

DIGITALE TRANSFORMATION

Wie meistern CIO diese Mammutaufgabe in ihren Unternehmen?

AUTOMATISIERUNG

Was können Cognitive Computing und Künstliche Intelligenz mittlerweile?

CONTAINERISIERUNG

Wie verändert der Programmierungs- und Paketierungs-Trend die IT?

HandelsblattJournal

Eine Sonderveröffentlichung von EUROFORUM Deutschland

DEZEMBER 2018 | WWW.HANDELSBLATT-JOURNAL.DE

FUTURE IT-MANAGEMENT

Trends & Innovationen

KI als Treiber der Digitalisierung

S.3

euroforum

Medienpartner

Handelsblatt
Substanz entscheidet.

The Need for Speed

Wie Container-Technologien die Entwickler-Produktivität steigern

von Henning Jacobs & Sebastian Scheele

Nicht die Großen fressen die Kleinen, sondern die Schnellen überholen die Langsamen: Passender als dieses Zitat des ehemaligen Vorstandsvorsitzenden der BMW AG Eberhard von Kuenheim lassen sich die Herausforderungen, vor denen viele Unternehmen angesichts der zunehmenden Digitalisierung aller Wirtschaftsbereiche stehen, kaum zusammenfassen. Mit teils großer Verunsicherung schauen traditionelle Unternehmen auf die neue, hungrige Konkurrenz aus Tech-Start-ups. Deren Speerspitze: disruptive Geschäftsmodelle auf Basis kundenorientierter Software und ihre Agilität. Eine leistungsstarke, moderne Softwareentwicklung wird immer stärker zum Schlüsselfaktor für die eigene Wettbewerbsfähigkeit. Dabei kommen der Einführung von DevOps und Continuous Delivery, sowie Container-Technologien wie Docker und Kubernetes eine zentrale Bedeutung zu.

Die alte Welt – Lost in Translation

Traditionell folgt die Softwareentwicklung einer klassischen Unterteilung von Funktionen und einem linearen Prozess mit vielen menschlichen

Schnittstellen: Zu Beginn antizipiert die Business-Seite die Kundenbedürfnisse, legt detaillierte Anforderungen an die Anwendung fest, und gibt diese an die IT weiter. In wochenlanger Einzelarbeit programmieren Entwickler dann an jeweils ihrer Kopie der Applikation, ohne sich möglicher Überschneidungen bewusst zu sein. Danach werden die Änderungen in eine gemeinsame Basis integriert, das sogenannte Mergen. Je länger mit diesem Schritt gewartet wird und je größer das Team, desto zahlreicher die Anzahl an Konflikten im Code, die dabei gelöst werden müssen.

Nachdem die Fehler in tagelanger Kleinarbeit behoben wurden, wird das neue Release an das QA- und das Operations-Team übergeben, die die neue Version in vielen Schritten manuell testen und schließlich in den Betrieb nehmen. Bis es allerdings soweit ist, sind in der Regel viele Besprechungen mit dem Entwicklungsteam und der Business-Seite notwendig, weil Fehler offenbar werden, es Unklarheiten gibt, oder die Anforderungen an die neue Version nachverhandelt werden. Als Folge dieser langwierigen Diskussionen sind Entwickler geneigt, den beschwerlichen Weg nur selten zu gehen und so viele Änderungen wie möglich in einem Release unterzubringen, was

aufgrund der höheren Komplexität nur noch mehr Handarbeit, Testläufe und Fehlerbehebung mit sich bringt. Das Resultat: lange und für alle Seiten nervenaufreibende Release-Zyklen, qualitativ minderwertige Software und im Zweifelsfall neue Features, die die Welt, beziehungsweise der Kunde, nicht braucht.

