

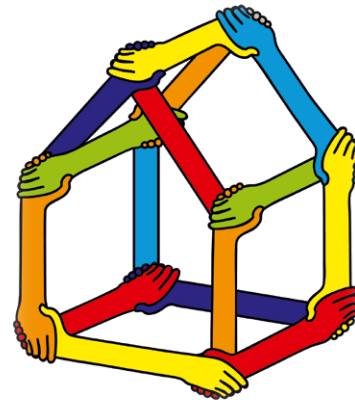


Releasemanagement mit BPMN

DevOps meets BPM

Masroor Ahmad, Liesa-Rebecca Thiel

DevOps bedeutet heute in der Umsetzung vor allem „Continuous Delivery“ und „Infrastructure as Code“. Bei komplexen Softwaresystemen stellt die Deployment-Pipeline jedoch Anforderungen an Koordination und Überwachung aller Akteure, die weit über bisherige Tool- und Abteilungsgrenzen hinaus gehen. Mit dem Einsatz einer Workflow-Engine öffnen sich neue Möglichkeiten der Integration von Entwicklung, Betrieb und Qualitätssicherung ganz im Sinne von DevOps 2.0.



Das Thema DevOps ist schon länger in den Entwickleretagen und Serverräumen präsent. Obwohl sich DevOps-Spezialisten und einschlägige Fachzeitschriften (vgl. [Pes12]) noch immer um eine Definition bemühen, lässt sich eine wichtige Kernaussage feststellen: Es ist gut für ein Unternehmen, wenn Entwicklungsabteilungen (*Development*) und Betrieb (*Operations*) näher zusammenrücken. Diese Aufgabe gelingt durch eine stärkere Kommunikation zwischen den Abteilungen und erfordert konsequent festgelegte Regeln und Prozesse.

Historisch betrachtet fügt sich der DevOps-Gedanke nahtlos in die stetige Weiterentwicklung des Softwareherstellungsprozesses ein, der bei den Entwicklern begann und nun auch im Betrieb ankommt. Standen am Anfang der Automatisierungsgrad und die technische Reife des Entwicklungsprozesses im Vordergrund, gesellten sich dank der agilen Bewegung ganz-

heitliche Forderungen im Rahmen des gleichnamigen Manifests dazu: „Menschen und Interaktionen sind wichtiger als Prozesse und Werkzeuge“. Dies führt zu neuen Herausforderungen bei der Umsetzung der technischen Infrastruktur.

Während die Automatisierung und Konsolidierung neuer Methoden und Konzepte fortschreitet („Continuous Delivery“-Strategie oder „Infrastructure as Code“), fand man bisher keine Lösung, um die Lücken der Kommunikation und Integration zwischen Tools und menschlichen Akteuren zu schließen. Wie in Abbildung 1 vereinfacht dargestellt, fehlt ein umfassendes Tool, das die Entwicklungs- und Auslieferungsprozesse steuert („Process as Code“).

DevOps als Geschäftsprozess

In einem Unternehmen gibt es menschen- und maschinengestützte Aktivitäten, die im Rahmen eines „Geschäftsprozesses“ ein Ergebnis erzielen möchten: Umsatz und Gewinn. Auch Vertreter der Blogger-Szene wie [Edw10] bestätigen, dass das Thema „DevOps“ kein technologisches Problem betrachtet, sondern ein „Geschäftsproblem“. Dies gibt Anlass, sich mit dem BPM-Ansatz auseinanderzusetzen – einem Feld, das bisher weitestgehend von Business-Analysten und Domänenexperten dominiert wird. „Softwareentwicklung“ hat als Geschäftsdomäne schon immer mit Entwicklern, Testern und Systemadministratoren als ihren Fachexperten existiert. Daher fördert das Studium der klassischen Disziplin der Geschäftsprozessmodellierung unter dem DevOps-Aspekt neue Erkenntnisse zutage.

Im Folgenden wird genauer untersucht, wie sich die einzelnen Maßnahmen konkret zum DevOps-Gedanken verhalten.

