

REALTECH

EMPOWER YOUR ENTERPRISE

WHITEPAPER

DEVOPS FÜR SAP
TEIL 1

AUTOREN



Tom Schneider ist Technischer Architekt bei der REALTECH Deutschland GmbH. Er unterstützt Unternehmen bei der Einführung von SAP Automatisierungslösungen.

REALTECH
EMPOWER YOUR ENTERPRISE



Sören Keuntje ist Business Consultant bei der REALTECH Deutschland GmbH. Er unterstützt Unternehmen bei der Implementierung von SAP Automatisierungslösungen.

REALTECH
EMPOWER YOUR ENTERPRISE



Helmut Stork ist Partner der NextGen Leadership Group. Er begleitet Unternehmen bei der Entwicklung agiler Führungsstrategien zur Erzeugung kollaborativer, selbststeuernder Teamstrukturen.

**NextGen
Leadership
Group**

INHALT

1. MOTIVATION	SEITE 5
2. PHILOSOPHIE	SEITE 7
3. ÜBERTRAGEN DER PHILOSOPHIE IN DIE SAP-WELT	SEITE 11
4. WIE SIEHT EIN ERFOLGREICHER START AUS?	SEITE 19

ABBILDUNGSVERZEICHNIS

ABBILDUNG 1: SYNCHRONE AUSRICHTUNG UND INTRINSISCHE MOTIVATION ALS TREIBER DER VERÄNDERUNG	SEITE 8
ABBILDUNG 2: IT-LEBENSZYKLUS DES DEVOPS FRAMEWORKS	SEITE 13
ABBILDUNG 3: DEVOPS FOR SAP: PRINCIPLES & TOOLS	SEITE 20
ABBILDUNG 4: ERFOLGSFAKTOREN BEI DER DEVOPS-EINFÜHRUNG FÜR SAP	SEITE 21



MOTIVATION

DIE DIGITALE REVOLUTION VERÄNDERT UNSER WIRTSCHAFTSLEBEN!

Die Digitalisierung markiert den größten Wandel unseres Wirtschaftens seit der Industriellen Revolution. Die immer weiter fortschreitende Technisierung unseres Alltags hat auch die Bedürfnisse und das Verhalten von Menschen verändert.

Spätestens der Erfolg sozialer Medien beweist den nachhaltigen Einfluss digitaler Produkte auf unser Leben. Die Monetarisierung solcher digitaler Dienstleistungen wird komplexer und ist zunehmend datengetrieben. Vorreiter wie Uber, Airbnb oder Zalando sind mit digitalen Geschäftsmodellen bereits seit Jahren erfolgreich. Für etablierte Unternehmen bietet sich hier eine Chance zur Erneuerung: einerseits durch die konsequente Automation der bestehenden Prozesse mittels neuer Technologien, andererseits durch die Integration von elektronischen Geräten wie Sensoren und Aktoren mit Unternehmenssoftware (vgl. Internet of Things). Das vorrangige Ziel ist das Finden neuer, IT-basierter Geschäftsprozesse. Darüber hinaus existieren Chancen aber auch Notwendigkeiten zur Etablierung neuer digitaler Geschäftsmodelle und Produkt- bzw. Servicestrategien. Unternehmen stehen somit global vor der Aufgabe, ihr bisheriges Wirtschaften zu prüfen und an diesen Wandel anzupassen.

Es muss klar sein, dass im Rahmen der Digitalisierung heute die Grundsteine für den globalen Wettbewerb von morgen gelegt werden. Der Auftrag für etablierte Unternehmen lau-

tet, intelligentere Softwarelösungen schneller in Nutzung zu bringen und durch kreative Ideen völlig neue Geschäftskonzepte und Märkte zu schaffen.

IHR DIGITALER KERN MUSS AUF SCHNELLERE ENTWICKLUNGEN ANGEPASST WERDEN

Der digitale Kern einer Großzahl europäischer Unternehmen sind ihre SAP ERP-Systeme, welche üblicherweise alle wesentlichen Unternehmensfunktionen abbilden. 1979 markierte die erste R2-Installation eine Zäsur in der Geschichte von Unternehmenssoftware. 39 Jahre später lässt sich feststellen, dass SAP zu seiner eigenen Sparte innerhalb der IT-Welt geworden ist: mit eigener Architektur, eigener Terminologie, eigenen Nutzern, eigenen Abteilungen, eigener Kultur und eigener Denkart. Der Aufbau von SAP-Lösungen sowie SAP-Teams folgt mittlerweile einem eigenen Standard. Durch diese Entkoppelung der SAP-Welt von der restlichen IT-Welt sind viele wichtige Entwicklungen der Branche an dieser eigenen SAP-Welt vorbeigegangen. Objektorientierung, serviceorientierte Architektur, Cloud Hosting und Agile Development sind die prominentesten Beispiele für Paradigmenwechsel, die nachträglich bei SAP adaptiert wurden. Unglücklicherweise entstanden so häufig unsichtbare Mauern innerhalb von IT-Abteilungen. Innovation Hubs brechen in vielen Unternehmen aus der etablierten Plan-Build-Run-Struktur aus und bedienen sich bereits heute der Vorteile agilen Arbeitens – was vielfach zu einer bimodalen IT geführt hat.

Das innovative Framework DevOps vereint eine schnelle Auslieferung von Software und einen sicheren Systembetrieb.

Agile Softwareentwicklung etablierte – beinahe wie in einer Spiegelwelt zu SAP – eine kontinuierliche Lieferfrequenz kleiner Funktionsupdates als De-facto-Standard (im Kontrast zu sich stetig verschiebenden „Mammut-Releases“). Die Folge: SAP-Systeme sind weiterhin einzig auf das Stützen der produktiven Geschäftsprozesse ausgerichtet, nicht auf einen schnellen und schlanken Entwicklungsprozess, der Unternehmen mit neuen digitalen Ideen und Geschäftsmodellen versorgt. Der Widerspruch zwischen starren SAP-Architekturen und -Innovationen muss nun aufgebrochen werden.

SAP sollte von Unternehmen als ihr digitaler Kern erkannt und damit als wichtiger Erfolgsfaktor begriffen werden. Mit den passenden technologischen und kulturellen Ansätzen können SAP-Architekturen so zum Antreiber des digitalen Wandels gemacht werden. DevOps wird mittel- bis langfristig auch in SAP technische Schulden abbauen, um im SAP-Umfeld ebenfalls die Vorteile agiler Methoden zu nutzen.

KULTUR DER ZUSAMMENARBEIT WIRD ZUM SCHLÜSSELFAKTOR

Herkömmliche Führungs- und Organisationsprinzipien sind unzureichend. Ihre Aktionsparameter zielen auf Stabilität und Effizienz: Fragmentierung, Hierarchiebindung, Fehlervermeidung, starre Prozesse, Planungssicherheit. Die dadurch geschaffenen Strukturen sind „Slow by Design“.

Führungs- und Organisationsprinzipien müssen Anpassung und Innovation ermöglichen. Technologische und organisatorische Flexibilität ist Bedingung, um Volatilität und Wandel der Märkte zu kontern. Dies erfordert veränderte Führungs- und Organisationsformate, die Agilität bewirken, d.h. Anpassungsfähigkeit und Emergenz ermöglichen, d.h. das Erzielen von Innovation durch ganzheitliches Zusammenwirken. Ihre Aktionsparameter zielen auf: Reaktionsfähigkeit, autonomes Agieren, hohe Beziehungs- und Kommunikationsdichte zwischen den Akteuren, funktionale und hierarchische Integration, Vielfalt, Perspektivenwechsel. Die durch sie geschaffenen Strukturen lassen radikale Beschleunigung und ein hohes Maß an Kreativität entstehen.

2

PHILOSOPHIE

DEVOPS SETZT AGILE FÜHRUNGS- UND STEUERUNGSSTRATEGIEN VORAUS

Eine erfolgreiche und nachhaltige DevOps-Philosophie ist eng verbunden mit der Etablierung neuer Paradigmen in Führung und Teamsteuerung. Agile, verhaltensorientierte Führungsstrategien stehen im Mittelpunkt. Diese erzeugen Veränderungenergie und befähigen nachdrücklich zur Transformation. Sie bewirken, dass Teams ihre Ressourcen im gemeinsamen Denken aktivieren und nutzen – und innovativ werden.

