WHITEPAPER

Der Weg zum DevOps-Erfolg

Von Datensammlungen zu Antworten – so gelangen Sie zum Erfolg mit DevOps

splunk>
turn data into doing®

So schnell wie das Internet ist auch die digitale Wirtschaft. Entsprechend müssen Unternehmen neue Software und verteilte Services schneller denn je bereitstellen und weiterentwickeln – mit agiler und iterativer Software-Entwicklung und -Bereitstellung. Dank häufigen, kleineren Updates können Entwickler schnell neue Funktionen austesten.

Häufige Updates sind grundsätzlich sinnvoll:

- Die Apps bleiben aktuell und bei den Kunden präsent.
- Sie können auf Bedrohungen oder operative Notwendigkeiten rasch reagieren.
- Das Risiko, dass Sie kritische Komponenten beschädigen, wird minimiert.

Im digitalen Business, wo sich Firmen mit Cloud-Services und serviceorientierten Workloads, mobilen Apps und Social Media laufend neu erfinden, sind schnelle Release-Zyklen gang und gäbe. Diese Zyklen haben neue Architekturen, Entwurfsmuster, Anwendungsstacks und Continuous Delivery Release Trains hervorgebracht.

DevOps ist ein Ansatz der Zusammenarbeit, der sich auf den gesamten Lebenszyklus bezieht. DevOps ist die Art und Weise, wie moderne Unternehmen Anwendungen entwickeln, bereitstellen und Support leisten. Unternehmen mit DevOps agieren schneller und flexibler und können so mit den Kundenbedürfnissen in einer sich rasch wandelnden digitalen Welt Schritt halten.

Aber die Hoffnungen der DevOps-Anhänger zerschellen allzu oft an den Klippen der Realität. DevOps kann seine Versprechen nicht immer einlösen – oft aufgrund von technologischer und bürokratischer Komplexität. Eine agile, mobile, datengestützte Kultur samt unterstützenden Prozessen aufzubauen, ist leichter gesagt als getan. Die Komplexität von Infrastruktur und Anwendungen, App-Telemetrie und Datenanalysen erweist sich häufig als Hürde auf dem Weg zum DevOps-Erfolg.

Daten sind weit mehr als nur Logs. Daten kommen aus Anwendungen, Clouds, Infrastrukturen, Containern und Mobil-Apps, aus dem IoT, aus Streaming-Netzwerken und vielem mehr. Die Performance und sämtliche Aktivitäten aller Apps, Systeme, Kunden, APIs und Produktionsanlagen generieren Logs, Metriken, Traces und Events.

Das übliche Performance-Monitoring reicht für den DevOps-Erfolg also nicht aus. Vielmehr ist eine Analyse all dieser Daten erforderlich, damit Entwickler und Operations-Teams sich auf den geschäftlichen Erfolg konzentrieren können, statt ständig Feuerlöscher zu spielen. Dann finden sie Antworten in Echtzeit und quer durch den gesamten Technologie-Stack. So können die Teams die Auswirkungen auf die Endanwender schnell erkennen und Problemen vorbeugen, die dem Geschäft schaden könnten.

Die DevOps-Realität und der Acht-Phasen-Zyklus

DevOps bringt agile Entwicklungsmethoden mit sich, serviceorientierte, flüchtige Workloads und Prozesse in andauernder Iteration. Es wird laufend Code definiert, getestet, bereitgestellt, überwacht und gemessen. Wo Probleme auftreten, wird der Code geändert, und der Prozess beginnt von Neuem – immer wieder.

„Code" bezieht sich hier nicht nur auf Anwendungscode, sondern ebenso auf die deklarative Architektur, die Netzwerkkonfiguration, das Monitoring etc. Der gesamte Prozess erfordert Geschwindigkeit, Agilität und ein Mindset, das nicht große, grundlegende Software-Releases anpeilt, sondern inkrementelle Optimierung und dabei wiederholtes Scheitern akzeptiert.

Der Schlüssel zu einem erfolgreichen DevOps-Feedback-Loop sind Daten – das Messen und Analysieren von Nutzung, Leistung, Fehlern und Business-Schlüsselmetriken. Observability-Plattformen, die flexibel genug sind, sodass sie beliebige Daten aufnehmen, KI-gestützt analysieren und als Reports ausgeben können, passen hervorragend zu solchen DevOps-Prozessen.

