Qualität, Nutzen und Geschwindigkeit von Softwareentwicklungsprojekten mit DevOps erhöhen

Softwareprojekte unchained: Mit DevOps auf die Überholspur!

Ralf Neubauer
Berater bei der msg group

Management Summary

- Der Erfolg eines Unternehmens hängt heute maßgeblich von den Fähigkeiten und der Kundenorientierung der IT ab. DevOps bietet hier die Möglichkeit, schnell den Kundenwünschen entsprechend zu reagieren.
- DevOps setzt auf den agilen Vorgehensweisen auf und vereint eine klassische, funktional getrennte Betrachtung von Entwicklung (Development) und Betrieb (Operations).
- Es kann z.B. helfen, die Time-to-market für Ihre Produkte zu verbessern, die Bedürfnisse der Kunden besser zu erkennen und flexibler auf Marktanforderungen zu reagieren. Das an den "5 Phases of DevOps" orientierte Reifegradmodell unterstützt auf dem Weg zum "High Performance Delivery".
- Die Phasen beinhalten z.B. Continuous Backlog, Continuous Integration und Continuous Delivery. Die Visualisierung in Form einer Journey Map fördert den Transformationsprozess.
- Mit dem Reifegradmodell können Sie ermitteln, auf welcher Stufe Sie sich in Ihrer Entwicklung befinden, sich neue Ziele setzen und diese entsprechend vorantreiben.
- DevOps dient dem Ansatz der kontinuierlichen Verbesserung (Continuous Improvement) und der Optimierung des Flows sowie des am Kundennutzen ausgerichteten Value Streams.

4 Uhr morgens. Der Wecker bereitet sich noch in aller Ruhe auf seinen Einsatz in zwei Stunden vor, die einjährige Tochter schläft zum ersten Mal seit langem durch. Bis das Telefon klingelt.

Noch bevor Sie richtig wach sind, schlägt der Hund an und ihre Tochter steht schreiend in ihrem Bettchen. Aber es kommt noch schlimmer: Totalausfall der weltweiten Bandversorgung in der Produktion ihrer Firma. Keiner weiß, woran es liegt. Als Sie im Büro ankommen, hören Sie in der ersten Stunde ausschließlich gegenseitige Schuldzuweisungen zwischen dem IT-Betrieb und der Entwicklungsabteilung, die in der letzten Woche ein neues Release zur Verfügung gestellt hat. Am nächsten Abend, als die Systeme nach intensiver Analyse und der Arbeit von vielen beteiligten Kollegen wieder

laufen, stellen sie fest, dass sie zur endgültigen Behebung des Problems Anpassungen in die Software einbauen müssen, sonst drohen weitere Ausfälle. Ihr regelmäßiger Releasezyklus für die betroffene Software sieht ein Update in ca. 6 Monaten vor, da erst entsprechend Abstimmungen mit Fachbereichen und umfangreiche Tests durchgeführt werden müssen.

So schlimm wird es schon nicht kommen? Dieses Beispiel wird leider in vielen Unternehmen tagtäglich zur bitteren Realität und diese wirkt sich nicht nur auf die persönliche Situation der Betroffenen aus. Beispiele wie dieses verhindern Wachstum in Unternehmen, sie bremsen Innovationen, sie sorgen für schlechtere Wettbewerbspositionen und fressen ihren Deckungsbeitrag auf.

Als IT-Berater und Systemintegratoren stehen wir regelmäßig vor der Aufgabe, für unsere Kunden solche Situationen, wie eingangs beschrieben, zu analysieren, Verbesserungen zu identifizieren und in den Vorhaben umzusetzen. Eine klassische, an den typischen Phasen eines Softwareprojektes orientierte Betrachtung der einzelnen Bereiche, wie z.B. Anforderungsspezifikation, Entwicklung und Betrieb, ist dabei nicht zielführend. Um Probleme dieser Art zu lösen, braucht es einen holistischen, systemischen Ansatz.

DevOps bietet eine Lösung

Der State of DevOps Report [1] untersucht jährlich die Einflussfaktoren auf die Performance der IT in Unternehmen. Anhand von wenigen Kennzahlen wird klar, wie sich Top-Performer von Low-Performern unterscheiden. Während heute in vielen Firmen noch zwei- oder dreimal im Jahr ein Softwaresystem mit neuen Funktionen in Produktion gestellt wird (Deployment), schaffen das die Top-Performer alle paar Sekunden. Die Kennzahl "Time to restore" beschreibt die Zeit, die notwendig ist, um einen Service nach einem Ausfall wiederherzustellen. Die Gruppe der Top-Performer schafft das durchschnittlich 2604 Mal schneller als die Low-Performer.

