung der Eingabedateien übereinstimmt (siehe Listing 1).

Data Pump

Wenn man über den Datentransfer zwischen Oracle-Datenbanken spricht, kommt man in der Regel an Data Pump nur schwer vorbei. Der Austausch zwi-

```
$> file input.csv
UTF-8 Unicode text
$> export NLS_LANG=GERMAN_GERMANY.AL32UTF8
```

Listing 1: Korrektes Setzen der NLS_LANG-Umgebungsvariable

```
SQL> CREATE VIEW EMP_DEPT10 AS
SELECT empno, ename, job, mgr, deptno FROM EMP
WHERE deptno = 10;

SQL> SELECT * FROM emp_dept10;
  EMPNO  ENAME      JOB          MGR          DEPTNO
-----  -
  7782   CLARK      MANAGER      7839          10
  7839   KONG       PRESIDENT    7839          10
  7934   MILLER     CLERK        7782          10

$> expdp scott/tiger DIRECTORY=DATA_PUMP_DIR VIEWS_AS_TABLES=EMP_DEPT10
```

Listing 2: Export einer View als Tabelle mithilfe von Data Pump

schen heterogenen Systemen ist dabei allerdings nicht möglich, da die Daten intern in dem proprietären Stream-Format des Direct Path API gespeichert werden. Der Funktionsumfang und eine detaillierte Beschreibung würden den Rahmen dieses Artikels sprengen. Entsprechend werden wir nur auf wenige Funktionen eingehen, die die Flexibilität und die Möglichkeiten von Data Pump demonstrieren.

Data Pump wurde mit Oracle 10g eingeführt und ist die Ablöse für das alte Export/Import Utility. Im Gegensatz zu seinem Vorgänger läuft Data Pump als Serverprozess. Der Grund dafür ist, dass Data Pump vollständig in PL/SQL (DBMS_DATAPUMP) geschrieben wurde und die beiden Kommandozeilenwerkzeuge expdp und impdp lediglich für die Interaktion mit dem Data Pump API verwendet werden.

Data Pump unterstützt folgende Modi, die auf unterschiedlichen Objektebenen agieren:

- Full
- Tablespace
- Transportable Tablespace
- Schema
- Table

Bei dem Export einer Tabelle war es in der Vergangenheit nicht möglich, einzelne Spalten zu entfernen. Seit Oracle 12c Release 1 kann dies über den Umweg einer View und den Parameter VIEWS_AS_TABLES erfolgen. Hierbei wird eine View definiert, die dann als eigenständige Ta-

belle exportiert werden kann. Dadurch ist es möglich, die Ergebnismenge einer (komplexen) SQL-Abfrage zu exportieren. Um die Datenmenge zu reduzieren, kann der QUERY-Parameter verwendet werden, der entweder eine globale oder eine auf bestimmte Tabellen beschränkte WHERE-Klausel definiert.

Listing 2 zeigt einen Export der View EMP_DEPT10, beschränkt auf Mitarbeiter der Abteilung 10.

Beim Datenimport spielt Data Pump seine Stärken aus. Metadaten werden als

```
SQL> CREATE OR REPLACE PACKAGE DATA_TRANSFORM IS
  FUNCTION HIDE_NAME (name IN VARCHAR2) RETURN VARCHAR2;
END DATA_TRANSFORM;
/

SQL> CREATE OR REPLACE PACKAGE BODY DATA_TRANSFORM IS
  FUNCTION HIDE_NAME (name IN VARCHAR2) RETURN VARCHAR2 IS
  BEGIN
    RETURN SUBSTR(name, 0, 1) || '.';
  END HIDE_NAME;
END DATA_TRANSFORM;
/

$> impdp scott/tiger DIRECTORY=DATA_PUMP_DIR TABLES=EMP_DEPT10 REM-
AP_TABLE=EMP_DEPT10:EMP_DEPT10_ANON REMAP_DATA=EMP_DEPT10.ENAME:DATA_
TRANSFORM.HIDE_NAME

SQL> SELECT * FROM EMP_DEPT10_ANON;
  EMPNO  ENAME      JOB          MGR          DEPTNO
-----  -
  7782   C.         MANAGER      7839          10
  7839   K.         PRESIDENT    7839          10
  7934   M.         CLERK        7782          10
```

Listing 3: Anonymisierung von Daten während eines Data-Pump-Imports

XML-Daten in dem Dumpfile gespeichert, die bei einem Import mittels XSLT verändert werden können. Es kann zum Beispiel ein Schemawechsel durch REMAP_SCHEMA-Parameter erfolgen. Ist eine Datenanonymisierung notwendig, so kann dies mithilfe des REMAP_DATA-Parameters erfolgen. Hierbei wird eine PL/SQL-Funktion aus einem Package (zwingend erforderlich) aufgerufen, die die Daten nach Belieben verändert. In Listing 3 wird die Funktion HIDE_NAME verwendet, um die Nachnamen auf den ersten Buchstaben zu reduzieren.

Für weitere Informationen empfiehlt sich ein Blick in den Oracle Database Utilities Guide [1].

SQL*Loader

Wie der Name bereits vermuten lässt, besteht die Hauptaufgabe des SQL*Loader darin, Daten aus externen Dateien in die Datenbank zu laden. Aufgrund der sehr flexiblen Parsing Engine können nahezu beliebig formatierte Dateien geladen werden.

Die Konfiguration wird in der Regel durch ein SQL*Loader Control File definiert. Bei einfacheren Ladevorgängen kann ab Oracle 12c Release 1 auch der SQL*Loader Express Mode verwendet werden, bei dem eine als CSV formatierte Datei direkt in die dazugehörige Tabelle

geladen wird. Dabei muss der Name der Eingabedatei dem Namen der Zieltabelle entsprechen. Listing 4 demonstriert dies anhand der Tabelle EMP.

Sollte die Eingabedatei in einem Verzeichnis liegen, für das kein logisches DIRECTORY-Objekt in der Datenbank existiert, versucht SQL*Loader dieses anzulegen. Dies funktioniert jedoch nur, wenn der Benutzer das Privileg CREATE ANY DIRECTORY besitzt.

Listing 5 zeigt ein SQL*Loader Control File eines komplexeren Ladevorganges für die Tabelle EMP. Dabei werden die folgenden Dinge durchgeführt.

- Einlesen mehrerer Eingabedateien durch Angabe einer Wildcard
- Filterung auf die Abteilungsnummer 10
- Transformation der Spalten-ID (Erhöhung der ID um 1000) und ENAME (Umwandlung in Großschreibung)

```
$> vi EMP.dat
$> sqlldr scott/tiger TABLE=EMP
```

Listing 4: SQL*Loader Express Mode

```
$> vi emp.ctl
LOAD DATA
INFILE 'emp*.csv'
APPEND
INTO TABLE EMP
WHEN deptno = '10'
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
(
  EMPNO ":EMPNO + 1000",
  ENAME "UPPER(:ename)",
  JOB,
  MGR,
  HIREDATE DATE "dd.mm.yyyy",
  SAL,
  COMM,
  DEPTNO
)
$> sqlldr scott/tiger CONTROL=emp.ctl
```

Listing 5: Komplexes SQL*Loader-Beispiel

SQL Developer

Der SQL Developer, der über das Oracle Technology Network bezogen werden kann [2], bietet eine Vielzahl von Funktionen, um mit einer Oracle-Datenbank zu

interagieren. Hierzu zählt auch die Möglichkeit, das Ergebnis einer beliebigen Abfrage über den Query-Result-Reiter und dessen Kontextmenüeintrag Export zu

exportieren. Für den Export stehen zahlreiche unterschiedliche Ausgabeformate zur Verfügung, wie zum Beispiel CSV, JSON, XML sowie Microsoft Excel.

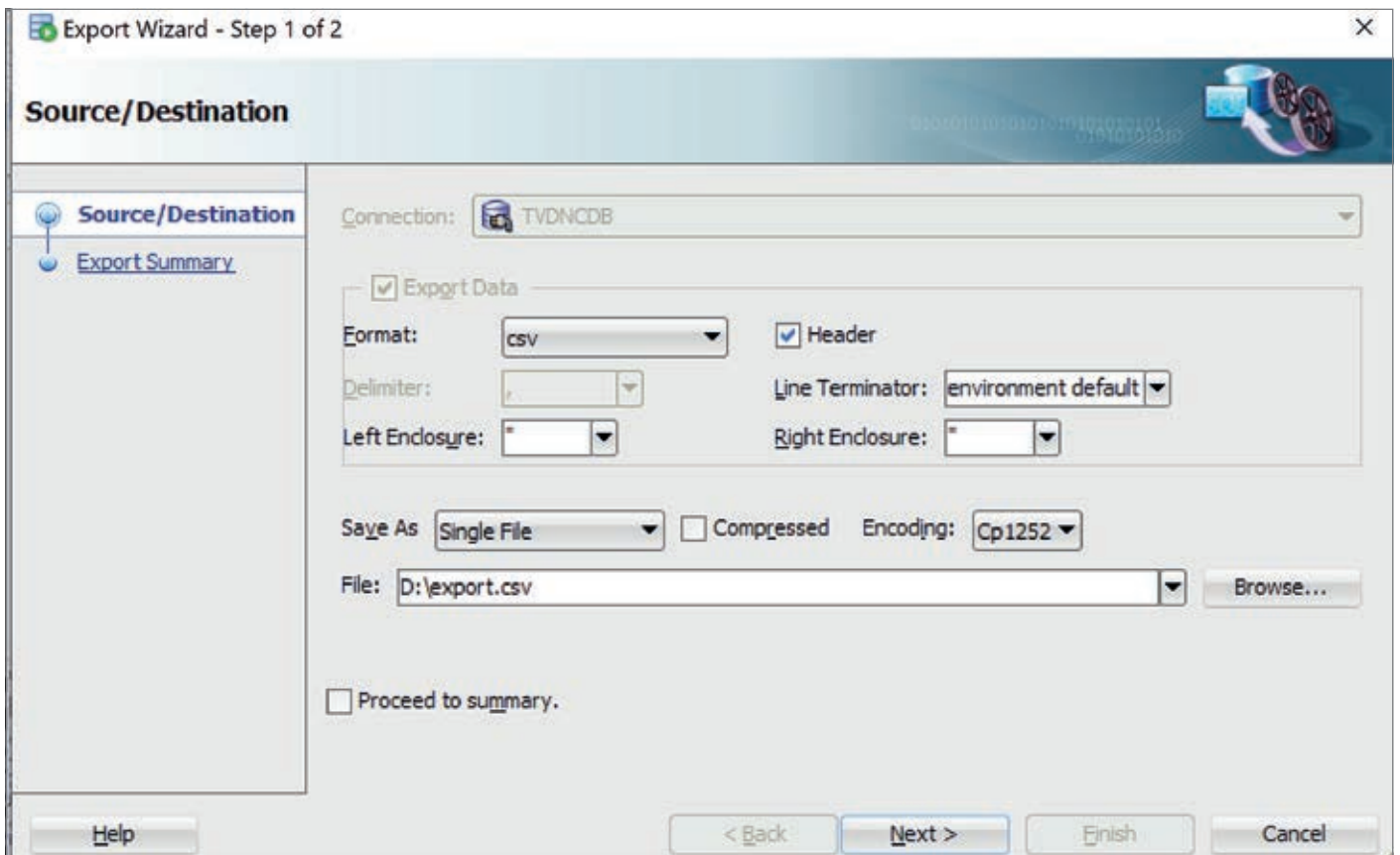


Abbildung 1: Export Wizard des SQL Developer (Quelle: Christian Gohmann)

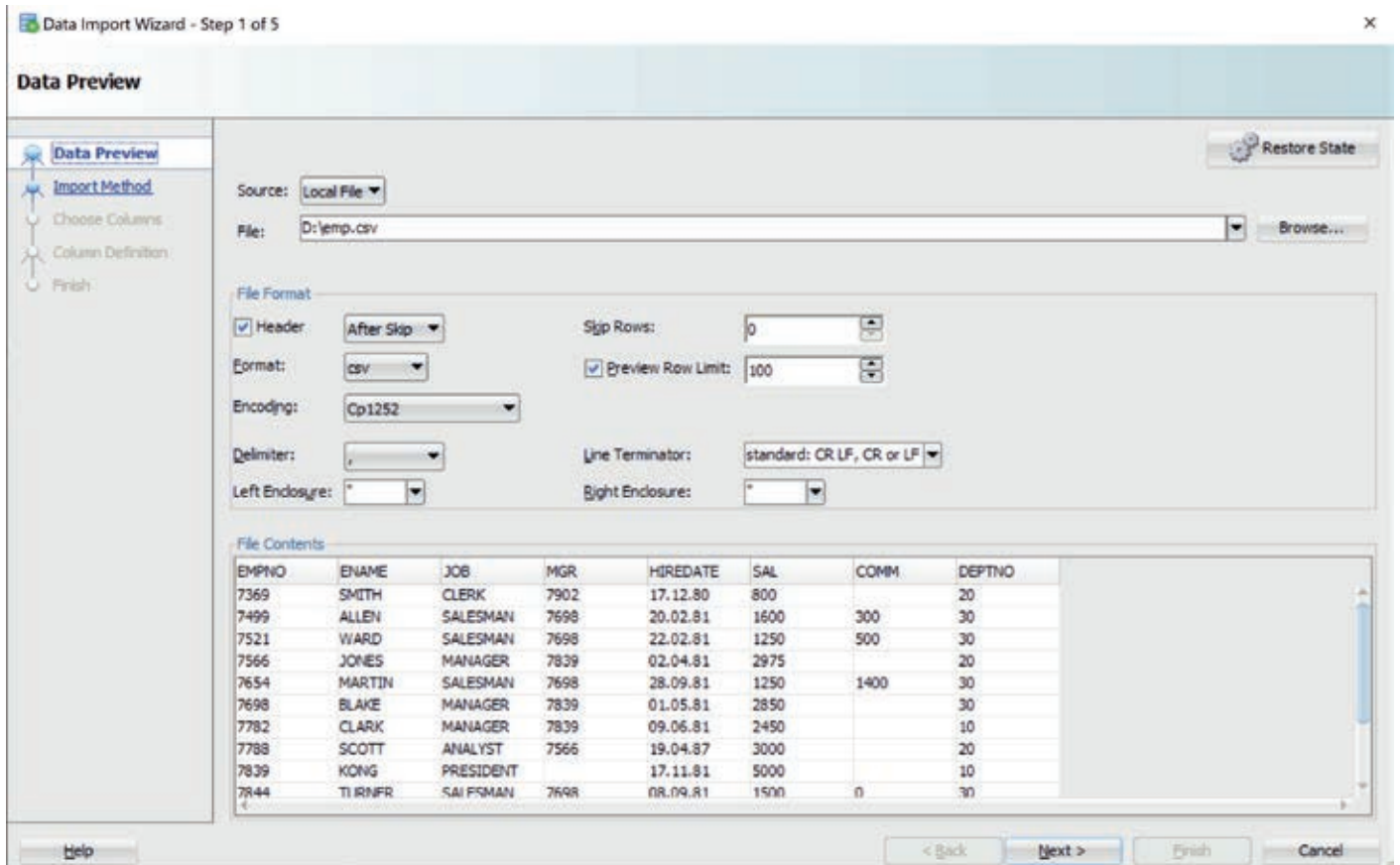


Abbildung 2: Data Import Wizard des SQL Developer (Quelle: Christian Gohmann)

Abbildung 1 zeigt den Export der Tabelle EMP als CSV-Datei.

