OPITZ CONSULTING

Digitale Service Manufaktur

Application Lifecycle Management neu ausrichten

Wege zu mehr Effizienz und Innovation im ALM

Whitepaper

Application Lifecycle Management neu ausrichten

Wege zu mehr Effizienz und Innovation im ALM

Über den Autor

Rolf Scheuch – Rolf Scheuch ist Diplom-Mathematiker und hat OPITZ CONSULTING mitbegründet. Dort ist er heute als Chief Strategy Officer tätig und arbeitet zudem als Management Coach und Autor zu Themen wie geschäftszielorientierte IT-Strategie und organisatorische Implementierung von Initiativen im Umfeld des Business- und IT-Alignments.

Kontakt

Rolf Scheuch,
Chief Strategy Officer
OPITZ CONSULTING Deutschland GmbH
Standort Gummersbach
Kirchstraße 6, 51647 Gummersbach
rolf.scheuch@opitz-consulting.com
+49 (0)2261 6001-1223

Impressum

OPITZ CONSULTING Deutschland GmbH
Kirchstr. 6
51647 Gummersbach
+49 (0)2261 6001-0
info@opitz-consulting.com

Disclaimer

Inhalt

Vorwort

Der Autor problematisiert die aktuellen Ansätze beim Application Lifecycle Management (ALM) und fordert, die Erfolgsfaktoren „Effektivität" und „Time to Market" mehr als bisher einzubeziehen und damit ein Umdenken zu bewirken, hin zu einem an Wertschöpfung und Produkt orientierten ALM. Hierbei spielen die Lessons Learned aus dem Umfeld der agilen Ansätze und dem neuen Paradigma des Continuous Delivery eine wesentliche Rolle.

In seinen Thesen zeigt der Autor die Schwächen „typischer" ALM-Ansätze auf und erklärt, warum diese aus seiner Sicht nicht zu der erhofften Zufriedenheit führen. Das Whitepaper schließt mit einer Diskussion und gibt einen Ausblick auf mögliche Ansätze.

Wo liegt das Problem?

Die heutige Applikationslandschaft ist komplex, tendenziell zu mächtig und unterliegt steter Veränderung. Komplexität und Geschwindigkeit werden sich durch den Druck des Marktes zur permanenten Erneuerung und die Veränderung der Geschäftsmodelle zukünftig noch weiter erhöhen. Obwohl Wandel ein natürlicher Vorgang im Wirtschaftsleben ist, wird es in vielen Industriesegmenten zunehmend schwierig, sichere Prognosen zu stellen. Sinkende Interaktionskosten und der Wegfall von Handelsbarrieren verstärken den Druck auf viele Geschäftsmodelle. Standardisierungen der Wirtschaftsdaten in vielen Bereichen durch Industrienormen und die Regelwerke der Compliance, wie auch die stetig steigenden und neuen Möglichkeiten der Informationstechnologie, heizen diesen Wandel zusätzlich an und erhöhen den Druck zur Anpassung der Applikationslandschaft.

Die Applikationsentwicklung und -wartung steckt in einem Dilemma: Die klassischen Ansätze zur Wartung und Modernisierung von Applikationen binden einen erheblichen Anteil des IT-Budgets, sind in einem hohen Maße arbeitsteilig und lähmen somit die notwendigen Innovationen und Anpassungen der IT-Lösungen an die sich verändernden Geschäftsmodelle. Der eigentliche Grund dieses Missstands liegt in den vordergründigen Zielen des ALM: Kostenminimierung und Effizienzstreben.

Typische Anzeichen einer schleichenden Verschlechterung der Situation sind:

- Die IT-basierende Innovation nimmt im Unternehmen ständig ab.
- Die Kosten für Entwicklung und Wartung steigen beständig ohne eine adäquate Steigerung des Wertschöpfungsanteils der IT-Lösung.
- Der „Technical Debt" der Applikationen steigt beständig.

Treten diese Indikatoren im Unternehmen auf, liegt die Notwendigkeit zur Überarbeitung der bestehenden Applikationsstrategie nahe.

Begriffsbestimmung ALM

In der wissenschaftlichen Literatur findet man zurzeit keine grundlegende oder einheitliche Definition für ALM. Die Anbieter der ALM-Produkte nutzen diesen Freiraum, um die Definition des ALM auf ihre Produktvermarktung auszurichten. Diese Definitionen spiegeln die unterschiedlichen Sichtweisen wider und beschränken sich dabei auf die optimale Unterstützung des Softwareentwicklungsprozesses, auf die Wartung oder auf den Betrieb von Applikationen.