Sowohl in der Softwareentwicklung als auch im Innovationsmanagement sind es agile Führungs- und Steuerungsmechanismen, die seit geraumer Zeit das Erzielen überragender Ergebnisse ermöglichen. So können mit verhaltensorientierten Methoden zum Beispiel im Innovationsmanagement Produkte schnell – typischerweise in einem Drittel der Zeit – konzipiert und zur Marktreife gebracht werden.

SCHWARMINTELLIGENZ LÄSST SELBSTSTEUERENDE TEAMSTRUKTUREN ENTSTEHEN

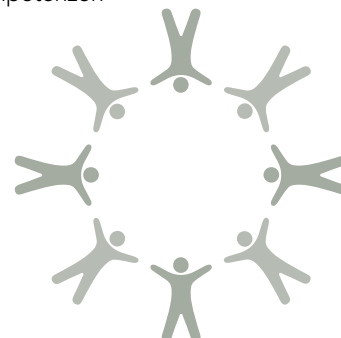
Agile DevOps-Organisationen verfügen über schwarmintelligente Eigenschaften. Genau diese sind es, die ihnen die einzigartige kreative Wirkungsweise sowie die Fähigkeit zu radikaler Beschleunigung verleiht.

Schwarmintelligente Systeme lassen sich wie folgt kennzeichnen: Sie sind selbstorganisierend, bewegen sich an der Grenze zum Chaos, haben als Ganzes Eigenschaften, die ein einzelner Teil nicht besitzt, und verfügen über keine inneren Grenzen. Sie sind adaptiv im Austausch mit ihrer Umgebung, erforschen ihre mögliche Zukunft und besitzen Eigen-

kontrolle. Schwarmintelligente Systeme passen sich ständig an die Grenzen und Eigenschaften ihrer Umwelt an, schließlich schaffen sie Ordnung.

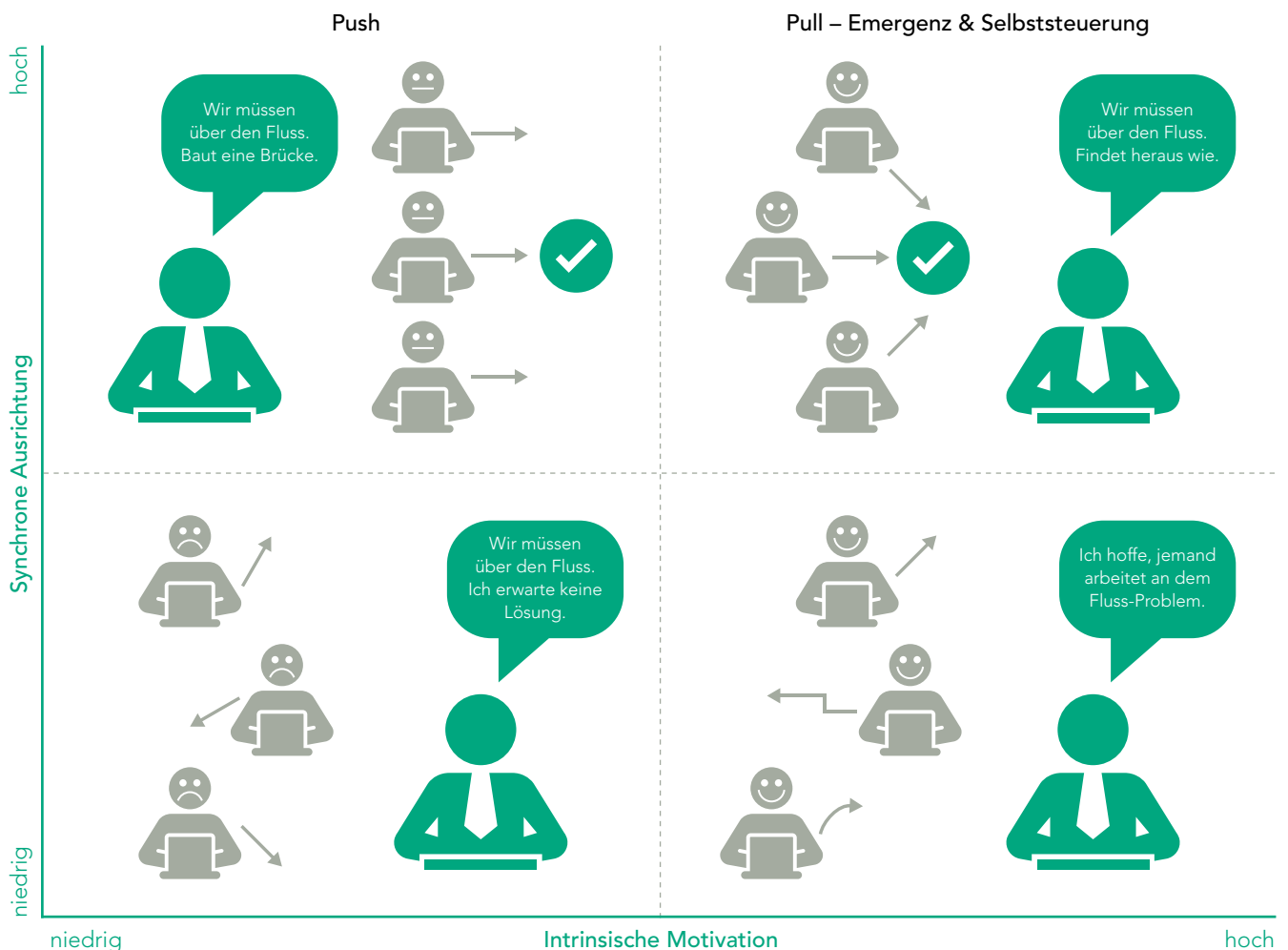
Führungskräfte agiler DevOps-Organisationen stimulieren diese Eigenschaften gezielt in ihren Teamstrukturen und lassen so Emergenz und Selbststeuerung entstehen. Die zehn nachstehenden Merkmale beschreiben das Verhaltensmuster und die agilen Eigenschaften selbststeuernder Teams:

- Reagieren spontan auf Veränderungen
- Vision und Werte bestimmen ihr Handeln
- Betrachten das Ganze im Verbund seiner Einzelteile
- Sind empathisch im Austausch mit anderen
- Schätzen Vielfalt wert
- Folgen autonom ihren Überzeugungen
- Eröffnen sich durch Fragen neue Möglichkeiten
- Sprengen im Perspektivenwechsel Wände weg und geben alte Paradigmen auf
- Sehen Fehlschläge als Chancen
- Erzielen Durchbrüche durch Reflexion ihrer Stärken und Kompetenzen



FROM PUSH TO PULL

Während herkömmliche Führungsmethoden Druck erzeugen, erzeugen agile Führungsmethoden Zug. Mit Umschalten von Druck auf Zug – FromPush2Pull – befähigt die DevOps-Führungskraft zur Veränderung von innen heraus, d. h., kein operatives Interagieren von außen, vielmehr intrinsische Motivation und Eigensteuerung des synchron ausgerichteten Teams. Es entstehen Commitment und die Fähigkeit zur Veränderung.



Synchrone Ausrichtung und intrinsische Motivation schaffen Emergenz & Selbststeuerung © 2018 REALTECH Deutschland GmbH.

AGILE FÜHRUNG IM SOFTWARE DEVELOPMENT & DEPLOYMENT LIFECYCLE CASE

Ausgehend von einer bestimmten Situation, die der Veränderung bedarf, wählt die DevOps-Führungskraft spezifische Maßnahmen aus, die aufgrund ihrer agilen Hebelkraft eine gewünschte Transformation bzw. einen Zustandswechsel bei den beteiligten Akteuren bewirken. Dabei werden agile Eigenschaften in konkrete Anwendungsformen gebracht, um in kritischen Handlungsfeldern des Software Development & Deployment Lifecycle (SDLC) erfolgswirksam zu sein.

Situation

- Best Practice in Midlife-Innovation (stufenweise, kontinuierliche Verbesserung) und
 - lange SDLC-Zyklen
- erschweren die Bereitstellung neuer digitaler Geschäftsmodelle, disruptiver Produkt- & Service-konzepte sowie erweiterter Kunden- & Anwenderfunktionen in stetig kürzeren Zeitabständen.

