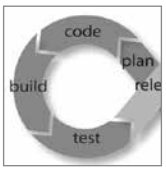
ISACA®
Trust in, and value from, information systems
Switzerland Chapter
Nr. 01 | März 2015 | www.isaca.ch
NEWSLETTER



Entwicklung
Auswirkungen der Begriffe DevOps und Continous Delivery auf die IT-Governance **Seite 61**

Ausbildung IT-Governance
ISACA Switzerland Chapter bietet eine berufsbegleitende Ausbildung in IT Governance an. **Seite 63**

ISACA-Training
Aktuelle Aus- und Weiterbildungsmöglichkeiten für Mitglieder und Nicht-Mitglieder **Seite 63**

Continous Delivery, DevOps und die Auswirkungen auf die IT-Governance

DevOps und Continous Delivery sind Begriffe, die in den letzten Jahren immer mehr in den Vordergrund gerückt sind. Nachfolgend werden die Begriffe kurz erläutert und einige Auswirkungen auf die IT-Governance hervorgehoben.

Von Eyup Koç

Continuous Delivery ist eine Sammlung von Techniken, Verfahren und Werkzeugen, die insbesondere den Softwarelieferprozess verbessern soll. Testautomatisierung, kontinuierliche Integration (Continuous Integration) und kontinuierliche Installation (Continous Deployment) erlauben die agile Entwicklung von Software, die durch automatisierte und damit häufige Releases auf Entwicklungs-, Test-, Integrations- und Produktivumgebungen eingespielt werden kann. Die Automatisierung der Test- und Lieferprozesse ist die Grundlage für eine schnelles, zuverlässiges und wiederholbares Ausrollen des Softwarecodes. Die kontinuierliche Integration beschreibt dabei den Prozess des fortlaufenden Zusammenfügens von Komponenten zu einer Anwendung.

Die Releasezyklen sind also kurz und Innovationen können rasch von der Entwicklung über das Testen in die Produktion überführt werden. Das Deployment der Komponenten verläuft voll automatisiert. Continous Delivery hat sich in der Softwareentwicklung als Bestandteil der agilen Softwareentwicklung breit durchgesetzt und erobert nun immer mehr auch die Infrastruktur (Infrastructure as a Code, bzw. Everything as a Code).

Der Begriff DevOps ist ein Kunstwort und setzt sich zusammen aus «Dev», der die Softwareentwickler (Developers) repräsentiert, und «Ops», der für den IT-Betrieb (Operations) steht. Die Kombination zum gemeinsamen «DevOps» symbolisiert hiermit einen Schulterschluss zwischen Entwicklung und IT-Betrieb. Ein Zusammenrücken der beiden in der traditionellen Wahrnehmung grundverschiedenen Bereiche Softwareent-

wicklung und IT-Betrieb steht im Vordergrund. (Quelle: Java Magazin, die DevOps Bewegung).

Während bei Continous Delivery mehr die Verfahrenskette und die Werkzeuge von der Entwicklung bis zum Deployment in die Produktion im Vordergrund stehen, ist DevOps eher eine kulturelle Angelegenheit. DevOps schreibt keine Prozesse oder Tools vor. Die Zusammenarbeit zwischen Entwicklung und Betrieb steht im Vordergrund. Ein Softwareentwickler ist damit auch für den Betrieb mitverantwortlich, ein Betreiber ist auch in die Entwicklung involviert. Diese Zusammenarbeit dient einem gemeinsamen Verständnis für beide Seiten und insbesondere für das zu entwickelnde Produkt.

DevOps basiert gemäss Gartner u.a. auf den Prinzipien «Gemeinsame Verantwortung für den Erfolg», «eine Fehlerkultur ist die Grundlage für eine ständige Verbesserung» und «DevOps ist eine Kultur, kein Tool, kein Jobtitel».

