



DevOps

Schnell, zuverlässig und sicher von der Idee zur Realisierung

Björn Kasteleiner · Alexander Schwartz

Software differenziert

Für Standardsoftware gilt das Versprechen: Einmal eine Software einkaufen und anschließend von regelmäßigen Verbesserungen profitieren. Der Kunde profitiert von regelmäßigen Softwareupdates, neuen Funktionen und Sicherheitsupdates. Die Standardsoftware hat Standardfunktionen und Schnittstellen, für die fertige Integrationen zu anderen Softwarepaketen existieren oder mit denen Daten mit Anwendern derselben Software ausgetauscht werden können. Die Kosten stehen im Vordergrund, die sich auf den eigenen Betrieb und die Wartungskosten und Lizenzkosten des Herstellers aufteilen. Die eigenen IT-Kosten kann der Kunde weiter minimieren, wenn er die Software „as a service“ aus der Cloud bezieht. Meist kann er damit durch flexiblere Lizenzierung Kosten senken.

Kurz nach dem Jahrtausendwechsel – die Dot-com-Blase war geplatzt – fasst dies Nicolas Carr unter „IT doesn't matter“ zusammen. Ein Vorsprung durch Individualentwicklung wird dabei von den Mitbewerbern schnell eingeholt und ist nur mit großen Kosten zu halten. Es wäre das Ende jeder Individualentwicklung gewesen [2].

Was aber, wenn Software nun das Einzige ist, das die eigene Firma vom Wettbewerber differenzieren kann? Wenn jede neue Idee zu einer Verbesserung eine Änderung an der Software nach sich zieht?

In diesem Fall werden nicht die Kosten minimiert, sondern z. B. die Zeitspanne von der Produktidee bis zu ihrer Verprobung in Produktion. Je nachdem, wie dieser Prozess optimiert und automatisiert ist, kann dies mehrmals täglich erfolgen.

DevOps beschreibt eine Organisationsform, die ihren Prozess auf schnelle Lieferung von Softwareänderungen optimiert.

Wer DevOps braucht

Differenzierung über Software betrifft alle Unternehmen, deren Hauptprodukt Software ist: Unternehmen wie MyTaxi, Uber oder AirBnB bauen ihr komplettes Unternehmensmodell auf Software auf. Auch bei Unternehmen wie DHL, Hermes, Deutsche Lufthansa oder Deutsche Bahn sind bequeme Buchung der Services und Informationen zur Pünktlichkeit des Pakets, des Flugzeugs und des Zugs wichtig für den Kunden. Bei vielen Prozessen innerhalb der Unternehmen, die ein Kunde nur indirekt bemerkt, steckt Individualsoftware, bei denen jede Verbesserungsidee eine Softwareänderung nach sich zieht.

Eine Investition in einen auf Geschwindigkeit optimierten Softwareänderungsprozess tätigen aber nur die Unternehmen, die mit anderen Unternehmen in einem starken Wettbewerb stehen. Von ihrer Investition erwarten sie, dass sie durch schnellere Umsetzung der Produktideen am Markt der Konkurrenz Marktanteile abnehmen und entsprechende Margen realisieren können. Gerade bei Produkten mit Netzwerk- und Skaleneffekten (wie z. B. AirBnB und MyTaxi) ist dies möglich.

<https://doi.org/10.1007/s00287-019-01173-2>
© Springer-Verlag Berlin Heidelberg 2019

Björn Kasteleiner · Alexander Schwartz
msg systems ag,
Robert-Bürkle-Straße 1, 85737 Ismaning/München
E-Mail: {bjoern.kasteleiner, alexander.schwartz}@msg.group

Alle „Aktuellen Schlagwörter“ seit 1988 finden Sie unter:
<http://www.is.informatik.uni-wuerzburg.de/as>

Erwartungen an DevOps

Die Erwartung ist, neue Produktideen schneller in Produktion zu bringen als die Konkurrenz. Allerdings führt nicht jede Produktidee automatisch zu mehr Umsatz oder mehr Kunden [5].

Daher liegt der Fokus auf einem verbesserten Lernen der Organisation: Produktideen sind zunächst nichts anderes als Hypothesen. Je schneller die Organisation Hypothesen einzeln oder parallel in Produktion bringt und aussagekräftige Metriken über den Erfolg oder Misserfolg sammeln kann, desto früher können aus diesen Informationen neue Hypothesen gebildet werden, und der Zyklus beginnt erneut. Das Ergebnis ist eine Organisation, die ihre Kunden nach und nach immer besser versteht [1].