IT-Bereiche in Unternehmen sind typischerweise funktional organisiert. Das heißt, es gibt einerseits Organisationseinheiten, die sich mit der Entwicklung und Bereitstellung von Softwaresystemen befassen und z.B. das Projektgeschäft betreuen. Und andererseits gibt es die Einheiten, die Ergebnisse dieser Projekte dann im operativen Betrieb betreuen. Oftmals handelt es sich dabei um sogar rechtlich eigenständige Einheiten, mit eigenen Zielen, eigenen Incentives und getrennten Verantwortlichkeiten. Die Entwicklung ist bestrebt – nicht selten auf massiven Druck der Fachbereiche hin – neue Softwareversionen schnell in die Produktion zu bringen. Der Betrieb andererseits, pocht auf notwendige Stabilität und möchte diese keinesfalls durch eventuell unbedachte Anpassungen in neuen Softwareversionen gefährden. Durch diese gegenläufigen Zielsetzungen entsteht die sogenannte "Wall of Confusion" (Bild 1) zwischen beiden Bereichen. Eine ähnliche Wand gibt es nach wie vor oft zwischen Fachbereichen und der IT-Entwicklung. Die Situation war hier ähnlich, allerdings konnte diese Wand durch den Einsatz agiler Methoden und eine damit erreichte höhere Integration der Fachbereiche in die IT-Projekte schon durchlöchert werden.

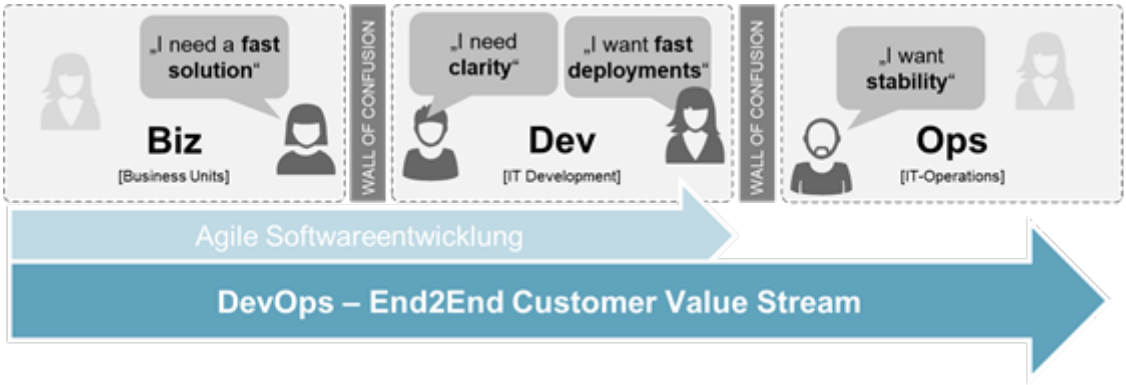

Bild 1: Walls of Confusion

DevOps setzt auf den agilen Vorgehensweisen auf und vereint die klassische funktional getrennte Betrachtung von Entwicklung (Development) und Betrieb (Operations). Als Ansatz zur Prozessverbesserung ist DevOps damit in den letzten Jahren vor allem im Bereich der Softwareentwicklung in der Breite angekommen. Der Ansatz selbst, kann jedoch auch auf weitere Branchen und Produktionsbereiche angewendet werden.

Gelingt diese Zusammenführung von Entwicklung und Betrieb, entstehen Potentiale, besonders für Effizienz und Geschwindigkeit sowie für die inhaltliche Entwicklung und Qualität des betrachteten Produktes. DevOps, richtig umgesetzt, sorgt für eine höhere Geschwindigkeit bei mindestens gleichbleibender, meist aber deutlich gesteigerter Qualität.

DevOps hat seine Wurzeln im Lean Management. Analog dazu betrachten wir den Softwareentwicklungsprozess nicht mehr aus dem Blickwinkel eines Projektes. Ein Projekt ist eine einmalige Sache, mit einem definierten Start, einem definierten Ende und festgelegten Ergebnissen. Wenn wir die DevOps Brille aufsetzen, ist der beste Vergleich eine Produktionslinie. Ich stelle mir dabei immer eine Produktionslinie bei einem Fahrzeughersteller vor. In vielen Einzelschritten wird ein Fahrzeug entwickelt, zusammengebaut und schlussendlich an den Kunden übergeben. Auch für die Fahrzeughersteller ist nach der Übergabe an den Kunden nicht Schluss. Sie betreuen ihre Produkte während der Kunde sie nutzt, warten diese und versuchen aus dem Feedback ihrer Kunden zu lernen. Natürlich ist das Entwickeln einer Software ein immer neuer kreativer Prozess und damit nicht vollständig mit einer industriellen Produktion zu vergleichen. Aber jenseits der kreativen Anteile gibt es viele, sich ständig wiederholende Schritte, die deutlich von einer solchen Sichtweise profitieren können. Aus diesen ergeben sich die Effizienzpotentiale, die wir heben wollen.