DevOps – Transformers

Im Gegensatz dazu kommen leistungsstarke IT-Unternehmen auf mehrere tausend Releases am Tag. So nimmt beispielsweise Amazon alle 11,7 Sekunden neuen Code in Betrieb. Aber führende IT-Unternehmen unterscheiden sich nicht nur durch ihre Release-Häufigkeit: sie produzieren auch 60 mal weniger Fehler und erholen sich 168 mal schneller von diesen. Das Geheimnis hinter dieser Leistungsstärke sind neue Prozesse und Methoden rund um die Schlagwörter DevOps und Continuous Delivery. Denn eigentlich steckt hinter diesen Begriffen nicht weniger als ein umfassender Wandel der Firmenkultur, der alle Aspekte der täglichen (Zusammen-)arbeit verändert und deshalb an erster Stelle ein Umdenken aller Beteiligten erfordert.

Der DevOps-Grundgedanke ist, wie im Begriff schon angelegt, der Schulterschluss zwischen

Entwicklern und dem IT-Betrieb. In der Praxis bedeutet das vor allem, dass Prozesse und Entscheidungen automatisiert und menschliche durch Programmierschnittstellen ersetzt werden. Der Entwickler wird zum internen Kunden des Plattform-Teams, das ihm ein Self-Service-Portal zur Verfügung stellt. Darüber kann er sich die notwendige Entwicklungsumgebung – ganz ohne Expertenwissen und Bereitstellungszeit – selbst konfigurieren und managen, und sich voll und ganz auf das Programmieren konzentrieren.

Im Gegenzug trägt er dann die Ende-zu-Ende-Verantwortung für die Weiterentwicklung und den Betrieb seiner Anwendung. Mit DevOps einher geht die Einführung von Continuous Delivery und agilen Entwicklungsmethoden. Continuous Delivery ist die Sammlung von Prozessen, Techniken und Tools, die es ermöglichen, kontinuierlich Änderungen an den Kunden auszuliefern. Grundlage dafür ist der Aufbau einer sogenannten Deployment-Pipeline, also ein fest definierter und weitgehend automatisierter Prozess, den jeder neue Code durchlaufen muss, bevor er in Betrieb geht. Wenn auf diesem Weg keine Fehler entstehen, wird die Änderung nach der oftmals rein formalen Freigabe durch den Product Owner ohne weitere manuelle Eingriffe an den Kunden ausgespielt. Dieser wiederum gibt, auch über sein Nutzungsverhalten neuer Features, kontinuierlich Feedback, so dass das Produkt direkt an seinen Bedürfnissen weiterentwickelt werden kann. Dadurch ermöglichen DevOps- und Continuous Delivery-Konzepte letztendlich nicht nur schnellere Releases, sondern auch bessere Produkte.

Container – harder, better, faster, stronger

Wie aber kommen dabei Container ins Spiel? Container sind, analog zur Schifffahrt, standardisierte Verpackungen für Software. Sie beinhalten die Anwendung samt ihrer Abhängigkeiten und können durch ihre Einheitlichkeit einfach und zuverlässig von einer Umgebung in die andere verschoben werden. Das war mit virtuellen Maschinen zwar bereits in der Vergangenheit möglich, allerdings simulieren diese ein gesamtes Betriebssystem und sind damit wesentlich schwerfälliger und ressourcen-intensiv.

Für DevOps und Continuous Delivery sind Container ein Glücksfall: Wenn eine Anwendung im Container entwickelt, getestet und ausgeliefert wird, verändert sich die Umgebung über den ganzen Prozess nicht und ist dadurch deutlich weniger pflegeintensiv und fehleranfällig. Außerdem kann so ziemlich jede Art von Anwendung in einem Container betrieben werden, egal in welcher Sprache sie geschrieben ist. Wenn Anwendungen dann noch in kleinere funktionale Einzelteile, wie beispielsweise die Bezahl- oder die Suchfunktion, runtergebrochen werden, können diese weitgehend unabhängig voneinander entwickelt und aktualisiert werden, ohne dass für die anderen Bestandteile ein Ausfallrisiko besteht. Durch all diese Eigenschaften erhöhen Container die Flexibilität, Agilität und Schnelligkeit der Softwareentwicklung und setzten sich zunehmend als Standard durch. So bezeichnete das Online-Portal von Forbes 2016 Container als disruptivste Technologie des Jahrzehnts.