Ziel

- Zeitnahe softwaretechnische Implementierung neuer Anforderungen, kurze SDLC-Zyklen
- New-Life-Innovation (radikal, überproportionale Wachstumsperspektiven)

Agile Stellhebel

- Bildung crossfunktionalen Teams, bestehend aus Business Unit, Entwicklung, Qualitätssicherung, Operations, IT-Sicherheit und Kunde
- Where to go: Gemeinsame Vision setzen, in messbare Ziele unterteilen und in kurzen Zyklen mit dem Kunden/Anwender hinsichtlich Fortschritt abgleichen
- Ganzheitliche Betrachtung der Wertströme und Wirkbeziehungen (Value Stream Mapping); Measuring: Visibility und Traceability
- Kontinuierliche Entwicklung in iterativen Schritten und Bereitstellung inkrementeller Releases
- Klare Rollenzuteilung, wechselseitige Kenntnis der unmittelbaren Interaktionspartner, Wahl direkter Kommunikationswege, Ausrichtung auf gemeinsames Projektziel (Kohäsion, Separation, Alignment)
- Rückmeldung aktiv einholen und aktiv geben (permanente Feedback-Loops)
- Upstream-gerichtete funktionale Integration; Fail Early, Fail Fast; Fehler in der frühen Phase sind zulässig; späte Fehlerquellen, die das Betriebsrisiko erhöhen, werden nach vorne gezogen (Front-loading)
- Entscheidungskompetenz liegt im Team, d. h., Entscheidungen entstehen im gemeinsamen, interaktiven Prozess der Akteure, die über einen hohen Freiheitsgrad verfügen

Zustandswechsel

- von divergierenden Zielsetzungen zu gemeinsamer Vision
- von technischer zu marktbewusster Betrachtung
- von einer Summe von Einzelleistungen zu einem emergenten Teamprozess
- von konventionellen zu unkonventionellen Denkansätzen
- von zentralisierter Entscheidung zu dezentralem Wirkungsverbund unter Kunden- und Marktzug
- vom Silodenken zum integrativen Agieren
- von Fragmentierung zur Synchronisation
- von frühzeitiger Planfixierung zu kurzfristiger Anpassungsbereitschaft
- von passiver zu aktiver Kommunikation
- von unterschiedlicher Verbindlichkeit seitens der Akteure zu einem All-round-Commitment
- vom ‚Empfänger-Mode‘ zum aktiven Engagement

PARADIGMENWECHSEL IM SELBSTVERSTÄNDNIS DER DIGITAL LEADERS

Mit der veränderten Führungsrolle FromPush2Pull beginnt für die DevOps-Führungskraft ein neues Selbstverständnis. Sie wird zum Befähiger von offener Kollaboration, Emergenz und Innovation. Dies stellt veränderte Führungskompetenzen in Erwartung: vom Führen im herkömmlichen Sinn zum Rahmen schaffen, vom Entscheiden zum Wahrnehmen, vom Linearen zum Vernetzten, vom Eindeutigen zum Ambivalenten, von festgelegter Erwartungshaltung zu offenem, agilem Mindset. Um Menschen in die Next Economy mitzunehmen, muss die DevOps-Führungskraft agile Eigenschaften in der eigenen Person verwirklichen. Persönliche Entwicklung und die Entwicklung neuer Organisationsformate gehen Hand in Hand.

Verhaltensorientierte Maßnahmen führen zu radikaler Beschleunigung und hoher Innovationskraft.

ÜBERTRAGEN DER PHILOSOPHIE IN DIE SAP-WELT



DEVOPS FÜR SAP: BEWÄHRTE VORGEHENSWEISEN UND METHODEN INTELLIGENT VERÄNDERN UND AN HEUTIGE ERFORDERNISSE ANPASSEN.

Höhere Wertschöpfung durch IT als Ziel

DevOps ermöglichen eine höhere Wertschöpfung der Unternehmens-IT durch einen schnelleren Entwicklungsprozess. Qualitativ höherwertige Eigenentwicklungen und automatisierte Releaseprozesse führen auf diese Weise auch in SAP on premise zu einer höheren Betriebssicherheit der bestehenden Systemlandschaft.

Folgende messbare Verbesserungen werden mit DevOps für SAP verfolgt:

- Höhere Betriebssicherheit des digitalen Kerns SAP
- Bessere Qualität der Software-Inkremete und bessere Qualität im Entwicklungsprozess
- Höhere Performance in den Entwicklungsintervallen
- Bessere Zusammenarbeit

Die Automatisierung bestehender Unternehmensprozesse in SAP ist abhängig von der Organisationsstruktur des jeweiligen Unternehmens. Diese Struktur muss nun für die agilen Prozesse und schnelleren Prozesslaufzeiten von DevOps befähigt werden.

Herausforderung: Zusammenspiel zwischen Kultur, Menschen und Technologie

In klassisch aufgestellten SAP-Betriebsorganisationen sind neben technischen Voraussetzungen eine Reihe nicht-technischer Spezifika zu beachten. Um in der Vergangenheit das Risiko eines Produktionsstillstands durch Eigenentwicklungen zu minimieren, wurde Flexibilität im SAP-Entwicklungsprozess durch starre Releasezyklen mit langen Testzeiträumen gehemmt. Releasezyklen von mehreren Monaten sind so zur Normalität geworden. Die große Mehrzahl an SAP-Kunden betreibt derzeit mehrere, voneinander getrennt agierende Entwicklungs- und Betriebsteams. Entwickler schreiben ihren ABAP- oder Java-Quellcode auf dem dafür vorgesehenen Entwicklungssystem und gegenüberstehende SAP Basis-Teams kümmern sich um einen möglichst störungsfreien Systembetrieb der Produktivumgebung. Jede Eigenentwicklung, die in das produktive System transportiert wird, stellt für die SAP Basis in erster Linie ein Betriebsrisiko dar. Analog ist jede Anforderung aus der SAP Basis für Entwickler eine ungeplante Verzögerung. Leider ist dies Alltag und geht zu Lasten des Kunden. Neue Geschäftsanforderungen werden nicht in akzeptabler Zeit umgesetzt und stauen sich in der Development Pipeline. Die so entstandenen „Mammut-Releases“ erhöhen das Risiko von Systemausfällen. Geht die Produktivsetzung dieser großen Upgrades schief, zieht dies kritische Produktionsausfälle nach sich. Zudem fehlt den ausführenden Mitarbeitern bei wenigen Produktivsetzungen oft schlicht die Routine für diesen Prozess. Dadurch leidet die Arbeitsqualität bei allen Beteiligten, Entwicklern und Administratoren, sowie bei Kunden, die auf neue Funktionalitäten warten.

IT-Abteilungen stehen also vor der Herausforderung, mit begrenzten Ressourcen zunehmend komplexere SAP-Landschaften immer schneller an sich stetig verändernde Anforderungen anzupassen.

Dies ist ein Spagat, den nur ein grundlegender Methodenwechsel möglich macht:

Eigenverantwortung und Kollaboration

Zum Kern der DevOps-Philosophie gehört auf sozialer Ebene, talentierte und eigenverantwortliche IT-Mitarbeiter zu fördern und ihnen mehr Hoheit in ihren Fachbereichen zu gewähren. Als Basis dient eine gemeinsame Vision des ganzen Teams, ein übergeordnetes Ziel, das regelmäßig besprochen und stetig verfeinert wird. Sogenanntes ‚Swarming‘, also frühes Einbeziehen von möglichst vielen fähigen Mitarbeitern zur schnellen Lösung eines schwierigen technischen Problems, führt im Regelfall zu einer frühen Problemlösung bei noch niedriger Eskalationsstufe. Wird das Problem im konventionellen Betriebsmodus nicht gelöst, steigt der Eskalationsdruck, und letztendlich arbeitet auch hier das ganze Team an der Lösung. Die Lieferung verzögert sich und die Frustration bei Kunden und im Team ist hoch. Hier hilft eine Anpassung der eigenen Arbeitsweise an die sich schnell und stetig ändernden Anforderungen.

Als erstes gemeinsames Ziel kann ein Agiles SAP formuliert werden.

Sinnvolle Entscheidungsfindung

Genehmigungsprozesse dürfen nicht hemmen und Entscheidungen müssen fachlich getroffen werden. So sollten im Rahmen von DevOps für SAP nur fachlich wirklich notwendige Genehmigungsschritte in Prozessen beibehalten werden. Alle anderen Genehmigungen können mithilfe intelligenter, automatisierter Prozesse durch Tools abgebildet werden. Flache Hierarchien bedeuten in diesem Sinne auch

eine Rückkehr zu fachbasierten Entscheidungen. Denn mit der Wegnahme von rein verwaltenden Prozessschritten tritt die fachliche Umsetzung der Arbeit – etwa Coding, Testing und Administration – wieder in den Vordergrund. Zudem sind Entscheidungen im Sinne des gesamten Unternehmens zu treffen, nicht nur im Sinne einzelner Abteilungen. Dies kann durch ein punktuelles, themenbezogenes Zusammenziehen der notwendigen Experten aus den jeweiligen Fachgebieten realisiert werden.