Analog dem Lean Management betrachten wir mit DevOps nunmehr den konsequent am Bedarf des Kunden ausgerichteten Value Stream und zwar von der Identifikation und Formulierung der Anforderungen bis zum operativen Betrieb des Produktes (der Software oder der Lösung) beim Anwender. Aufgrund dieser umfassenden End-2-End Betrachtung verzichten wir auch darauf, weitere Ergänzungen des DevOps Begriffs zu verwenden, wie z.B. BizDevOps, was die Integration über die erste "Wall of Confusion" zwischen Fachbereich und IT-Entwicklung in Bild 1 beschreibt.

Das vollständige Potential dieser integrierten Sichtweise kann nur dann realisiert werden, wenn die entsprechende Verantwortung der einzelnen Phasen dabei auch in einer Hand liegt. Die Personen, welche die Entwicklung steuern und die Anforderungen priorisieren, sollten auch diejenigen sein, die nachts alarmiert werden, wenn die Services z.B. aufgrund von Qualitätsproblemen nicht zur Verfügung stehen.

DevOps kann Ihnen helfen, wenn Sie

- Produkte über die Lieferung hinaus bei Ihren Kunden betreuen
- die Bedürfnisse Ihrer Kunden besser verstehen wollen
- die Time-to-market für Ihre Produkte verbessern müssen
- Qualitätsprobleme in Ihren Services lösen wollen
- auf Marktanforderungen flexibel reagieren müssen

DevOps in der Praxis umsetzen

Vor allem für bestehende Vorhaben ist es oft nicht einfach, Ansatzpunkte systematisch zu identifizieren und daraus sinnvolle Maßnahmen und Chancen abzuleiten. DevOps ist ein Weg, den man geht, keine Checkliste, die man Punkt für Punkt abarbeitet. Um unseren Projekten hier trotzdem ein Handwerkszeug zu geben, sich zu orientieren und Potentiale zu erkennen, haben wir ein Reifegradmodell entwickelt. Im Rahmen von wenigen Workshops helfen wir den Vorhaben auf dieser Grundlage den eigenen Status Quo zu ermitteln und eine Roadmap für die weitere Entwicklung zu entwerfen. Die nachfolgenden Schritte beschreiben die Vorgehensweise, danach folgt die Erläuterung der einzelnen Werkzeuge:

Symptomaufnahme / Problembeschreibung (optional)

Bei bereits laufenden Vorhaben gibt es häufig konkrete Anlässe in Form von Problemen, wie sie z.B. am Anfang des Artikels beschrieben wurden, die dazu führen, dass man über Verbesserungen nachdenken muss. In einem ersten Schritt hilft es, für die nachfolgenden Schritte diese Symptome nochmals aufzunehmen, klar zu beschreiben und abzugrenzen. Im weiteren Verlauf kann man dann jeweils auf diese Beobachtungen referenzieren und prüfen, ob sich die erlebten Probleme mit den identifizierten Maßnahmen beheben lassen. Optional ist dieser Schritt deshalb, da wir mit unserer Vorgehensweise auch Verbesserungen identifizieren können, wenn das Projekt eigentlich "gut läuft ". Nur weil z.B. Ineffizienzen nicht augenfällig geworden sind, heißt es nicht, dass sie nicht da sind.

Bewertung des Ist-Reifegrads des Projektes

Unser Reifegradmodell haben wir in den "5 Phases of DevOps" beschrieben. Es stellt einen idealisierten Reifeprozess eines Vorhabens dar, den es durchläuft bis es den Zielzustand erreicht hat. Um zu einer Einschätzung

des aktuellen Reifegrades zu kommen, führen wir eine Bewertung des Projektes in verschiedenen Dimensionen durch, die im Sinne eines holistischen Ansatzes alle relevanten Aspekte der Organisation miteinbeziehen.

Definieren des anvisierten Ziel-Reifegrads

Basierend auf den "5 Phases of DevOps" wird der gewünschte Zielzustand für das Vorhaben definiert. Je nachdem, welches Vorhaben oder Projekt man betrachtet, kann es sein, dass ein Erreichen des höchsten Reifegrads gar nicht erstrebenswert ist. Wird z.B. ein System betrachtet, dessen End-of-Life Datum bereits bekannt ist, muss man Ziele, z.B. unter Einbeziehung der Wirtschaftlichkeit, anders bewerten, als wenn das System gerade erst ausgerollt wird.

Identifizieren und Bewerten von Maßnahmen

Wir haben einen Orientierungsrahmen geschaffen. Diesen verwenden wir, um den beteiligten Mitarbeitern zu erläutern, wie wir DevOps verstehen und was im Sinne von DevOps einen Mehrwert liefert. Dieser Rahmen ermöglicht uns auch eine Bewertung und Priorisierung von identifizierten Maßnahmen, bezogen auf ihren Wertbeitrag.