Risikominimierung durch maschinelle Koordination

Der Konflikt zwischen der Entwicklungsabteilung und dem Betrieb liegt in den jeweiligen Zielen und Zuständigkeiten begründet. Die Entwicklungsabteilung hat Interesse daran, möglichst schnell viele Änderungen produktiv zu schalten. Für den Betrieb hingegen ist jede Änderung am Produktivsystem mit Unsicherheiten verbunden, die es zu vermeiden gilt. Ein erster Ansatz zur Auflösung dieses Zielkonflikts besteht in der Reduzierung dieser. BPM kann helfen, menschliche Fehler zu minimieren, indem es die beteiligten Akteure von Beginn an „führt“. Durch die Process Engine werden die Mitarbeiter an ihre jeweiligen Aufgaben erinnert und die Erledigung wird, ähnlich wie

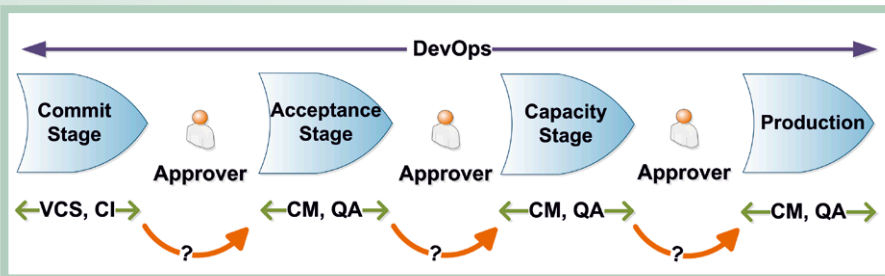


Abb. 1: Deployment-Pipeline mit Lücken in der Kommunikation

BPM, BPMN und die Process Engine

Beim *Business Process Management (BPM)* handelt es sich um einen systematischen Ansatz, um Prozesse zu modellieren, sie auszuführen und zu überwachen – mit dem Ziel, sie einfacher steuern und verbessern zu können.

Mit *Business Process Model and Notation (BPMN)* steht eine grafische Spezifikationsprache zur Verfügung, welche die Weitergabe der Prozessdefinitionen und damit die Kommunikation und toolgestützte Kollaboration zwischen den Prozessbeteiligten (Akteuren) erheblich vereinfacht.

Die *Business Process Engine* bezeichnet das Programm, das sowohl die Ausführung maschineller Aktivitäten, als auch die Koordination der menschlichen Akteure sicherstellt.