Dabei sind Container streng genommen keine neue Technologie, ihre Geschichte reicht zurück bis in die 70er Jahre. Ihr allmählicher Siegeszug startete jedoch erst mit der Veröffentlichung von Docker-Containern 2013 und der Open Source-Orchestrierungs-Lösung Kubernetes rund ein Jahr später. In Deutschland hat

„Wenn eine Anwendung im Container entwickelt, getestet und ausgeliefert wird, verändert sich die Umgebung über den ganzen Prozess nicht und ist dadurch deutlich weniger pflegeintensiv und fehleranfällig.“



Henning Jacobs,
Head of
Developer
Productivity,
Zalando



Sebastian Scheele,
Mitgründer und
CEO, Loodse

Zalando als eines der ersten Unternehmen umfassend auf moderne Container-Technologien gesetzt und damit die schnelle Expansion der Fashion-Plattform überhaupt erst ermöglicht.

Unsere Cloud-Native Reise bei Zalando – Catch Us If You Can

Zwei Jahre nach Gründung stehen bei Zalando 2010 alle Zeichen auf Wachstum: der damals größte Schuhversender Deutschlands erweitert sein Produktportfolio um den Bereich Mode, geht in den Niederlanden und Frankreich an den Markt und zieht mit seinem knapp 200-köpfigen Team in ein neues Büro in den Prenzlauer Berg um. Die Herausforderung an die IT ist so klar wie ambitioniert: die weitere Expansion des Unternehmens sicherzustellen und Zalando nach und nach zur führenden Fashion-Plattform mit vielen integrierten Angeboten wie Same-Day-Delivery oder individuelle Style-Beratung weiterzuentwickeln.

Schon bald stieß das IT-Team mit seinen damals zwei Rechenzentren an die Grenzen der Skalierbarkeit. Außerdem konnten die Teams aufgrund der unflexiblen Infrastruktur und den damit einhergehenden Abhängigkeiten nicht autonom agieren, was die Entwickler-Produktivität erheblich beeinträchtigte. Zusätzlich konnten Anwendungen damals nur in zwei Sprachen, Java oder Python, geschrieben werden. Alles in allem also kein akzeptabler Zustand für ein aufstrebendes Fashion-Unternehmen. Aus diesem Grund entschloss sich Zalando zu einem klaren Schnitt mit dem Ziel eine langfristig wettbewerbsfähige Infrastruktur mit selbstverantwortlichen Teams aufzubauen.

Die Teams begannen, alle Anwendungen für die Cloud von Grund auf neu zu schreiben und die gesamte Infrastruktur mit Docker-Containern auf Amazon Web Services (AWS) aufzusetzen. Gleichzeitig wurden DevOps-Methoden und -Prozesse umgesetzt und damit Ende-zu-Ende-Verantwortung an Entwicklungsteams übergeben. Für das Management der Container setzte Zalando zunächst auf eine eigene Cloud-Lösung, doch mit stetig steigender Anzahl an Entwicklern und Teams wuchs auch der operative Aufwand. Aus diesem Grund kam 2016 schließlich noch die Einführung der Orchestrierungs-Lösung Kubernetes hinzu. Damals noch die Wette auf das – zugegebenermaßen aussichtsreichste – Pferd im Stall, hat sich Kubernetes inzwischen cloud-übergreifend als de-facto Standard für Container-Management etabliert.

Heute schaut das IT-Team von Zalando auf einen langen und stellenweise steinigen Weg zurück. Nichtsdestotrotz, der Aufwand hat sich gelohnt: In den letzten Jahren hat das Unternehmen Tausende von Entwicklern eingestellt, die in Hunderten von Entwicklungsteams unabhängig voneinander Software entwickeln und mehrere tausend Mal pro Woche neu veröffentlichen. Container bieten dabei ein einheitliches Release-Format und ermöglichen zusätzlich eine effiziente Nutzung der Cloud-Ressourcen. Der Umstieg auf Kubernetes sorgt für eine zukunftssichere Plattform mit der sich die Release-Häufigkeit, Produktivität und Entwickler-Zufriedenheit auch zukünftig weiter steigern lässt. Need for Speed at Zalando? Mission accomplished!