Bei der Identifikation von möglichen Verbesserungsmaßnahmen orientieren wir uns am Reifegradmodell und den dort beschriebenen Dimensionen.

Abhängig vom Wissensstand der beteiligten Mitarbeiter, wird der Orientierungsrahmen bereits vor Einstieg in die Workshops in einer initialen Schulungsmaßnahme vermittelt.

Umsetzung der Maßnahmen

Nach der Identifikation und Priorisierung der Maßnahmen können diese in den Vorhaben eingeplant und umgesetzt werden. Abhängig von der identifizierten Maßnahme und der betroffenen Dimension, müssen entsprechende Transformationspläne erstellt werden. Ist z.B. bereits eine agile Vorgehensweise im Projekt etabliert, müssen Epics und Stories definiert und zur Umsetzung der Maßnahmen in Sprints eingeplant werden.

Regelmäßige Überprüfung

Der hier beschriebene Durchlauf kann mit unterschiedlichen Schwerpunkten immer wieder angewandt werden, um Nachhaltigkeit und kontinuierliche Verbesserung zu etablieren. Idealerweise wird er zu einer regelmäßigen Routine. Es empfiehlt sich, für die Durchführung immer wieder auch externe Experten hinzuzuziehen, die mit einem Blick von außen bewusst neue Impulse setzen können.

5 Phases of DevOps: Der Weg zum High Performance Delivery

Wie schon erwähnt, verstehen wir DevOps nicht als ein definiertes Ziel, welches man mit Hilfe einer Checkliste erreicht. Wir verstehen DevOps als eine Reise. Auf dieser Reise können sich die Dinge verändern und sich weiterentwickeln. Trotzdem wird es ein paar Meilensteine geben, die stabil bleiben.

Als wir begonnen haben, uns mit dem Thema auseinanderzusetzen, haben wir schnell festgestellt, dass unsere Projekte immer wieder danach fragen "Wo stehen wir denn in Bezug auf DevOps?", gefolgt von der Frage " Wohin können wir uns entwickeln?". Die Antwort auf diese Fragen ist je Vorhaben unterschiedlich und lässt sich nicht pauschal beantworten. Trotzdem hat sich gezeigt, dass es den Projekten hilft, eine Karte zu haben, auf der sie sich orientieren können. Im ersten Schritt haben wir dafür ein Reifegradmodell entwickelt, welches verschiedene Entwicklungsstufen im Kontext zu DevOps abbildet.

Als wir uns letztes Jahr für die Teilnahme an der DevOpsCon entschieden haben, waren wir auf der Suche nach einer geeigneten Visualisierung. Nach einigen Diskussionen ist die Idee entstanden, tatsächlich eine Reise abzubilden, in Form einer Schatzkarte (Bild 2).

Die einzelnen Stationen der Karte bilden die Reifegrade ab, die wir in unseren Vorhaben erreicht haben oder erreichen wollen. Sie erlauben ein Ermitteln des Ist-Zustands und ein Anvisieren des gewünschten nächsten Ziel-Zustands.

Die Reifegrade, die wir gewählt haben, orientieren sich am Fluss einer Anforderung durch den Value Stream bis zum operativen Betrieb. Um in einer Bewertung alle Aspekte einer ganzheitlichen Sicht abdecken zu können, haben wir relevante Betrachtungsdimensionen definiert (siehe Tabelle 1) und diese je Reifegrad mit Kriterien hinterlegt. Anhand dieser Kriterien kann ein Projekt den Erfüllungsgrad für sich selbst bestimmen und so seinen eigenen Reifegrad. Die Betrachtungsdimensionen umfassen dabei Organisation & Kultur, Prozesse, Technologie & Architektur und Anwender-/Kundenorientierung.

Natürlich wäre es möglich, jetzt ein mathematisches Modell dahinter zu legen, das aufgrund der Erfüllung bestimmter Kriterien einen bestimmten Reifegrad (von sagen wir z.B. 2,34) ermittelt. Diese Absicht hatten wir nicht und halten das auch nicht für zielführend. Das Reifegradmodell soll Anhaltspunkte liefern und eine Grundlage dafür sein, in einem Team einen Denk- und Diskussionsprozess in Gang zu setzen. Weiterhin ist es auch möglich, in einer Betrachtungsdimension schon sehr fortgeschritten zu sein, während in einer anderen der Reifegrad noch vergleichsweise gering ist. Die Darstellung einer so durchgeführten Bewertung in Form eines Spinnendiagramms, hat sich dabei als hilfreich erwiesen (siehe Bild 3: Bewertung Reifegrad und Potential).