Komplexität ist der Feind aller Agilität. Und Agilität ist die Grundlage für DevOps. Ein typischer DevOps-Prozess besteht aus acht Phasen mit jeweils eigenen Tools (Abb. 1). Da diese Tools häufig Open Source oder Einzellösungen sind, haben sie untereinander wenig oder gar keine Integration. Das führt zu Überschneidungen und Ineffizienzen, welche die Transparenz trüben und die Agilität lähmen. Agilitätseinbußen bedeuten aber, dass Probleme länger als nötig bestehen bleiben. Sie werden ausgebremst, und das kostet Sie bares Geld.

Abbildung 1: Die wichtigsten Phasen eines typischen DevOps-Prozesses.

Die DevOps-Phasen werden üblicherweise in der Form eines Unendlich-Zeichens dargestellt, weil es die Koppelung der beiden klassischen Prozessverantwortungsbereiche gut veranschaulicht. Die linke Seite zeigt den Zyklus von Entwicklung und Test („Dev"), die rechte den Zyklus von Bereitstellung und Betrieb („Ops"). Trotzdem steht aus Tool-Perspektive jeder einzelne Schritt im Zyklus für eigene Produktmärkte, und in jeder Phase kommen oft ein Dutzend oder mehr Tools zum Einsatz.

Für Entwickler- und Operations-Teams ist es natürlich toll, dass sie so viel Produktauswahl haben, weil jedes Team nach Bedarf und Vorlieben wählen kann. Doch das Sammelsurium, das sich daraus ergibt, ist ein Albtraum für Projektmanager, DevOps-Verantwortliche und Führungskräfte. Ein derartiger Mix von Systemen, von denen jedes isoliert Daten ausspuckt, macht die Gesamtprozesse praktisch undurchschaubar und führt zu weiterem Tool-Wildwuchs. Hinzu kommt, dass Dev- und Ops-Teams keine gemeinsame Sprache und keine gemeinsamen Troubleshooting-Abläufe haben, was bei Ausfällen eine längere MTTR (Mean Time To Repair) zur Folge hat.

Der Mangel an gemeinsamen Tools bedeutet oft, dass man keine Möglichkeit hat, zu beurteilen, ob ein Eingriff wirklich das Grundproblem behebt oder nur ein vorübergehender Patch ist. Wenn die Teams nicht vollständigen Einblick in die Gesamtheit der Daten aus Entwicklung, Release und Nutzung haben, können sie die Qualität, Leistung und Sicherheit von Apps gar nicht beurteilen. So kann z. B. eine Security-Monitoring-Software eine Schwachstelle erkennen, aber ohne die Möglichkeit, das Problem zum Software-Ursprungsmodul zurückzuverfolgen, lässt sich die Schwachstelle nur schwer beheben.

Darüber hinaus können Unternehmen, denen der Einblick in die DevOps-Metriken fehlt, nicht die Agilität erreichen, die sie brauchen, wenn sie neue Ideen ausprobieren wollen oder iterativ auf ein verändertes Geschäftsumfeld reagieren müssen. Das Ergebnis ist schlechte Qualität, und das bedeutet Kunden, die unzufrieden sind und am Ende abwandern.

Disparate Build-Pipelines machen bei DevOps mehrfache, zeitaufwendige Übergaben im gesamten Prozess der Anwendungsbereitstellung erforderlich. Diese Übergaben bergen die Gefahr, dass sich redundante Funktionen einschleichen und aufgrund lästiger Aufgabenwechsel die Effizienz leidet. Im Schnitt dauert es nach einer Unterbrechung 10 bis 15 Minuten, bis man wieder zur Arbeit am Code zurückkehren kann.*

Wenn die Entwickler beim ganz normalen Troubleshooting zu ständigen Kontextwechseln gezwungen sind, hat dies ausgesprochen negative Auswirkungen auf die Produktivität. Ein fragmentierter Delivery-Workflow bedeutet auch, dass die Zusammenarbeit ins Hintertreffen gerät, dass es Lücken zwischen den Abteilungen gibt, weil die Teams eigene Technologien verwenden und das Projekt als Ganzes aus einer eigenen Perspektive betrachten. Insgesamt untergraben die Reibungsverluste bei Kommunikation und Informationsaustausch das Vertrauen und die Zusammenarbeit – und das steht dem Grundgedanken von DevOps diametral entgegen.

* Programmer Interrupted, ninlabs research, 19. 01. 2013.

Ohne eine Observability-Plattform, die datengestütztes Feedback liefert, kann DevOps kaum erfolgreich sein.

Das Potenzial von DevOps voll ausschöpfen

Für die meisten Führungskräfte ist DevOps nur ein Buzzword. Wer mit dem Konzept nicht gut vertraut ist, hat dann oft unrealistische Erwartungen. Man erhofft sich alles, was DevOps leisten kann. Wo aber das Verständnis für die Prinzipien, Prozesse und Herausforderungen fehlt, sind Enttäuschungen vorprogrammiert.