Neben dem Export von Daten bietet der SQL Developer die Möglichkeit, Daten aus externen Dateien in eine bestehende Tabelle zu laden. Hierfür wählt man die gewünschte Tabelle aus und ruft über das Kontextmenü den Punkt Import Data auf.

Der Import-Assistent bietet ein breites Spektrum an Funktionen an, wie zum Beispiel das Verwenden unterschiedlicher Import-Methoden, das Abwählen nicht benötigter Spalten oder das Mappen der Eingabespalten auf die Spalten der Zieltabelle. Ein weiterer Vorteil ist die Vorschau

der Daten auf Basis der getätigten Einstellungen. Dadurch sieht man schnell, ob Umlaute oder Sonderzeichen korrekt dargestellt werden. Ist dies nicht der Fall, muss das Encoding gegebenenfalls geändert werden. *Abbildung 2* zeigt einen CSV-Import in die Tabelle EMP.

CSV-Dateien mit SQL*Plus / SQLcl generieren

SQL*Plus wurde über die Jahre hinweg um zusätzliche Funktionen erweitert. Mit der Version 12c Release 2 kam die Unter-

stützung für die Darstellung einer Ergebnismenge als Comma Separated Values (CSV) dazu. Damit ist es möglich, einen CSV-Export einer oder mehrerer Tabellen direkt aus SQL*Plus heraus zu erstellen. Zum Aktivieren dieser Funktionalität muss das Kommando SET MARKUP CSV verwendet werden. Über weitere optionale Parameter kann das Trennzeichen (DELIMITER) festgelegt werden oder ob Zeichenketten mit Anführungszeichen umschlossen werden sollen (QUOTE). *Listing 6* zeigt die Verwendung anhand einer Abfrage gegen die EMP-Tabelle.

Wird ein Trennzeichen benutzt, das in Spalten verwendet wird, so wird dies nicht mit einem Escape-Zeichen versehen und führt entsprechend zu einer falsch formatierten CSV-Ausgabe. Folgende COLUMN-Kommandos können zur weiteren Formatierung verwendet werden.

- COLUMN FORMAT
- COLUMN HEADING
- COLUMN NULL

```
SQL> SET MARKUP CSV ON DELIMITER ";" QUOTE ON
SQL> SELECT * FROM scott.emp WHERE ename = 'SCOTT';

"EMPNO";"ENAME";"JOB";"MGR";"HIREDATE";"SAL";"COMM";"DEPTNO"
7788;"SCOTT";"ANALYST";7566;"19-APR-87";3000;;20
```

Listing 6: CSV-Ausgabe in SQL*Plus

```
$> sqlplus -S -M „CSV ON“ scott/tiger @select_data.sql
```

Listing 7: Aktivierung CSV-Ausgabe für SQL*Plus

SQL*Plus bietet die Möglichkeit an, direkt beim Start die CSV-Ausgabe zu aktivieren, (siehe *Listing 7*). Dies ist vor allem hilfreich,

wenn die Generierung von CSV-Dateien automatisiert werden soll.

SQL Developer bzw. SQLcl bieten ein weitaus größeres Spektrum unterschiedlicher Ausgabenformate, von zum Beispiel XML, JSON bis hin zu klassischen INSERT-INTO-Befehlen. Das gewünschte Ausgabeformat kann entweder über das Kommando SET SQLFORMAT aktiviert oder durch die Angabe eines Kommentars (z.B. /* csv */) bei der Ausführung der SQL-Anweisung aktiviert werden.

External Tables

Oracle bietet mit den External Tables einen speziellen Objekttyp an, der es ermöglicht, auf Daten außerhalb der Datenbank lesend zuzugreifen.

Die Daten liegen dabei in einfachen Textdateien, die nahezu beliebig formatiert sein können. Im Data Dictionary werden lediglich Meta-Informationen über die Tabelle abgelegt. Eine External Table kann wie eine herkömmliche Tabelle in SQL-Anweisungen verwendet werden.

Der Zugriff auf die Daten erfolgt über einen der folgenden sogenannten Access Drivers, die unterschiedliche Funktionalitäten bieten.

- ORACLE_LOADER (Standard)
- ORACLE_DATAPUMP
- ORACLE_HIVE (12.2)
- ORACLE_HDFS (12.2)

Der Access Driver ORACLE_LOADER wird in *Listing 8* verwendet, um die als CSV-Dateien gespeicherten Kundendaten der Datenbank zugänglich zu machen. Dieser Access Driver bietet eine Untermenge der Funktionalitäten des SQL*Loader, inklusive dessen Data-Mapping-Möglichkeiten.

Die PREPROCESSOR-Anweisung erlaubt die Ausführung eines Betriebssystem-Skriptes. Damit ist es zum Beispiel möglich, eine gezippte Datei automatisch zu entpacken, bevor auf die Daten zugegriffen wird.

Die Unterstützung von partitionierten External Tables kam mit Oracle 12c Release 2 dazu. Dabei werden mehrere Dateien zugrunde gelegt, die dann als Partitionen deklariert werden. Ob der Inhalt der Dateien dem gewählten Partitioning Key entspricht, wird dabei nicht geprüft –

```
SQL> CREATE DIRECTORY ext_data_dir AS '/data';
SQL> CREATE TABLE emp_xt4 (
  empno NUMBER(4),
  ename VARCHAR2(10),
  deptno NUMBER(2)
) ORGANIZATION EXTERNAL (
  TYPE ORACLE_LOADER
  DEFAULT DIRECTORY ext_data_dir
  ACCESS PARAMETERS (
    RECORDS DELIMITED BY NEWLINE
    PREPROCESSOR 'unzip_emp.sh'
    FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
  )
) REJECT LIMIT UNLIMITED
PARTITION BY LIST (deptno)
(PARTITION emp_dept10 VALUES (,10) LOCATION ('emp_deptno10.csv'),
PARTITION emp_dept20 VALUES (,20) LOCATION ('emp_deptno20.csv'));
```

Listing 8: Beispiel einer partitionierten External Table

```
SQL> CREATE TABLE emp_xt ORGANIZATION EXTERNAL
(TYPE ORACLE_DATAPUMP DEFAULT DIRECTORY data_dir LOCATION ('ext_xt.dmp'))
AS SELECT empno, ename FROM emp WHERE deptno = 10;

SQL> CREATE TABLE emp_deptno10_xt (empno NUMBER(4), ename VARCHAR2(10))
ORGANIZATION EXTERNAL
(TYPE ORACLE_DATAPUMP DEFAULT DIRECTORY data_dir LOCATION ('ext_xt.dmp'));
```

Listing 9: Entladen und Laden von Daten

es liegt somit in den Händen des DBA, die Datenkonsistenz sicherzustellen.

Mithilfe des ORACLE_DATAPUMP Access Driver kann die Ergebnismenge einer SQL-Anweisung in eine externe Datei geschrieben werden. Die erzeugte Datei kann dann wiederum als External Table eingebunden werden. *Listing 9* zeigt das Entladen und Laden der Tabelle EMP.

Auch wenn der Name gegebenenfalls etwas anderes suggeriert, kann die generierte Dump-Datei nicht von Data Pump gelesen werden (ORA-39323).

Ab Oracle 18c ist es nicht mehr zwingend erforderlich, ein Objekt zu erzeugen. Stattdessen kann eine Inline External Table verwendet werden, die in der FROM-Klausel einer SQL-Anweisung definiert wird.

Quellen

- [1] Oracle Database Utilities 19c: <https://docs.oracle.com/en/database/oracle/oracle-database/19/sutil/index.html>
- [2] SQL Developer Produktseite: <https://www.oracle.com/database/technologies/appdev/sql-developer.html>

Über den Autor

Christian Gohmann ist Senior Consultant und Trainer bei der Trivadis GmbH in Düsseldorf. Seine Schwerpunkte liegen im Bereich Oracle und den Themengebieten Architektur, Implementation und Migration – spezialisiert auf Hochverfügbarkeitslösungen, wie zum Beispiel Oracle Real Application Cluster oder Data Guard.

Neben seinen Consulting-Tätigkeiten ist Christian Gohmann hauptverantwortlicher Entwickler des von Trivadis vertriebenen Produktes TVD-Backup, das ein intelligentes Backup und Recovery von Oracle-Datenbanken ermöglicht.



Christian Gohmann
christian.gohmann@trivadis.com



Klick, Klick, Test – Automatisierte Tests mit APEX

Kai Donato, MT AG

Heutzutage müssen wir als Entwickler nicht nur qualitativ hochwertigen Programmcode entwickeln, sondern auch für eine ausreichende Testabdeckung sorgen. Hierzu gibt es eine Vielzahl von Möglichkeiten, die alle ihre Vor-, aber leider auch Nachteile haben. Dieser Artikel beschreibt einen neuen Ansatz, der es uns möglich macht, zum einen schnell erfolgreich zu testen und zum anderen alle Stärken des Oracle Application Express (APEX) zu instrumentalisieren.

Status Quo

Viele Entwickler beantworten die Frage nach der aktuell eingesetzten Teststrategie mit einem Satz in etwa wie: „Der Fachbereich bekommt den Zugriff und gibt uns dann ein ‚okay‘ oder ‚nicht okay‘“. Im Gegensatz zu dem reinen Abnahmetest durch den Fachbereich gibt es in hochkritischen Anwendungen umfangreiche Test-Suites, die in mühevoller Arbeit entwickelt wurden. Häufig wird dabei auf existierende Frameworks

zurückgegriffen, die mit ihrer eigenen Sprache und natürlich auch mit Vor- und Nachteilen daherkommen. Häufig hören wir an dieser Stelle Produktnamen wie Selenium, Ranorex, Cucumber oder Katalon. Unbestritten kann man sagen, dass es sich bei den genannten Produkten um sehr mächtige Werkzeuge handelt, denen es aber genau an einer Sache mangelt: dem direkten Bezug zum Oracle Application Express.

Die Metadaten, anhand derer eine APEX-Anwendung aufgebaut wird, befin-

den sich, sauber strukturiert, in der Datenbank und stellen eigentlich eine hervorragende Basis für unsere Testkriterien dar. Dazu später mehr.

Was ist die Grundlage für erfolgreiche Tests?

In einem Lastenheft oder einer Story im Backlog eines Scrum-Projekts steht im Regelfall immer eine Textpassage, die be-

schreibt, welcher Sachverhalt erfüllt werden muss, damit eine Funktion oder ein Feature als erfolgreich entwickelt gilt. Dazu gehört unter anderem auch eine Beschreibung dessen, was auf der Oberfläche der Anwendung zu sehen beziehungsweise nicht zu sehen ist. Ebenso wird beschrieben, was passieren soll, wenn eine Aktion vorgenommen wurde.

Dies stellt dann die Grundlage für die Entwicklung eines Features, aber auch für die Entwicklung von Testszenarien dar. Der Entwickler plant, setzt das Angeforderte um und führt zunächst einen „Entwickler-Smoke-Test“ durch, der im Groben die Funktionalität des Entwickelten überprüft. Kurz gesagt: Das Artefakt wird entsprechend der Interpretation der Story vom Entwickler getestet und dann für weitere Tests freigegeben. Das, was nun noch fehlt, ist genau das, was ein „Projekt“ zu einem „erfolgreichen Projekt“ machen kann – automatisierte Tests, die im besten Fall regelmäßig ausgeführt werden und somit die Integrität und die Fehlerfreiheit einer Anwendung dauerhaft gewährleisten können.

Die beliebtesten Frameworks, mit denen heutzutage getestet wird

In den meisten Fällen setzen Unternehmen auf ein Framework, das bereits seit langer Zeit existiert – wie zum Beispiel Selenium. Selenium und Selenium Grid bieten ein weites Spektrum an unterstützten Browsern an, die vorher aufgenommene Tests ausführen können. Hierbei wird üblicherweise

ein Browser-Plug-in installiert, das Schritt für Schritt aufnimmt, welche Aktivitäten im Browser durchgeführt werden. Dies dient im Anschluss als Drehbuch für Tests, die dann automatisch ausgeführt werden.

In vielen Fällen steht auch kein Browser-Plug-in zur Verfügung, da das Test-Framework dies nicht unterstützt. In diesen Fällen werden manuell Tests programmiert, die mehr als nur zeitaufwendig sind. Wer in seinem Projekt für eine lückenlose Testabdeckung sorgen will, muss einen Teil der Entwicklungszeit aufschlagen, um Testfälle sorgfältig auszuprogrammieren. Neben diesem sehr bekannten und häufig genutzten Framework haben in den vergangenen Jahren immer wieder neue Frameworks das Licht der Welt erblickt, die alle ihre Daseinsberechtigung haben, jedoch nur begrenzt sinnvoll für den Einsatz mit APEX sind.

Was fehlt uns und was brauchen wir?

Die genannten Frameworks können alle genutzt werden, um Web- und Mobile-Anwendungen zu testen. Sie haben allerdings auch alle ihre Eigenheiten. Dazu gehört zum Beispiel die Syntax, die verwendet wird, um sequenzielle Abläufe zu programmieren. Im Regelfall muss man für jedes dieser Frameworks einen Aufwand betreiben, um es auf APEX zuzuschneiden. Häufig muss man dabei auch auf Funktionalitäten verzichten, die sich im Kontext von APEX nicht abbilden lassen. Dazu kommt, dass der Autor der Tests einen Sachverhalt inklusive aller daran beteiligten Komponenten niederschreibt und somit ein rela-

tiv starres Objekt ablegt. Um genau dieses „Starre“ aus unseren Tests herauszubekommen und vor allem von einer immensen Zeitersparnis zu profitieren, sollten wir uns eine der Stärken von APEX zunutze machen: die Metadaten!

Alle Informationen, die eine APEX-Anwendung ausmachen, befinden sich in der Datenbank; somit können wir beispielsweise einfach ermitteln, welche Komponenten sich auf einer APEX-Seite befinden, und diese für unsere Tests verwenden. Mithilfe von mehr oder weniger einfachen Select-Statements können wir alle nötigen Meta-Informationen ermitteln und für eine deklarative Erstellung von Testfällen zur Verfügung stellen.

Das Metadaten-Modell von APEX ist umfangreich und gut strukturiert, sodass wir stets auf den „Ist-Zustand“ einer Anwendung schauen.

