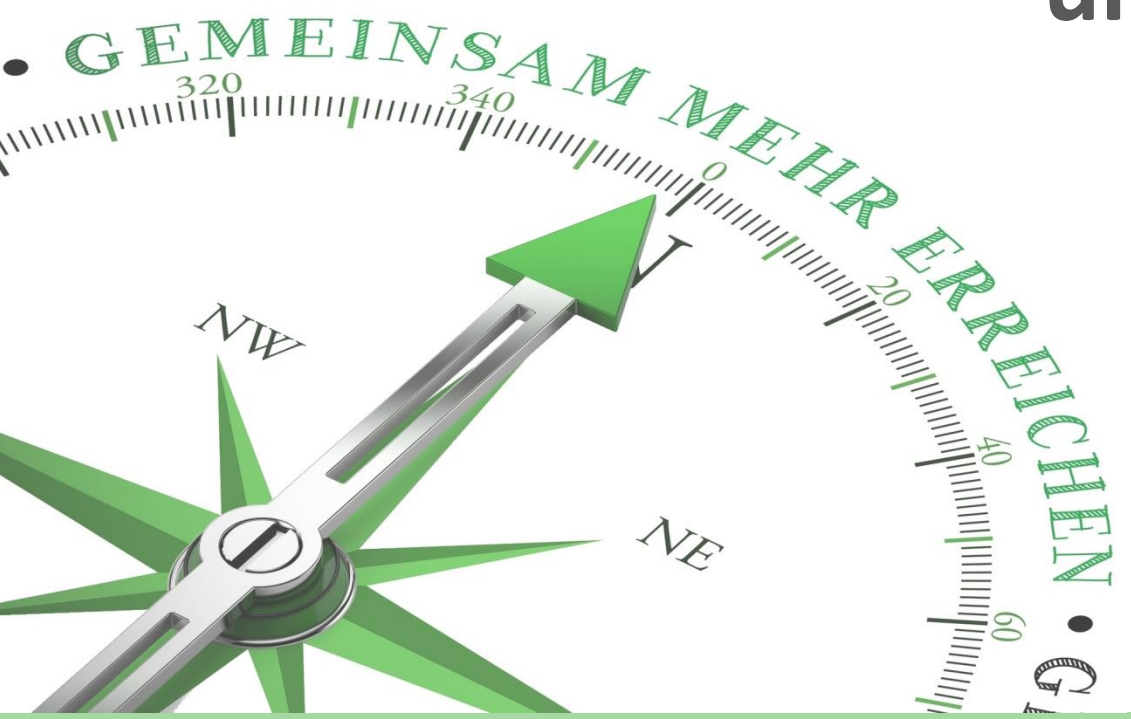
avision
REVIVING YOUR SOFTWARE

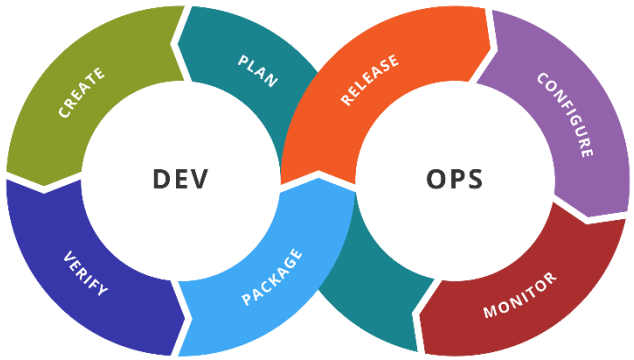
DevOps und die Sache mit dem „Flow“



Avision GmbH, 21.02.2018 Jürgen Bernert

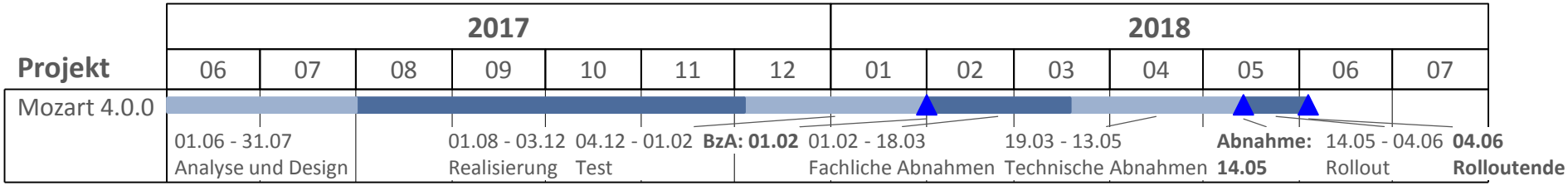
DevOps – was ist das eigentlich?