Das zweite gemeinsame Ziel stellen automatisierte Genehmigungs- und Testverfahren dar.

Interdisziplinäre Teams

Wer Genehmigungsprozesse abbaut, muss an deren Stelle neue Mechanismen zum Abfangen von erwartbaren Problemen setzen. So sollten beispielsweise Administratoren, Qualitätssicherung, Information Security und anfordernder Fachbereich bereits im frühen Entwicklungsprozess mitarbeiten, sodass rückwärtsgewandte, aufwändigere Prüfungen im Nachhinein vermieden werden. Auf diese Weise können agile Ansätze, wie DevOps für SAP, Silos im Unternehmen reduzieren. Sie fördern eine neue, offene und zielorientierte Unternehmenskultur. Das Hauptaugenmerk bei der Einführung einer agilen SAP-Betriebsorganisation liegt somit auf den handelnden Personen, danach erst auf Tools und Plattformen. Eine DevOps-Einführung soll in diesem Rahmen SAP- und Non-SAP-Teile des Unternehmens wieder harmonisieren. Wenn im Folgenden von interdisziplinären Teams gesprochen wird, beinhalten diese je ein Mitglied aus Entwicklung (Dev), Basis-Administration (Basis), Quality Assurance (QA), Information Security (InfoSec) sowie anforderndem Fachbereich (Kunde).

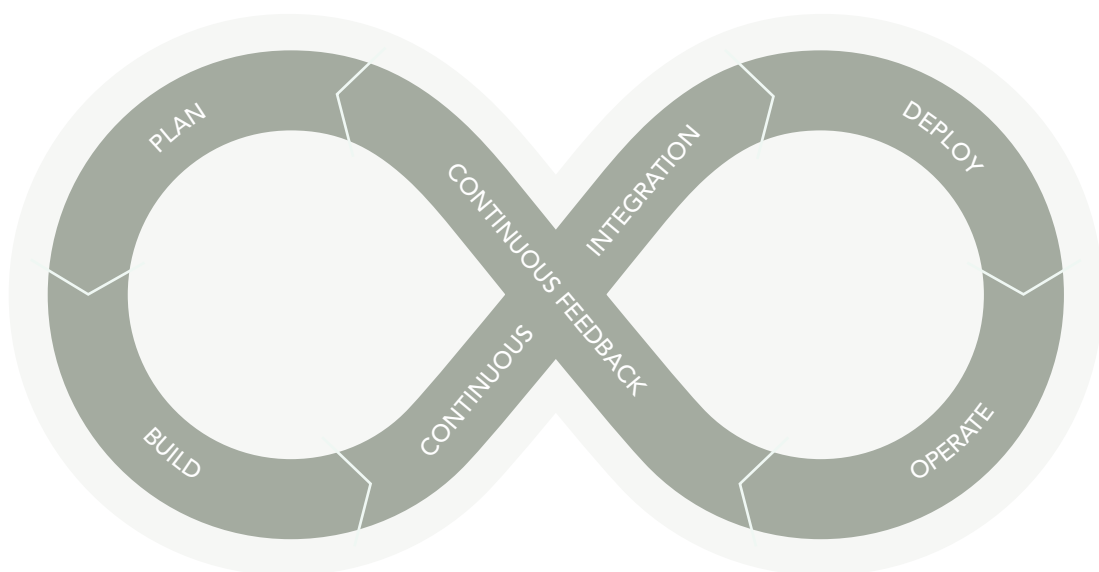
Das dritte gemeinsame Ziel ist die Etablierung von themen- und rollenübergreifenden Teams.

DEVOPS VEREINT EINE SCHNELLE AUSLIEFERUNG VON SOFTWARE UND EINEN SICHEREN SYSTEMBETRIEB

Der klassische DevOps-Lifecycle besteht aus einem Phasenmodell, welches kontinuierlich abläuft und eine Symbiose aus Entwicklungs- und Betriebsprozess bildet. Je nach Quelle bzw. Lösungsanbieter variiert das aufgezeigte Modell in der Anzahl der Phasen und deren Benennung, gibt aber stets die gleiche Kernphilosophie wieder.

Im Folgenden wird das DevOps-Framework der Firma Atlassian, einer der führenden Anbieter von DevOps-Tools in der Non-SAP-Welt, betrachtet:

DevOps für SAP heißt bewährte Vorgehensweisen und Methoden intelligent zu verändern und an heutige Erfordernisse anzupassen.



IT-Lebenszyklus des DevOps Frameworks.

- Die erste Phase, Plan, beinhaltet Entwicklungsaktivitäten von der Anforderungsanalyse über die Erstellung eines Release-Plans und des Business Case. Das Feedback aus dem Betrieb fließt kontinuierlich in die nächste Plan-Phase mit ein. Über alle Phasen hinweg wird in interdisziplinären Teams zusammengearbeitet, sodass Kunde, QA, InfoSec und Basis von Anfang an in die Entwicklungstätigkeiten miteinbezogen und zeitraubende Prozessübergänge zwischen einzelnen Silos vermieden werden.
- Auf die Phase Plan folgt Build, vereinzelt auch als Create bezeichnet. In dieser Phase wird ein Stück Software als Release Candidate erstellt. Dazu wird der Quellcode der Software geschrieben und paketierrt. Kollaboratives Programmieren kann durch eine Multicode- bzw. Versionsmanagementlösung gestützt und teilweise automatisiert werden.
- Die dritte Phase, Continuous Integration bzw. Verify, beinhaltet die Testaktivitäten der Software-Qualitätssicherung und anschließende Freigabe des Releases. Miteinbezogen wird die Konfiguration der Software auf der Infrastruktur. Zentraler Punkt ist die Automatisierung mittels Tools, sodass beispielsweise Informationen teamübergreifend geteilt werden und dadurch Übergänge der Verantwortlichkeit zwischen den Teams zügig und transparent stattfinden.
- Daran anknüpfend wird das Software-Release in der Phase Deploy automatisiert in die Produktionsumgebung ausgerollt, also den Anwendern zur Verfügung gestellt. Dies beinhaltet auch die Sicherstellung einer möglichen, zeitnahen Wiederherstellung der produktiven Systeme auf einen vorherigen Zeitpunkt (Rollback).
- Die ausgerollte Software wird in der Phase Operate unter Zuhilfenahme von Automatisierungstools des täglichen Betriebs, wie einfachen Reboots, Backup- und Recovery-Vor-

gängen, oder der massenhaften Anlage von Usern betrieben. Ebenso ist ein detailliertes Monitoring notwendig, das neben System- auch Prozessdaten sammelt, um eine Echtzeitevaluation des Betriebs zu gewährleisten.

- In der – vermeintlich – letzten Phase, Continuous Feedback bzw. Reporting, werden Metriken und Statistiken zur Verfügung gestellt, auf deren Grundlage Performance und End-user-Experience beurteilt werden. Das daraus resultierende Feedback wird wieder in die erste Phase, Plan, gegeben. Auf diese Weise stellen DevOps durch stetige Verbesserungen und schnelle Releasezyklen die Wettbewerbsfähigkeit des Unternehmens sicher.

DIE ÜBERTRAGUNG DES DEVOPS-FRAMEWORKS AUF SAP

SAP ERP On-premise-Architekturen liefern im Umfeld von DevOps grundlegende Funktionalitäten, die aber zur Umsetzung des Gesamtkonzepts nicht ausreichen. In diesem Kapitel werden diese grundlegenden Funktionalitäten dargestellt und es wird beschrieben, welche weiteren Funktionalitäten darüber hinaus notwendig sind.