Dasselbe gilt auch für externe Impulse durch Wettbewerber oder neue Technologien: Je kürzer der Feedbackzyklus ist, desto schneller kann die Organisation reagieren.

Gibt es in einer Organisation nur wenige Releases pro Jahr, so bleibt den Fachabteilungen nur die Möglichkeit, neue Produktideen komplett implementieren zu lassen, inklusive der Funktionen, die nur vielleicht gebraucht werden, weil eine kurzfristige Nachlieferung nicht möglich ist. Können Releases täglich in Produktion gegeben werden, so kann die Fachabteilung zunächst eine minimale Version der Produktidee in Produktion geben und Metriken sammeln. Ist die Produktidee erfolgreich, so können weitere Teile der Idee umgesetzt werden. Ist sie es nicht, so kann diese Hypothese verworfen werden. Die dadurch eingesparten Ressourcen in der Fachabteilung und in der IT können für eine variierte oder eine ganz andere Produktidee verwendet werden [4, S. 241ff.].

Wofür DevOps steht

DevOps ist zunächst nur eine Kombination der Worte „Development“ und „Operations“. Es steht für das Zusammenrücken von Entwicklung und Betrieb innerhalb einer Organisation: Der Bruch zwischen

beiden Abteilungen wird überwunden, um eine fertig entwickelte Softwareänderung ohne Reibungsverluste und in hoher Qualität in Produktion zu bringen. Allerdings steht DevOps für viel mehr: Es steht für die Veränderung der gesamten Organisation, um den Lernzyklus der Organisation zu beschleunigen. Das schließt auch die Zusammenarbeit mit Testern, IT-Security und Fachabteilungen ein.

Wie DevOps funktioniert

Damit DevOps funktioniert, sind zwei Punkte essenziell: Automatisierte Prozesse von hoher Qualität und minimale Übergaben.

Automatisierung hat dabei viele Facetten: Continuous Integration steht für die vollautomatische Erstellung der zu installierenden Softwareartefakte. Testautomatisierung kümmert sich um Unit-, Integrations- und Ende-zu-Ende-Tests. Continuous Delivery kümmert sich um die automatisierte Installation der Artefakte in den Teststufen und in Produktion. Infrastruktur-as-Code steht für Test- und Produktionsumgebungen, die komplett automatisiert bereitgestellt und aktualisiert werden können. Metriken liefern technische und fachliche Informationen zu der in Produktion laufenden Software: Sie ermöglichen A/B-Tests, um den alten Prozess parallel zum neuen Prozess zu betreiben und so aussagekräftige Vergleiche anstellen zu können.

Automatisierung kann dabei Kopffmonopole verringern und Liegezeiten reduzieren. Allerdings wird sie nur dann die notwendige Akzeptanz finden, wenn sie zuverlässig und schnell ist – sei es bei der Installation der Software als auch bei der Ausführung der Tests.

Übergaben werden mit einer produktzentrierten Organisationsstruktur minimiert: Verschiedene Produktteams verantworten für ihre jeweiligen Teile alle Phasen wie fachliches und technisches Design, Umsetzung, Test, Installation und Betrieb (Abb. 1). Durch räumliche Nähe und personelle Verflechtung berücksichtigen sie Erkenntnisse in einer Phase direkt in anderen Phasen.

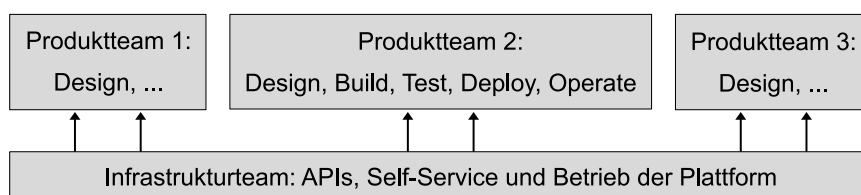


Abb. 1 Produktteams verantworten die Lieferung der Funktionalität und werden dabei vom Infrastrukturteam unterstützt

Unterstützt werden die Produktteams durch ein oder mehrere Infrastrukturteams: Diese stellen Funktionen über APIs oder Self-Service-Plattformen bereit. Die Bereitstellungszeiten z. B. von neuen Servern oder Datenbanken werden dadurch auf Sekunden oder Minuten reduziert. Teile davon können ggf. direkt über Cloudservices eingekauft werden. Dabei kann das Produktteam wählen, ob es standardisierte Infrastrukturkomponenten nutzt oder diese Services selbst bereitstellt, wenn dies z. B. für die Umsetzung einer Produktidee notwendig ist.

