

[continuous|agile|lean] delivery auf dem Oracle Redstack

Tobias Weih
Initiative „Venisolvere“
Nürnberg

Schlüsselworte

- Continuous Delivery,
- Automated Infrastructure Provisioning,
- Oracle Cloud Control

Einleitung

Es ist geschäftskritisch die Durchlaufzeit von Änderungen an Software zu reduzieren um eine neue Version so schnell wie möglich in ein funktionierendes System zu wandeln. Während manche Unternehmen sich heute noch Zykluszeiten von mehreren Monaten erlauben, bringen andere schon mehrfach täglich eine neue Version in Produktion.

Zur Verkürzung der cycle-time und zur Reduktion des Release-Risikos ist eine konsequente Vollautomatisierung der Delivery und Quality Assurance über alle Stages hinweg essentiell – und das sowohl für Dev als auch Ops! Wie lange dauert ein Change in Ihrer Organisation, eine Änderung an einer einzigen Zeile Code in Produktion zu bringen?

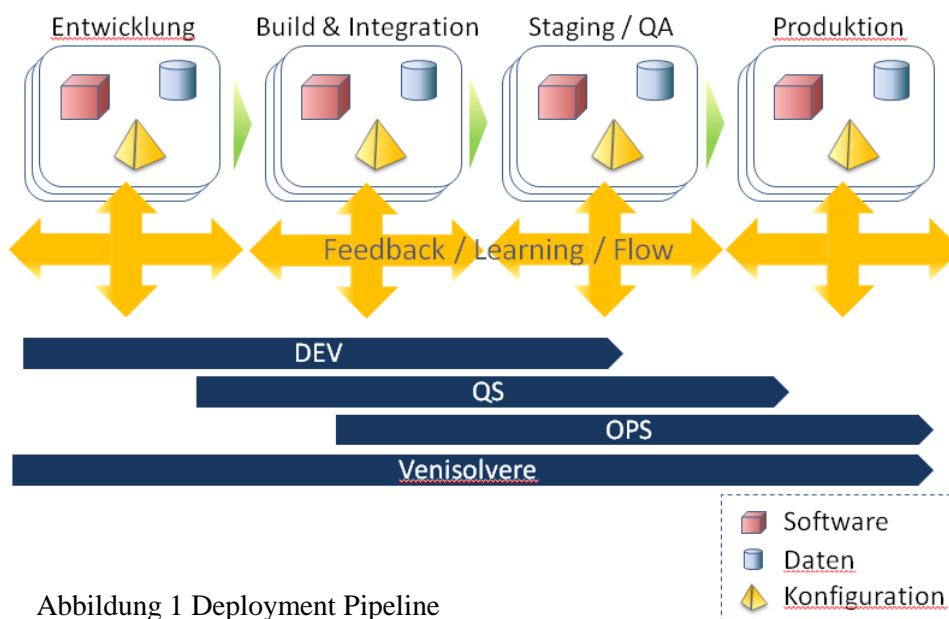


Abbildung 1 Deployment Pipeline

Im Geschäftsbereich Software Engineering (SE) reifen – insbesondere in der Entwicklung (Dev) - seit mehr als einem Jahrzehnt agile Praktiken und Methoden. Diese haben sich unter dem Stichwort Continuous Integration (CI) auch in einer unverzichtbaren Werkzeugkette niedergeschlagen. Im Betrieb (OPS) von Middleware hingegen, hinkt die Realität den Erfordernissen noch deutlich hinterher.

Für Middleware-Ops muss eine die Deployment-Pipeline implementierend Werkzeugkette vor Allem individuelle Konfigurationsprozesse abbilden können und diesen durch kontinuierlich und dezentral (weiter)entwickelter Automatisierungsketten und –bausteinen flexibel halten.

Neben einem anpassbaren Datenmodell in XML zur dezentral verantworteten Datenhaltung und inter-system-kommunikation über Webservices ist die strikte Versionierung der Konfiguration und die Nachvollziehbarkeit von Änderungen durch Logging und Reporting ist schlicht notwendig.

Nicht zuletzt ist auch eine Integrationen in weitere Prozessbereiche wie z.B. Product Lifecycle Management (PLM), IT-Störungsmanagement, etc. wünschenswert. So wird ein zentraler IT-Betrieb, bei gleichzeitiger dezentraler Konfiguration für Projekte, Mandanten oder Gruppen möglich.

Continous Delivery

Jeder kennt und viele fürchten diese Wochenenden, an denen ein System Upgrade ausgerollt werden soll, bevor am Montag sich die Sonne über den Horizont schiebt. Was nutzt ein integrierter (Oracle) Middleware Stack, wenn der Entwicklungs-, Deploy- und Releaseprozess nicht integriert sind? Wie stopft man das lang vergessene Loch zwischen Development und Operations? Wie bekommt man diese letzte Meile in den Griff? Eine funktionierende CI ist dabei nur ein erster Schritt! Neben Code sind Konfiguration, Middleware-Stack, Daten und Systemkontext fragil, komplex und kritisch.

„Deploy the same artifacts in every environment“

„Keep everything in version control“

„Automate almost everything“

Die Continous Delivery beschreibt den Prozess für DevOps. Ziel ist die Integration der verschiedenen Aufgaben und Funktionen in einer Deployment Pipeline, die Bereiche des Change-, Build-, Requirements-, Integrations-, Release- und Test-Management unter agiler Maßgabe implementiert. Dazu gehört:

- continuous integration and deployment
- data management
- configuration management
- environment management
- automated testing
- release management

Agile Delivery

Der zunehmende Einsatz komplexer Middleware verschiebt maßgebliche Aufwände aus der Softwareentwicklung (Dev) in den Betrieb (Ops). Auch die Middleware und damit Middleware-Ops, ist zunehmen im Sog dynamischer und sich entwickelnder Anforderungen. Und das meint wandelbare, sich im Zeitablauf entwickelnde Anforderungen an Middleware und deren effiziente und qualitativ hochwertige Umsetzung.