- **DevOps** ist ein Kunstwort aus Development und IT-Operations und wurde von Patrick Debois zum ersten Mal Oktober 2009 für eine Konferenz in Belgien (DevOpsDays) verwendet (https://blog.newrelic.com/2014/05/16/devops-name/).
- Es bezeichnet die verstärkte Zusammenarbeit zwischen Entwicklung und Betrieb mit Verwendung von Techniken, die auch in der agilen Softwareentwicklung verwendet werden. Daher wird es oft auch als Erweiterung der agilen Softwareentwicklung gesehen.
- Im Gegensatz zur agilen Softwareentwicklung, die vorrangig die Zusammenarbeit im Team zur Erstellung von Software verbessert hat, ist DevOps ein Wandel der Unternehmenskultur, der nicht nur Development und IT-Operations beeinflusst, sondern auch Unternehmenswerte wie Vertrauen, Offenheit, Wertschätzung usw.



Und warum DevOps?

Dazu müssen wir uns die Vorgehensweisen in der Softwareentwicklung ansehen:



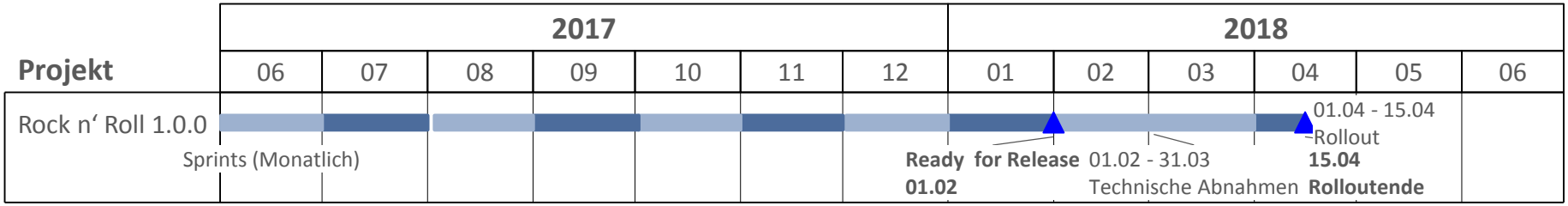
Das Projekt „Mozart“ entwickelt mit klassischem Wasserfallverfahren Software:

- Jedes Release dauert ca. ein Jahr bis in einem komplexen Rollout, der selten fehlerfrei abläuft, alles installiert wird.
- Während des Projektverlaufs kommt es immer wieder zu Knowledge Transfers zwischen unterschiedlichen Teams (Analyse und Design -> Realisierung -> Test -> Abnehmer der Software (Kunde) -> IT-Operations.
- Da fast kein Rollout fehlerfrei durchläuft werden die Genehmigungsprozesse immer komplizierter und es müssen immer längere Checklisten ausgefüllt werden. Genehmigungen erfolgen nun auf einer Hierarchieebene, die sehr weit von der Entwicklung entfernt ist und die Änderungen technisch auch nicht mehr einschätzen kann.
- Da der Abnahme- und Rolloutprozess so aufwändig ist, werden immer mehr Features in ein Release gepackt, dass daraufhin immer komplexer wird.

Fazit: Jeder Rollout ist immenser Stress für alle Beteiligten und Ausgangspunkt für das „Blame Game“ bei dem sich alle gegenseitig beschuldigen, wenn etwas schiefläuft. Die beteiligten Entwickler erhalten erst Monate nach Abschluss ihrer Arbeit ein Feedback zur Qualität.

Und warum DevOps?