Automatisiertes Ausführen von automatisiert erstellten Tests

So exotisch wie der Titel dieses Absatzes war auch die Idee, die hinter einem Tool steckt, das in diesem Artikel vorgestellt werden soll. Wir machen uns die Nachteile der bekannten Test-Frameworks und die Vorteile von APEX zunutze und entwickeln etwas Großes! Mit einer Handvoll neuer Technologien und vielen bekannten Faktoren vereinfachen wir die Schritte, die uns sonst viel Zeit kosten. Nach ersten Gehversuchen mit „Puppeteer“, dem Framework für automatisierte Tests im Google Chrome-Browser, und diversen Tests mit CodeceptJS haben wir etwas Neues geschaffen. Mithilfe von Node.js und dem Templating-

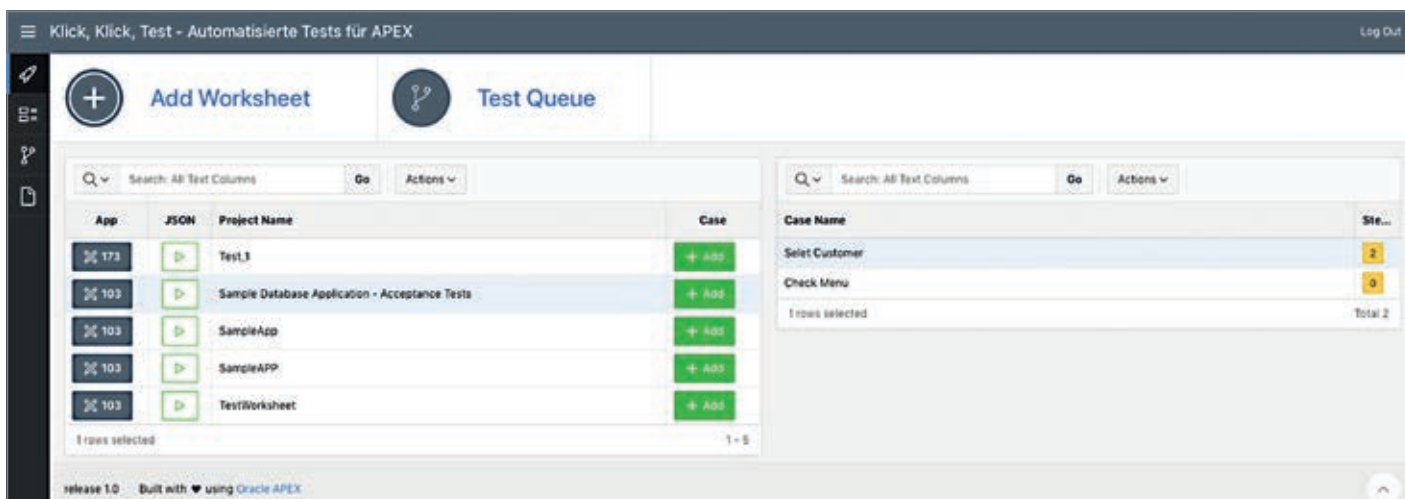


Abbildung 1: Die Übersicht (Quelle: Kai Donato)

Test Relevant	Item-Name	Item-Type	Item-Value	STATIC-ID	Action	Default-Value	Region Name	Data Type	Sort-Order
Yes	P45_PROJ_NAME	Forme-Item	Test Projekt	P45_PROJ_NAME	MouseClicked	-	Worksheet Details	VARCHAR2	1.
Yes	P45_PROJ_DESCR	Forme-Item	Projekt-Beschreibung	P45_PROJ_DESCR	Keyboard-Tab	-	Worksheet Details	VARCHAR2	2.
Yes	P45_APPLICATION_ID	Forme-Item	1000	P45_APPLICATION_ID			Worksheet Details	VARCHAR2	3.
Yes	P45_DOMAIN	Forme-Item	-	P45_DOMAIN		http://apex.oracle.com/ftp	Worksheet Details	VARCHAR2	4.
Yes	P45_LOGIN	Forme-Item	-	P45_LOGIN		1	Worksheet Details	VARCHAR2	5.
	P45_OWNER	Forme-Item	-	P45_OWNER		-	Worksheet Details	VARCHAR2	6.
	P45_PASSWORD	Forme-Item	-	P45_PASSWORD		-	Worksheet Details	VARCHAR2	7.
	P45_SCREENSHOT	Forme-Item	-	P45_SCREENSHOT		0	Worksheet Details	VARCHAR2	8.
	P45_PDF_OUTPUT	Forme-Item	-	P45_PDF_OUTPUT		0	Worksheet Details	VARCHAR2	9.
	P45_SIZE_X	Forme-Item	-	P45_SIZE_X		1440	Worksheet Details	NUMBER	10.
	P45_SIZE_Y	Forme-Item	-	P45_SIZE_Y		800	Worksheet Details	NUMBER	11.
	P45_EMULATE_DEVICE	Forme-Item	-	P45_EMULATE_DEVICE		-	Worksheet Details	VARCHAR2	12.
	P45_DELAY	Forme-Item	-	P45_DELAY		0	Worksheet Details	NUMBER	13.
	P45_HEADLESS	Forme-Item	-	P45_HEADLESS		0	Worksheet Details	VARCHAR2	14.
Yes	Cancel	Button	-	B130002479938602885	MouseClicked	-	Buttons	-	15.
	Delete	Button	-	B130001964793602885		-	Buttons	-	16.
	Apply Changes	Button	-	B130001874959602885		-	Buttons	-	17.
	Create	Button	-	B130001760092802885		-	Buttons	-	18.

Abbildung 2: Der Test-Designer (Quelle: Kai Donato)

Framework „Handlebars“ kann dynamisch ein Test-Code für User-Acceptance-Tests erzeugt werden, der wiederum in vielen Umgebungen unter verschiedenen Voraussetzungen ausgeführt werden kann. Um es dann noch abzurunden, kann mittels Docker für die angemessene Unterbringung der Testumgebung gesorgt werden.

Das Tool

Den Einstieg in die Anwendung, die uns das einfache Erstellen von Tests ermöglicht, erhalten wir über die Übersichtsseite, die den Aufbau der sogenannten Test-Worksheets verdeutlicht. Mithilfe eines Worksheets werden mehrere Test-Cases gruppiert und als eine Art Arbeitsmappe für eine zu testende Anwendung gesehen. Innerhalb eines Test-Case werden wiederum Test-Schritte (Steps) angeordnet (siehe Abbildung 1). Dies wird gleich näher erläutert, wenn wir einen Blick unter die Haube werfen.

Der Test-Designer

Mit dem Test-Designer sind wir in der Lage, alle Komponenten einer zu testenden APEX-Seite auf einen Blick zu sehen und für eine Anwendung auszuwählen. Hierzu wird, wie in Abbildung 2 zu sehen

ist, ausgewählt, welche Komponenten relevant für diesen Test sind. In der ersten Spalte dieses Grid wird hierzu „Test Relevant“ auf „Yes“ gesetzt. Hinzukommen werden zum einen weitere Meta-Daten für die jeweiligen Komponenten und zum anderen wird die Möglichkeit gegeben, Eingabefelder mit entsprechenden Daten zu versehen. So wählen wir in dem oben gezeigten Beispiel ein Textfeld aus, das den technischen Namen „P45_PROJ_NAME“ besitzt, und geben in das Feld „Item-Value“ einen Text-String ein, der innerhalb dieses Tests verwendet werden soll. In der weiteren Entwicklung ist vorgesehen, dass sich diese Felder auch mehrfach belegen lassen, sodass sich mehrfache Tests pro Komponente einfach erstellen lassen.

Da erfahrene Entwickler stets dazu raten, für jede Komponente eine sogenannte „Static ID“ zu vergeben, kann die Engine, die unsere Tests generiert, diese verwenden, um eine Komponente eindeutig zu identifizieren. Aber keine Angst – sollten Sie diese „Static ID“ nicht auf jeder Komponente hinterlegt haben, so wird im Test-Designer die jeweilige ID angezeigt, die APEX selbst vergeben hat. Das kann eine kryptische Zeichenfolge, aber auch ein logisch vorbelegter Name sein, wie es beispielsweise bei Textfeldern und einigen anderen Komponenten in APEX der Fall ist. Doch Vorsicht: Automatisch erzeugte IDs für Elemente können sich

je nachdem, wie Sie die Anwendung exportieren oder importieren, ändern!

Ist ein Test-Case im Test-Designer erstellt und gespeichert worden, so wird anhand dieser Eingaben und ausgewählter Komponenten dafür gesorgt, dass ein automatisiertes Erstellen der Tests möglich wird. Hierzu benötigen wir das sogenannte Manifest.

Das Manifest

Eine wichtige Rolle für die Definition von Test-Skripten ist ein Manifest, das so abstrakt ist, dass es alle Informationen für die auszuführenden Tests beinhaltet, jedoch so generisch, dass es automatisiert weiterverarbeitet werden kann. Zu den enthaltenen Informationen gehören auch die im letzten Absatz genannten APEX-Items. Ein Manifest kann wie in Listing 1 dargestellt aussehen:

In diesem Manifest sehen wir zum einen Bestandteile, die immer vorhanden sind, wie zum Beispiel die URL, bei der ein Test starten soll, oder auch Informationen darüber, wie und mit welchen Daten sich in der Anwendung angemeldet werden soll. Darüber hinaus befindet sich aber auch der Punkt „steps“ auf der obersten Ebene. Innerhalb dieses Attributs „steps“ befindet sich eine Sammlung aus generischen Ob-

```

{
  "base_url": "http://apex.kai-donato.de/ords/f?p=",
  "test_project_name": "SampleApplication123",
  "login": {
    "needed": 1,
    "login_pageid": 101,
    "usr_fn": "scott",
    "pwd_fn": "secret"
  },
  "app_id": 103,
  "steps": [{
    "onpage": 1,
    "isModalPage": 0,
    "type": "fillForm",
    "items": ["P1_SEARCH"],
    "item_sub_type": ["NATIVE_TEXT_FIELD"],
    "values": ["Bug"]
  }]
}

```

Listing 1: Das Manifest

```

Feature('Sample Application Login and Search');
Scenario('Main-Szenario', (I) => {
  I.amOnPage('http://apex.kai-donato.de/ords/f?p=103');
  I.waitForElement('#P101_USERNAME', 5);
  I.waitForElement('#P101_PASSWORD', 5);
  I.appendField('#P101_USERNAME', 'scott');
  I.appendField('#P101_PASSWORD', 'secret');
  I.click('Sign In');
  I.wait(10);
  I.dontSeeElement('.t-Alert-wrap');
  I.saveScreenshot('Auth26-05-2019.png');
  I.waitForElement('#P1_SEARCH', 5);
  I.appendField('#P1_SEARCH', 'Bug');
  I.saveScreenshot('fillFormstep-026-05-2019.png');
});

```

Listing 2: CodeceptJS-Code

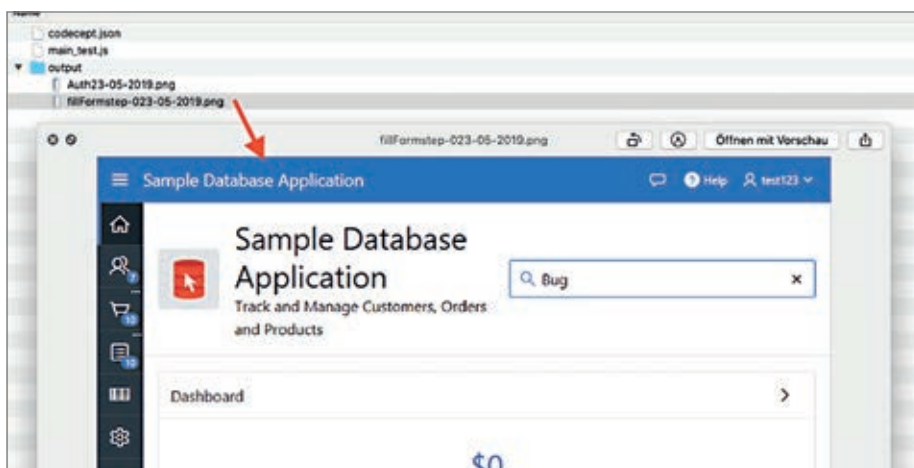


Abbildung 3: Blick auf das Dateisystem nach der Ausführung eines Tests (Quelle: Kai Donato)

jekten, die dann die einzelnen Testschritte beschreiben. So können wir beispielsweise den Typ eines Testschritts nutzen, um festzustellen, welche Interaktion im Laufe des Tests erfolgen soll. Dies ist notwendig

für die Übersetzung in die einheitliche Testsprache, die wir im folgenden Absatz über CodeceptJS kennenlernen werden. Ebenfalls innerhalb eines Testschritts befindet sich, sofern nötig, eine Auflistung beteilig-

ter Elemente. Bei dem oben gezeigten Manifest ist beispielsweise das Befüllen eines Eingabefeldes gefordert und demnach ein Element (P1_SEARCH) und der einzugebende Text-String (Bug) hinterlegt. Zusammen mit dem Typ des Testschritts (fillForm) ist nun alles bekannt, um einen entsprechenden Testschritt zu erzeugen. Und das übrigens nicht nur einmalig, sondern beliebig häufig, wenn sich etwas an den Test-Templates ändert. Nehmen wir also an, dass ein Test aufgrund von Änderungen in einer neuen Version von APEX nicht mehr zuverlässig funktioniert, so muss lediglich das Test-Template so modifiziert werden, dass es eine Komponente gemäß der neuen Funktionsweise testen kann. Anschließend werden die entsprechenden Tests anhand der existierenden Manifest-Dateien erneut generiert und können dann wie gewohnt ausgeführt werden.

CodeceptJS

Mit einem Framework wie „CodeceptJS“ lassen sich die zu Beginn des Artikels genannten Nachteile der Test-Frameworks aus der Welt schaffen und mit einer einheitlichen Syntax viele Test-Engines für die Ausführung unserer Skripte nutzen. Das folgende einfache Test-Skript in *Listing 2* verdeutlicht, wie einfach der CodeceptJS-Code selbst zu lesen ist und nach welchem Schema er aufgebaut ist:

Dieses Test-Skript ist nicht nur leicht zu lesen, es ist auch im Aufbau so konzipiert worden, dass es sich unter anderem mit den gängigen Treibern ausführen lässt. Hierzu gehören zum Beispiel:

- WebDriver
- Puppeteer
- Protractor
- Appium

Diese sogenannten Treiber, auch Engines genannt, stellen die Basis für das Testen in Google Chrome, Mozilla Firefox, Apple Safari, Microsoft Internet Explorer, Microsoft Edge oder sogar auf mobilen Geräten dar.

Wer sich das Testskript etwas genauer ansieht, kann dort einiges bereits Bekannte erkennen. Unter anderem sind dort die Namen von APEX-Items sowie APEX-spezifische CSS-Klassen zu sehen. Diese kommen aus den Metadaten unserer APEX-Anwendung, doch wo werden diese definiert?