Ferner beachten diese Ansätze nicht, dass die Methodik und die Vorgehensmodelle sich im Zeitablauf verändern können und meist auch verändern müssen, wenn Applikationen im Lebenszyklus Strategieänderungen erfahren.

Eine Applikation kann etwa als Innovation mit unklaren Anforderungen starten und agile Ansätze zur initialen Erstellung der Lösung nutzen, um dann im Lebenszyklus zu formaleren Releasezyklen zu wechseln, die Dynamik zu drosseln und Kostensenkungen zu verfolgen. Andererseits ist es aus Sicht der Investitionssicherung üblich, umfangreiche Modernisierungen an bestehenden Legacy-Systemen vorzunehmen. Hier passt sich das Vorgehen wieder an die individuellen Rahmenbedingungen an.

Chappell und Ballas haben ihre Definitionen von ALM folgendermaßen formuliert: *"An application's lifecycle includes the entire time during which an organization is spending money on this asset, from the initial idea to the end of the application's life." (David Chappell, 2008) "The conscious planning and management of implemented applications and software components which enable efficient and effective business processes throughout the enterprise."* (Linda Ballas, 2010)

Die Aussage von Dave Chappell hebt die gesamte Zeitdauer des ALM hervor und nicht nur die Wartungsphase oder Entwicklungsphase. Linda Ballas Aussage drückt die Bedeutung der Applikationsstrategie und der Governance bei einer zielgerichteten Planung der Weiterentwicklung aus, die sich am Beitrag zur Wertschöpfung („effective business processes") durch die Applikation orientieren soll.

Ich selbst bevorzuge diese Definition von ALM: *„ALM ist das Management des Assets „Applikation", über den gesamten Lebenszyklus von der Idee bis zum End-of-Life, mit dem Zweck, die Anwendungssysteme zeitgerecht, verlässlich und anforderungsbezogen zu liefern und gleichzeitig den Wertbeitrag der Applikation kontrolliert und an den Bedürfnissen des Geschäfts ausgerichtet zu gestalten."*

Im Mittelpunkt steht die Beachtung des Beitrags des Produkts an der Wertschöpfung. Die Definition lässt offen, ob der Wertbeitrag kontrolliert gesenkt wird, etwa im Rahmen des Sundowning der Applikation, mittels Modernisierung oder Weiterentwicklung bewusst gesteigert wird oder durch grundlegende Wartungsaufgaben nur gehalten wird. Entscheidend ist die Aussage, dass die Veränderung des Produkts und damit des Wertbeitrags der Anwendungen kontrolliert und planvoll erfolgt. Somit ist auch klar, dass die klassischen Kennzahlen des Projektmanagements zur Erfolgsmessung hier nur bedingt weiterhelfen. Die Kennzahlen sind nur eine Bewertung der Abwicklung. Entscheidend für ein erfolgreiches ALM und das implizite Produktmanagement ist der Nutzen für die Fachbereiche. Hierzu müssen belastbare Modelle für Kennzahlen mit dem Fachbereich entwickelt und die Ausprägung verfolgt werden.

Aus der Welt der Softwareentwicklung tauchen zurzeit einige neue Begrifflichkeiten auf, wie Lean ALM, Agile ALM, DevOps und ALM 2.0 oder sogar ALM 3.0. Diese Ansätze weiten die kooperativen und agilen Ansätze aus der Entwicklung auf den Bereich der Wartung aus. In der Konsequenz wird ALM als Einheit von Entwicklung, Weiterentwicklung und Wartung verstanden.

Dies erfolgt nun in zwei Richtungen:

Zum einen werden die agilen Ansätze der Entwicklung zur Steigerung der Adaptivität und zur Reduktion der Komplexität auf den Bereich des Deployments und des Betriebs der Lösung ausgedehnt. Der Regelkreis der Entwicklung zieht nun die bislang starre Schnittstelle zum Betrieb mit den etablierten Hand-over-orientierten Deployment-Prozessen ein und fordert eine kontinuierliche Inbetriebnahme der Lösungen (Continuous Deployment), die an die kurzen Entwicklungszyklen angepasst sind. Dies erfordert ein hohes Maß an Automatisierung bei der Qualitätssicherung und beim produktiven Deployment und gehört zum Aufgabenbereich der Mitarbeiter mit der Rolle „DevOps“.