Die beste Chance, dieser Falle zu entgehen, besteht darin, den DevOps-Erfolg anhand von Metriken zu messen, darzustellen und zu kommunizieren – und ihn zugleich durch datengestützte Erkenntnisse weiter zu verbessern. Mit anderen Worten: Ohne eine Observability-Plattform, die datengestütztes Feedback liefert, kann DevOps kaum erfolgreich sein.

DevOps wird dann wertvoll, wenn nicht nur das Schlagwort im Unternehmen kursiert, sondern wenn Softwarequalität und Bereitstellung eine gemeinsame Sprache finden und einen gemeinsamen Ansatz verfolgen.

DevOps in der Praxis

Erfolgreiche Unternehmen richten sich auf die Kundenbedürfnisse aus. Das erfordert eine Observability-Plattform, die fortlaufend Feedback liefert und damit alle Bereiche des Unternehmens unterstützt: IT, App-Entwickler, Geschäftsentscheider, Sicherheitsteams, Auditoren etc. Mit Erkenntnissen und Entscheidungsunterstützung auf der Grundlage belastbarer Daten versetzt eine Observability-Plattform Unternehmen in die Lage, schnell zu agieren, um Kosten zu sparen, gute Ideen umzusetzen und schlechte auszusortieren.

Wer das richtig angehen will, braucht Daten, Tools und Prozesse, die allen Beteiligten kontinuierlich Einblick in sämtliche Workloads geben. Mit DevOps können Sie auf die so gewonnenen Erkenntnisse schneller reagieren, und zwar quer durch die Geschäftsbereiche, Anwendungen und die kompletten IT-Operations.

In der Praxis hat die erfolgreiche Kombination eines DevOps-Modells mit umfassenden, KI-gestützten Analysen drei wesentliche Vorteile: schnellere Anwendungsbereitstellung, bessere Anwendungsqualität und bessere Business-Qualität.

Kernkompetenzen

Agilität und Kundenzufriedenheit erfordern von DevOps-Teams vier wichtige Fähigkeiten:

1. **Observability:** Jedes Element des DevOps-Workflows muss so instrumentiert sein, dass die Daten zur weiteren Analyse in einem Master-Repository erfasst werden. Ohne Daten lassen sich Probleme weder verstehen noch beheben.
2. **Iteration:** Code-Fixes und Verbesserungen müssen anhand der korrelierten Daten aus der gesamten Tool-Kette schnell ermittelt, selektiert und entwickelt werden.
3. **Zusammenarbeit:** Eine schnelle Bereitstellung funktioniert nur, wenn die DevOps-Teams auf dem gleichen Stand sind, die gleichen Daten verwenden und mit den gleichen Messwerten arbeiten.
4. **Optimierung:** Die Führungsebene muss sich unablässig um Prozessoptimierung bemühen und dazu faktenbasierte Entscheidungen treffen. Zur Prozessoptimierung sind datengestützte Antworten auf Fragen wie diese erforderlich:
   - Welches „Idea-to-Cash"-Verhältnis haben wir, wie stark ist unsere End-to-End-Ertragskraft?
   - Wie lange dauern die einzelnen Phasen unserer Bereitstellungspipeline?
   - Wie viel Zeit verbringen die Teams mit dem Schreiben, Testen und Überprüfen von Code?
   - Welche Entwicklungsteams sind die produktivsten?

Die Vorteile von datengestützten DevOps schlagen sich direkt im Geschäftsergebnis nieder: Durch größere Effizienz und Entwicklungsproduktivität schafft das Unternehmen eine schnellere Anwendungsbereitstellung, geringere Kosten, größere Kundenzufriedenheit und höheren Umsatz.

Bessere Apps dank Observability-Erkenntnissen in Echtzeit

Die Daten aus der gesamten DevOps-Tool-Kette liefern Informationen, die proaktive Antworten auf Probleme schon früh im Entwicklungs- und Testzyklus möglich machen. Mit Observability können Entwickler und Operations-Teams Probleme lösen, bevor nicht funktionierenden Features oder abstürzenden Apps die Kunden und Anwender frustrieren.

Solche proaktiven Maßnahmen erfordern eine gemeinsame Observability-Plattform, die alle, auch die DevOps-Teams, als „einzige Quelle der Wahrheit" (Single Version of The Truth) zugrunde legen – etwa so, wie Entwickler mit einer Quellcode-Versionsverwaltung arbeiten, damit sie ihre Arbeitsergebnisse konsolidieren und teilen können.