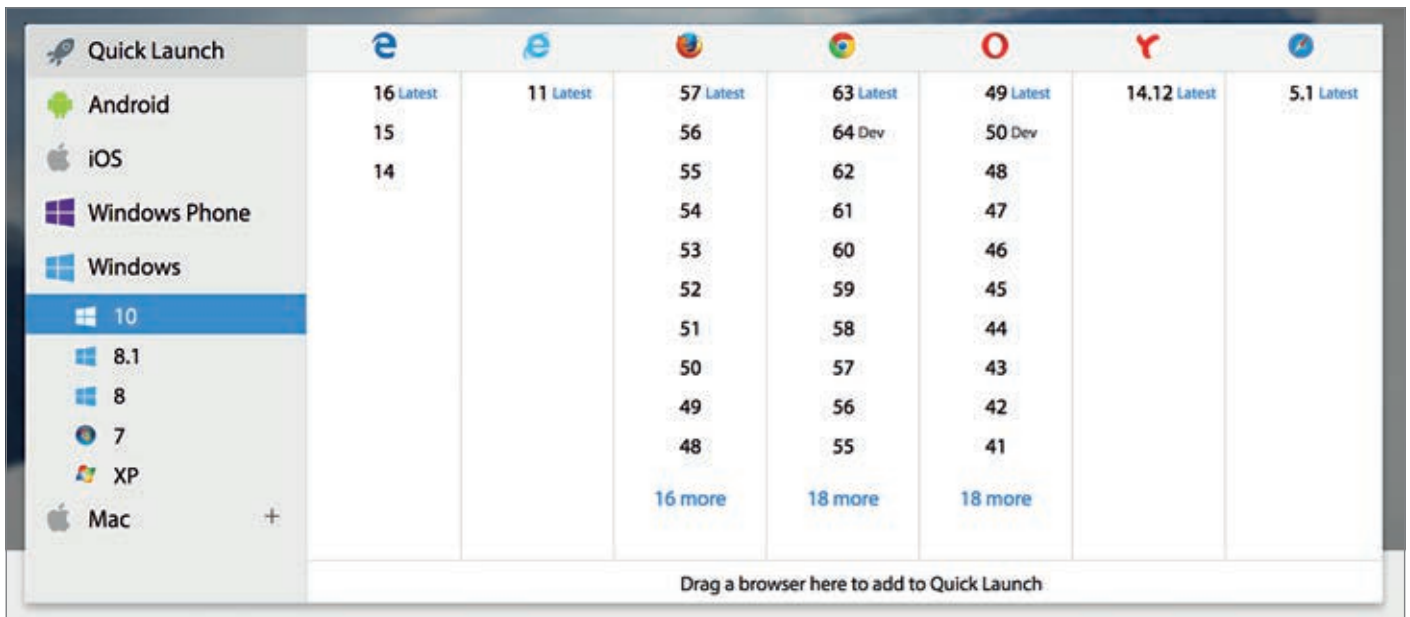


Abbildung 4: Auszug aus den verfügbaren Browsern auf Browserstack.com (Quelle: Kai Donato)

Ausführung der Tests

Die Ausführung der generierten Tests ist zum Zeitpunkt dieses Artikels auf drei Arten möglich. Zum einen können die Tests in einem Docker-Container ausgeführt werden, der sich innerhalb der eigenen Infrastruktur befindet. Hierbei wird ein vorbereitetes Docker-Image von CodeceptJS verwendet, das bereits alle notwendigen Treiber und Softwarepakete beinhaltet. Hierbei wird beispielsweise ein „Headless Chrome“ gestartet, das nicht mehr und nicht weniger als ein voll funktionsfähiger Google-Chrome-Browser ist, mit der Be-

sonderheit, dass es keine grafische Oberfläche gibt, die der Benutzer betrachten kann. Was jedoch möglich ist, ist die Erstellung von Screenshots, die uns hinterher Aufschluss darüber geben, was in dem Browser bei dem jeweiligen Testschritt angezeigt wurde. Das hat zur Folge, dass der Docker-Container ohne die nötigen Ressourcen eines Arbeitsplatz-Computers auskommt und dazu auch noch vollkommen unabhängig vom Host-System läuft. Somit kann für jeden Test ein neuer „frischer“ Container gestartet werden, der sich immer in derselben Ausgangssituation befindet. Dieser Docker-Container

kann aber auch dafür genutzt werden, die Tests weiter zu orchestrieren. Nehmen wir an, wir besitzen ein Selenium Grid, das bereits eigenständig aufgesetzt wurde und eine Vielzahl von Browsern bereitstellt. In der Definition unserer Tests können wir darauf verweisen, dass der jeweilige Test nicht im Container selbst ausgeführt werden soll, sondern alles Nötige an das Selenium Grid weitergibt und eine Ausführung von dort an im Grid stattfinden soll. Alle relevanten Testergebnisse und auch alle Screenshots landen nach der Ausführung des Tests wieder im Docker-Container und werden vor dem Beenden des Prozesses wie gewohnt an APEX weitergeleitet und auf dem Dateisystem abgelegt.

Da es sich in erster Linie um User-Acceptance-Tests handelt, kann abgesehen von den Tests innerhalb der eigenen Infrastruktur von jedem Ort aus getestet werden, von dem die zu testende Anwendung erreichbar ist. Hierzu können wir mithilfe des Anbieters „Browserstack“ (siehe unter <https://browserstack.com>) testen. Hierbei wird anstelle des im Container befindlichen Browsers oder des Selenium Grid ein externer Dienst aufgerufen, der die Tests dann in unterschiedlichen Browsern auf diversen Plattformen ausführt. Dazu verlassen dann unsere Test-Skripte inklusive möglicher Login-Informationen die eigene Infrastruktur.

Ich denke, zu all diesen Möglichkeiten bieten sich Anwendungsfälle. Sicherlich ist das Testen innerhalb der eigenen

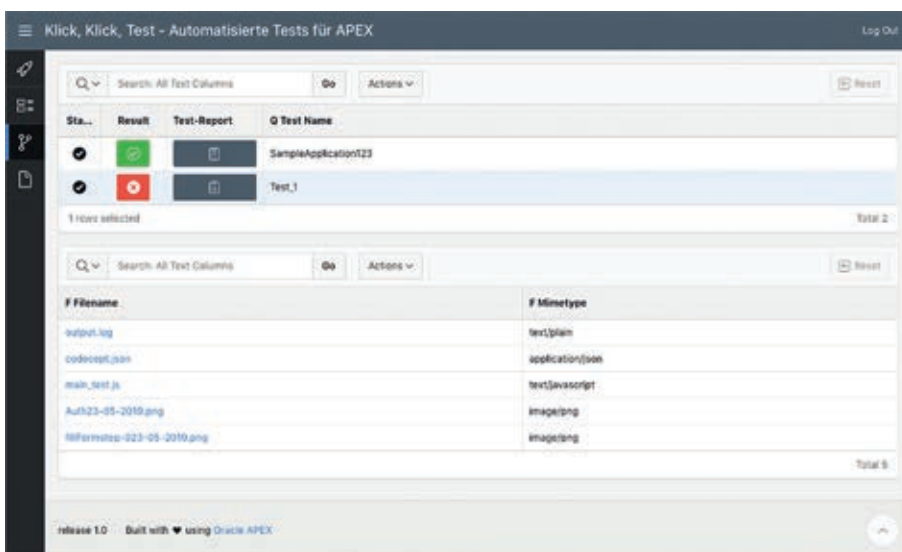


Abbildung 5: Ausgeführte und laufende Tests werden übersichtlich in der Queue dargestellt (Quelle: Kai Donato)



Abbildung 6: Schematische Darstellung des Test-Prozesses (Quelle: Kai Donato)

Infrastruktur häufiger von Vorteil und mit Sicherheit auch oft nötig. Es gibt jedoch auch die Fälle, in denen man davon profitieren kann, dass Tests von außerhalb ausgeführt werden. Eine Vielzahl von Betriebssystemen und Browsern stehen zur Verfügung, um eine möglichst hohe Testabdeckung zu ermöglichen. Die Server befinden sich allerdings, wie bereits gesagt, nicht innerhalb der eigenen Infrastruktur und sollten gerade im Zusammenhang mit sensiblen Daten mit Vorsicht genutzt werden.

Die Queue

Egal welche der oben genannten Ausführungsoptionen man verwendet, landen alle Testergebnisse wieder in unserer Anwendung und können dort ausgewertet werden.

Nachdem ein Test für die Ausführung bestimmt oder automatisch gestartet wurde, taucht ein neuer Eintrag in der Test-Queue auf und zeigt dessen Status in Echtzeit an. Nach Abschluss des Tests kann eingesehen werden, welche Dateien erzeugt wurden; sowohl die Log-Ausgaben als auch erstellte Screenshots können angezeigt werden.

Module

Der deklarative Ansatz dieser Lösung steht klar im Vordergrund. Der Endbenutzer soll, wie er es in APEX gewohnt ist, auf deklarative Art und Weise Tests erstellen und sich so wenig wie möglich mit den technischen Einzelheiten, die im Hintergrund passieren, auseinandersetzen. Durch den modularen Aufbau der Engine im Hintergrund kann in Form von Templates dafür gesorgt werden, dass spezielle, im Projekt angeforderte Testroutinen einmalig ausprogrammiert werden und anschließend deklarativ in der Oberfläche des Tools verwendet werden. Dies ist zum einen Aufwand, der projektspezifisch geleistet werden muss, jedoch auch die Flexibilität, die man von einem Tool erwarten sollte. So ist es auch möglich, die im Framework enthaltenen Oberflächen-Tests um Integrations- und Modultests zu erweitern.

Fazit

Aus einer Idee und vielen Experimenten konnte eine Lösung geschaffen werden, die uns in Zukunft viel Zeit sparen kann. Als APEX-Entwickler haben wir die de-

klarative Arbeit schätzen und lieben gelernt und sind es gewohnt, mit wenigen Klicks ein tolles Ergebnis zu schaffen. Das vorgestellte Tool soll eine weitere Lücke schließen und den deklarativen Ansatz auch für das Thema Testing ermöglichen. Es gibt viele Features, die sich in der Pipeline befinden, und so wird beispielsweise auch das visuelle Testen in Form von Screenshot-Vergleichen Einzug in diese Lösung finden. Zu guter Letzt bin ich gespannt darauf, was Sie zu diesem Thema zu sagen haben und wie diese Lösung bei Ihnen eingesetzt werden kann/könnte!



Kai Donato
kai.donato@mt-ag.com

Stateless Packages in PL/SQL und Hochverfügbarkeit von Applikationen

Athanasios Manolopoulos, Sopra Steria

Ein tieferes Verständnis für die effiziente Nutzung der erweiterten PL/SQL-Funktionen erhalten Entwickler bei der TÜV Rheinland Akademie. Als Oracle Authorized Education Center (OAEC) ist die TÜV Rheinland Akademie Ihr zuverlässiger Anbieter von Oracle-Schulungen. Wir bieten Ihnen mehr als klassische Präsenzseminare an. Mit unseren digitalen Lernformen können Sie jederzeit und von jedem Ort aus an Ihrem Oracle-Seminar teilnehmen. Weitere Informationen unter www.tuv.com/it-training



Einleitung

Der Erstaufwurf von geänderten und neu kompilierten PL/SQL-Unterprogrammen aus zuvor geöffneten Sessions verursacht in manchen Fällen den harmlosen, aber unangenehmen Fehler ORA-04068 (der Zustand der Pakete hat sich geändert). Obwohl der Fehler beim erneuten Aufruf des Unterprogramms verschwindet, wirkt er auf einen Nicht-PL/SQL-Routinier oder auf einen Endbenutzer erst einmal verstörend. Etwas komplexer wird die Lage, wenn der PL/SQL-Code aus anderen Programmierumgebungen aufgerufen wird (z.B. Servlets, Micro-Services etc.). Applikationen müssen in diesem Fall neu gestartet werden. Eine daraus resultierende Problematik lautet: Eine einfache Re-Kompilierung von PL/SQL-Packages gefährdet die Hochverfügbarkeit einer Applikation. Dieser Artikel widmet sich diesem Problem und zeigt Wege und Lösungen zu seiner Vermeidung.

Package-Zustände

Aus der Oracle-Dokumentation entnehmen wir: Ein Package gilt als zustandsbehaftet, wenn es öffentliche und/oder private globale Package-Variablen enthält. Sein Zustand ergibt sich aus den Werten, die zu einem bestimmten Zeitpunkt in allen Package-Variablen gespeichert sind. Default-mäßig gilt

der Package-Zustand für die Dauer der Benutzersession; die Package-Variablen werden in der User Global Area gespeichert. Der Package-Zustand ändert sich, wenn ein Package-Unterprogramm, dessen Code Package-Variablen manipuliert, erneut kompiliert wird. Dies lässt sich anhand eines einfachen Beispiels leicht feststellen. Wir führen folgende Testschritte aus:

1. Wir öffnen ein SQL-Developer-Fenster (im Nachfolgenden Fenster No 1

genannt), wir loggen uns in einem Testschema ein und erstellen ein Package, das eine globale Variable namens `gv_n` im Package-Body enthält. Des Weiteren enthält das Package eine einfache öffentliche Prozedur namens `doit`, die lediglich eine Bildschirmausgabe der globalen Variable macht. Die Anweisung „null“, die danach folgt, steht symbolisch und stellvertretend für den eventuellen Rest-Code der Prozedur, der uns hier nicht

```
create or replace package pkg_demo as
  procedure doit;
end;
/
create or replace package body pkg_demo as

  gv_n number := 5;

  procedure doit is
  begin
    DBMS_OUTPUT.put_line(gv_n);
    null;
  end;

end;
/
```

Listing 1

```
set serveroutput on
execute pkg_demo.doit
```

Listing 2

weiter interessieren sollte (siehe Code im Listing 1).

2. Danach öffnen wir ein weiteres SQL-Developer-Fenster (im Nachfolgenden Fenster No 2 genannt), wir loggen uns im gleichen Schema ein und rufen die Prozedur doit gemäß dem Code im Listing 2 auf. Der Aufruf läuft erfolgreich mit Ausgabe des Variablenwerts.
3. Im Fenster No 1 kommentieren wir im Code der Prozedur doit die Anweisung „null“ aus (siehe Code im Listing 3).
4. Im Fenster No 2 rufen wir die Prozedur doit gemäß dem Code im Listing 2 erneut auf. Diesmal bekommen wir den berüchtigten ORA-04068-Fehler (siehe Listing 4).

Grund für dieses Verhalten ist: Wir haben es mit einer Prozedur zu tun, in deren Code globale Package-Variablen referenziert werden. Jede Änderung im Code dieser Prozedur, so klein und unbedeutend sie auch sein mag (in unserem Beispiel das Auskommentieren einer null-Anweisung), hat eine Änderung des Package-Zustands zur Folge und erzeugt bei ihrem nächsten Aufruf aus einer zuvor geöffneten Session den ORA-04068-Fehler.

Das gleiche Verhalten fände statt, wenn die Variable gv_n öffentlich wäre (Variablendefinition befindet sich hier in der Package-Spezifikation). Es stellt sich nun die Frage: Wie können wir diesen lästigen Fehler loswerden? In den folgenden Abschnitten werden wir uns den unterschiedlichen Lösungen zu diesem Problem widmen.

Lösungsvariante No1: PRAGMA SERIALLY_ REUSABLE

Diese Lösungsvariante besteht in der Verwendung der Compiler-Anweisung PRAGMA SERIALLY_REUSABLE in der Package-Spezifikation und im Package-Body (siehe Code im Listing 5).

Zum Testen der Wirkung dieser PRAGMA-Anweisung wiederholen wir den Test des ersten Kapitels:

1. Im Fenster No 1 kompilieren wir den Code des Listing 5.
2. Im Fenster No 2 rufen wir die Package-Prozedur doit auf (siehe Listing 2). Dieser Erstaufruf läuft erfolgreich.

```
create or replace package body pkg_demo as

  gv_n number := 5;

  procedure doit is
  begin
    DBMS_OUTPUT.put_line(gv_n);
    --null;
  end;

end;
/
```

Listing 3

```
ORA-04068: existing state of packages has been discarded
ORA-04061: existing state of package body "HR.PKG_DEMO" has been invalidated
ORA-04065: not executed, altered or dropped package body "HR.PKG_DEMO"
ORA-06508: PL/SQL: could not find program unit being called: "HR.PKG_DEMO"
ORA-06512: at line 1
04068. 00000 - "existing state of packages%*s%*s%*s%*s has been discarded"
*Cause: One of errors 4060 - 4067 when attempt to execute a stored
        procedure.
*Action: Try again after proper re-initialization of any application's
        state.
```

Listing 4

```
create or replace package pkg_demo as
  PRAGMA SERIALLY_REUSABLE;
  procedure doit;
end;
/

create or replace package body pkg_demo as

  PRAGMA SERIALLY_REUSABLE;

  gv_n number := 5;

  procedure doit is
  begin
    DBMS_OUTPUT.put_line(gv_n);
    null;
  end;

end;
/
```

Listing 5

3. Im Fenster No 1 kommentieren wir die Anweisung „null“ aus (siehe Listing 6).
4. Im Fenster No 2 rufen wir erneut die Package-Prozedur doit auf. Dieser Aufruf läuft diesmal ebenfalls erfolgreich.

