



□ Sabine Bernecker-Bendixen

(sabine@getnext-it.com)

arbeitet als Senior Consultant für getNext IT. Sie ist als internationale Sprecherin, Autorin und Trainerin zu den Themen Culture, Soft Skills, Leadership und Kommunikation tätig. Sie berät und unterstützt IT-Unternehmen, Teams und Führungskräfte auf dem menschlichen Weg zu mehr Erfolg.

DevOps – Engagement ist gefragt!

DevOps ist als „Buzzword“ in aller Munde. Überall dort, wo Hypes identifiziert werden, werden Tools und Zertifizierungen als Lösung aller Probleme und Herausforderungen hoch gehandelt. Doch gerade DevOps-Pioniere wie Patrick Debois, John Willis oder Damon Edwards stellen immer wieder die Bedeutung der Menschen und deren Verhalten für das Gelingen des DevOps-Experimentes heraus. Tools können Prozesse und Abläufe vereinfachen und unterstützen, doch die Menschen in dem System müssen DevOps leben, um das Experiment erfolgreich durchzuführen – nicht nur auf der technischen Seite.

DevOps – die Lösung aller Probleme?

Durch Marktentwicklungen wie Continuous of Everything, Microservices und Cloud sowie die damit verbundene Schnelligkeit sehen sich viele Softwareentwicklungsfirmen vor großen Herausforderungen: „Wie schaffen wir es, eine marktgerechte Geschwindigkeit in unsere Entwicklung zu bringen, ohne Qualitätseinbußen hinzunehmen?“

Neben agilen Methoden wie Scrum kommt an dieser Stelle sehr häufig der Begriff DevOps ins Spiel. Die Grundidee, durch das Zusammenrücken von Entwicklung und IT-Betrieb die Entwicklungszyklen deutlich zu verkürzen, um damit früher und häufiger releasen zu können, hört sich verlockend an. Der Markt bietet eine Vielzahl an Collaboration-Tools, die scheinbar auf Tastendruck das effektive Zusammenarbeiten der Menschen sicherstellen und bestehende Silos auflösen. Es scheint so simpel!

Doch die tägliche Praxis wird zeigen, dass „Komm, ab morgen machen wir DevOps“ nicht die Wunderwaffe zur Pulverisierung aller Probleme darstellt und schon gar nicht sofort zum Erfolg führen wird.

DevOps bedarf einer intensiven Vorbereitung – der gesamten Organisation *und* der Menschen.

Weiterhin muss DevOps auf den Bedarf des jeweiligen Unternehmens angepasst werden, da es nicht den einen Weg – keinen „one-size-fits-all-Ansatz“ – gibt. Die Tatsache, dass DevOps anpassbar ist, stellt eine große Chance für Unternehmen unterschiedlichster Strukturen und Größen dar. Es bedeutet gleichzeitig aber auch, dass es keine Checkliste gibt, die es während der Umsetzung abzhaken gilt.

Der Artikel „What Team Structure is Right for DevOps to Flourish“ von Matthew Skelton zeigt sehr deutlich, dass verschiedene Ansätze denkbar sind [Ske13]. Die Organisationen müssen anfangen, sich mit ihrer Zukunft des „how to work“ intensiv zu beschäftigen. Sie müssen aber auch ihre Vergangenheit „aufräumen“ und sich von alten, nicht mehr nützlichen Mustern trennen. Auch dies bezieht sich sowohl auf die organisatorische, die technische als auch die menschliche Seite.

Der DevOps-Weg ist geprägt von:

- der Auseinandersetzung mit den eigenen Stärken und Schwächen des Un-

ternehmens, der Teams und der Menschen,

- Veränderungen in der Organisation, den Abläufen und im Verhalten aller Beteiligten,
- der Auseinandersetzung mit den Zielen und Werten im Unternehmen und der einzelner Stakeholder,
- der Identifikation von Flaschenhälsen und der damit verbundenen Aufarbeitung,
- Erfolgen und Misserfolgen, Fortschritten und Rückschlägen, Motivation und Frust,
- der steten Optimierung und Schaffung einer lernenden Organisation,
- der kreativen Gestaltung des eigenen DevOps-Fittings.