Auf der anderen Seite fordern diese Ansätze eine starke Einbeziehung der Fachbereiche bei Entwicklung und Change Management, um die aus Sicht des Fachbereichs „richtige“ Software zu erstellen.

Die Schwächen aktueller ALM-Ansätze

Im Folgenden stelle ich vier kontroverse Thesen auf und erläutere diese im Anschluss eingehend. Im Hintergrund steht meine Kritik am scheinbar immer noch unerschütterlichen Glauben an die Möglichkeit, Effizienz und Effektivität durch einen arbeitsteiligen und sequentiellen Prozess im ALM zu steigern. In Anlehnung an Peter F. Drucker stelle ich diesem weitverbreiteten Glauben eine andere Aussage gegenüber: Die moderne Softwareentwicklung und -wartung erfolgt durch interdisziplinäre „Knowledge-Worker“, deren Produktivität und Steuerung durch Zielorientierung und Selbstorganisation erfolgt. Zwischen der aktuellen Situation mit ihrer Sichtweise auf die Softwareproduktion als arbeitsteiligen Prozess und den aktuellen Notwendigkeiten des ALM klafft eine Lücke.

Aus dieser Feststellung ergeben sich für mich vier teilweise provokante Thesen:

1. Das Denkmuster der „Software-Fabrik“ führt in die falsche Richtung.
2. Die strikte Trennung von Entwicklung, Betrieb und Support verhindert Innovation.
3. Off-shoring und agile Ansätze für das ALM passen nicht wirklich zusammen.
4. Umfassendes Tooling ist keine Lösung für das ALM.

These 1: Das Denkmuster der „Software-Fabrik“ führt in die falsche Richtung.

Im ersten Jahrzehnt dieses Jahrhunderts haben viele große Beratungshäuser und vorrangig auch die ausländischen Outsourcing-Häuser mit dem Schlagwort „Software-Fabrik“ geworben. Software wird in der Fabrik gefertigt mit einem optimierten Produktionsprozess. Dies sollte durch die enge Analogie zur industriellen Produktion ausgedrückt werden und die Effizienz stand dabei im Mittelpunkt.

Die Erfahrungen der letzten Dekade geben jedoch ein gemischtes Bild wieder. Allen Einschätzungen gemeinsam ist, dass der „Fabrik-Ansatz“ nicht die Effektivität der Software und damit den erhofften Beitrag der IT an der Wertschöpfung erzielt hat. Und das liegt letztlich an den großen Unterschieden bei der Messung des Erfolgs.

Seitens der IT herrschte bislang eine lieferantenorientierte Projektsicht vor: In Zeit, Qualität und Budget wird ein fest definierter Funktionsumfang umgesetzt und implementiert. An diesen Kriterien misst sich der Erfolg des Projekts. Effektivität ist nicht die Aufgabe einer solchen „Fabrik“: Anforderungs-Junk In, Software-Junk Out. Veränderungsbereitschaft in den Projekten ist nicht erwünscht. Ferner hat die IT sich als Service-Organisation mit SLAs und ITIL-Prozessen ausgerichtet, was nur selten zu einer Partnerschaft mit gemeinsamer Erfolgsmessung führt – und damit auch nicht zu einem verbesserten Beitrag der IT-Systeme an der Wertschöpfung!

Der Erfolg der agilen Ansätze zeigt deutlich die Lücken dieser Sichtweise auf, denn bei diesen Vorgehensweisen erfolgt die Integration des Fachbereichs, der Entwicklung und der Qualitätssicherung in einem Team.

- Take Away: Neben der Effizienz ist auch die Effektivität wichtig!

These 2: Strikte Trennung von Entwicklung, Test und Betrieb verhindert Innovation.
In der Regel erfolgt eine strikte Trennung von Entwicklung, Qualitätssicherung, Testmanagement und Betrieb. Wie am Fließband übergibt ein Bereich sein Gewerk an die nächste Stelle und kommuniziert die Fertigstellung mit einer Abnahmeprozedur. Folge: Die IT-Organisation erlahmt an den internen Prozeduren. Der Ausweg sind meist wenige, aber dafür entsprechend große Releasewechsel – oft nur einmal pro Halbjahr. Hierdurch verlängert sich der „Time to Market" einer neuen Idee, die Innovation erlahmt und den Fachbereichen geht wertvolle Zeit für Innovationen und eine verbesserte Wertschöpfung in den Geschäftsprozessen verloren.