Investitionen in DevOps

Wer seine Organisation in Richtung DevOps umbauen möchte, der hat die klassischen Anforderungen des Change Managements: Die Veränderung als notwendig zu begründen („sense of urgency“), das Ziel zu formulieren („clear and shared vision“), die ersten Schritte aufzuzeigen („actionable first step“) und die dafür notwendigen Freiräume zu schaffen („capacity to change“) [6].

Alles in allem ist es eine Investition in Mitarbeiterinnen und Mitarbeiter, Prozesse und Technologie.

Für die Mitarbeiterinnen und Mitarbeiter bedeutet die neue Organisation eine Umstellung ihrer Arbeitsinhalte. In den Produktteams steigt die Breite der Aufgaben stark an: Fachlichkeit (wie funktioniert mein Kunde) steht ebenso im Mittelpunkt wie Methodik verschiedener Bereiche (User Experience, Architektur, Testmethodik, Monitoring) und ein Satz neuer Technologien.

Die querschnittlich aufgestellten Produktteams bringen alle Kompetenzen mit, um für das Produkt neue Ideen in Produktion zu überführen. Über APIs, Self-Service-Funktionalität und Automatisierung sinkt der Kommunikationsbedarf an den früheren Übergabepunkten. Gleichzeitig steigt der Kommunikationsaufwand, um das methodische Wissen innerhalb des Unternehmens weiterzugeben und weiterzuentwickeln: Communities of Practice verbinden Mitarbeiterinnen und Mitarbeiter in verschiedenen Produktteams mit gleichem methodischen Schwerpunkt (z. B. Build-Automatisierung oder User Experience). Der Wissensaustausch erfolgt in regelmäßigen Treffen. Gleichzeitig treiben diese Communities of Practice übergreifende Themen voran. Ansätzen wie das Scaled Agile Framework (SAFe) [8] beschreiben, wie Aktivitäten über Produktteams hinweg koordiniert werden können.

Einige dieser Ansätze wurden bei Spotify ausführlich beschrieben [9]. Die Praxis zeigt jedoch, dass jedes Unternehmen sein eigenes Vorgehen entwickeln muss, da ein von außen vorgegebenes Modell – sei es Spotify oder SAFe – oft unpassend und der falsche Ansatz ist. Das „DevOps Handbook“ [4] zeigt anhand von vielen Beispielen, wie verschiedene Unternehmen jeweils den für sie passenden Weg gefunden haben.

Bei den Prozessen gilt es, Experimente möglich zu machen. Dies bedingt, dass Verantwortungen erweitert und Freiräume geschaffen werden. Die Produktteams müssen ihre Experimente selber planen, durchführen und überwachen können. Formale Abnahmen und Übergaben entfallen zugunsten von automatischen Tests und Deployments. Die Nachvollziehbarkeit wird dabei durch die Versionshistorie der Test- und Deployment-Scripte und der Ausführungsprotokolle sichergestellt. Eine große Anzahl von Experimenten bedingt aber auch, dass Fehler gemacht werden. Anstatt auf zusätzliche und möglicherweise sogar manuelle Kontrollen zu setzen, die den Prozess verlangsamen, müssen die Prozesse die Auswirkungen von Fehlern begrenzen. Dies gelingt durch schnelle und frühe Erkennung von Fehlern (z. B. durch Canary Releases [7]) und schnelle Wiederherstellung (Automatisierung der Deployments und Green/Blue-Deployments [3]).

Im Bereich Technologie werden die bestehenden Technologien des Unternehmens darauf überprüft, ob sie die Mitarbeiterinnen und Mitarbeiter und die geforderten Prozesse unterstützen. Unterstützt z. B. eine Datenbank eine automatische Bereitstellung via Self-Service oder API? Ermöglicht das Testwerkzeug einen leichtgewichtigen Ansatz und ist es gut erlernbar für ein querschnittlich besetztes Team? Wie schnell kann ein Artefakt in Produktion automatisch ausgerollt oder eine alte Version wiederhergestellt werden? Wie können die Infrastrukturteams den Produktteams eine stabile Infrastruktur bereitstellen, die gleichzeitig genügend Freiräume für Innovation bietet? Wie können in allen Bereichen die notwendigen Metriken gesammelt und in den Produktteams zur Überwachung des Betriebs und zur Überprüfung ihrer Hypothesen zur Verfügung gestellt werden?