Die Instrumentierung und Analyse des gesamten DevOps-Prozesses bringt die tatsächlichen Leistungs-, Nutzungs- und Fehlerdaten ans Tageslicht, die für die Optimierung des Endprodukts und der gesamten Bereitstellung entscheidend sind. Eine gemeinsame Datenplattform ermöglicht die übergreifende Korrelation von Informationen über Tools und Infrastrukturen hinweg und kann proaktiv auf Probleme hinweisen. So kann etwa die Korrelation von Daten aus Codeüberprüfungen und Performance Monitoring Probleme aufzeigen, ehe die Anwender mit Bug-Meldungen beginnen.

Allerdings ist zur Konsolidierung von Daten aus dem gesamten DevOps-Bereitstellungszyklus eine Plattform nötig, die in Echtzeit Daten von sämtlichen Tools erfassen kann, die im oben beschriebenen Acht-Phasen-Zyklus zum Einsatz kommen. Und sowohl die Entwicklungstools als auch die Daten selbst ändern sich ständig.

Daten sind der Rohstoff der DevOps-Messung. Aber sie müssen aus objektiven Metriken stammen, die quantifizieren, ob der Code den Spezifikationen und SLAs entspricht. Nach der Code-Bereitstellung müssen die Teams verstehen können, was in ihren Anwendungen geschieht, wenn die einzelnen Releases die Bereitstellungspipeline durchlaufen. Wenn Sie nicht wissen, was im Inneren Ihrer Anwendung geschieht, können Sie auch Ihre Pipeline nicht verstehen oder wie Pipeline-Events mit der App-Performance und der User Experience zusammenhängen. Observability-Erkenntnisse in Echtzeit zur Analyse des gesamten Anwendungsbereitstellungszyklus rückt die folgenden Ziele in Reichweite:

- Konsistente, messbare und nachvollziehbare Benchmarks für Programmfehlerraten über Entwicklungsteams und Code Releases hinweg.
- Besser sichtbare Metriken der Qualitätssicherung; Probleme werden vor der Produktionsfreigabe erkannt.
- Schnellerer Durchblick bei der weiter wachsenden Menge verteilter Microservices; Probleme können im Moment ihres Auftretens lokalisiert werden.
- Optimierung der Cloud-Ausgaben.
- Schnellere, präzisere Alerts, gezieltes Troubleshooting und rasche Erkenntnisse dank In-stream-KI.
- Engere Einbindung der Security in die DevOps-Prozesse (das geht in Richtung DevSecOps).

Der Code wird dann sicherer und besser kompatibel, weil die Entwicklerteams Sicherheitsrisiken bereits zu einem frühen Zeitpunkt im Build-Zyklus erkennen und beseitigen. Datengestützte Echtzeiterkenntnisse aus den Build Pipelines der Anwendungen machen es möglich, dass die Sicherheitsteams gefährliches Verhalten erkennen und an das zuständige Team melden können, bevor das Problem in die Codebasis eingecheckt wird. Automatisierte Prozesse zur Klärung solcher Beobachtungen machen Security zu einem wesentlichen Bestandteil des Entwicklungsprozesses. In der modernen Anwendungsentwicklung sind sie unerlässlich.

Bessere Geschäftsergebnisse erzielen

Die Vorteile von datengestützten DevOps und Observability wirken sich unmittelbar auf das Geschäftsergebnis aus: Durch die Effizienz- und Produktivitätssteigerung erreicht das Unternehmen eine schnellere Anwendungsbereitstellung, geringere Kosten, größere Kundenzufriedenheit und einen höheren Umsatz.

Unternehmen, die eine umfassende Observability-Plattform verwenden und sie auch jenseits des klassischen Einsatzgebiets IT Operations nutzen, profitieren von den folgenden Vorteilen:

- Echtzeiteinblick in Nutzung, Leistung, Fehler und Sicherheitsvorfälle bei neuen Anwendungsreleases – ohne Verzögerung von Stunden oder gar Tagen wie beim Ad-hoc-Reporting.
- Drastisch verringerte MTTR, mit schnellerer Problemlösung in Produktion (bis zu 70 %) und Vorproduktion (bis zu 40 %)
- Gesteigerte Effizienz durch automatisierte Datenerfassung und -analyse. Dev- und Ops-Teams können sich auf die Unternehmensziele konzentrieren, statt ihre Zeit mit der Konstruktion und Wartung von Monitoring-Tools zu verschwenden.