Wer glaubt, dass die Herausforderungen und Probleme vorbei sind, wenn „erstmal DevOps gemacht wird“, der irrt gewaltig, denn: *DevOps ist der Weg, nicht das Ziel!*

Ganz im Sinne des „Continuous of Everything“ ist ein Hauptaspekt bei DevOps die ständige, kontinuierliche Verbesserung. Lernen wird damit nicht nur eine Aufgabe für die einzelnen Individuen, sondern vielmehr für die Teams und Unter-

nehmen in ihrer Gesamtheit – die lernende Organisation.

Allerdings sind zum Erreichen von Zielen wie ständige, kontinuierliche Verbesserung und lernende Organisation zunächst einmal grundlegende Basiswerte notwendig.

„Organization learning requires safety to make mistakes and deal in differences“, schreibt Daniel Mezick in seinem Buch „The Culture Game“ [Mez12]. Wer also DevOps ernsthaft betreiben möchte, um erfolgreich(er) am Markt bestehen und den neuen Erwartungen gerecht werden zu können, muss zunächst dafür sorgen, Mitarbeitern Sicherheit zu vermitteln. Nur dann können Fehler genutzt werden, um sich stetig weiter zu verbessern; nur dann können „Diversity & Differences“ als Bereicherung willkommen geheißen werden; nur dann kann Wissen von einzelnen Personen geteilt und somit zum Teamwissen werden.

Fehlt das Gefühl der Sicherheit und wird es ersetzt durch das Gefühl der Unsicherheit oder gar der Angst, sorgt der älteste und primitivste Teil unseres menschlichen Gehirns – das „Reptiliengehirn“ – und das limbische System mit der darin eingebetteten Amygdala dafür, dass der Mensch in den Abwehr- und Verteidigungsmodus umschaltet. Ergebnis ist, dass der Fokus nun automatisch auf das gerichtet wird, was verunsichert oder verängstigt. Alle anderen positiven Faktoren werden ausgeblendet und haben keine Bedeutung mehr.

Ein Arbeitsumfeld, in dem Mitarbeiter sich fragen müssen, „was passiert mit mir, wenn ich einen Fehler mache und ihn auch noch eingestehe“, baut nicht nur Druck auf, sondern auch eine Angst und Unsicherheit, die dem Ziel der ständigen Verbesserung entgegensteht. Die Bereitschaft, sich neuer, unbekannter Aufgaben und Themen anzunehmen und zu experimentieren, sinkt gegen Null. Zusätzlich neigen Menschen in derart repressiven Umgebungen dazu, Wissen für sich zu behalten, um sich als Mitarbeiter „unersetzbar“ zu machen und sich so einen persönlichen Vorteil zu verschaffen. Damit rücken der erste und letzte Buchstabe von CAMS (Culture, Automation, Measurement and Sharing) ins Blickfeld – *Culture und Sharing*.

Wie im Akronym CAMS stellen Kultur und die Idee des Teilens von Wissen auch die Klammer um DevOps dar, die das System zusammenhält und überhaupt erst zum Laufen bringt. „Ohne Kultur ist Dev-

Ops verschwendete Zeit“, könnte die freie Zusammenfassung eines Meetup-Vortrages von John Willis sein [Ske/Wil13].

Can't buy you culture!

Kultur beschreibt die Art, wie Menschen innerhalb eines Teams und einer Organisation miteinander umgehen und welche Werte gelebt werden. Dies kann auch erweitert werden durch den Umgang mit externen Stakeholdern wie zum Beispiel Kunden oder Anwendern. Für die Kultur spielt die Verständigung auf gemeinsame Ziele eine zentrale Rolle, denn die Kultur schafft den Rahmen, um diese gemeinsamen Ziele zu erreichen. Dennoch fällt es vielen Unternehmen sehr schwer, sich mit dem fundamentalen Element Kultur nachhaltig auseinanderzusetzen.

„For three or four years I have been going around saying “Culture, culture, it's got to be culture”, and people look at you with stare-y eyes going “Shut the hell up” or “Yeah, we get it John. Our culture, absolutely, yeah. NOW CAN WE GET THAT CHEF RECIPE RUN?”. People think [culture] is a ‘kittens and puppy dogs’ thing which no-one really wants to deal with because it's HARD“ [Wil13].