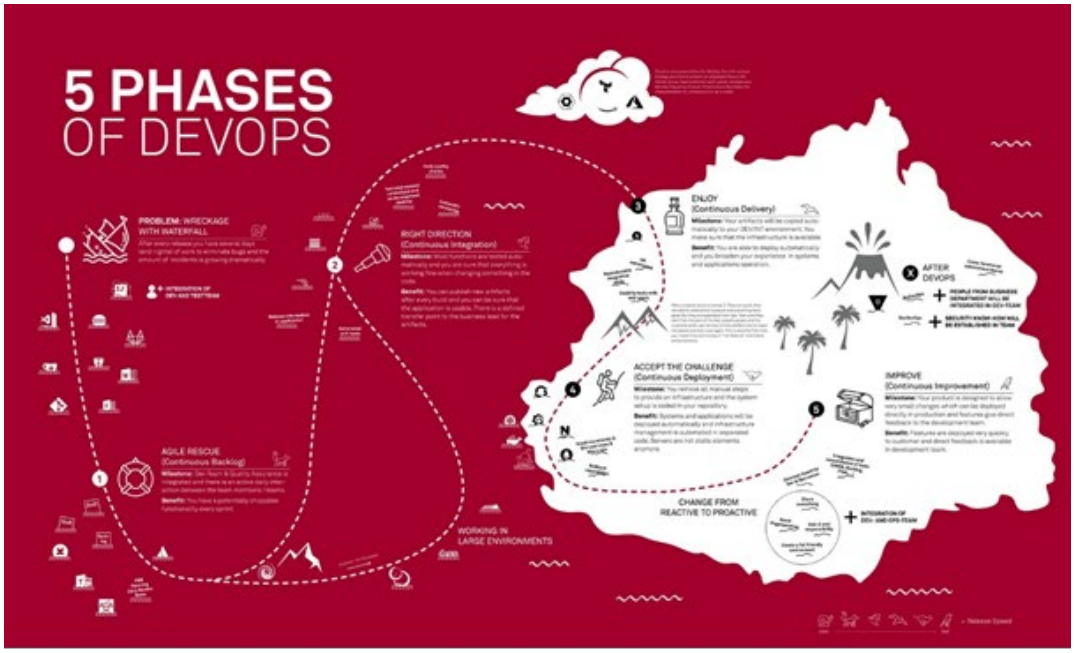

Bild 2: 5 Phases of DevOps – Der Weg zum High Performance Delivery

Problem: Wreckage with Waterfall

Das ist der Ausgangspunkt. Sozusagen die Phase 0. Hier sind über alle Betrachtungsdimensionen hinweg wesentliche Grundelemente der Reifegrade noch nicht erfüllt. Wenn man so will, ist das unser Beispiel vom Anfang des Artikels. Wasserfall (eine Entwicklungsstufe folgt der nächsten, keine oder wenig Rückkopplung zwischen den einzelnen Elementen des Value Streams), getrennte Verantwortlichkeiten, manuelle Tätigkeiten, hohe Unsicherheiten im Auslieferungsprozess und der erwarteten Qualität des Produktes kennzeichnen diese Phase.

Phase 1: Agile Rescue – Continuous Backlog

Agilität ist eine der wesentlichen Voraussetzungen für ein erfolgreiches Adaptieren von DevOps. Phase 1 setzt voraus, dass das Team auf Basis agiler Vorgehensweisen arbeitet. Wesentliches Steuerungsinstrument für die Arbeit des Teams ist das Backlog, welches alle anstehenden Tätigkeiten beinhaltet. In dieser Phase arbeiten Fachbereich und Entwicklung durch die agile Vorgehensweise bereits integriert. Der Product Owner definiert die Anforderungen und priorisiert diese gemäß den Bedarfen der Anwender für die Entwicklung. Ziel ist es hier ein funktionsfähiges Inkrement nach jedem Sprint zu erzeugen.

Phase 2: Right Direction – Continuous Integration

Sobald Software oder auch andere Produkte durch ein oder mehrere Teams erstellt und bearbeitet werden, müssen die jeweils einzelnen Teile immer wieder integriert werden. In der Vergangenheit war das oft ein losgelöster, zeitversetzter Prozess. Erst wenn dieser durchlaufen war, erhielt der einzelne Entwickler ein Feedback darüber, ob seine Anpassungen richtig in das System übernommen werden konnten und das System mit diesen Anpassungen noch funktioniert. Dieser Versatz verhindert einen

kontinuierlichen Fluss im System, zu spät erkannte Probleme müssen aufwändiger nachgearbeitet werden im Vergleich zu einem sofortigen Feedback. In dieser Phase wird begonnen diese Prozesse zu automatisieren und mit ersten automatisierten Tests und Quality Checks abzusichern.