Nach dem letzten Testlauf stellen wir fest: Code-Änderungen verursachen keine Veränderung mehr auf dem Package-Zustand. Der ORA-04068 bleibt aus. Was ist nun anders? Die Erklärung dazu liefert wieder die Oracle-Dokumentation:

Die Compiler-Anweisung PRAGMA SERIALLY_REUSABLE veranlasst den Datenbankserver, die Package-Variablen in der System Global Area (SGA) statt der User Global Area zu speichern. Dadurch werden die Package-Variablen gemeinsam von allen Sessions verwendet. Der Package-Zustand ist in diesem Fall nur für die Dauer eines Unterprogrammaufrufs gültig und das Package wird auf diese Art und Weise „zustandslos“ (stateless). Eine Re-Kompilierung des Packages wirkt sich

```

create or replace package body pkg_demo as

    PRAGMA SERIALLY_REUSABLE;

    gv_n number := 5;

    procedure doit is
    begin
        DBMS_OUTPUT.put_line(gv_n);
        -- null;
    end;

end;
/

```

Listing 6

```

create or replace package pkg_demo as
    PRAGMA SERIALLY_REUSABLE;
    procedure doit;
    function twice(p_n number) return number;
end;
/

create or replace package body pkg_demo as

    PRAGMA SERIALLY_REUSABLE;

    gv_n number := 5;

    procedure doit is
    begin
        DBMS_OUTPUT.put_line(gv_n);
        null;
    end;

    function twice(p_n number) return number is
    begin
        return p_n * 2;
    end;

end;
/

```

Listing 7

auf seinen Zustand nicht mehr aus und der ORA-04068 verschwindet.

Diese Lösungsvariante erfordert minimalen Eingriff im Package-Code, hat aber wiederum auch eine Einschränkung: Funktionen, die in Serially Reusable Packages definiert sind, können nicht in SQL-Anweisungen verwendet werden. Dies lässt sich gut durch Erweiterung unseres Beispiels beobachten. Wir führen zu diesem Zweck folgende Testschritte aus:

1. Im Fenster No 1 fügen wir dem Package pkg_demo eine öffentliche Funktion namens „twice“ hinzu. Diese Funktion ist recht einfach gestrickt: Sie akzeptiert ein numerisches Argument als Eingabeparameter, multipliziert

den Eingangswert mit zwei und liefert den neuen Wert als Ergebnis zurück (siehe Listing 7). Diesen Code kompilieren wir im Fenster No 1.

2. Im Fenster No 2 rufen wir diese Funktion innerhalb einer SQL-Anweisung auf (siehe Listing 8). Die Ausführung dieser SQL-Anweisung verursacht den Fehler, der in Listing 9 zu sehen ist.

Diese Fehlermeldung liefert den Nachweis für die oben erwähnte Einschränkung: Funktionen von Serially Reusable Packages können nicht aus SQL-Anweisungen aufgerufen werden. Inwieweit man jedoch diese Einschränkung als Nachteil betrachten kann, liegt im Ermessen des Einzelnen. Der Autor vertritt die

Ansicht, dass PL/SQL-Funktionsaufrufe in SQL-Abfragen generell nicht verwendet werden sollten. Jeder PL/SQL-Aufruf innerhalb von SQL-Abfragen verursacht nämlich einen Kontextwechsel zwischen der SQL- und der PL/SQL-Engine und sorgt für eine Performance-Verschlechterung. Je größer die Ergebnismenge ist, desto spürbarer wird die Performance-Einbuße.

Der Verzicht auf PL/SQL-Funktionsaufrufe innerhalb von SQL-Abfragen ist in der Praxis einfacher realisierbar, als es sich anhört. Der Autor hat es sehr häufig erlebt, dass sich der größte Teil der in den SQL-Abfragen einer Applikation verwendeten PL/SQL-Funktionen direkt in SQL programmieren ließe. Manche Funktionen lassen sich durch eine kombinierte Verwendung von Single-Row SQL-Funktionen, Arithmetik, Datumsarithmetik und IF-THEN-ELSE-Logik (CASE-Ausdrücke) ersetzen. Andere Funktionsaufrufe könnten durch zusätzliche JOIN-Operationen vermieden werden. Die perfekte Beherrschung der Abfragesprache SQL ist dabei eine unabdingbare Voraussetzung.

Nichtsdestotrotz sei hier fairerweise erwähnt, dass sich nicht jede PL/SQL-Funktion mit SQL-Mitteln realisieren lässt. Die Berechnung von Phonetik-Codes von Nachnamen oder die Berechnung der Abgeltungssteuer für ein bestimmtes Bankkonto sind hoch komplexe Vorgänge, die sich nicht allein mit SQL-Mitteln realisieren lassen. In solchen Fällen bleibt uns die Alternative, die gesamte Logik in PL/SQL zu verlagern. Dabei bedienen wir uns der Methode des Bulk Binding. Der Autor hat zu diesem Thema einen Artikel in der Ausgabe vom Januar 2018 des Red Stack Magazin veröffentlicht, wo er es im Detail behandelt. Dieser Artikel ist in elektronischer Form vorhanden und kann beim Autor bei Bedarf jederzeit angefordert werden.

Jetzt aber zurück zum Hauptthema mit einer kleinen Zusammenfassung der bisherigen Erkenntnisse: Die erste Möglichkeit, PL/SQL-Packages als zustandslos (stateless) zu implementieren, ist, die Compiler-Anweisung PRAGMA SERIALLY_REUSABLE zu verwenden. Dabei muss man im Hinterkopf behalten, dass man Funktionen eines zustandslosen PL/SQL-Packages nicht in SQL-Anweisungen verwenden kann. Eine weitere Ein-

schränkung betrifft verschachtelte Aufrufe. Unterprogramme eines zustandslosen Packages dürfen ihrerseits keine Unterprogramme anderer Packages aufrufen, die zustandsbehaftet sind. Der gesamte Aufruf-Stack muss ausschließlich aus Aufrufen von Unterprogrammen zustandsloser Packages bestehen.

Wenn das Kind schon in den Brunnen gefallen ist: die Lösungsvariante No 2

Bezüglich der Lösungsvariante No 1 gibt es ein Problem, das nicht zu unterschätzen ist: Was ist, wenn wir ein altes PL/SQL-Package nachträglich in ein zustandsloses verwandeln möchten, dessen Funktionen aber in etlichen SQL-Statements unserer Applikation schon verwendet werden? So eine Situation taucht öfter bei langjährigen Projekten auf, bei denen der Code von einer großen Anzahl unterschiedlicher Personen gepflegt wurde und mit der Zeit zu einem unübersehbaren Koloss gewachsen ist. In diesem Fall wäre zwar die Umwandlung des Packages in ein zustandsloses durch die PRAGMA-Anweisung schnell und einfach, dies würde allerdings ein umfassendes Re-Design großer Teile der Applikation nach sich ziehen. Man müsste erst mal alle davon betroffenen SQL-Statements identifizieren und anschließend für jedes davon eine Lösung implementieren. Eine Mammutaufgabe, für die höchstwahrscheinlich auch kein Budget zur Verfügung steht. In diesem Fall muss eine Alternative her und die lautet:

Entfernung der globalen Variablen und Konstanten aus dem Package-Code. Dies klingt zwar etwas restriktiv, ist aber realisierbar. Im Nachfolgenden werden einige Methoden vorgestellt, die veranschaulichen, wie die Implementierung dieses Vorhabens aussehen könnte.

Erster Teil: Entfernung der globalen Variablen

Als Erstes widmen wir uns der Problematik der globalen Variablen. Der Code des *Listing 8* erweitert das ursprüngliche Beispiel um ein paar weitere lokale Package-Prozeduren namens `init`, `proceed` und `print`. In der Prozedur `init` wird die globale Variable `gv_n` initialisiert, in der Prozedur `proceed` wird sie manipuliert und in der

```
select pkg_demo.twice(5) from dual;
```

Listing 8

```
ORA-06534: Cannot access Serially Reusable package "HR.PKG_DEMO"
ORA-06512: at line 1
06534. 00000 - "Cannot access Serially Reusable package %s"
*Cause:   The program attempted to access a Serially Reusable package
in
          PL/SQL called from SQL context (trigger or otherwise). Such
an
          access is currently unsupported.
*Action:  Check the program logic and remove any references to Serially
Reusable
          packages (procedure, function or variable references) which
might happen
          in PL/SQL called from sql context (trigger or otherwise).
```

Listing 9

```
create or replace package pkg_demo as
  procedure doit;
  function twice(p_n number) return number;
end;
/

create or replace package body pkg_demo as

  gv_n number;

  procedure init is
  begin
    gv_n := 5;
  end;

  procedure proceed is
  begin
    gv_n := gv_n + 4;
  end;

  procedure print is
  begin
    DBMS_OUTPUT.put_line(gv_n);
  end;

  procedure doit is
  begin
    init;
    proceed;
    print;
    null;
  end;

  function twice(p_n number) return number is
  begin
    return p_n * 2;
  end;

end;
/
```

Listing 10

Prozedur print ausgegeben. Die Main-Prozedur doit ruft alle drei nacheinander auf. Die Prozedur doit enthält außerdem weiterhin unsere Testanweisung „null“. Das Package wird diesmal ohne die Compiler-Anweisung PRAGMA SERIALLY_REUSABLE erstellt (siehe Listing 10).

Wir wiederholen die uns bekannten vier Testschritte, nachdem wir zuvor die SQL-Developer-Fenster No 1 und No 2 geschlossen und erneut geöffnet haben:

1. Im Fenster No 1: Code von Listing 10 kompilieren
2. Im Fenster No 2: Package-Prozedur doit gemäß Listing 2 aufrufen (läuft erfolgreich)
3. Im Fenster No 1: Anweisung „null“ auskommentieren und erneut kompilieren
4. Im Fenster No 2: Package-Prozedur doit erneut aufrufen (terminiert mit dem ORA-04068-Fehler)

Der Fehler des Testschritts No 4 kam wie erwartet, da das Package aufgrund des Vorhandenseins der globalen Variable gv_n immer noch als zustandsbehaftet gilt. Wir werden nun als Nächstes alle benötigten Code-Änderungen im Package Body vornehmen, um das Package in ein zustandsloses umzuwandeln. Dabei gehen wir folgendermaßen vor:

- Die globale Variable gv_n wird durch die lokale Variable lv_n im Body der Prozedur doit ersetzt.
- Die Prozeduren init, proceed und print bekommen einen numerischen Parameter namens p_n, der zur Weiterreichung des Wertes von lv_n dient.

Diese Änderungen werden in Listing 11 veranschaulicht.

Wir führen anschließend den vorigen Test mit den vier Schritten nochmal aus. Wir stellen fest, dass der Testschritt No 4 diesmal fehlerfrei terminiert. Dies ist der Nachweis, dass unser Package jetzt zustandslos ist. Auch der Aufruf der Package-Funktion twice aus der SELECT-Anweisung aus Listing 8 läuft diesmal fehlerfrei und liefert ein Ergebnis:

```
select pkg_demo.twice(5) from
dual;
PKG_DEMO.TWICE(5)
```

10

Zweiter Teil: Entfernung der globalen Konstanten

Wir befassen uns als Nächstes mit der Behandlung der Package-Konstanten. Die Vorgehensweise ist hier recht einfach: Wir ersetzen Konstanten durch Funktionen, die eine Konstante als Rückgabewert liefern. Der Code der Listings 12 und 13 zeigt, wie sich dieses Vorhaben implementieren lässt. Listing 12 zeigt die Ausgangslage und Listing 13 das Ziel.

Derartige Funktionen wie die des Listing 13 eignen sich außerdem hervorragend für Caching (siehe RESULT_CACHE-Klausel in der Funktionssignatur). Ihr Aufruf wird dadurch äußerst performant.

Vergleich der beiden Lösungsvarianten

Die hier beschriebenen Implementierungen der beiden Lösungsvarianten haben Vor- und Nachteile, auf die im Folgenden kurz eingegangen wird.

Vorteil der ersten Lösungsvariante: Minimaleingriff am Package. Zu ihrer Implementierung ist lediglich eine PRAGMA-Anweisung erforderlich. Nachteil: Eventuell wird der Rest der Applikation betroffen, wenn Funktionen aus diesem Package in SQL-Statements aufgerufen werden. In diesem Fall wird eventuell ein ausgeweiteter Eingriff in großen Teilen der Applikation erforderlich.

Vorteil der zweiten Lösungsvariante: Der Gesamteingriff beschränkt sich nur aufs Package. Der restliche Applikationscode bleibt von diesem Eingriff unberührt. Nachteil: Zusätzlicher Aufwand beim Kodieren des Packages durch Verzicht auf bestimmte Sprachelemente.

Nachwort

Der derzeitige Trend um die Entwicklung von Micro-Services hat die verstärkte Forderung nach hochverfügbaren Applikationen nach sich gezogen. In vielen Entwicklerkreisen, besonders bei Kollegen mit einer Java-Spezialisierung, herrscht der falsche Eindruck vor, dass Änderungen an Datenbankmodulen (Packages) immer mit einer Downtime der Applikation verbunden sind, was zu einer generellen Abneigung gegen die Verwendung von PL/SQL führt. Dieser Artikel wurde mit der Ab-

sicht verfasst, dieses Missverständnis aus dem Weg zu räumen. Software Patches, bei denen datenbankseitig lediglich eine Re-Kompilierung von PL/SQL-Packages erforderlich ist, könnten bei laufender Applikation ohne Downtime durch eine einfache Re-Kompilierung installiert werden. Voraussetzung für den nahtlosen Übergang ist jedoch die Zustandslosigkeit der PL/SQL-Packages. Hier wurden einige Methoden zu ihrer Implementierung vorgestellt. Der Weg dorthin ist zwar nicht immer einfach, aber in jeder Hinsicht realisierbar.

Info zum Autor

Name: Athanasios Manolopoulos
Firma: Sopra Steria Consulting
Nebentätigkeit: Oracle Trainer für SQL, PL/SQL, Datenbankadministration und Performance Tuning.

Sämtliche Listings mitsamt den drei noch fehlenden finden Sie unter folgendem Link:

http://www.doag.org/go/redstack/201904_listings_manolopoulos



Athanasios Manolopoulos
a.manolopoulos@soprasteria.com

Teile und herrsche: Partitionierung in der Oracle-Datenbank

Dierk Lenz, Herrmann & Lenz Services GmbH

Die Partitioning-Option ist eine kostenpflichtige Option der Enterprise Edition der Oracle-Datenbank. Daher sind wir de facto im Bereich der hochpreisigen Lizenzen. Partitionierung ist daher in eher großen Umgebungen zu finden; oft im Umfeld von Unternehmenslizenzen.

Bei der Einführung von Partitionierung ist häufig zu beobachten, dass mit der Einstellung „Wir partitionieren jetzt irgendwie und alles wird gut!“ gearbeitet wird. Der zweite Teil des Satzes will sich aber nicht so recht einstellen. In diesem Artikel werden einige grundsätzliche Fakten über Partitionierung, einige neuere Features sowie ein paar grundlegende Zusammenhänge dargestellt.