System Engineering ist im Betrieb (OPS) hingegen noch immer weitgehend einem statischen Idealbild verhaftet. Und die notwendigen Schritte hin zu einer anforderungsgetriebenen zunehmend mächtigen und komplexen Middleware stehen aus. Jede Änderung ist ein maßgebliches Risiko, das die Stabilität gefährdet; im Zeitalter eines „agile“ Megatrends.

Lean Delivery

Eines der Dinge, die Lean Management lehrt ist, das Ganze zu optimieren. Dazu gehört es, kurze und effektive feedback-Schleifen zu etablieren, die eine kontinuierliche und dezentrale Entstehung und Verbesserung ermöglichen. Der natürlich iterative Evolutionsprozess gilt gleichermaßen in beiden Bereichen; dem Software- und System-Engineering. Im Geiste des Lean Management gilt es, drei wesentliche Punkte in einer Continous Delivery zu implementieren: Feedback, Learning, Flow!

Automated Infrastructure Provisioning mit Cloud Control

Die oben genannten Lösungsansätze für die Bereitstellung der Software benötigen natürlich auch eine Plattform auf diese Software deployed werden kann. Hierbei stellen sich oft die folgenden Probleme: Bereitstellung von Hardware, OS, Storage, Netzwerkapparaten usw. erfordert Expertenwissen. Um dies in Kombination mit Oracle Software zu einem lauffähigen System zusammen zu bringen ist im Enterprise Umfeld oft ein langer und mühsamer Weg. In einfachen Umgebungen wird dann regelmäßig eine Kopie eines existierenden Systems benutzt. Dies ist aber in komplexen FMW und Fusion Apps Umgebungen nicht mehr machbar da es hunderte von Punkten gibt bei denen Feinheiten aufeinander abgestimmt werden müssen.

Der Lösungsweg von Venisolvere geht dahin um (abhängig vom Einsatzgebiet) mit vorkonfigurierten VM bzw. Scripted Oracle/OS Installationen Standardbausteine anzubieten. Bei häufig gefragten Komponenten (WLS, SOA Suite) ist ein VM Template der bessere Weg, bei anderen Komponenten bieten die Ready-To-Build Skripte einen Ausweg. Diese Bausteine werden dann im CloudControl eingebunden. Wichtig ist hierbei ein sogenanntes Late-Binding von Parametern, um die Bausteine genau so fertig zu stellen (finishing touch) wie sie in der zu bauenden Umgebung benutzt werden sollen. Die Kombination aus IaaS und PaaS bietet die Flexibilität die zurzeit oft benötigt wird aber selten geliefert werden kann.

Weiterhin sind diese Bausteine nicht statisch - sie unterliegen den Anforderungen des Lean-Konzeptes. Verbesserungen, Feedback und Optimierung sorgen für neue besser passende Bausteine.

Durch das Anbieten der Venisolvere Lösung werden die Übergänge zwischen den Entwicklern und DevOps besser definiert. Der Fokus der Entwickler bleibt auf dem Code während das DevOps Team mit einem Katalog der Angebote eine passende und gewünschte Ziellandschaft zur Verfügung stellt. Gerade beim DevOps Team geht dann der Weg vom reaktivem ("muss so schnell wie möglich fertig sein") zum automatisierten Bereitstellen der Umgebungen und zur Unterstützung (lean: Lernen, Verbessern, Wiederholen) der Entwickler ("Extra Umgebung für Fehlersuche").

Am sinnvollsten ist der Einsatz der Venisolvere Lösung bei Engineerd Systems (Exa*) aber auch in konventionellen Systemen ist dies gut ein zu setzen. Dies kommt natürlich durch den Einsatz der Provisionierungsmöglichkeiten des CloudControl.

Der große Unterschied bei Venisolvere ist die Positionierung des CloudControl. Da wir schon beim Start des Projektes CloudControl einbinden gibt es im Laufe des Projektes deutliche Vorgehensweisen wenn es um die Unterstützung des Lebenszyklus geht. Ob Bereitstellung von Umgebungen, Management von Artefakten und Konfigurationsversion, Performanceanalyse oder normalen Betrieb geht, die Werkzeuge und Übersicht schon vorhanden.

Das Ziel ist durch diese "Baustein-Sicht" ein Self-Service der Plattform zu ermöglichen. Ein Projekt entscheidet dann selber wann es welche Ressourcen benötigt. Durch Chargeback kann dann mit den Abnehmern entschieden werden ob Ressourcen noch benötigt werden.

Kontaktadressen der Initiative „Venisolvere“

Tobias Weih
CABP – Computer Aided
Business Processes
Thalkirchner Str. 25
D-80337 München

Telefon: +49 (0)89-729973 66
tobias.weih@cabp.de
www.cabp.de

Andreas Chatziantoniou
Foxglove-IT BV

Texel 18
NL-3524 AP Utrecht

Telefon: + 31623259167
andreas@foxglove-it.nl
www.foxglove-it.nl

Dirk Ruchatz
ITCSD

Lange Zeile 6
D-90519 Nürnberg

Telefon: +49 911 225251
+49 151 1475 1764
itcsd@gmx.de