Doch woran liegt es, dass die Kultur oftmals – bewusst oder unbewusst – vernachlässigt bis vergessen wird? Eine wahrscheinliche Antwort hierauf: *Kultur kann nicht gekauft werden!*

Jedes Unternehmen muss sich individuell mit der Frage nach der eigenen Kultur auseinandersetzen. „Was können wir? Was wollen wir? Wie können wir unsere Ziele erreichen? Was passt zu uns?“, sind nur einige Fragen, mit denen sich die Unternehmen beschäftigen müssen. Hierzu ist eine Analyse des Status quo auf allen Ebenen zwingend erforderlich. Alles in Frage zu stellen, bedeutet nicht, dass alles Alte und Etablierte automatisch schlecht und falsch ist. Es ist vielmehr die Grundlage dazu, Stärken zu benennen, Flaschenhälse zu identifizieren, Verbesserungspotenziale aufzudecken und Möglichkeiten zu erkennen.

Hierbei werden sowohl Strukturen und Arbeitsprozesse, besonders aber auch die Verhaltensweisen und Rollen der einzelnen Beteiligten oder Gruppen genauestens betrachtet. Dieser Prozess ist von (Selbst-)Reflexion geprägt und es entsteht neben dem manifestierten Selbstbild das Bild des Status quo. Je mehr das Selbstbild vom Bild des Status quo abweicht, umso schmerzhafter kann sich diese Phase gestalten.

Haben sich dennoch alle Beteiligten auf Rahmen und Werte für die Kultur geeinigt und gegebenenfalls sogar euphorisch Leitsätze daraus entwickelt und verschriftlicht, zeigt sich häufig die Kluft zwischen Theorie und Praxis: *Die Implementierung von Kultur braucht viel Zeit und ist aufwendig!*

Eine Kultur muss wachsen und sich entwickeln, sie ist nicht plötzlich von heute auf morgen da. Sie entwickelt sich an Zielen und Leitbildern und muss in den Köpfen und Herzen der Stakeholder Einzug halten, dort zur Überzeugung werden.

Eine Veränderung der Kultur geht regelmäßig mit einer Veränderung von alten Mustern einher. Ein wertschätzendes Feedback wird zu einem der wichtigsten Instrumente in der Zusammenarbeit, Kommunikation ein gefragter Skill abseits der berufstypischen Fähigkeiten von „Devs“ und „Ops“. Transparenz und Offenheit, eine positive Fehlerkultur sowie Respekt und Empathie müssen konsequent trainiert und eingefordert werden. Rückschritte und Fehlschläge gehören ebenso dazu wie kleine und große Erfolge.

Kanban zum Beispiel stellt ein hilfreiches Instrument in diesem Prozess zur Visualisierung von Arbeit und dem Auftreten von Flaschenhälsen dar. Gerade in Zeiten der Veränderung kann Kanban dazu beitragen, Überlastungen des Teams und einzelner Mitglieder zu minimieren. Es bietet zusätzlich die Möglichkeit, den Informationsaustausch im Team zu fördern und dadurch eine stete Verbesserung zu erzielen.

Durch bewusst gesetzte Slack-Zeiten und einem Regelrahmen, wie Slack genutzt werden soll, bietet es zusätzlich die Möglichkeit:

- über den eigenen Tellerrand zu schauen,
- Wissen zu teilen und zu erweitern,
- unterstützend tätig zu werden oder
- ganz einfach etwas Neues zu probieren.

Slack kann somit das Team stärken und Entwicklungen fördern. Doch auch Kanban muss gut überlegt sein und mit dem Team trainiert werden. Hier kommt *Improvement Kata* [Hum15] ins Spiel. Kata ist vielen aus dem Bereich der asiatischen Kampfkünste bekannt und bezeichnet eine stete Wiederholung von bestimmten Bewegungsmustern, um diese zur Routine