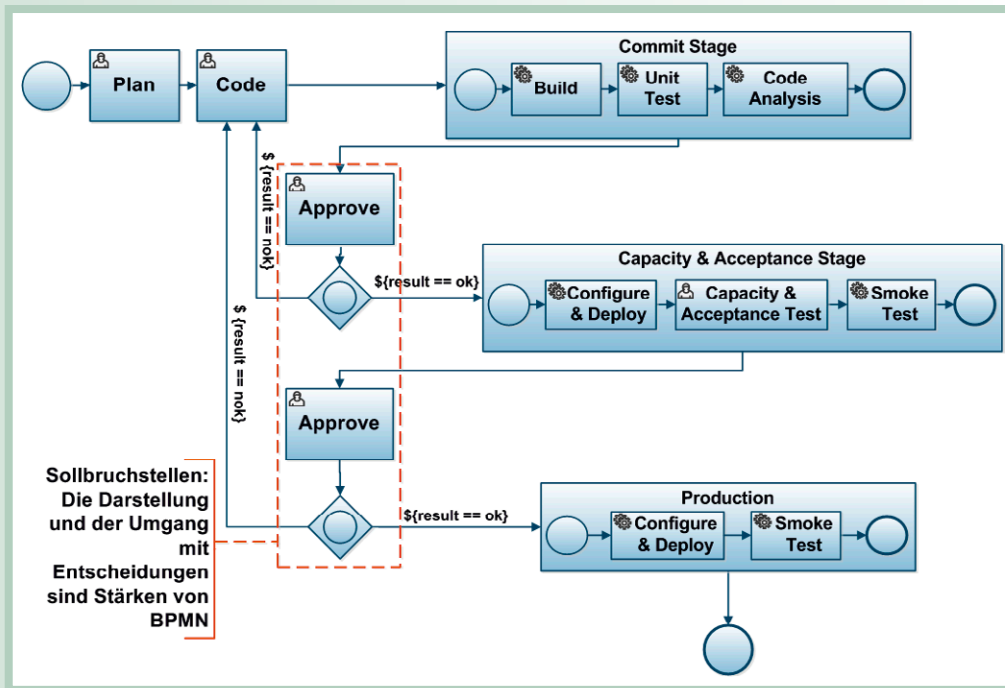


Abb. 2: Eine typische Deployment-Pipeline in BPMN

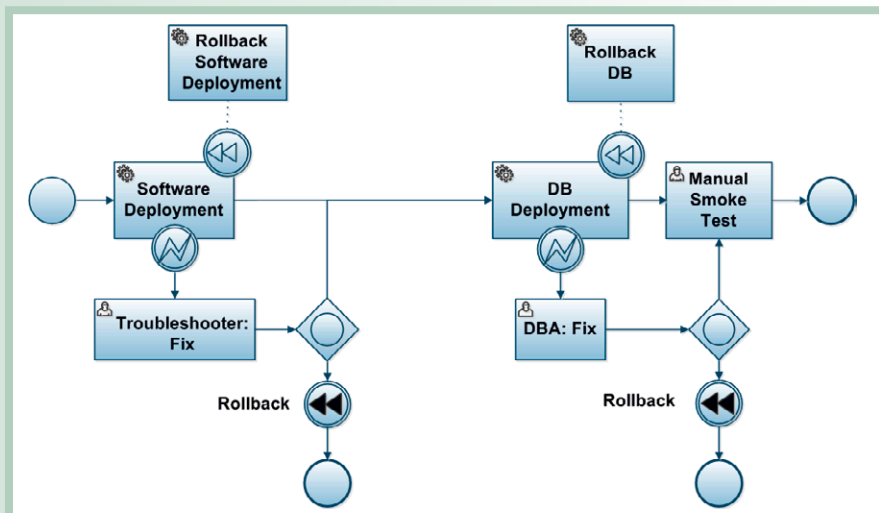


Abb. 3: Ein typischer DevOps-Prozess in BPMN mit Boundary Events und Compensation Tasks

bei einer ToDo-Liste, verfolgt und dokumentiert. Damit ist eine doppelte Ausführung ebenso ausgeschlossen wie die Nichterledigung. Die Teams können sich auf ihre Kernkompetenzen konzentrieren, und die Einhaltung unternehmensinterner Freigabeprozesse wird durch die Engine sichergestellt.

Eine typische Deployment-Pipeline aus dem Betrieb zeigt Abbildung 2 (auf Pools und Lanes wurde in dieser und allen folgenden Abbildungen der Einfachheit halber verzichtet). Manuelle Tasks wechseln sich mit maschinellen Tasks ab. Wenn der Kontrollfluss eine manuelle Task erreicht, werden alle zuvor festgelegten Akteure benachrichtigt. Die Erledigung einer Aufgabe muss der Engine zurückgemeldet werden, damit der Prozess fortschreiten kann.

„Blames“ durch Verantwortlichkeiten verhindern

Eine Lösung gegen das Problem der Unkultur gegenseitiger Schuldzuweisungen kann auch BPM nicht liefern. Doch es för-

dert die Agilität, um dem „Blame Game“ zwischen Dev und Ops entgegenzuwirken: Die meisten Fehlersituationen in produktiven Systemen entstehen während des Deployment-Vorgangs oder unmittelbar danach; bezogen auf einen Prozess, in dem beide Abteilungen gleichermaßen einwirken (vgl. [Hüt12]). Die Scrum- und Kanban-Boards der Unternehmen haben jedoch nur die Entwicklung der Software im Fokus.

Mit der Gestaltung von ereignisgetriebenen Workflows lässt sich ein agiles Board aufbauen, das alle wichtigen Akteure zusammenführt. Die Engine kann auf Grundlage aktueller Monitoringwerte eine präzise Definition der Akteure und Aktivitäten für den Start eines „Deeskalationsprozesses“ ermitteln. Haben sich

alle Teilnehmer auf einen Satz vertikaler Prozesse im Incident Management geeinigt, besitzt die Prozessdefinition eine stabile und konfliktmindernde Autorität über die zeitlichen Ressourcen der Teammitglieder von Entwicklung und Betrieb (s. Abb. 3). Auf diese Weise führt man einen geregelten Umgang mit Fehlern ein, dessen Notwendigkeit bereits DevOps-Pionier Flickr in [AllHam09] erkannt hat.

Automatisierungsstrategie

DevOps beschreibt die Automatisierung der Prozesse als wichtigen Stützpfiler für die reibungslose Zusammenarbeit der Abteilungen. Ohne die vollautomatische Ausführung von Build und Deployment ist eine Deployment-Pipeline nicht zu realisieren. Die Fehlerquote sinkt drastisch, wenn wiederkehrende und komplexe Aufgaben nicht mehr von Hand ausgeführt, sondern der Maschine überlassen werden.

Eine Automatisierung lässt sich jedoch, so die Autoren des Continuous-Delivery-Standardwerks [Hum10], niemals vollständig umsetzen. Es gibt Konstellationen, die menschliches Eingreifen erfordern. Beispiele dafür sind das explorative Testen oder die Freigabe eines Artefakts.