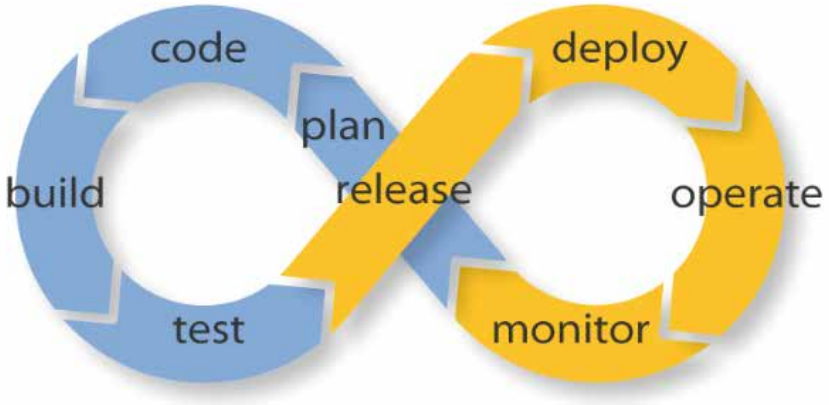
Für die Entwickler bedeutet dies insbesondere, dass sie nicht nur für Bereitstellung des Codes bis zum Testen verantwortlich sind, sie sind verantwortlich bis und mit erfolgreichem Deployment. Der Betrieb muss wiederum in der Entwicklung beteiligt sein, dies benötigt zusätzliche Fähigkeiten wie Code lesen und verstehen.

Einige der IT-Governance Grundsätze (Entwickler entwickeln, Betreiber betreiben) kommen ins Wanken, denn Entwickler sind für den Code End-to-End verantwortlich, Betreiber werden in die Entwicklung einbezogen. Die harte Trennung, auf die zum Beispiel im Rahmen von ITIL Einführungen jahrelang hingearbeitet wurde, wird nun quasi obsolet.

Aus einer Risikobetrachtung heraus hat jedoch Continous Delivery enorme Vorteile und bietet Lösungen für viele aktuelle Governance-Probleme in der IT:

➤ Viele kleine Änderungen sind im Grundsatz weniger riskant als wenige grosse Änderungen. Die Unsicherheiten (und manuelle Eingriffe auf der Produktionsplattform) nach einem Releasewochenende werden mit immer höherem Reifegrad massiv kleiner.

➤ Dadurch, dass alles Code-basiert ist, können Änderungen jederzeit rückgängig gemacht werden (Rollback). Es gibt Re-Deployment des vorherigen Releases. Dies war in der Praxis im klassischen Ansatz nicht mehr möglich, es gab zumindest in heterogenen IT Landschaften nur eine Vorwärtsstrategie.



➤ Es werden keine manuellen Eingriffe in der Produktion vorgenommen. Die Incidentrate wird garantiert sinken.

➤ Anwendungen werden nicht mehr monolithisch gebaut, es werden vermehrt Codefragmente wiederverwendet. Dies führt zu einer höheren Qualität der Anwendungen.

➤ Codefragmente und Anwendungen werden automatisiert getestet. Damit kann sichergestellt werden, dass auf der Test- und Integrationsumgebung durchgehend getestet wird. Die in die Produktion deployten Anwendungen haben eine höhere Qualität bezüglich Verfügbarkeit und Integrität.

➤ Ein wesentlicher Vorteil ist die Agilität. Das IT Business Alignement funktioniert viel besser. Die Time-to-Market sinkt relevant und verringert zumindest die Marktrisiken stark.

Continous Delivery und DevOps sind relativ neu. Es haben sich noch nicht viele Unternehmungen damit auseinandergesetzt. Interessant aus einer Governance-Sicht ist der Aspekt, dass die ersten Versuche in diese Richtung praktisch in jeder Unternehmung bottom up erfolgen und sich schleichend verbreiten. Dies ist ein Gegensatz zu ITIL, welches in der Regel immer top down gesteuert wurde.

Ein Nachteil aus Sicht IT-Governance ist, dass zur Zeit noch keine allgemein formulierten Standards oder gar Frameworks bestehen. Es sind erste Reifegradmodelle und Highlevel Prozessbeschreibungen vorhanden. Auch die Literatur beschränkt sich in der Regel auf Blogeinträge, wissenschaftliche Journals, Whitepaper von Toolproduzenten und einige wenige Bücher. Daher ist es schwierig voraus zu sehen, wie sich bestehende Standards wie ITIL anpassen werden müssen. Auch Continous Delivery wird beispielsweise Change Management und Incident Management Prozesse benötigen, jedoch kaum in der bestehenden Form nach ITIL. Zusätzlich ist zu beachten, dass nicht alles sofort nach dem Continous Delivery Ansatz erstellt werden kann, dies bedeutet, dass zumindest in den nächsten Jahren eine Koexistenz zwischen den aktuellen bekannten und neuen Frameworks vorhanden sein wird.