Im Detail lässt sich die DevOps-Philosophie auf den bestehenden technischen Releaseprozess in SAP wie folgt übertragen: Die Phase Plan kann beispielweise im SAP Solution Manager durch das Anforderungsmanagement abgedeckt werden. Die Build-Phase spielt sich im ABAP-Editor des jeweiligen Entwicklungssystems ab. Die anschließende Phase, Continuous Integration, wird rudimentär durch das ABAP Test Cockpit, den Code Inspector und den Package Builder unterstützt. Das Deployment erfolgt in der Phase Deploy über den

Transport Organizer bzw. das SAP Transport Management System (STMS). In der Phase Operate kann beispielsweise über das Landscape Virtualization Management (LVM) eine produktive Umgebung automatisiert bereitgestellt und konfiguriert werden. Der Kreis schließt sich wieder in der Phase Continuous Feedback. Das notwendige Feedback aus dem Betrieb kann ebenfalls im SAP Solution Manager dokumentiert und Entwicklern bereitgestellt werden. Damit bietet die SAP schon grundlegende Funktionalitäten an, die jedoch im Rahmen der notwendigen Toolchain bisher nicht integrativ zusammenarbeiten und somit nicht in Gänze das DevOps-Framework unterstützen. Um dies zu erlangen, werden beispielsweise parallelisierbare Prozessfunktionalitäten, integrierte automatisierte Testverfahren und Möglichkeiten zum Aufbau von Testumgebungen inklusive Rollback und Schnittstellen zu Fremdsystemen benötigt.

Technische Architektur: Standardmäßig vorhandene Entwicklungs- und Test-Systeme

Standardmäßig sind meist dreistufige SAP-Umgebungen für Entwicklung (D), Test (Q) und Produktion (P) verfügbar. D- und Q-Systeme werden in SAP-Landschaften, im Gegensatz zu anderen Softwarearchitekturen, nicht auf Ad-hoc-Anforderung hin provisioniert, sondern in der initialen Installation mitausgeliefert. Das entschärft die Komplexität der Infrastruktur im DevOps-für-SAP-Ansatz, da virtuelle Server nicht als Infrastructure-as-a-Code provisioniert und SAP als Applikation nicht extra installiert werden müssen, sondern bereits verfügbar sind. Deployment, Integration und Konfiguration einer Softwarekomponente werden über das STMS abgebildet. Auf Basis dieser Verteilungs-

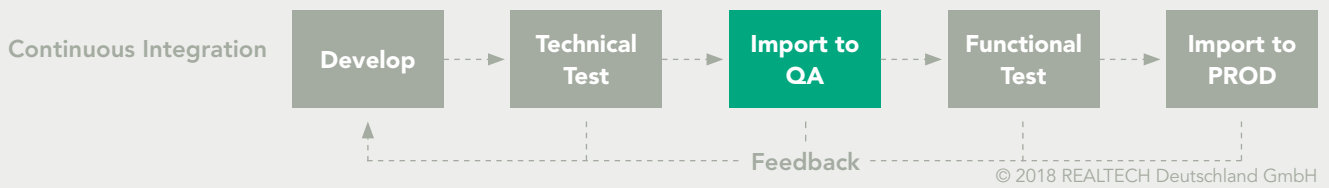
logik zwischen D-, Q- und P-Systemen, ergänzt durch zusätzliche intelligente Automatisierung, lassen sich so agile Entwicklungs- und Betriebsabläufe bereits heute in SAP abbilden.

Transport Management: Automatisiertes Release and Deploy von SAP-Entwicklungen

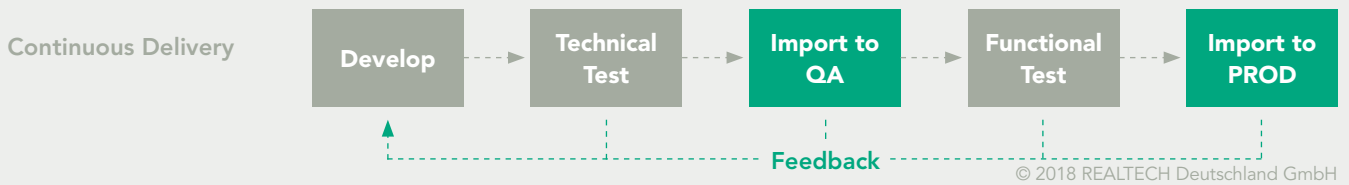
Eine fertige SAP-Entwicklung ist stets zu testen und anschließend freizugeben, bevor sie dem Produktivbetrieb zur Verfügung gestellt werden kann. Dem Import der Entwicklung in das Q-System folgt ein fachlicher Test und – im Erfolgsfall – die Freigabe zur Produktivsetzung. Dabei liegt der bisherige Prozessfokus auf einem Einhalten der komplexen Genehmigungslogik zum Wohle eines sichereren Betriebs. Jede Produktivsetzung erfordert derzeit eine mehrfache Freigabe.

Innerhalb eines DevOps-für-SAP-Frameworks sollen diese nun einfach, schnell, möglichst ohne Risiko und vermeidbare Genehmigungsschritte von den Entwicklungs- in die Zielsysteme transportiert werden. So ergeben sich drei Ausbaustufen von agilen SAP Development Pipelines. Im Folgenden wird – aus Gründen der Übersichtlichkeit – von einem Transportprozess innerhalb einer einfachen, dreigliedrigen DQP-Landschaft inklusive technischem Test vor Q-Import und funktionalem Test vor Produktivsetzung ausgegangen. Wünschenswert wäre es, im Falle von Changes mit unkritischen Objekten vollautomatisiert Releases einzuspielen, beispielsweise beim Implementieren von SAP Notes, die auf Objekte zugreifen, die keine unternehmenskritischen Geschäftsprozesse beeinflussen.

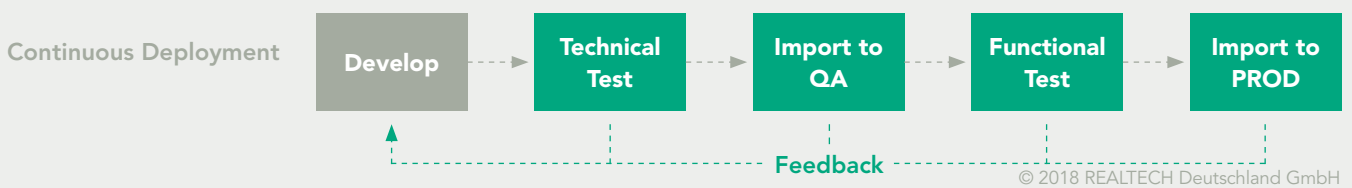
Die Ausprägungen der SAP Development Pipelines werden nachfolgend näher beschrieben:



Die Grundstufe, Continuous Integration, bezeichnet eine Transportautomatisierung zwischen D- und Q- System. Nach Fertigstellung der Entwicklung werden Transportaufträge automatisch in einen vordefinierten Genehmigungsworkflow aufgenommen und in das Q-System importiert. Technische Tests innerhalb der Entwicklung realisieren Qualitätschecks, eine Prüfung auf kritische Objekte, Kollisionen, Versionsüberholer, Objektabhängigkeiten sowie Transportvollständigkeit. Durch zusätzliche Systembenachrichtigungen ist ein direktes Feedback bzw. eine Vorhersage innerhalb des SAP-Systems an die Entwickler über den korrekten Import ihrer Entwicklung in das Q-System abbildbar.



Die erste Ausbaustufe, Continuous Delivery, beinhaltet einen automatisierten Import des Transportauftrags in das Q-System nach erfolgreich durchlaufenen technischen Tests. Der anschließende Import in die Produktionsumgebung erfolgt nach erfolgreicher Abnahme der funktionalen Tests. Sind alle Testfälle der Testfall-Bibliothek im Test-Management-System erfolgreich durchlaufen und liegt eine Abnahme durch den Testmanager vor, steht einer Produktivsetzung der getesteten SAP-Entwicklung nichts mehr im Weg. Eine weitere Integration in Service Portal und ITSM-Prozesse ermöglichen einen zentral gesteuerten, halb-automatisierten Test basierend auf ITIL. Auf diese Weise ist der gesamte Software-Lebenszyklus abgebildet – von der Anforderung über die Entwicklung und die Tests bis hin zur Produktivsetzung der SAP-Entwicklung.



Die vollständig automatisierte Lösung aller organisatorischer Genehmigungen inklusive vollautomatisierter Prozessübergänge wird als Continuous Deployment bezeichnet. Die finale Ausbaustufe der SAP Development Pipeline synchronisiert alle Datenquellen der SAP-Entwicklung und nutzt sie zu einer weitestgehenden Automatisierung des STMS. An dieser Stelle werden neben technischen Tests vor Q-Import ebenso fachliche Tests unter Zuhilfenahme des SAP Solution Manager (beispielsweise) und dessen Teststeuerungsfunktionen zur Automatisierung des Transportkonzepts durchgeführt. Dazu kommt eine umfassende Integration bestehender Technologien und Prozesse. Änderungen werden an allen relevanten Datenquellen der SAP-Entwicklung synchron gehalten. Redundanz im Ablauf wird reduziert und das tatsächliche Potenzial der SAP-Landschaft besser ausgeschöpft.