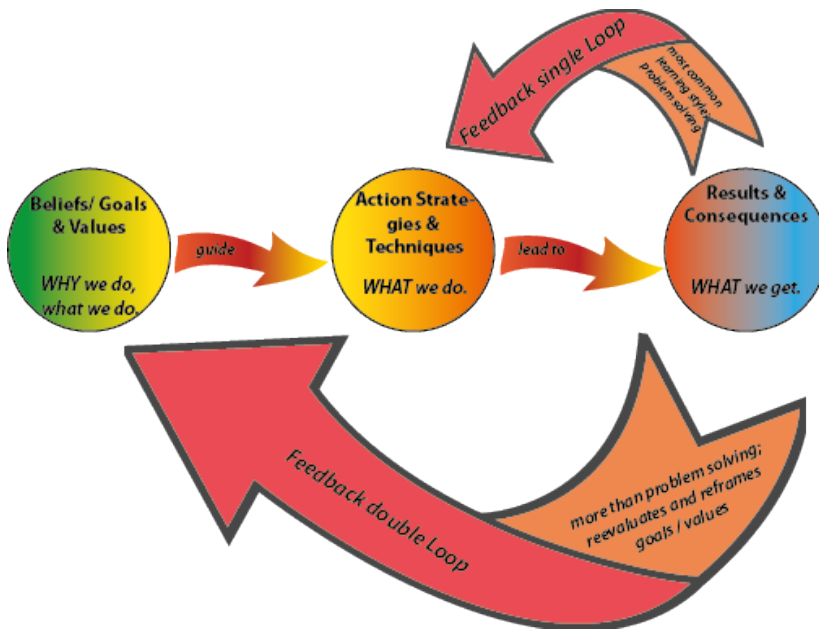


Abb.: „Single-Loop and double-loop learning“ (nach Chris Argyris) im Improvement Kata

werden zu lassen.

Im kulturellen Zusammenhang beschreibt Improvement Kata das immer wiederkehrende Trainieren von Denk- und Handlungsmustern, um die Werte der Kultur lebendig werden zu lassen. Das Improvement Kata zielt auf die folgenden drei Fragen ab, die sich jeder Beteiligte stets stellen soll:

1. Warum wir etwas tun – Glaube / Überzeugung / Werte / Ziele
2. Was wir tun – Aktion
3. Was wir davon haben – Resultate / Ergebnisse.

Dadurch befähigt das Improvement Kata die Beteiligten, sich selber, die Ziele und das Handeln zu hinterfragen und weiter zu verbessern. Feedback kann dann Auswirkungen sowohl auf den Bereich der Aktion haben (lösungsorientiert) als auch auf die Frage des „Warums“. Im Kontext von DevOps und Kanban können mithilfe von Improvement Kata beispielsweise folgende positive Routinen trainiert werden:

- das Verfolgen der gemeinsamen Ziele,
- hilfreiches und wertschätzendes Feedback geben,
- Feedback annehmen,
- Fehler offen und rechtzeitig ansprechen,
- Übernahme von Verantwortung,
- Qualitätsbewusstsein,
- Kaizen-Mentalität,

- Leadership-Mentalität,
- Meeting-Verhalten,
- die Annahme der Veränderung als Norm,
- „Notfall“-Kit.

Auch ein Handeln nach dem PDCA-Zyklus – auch Demingkreis genannt – kann helfen, die Einführung von neuen Routinen zielführend und erfolgreich umzusetzen. Plan-Do-Check-Act beschreibt das Vorgehen im Rahmen der kontinuierlichen Verbesserung. Als erster Schritt erfolgt die *Planung* unter Einbindung der am Prozess beteiligten Stakeholder-Parteien.

Es folgt die Phase des *Do*, in der die Ergebnisse des Plans ausprobiert werden, gefolgt vom *Check*. Hier wird die Wirksamkeit und Umsetzbarkeit sorgfältig geprüft und gegebenenfalls angepasst, bevor der Plan zur allgemeinen Umsetzung – *Act* – freigegeben wird. Durch das Einbeziehen der am Prozess beteiligten Parteien wird zunächst einmal eine breitere Akzeptanz geschaffen. Diese wird noch verstärkt durch die ausdrückliche Testphase und deren Analyse.

Die Analyse schafft zusätzlich Fakten, die letzte Vorbehalte beseitigen können. Die Mitarbeiterzufriedenheit wird verbessert. Der Grundsatz des PDCA-Zyklus findet sich auch in der DevOps-Strategie wieder. Nach der Planung mit den Stakeholdern erfolgt die Entwicklung der Software, die sehr zeitnah getestet wird. Die Ergebnisse der Tests fließen in die Entwicklung mit ein, bis eine Qualität er-

reicht ist, die ein Release zulässt.