Doch die Erkenntnis, dass nicht alle Aktivitäten automatisch ablaufen können, steht nicht im Widerspruch zum Prozessgedanken. Die Sollbruchstellen, sprich: die manuellen Tasks in der Prozessdefinition, sind letztendlich Ausdruck einer ausgereiften BPM-Lösung, in der die Process Engine steuernde und integrierende Aufgaben besitzt. Zudem entscheidet das Abstraktionsniveau der jeweiligen Prozessbeschreibungen über den möglichen Grad der Automatisierung. Das caBPMN-Framework aus [FreRü12] beschreibt beispielsweise eine Methode, in der die Prozesse ausgehend vom strategischen Pro-

DevOps-Reports

- ▼ Wie lange braucht eine Codeänderung vom Check-in bis zur Produktivschaltung?
- ▼ Wie viele Hotfixes wurden letzten Monat ausgeliefert?
- ▼ Wie lange benötigen die Troubleshooter im Durchschnitt zur Behebung eines Fehlers in Produktion?
- ▼ Hat sich die Umstellung des Deployments auf ein neues Verfahren schon ausgezahlt?

zessmodell in die operative Ebene überführt werden, um dann in einem technischen Prozessmodell zu münden. Das operative Prozessmodell reduziert damit den Abstimmungsbedarf zwischen den Mitarbeitern und Abteilungen. Erst das technische Prozessmodell korreliert mit der IT-Spezifikation und enthält ausführbaren Code. Liegen die Prozessbeschreibungen in einer guten Qualität vor, lassen sich manuelle Tasks im Rahmen eines iterativen Vorgehensmodells sukzessive minimieren und, ohne Anpassung des gesamten Prozesses, durch maschinelle Tasks ersetzen.

Ermittlung von Metriken

Das „M“ aus dem DevOps C.A.M.S. (vgl. [Will10]) steht für „Measurement“ und beschreibt die Messbarkeit der involvierten Betriebsmittel und Prozesse. Die Erhebung von Metriken dient nicht nur als wichtiges Instrument des höheren Managements, sondern liefert auch den Entwicklern und Architekten wertvolle Informationen.

Auf Basis dieser Erkenntnis schlägt [ShDeGa] eine Klassifikation von Messzahlen vor, die im Softwaregeschäft typischerweise anfallen. Die Tools für die Erhebung von Service-Level-Metriken sind bekannt (z. B. Maintenance-Applikationen wie Zabbix), ebenso die Methoden zur Berechnung wirtschaftlicher Kennzahlen. Eine Ermittlung von Zahlen, die den „Team Level“ beschreiben, ist hingegen deutlich anspruchsvoller. Zwar lässt sich die Qualität der zwischenmenschlichen Beziehungen von Entwicklern und Testern nicht objektiv messen, aber mit Hilfe von historischen Prozessdaten lassen sich wichtige Trends erkennen. So dringt man in Bereiche vor, die sich einer objektiven Messung und Kontrolle klassischerweise entziehen. Dies macht einen verantwortungsvollen Umgang mit diesen Informationen erforderlich.

Verbindung von BPM und DevOps

Nach Identifizierung der Prozesse stellt sich aus Architektursicht die Frage, wie die Process Engine in die bestehende Toollandschaft integriert werden soll. Diese Architekturentscheidungen, die es individuell abzuwägen gilt, sind Grundvoraussetzung für den Einsatz von BPM und beeinflussen entscheidend die Ausrichtung der Prozessworkflows.

Grundlage für die Entscheidungen können folgende Fragen bilden:

- ▼ Wie stark soll der Prozessgedanke die tägliche Arbeit der Entwickler und Systemadministratoren durchdringen?

- ▼ Welche funktionalen Teilaspekte sollten aus den bestehenden Tools herausgelöst und in die Workflow Engine integriert werden?
- ▼ Welche Aufgaben sollte man besser nicht der Engine überlassen?
- ▼ Was ist das Leading Tool und wer liefert das zentrale Dashboard?