Die Idee

Oracle-Tabellen werden auch als heap tables bezeichnet, also umgangssprachlich als ungeordnete Haufen von Daten. Diese Eigenschaft ist durchaus wörtlich zu nehmen: Es gibt keine vorgeschriebene Ordnung der Datensätze innerhalb einer Tabelle. Insbesondere gilt dies für Tabellen, in die Daten regelmäßig einerseits per INSERT hineinlaufen und andererseits per DELETE wieder entfernt werden.

Betrachtet man die Menge der aktuellen Datensätze, so sind diese in einer solchen Struktur auf viele Blöcke verteilt. Das führt wiederum dazu, dass das Cache-Verhalten einer solchen Tabelle nicht besonders gut ist. Typische Operationen wie das DELETE der ältesten Datensätze erstrecken sich über viele Blöcke und sind demnach sehr teuer. Des Weiteren ist die Administration schwierig, da praktisch immer der Gesamtdatenbestand bearbeitet werden muss.

Die Idee der Partitionierung ist nun, den Datenbestand für die Anwendung transparent in kleinere Stücke aufzuteilen, wobei die Aufteilung nach einer logischen Vorgabe erfolgt. In Oracle-Sprache heißt das, dass die Eins-zu-eins-Beziehung zwischen Tabelle und Segment aufgebrochen wird: Die Tabelle wird zunächst logisch in Partitionen auf-

geteilt. Jede Partition bekommt dann ein eigenes Segment. Gleiches gilt für die Indizes, falls diese ebenfalls partitioniert sind.

Das Ganze passiert vollkommen transparent für die Anwendung, SQL-Befehle müssen also in keiner Weise verändert werden, um mit der Partitionierung klarzukommen. Doch Vorsicht, die Transparenz birgt Gefahren! Wie wir später noch sehen werden, gibt es sehr wohl Zusammenhänge zwischen der Art der Partitionierung und der Formulierung der SQL-Befehle. Die Partitionierung bietet viele Vorteile, wie zum Beispiel erheblich besseres Cache-Verhalten, wenn eine zeitbasierte Partitionierung gewählt wird und die aktuelle Partition die aktuellen Datensätze enthält. Das Löschen alter Datensätze kann gegebenenfalls durch das Löschen einer ganzen Partition ersetzt und somit extrem beschleunigt werden. Zudem kann die Parallelverarbeitung bei Full Table Scans die Partitionierung nutzen und mehrere Partitionen gleichzeitig verarbeiten.

Die Basics

In diesem Abschnitt werden die wichtigsten Begriffe und Techniken bei der Partitionierung kurz angerissen. Zur Vertiefung kann zum Beispiel der VLDB and Partitioning Guide der Oracle-Dokumentation herangezogen werden (*siehe <https://docs.oracle.com/en/database/oracle/oracle-database/19/vldb/index.html>*)

Der Standard: Range-Partitionierung

Bei der Range-Partitionierung wird jeder Partition ein Wertebereich der Partitio-

nierungsspalte(n) zugeordnet. Über die VALUES-LESS-THAN-Klausel wird die offene obere Intervallgrenze definiert, die geschlossene untere Intervallgrenze ergibt sich implizit über das darunter liegende Intervall.

Als Partitionierungsspalten werden hierbei oft DATE- oder TIMESTAMP-Spalten verwendet. Somit werden zeitbasierte Partitionen (z.B. Monatspartitionen) definiert.

Nützlich: List-Partitionierung

Wenn die möglichen Werte der Partitionierungsspalte lediglich eine überschaubare Menge von Werten annehmen kann (z.B. Vertriebsgebiete, Farben usw.), dann kann mit der List-Partitionierung jedem Wert eine Partition zugewiesen werden.

Hash-Partitionierung

Bei der Hash-Partitionierung wird neben der Partitionierungsspalte lediglich die Anzahl der gewünschten Partitionen angegeben. Die Datensätze werden nicht über Ranges oder Listen, sondern über eine Hash-Funktion auf der Partitionierungsspalte auf die Partitionen verteilt.

Dahinter steckt das Bestreben, die Datensätze möglichst gleichmäßig auf die Partitionen zu verteilen, sodass mit parallelen Scans so effizient wie möglich zugegriffen werden kann.

Subpartitionierung

Wenn aus irgendeinem Grund eine Ebene von Partitionierung nicht genug ist, dann kann man eine zweite Ebene, ge-

nannt Subpartitionierung, einziehen. Die Idee hierbei ist, dass man zum Beispiel die Vorteile der Range-Partitionierung für die Administration mit den Vorteilen der Hash-Partitionierung für die parallele Datenverarbeitung kombinieren kann. Grundsätzlich kann man die Partitionierungsarten Range, List und Hash beliebig miteinander kombinieren. Ob alle Kombinationen sinnvoll sind, sei dahingestellt.

Automatismen

Traditionell mussten neue Partitionen beispielsweise für eine zeitbasierte Range-Partitionierung manuell vorkreiert werden. Meist existiert eine MAXVALUE-Partition für alle Datensätze mit Werten, die außerhalb der definierten Wertebereiche liegen. Durch einen SPLIT der MAXVALUE-Partition entsteht eine neue Partition oberhalb der bestehenden Partitionen. Falls man das Erstellen der neuen Partition nicht rechtzeitig durchführt, ist die MAXVALUE-Partition selbst mit Daten gefüllt. Der SPLIT ist dann weiterhin möglich, aber sehr aufwendig, da die Datensätze physisch bewegt werden müssen.

Mit der Intervallpartitionierung definiert man ein Wertebereich oberhalb der größten bestehenden Partition. Eine MAXVALUE-Partition darf es nicht geben. Durch die Intervallpartitionierung werden nun automatisch neue Partitionen angelegt, sobald Werte für einen hierdurch definierten Wertebereich vorhanden sind.

In ähnlicher Art und Weise sorgt die automatische List-Partitionierung dafür, dass eine neue Partition angelegt wird, wenn die Werteliste erweitert wird und zum Beispiel eine neue Farbe hinzukommt.

Eigenschaften von Partitionen

Ein großer Vorteil von Partitionierung ist, dass Partitionen unterschiedliche Storage-Eigenschaften haben können. So können Partitionen einer Tabelle in verschiedenen Tablespaces liegen. Damit können beispielsweise alte und nicht mehr häufig benutzte Partitionen auf günstigeren und nicht ganz

so schnellen Storage verschoben werden. Falls die Advanced-Compression-Option vorhanden ist, können diese älteren Partitionen auch stärker komprimiert werden.

Auch LOB-Partitionen können unterschiedliche Eigenschaften haben, etwa die Speicherung als BasicFile oder SecureFile LOBs. Hier ergibt sich die Möglichkeit, die Migration auf SecureFile LOBs partitionsweise durchführen zu können.

Partitionen der gleichen Tabelle können jedoch nicht in Tablespaces mit unterschiedlicher Blockgröße liegen.

Partitionen können einzeln verwaltet werden. Beispiele hierfür sind: Export einzelner Partitionen sowie Move, Truncate usw.

Indexpartitionierung

Oft wird bei Partitionierungskonzepten nicht darüber gesprochen, ob und wie Indizes partitioniert werden. Dabei ist dieser Aspekt entscheidend für den Erfolg der Partitionierung.

Zunächst ist festzustellen, dass Indizes nicht partitioniert werden, wenn dies beim CREATE INDEX nicht explizit angefordert wird. Demnach sind Indizes standardmäßig nicht partitioniert.

Sollen Indizes partitioniert werden, unterscheidet man zwei Fälle:

1. Wenn die Partitionierungskriterien identisch zur Tabelle sind, spricht man von einem lokal partitionierten Index. Um dies zu erreichen, fügt man LOCAL zum CREATE INDEX hinzu.
2. Wenn der Index nach eigenen Kriterien partitioniert werden soll, spricht man von globaler Partitionierung. Hierzu erhält der Index seine eigene PARTITION-Klausel.

Der größte Nachteil von nicht beziehungsweise global partitionierten Indizes gegenüber den lokal partitionierten ist, dass beim DROP PARTITION für eine Tabelle nicht einfach die passende Indexpartition mitgelöscht wird, sondern die Datensätze wie beim DELETE aus dem Index oder den Indexpartitionen entfernt werden müssen.

Voraussetzungen

Es sei vorab festgestellt, dass Partitionierung nur ab einer gewissen Tabellengröße Sinn ergibt. Aus meiner Sicht liegt diese Grenze bei einigen 100GB. Dadurch dass bei LOBs gewisse Operationen erheblich länger dauern (z.B. Export/Import), kommen insbesondere Tabellen mit LOB-Attributen in die Betrachtung.

Die zweite Bemerkung: Ein Partitionierungskonzept muss unbedingt getestet werden. Meist ist die Einführung von Partitionierung aufwändig. Sollte die Partitionierung nicht gut funktionieren, so wird man diese so schnell auch nicht wieder los. Also müssen das Verhalten der Anwendung sowie die geplanten Administrationsverfahren (z.B. Löschen alter Partitionen) ausgiebig getestet werden.

OLTP- vs. DWH-Anwendungen

Ein grundsätzlicher Unterschied zwischen OLTP-artigen und DWH-artigen Anwendungen ist, dass man im ersten Fall praktisch ausschließlich indizierte Zugriffe haben möchte und Full Table Scans, insbesondere mit paralleler Ausführung, unter allen Umständen vermeidet. Im DWH-Fall weiß man, dass je nach Aufgabenstellung Full Table Scans erforderlich sind. Hierfür ist Parallelverarbeitung ein willkommenes Mittel.

Die Hash-Partitionierung unterstützt insbesondere die Parallelverarbeitung im großen Stil, sodass man diese in den meisten Fällen nur dann einsetzen sollte, wenn man diese Art der Verarbeitung auch haben will – also in DWH-artigen Anwendungen. Andersherum: Hash-Partitionierung ergibt bei OLTP-artigen Anwendungen oft nur wenig Sinn.

Administrierbarkeit

Ein sehr smartes Feature ist das Reference-Partitioning. Hierbei wird eine Detail-Tabelle (also eine mit einem Fremdschlüssel auf die Master-Tabelle) nach dem Kriterium der Master-Tabelle partitioniert. Eine Tabelle mit Auftragspositionen könnte also, genau wie die Auftragsstabelle, nach Auftragsdatum monatsweise partitioniert sein, auch wenn das Auftragsdatum bei den Auftragspositionen gar nicht vorhanden ist.

```
WHERE datum BETWEEN TO_DATE('01.03.2019') AND TO_DATE('25.05.2019')
```

Listing 1

Hierdurch bekommt man diverse Vorteile: Joins zwischen den Tabellen können per Definition partitionsweise durchgeführt werden, das Löschen einer Partition bei den Aufträgen führt implizit zur Löschung der passenden Partition bei den Auftragspositionen.

Es gibt jedoch einen Nachteil: Der Fremdschlüssel, der die Grundlage der Partitionierung ist, kann nicht mehr abgeschaltet oder gelöscht werden. Damit sind gegebenenfalls Reorganisationsmaßnahmen nicht mehr ohne Weiteres möglich.

Daher müssen für ein solches Feature die (hervorragenden) Vorteile gegenüber den Nachteilen gründlich abgewogen werden.

Performance Feature No 1: Partition Pruning

Mit Partition Pruning wird die Möglichkeit des Optimizers bezeichnet, abhängig von den mitgelieferten Werten in WHERE-Bedingungen nur auf die in Frage kommenden Partitionen zuzugreifen, wenn beispielsweise bei einer monatsweise partitionierten Tabelle die Bedingung aus *Listing 1* enthalten ist:

Dann würde auf die Partitionen für März, April und Mai 2019 zugegriffen, auf alle anderen Partitionen – egal wie viele – nicht!

Im Umkehrschluss gilt aber, dass Partition Pruning nicht stattfindet, wenn das Partitionierungskriterium nicht in der WHERE-Klausel vorkommt!

Daher gilt als eine der wichtigsten Regeln für ein Partitionierungskonzept, dass das Partitionierungskriterium in einem Großteil der SQL-Befehle als Filter vorkommen muss. Dies ist insbesondere für Indizes wichtig: Indexpartitionen verhalten sich wie eigenständige Indizes. Findet für einen Indexzugriff kein Partition Pruning statt, dann müssen so viele Indexzugriffe durchgeführt werden, wie Indexpartitionen vorhanden sind. Da dies bis zu Hundert sein können, wird sich die Antwortzeit für Abfragen durch ein falsch gewähltes Partitionierungskriterium stark verschlechtern.

Partitionierte eindeutige Indizes

Eindeutige Indizes (oder nicht eindeutige, die für Primärschlüssel-Constraints ver-

wendet werden) können nur dann partitioniert sein, wenn die Indexspalten Teil des Partitionierungskriteriums sind. Dies ist der Tatsache geschuldet, dass andernfalls die Prüfung aller Partitionen für den Eindeutigkeits-Check notwendig wäre, was fatal für die INSERT-Performance wäre.

Hier muss oft davon Abstand genommen werden, möglichst lokal partitionierte Indizes zu verwenden, stattdessen benutzt man – je nach sonstigen Anforderungen – nicht partitionierte oder globale Indizes. Das Löschen von Tabellenpartitionen wird erheblich langsamer.

Wie viele Partitionen sollen es sein?

Zu dieser Frage gibt es naturgemäß keine exakte Antwort. Typischerweise sind einige Dutzend bis einige Hundert Partitionen für eine Tabelle in Ordnung. Bei Tausenden oder gar Zehntausenden Partitionen geht oft die Administrierbarkeit verloren. Konkret hat sich in einer produktiven Datenbank ein DROP TABLE für eine Tabelle mit mehr als 70.000 Partitionen als nicht durchführbar erwiesen. Hierfür wurde dann Partition für Partition gelöscht; die Laufzeit erstreckte sich über viele Stunden.

Eine grundsätzliche Empfehlung ist, die Partitionierung nicht zu feingranular zu halten, also eher Monats- oder Quartalspartitionen als Stundenpartitionen.

Partitionen und Statistiken

Bei der Erstellung von Statistiken unterscheidet Oracle Statistiken auf Tabellen- und auf Partitionsebene. Diese werden unabhängig voneinander berechnet. Bei der Berechnung der Statistiken auf Tabellenebene ist ein Full Scan auf der Gesamtdatenmenge durchzuführen – bei wirklich großen Tabellen ist dies aber oft gar nicht möglich.

Hierfür gibt es die Statistikpräferenz INCREMENTAL, die auf TRUE gesetzt werden kann. Dann werden Statistiken nur noch auf Partitionsebene berechnet; die Partitionsstatistiken werden dann in die globalen (Tabellen-)Statistiken hineingerechnet. Hierzu müssen jedoch mehr Details über die Statistiken gespeichert werden.

Vorsicht: Nach dem Ändern von INCREMENTAL müssen alle Statistiken neu berechnet werden!

What's new?

Ein lang ersehntes Feature ist mit Version 18c implementiert worden: Die Partitionierungsstrategie einer Tabelle kann nun mit einem ALTER TABLE ... PARTITION geändert werden. Dies ermöglicht auch die Partitionierung einer vorher nicht partitionierten Tabelle.

Die Änderung kann online erfolgen. Man benötigt während des Vorgangs den Platz für Daten und Indizes zweimal, da die neue Struktur parallel zur bestehenden erstellt wird.

Indizes werden lokal partitioniert, wenn die Indexspalten Präfix des Partitionierungskriteriums sind, ansonsten bleiben sie global. Mit einer UPDATE-INDEXES-Klausel kann man jedoch eigene Partitionierungskriterien für Indizes vorgeben.