Phase 3: Enjoy – Continuous Delivery

Mit Continuous Delivery ermögliche ich eine jederzeitige Auslieferbarkeit meines Produktes. Ich bin jederzeit in der Lage durch das Anstoßen eines automatisierten Prozesses das System auf einer Testumgebung, einer Integrationsumgebung für Anwendertests oder der Produktionsumgebung für die Nutzung durch die Anwender bereitzustellen.

Spätestens in dieser Phase werden auch Entwicklung und Betrieb integriert. Das Backlog dient dann als gemeinsamer Arbeitsspeicher. Wir unterschieden im Betrieb zwischen planbaren Aktivitäten, wie z.B. dem Einspielen eines notwendigen Patches wegen einer Versionserhöhung, und nicht planbaren Aktivitäten, wie z.B. aufkommenden Incidents, die vom Team behoben werden müssen. Alles was planbar ist, muss im Backlog gemeinsam für das Team priorisiert werden. Das Backlog ist der Arbeitsvorrat für das Team. Durch die Betrachtung von Entwicklungs- und Betriebstätigkeiten, wird der für die Priorisierung der Arbeit verantwortliche Mitarbeiter (im agilen Vorgehen nach Scrum ist das der Product Owner) gezwungen, beide Aspekte zu berücksichtigen und gegeneinander abzuwägen: "Was ist aktuell wichtiger und erzeugt den größeren Mehrwert für meine Anwender?"

Phase 4: Accept the Challenge – Continuous Deployment

Jede Änderung am Produkt wird sofort den Anwendern produktiv zur Verfügung gestellt. Es gibt keine manuelle Absicherung, kein Change-Advisory-Board, welches die Änderungen genehmigen müsste. Sie haben Prozesse und Qualitätssicherung so weit optimiert, dass Sie jede Änderung sofort verfügbar machen können, ohne Angst haben zu müssen, dass ihr Produkt für Fehler sorgt.

Phase 5: Improve – Continuous Improvement

Sind die Prozesse und Vorgehensweisen grundsätzlich implementiert, rückt die Frage in den Vordergrund, wie diese nun ständig verbessert werden können. Dazu werden Key Performance Indikatoren bestimmt, Messverfahren etabliert, Daten analysiert und weitere Maßnahmen identifiziert und umgesetzt.

	Organisation & Kultur	Prozesse	Technologie & Architektur	Anwender-/ Kundenorientierung
Phase 1: Continuous Backlog				

Phase 2: Continuous Integration				
Phase 3: Continuous Delivery				
Phase 4: Continuous Deployment				
Phase 5: Continuous Improve-ment				

Tabelle 1: Reifegrade und Betrachtungsdimensionen

Mögliche Maßnahmen für die verschiedenen Bereiche innerhalb der Phasen können sein:

- **Organisation & Kultur**:
  Ausprägung eines agilen Mindsets, benötigte Skills, Klärung der Stärken des Team-Aspektes, Klären von Entscheidungswegen und Verantwortlichkeiten etc.
- **Prozesse**:
  Definieren von schlanken Prozessen, Etablieren von KPIs, Performance-Messung, Priorisierung und Automatisierung
- **Technologie & Architektur**:
  Nutzung moderner Technologien (Cloud, Microservices etc.) sowie integrierter Toolsets
- **Anwender-/Kundenorientierung**:
  Einbindung von Anwendern und Kunden, Mehrwertorientierung und Feedback

Darstellung der Bewertung

Für die Darstellung der Bewertung hat sich ein Spinnendiagramm (s. Bild 3) bewährt. In diesem kann zum einen die aktuelle Bewertung des Ist-Standes eingetragen werden und zum anderen auch die im nächsten Schritt angestrebten Verbesserungen in den jeweiligen Betrachtungsdimensionen.

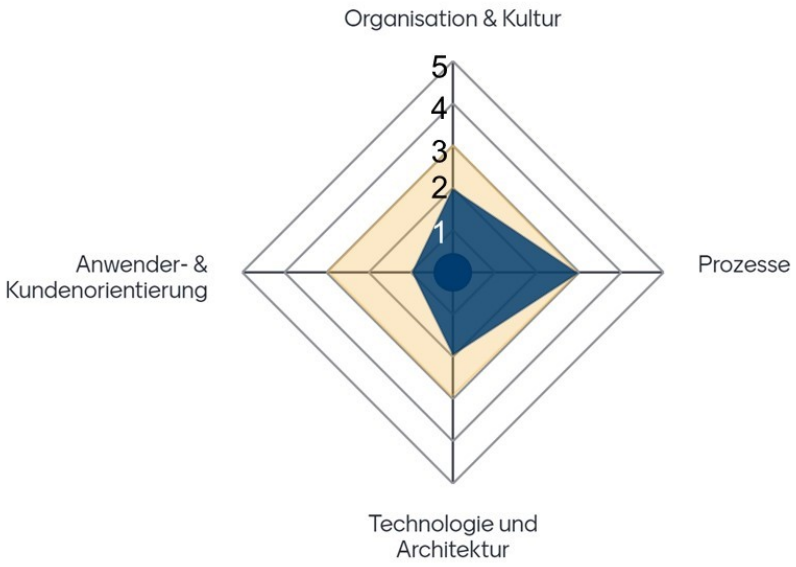

Bild 3: Bewertung Reifegrad und Potential

Orientierung - Wann wirkt etwas im Sinne von DevOps positiv?