Zusätzlich muss ein DevOps-Engineer dazu stets folgende Aspekte berücksichtigen:

- ▼ Auf abstrakter Ebene bildet jedes DevOps-Tool einen eigenen Geschäftsprozess ab. Zum Beispiel beginnt der automatische Snapshot-Build, auf dem CI-Server, mit einem definierten Start Event (Code-Check-in) und endet mit einem Ergebnis (gebaute Artefakte im Artefact-Repository). Es ist nicht Aufgabe einer Process Engine, die bereits existierenden Prozesse abzulösen. Stattdessen obliegt ihr die Aufwertung, Kapselung und Integration der Prozesse in die bestehende DevOps-Toollandschaft.
- ▼ Zu den Hauptaufgaben einer Process Engine gehören in diesem Kontext Kollaboration, Koordination und Parametrisierung der Prozesse. Hierzu zählt auch die Benachrichtigung menschlicher Akteure bei Entscheidungsbedarf.
- ▼ Da die Process Engine eine sehr zentrale und wichtige Rolle im Build- und Deployment-Verfahren einnimmt, sind eine gute Performance und Ausfallsicherheit unverzichtbar.
- ▼ Weil die Process Engine im DevOps-Umfeld einen stark integrativen Charakter besitzt, ist der Einsatz einer möglichst offenen BPM-Suite empfehlenswert. Zumindest im Java-basierten Umfeld haben die leichtgewichtigen Open-Source-Lösungen gegenüber den proprietären BPM-Software-Suiten einen entscheidenden Vorteil. Sie sind als eine lose Sammlung von Java-Tools konzipiert und bieten die notwendige Freiheit zur Anbindung und Integration von Fremdsystemen.

Eine Integration der Process Engine kann wie folgt aussehen: Die grafische Benutzungsschnittstelle der BPM-Lösung übernimmt die Rolle eines Dashboards, das Einblick in die laufenden Prozessinstanzen gewährt und als Anlaufpunkt für die menschlichen Akteure fungiert. Die Process Engine kommuniziert in diesem Beispiel über eine REST-Schnittstelle mit den restlichen Tools (s. Abb. 4).

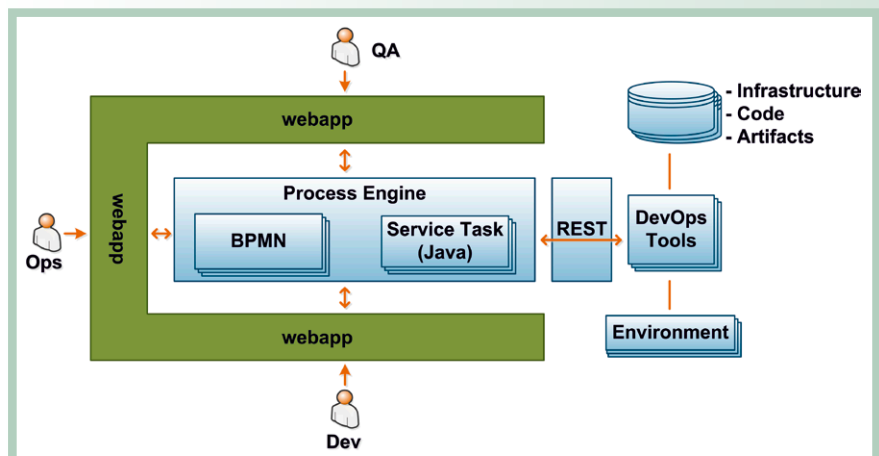


Abb. 4: Process Engine in der DevOps-Toollandschaft

Praxisbeispiel: Releasewechselprozess in der DVAG

Im Sinne der Geschäftsprozesssicht setzt sich der Softwareentwicklungs- und Bereitstellungsprozess aus den fol-