Fail-Fast – das Mantra von DevOps

Fail-Fast, Fehler frühzeitig zu erkennen, um sie zu beheben, bevor sie einen größeren Schaden anrichten, ist das Mantra im „DevOps-Movement“. Doch von Kindheit an haben wir gelernt, dass es für uns meist negative Konsequenzen wie Strafen oder schlechte Noten nach sich zieht, wenn wir Fehler machen und diese auch noch zugeben oder gar offen ansprechen. Hinzu kommt, dass Fehler im gesellschaftlichen Kontext zunächst einmal als grundsätzlich schlecht angesehen und verurteilt werden.

Daher haben viele Menschen sich Strategien angeeignet, um Fehler zu kaschieren, in der Hoffnung, dass diese nicht bemerkt werden oder ihnen später nicht „in persona“ zugeordnet werden können. Doch Fehler passieren und sind Teil der Entwicklung: *Nicht die Fehler sind das Negative, sondern vielmehr der falsche Umgang mit ihnen.*

Wichtiger als die Frage, wer den Fehler gemacht hat, sind die Fragen, warum der Fehler passiert ist – Analyse – und was getan werden kann, um ihn später zu vermeiden. Weiterhin ist es wichtig, welche Auswirkungen ein Fehler auf ein System hat. Hier liegen unter Umständen enorme Verbesserungspotenziale und innovative Ansätze versteckt.

Wenn also „Fail-Fast“ ein wichtiger Bestandteil von DevOps ist, heißt das dann, dass möglichst viele Fehler gemacht werden sollen? Nein! Es bedeutet, dass über Fehler offen kommuniziert werden muss. Allen Stakeholdern muss klar sein, dass Fehler passieren können und werden, ein anschließendes „Blame Game“ aber nicht zielführend ist.

„Fail-Fast“ ist ebenfalls ein Appell an alle Beteiligten, Verantwortung für Fehler dahingehend zu übernehmen, dass sie diese als Teil des Prozesses akzeptieren und gemeinsam an der Lösung arbeiten. So schön die Idee von „Fail-Fast“ auch ist: noch wichtiger ist es, dass jeder einzelne dazu beiträgt, durch Nachdenken und Mitdenken vermeidbare und offensichtliche Fehler zu minimieren.

Engagement – ohne geht es nicht!

DevOps erfolgreich zu entwickeln, einzuführen und nachhaltig umzusetzen, erfordert die Bereitschaft auf allen Seiten, sich diesen Herausforderungen zu stellen. ITler sind es gewohnt, sich auf neue Techniken, Tools und Sprachen einzustellen. Doch die

Veränderungen durch und mit DevOps gehen weit über den technischen Bereich hinaus und betreffen in einem großen Teil die Persönlichkeit oder das eigene Verhalten.

Soft Skills, die in der klassischen Ausbildung eine untergeordnete oder gar keine Rolle gespielt haben, bekommen plötzlich einen hohen Stellenwert. Alte Muster aufzubrechen und neue anzutrainieren ist extrem anstrengend, zumal jeder Stakeholder neben den notwendigen Veränderungen gleichzeitig noch die täglichen Aufgaben erfüllen muss.

Kein Team, kein Unternehmen wird die Zeit haben, alle notwendigen Veränderungen losgelöst vom Geschäft durchzuführen; vielmehr müssen Veränderungen und Tagesgeschäft Hand in Hand abgewickelt werden. Die Belastung wird dadurch für alle Mitglieder zunächst ansteigen, bis die neuen Routinen greifen. Die Produktivität wird gleichzeitig in der Regel sinken.

Doch mit gestiegener Belastung bei gleichzeitig geringerer Produktivität ist die Wahrscheinlichkeit sehr groß, dass der Frust steigt. Die Gefahr des Motivationsverlustes ist hoch. An genau diesem Punkt entscheidet sich, ob der DevOps-Weg weiter beschritten wird und damit eine Chance auf Erfolg bekommt oder zu „*business as usual*“ zurückgekehrt wird.