Je nach Tabellengröße wird die Laufzeit für diese Aktion immens hoch sein; daher die Empfehlung, dies in einer Testumgebung ausführlich zu testen.

Mit 19c kann es hybrid partitionierte Tabellen geben, es können also Partitionen aus externen und internen Inhalten gemischt werden. Die externen Partitionen verhalten sich dabei etwa so wie die bereits bekannten externen Tabellen. Dies kann zum Beispiel nützlich sein, wenn ältere Daten nur noch in Dateiform vorliegen.

Fazit

Partitionierung bietet immense Möglichkeiten bei großen bis hin zu riesigen Tabellen. Eine gute Kommunikation zwischen Datenbankadministratoren, Datenmodellierern und Anwendungsentwicklern ist notwendig, um die Möglichkeiten der Partitionierung auszuschöpfen und das Projekt „Partitionierung“ erfolgreich zu machen.



Dierk Lenz
dierk.lenz@hl-services.de

Das Design von Formularen

Maximilian Liesegang, esentri AG

Formulare sind der Mittelpunkt fast aller Interaktionen. Sie stehen zwischen uns und dem, was wir versuchen zu erledigen. Täglich sind wir mit einer großen Anzahl von Formularen konfrontiert, wenn wir zum Beispiel privat oder beruflich mit IT-Anwendungen eine Tätigkeit durchführen wollen. Egal ob man ein Ticket kaufen, ein Hotelzimmer buchen oder etwas online bestellen möchte, höchstwahrscheinlich musste man währenddessen ein Formular ausfüllen. Formulare sind aber nur ein Mittel zum Zweck. Die Benutzer sollten in der Lage sein, sie schnell und verwirrungsfrei abzuschließen. In diesem Artikel lernen Sie praktische Tipps kennen, die Ihnen helfen, ein effektives Formular zu gestalten.

Formulare sind bescheiden. Es ist einfach ihr Ziel, ausgefüllt zu werden. Zwei Faktoren haben dabei einen großen Einfluss auf die Rate, in der Menschen dieses Ziel erfüllen. Zum einen die Wahrnehmung der Komplexität und zum anderen die Interaktionskosten eines Formulars.

Das erste, was Benutzer tun, wenn sie ein neues Formular sehen, ist abzuschätzen, wie viel Zeit zum Ausfüllen insgesamt benötigt wird. Benutzer tun dies, indem sie das Formular überfliegen. Die Wahrnehmung spielt eine entscheidende Rolle im Prozess der Schätzung. Je komplexer ein Formular aussieht, desto wahrscheinlicher ist es, dass der Benutzer den Prozess abbrechen wird.

Die Interaktionskosten sind die Summe der kognitiven und auch der physischen Anstrengungen, die Benutzer in die Interaktion mit dem Formular stecken, um ihr

Ziel zu erreichen. Sie haben einen direkten Zusammenhang mit seiner Benutzerfreundlichkeit. Je mehr Aufwand die Benutzer aufwenden müssen, um ein Formular auszufüllen, desto weniger benutzbar ist das Formular. Hohe Interaktionskosten können das Ergebnis von erschwerter Eingabe, der Unfähigkeit, die Bedeutung einiger Fragen zu verstehen, oder Verwirrung über Fehlermeldungen sein.

Diese Faktoren sind durch folgende Grundsätze positiv beeinflussbar: Reduziert die kognitive Belastung, macht es schwerer, Fehler zu begehen, und macht es menschlicher.

Schlechte Formulare kosten viel Geld

Diese Erfahrung durfte das amerikanische Online-Reisebüro Expedia machen.

Wie üblich, suchten und fanden Kunden auf ihrer Website die gewünschten Hotels, gaben ihre Rechnungsinformationen ein und drückten auf die „Kaufen“-Schaltfläche. Expedia konnte jedoch die große Mehrheit dieser Zahlungen nicht durchführen. Was war der Grund dafür?

Analysten begannen, die fehlgeschlagenen Zahlungen zu untersuchen, um festzustellen, welche Eigenschaften sie gemeinsam hatten. Die Antwort, so stellte sich heraus, war ganz einfach. Das Web-Formular hatte unter dem Feld „Name“ ein optionales Feld „Firma“. Diese Konstellation verwirrte einige Kunden, die das Feld „Firma“ mit ihrem Banknamen ausfüllten. Nach der Eingabe ihres Banknamens gaben diese Kunden dann in das Adressfeld die Adresse ihrer Bank und nicht ihre Privatadresse ein. Als es um die Adressverifizierung zur Verarbeitung der Zahlung ging, scheiterte diese,



Abbildung 1: Teilen Sie den Prozess auf mehrere Seiten auf (Quelle: Adam Silver) [1]

weil in dem Feld nicht die Adresse des Kreditkarteninhabers eingetragen wurde. Nachdem Expedia dies bemerkte, entfernten sie das entsprechende Feld sofort. Über Nacht stieg die Zahl der erfolgreichen Transaktionen an. Nach Angaben der Firma konnte so pro Jahr 12 Millionen Dollar mehr Umsatz gemacht werden.

Reduziert die kognitive Belastung

Im folgenden Abschnitt finden sich vier Tipps, um die kognitive Belastung der Nutzer zu reduzieren. So können die Interaktionskosten und die Komplexität der Formulare vermindert werden.

Eine Sache pro Seite bedeutet, einen komplizierten Prozess in kleine Stücke zu zerlegen und diese auf eine einzelne Seite zu verteilen. Anstatt beispielsweise Lieferadresse, Lieferoptionen und Zahlungsformulare auf einer langen Seite darzustellen, werden sie auf separaten Seiten dargestellt. Es geht dabei nicht zwingend darum, auf jeder Seite ein einziges Formularfeld zu haben, obwohl das auch möglich ist. Durch den ständigen Fortschritt wird die Vervollständigungsrate der Benutzer gefördert. Außerdem kann mit Fortschrittsbalken die Unsicherheit der Benutzer reduziert werden. Diese Balken schaffen Transparenz darüber, wie weit man von der Vervollständigung entfernt ist. Ein kleiner Hack ist es, den Fortschrittsbalken initial nicht bei 0% starten zu lassen. Untersuchungen haben ergeben, dass so die Vervollständigungsrate gesteigert werden kann.

Formularfelder gehören in eine Spalte. Tests zeigen, dass mehrspaltige Formularlayouts anfällig für Fehlinterpretationen sind. Die Folgen sind, dass Benutzer Felder überspringen, in denen sie tatsächlich Daten eingeben müssen, Daten in die falschen Felder eingeben oder einfach zum Stillstand kommen und rätseln, wie sie mit der Eingabe fortfahren sollen. Anstatt eine visuelle Neuorientierung des Benutzers zu erfordern, halten Sie ihn im Fluss, indem Sie sich an eine einzelne Spalte mit einer separaten Zeile für jedes Feld halten. Ausnahmen von dieser Regel gibt es auch. Kurze und logisch zusammen-

hängende Felder wie Stadt, Bundesland und Postleitzahl können in derselben Zeile angezeigt werden.

Es ist empfehlenswert, Beschriftungen über die entsprechenden Formularfelder zu setzen. Obwohl dies die Gesamtlänge des Formulars verlängert, erleichtert es das Überfliegen des Formulars, da der Benutzer das Feld in der gleichen Linie wie die Feldbeschriftungen sehen kann. Außerdem erspart es den Augen des Benutzers, zwischen der Beschriftung und dem Eingabefeld hin- und herzuspringen (siehe hierzu *Abbildung 3*). Diese Platzierung ermöglicht auch längere Feldbeschriftungen, da horizontaler Abstand kein Problem darstellt. Wenn es um die Formularlänge geht, kann man links neben den Textfeldern Feldbezeichnungen platzieren. Achten Sie jedoch in diesem Fall darauf, dass die Etiketten gleich lang sind und so nah wie möglich an den Textfeldern platziert werden. Wenn die Etiketten zu weit links liegen, kann es schwierig sein, das richtige Etikett dem entsprechenden Feld zuzuordnen. Wenn die Nähe ein solches Problem ist, warum platziert man dann nicht die Beschriftungen in den Feldern? Wenn man die Beschriftung als Platzhalter in das Formularfeld einfügt, verschwindet es, wenn Benutzer ihren Text eingeben. So müssen sie es sich merken, wenn sie das Feld ausfüllen wollen. Dies führt insbesondere dann zu einem Problem, wenn Benutzer mit der Tabulatortaste durch ein Formular navigieren. Wenn sie mit der Tabulatortaste zum nächsten Feld navigieren, ohne zu schauen, verpassen sie, welche Informationen benötigt werden.

Dropdowns können bei richtiger Verwendung sehr hilfreich sein. Sie sparen

Platz auf dem Bildschirm und verhindern, dass Benutzer fehlerhafte Daten eingeben, da sie nur zulässige Optionen zeigen. Sie verfügen über viele nützliche Funktionen wie Gruppierungen und eine leichte Tastatur-Navigation. Allerdings werden Dropdowns auch sehr häufig in den falschen Situationen eingesetzt. Der Benutzer leidet initial unter einem Mangel an Informationen. Er muss erst klicken, um alle verfügbaren Optionen zu sehen. Solange es fünf oder weniger Einträge gibt, sollte man eine Radio-Komponente verwenden. Mit dieser können Benutzer sofort sehen, wie viele Optionen sie haben und was diese Optionen sind, ohne eine weitere Aktion durchführen zu müssen. Besonders Ihre mobilen Nutzer werden es Ihnen danken. Auf der anderen Seite der Skala gibt es auch zu viele Optionen für ein Dropdown. Wenn es größer als 15 Optionen ist, wird es für den Benutzer schwierig, die Informationen zu scannen und sie zu navigieren. Riesige Dropdown-Listen können für Ihre Benutzer ein echter Alptraum sein, da sie anfangen müssen, innerhalb der Liste zu scrollen, weil sie den Teil der Optionen außerhalb des Sichtbaren verbergen. Diese miserable Benutzererfahrung verlangsamt den gesamten Prozess. Stattdessen sollte man bei langen Listen das Dropdown um ein Eingabefeld erweitern. Mit diesem Feld filtert der Benutzer die Einträge und kommt so schnell und zuverlässig an sein Ziel.

Macht es schwerer, Fehler zu begehen

Benutzer werden Fehler machen. Es ist unvermeidlich. Deshalb ist es wichtig, ein

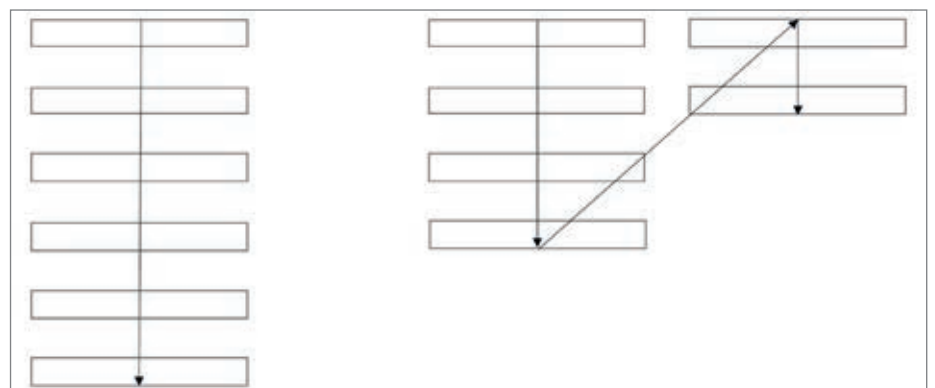


Abbildung 2: Einspaltige Layouts, wie links, sind angenehmer (Quelle: Maximilian Liesegang)

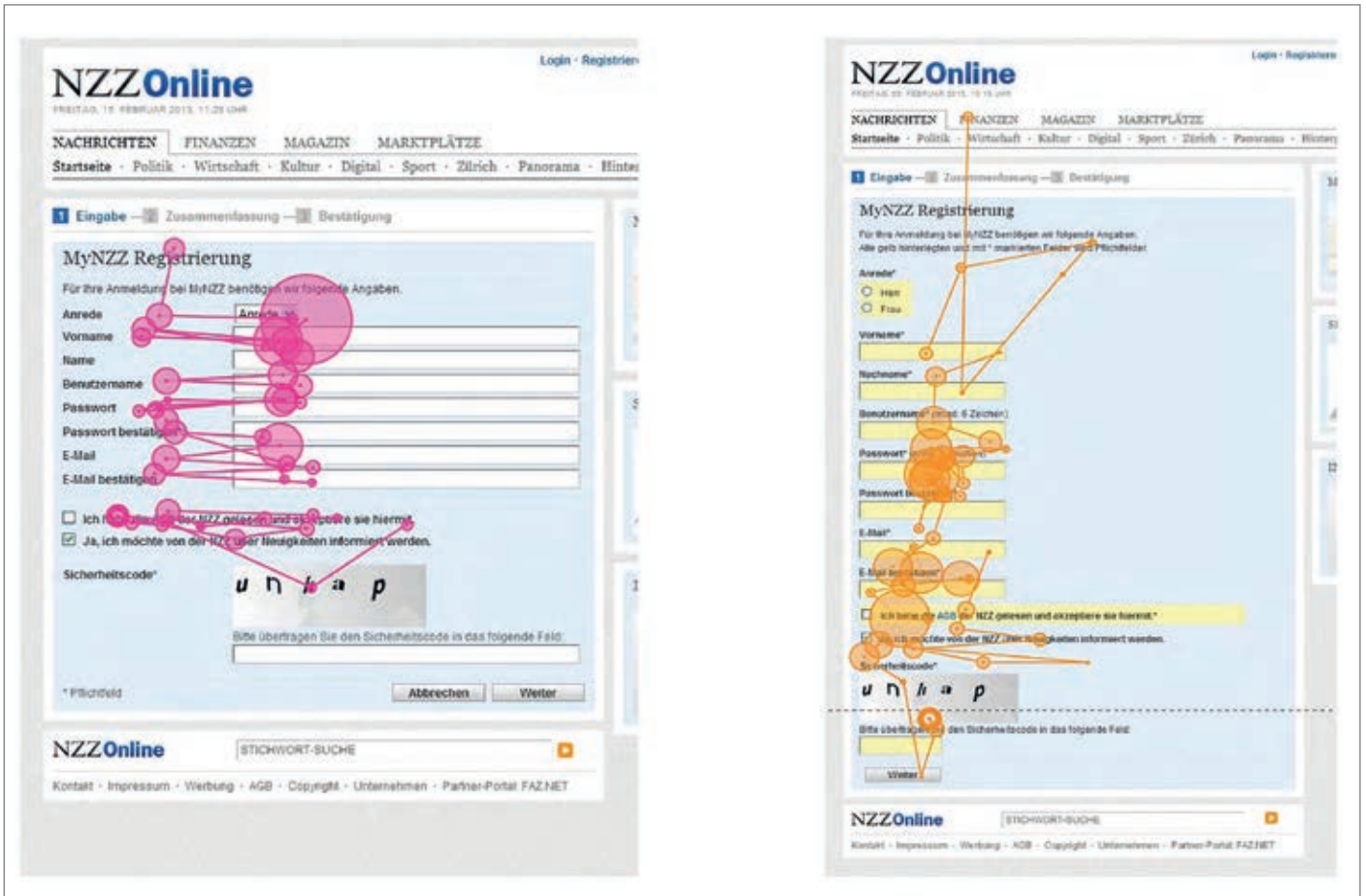


Abbildung 3: Eye-Tracking-Ergebnisse im Vergleich von Beschriftungen (Quelle: Mirjam Seckler et al.) [2]

Formular zu entwerfen, das die Benutzer in Momenten, in denen Probleme auftreten, unterstützt. Da das Thema „Fehler und Validierung“ einen eigenen Artikel verdient, sind im Folgenden nur zwei Dinge aufgelistet, die zu tun sind, um den Benutzern zu helfen.