Am Anfang der Überlegungen stand für uns die Frage: "Wann tue ich im Sinne von DevOps etwas Positives?" Wenn ich eine Verbesserung erzielen möchte, muss ich entscheiden können, ob sie – bezogen auf meine Ziele – einen Mehrwert liefert. Um diese Frage zu beantworten, haben wir einen Orientierungsrahmen entworfen, mit dessen Hilfe diese Frage, bezogen auf jede einzelne Maßnahme, beantwortet werden kann.

Hier haben wir uns auf der Suche nach einer Antwort ebenfalls zunächst die verfügbare Literatur angesehen. Aufbauend auf den Grundlagen des Lean Managements, beschreibt das Buch "The DevOps Handbook" [2] drei Wege oder Stoßrichtungen, die bei der Umsetzung von DevOps maßgebliche Rollen spielen.

Um in Workshops besser mit diesen Wegen arbeiten zu können, haben wir sie in unserem Orientierungsrahmen in Fragen umformuliert (s. Bild 4).

In der Diskussion mit Experten und den Ergebnissen von Interviews haben wir festgestellt, dass es im Kontext der Betrachtung des vollständigen Value Streams Sinn macht, einen weiteren Aspekt hervorzuheben: Die Frage nach der Priorisierung der Inhalte.

Mit diesen vier Fragen haben wir jetzt einen Orientierungsrahmen geschaffen. Jede Maßnahme, die ich für mein Vorhaben umsetzen möchte, muss einen positiven Beitrag im Kontext einer dieser Fragen liefern. Eine Maßnahme, die nicht positiv auf eine dieser vier Fragen wirkt, wird nicht umgesetzt.

Bild 4: Orientierungsrahmen

1. Wie kann ich den Fluss verbessern? (Flow)

Mit dieser Frage legen wir den Fokus auf die Verbesserung des Ablaufs in unserem Value Stream. Wie greifen die verschiedenen Produktionsschritte ineinander? Wie vermeide ich Verschwendung und reduziere Liegezeiten? Hier spielt aber auch z.B. Automatisierung eine große Rolle. Was muss ich tatsächlich manuell tun, was kann ich vereinfachen oder automatisieren?

2. Wie bekomme ich frühzeitig Rückmeldung? (Feedback)

Je früher ich Rückmeldung über mögliche Probleme in nachfolgenden Prozessschritten bekomme, umso schneller kann ich diese Probleme beheben. Damit steigere ich die Effizienz und erhöhe die Qualität. Eine mögliche Maßnahme, die positiv auf diese Frage wirkt, ist z.B. der Einsatz von Testautomatisierung in der Entwicklung. Im laufenden Betrieb betrachten wir das Monitoring des Systems, um Feedback über den Zustand zu erhalten, bevor Probleme auftreten. Maßnahmen können sich aber auch auf den Anwender beziehen und z.B. mit Hilfe von MVP (Minimum Viable Product) oder Experimenten untersuchen, wie neue Funktionen von Anwendern angenommen werden, bevor unnötig viel Ressourcen investiert werden.

3. Wie kann ich mich fortlaufend verbessern? (Continuous Improvement)

Sobald ich Abläufe, Prozesse und Strukturen einmal etabliert habe, ist das Ziel die kontinuierliche Verbesserung dieser. Konnte ich die Releasezyklen von dreimal jährlich auf monatliche Releases verbessern, kann man daran arbeiten, diese Zeiten weiter zu reduzieren. Auch das Schaffen einer fortlaufend lernenden und sich weiterentwickelnden Organisation gehört zu den Dingen, die hier positiv wirken.

4. Wie priorisiere ich richtig? (Value)

Diese Frage haben wir zusätzlich hinzugefügt, da sie aus unserer Sicht einen zentralen Stellenwert hat. Leider wird sie oft vernachlässigt. Hier geht es um die Frage, ob ich mich auf die Dinge konzentriere, die im Sinne meiner Kun-den oder Anwender tatsächlich jeweils den meisten Wert schaffen. Ein Kollege

von mir hat das sehr treffend auf den Punkt gebracht: "Wenn wir uns diese Frage nicht stellen, schaffen wir mit DevOps vielleicht nur eine Möglichkeit, die falschen Dinge schneller zu tun" - wir kommen schneller die Leiter hoch, diese steht jedoch an der falschen Wand.