Die agilen Methoden verbessern die Situation während der Entwicklung:



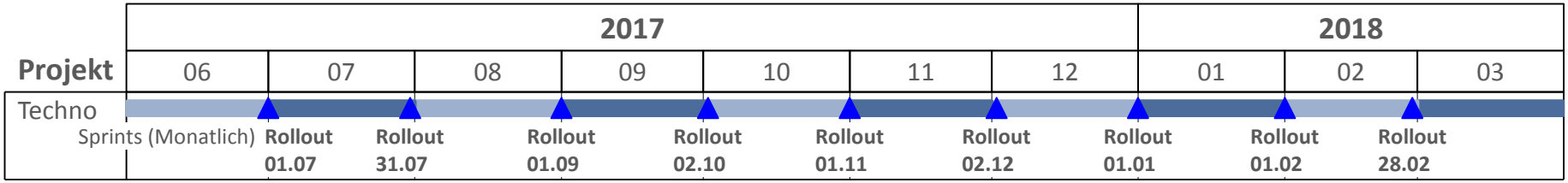
Das Projekt „Rock n' Roll" entwickelt mit agilen Methoden (z.B. Scrum) Software:

- Jedes Release dauert weniger als ein Jahr bis in einem komplexen Rollout, der selten fehlerfrei abläuft, alles installiert wird.
- Am Ende des Projektes erfolgt der Knowledge Transfer zu IT-Operations, obwohl bereits nach jedem Sprint ein „potentially releasable Product" bereit stehen würde.
- Da fast kein Rollout fehlerfrei durchläuft werden die Genehmigungsprozesse immer komplizierter und es müssen immer längere Checklisten ausgefüllt werden. Genehmigungen erfolgen nun auf einer Hierarchieebene, die sehr weit von der Entwicklung entfernt ist und die Änderungen technisch auch nicht mehr einschätzen kann

Fazit: Obwohl die agile Entwicklung bereits Analyse, Design, Realisierung, Test und auch Abnahme in kürzeren Phasen vereint und auch installierbare Produkte in kurzen Zyklen hervorbringt, ist die Time-to-Market nicht wesentlich kürzer.

Und warum DevOps?

DevOps erweitert das agile Vorgehen auf den Rolloutprozess:



Das Projekt „Techno“ entwickelt mit agilen Methoden (z.B. Scrum) und DevOps Software:

- Nach jedem abgeschlossenen Sprint erfolgt auch die Produktivsetzung der Software.
- Ein Knowledge Transfer erübrigt sich, IT-Operations ist Bestandteil des Entwicklungsteams.
- *„You build it, you run it*. —Werner Vogels, CTO, Amazon: Das Entwicklungsteam produziert qualitativ hochwertigere und zuverlässigere Software, da es auch für den produktiven Einsatz verantwortlich ist.

Fazit: DevOps sorgt für eine deutlich verringerte Time-to-Market und insgesamt für weniger Stress bei den Teams, schnelleres Feedback an die Entwicklung und weniger Probleme bei der Produkteinführung.

Und die Sache mit dem „Flow“?

avision
REVIVING YOUR SOFTWARE

DevOps basiert auf wenigen Prinzipien:

- Ein schneller und widerstandloser Arbeitsfluss (der „**Flow**“), auch als **Erster Weg** bezeichnet
- Feedback ohne Schuldzuweisungen, der **Zweite Weg**
- Und als **Dritter Weg**: Unternehmensweites Lernen und Experimentieren

Der „Flow“ ist der Erste Weg und beschreibt den widerstandslosen Arbeitsfluss, der ein stressfreies durchgängiges Arbeiten mit hoher Qualität ohne Unterbrechungen ermöglicht.

Warum „Flow“?

Ein schneller und widerstandsloser Arbeitsfluss sorgt dafür, dass sich Arbeit nicht an einer Arbeitsstation stapelt („Flaschenhals“) und weitere Arbeitsstationen warten müssen (das ist bei physischer Arbeit natürlich deutlich besser zu beobachten).

Die folgenden Punkte haben sich als besonders effektiv für einen funktionierenden „Flow“ gezeigt:

- Umgebungen durch Self-Service bereitstellen (und das bedeutet nicht, dass eine Mail an IT-Operations zum Aufsetzen einer Umgebung generiert wird)
- Automatisierte Deployments ohne manuelle Schritte
- Automatisierte Tests
- Architekturen, die durch lose Kopplung das Arbeiten mit mehreren Teams und selektives Deployment ermöglichen

Geht das noch besser?

Durch Verringerung von Batchgrößen wird der Arbeitsfluss optimiert:

- Kleine Batchgrößen werden schneller ausgeliefert. Bei der Softwareentwicklung bedeutet das: nur wenige Features in einen Sprint, aber dafür die Sprintdauer reduzieren
- Bei Fehlern sind nur wenige Features betroffen und Nacharbeiten gehen zügig und sind kostengünstig.
- Das Ziel ist der Single Piece Flow

Was hindert den „Flow“?