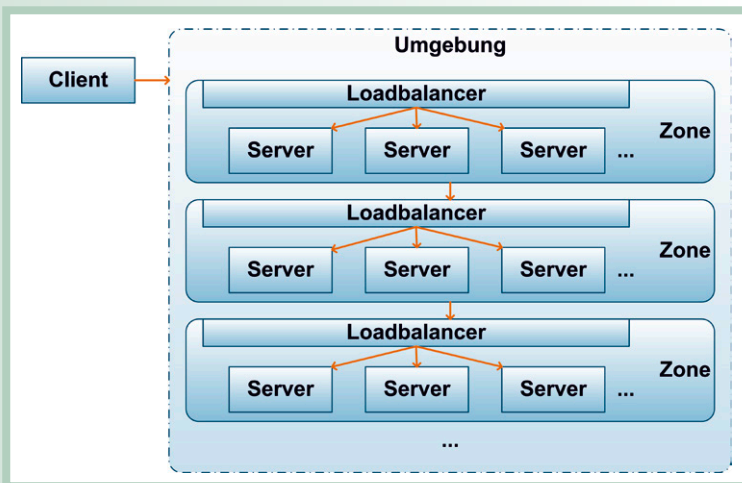


Abb. 5: Architektur des DVAG Online-Systems

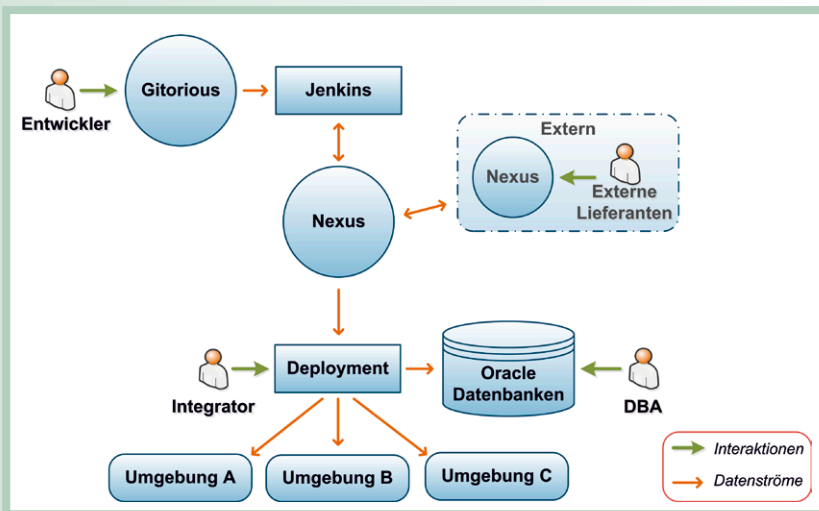


Abb. 6: Integrationsprozess der DVAG

genden Kernbereichen zusammen: Releaseplanung, Konfigurationsmanagement, Deployment-Pipeline (inkl. Releasewechsel), Troubleshooting und Qualitätssicherung. Vor allem bei großen Softwaresystemen mit mehrschichtigen Serverlandschaften stellt die Deployment-Pipeline die höchsten Anforderungen an die abteilungsübergreifende Koordination und Überwachung der beteiligten menschlichen Akteure und Maschinen.

Ein Softwareprodukt dieser Komplexität und Reichweite ist das „Online-System“ der Deutsche Vermögensberatung AG (DVAG). Die DVAG ist der größte eigenständige deutsche Finanzdienstleister mit über 6 Mio. Kunden, die von 3400 Direktionen und Geschäftsstellen betreut werden. Das DVAG Online-System unterstützt die Außendienstmitarbeiter in Beratung, Kommunikation, Verwaltung von Kundendaten und Vertriebssteuerung. Das 2006 eingeführte und ständig erweiterte System beruht auf der Smart-Client-Technologie und besteht aus einem sich selbst aktualisierenden Client sowie einer hochverfügbaren Serverumgebung (s. Abb. 5). Die zu deployenden Softwareänderungen durchlaufen im wöchentlichen Rhythmus mehrere Stages und Quality Gates, bis die hohen Ansprüche an ihre Qualität erreicht sind. In einem monatlichen Releasezyklus werden sie den Mitarbeitern des Außendienstes bereitgestellt. Neben dem Produktivsystem gibt es ein

gesondertes System für Schulungszwecke, welches einen eigenen Zweig inklusive Stages und Quality Gates bildet.

Verantwortlich für das Buildmanagement, Deployment und Releasemanagement bei der DVAG ist ein Team aus Integratoren. Das Buildmanagement umfasst alle Softwarekomponenten und Ressourcen, wie Java-Klassen, Migrationsskripte für die Datenbank sowie Komponenten von Drittanbietern. Hierzu kommt eine Maven-basierte Build-Infrastruktur zum Einsatz. Das Ergebnis eines Builds besteht aus einer Menge von Serververteilungen, die Out-of-the-Box laufen können (vgl. [Häf06]). Diese entstehen für alle Umgebungen und werden aus einem generischen und einem umgebungsabhängigen Anteil zusammengesetzt. Das anschließende Deployment verteilt die gebauten Artefakte und führt letzte Konfigurationsänderungen durch. Parallel dazu führen die Datenbankadministratoren

die vom Deployment bereitgestellten Datenbankskripte aus. Nach erfolgreichem Deployment finden automatisierte Tests statt. Komplexe Fachlichkeit wird direkt von der Qualitätssicherung getestet. Wöchentlich gebaute Versionen durchlaufen mehrere Stages und bilden dann die Basis für das monatliche produktive Release (s. Abb. 6). Parallel zu diesem Verfahren sorgt Continuous Delivery in eine gesonderte Umgebung für schnelles Feedback.