Die Communities of Practice können helfen, neue Technologien für die Organisation zu erarbeiten und z. B. Blueprints bereitstellen. Dies kann Teams beim Start beschleunigen, sofern die Blueprints von guter Qualität und ausreichend

dokumentiert sind. Allerdings benötigt auch das Verstehen und Anpassen von Blueprints in den Teams Zeit.

Risiken einer DevOps-Transformation

Jede Transformation einer Organisation erzeugt Kosten und hat Risiken. Kosten entstehen zum Beispiel durch Schulung und Weiterbildung, aber auch durch Kündigungen von Mitarbeiterinnen und Mitarbeitern (sei es aus Stress oder aus Angst vor Veränderung) und durch Ausfälle in Produktion durch den Einsatz von neuen Technologien.

Kosten entstehen auch durch Wissensverlust: Durch Kündigung von Mitarbeiterinnen und Mitarbeitern und die daraus resultierenden Neueinstellungen, durch Wechsel von Technologien und Werkzeugen, wodurch bestehendes Wissen seinen Wert verliert und durch fehlende Vernetzung und Prozesswissen innerhalb der neuen Organisation, wenn sich Ansprechpartner ändern.

DevOps tritt an, Prozesse zu beschleunigen. In der Transformation können Prozesse aber auch mehr Zeit benötigen: Sei es durch Umorganisation und fehlende Vernetzung und Prozesswissen, fehlende interne und externe Experten und Mitarbeiterinnen und Mitarbeiter, die an Schulungen und Weiterbildungen teilnehmen.

Ein professionelles Change Management sorgt durch schrittweise Umsetzung dafür, dass das Ziel in mehreren regelmäßigen Schritten erreicht wird, ohne dass die Organisation überlastet wird.

Wie DevOps gelingen kann

DevOps hilft Unternehmen, Produktideen schnell in Software umzusetzen und an den Markt zu bringen.

Es ist jedoch keine Änderung, die nur eine einzelne Abteilung betrifft, sondern sie betrifft die ganze Organisation. Grenzen zwischen funktional organisierten Abteilungen werden verschwinden. Da Experten am Markt schwer zu bekommen sind, liegt der Schlüssel darin, die eigenen Mitarbeiterinnen und Mitarbeiter weiterzubilden. Da Automatisierung der Schlüssel für schnelle Lieferung ist, stehen erlernbare und automatisierungsfähige Technologien im Mittelpunkt. Consultants für Change Management und Technologien können helfen, wenn sie als Katalysator für die eigene Organisation dienen und das Lernen unterstützen.

Hat ein Unternehmen DevOps erfolgreich umgesetzt, so wird es davon doppelt profitieren: Beim Vermarkten seiner Produkte bei seinen Kunden und beim Recruiting von engagierten Mitarbeiterinnen und Mitarbeitern in IT- und Fachabteilungen.

Literatur

1. Blank S (2013) Why the lean start-up changes everything. Harvard Business Review May 2013. <https://hbr.org/2013/05/why-the-lean-start-up-changes-everything>, last access: 3.4.2019
2. Carr N (2003) IT doesn't matter. Harvard Business Review, May 2003. <https://hbr.org/2003/05/it-doesnt-matter>, last access: 3.4.2019
3. Fowler M (2010) Blue green deployment. <https://martinfowler.com/bliki/BlueGreenDeployment.html>, last access: 3.4.2019
4. Kim G, Humble J, Debois P, Willis J (2016) The DevOps Handbook. How to create world-class agility, reliability, & security in technology organizations. IT Revolution Press
5. Kohavi R, Thomke S (2017) The surprising power of online experiments. Harvard Business Review September/October 2017. <https://hbr.org/2017/09/the-surprising-power-of-online-experiments>, last access: 3.4.2019
6. Redman T (2008) Data driven: profiting from your most important business asset. Harvard Business Review Press (ch. 3)
7. Santo D (2014) Canary release. <https://martinfowler.com/bliki/CanaryRelease.html>, last access: 3.4.2019
8. Scaled Agile Framework. <https://www.scaledagileframework.com/>, last access: 3.4.2019
9. Scaling Agile @ Spotify with tribes, squads, chapters & guilds. <https://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf>, last access: 3.4.2019