Wünschenswert wäre in diesem Zusammenhang ein Enterprise Layer, das die Integration der SAP-Systemlinie mit unterschiedlichsten Change-Management-Tools ermöglicht.

System Monitoring: Bindeglied zwischen Entwicklung und Betrieb

Der Gedanke eines kontinuierlichen Feedbacks sieht vor, dass Entwickler nach dem Transport ihrer Entwicklungen in das Produktivsystem in Echtzeit Feedback über deren Performance und Einfluss auf den Systembetrieb erhalten.

In SAP-Systemen liegen diese Informationen bereits vor und werden gesammelt, wurden aber in der Vergangenheit nicht automatisch für die Entwickler aufbereitet, sondern der Basis bereitgestellt. In Zukunft müssen performancerelevante KPIs im Anschluss an den Import ins Produktivsystem automatisch wieder zum Entwickler gelangen, denn letztendlich wird eben dieser Entwickler seinen Quellcode performancetechnisch optimieren – und dafür benötigt er jene Informationen darüber, wie sich beispielsweise die Antwortzeit einer Eigenentwicklung zusammensetzt, ob eine schlechte Antwortzeit daraus resultiert, dass die sequenziellen Lesezugriffe auf der Datenbank lange dauern, und seit welchem Release die Performance sich verbessert oder auch verschlechtert hat. Ziel ist, die Entwickler zu befähigen, selbständig die Qualität der Entwicklung durch dieses zeitnahe Systemfeedback zu optimieren und die Barriere zwischen Entwicklung und SAP Basis aufzulösen. Dadurch wird nicht nur die Qualität, sondern auch die Geschwindigkeit der Releasezyklen erhöht.

Alle bestehenden implementierten Lösungen von SAP basieren auf dem klassischen Ansatz der getrennten Teams und der daraus resultierenden Arbeitsweise in Silos. Keine Lösung ist durchgängig integriert, sodass dieses DevOps-Framework nicht über alle Phasen hinweg unterstützt wird. So bietet SAP mehrere Möglichkeiten, die beschriebenen Daten hinsichtlich Performance und Qualität der Entwicklungen zu analy-

sieren. Beispielsweise kann die Feedbackphase bereits über das Monitoring und Incident Management des SAP Solution Manager unterstützt werden. Darauf aufbauend ist eine automatisierte Performancemessung möglich und Tickets können an die zuständigen Mitarbeiter weitergeleitet werden. Doch auch dieses Konzept setzt wiederum auf dem klassischen Ansatz auf, dass Entwicklung und SAP Basis getrennt voneinander agieren und dass das SAP Basis-Team die Performance im laufenden Betrieb überwacht und die Entwicklung erst im Problemfall informiert. Nach wie vor ist es erforderlich, dass die Basis den Betrieb der produktiven Systeme sicherstellt. Dazu zählen das kontinuierliche Monitoring der Systeme über SAP-Programme wie die klassischen Transaktionen im System selbst, eine zentrale Monitoring-Lösung über den SAP Solution Manager, die aber auch oftmals durch Lösungen von Drittanbietern unterstützt wird.

Eine weitere zu betrachtende Möglichkeit im SAP Solution Manager ist das sog. Custom Code Management. SAP hat hiermit die Analysen über die Qualität der Eigenentwicklungen auf eine neue Stufe gehoben. Es können verschiedenste Analysen der Eigenentwicklungen vorgenommen und übersichtlich in einem frei konfigurierbaren Dashboard angezeigt werden. SAP verfolgt damit das Ziel der Annäherung der Kunden zurück an den SAP-Standard, sodass diese wiederum einfacher auf neue innovative SAP-Lösungen migrieren können. SAP-Kunden wird damit eine Möglichkeit zur Verbesserung der Qualität ihrer Eigenentwicklungen in Aussicht gestellt. Dies ist durchaus eine – wenn auch nicht direkt in den Releasezyklus eingebettete – Möglichkeit, dem Entwickler automatisch generiertes Feedback zur Verfügung zu stellen.

Es muss eine landschaftsspezifische Lösung geschaffen werden, die nach dem DevOps-Prinzip die Feedbackphase best-

Der Schlüssel zum Erfolg ist eine nahtlose Integration.

möglich mit Metriken über die Performance und Qualität in SAP-Landschaften unterstützt und die es dadurch den Kunden ermöglicht, agil zu entwickeln, schnell zu deployen und sofort ein Feedback über die Auswirkungen der Änderung zu erhalten. Das Ziel ist, bei immer kürzeren Releasezyklen die Stabilität des Systems nicht zu beeinträchtigen, darüber hinaus dem Entwickler durch ein nahtlos in den Releasezyklus integriertes Dashboard unmittelbar Feedback zu geben und folglich die Stabilität durch eine höhere Qualität der Eigenentwicklungen zu steigern. Genau auf diese nahtlose Integration über alle Phasen des DevOps-Framework hinweg warten SAP-Kunden bis heute. Stattdessen unterstützt SAP mit guten Lösungen aber nur punktuell im Entwicklungs- oder Betriebsprozess.

Soziale und technische Integration: Mehrwert durch Vernetzung

Im Rahmen eines DevOps-für-SAP-Frameworks soll eine intensive und offene Feedbackkultur im Unternehmen eingeführt und diese durch die intelligente Vernetzung der bestehenden Tools getragen werden.

Die Automatisierung des STMS im Allgemeinen sowie der Aufbau einer automatisierten Development Pipeline im Speziellen bedeuten implizit auch den Abbau von Genehmigungsworkflows. Ein Teil der bestehenden Kontrollmechanismen innerhalb des SAP-Betriebskonzepts fällt weg. Aufgefangen wird dies durch den Aufbau von interdisziplinären Teams und agilen Kommunikationsstrukturen. De facto erfordert dies beispielsweise einen Einbezug der Basis, QA und InfoSec ab der Grobkonzeption einer Idee für ein Software-Inkrement. Durch die Unterstützung mit einer passenden Toolchain soll der Aufbau eines kollektiven Bewusstseins unter allen SAP-Experten im Unternehmen gefördert werden.

Gemeinsame Kollaborations- und Planungssysteme, wie Jira, SharePoint, Microsoft Teams oder Project, stehen bei der Integration in den SAP-Betriebsablauf im Mittelpunkt. Über deren Anbindung an das hiesige, für SAP Change Requests verantwortliche Ticketsystem bzw. Service Portal lassen sich genehmigte Change Requests in den jeweiligen Tools automatisch in abhängige Aufgaben herunterbrechen und logisch verknüpfen. Sind alle Aufgaben im Planungssystem dann abgeschlossen – und damit auch alle Coding-Tätigkeiten – lässt sich neben dem Transport und damit der Freigabe für QA-Tests auch eine Rückspiegelung in das Service Portal realisieren. So wird sichergestellt, dass nach der Implementierung der benötigten Schnittstellen auch das jeweilige Ticket im Service Portal auf einen neuen Status gesetzt und in Abhängigkeit davon weiter prozessiert wird.

Eine mögliche Verknüpfung zwischen Transportauftrag in SAP und Change Request im Service Portal stellt sicher, dass kein undokumentierter, organisatorisch unerwünschter oder schlicht nicht zuordenbarer Auftrag verarbeitet wird. Durch diese Ende-zu-Ende-Inventarisierung und Synchronisierung aller Datenquellen des Change Managements, der Scrum-Sprintplanung, der Softwareentwicklung und schließlich des STMS wird der komplette Prozess einer Idee bis zur produktiven Inbetriebnahme abgebildet. Eine kundenspezifische Automatisierung der Prozessübergänge zwischen den einzelnen Systemen verschafft Unternehmen eine höhere Wertschöpfung durch zentrale IT-Systeme und befähigt zum weiteren Wandel von Prozessen und Strukturen.

WIE SIEHT EIN ERFOLGREICHER START AUS?