Die Sicherstellung der Qualität und Funktionsfähigkeit von Releases ist für die DVAG geschäftskritisch. Hinzu kommt die Anforderung nach einer möglichst kurzen Downtime. Daher benötigt der Deployment-Prozess für die Produktivumgebung einen ausführlichen Plan, der strikt eingehalten werden muss. Dieser legt fest, wann und von wem welche Tätigkeiten auszuführen sind – ob automatisch oder manuell. Zusätzlich müssen die Integratoren nach einem Releasewechsel in der Lage sein, Hotfixes ohne Downtime einzuspielen. Die verschiedenen Deployment-Szenarien stellen hohe Anforderungen an den Integrator und seine Werkzeuge. Das Vorgehen beim Deployment eines Hotfixes hängt stark von dessen Inhalt ab: Verteilt sich ein Hotfix auf verschiedene Server, ist die Reihenfolge beim Deployment zu berücksichtigen. Die zunehmende Komplexität, Fehleranfälligkeit und die Frage nach weiterer Optimierung inspirierte das Team der Integratoren zum Einsatz einer auf BPMN basierenden Process Engine.

Die verschiedenen Deployment-Szenarien stellen hohe Anforderungen an den Integrator und seine Werkzeuge. Das Vorgehen beim Deployment eines Hotfixes hängt stark von dessen Inhalt ab: Verteilt sich ein Hotfix auf verschiedene Server, ist die Reihenfolge beim Deployment zu berücksichtigen. Die zunehmende Komplexität, Fehleranfälligkeit und die Frage nach weiterer Optimierung inspirierte das Team der Integratoren zum Einsatz einer auf BPMN basierenden Process Engine.

Auswahl einer geeigneten BPM-Lösung

Bei der Suche nach einer geeigneten BPM-Software fiel die Wahl auf camunda BPM, einer Open-Source-Lösung in Java. Insbesondere die zugehörige Process Engine besitzt einen hohen Reifegrad und die aktive und hilfsbereite Community bietet ausreichend Investitionsschutz. Eingebettet in einen Web-Container (Tomcat), steuern und administrieren die Benutzer die Camunda Process Engine über die mitgelieferten Webapplikationen. Eine REST-Schnittstelle erlaubt die Integration von Fremdsystemen über ein standardisiertes Kommunikationsprotokoll.

Die Erweiterbarkeit der BPMN ermöglicht es den Integratoren, selbst entwickelte „Service Tasks“ (ausführbarer Java-Code) in die Prozessdefinitionen aufzunehmen. Diese bilden unternehmensspezifische fachliche Aspekte ab. Zur Erstellung der Prozessdiagramme wurde die mitgelieferte Werkzeugleis-

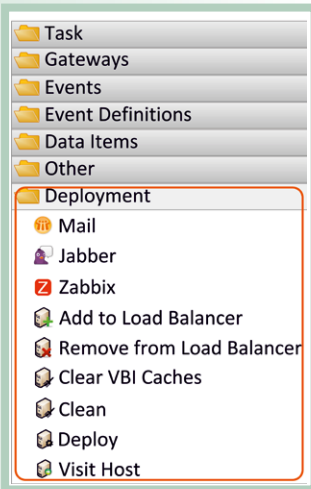


Abb. 7: Werkzeugleiste mit domänen-spezifischen Task-Elementen (in Rot)

te des Camunda Modelers, einer Eclipse-Erweiterung zur Gestaltung von BPMN-Diagrammen, um eigene benutzerdefinierte Elemente erweitert (s. Abb. 7).

Umsetzung des Deployment-Prozesses

Im Zuge einer Evaluationsphase entwarf das Team der Integratoren ein Prozessmodell für das Deployment des weniger komplexen Schulungsziels des DVAG Online-Systems. Es stellte sich heraus, dass eine Balance zwischen Detailgenauigkeit und Lesbarkeit der Prozess-

maschinelle Koordination ist eine große Hilfe für den Release-manager, vor allem, wenn die Choreografie von Abteilungen und Systemen eine Komplexität erreicht, die ein einzelner Mensch nicht mehr überblicken kann. Mit der Einführung von BPM im IT-Betrieb konnten die Integratoren der DVAG ein wichtiges Fundament für mehr Verständnis, Zusammenarbeit und Effizienz zwischen den Abteilungen legen. Ganz im Sinne eines DevOps 2.0.