Bewertung und Priorisierung der Maßnahmen

Wenn während der Workshops Maßnahmen aus den Projekten/Vorhaben identifiziert werden, nutzen wir ein einfaches Bewertungsschema um die Maßnahmen zu gewichten und gegeneinander zu priorisieren (s. Bild 5).

Für jede der vier Fragen bewerten wir je identifizierter Maßnahme, ob ein negativer (-1), neutraler (0), leicht positiver (1), positiver (2) oder stark positiver (3) Beitrag vorhanden ist. Aus dieser Bewertung berechnen wir je Maßnahme einen Score, der den Nutzenbeitrag der Maßnahme charakterisiert. Diesen Score multiplizieren wir im Anschluss noch mit einem Faktor, bezogen auf die Umsetzbarkeit hinsichtlich des Termins, der Zeit und der Kosten.

Aus diesem Vorgehen ergibt sich eine priorisierte Liste von Maßnahmen mit jeweils klar definiertem Wertbeitrag, die dann einer Entscheidung und weiteren Planung zugeführt werden kann.

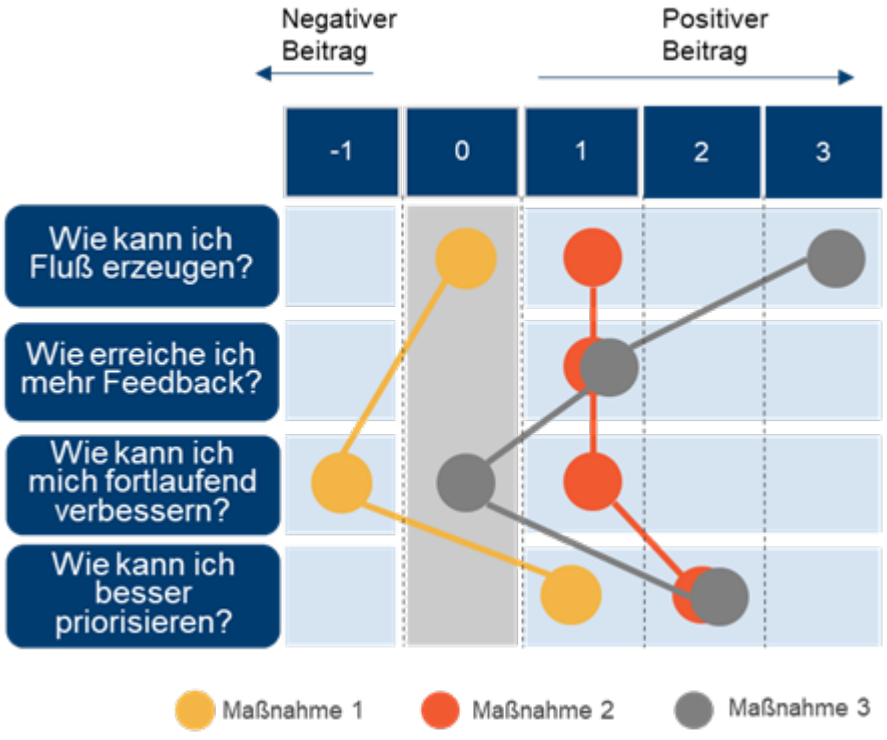

Bild 5: Bewertung von Maßnahmen anhand des Orientierungsrahmens

Fazit

DevOps ist ein weiterer Ansatz, um die bereits bekannten Prinzipien aus dem Lean Management in die Softwareentwicklung zu transferieren. Während Agilität hinterfragt, ob wir auf dem richtigen Weg sind (Effektivität) sorgt DevOps dafür, dass wir auf diesem Weg schneller und effizienter werden.

Um DevOps erfolgreich umzusetzen, bedarf es einer ganzheitlichen Betrachtung. Damit tun sich viele Projekte schwer. Ihnen fehlen die Landkarte und ein Bezugsrahmen als Grundlage für Entscheidungen. Hier setzen wir mit den 5 Phasen und dem Orientierungsrahmen an.

Unsere Erfahrung zeigt, dass mit diesem Vorgehen in nahezu jedem organisatorischen Setup Verbesserungen identifiziert werden können. Das volle Potential erreicht man jedoch nur, wenn tatsächlich die Verantwortung für den gesamten Value Stream in einer Hand liegt.

Literatur:

[1] State of DevOps Report 2019, Dr. Nicole Forsgren, Dr. Dustin Smith, Jez Humble, Jessie Frazelle

[2] The DevOps Handbook, Gene Kim, Jez Humble, Patrick Debois, John Willis