4

WELCHE SAP-KOMPONENTEN BEIM EINSTIEG UNBEDINGT BEDACHT WERDEN MÜSSEN

Im folgenden Abschnitt werden die wesentlichen vier Referenzkomponenten für eine DevOps-fähige und integrierte SAP-Landschaft (wie in Kapitel 3 beschrieben) dargestellt. Dazu bietet SAP bereits einige grundlegende Funktionalitäten. Diese machen die Basis für eine agile SAP-Systemlinie aus.

1. Transportautomatisierung ist die Automatisierung der SAP Development Pipeline mithilfe von STMS-Konfiguration, indem von Entwicklung bis Produktivsetzung nicht benötigte Genehmigungen gestrichen und automatisiert werden. Das Ziel ist die Reduktion von Wartezeit bei Prozess-übergängen zwischen verschiedenen beteiligten Teams. Der Weg dorthin ist die intelligente und landschaftsspezifische Automatisierung der Transport- und Genehmigungslogik in SAP anhand einer Integration in die bestehende Toolchain.

2. SAP-Telemetrie und -Metriken visualisieren Daten aus der SAP eigenen CCMS Monitoring Architektur und unterstützen so die agile Softwareentwicklung über alle Phasen hinweg. Als Mehrwert enthält der Entwickler – in nahezu Echtzeit – z. B. eine Auswertung über die entscheidenden Key Performance Indikatoren der transportierten Entwicklung im

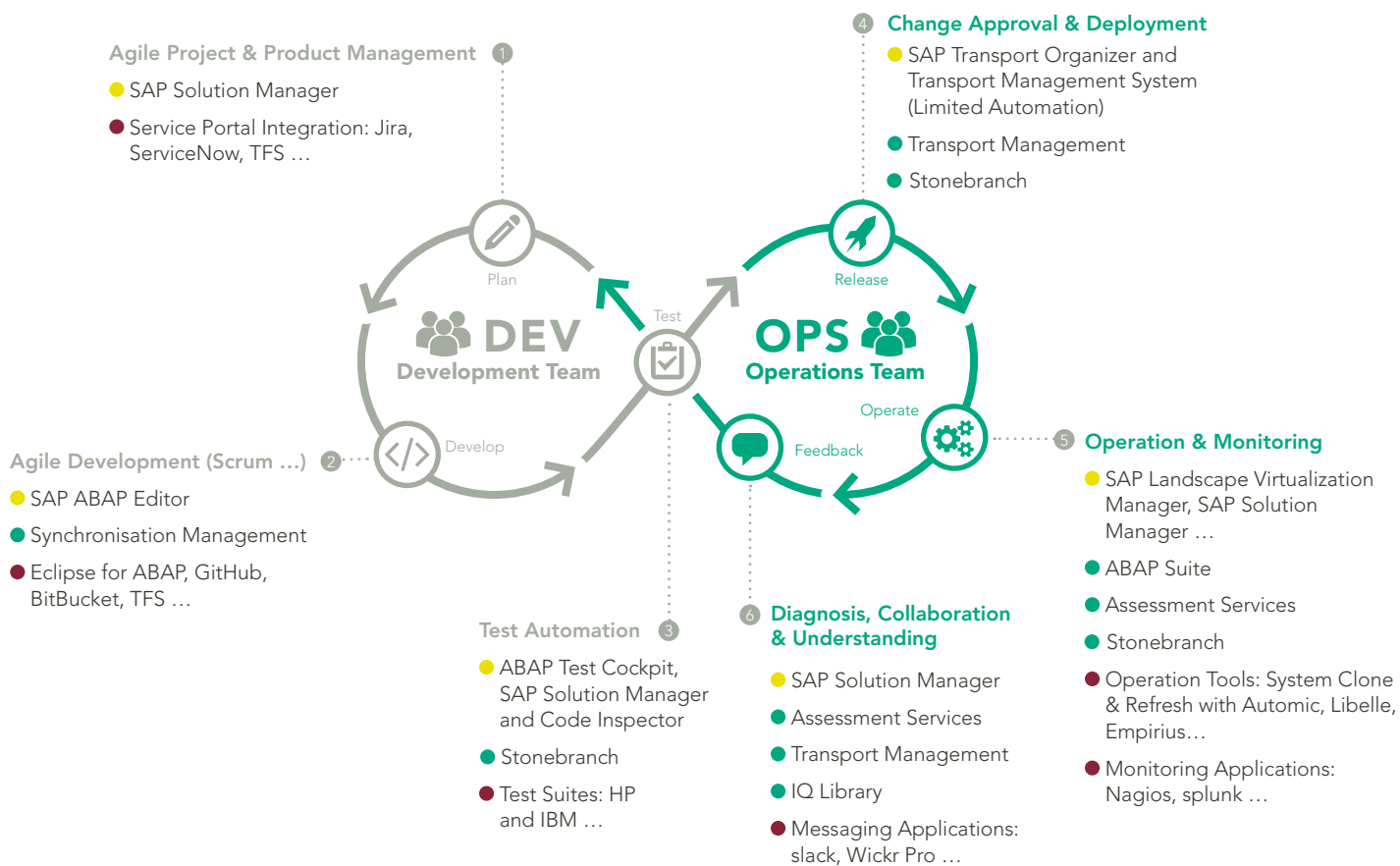
Produktivsystem. Anhand dieser Daten kann die Qualität der Entwicklungen sukzessive gesteigert werden, da Feedback über die aktuelle Performance einfließt und sich der Entwicklungsprozess so konsequent weiter verbessern lässt.

3. Kontinuierliches Feedback – insbesondere zwischen Entwicklung und Betrieb – muss als Kultur, Prozess und IT-Architektur in alle beteiligten Abteilungen getragen werden. Dies stellt sicher, dass in jeder Situation möglichst viel vorhandenes Wissen zur Problemlösung eingesetzt wird. Technisch kann das vor allem durch das Umsetzen von Schnittstellen zwischen SAP, Service Portal, Projektplanung und Scrum sowie durch die teamübergreifende Einführung von Kollaborationstools erreicht werden. Vor allem wird ein transparentes Arbeiten und eine hohe Eigenverantwortlichkeit erreicht.

4. Die **technische und prozessuale Integration** der SAP-Systemlinie mit Service Portal, Scrum und Versionsmanagement ermöglicht eine bessere Koordination der IT. Die Verknüpfung von Anfrage, Ticket, Entwicklungsaufgabe, Software-Inkrement und SAP-Transportauftrag ermöglicht einen möglichst vollständigen Überblick über Ablauf und Abhängigkeiten während aller Projekt- bzw. Produktphasen. Alle weiteren Systeme, die während des Entwicklungszyklus Daten zur Entwicklung speichern, sind ebenso Optionen für eine Integration in DevOps für SAP.

DEVOPS-LEBENSZYKLUS FÜR SAP-LANDSCHAFTEN

Der folgende grundlegende DevOps-Zyklus für SAP-Landschaften besteht aus sechs Prozessphasen, die iterativ durchlaufen werden. Das Feedback aus dem Betrieb bezüglich Performance oder anderer Auffälligkeiten im Nutzungsverhalten beeinflusst die agile Entwicklung und rundet somit den gesamten DevOps-Zyklus ab.