Dafür, dass die DevOps-Teams auf die geschäftlichen Ziele ausgerichtet bleiben, sind Continuous-Delivery-Prozesse mit häufigen Releases nötig, die messbar und mit den tatsächlichen Geschäftsergebnissen korreliert sind.

Diese schnellen Release-Prozesse verbessern die Customer Experience, weil der bereitgestellte Code leistungsfähiger und besser brauchbar ist. Besserer Code bedeutet zufriedenere Kunden, und zufriedene Kunden sind treue Kunden. Wer mit den Daten einer Observability-Plattform arbeitet und Metriken wie die Anwendungsnutzung und die Verkaufszahlen analysiert, kann die Kundenzufriedenheit direkt durch Code-Releases beeinflussen. Weitere geschäftsrelevante Observability-Metriken sind:

- Neukundenanmeldungen und Download-Raten,
- Umsatzveränderungen und Ausreißer (nach oben und unten) in Verbindung mit Produktreleases,
- Veränderungen beim Customer Engagement, im Abverkauf oder bei den Warenkorbabbrüchen.

Analytics zur Quantifizierung und Steigerung der DevOps-Vorteile

Erfolg in der digitalen Welt erfordert Agilität. Agilität wird durch effiziente DevOps-Release-Prozesse erreicht. Und mit einer Observability-Plattform holen Sie wirklich alles aus Ihrem DevOps-Programm heraus.

Wer von DevOps profitieren möchte, muss Reaktionsfähigkeit, Zusammenarbeit, Sicherheit und Richtlinien-Compliance fortlaufend optimieren. Denn diese Eigenschaften wirken sich über die schnelle Entwicklung innovativer Produkte unmittelbar auf die Reputation des Unternehmens und die Kundenzufriedenheit aus. Trotzdem kann die Einführung von DevOps viele Herausforderungen mit sich bringen, die oftmals das Ergebnis von komplexen Prozessen und Tool-Wildwuchs sind. Wo ein Tool die Sicherheitsdaten aufnimmt, ein anderes die Systemperformance-Daten und ein drittes die Anwendungsdaten, wird deren Korrelation vermutlich schwierig. Wir müssen es Führungskräften, IT-Entscheidern und Entwicklungsteams leicht machen, die Vorteile von DevOps zu verstehen und von den Vorteilen zu profitieren.

Das Geheimnis des DevOps-Erfolgs heißt quantifizierte Validierung. Die alte Weisheit „Was man nicht messen kann, kann man nicht verbessern“ gilt auch heute noch, und eine konsolidierte Observability-Plattform stellt das ideale System für Messungen und Auswertungen dar.

DevOps bietet beachtliche Vorteile, allerdings nur bei richtiger Umsetzung. DevOps ist kein Sprint, sondern ein Marathon. Wer DevOps-Erfolg will, muss nicht nur die Art und Weise der Software-Entwicklung ändern, sondern auch Release, Betrieb, Monitoring und Troubleshooting. Das ist eine ganze Menge, doch die Vorteile bei Produktivität, Entwickler- und Kundenzufriedenheit, Release-Geschwindigkeit und Zuverlässigkeit sprechen für sich.

Wer heute DevOps-Erfolg anstrebt, muss Observability als entscheidenden Bestandteil der Software-Entwicklung begreifen. Observability macht den Zustand der einzelnen Workloads im Kontext sichtbar und verständlich. Angesichts der heute vorherrschenden komplexen Cloud-nativen Workloads gibt Observability den DevOps-Teams die Chance, die Unsicherheiten und offenen Fragen rund um ihre Anwendungen mit Antworten in Echtzeit zu lösen – und den Stakeholdern des Unternehmens entscheidende Erkenntnisse zu liefern.

Mehr dazu finden Sie im Splunk-E-Book **Observability: Ein Leitfaden für Einsteiger.** Oder Sie testen die **Splunk Observability Cloud** gleich selbst (es ist kostenlos).

Erfahren Sie mehr über Splunk für DevOps oder testen Sie Splunk-Software **kostenfrei.**

splunk>

Vertrieb kontaktieren: www.splunk.com/asksales

www.splunk.de

Splunk, Splunk> und Turn Data Into Doing sind Marken und eingetragene Marken von Splunk Inc. in den Vereinigten Staaten und anderen Ländern.
Alle anderen Markennamen, Produktnamen oder Marken gehören den entsprechenden Inhabern. © 2021 Splunk Inc. Alle Rechte vorbehalten.

21-10758-SPLK-The Road to DevOps Success-A4-107-WP_GER