

DevOps

Dev (Developers) Ops (Operations) → Schulterschluss zwischen Softwareentwicklern und IT-Betrieb

Konflikt zwischen Entwicklung und Betrieb

Entwicklung von neuen Features ist attraktiv für die Entwickler, jedoch ist es Ihnen egal ob es auch auf den gewünschten Systemen problemlos läuft. Dadurch wird die IT gefordert und muss dann das Ganze mit aufwändiger Arbeit verfügbar machen. (Anreize beider Systeme sind gegensätzlich)

Falls ein Problem auftritt wird immer zuerst spekuliert bevor eine Lösung gesucht wird. Das Blame Game wird gestartet. Durch die Agilen Methoden wird dieses Problem noch verschärft, da die Entwicklungszyklen immer kürzer werden.

All diese Probleme hatte zur Wirkung, dass DevOps überhaupt entstanden ist. Entwickler werden von IT-Betrieben beigezogen um zusammen in Agilen-Methoden optimale Lösungen zu finden.

DEV-Ops Ziele/Grundprinzipien:

- **Zusammenarbeit**
Wichtigster Aspekt ist die Zwischenmenschliche Komponente (gegenseitiger Respekt)
- **Automatisierung**
Infrastructure as Code → Wichtiger Bestandteil für Tests da nun auch die Betriebe das Ganze testen können. Voraussetzung sind extra dafür entwickelte Tools welche von beiden Seiten genutzt wird.
 - Zentrale Verwaltung des Quellcodes unter Nutzung von Versionskontrolle
 - Hohe Transparenz und Vermeidung von Wissensinseln
 - Automatisierte Tests der Konfiguration von Servern und VMs
 - Automatisierte Tests von Deployment und anschliessender Verfügbarkeit von Systemen
 - Gemeinsame Nutzung von Konfiguration. Und Deployment-Vorschriften durch Entwicklung und Betrieb
- **Prozesse**
In diesem Bereich gibt es noch grosse Defizite, welche geklärt werden müssen. Da es Unterschiede zwischen KMUs und Grossunternehmen gibt ist es schwer geeignete Prozesse zu definieren Darum ist es umso wichtiger, dass es getan wird.

Missverständnisse und Kritik

Ziele von DevOps sind nachvollziehbar und auch nicht neu.

- DevOps ist nur Werbename um altes Problem neu anzupreisen
- DevOps ist keine Geldmaschine
- DevOps ist ein Freifahrtsschein für Entwickler Schaden anzurichten (Stimmt nicht es gibt Möglichkeiten diese einzuschränken und trotzdem Infrastructure as Code einsetzen.)
- Nicht Alleskönner erschaffen (Entwickler und Betrieb) sondern die Kommunikation fördern und verbessern
- Keinen Respekt voneinander (Zuerst muss man einander verstehen bevor man kritisieren kann. Problem ist nicht DevOps sondern wir selber)
- Cloud löst das Problem (ist nur eine weitere Schnittstelle, jedoch wird immer noch ein Betrieb benötigt.)
-

Was ist DevOps wirklich

Verfolgt gleiches Ziel wie agile Methoden einfach abteilungsübergreifend

Wegweisend sind folgende zwei Fragen:

1. Was ist der Anreiz für Entwicklung und Betrieb, viel Zeit und Aufwand zu investieren obwohl man schon zu wenig Zeit hat?
➔ Darum muss man zuerst einen Anreiz schaffen um das Ganze in Schwung zu bringen und die Realisierung voranzutreiben
2. Anreiz für Geschäftsführung?
➔ DevOps wird erst interessant wenn es einen (finanziellen) Mehrwert bringt.

Diese Fragen zeigen auf, dass das Problem des Anreizes für die, welche Entscheiden, nicht vorhanden sind und man nur eine Symptombekämpfung macht anstatt das Problem am Kern zu packen.

Darum ist es wichtig die bestehenden Probleme als ein Geschäftsproblem anzusehen.

Wie kann man die Leute überzeugen:

- Vorteile in Business Case aufzeigen
 - o Berichte von Erfolgstorys veröffentlichen damit die Entscheider darauf aufmerksam werden (Das gleiche welches man auch bei den Agilen-Methoden gemacht hat)
- Aufzeigen, dass man durch besserer Zusammenarbeit die Zufriedenheit der Kunden schneller und langfristiger erhält.

Was kommt mit DevOps auf uns zu

Developer:

- Mit Aspekten aus Betrieb auseinandersetzen
- Bessere Planung und Durchführung von Deployments (VMs für testst)
- Tools für Automatisierung einsetzen (Hilfe von Betrieb ➔ wir werden von Ihnen lernen)
- Mehr Verantwortung für unsere Software übernehmen
- Schnelle Troubleshooting unterstützen
- Flags einbauen um allfällige Features von aussen wider zu deaktivieren.

IT-Betrieb

- Mehr auf Automatisierung mit Infrastrukturen as Code und Versionskontrollen setzen.
➔ Softwareentwicklung wird uns dabei unterstützen
- Agile Methoden wie Kanban anwenden, möglicherweise gemeinsam mit Entwicklung
- Auf häufigere Releases einstellen. Diesen Prozess mit den Entwicklern abstimmen

Bei beiden werden wir uns auf zusätzliche Metriken einstellen müssen um das Ganze zu messen und der Geschäftsleitung aufzuzeigen, dass es eine Ersparnis bringt.