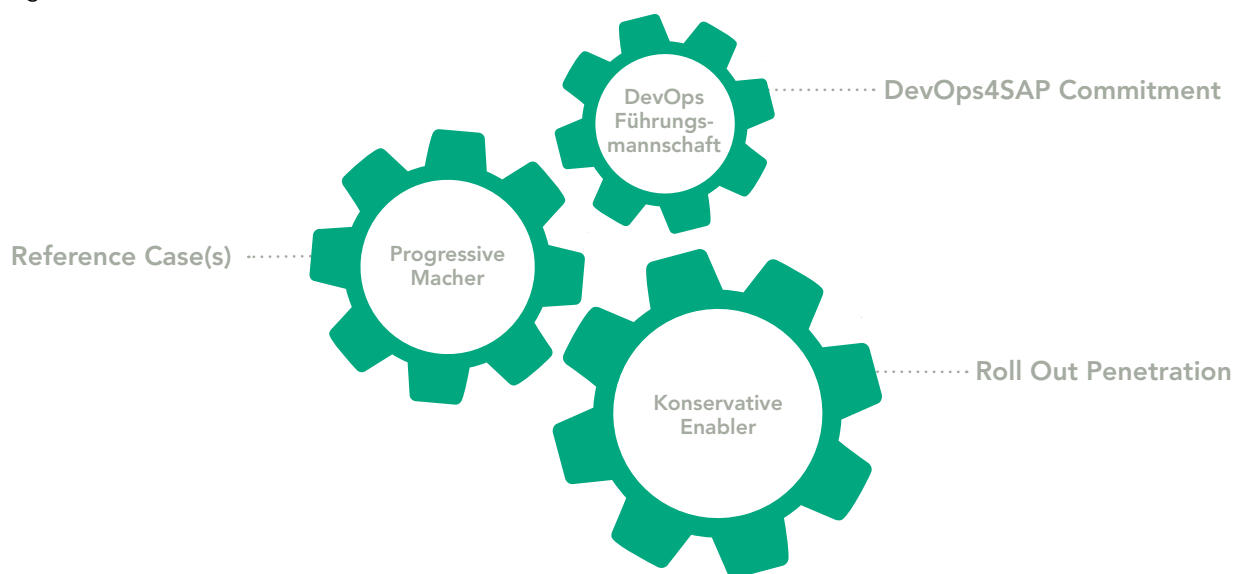


SCHRITTWEISES VORGEHEN UND KUPPLUNG/GAS-GESPÜR BEIM UMSCHALTEN

Der Weg von fragmentierten Entwicklungs- und Betriebsprozessen in einer klassischen SAP-Umgebung hin zu einer agilen DevOps für SAP-Organisation ist eine Reise. Diese Reise erfolgt in Phasen und verbessert über jeweils neue Iterationen die gesamte IT-Effektivität im digitalen Unternehmenskern der SAP-Systeme. Ziel der DevOps-Führungskräfte ist es, mit jeder Iteration eine positive Veränderung zu erzeugen.

Um den Prozess kraftvoll in Gang zu setzen, ist in drei Schritten vorzugehen. Diese erfolgen sequenziell und generieren miteinander die notwendige Schwungkraft für die Veränderung der Organisation.

Im ersten Schritt ist es die Aufgabe der Führungsmannschaft sich klar und unmissverständlich zu den Prinzipien einer DevOps für SAP-Organisation zu bekennen und die Umsetzung als gemeinsames Ziel zu konstatieren. Damit einher geht das veränderte Selbstverständnis in der Führungsrolle, d.h. das Umschalten von Druck auf Zug, damit Kollaboration und Emergenz entsteht. Verhaltensorientierte Führungskompetenzen und agiler Mindset sind zu etablieren und als Führungsstärken zu entwickeln (My Role – My Strength).



Erfolgsfaktoren bei der DevOps-Einführung für SAP. © 2018 REALTECH Deutschland GmbH.

Dem klaren Commitment des Führungskreises folgend werden im zweiten Schritt DevOps Reference Cases mit einzelnen Applikationen oder Applikationsportfolios bestimmt und durchgeführt. Unter gegebener Zielsetzung ist ein agiles Set-Up aufzusetzen, welches hohe Interaktion und Selbstorganisation im Team erzeugt. Das Team ist crossfunktional besetzt und besteht vornehmlich aus Akteuren mit hoher Agilitätsausprägung (Progressive Macher). Es werden Erfahrungen gesammelt und erste Ergebnisse mit DevOps für SAP erzielt, die in weitere Iterationen einfließen.

Im dritten Schritt erfolgt die Übertragung der DevOps-Prinzipien auf alle Software Development & Deployment Aktivi-

täten im klassischen SAP-Umfeld. Im Fokus dieses Roll Outs stehen insbesondere die konservativ eingestellten Mitarbeiter (Konservative Enabler), die sich eher in strukturierten als unstrukturierten Arbeitsumgebungen wohlfühlen. Es gilt, diese zu animieren, neue positive Erfahrungen durch dynamisch agile Arbeitsweisen aufzunehmen. Dem sukzessiven Verlassen der Komfortzone sollten erlebbare Positiveffekte folgen.

Der Auf- und Ausbau agiler Kompetenzen und Verhaltensmuster erfolgen durch gezielte Trainings-, Coaching- und On-the-job-Maßnahmen. Letzteres bedeutet, dass agile Organisationsformate in den operativen Projekten Anwendung finden. Darüber hinaus muss die Führungskraft den Mitarbeiter bzw. die Teams auf den nächsten Level führen, indem sie in Einzelsituationen die geeigneten agilen Stellhebel bedient.

Grundsätzlich ist bei der Planung des Einstiegs in DevOps zu beachten, ob und inwieweit im Unternehmen bereits Erfahrungen im Umgang mit agilen Führungs- und Steuerungsmethoden vorhanden sind. Entsprechend der gegebenen Erfahrungen und des agilen Reifegrades – insbesondere bei den betroffenen Akteuren – sind dann Inhalte, Umfang und Zeitrahmen der einzelnen Schritte festzulegen. Ein behutsames Vorgehen, das gewährleistet, dass alle Beteiligten mitgenommen werden und eine nachhaltige Transformation herbeigeführt wird, ist von großer Bedeutung. Das heißt, ein gutes Gespür für Kupplung und Gas ist angebracht.

Beim Übergang in die DevOps für SAP-Welt ist ein agiler Mindset auf Führungsebene zu etablieren.

FAZIT UND AUSBLICK AUF TEILE 2 UND 3 DES WHITEPAPERS

Traditionelle SAP-Betriebsorganisationen können sich dem steigenden Druck auf schnellere Veränderungen nicht dauerhaft entziehen. DevOps bietet auch für SAP on premise die Chance, eine gewachsene Architektur sinnvoll zu optimieren und für die digitale Welt zu rüsten. Die Notwendigkeit einer wesentlich schnelleren und automatisierten Auslieferung von SAP-Entwicklungen wird auf diese Weise in Einklang mit einem sicheren Systembetrieb gebracht. In diesem Whitepaper wurde dargestellt, welche Möglichkeiten SAP bereits heute bietet und was darüber hinaus im Hinblick auf die Integration der Komponenten notwendig ist. Dadurch wird eine kundenspezifische Anpassung des DevOps-für-SAP-Frameworks für den täglichen Systembetrieb und Eigenentwicklung ermöglicht.

In Teil 2 des Whitepapers zum Thema DevOps für SAP stellen wir Ihnen eine mögliche Toolchain für eine DevOps-fähige On-premise-Systemlandschaft vor.

Diese basiert auf den vier Komponenten:

- Transportautomatisierung
- Telemetrie und Metriken
- kontinuierliches Feedback
- technische und prozessuale Integration

Darauf aufbauend, wird sich der dritte Teil des Whitepapers mit Konzepten beschäftigen, die klassische SAP-Landschaften im eigenen Rechenzentrum und neue Cloud Services wie beispielsweise die SAP Cloud Platform kombinieren. Beide Welten, on premise und Cloud, können mit einer einheitlichen DevOps-Strategie harmonisiert werden.

SAP-Anwender, -Administratoren, -Entwickler und -Entscheider kommen so in den Genuss der Vorteile, die sich durch DevOps abseits von SAP entwickelt und bereits erfolgreich durchgesetzt haben.



© 2018 REALTECH Deutschland GmbH. Alle Rechte vorbehalten.

Die in diesem Whitepaper veröffentlichten Inhalte unterliegen dem deutschen Urheber- und Leistungsschutzrecht. Jede vom deutschen Urheber- und Leistungsschutzrecht nicht zugelassene Verwertung bedarf der vorherigen schriftlichen Zustimmung des Anbieters oder jeweiligen Rechteinhabers. Dies gilt insbesondere für Vervielfältigung, Bearbeitung, Übersetzung, Einspeicherung, Verarbeitung bzw. Wiedergabe von Inhalten in Datenbanken oder anderen elektronischen Medien und Systemen. Inhalte und Rechte Dritter sind dabei als solche gekennzeichnet. Lediglich die Herstellung von Kopien für den persönlichen und nicht kommerziellen Gebrauch ist erlaubt.

Alle Angaben und Beschreibungen in diesem Whitepaper stellen keine zugesicherten Eigenschaften dar. REALTECH übernimmt keine Verantwortung für Fehler und Lücken in diesem Whitepaper und kann die Inhalte jederzeit und ohne vorherige Ankündigung korrigieren oder ändern.



REALTECH DEUTSCHLAND GMBH
INDUSTRIESTRASSE 39C · 69190 WALLDORF · GERMANY
TEL. +49.6227.837.0 · FAX +49.6227.837.837
INFO@REALTECH.DE · WWW.REALTECH.DE