Stephan Kaps

(info@kitenco.de)
arbeitet als Software-Architekt, Entwickler und technischer Projektleiter. Weitere Schwerpunkte liegen in der Konzipierung und Optimierung von Software-Entwicklungsprozessen (Continuous Delivery), dem Application Lifecycle Management und DevOps. Zudem hat er langjährige Erfahrung als Java- und Host-Entwickler. Seine Leidenschaft sind neue Technologien und Methoden.

objektspektrum themenspecial: DevOps

DevOps – Mehr als nur Zusammenarbeit von Dev und Ops!

Was ist DevOps? Wie finde ich den passenden Einstieg? Aber ist das auch schon alles? Was kommt nach der ersten Initiative? Dieses Themenspecial will über den Tellerrand hinausschauen. Wie und an welcher Stelle kann die Personalabteilung (HR) bei der Entstehung eines DevOps-Teams beteiligt werden? Wie kann man das Verhalten der Beteiligten beeinflussen? Und vor allem, wie kann das Thema Sicherheit im DevOps-Umfeld gebührend berücksichtigt werden?

Viele Artikel und Konferenzen widmen sich dem Thema DevOps. Grundsätzlich geht es dabei um die engere Zusammenarbeit zwischen Entwicklung und Betrieb. Der Austausch von Wissen und Erfahrungen soll bewirken, dass jeder die Anforderungen, Bedürfnisse und Rahmenbedingungen des anderen kennen und respektieren lernt. Dadurch lernt man sich gegenseitig besser kennen und schätzen und arbeitet motivierter an dem gemeinsamen Erfolg. Es entsteht ein neues „Wirgefühl". Organisatorische Grenzen werden aufgeweicht und ein „Blame Game" vermieden. Daraus erwächst intrinsische Motivation für das Projekt.

Doch wie geht es weiter nach dem Satz „Wir machen jetzt DevOps"? Organisatorische und kulturelle Änderungen stehen bevor. Wer oder was kann dabei helfen, diese Veränderungen mit wenig „Schmerzen" einzuführen und die Widerstände „sanft" zu beseitigen?

Anna Löw von Giant Swarm liefert *sechseinhalb Überlegungen zur HR-seitigen Unterstützung neu gegründeter DevOps-Teams*. Der Hintergrund dieser Überlegungen ist, dass das DevOps-Team dazu neigt, sich nur auf die technischen Herausforderungen zu konzentrieren. Dieser Gefahr soll begegnet werden.

Sabine Bernecker-Bendixen macht in ihrem Artikel darauf aufmerksam, dass DevOps nur der Weg ist, nicht das Ziel. Engagement ist gefragt, um eine Kultur der kontinuierlichen Verbesserungen zu etablieren und um nicht nach einiger Zeit wieder zu „business as usual" zurückzukehren. Doch Kultur ist nicht kaufbar, sondern deren Implementierung benötigt viel Zeit und ist aufwendig.

Dave van Herpen, **Robert den Broeder** und **Alexander van Ewijk** stellen ein Instrument vor, mit dem es möglich ist, Verhaltensänderungen im DevOps-Kontext herbeizuführen. Sie gehen der Frage nach, wie man es anpacken muss, damit eine „gewünschte Kultur" entsteht.

DevOps ist eine der Voraussetzungen für Continuous Delivery (CD). CD ist inzwischen weit verbreitet. Zu Recht, denn neben der Möglichkeit, sehr frühzeitig Feedback zu neuen Entwicklungen zu erhalten, erlaubt CD durch Automatisierung von Build-, Deploy- und Testprozessen schnell, zuverlässig und wiederholbar Software auszuliefern, qualitativ hoch-

wertig, mit niedrigem manuellen Aufwand und geringem Risiko.

Eine der Praktiken bei Continuous Delivery ist „Infrastructure as Code“, also ein Konfigurationsmanagement, um gegebenenfalls komplette Umgebungen in unterschiedlichen Konfigurationen auf- und abbauen zu können, und zwar per Knopfdruck oder komplett automatisiert.

Kai Weingärtner beschreibt in seinem Artikel, wie ein erster Einstieg in DevOps gelingen kann, mithilfe der „Infrastructure as Code“-Praktik. Durch Vereinheitlichung und Automatisierung der Umgebungserstellung und dem Deployment können schnell Erfolge erzielt werden, ohne gleich eine Kulturrevolution auszurufen.

Doch wollen wir unsere Software kontinuierlich ausliefern, müssen wir auch kontinuierlich Sicherheitstests durchführen!

Continuous Security Testing bedeutet, statische und dynamische Analysen bereits während der Entwicklung durchzuführen, um frühzeitig und regelmäßig Sicherheitsmaßnahmen umzusetzen, bevor manuelle Prüfungen wie Penetrationstests zum Einsatz kommen. Um eine Anwendung bereits während der Entwicklung auf das Vorhandensein sicherheitskritischer Schwachstellen hin überprüfen zu können, ist eine Integration in den Entwicklungsprozess und somit eine kontinuierliche und am besten automatisierte Prüfung notwendig.

Durch Initiativen wie Continuous Integration sind Code-Analysen inzwischen weit verbreitet und quasi Standard. Doch werden bei diesen Analysen auch Sicherheitsschwachstellen geprüft? Wenn ja, welche? Es existieren diverse Kataloge mit bekannten Verwundbarkeiten, die man für das Aufstellen eines Secure Coding Guides heranziehen kann. Hinzu kommen diverse Tools, um automatisiert im Entwicklungsprozess Security-Checks durchzuführen [Kap15].

Dieses Themenspecial setzt einen Schwerpunkt auf Security DevOps (SecDevOps oder DevSecOps), also die Erweiterung von DevOps um Sicherheitsaspekte.

Laut **Andreas Falk** muss die Sicherheit der Anwendung unbedingt mit der Deployment-Frequenz Schritt halten. Dafür ist es notwendig, Sicherheitsaspekte im gesamten Entwicklungsprozess zu berücksichtigen, also bereits bei der Erstellung des Backlogs, über die Entwicklung bis hin zum Betrieb.

Dr. Achim D. Brucker geht auf die Anforderungen ein, die Werkzeuge für Sicherheitsprüfungen erfüllen sollen und betont dabei, dass die Werkzeuge vor allem dem Entwickler helfen sollen, sichere Software zu entwickeln und nicht als Hindernis empfunden werden.

Christian Schneider beschreibt Möglichkeiten, wie agile Projekte durch den Einsatz bestimmter Open-Source-Werkzeuge Ihre Sicherheit projektbegleitend erhöhen können. Dabei geht er auf statische und dynamische Analysen ein und stellt ein sogenanntes Maturity-Modell vor, mit dem man seinen eigenen Standpunkt bzgl. einer Security DevOps-Einführung evaluieren kann.

Im Umfeld von DevOps wächst die Erkenntnis, dass neben Entwicklung und Betrieb weitere Bereiche einer Organisation beteiligt werden müssen. Einige davon haben wir in diesem Themenspecial beleuchtet. Andere fehlen noch, wie z. B. das Business (BizDevOps [BDO]). Für die Zukunft wird bereits von DevOps 2.0 gesprochen. Dies beinhaltet die Integration von nicht-technischen Akteuren, wie dem Marketing- oder Sales-Team. ■

Viel Spaß beim Lesen wünscht Ihnen

Stephan Kaps

Literatur

[Kap15] S. Kaps, Sichere Softwareentwicklung – Ein Einstieg in Secure Coding and Continuous Security Testing, ObjektSpektrum 4/2105.

[BDO] BizDevOps, siehe http://www.bizdevops.org