Übergaben stören den „Flow“:

- Jede Übergabe wird begleitet von Knowledge Transfers, Dokumenten, Zeitverzögerungen durch Urlaube usw.
- Übergaben sorgen für Wartezeiten, das kann in extremen Fällen zu deutlichen Verzögerungen im Arbeitsfluss führen (Bei Microsoft addierten sich diese Wartezeiten im Jahr 2005 bei einer 20 Tage Aufgabe auf zusätzliche 135 Tage[1])
- Selbst im besten Fall wird bei der Übergabe Wissen verloren gehen

Work in Progress ist eines der größten Hindernisse im Flow

- Reduzierung der Work in Progress (WiP): „Stoppen Sie das Beginnen, Beginnen Sie mit dem Beenden“ (David J. Andersen: *Kanban: Successful Evolutionary Change for Your Technology Business*)

1) “*From Worst to Best in 9 Months: Implementing a Drum-Buffer-Rope Solution in Microsoft’s IT Department*“ von David J. Anderson und Dragos Dumitriu

Was hilft den „Flow“?

- Arbeit sichtbar machen, z.B. mit Hilfe von Kanban Boards
- Regelmäßigkeit und Wiederholung
  Das gilt auch für Notfallübungen und Verbesserungen
  (Verbesserungs-KATA von Mike Rother)
- Standardisierungen
- Wartezeiten minimieren
- Tätigkeiten auf das Ziel ausrichten, unnütze Arbeit eliminieren
- Automatisierungen

Und was ist mit „Wartung“?

avision
REVIVING YOUR SOFTWARE

DevOps kennt keine „Wartung“ im Sinne einer zusätzlichen Zwischenschicht zwischen Development und IT-Operations:

Jede zusätzliche Übergabe und jeder zusätzliche Prozess behindert den Flow

Know-How und Tätigkeiten im produktiven Betrieb, z.B. Auswertung und Validierung von Telemetriedaten einer Anwendung verteilen sich auf Mitarbeiter im Development und IT Operations.

Trotzdem löst DevOps nicht ITIL ab: *„Auch wenn manche DevOps als Gegenbewegung zu ITIL (IT Infrastructure Library) und ITSM (IT Service Management) sehen, sind DevOps und ITIL ebenfalls miteinander kompatibel. ITIL und ITSM bleiben weiterhin die besten Umsetzungen der Prozesse, die in IT-Operations genutzt werden, und tatsächlich beschreiben sie viele der Fertigkeiten, die IT-Operations braucht, um einen Arbeitsfluss im DevOps-Stil zu unterstützen.“*[1]

1)„Projekt Phoenix: Der Roman über IT und DevOps - Neue Erfolgsstrategien für Ihre Firma“ von Gene Kim, Kevin Behr und George Spafford

Wie geht es weiter?

avision
REVIVING YOUR SOFTWARE

DevOps ist eine Veränderung der Unternehmenskultur. Aber die Anstrengung lohnt und es gibt viele erfolgreiche Unternehmen, die mit DevOps deutlich flexibler und schneller am Markt agieren können.

Firma	Auslieferungs-frequenz	Code Deployment Lead Time	Zuverlässig-keit	Reaktion auf Kundenwünsche
Amazon	23.000/Tag	Minuten	hoch	hoch
Google	5.500/Tag	Minuten	hoch	hoch
Netflix	500/Tag	Minuten	hoch	hoch
Facebook	1/Tag	Stunden	hoch	hoch
Twitter[a]	3/Woche	Stunden	hoch	hoch
Klassische Firma	1/alle 9 Monate	Monate oder Quartale	niedrig/ mittel	niedrig/mittel

[a] Für ein monolithisches Ruby on Rails-Frontend.

Quelle:
Projekt Phoenix von Gene Kim, Kevin Behr und George Spafford

Auch interessant:

- **Das DevOps Handbuch** von Gene Kim, Jez Humble, Patrick Debois & John Willis, (O'Reilly 2017, ISBN 978-3-96009-047-2)
- **Projekt Phoenix: Der Roman über IT und DevOps - Neue Erfolgsstrategien für Ihre Firma** von Gene Kim, Kevin Behr und George Spafford (O'Reilly 2015, ISBN 978-395875-175-0)
- https://www.codecentric.de/publikation/die-devops-bewegung/

Der erste Weg: „Flow“ – Gibt’s auch einen zweiten?

Danke für die Aufmerksamkeit!

Fortsetzung folgt mit:

DevOps – Das Feedback