DER AUTOR

Eyup Koç ist Wirtschaftsinformatiker und Betriebsökonom. Er ist Leiter Cloud Enablement und Consulting bei Swisscom. Eyup hat unter anderem in früheren Positionen als Leiter IT Risk und Compliance Management eines Finanzinstituts und als SOX Projektleiter gearbeitet. Eyup hat die CISA Prüfung erfolgreich abgelegt, ist ISO 20000 und ISO 27001 Auditor. Nebenberuflich ist Eyup Dozent an verschiedenen Fachhochschulen in Projektmanagement, IT Governance und Cloud Computing tätig.

Informationen des Verbands

Haben Sie Ihre IT im Griff? Ein CGEIT muss her!

Wie der beiliegende Artikel von Eyup Koç aufzeigt, ist das Thema «IT-Governance» extrem breit gefächert. Frei übersetzt bedeutet IT-Governance die «IT im Griff haben». Da beliebig viele Themen in der IT schief gehen können, müssten auch beliebig viele Themen in der IT gesteuert werden. Auf den ersten Blick drängt sich die Verwendung möglichst vieler entsprechender Methoden und Standards auf: von ITIL über Prince oder ISO27001 bis hin zu COSO – oder allenfalls die Verwendung von COBIT, das eine Synthese der wichtigsten Standards darstellt.

Wichtiger als ein Standard ist aber, auf den verschiedenen Ebenen der Unternehmenshierarchie über Mitarbeitende zu verfügen, die ein gesundes Mass an theoretischem Wissen und praktischer Erfahrung aufweisen. Was alles dazu gehört, ist im CGEIT-Berufsbild von ISACA sehr gut beschrieben. Das 2007 herausgegebene und 2013 erstmals aktualisierte Berufsbild ist eine ausgezeichnete Basis für die Festlegung der für IT-Governance notwendigen Fähigkeiten.

Das ISACA Switzerland Chapter bietet in der Schweiz bereits seit 2009 eine berufsbegleitende Aus- und Weiterbildung für IT-Governance an und bereitet damit die Interessenten auch auf das ISACA-Examen vor. Der Kurs besteht aus einem strukturierten Selbsstudium, einem Block mit Präsenzunterricht jeweils im Juni/Juli sowie einem kompakten Prüfungstraining im Oktober.

Neben dem im Unterricht eingesetzten Governance-Framework COBIT (in der Version 5) erhalten die Teilnehmenden verschiedene Fachbücher sowie weit über 2000 Seiten mit Unterlagen zur persönlichen Vorbereitung und als Nachschlagewerk.

WEITERE INFORMATIONEN

Informieren Sie sich auf unserer Website www.isaca.ch

ISACA-TRAINING

Datum	Code	Hauptthema – Kurstitel
Ab 15.06.2015 Selbststudium ab sofort	CISA-VK	CISA Vertiefungskurs 2015 (14 Tage)
Ab 15.06.2015 Selbststudium ab sofort	CISM-VK	CISM Vertiefungskurs 2015 (13 Tage)
Ab 15.06.2015 Selbststudium ab sofort	CRISC-VK	CRISC Vertiefungskurs 2015 (12 Tage)
Ab 15.06.2015 Selbststudium ab sofort	CGEIT-VK	CGEIT Vertiefungskurs 2015 (12 Tage)
www.isaca.ch		
30.03.-01.04.2015		COBIT 5 Assessor: Public course
30.03.-01.04.2015/29.05.2015		CGEIT – exam preparation course / Exam: 13.06.2015
www.glenfis.ch		

IMPRESSUM ISACA NEWS

ISACA

Herausgeber, Redaktion: ISACA Switzerland Chapter
Adresse: Sekretariat ISACA c/o BDO AG, Biberiststrasse 16, 4501 Solothurn
Erscheinungsweise: 4x jährlich in Swiss IT Magazine
Mitgliedschaft: Wir begrüssen alle, die Interesse an Audit, Governance und Sicherheit von Informationssystemen haben. Es ist nicht notwendig, dass Sie Sicherheitsspezialist oder Revisor sind, um bei uns Mitglied zu werden. Weitere Informationen finden Sie unter www.isaca.ch

Copyright: © Switzerland Chapter der ISACA