Das Engagement, das von jedem Einzelnen hierbei erforderlich ist, geht über das normale Maß hinaus, entscheidet aber über Erfolg oder Misserfolg. Führungskräfte, Manager und Mitarbeiter sind hierbei gleichermaßen gefordert.

An sich selber zu arbeiten, gemeinsam mit dem Team neue Abläufe zu verinnerlichen und dabei gleichzeitig das Ziel der steten Verbesserung zu verfolgen, verlangen Durchhaltevermögen und Ausdauer. Sich verantwortlich zu fühlen für das eigene Handeln, eigene und fremde Fehler und für den Teamerfolg, ist in der Regel mehr als alles, was bislang gelebt wurde, umso mehr, wenn es sich auch noch um verteilte Teams handelt. Gemeinsame Ziele höher zu bewerten als die individuellen, ist für

viele Beteiligten – auf allen Hierarchieebenen – eine neue Erfahrung.

Eine Möglichkeit, den Teams diese Phase zu vereinfachen, sind Workshops und Trainings, um gemeinsame Ziele und das Vorgehen besser abzustimmen sowie Retrospektiven zu den bislang durchgeführten Aktionen abzuhalten. Auch das Coaching und die Begleitung durch außenstehende Personen sind hilfreich, um festgefahrene Situationen zu lösen, Motivation zu schaffen, Überlastungen rechtzeitig zu erkennen und Blickwinkel zu verändern.

Gleichzeitig kann ein Coach Aufgaben wie Reflexion und Moderation von Workshops und Meetings übergangsweise übernehmen und damit Entlastung schaffen. Parallel dazu erhalten alle Teammitglieder die Chance, diese neuen Aufgaben und Rollen zu trainieren und in sie hineinzuwachsen. Die Annahme, dass Teams diesen Weg alleine gehen müssen, ist schlichtweg falsch. Jede Hilfe, die das Team zusätzlich erhalten kann, erhöht die Chance auf Erfolg, verkürzt die Umsetzungszeit, erhöht die Motivation und spart damit Kosten ein.

DevOps – nur im ganzen Unternehmen oder auch als Team?

Eine Frage, die immer wieder auftaucht, ist die, ob ein ganzes Unternehmen Dev-

Ops adaptieren muss oder auch nur ein einzelnes Team derart arbeiten kann. Die Antwort hierauf ist einfach. Zwar kann ein Team alleine nach der DevOps-Idee arbeiten, allerdings sollten grundlegende Werte wie Respekt, Wertschätzung, Transparenz und eine positive Fehlerkultur fest im gesamten Unternehmen verankert sein.

So wie es in einem Getriebe zu Problemen bis hin zu Fehlfunktionen führen kann, wenn ein einzelnes Zahnrad losgelöst aus dem Gesamtkontext optimiert wird, kann auch eine losgelöste Optimierung einer einzelnen Unternehmenseinheit zu Problemen im Gesamtablauf führen. Vielmehr sollten Unternehmen die Chancen nutzen, die DevOps bietet und einen ganzheitlichen Ansatz – wie im Systemdenken propagiert – wählen.

Hierzu wäre in einem ersten Schritt die Erarbeitung und Schaffung einer gemeinsamen, wertschätzenden und auf Verbesserung ausgerichteten Kultur ein enormer Mehrwert für viele Organisationen. Unternehmen, die Gefallen an dieser Entwicklung gefunden haben, begeben sich auf die Reise zu BizDevOps und führen die Idee der Kollaboration auf eine neue Ebene. ■

Twitter: @SabineBendixen

Quellen

[Ske13] <http://web.devopstopologies.com>

[Mez12] Mezick, Daniel: *The Culture Game – Tools for the Agile Manager*; ISBN 978-0-984-8753-0-6

[Ske/Wil13] <https://web.experiencedevops.org/2013/10/07/john-willis-botchagalupe-on-devops-culture-culture-its-got-to-be-culture/>

[Wil13] <https://www.youtube.com/watch?v=kbvMmNrZ8Kk>

[Hum15] Humble, Jez; Molesky, Joanne; O'Reilly, Barry: *Lean Enterprise – How High Performance Organizations Innovate at Scale*; O'Reilly 2015 ISBN: 978-1-449-36842-5