Literatur und Links

[AllHam09] J. Allspaw, P. Hammond, 10+ Deploys Per Day: Dev and Ops Cooperation at Flickr, Velocity, 2009
 [Edw10] D. Edwards, <http://dev2ops.org/2010/11/devops-is-not-a-technology-problem-devops-is-a-business-problem/>
 [FreRü12] J. Freund, B. Rücker, Praxishandbuch BPMN 2.0, Carl Hanser Verlag, 2012

[Häf06] F. Häfner, Herausforderungen an Build Management und Deployment in großen Projekten, JAX 2006, http://www.compeople.de/files/haefner_build-unddeployment.pdf
 [Hüt12] M. Hüttermann, DevOps for Developers, Apress, 2012
 [Hum10] J. Humble, D. Farley, Continuous Delivery, Addison-Wesley, 2010
 [Pes12] P. Peschlow, Die DevOps-Bewegung, in: Java Magazin 1.12, 2012
 [ShDeGa] A. Shafer, P. Debois, I. Gat, DevOps Metrics, <http://de.slideshare.net/jedi4ever/devops-metrics>
 [Will10] J. Willis, 2010, <http://www.getchef.com/blog/2010/07/16/what-devops-means-to-me/>

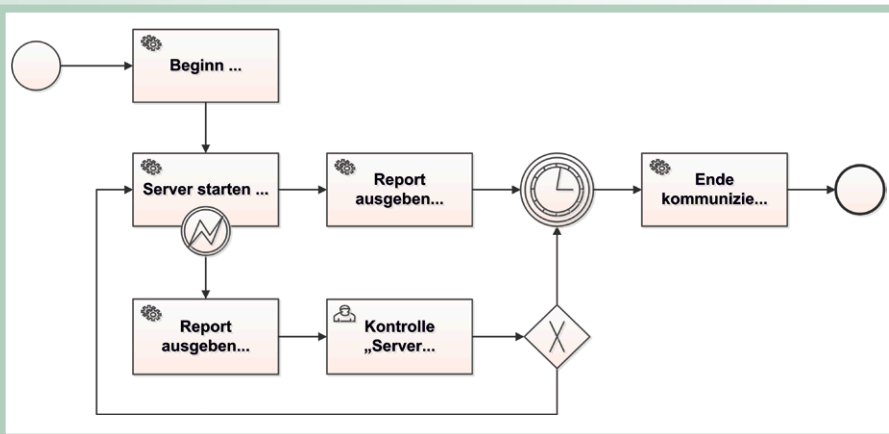


Abb. 8: Auszug aus einem ausführbaren Releaseworkflow – einer von insgesamt 51 Subprozessen

diagramme gefunden werden musste. Nach einer Entwicklungszeit von sechs Monaten, in denen das Team unter anderem die Anbindung der Process Engine in die bestehende technische Infrastruktur realisierte, entstand auf Grundlage zahlreicher Interviews und Meetings eine Sammlung hochwertiger und stabiler Prozessdefinitionen (s. Abb. 8). Diese können jetzt mit gleichbleibend hoher Qualität von jedem Teammitglied ausgeführt werden. Darüber hinaus existiert nun eine beispiellose Bündelung von Expertenwissen in der BPMN.

Eine Übersicht über den aktuellen Stand der laufenden Deployments ist für den Releasemanager nur noch einen Mausklick entfernt. Auch Releasewechsel-Prozesse, die sich über mehrere Tage erstrecken, erfordern keine Kontrolle und Aufmerksamkeit, weil die betroffenen Akteure automatisch und planmäßig via E-Mail und Chat benachrichtigt werden. Der anstrengende Koordinierungsaufwand, der in der Vergangenheit hauptsächlich durch Zuruf oder via Telefon erfolgte, entfällt nun.

Fazit

Aufgrund der gewonnenen Erfahrungen lässt sich gut belegen, dass die Einführung von BPM in den Betrieb zu mehr DevOps führt. Je besser die Schnittstellen zwischen den Abteilungen durch die ausführbaren Prozessmodelle abgebildet werden, desto besser funktioniert die Zusammenarbeit. Die



Masroor Ahmad ist freiberuflicher Trainer und Softwareberater im Java-Umfeld und unterstützt Unternehmen bei der Planung und Entwicklung komplexer Geschäftsanwendungen. Seine Schwerpunkte liegen im Aufbau von Build- und Deployment-Infrastrukturen sowie dem Qualitätsmanagement. Aktuell unterstützt er als externer Mitarbeiter das Konfigurations- und Releasemanagement des DVAG Online-Systems.
 E-Mail: kontakt@masroor.de

Liesa-Rebecca Thiel ist Angestellte bei der ATLAS Dienstleistung für Vermögensberatung GmbH, ein Dienstleister der Deutschen Vermögensberatung AG. Sie leitet dort das Team der Integratoren, welches für das Releasemanagement des DVAG Online-Systems, inklusive des Build- und Deployment-Managements, zuständig ist.
 E-Mail: liesa-rebecca.thiel@dvag.com