Indem Sie sinnvolle Standardantworten auf Fragen bereitstellen, sparen Sie den Benutzern kognitive Arbeit. Er muss nicht mehr den Aufwand betreiben, über die Antwort nachzudenken oder sie einzugeben. Das Ausfüllen von Formularen macht nie Spaß, aber wenn dieses Muster die Zeit halbiert, die ein Benutzer braucht, um es durchzuarbeiten, wird

er dankbar sein. Möglicherweise werden die meisten Benutzer auf eine bestimmte Weise antworten. Oder der Benutzer hat bereits genügend Kontext-Informationen für die Benutzeroberfläche bereitgestellt, um genau vorherzusagen, welche Antwort er geben wird. Wenn beispielsweise jemand eine Postleitzahl eingibt, können Sie aus genau dieser Nummer die Stadt ableiten. Genauso kann man den Namen einer Person bei Adressen oder Zahlungsinformationen wiederverwenden. Bei semi-relevanten Fragen kann man vielleicht nicht erwarten, dass der Benutzer die Antwort weiß oder sich um sie kümmert, und „was

auch immer das System vorgibt“, reicht völlig aus. Wählen Sie aber keinen Standardwert, nur weil Sie der Meinung sind, dass Sie kein Feld leer lassen sollten. Tun Sie dies nur, wenn Sie sich ziemlich sicher sind, dass die meisten Benutzer dieses nicht ändern werden. Sonst wird man natürlich unnötigerweise zusätzliche Arbeit schaffen.

Ein kurzer Hilfetext in vollständigen Sätzen, der dauerhaft an der Oberfläche sichtbar ist und der eine Funktion oder das Ergebnis von Aktionen erklärt, kann ein sehr hilfreiches Konzept sein. Es wird neuen Benutzern helfen, die Dinge besser zu verstehen, und beste-



Abbildung 4: Links eine Radio-Komponente, rechts ein Input-Dropdown (Quelle: Maximilian Liesegang)



Abbildung 5: Komplette Sätze als Hilfetext sind sehr hilfreich (Quelle: Maximilian Liesegang)

hende Benutzer bei ihrer Arbeit unterstützen. Überraschenderweise haben viele Anwendungen heutzutage extrem kurze Bezeichnungen ohne Erklärungen neben ihren Formularfeldern. Wenn die Benutzer das Feld jedoch nicht verstehen, sind sie komplett verloren. Eine weitere Ausprägung der Hilfetexte sind präzisere Bezeichnungen der Formularfelder. So sollte man in einem Login-Formular das Feld „Benutzername“ durch „E-Mail-Adresse“ ersetzen, sollte das System nur E-Mail-Adressen zum Login benutzen. Allgemein gilt es, vieles, was implizit ist, explizit zu machen und auszuschreiben. Die Nutzer werden es Ihnen danken.

Gestaltet es menschlicher

Es ist wichtig, dass Formulare barrierefrei sind, um allen Benutzern gleichen Zugang und gleiche Chancen zu bieten. Ein zugängliches Formular kann Menschen mit Einschränkungen helfen, sich aktiver als je zuvor zu beteiligen.

Verlassen Sie sich bei der Kommunikation nicht auf Farbe. Es gibt eine große Anzahl von Menschen, die einen gewissen Grad an Farbenblindheit besitzen. Achten Sie deshalb bei der Anzeige von Validierungsfehlern oder Erfolgsmeldungen darauf, dass das Feld nicht ausschließlich grün oder rot markiert ist. Wo immer Farbe verwendet wird, versuchen Sie auch zusätzlich Text und Symbole anzuzeigen, um dem Benutzer die entsprechende Nachricht zu übermitteln.

Das „Required“-Attribut aus HTML sorgt dafür, dass viele Formulare das Symbol * benutzen. Aber die Formulierung „Optional“ nach einer Beschriftung ist viel klarer als jedes visuelle Symbol, das Sie je verwenden könnten. Es wird immer Nutzer geben, die sich fragen, was dieses Sternchen bedeutet. Und diese Nutzer werden nach einer Legende suchen, die Dinge erklärt.

Stellen Sie sicher, dass Ihr gesamtes Formular mit der Tabulatortaste navigierbar ist. Während viele Benutzer die Tabulatortaste verwenden, um schneller durch Formulare zu navigieren, ist dies besonders wichtig für manche Benutzer, die sich auf eine Software verlassen, die die Tabulatorfunktion verwendet, um von einem Feld zum nächsten zu wechseln.

Wenn Sie Fehlermeldungen schreiben, konzentrieren Sie sich darauf, die Frustration zu minimieren, die Benutzer empfinden, wenn sie ein Problem mit einem Formular haben. Hier sind ein paar Regeln zum Schreiben effektiver Fehlermeldungen: Die Art und Weise, wie Sie eine Fehlermeldung übermitteln, kann einen enormen Einfluss darauf haben, wie Benutzer sie wahrnehmen.

Eine Fehlermeldung wie „Sie haben eine falsche Nummer eingegeben“ gibt dem Benutzer die ganze Schuld, was dazu führen kann, dass der Benutzer frustriert ist. Schreiben Sie stattdessen etwas, das neutral oder positiv klingt. Eine neutrale Meldung könnte sein: „Diese Zahl ist falsch.“ Des Weiteren vermeiden Sie Fehlermeldungen wie „User Input Error: 1034“. Diese sind kryptisch und beängstigend. Schreiben Sie wie ein Mensch, nicht wie ein Roboter. Verwenden Sie menschliche Sprache und erklären Sie, was genau der Benutzer oder das System falsch gemacht hat und was genau der Benutzer tun sollte, um das Problem zu lösen.

Fazit

Wie wir sehen können, ist es nicht leicht, ein gutes Formular zu entwerfen. Riskieren Sie nicht, dass Benutzer enttäuscht werden oder wertvolle Zeit damit verschwenden, herauszufinden, wie Ihr Formular funktioniert. Der Autor hofft, mit dem Artikel ein paar Tipps gegeben zu haben, die es Ihnen in Zukunft erleich-

tern, effektivere Formulare zu gestalten. Es ist nicht so, dass man besonders künstlerisch begabt sein muss, um die Oberfläche eines Formulars besser zu machen. Das Wichtigste ist, dass man bei der Entwicklung von Anwendungen niemals vergisst, dass Menschen diese am Ende benutzen sollen. Zumindest solange wir die Singularität noch nicht erreicht haben.

Quellen

- [1] Adam Silver (2017): One Thing Per Page. Smashing Magazine <https://www.smashingmagazine.com/2017/05/better-form-design-one-thing-per-page>
- [2] Mirjam Seckler et al. (2014): Designing Usable Web Forms – Empirical Evaluation of Web Form Improvement Guidelines. Conference on Human Factors in Computing Systems



Maximilian Liesegang
maximilian.liesegang@esentri.com



Wir begrüßen unsere neuen Mitglieder

Persönliche Mitglieder

- › Andre Homberg
- › Katharina BÜchse
- › José-Luis Quiro Zamora
- › Manfred Schmid
- › Uwe Sponholz

Firmenmitglieder DOAG

- › PROSOZ Herten GmbH, Bernhard Steffen

Termine



September

02. + 03.09.2019
Go DevOps 2019
 Berlin

05.09.2019
Regionaltreffen Karlsruhe
 Reiner BÜnger & Jochen Kutscheruk

10.07.2019
Regionaltreffen Berlin/Brandenburg
 Michel Keemers
 Berlin

13.09.2019
DOAG Datenbank Webinar
 online

16.09.2019
Regionaltreffen München/Südbayern
 Andreas Ströbel

17. + 18.09.2019
Berliner Expertenseminar: Die Exadata effizient nutzen?
 Dozent: Lothar Flatz
 Berlin

17.09.2019
Regionaltreffen Hannover
 Andreas Ellerhoff

17.09.2019
DOAG Logistik + IT
 Frankfurt am Main

19.09.2019
Regionaltreffen Rhein-Neckar
 Frank Stöcker
 Mannheim

20.09.2019
NextGen Webinar: Abstimmung von Organisationsform und Systemarchitektur für eine erfolgreiche Produktentwicklung
 online

25.09.2019
DOAG Data Centric Day
 Christian Schwitalla
 Köln

27. + 28.09.2019
Vorstandssitzung 3/2019
 DOAG-Vorstand
 Flonheim-Uffhofen



Oktober

07.10.2019
Regionaltreffen Bielefeld/Münster/Osnabrück
 Andreas Kother & Klaus Günther

10.10.2019
Regionaltreffen NRW
 Martin Schmitter & Torsten Rosenwald

11.10.2019
DOAG Datenbank Webinar: Direkte Anbindung von Active Directory (AD) an die Oracle 18C Datenbank (Single Sign On)
 Dozent: Dirk Krautschick
 online

15. + 16.10.2019
Berliner Expertenseminar: Oracle Cloud Infrastructure
 Dozent: Martin Berger
 Berlin

15.10.2019
DOAG Security Day
 Bruno Cirone
 Hamburg

Impressum

Red Stack Magazin wird gemeinsam herausgegeben von den Oracle-Anwendergruppen DOAG Deutsche ORACLE-Anwendergruppe e.V. (Deutschland, Tempelhofer Weg 64, 12347 Berlin, www.doag.org), AOUG Austrian Oracle User Group (Österreich, Lassallestraße 7a, 1020 Wien, www.aoug.at) und SOUG Swiss Oracle User Group (Schweiz, Dornacherstraße 192, 4053 Basel, www.soug.ch).

Red Stack Magazin ist das User-Magazin rund um die Produkte der Oracle Corp., USA, im Raum Deutschland, Österreich und Schweiz. Es ist unabhängig von Oracle und vertritt weder direkt noch indirekt deren wirtschaftliche Interessen. Vielmehr vertritt es die Interessen der Anwender an den Themen rund um die Oracle-Produkte, fördert den Wissensaustausch zwischen den Lesern und informiert über neue Produkte und Technologien.

Red Stack Magazin wird verlegt von der DOAG Dienstleistungen GmbH, Tempelhofer Weg 64, 12347 Berlin, Deutschland, gesetzlich vertreten durch den Geschäftsführer Fried Saacke, deren Unternehmensgegenstand Vereinsmanagement, Veranstaltungsorganisation und Publishing ist.

Die DOAG Deutsche ORACLE-Anwendergruppe e.V. hält 100 Prozent der Stammeinlage der DOAG Dienstleistungen GmbH. Die DOAG Deutsche ORACLE-Anwendergruppe e.V. wird gesetzlich durch den Vorstand vertreten; Vorsitzender: Stefan Kinnen. Die DOAG Deutsche ORACLE-Anwendergruppe e.V. informiert kompetent über alle Oracle-Themen, setzt sich für die Interessen der Mitglieder ein und führen einen konstruktiv-kritischen Dialog mit Oracle.

Redaktion:

Sitz: DOAG Dienstleistungen GmbH
(Anschrift s.o.)
ViSdP: Mylène Diacquenod
Redaktionsleitung: Martin Meyer
Kontakt: redaktion@doag.org
Weitere Redakteure (in alphabetischer Reihenfolge): Lisa Damerow, Mylène Diacquenod, Marina Fischer, Sanela Lukavica, Robert Marz, Yann Neuhaus, Fried Saacke

Titel, Gestaltung und Satz:

Alexander Kermas
DOAG Dienstleistungen GmbH
(Anschrift s.o.)

Fotonachweis:

Titel: © Daniil Peshkov | <https://de.123rf.com>
S. 7: © juliatim | <https://de.123rf.com>
S. 16 Logo: © dotCloud, Inc. | <https://de.wikipedia.org>
S. 16 Hintergrund: © Harryarts | www.freepik.com
S. 20: © Andrea Danti | <https://de.fotolia.com>
S. 30: © Sergey Nivens | <https://de.123rf.com>
S. 38: © kasto | <https://de.fotolia.com>
S. 44: © sentavio | <https://de.123rf.com>
S. 50: © elenabsi | <https://de.123rf.com>
S. 55: © greyjj | <https://de.123rf.com>
S. 61: © neyro2008 | <https://de.123rf.com>
S. 66: © tuulijumala | <https://de.fotolia.com>
S. 73 Figuren ©: apinan | <https://de.123rf.com>
S. 73: Hintergrund: © Dzianis Kuryanovich | <https://de.123rf.com>

Anzeigen:

Simone Fischer,
DOAG Dienstleistungen GmbH
(verantwortlich, Anschrift s.o.)
Kontakt: anzeigen@doag.org
Mediadaten und Preise unter:
www.doag.org/go/mediadaten

Druck:

adame Advertising and Media GmbH,
www.adame.de

Alle Rechte vorbehalten. Jegliche Vervielfältigung oder Weiterverbreitung in jedem Medium als Ganzes oder in Teilen bedarf der schriftlichen Zustimmung des Verlags.

Die Informationen und Angaben in dieser Publikation wurden nach bestem Wissen und Gewissen recherchiert. Die Nutzung dieser Informationen und Angaben geschieht allein auf eigene Verantwortung. Eine Haftung für die Richtigkeit der Informationen und Angaben, insbesondere für die Anwendbarkeit im Einzelfall, wird nicht übernommen. Meinungen stellen die Ansichten der jeweiligen Autoren dar und geben nicht notwendigerweise die Ansicht der Herausgeber wieder.

Inserentenverzeichnis

B4BMEDIA.NET AG www.e3zine.com	S. 29	DOAG e.V. www.doag.org	S. 49, U 2	MuniQsoft Training GmbH www.muniqsoft-training.de	S. 3
dbi services sa www.dbi-services.com	S. 19	Logicalis GmbH www.de.logicalis.com	U 3	Trivadis AG www.trivadis.com	U 4
		MuniQsoft Consulting GmbH www.muniqsoft-consulting.de	S. 37		

Java-Support

ist kostenpflichtig!



Jetzt bei dem deutschen Oracle License Management Partner informieren:

- » Welche Kosten entstehen?
- » Wie ändert sich die Java Support Roadmap?
- » Was müssen Unternehmen jetzt tun?

WIR LEBEN DIGITALISIERUNG.



Die Digitale Transformation macht vor keiner Branche halt und verändert Wertschöpfungsketten und Strukturen auch Ihres Unternehmens. Sie müssen sich neuen Herausforderungen stellen. In andere Richtungen denken. Ihr Geschäftsmodell anpassen und weiterentwickeln, damit Sie auch in Zukunft wettbewerbsfähig bleiben. Wir sind mittendrin im Geschehen und gestalten partnerschaftlich Ihren Weg ins Zeitalter der Digitalisierung. Sprechen Sie mit uns.

m.trivadis.com/digitalisierung | info@trivadis.com

BASEL | BERN | BRUGG | BUKAREST | DÜSSELDORF | FRANKFURT A.M.
FREIBURG I.B.R. | GENÈVE | HAMBURG | KOPENHAGEN | LAUSANNE
MANNHEIM | MÜNCHEN | STUTTGART | WIEN | ZÜRICH

trivadis