Das neue Paradigma des „DevOps" verspricht hier eine Lösung. Entwicklung, Test und „Operations" (Betrieb) agieren in enger Abstimmung innerhalb eines Regelkreises. Mit diesem Denkmuster sind jedoch auch Schwierigkeiten verbunden: Continous Build und Deployment der Software muss möglich sein, alle Testszenarien müssen automatisierbar und die Deployment Prozesse ebenfalls automatisiert sein. Und noch nicht schwierig genug, kommt nun erst die eigentliche Herausforderung: Ein begleitendes Veränderungsmanagement muss die Akzeptanz häufiger Releasewechsel und damit verbundener Veränderungen bei den relevanten Anwendern erreichen.

- Take Away: Neben der Effizienz sollte man auch den „Time to Market" beachten!

These 3: Umfassendes Tooling ist keine Lösung für das ALM.
Eingangs hatte ich bereits auf die unterschiedlichen Definitionen des ALM hingewiesen. Einige Produkthersteller nutzen diese Uneinheitlichkeit geschickt aus, um ihre Funktionalität als Lösung für ALM zu vermarkten. Somit impliziert die Toolauswahl auch Vorgehensweise, Philosophie und Prozesse des ALM. Da sich jedoch im Lebenszyklus der Applikation auch die Notwendigkeit des Entwicklungsansatzes verändern kann, erscheint das rigide Einhalten einer Vorgehensweise nicht zielführend.

Ich halte es für angebracht, den komplexen Entwicklungsprozess mit einem Lean-Management-Ansatz zu versehen, der sich stetig verbessert und neu ausrichtet. Somit wäre das Tooling eher als eine Integrationsleistung von einer „Familie" an Werkzeugen zu sehen. Die Tools dieser „Werkzeugfamilie" werden aufeinander abgestimmt, unterliegen einer permanenten Veränderung und optimieren den eigenen ALM-Ansatz.

ALM ist somit weder ein Produkt noch eine Sammlung von Tools, sondern es handelt sich um einen methodischen Ansatz, der Handlungsfelder aufzeigt und etablierte, meist isoliert verwendete Werkzeuge, Methoden und Vorgehensweisen zu einer ganzheitlichen Methode verbindet.

- Take Away: Das ALM-Tooling ordnet sich der Vorgehensweise unter!

These 4: Off-Shoring und agile Ansätze für das ALM passen nicht wirklich zusammen.
Die letzte These ist wahrscheinlich die provokanteste. Maßnahmen wie Outsourcing oder Off-Shoring verfolgen im Allgemeinen das Ziel, Prozessschritte an spezialisierte Dritte zu übergeben, um die eigene Fertigungstiefe zu reduzieren. Im Denkmuster der „Software-Fabrik" mag dies ebenfalls funktionieren, da die Anforderung dort fix, fachlich einwandfrei und in sich abgeschlossen vorliegt. Die Voraussetzung dafür muss in der Realität aber erst mit hohem Aufwand geschaffen werden, und so stellt sich die Frage, ob die Organisation ihre Kraft wirklich in die Spezifikation, das Projekt- und Qualitätsmanagement investieren – oder ob sie nicht stattdessen eher in kleineren Schritten und stets überprüfbar, ein einheitliches Ziel verfolgen sollte.

Folgt man den agilen Ansätzen, so ist das Team beim ALM eng an den Fachbereich angebunden und entwickelt mit diesem gemeinsam die Lösungen. Diese enge und fachlich orientierte Zusammenarbeit scheint bei einem Off-Shoring mit agilen Ansätzen nicht möglich zu sein.

- Take-Away: Off-Shoring und kollaborative ALM-Ansätze widersprechen sich!

Alternative: Produktzentrischer ALM-Ansatz

Im Fokus des ALM, auch gemäß der gewählten Definition, steht das (Software-)Produkt, das aus einem oder mehreren Applikationen besteht, mit seinem Beitrag an der Wertschöpfung durch IT-Unterstützung der relevanten Geschäftsprozesse.

Ziel ist es, ein ALM mittels leichtgewichtiger Prozesse zu implementieren, die eine stetige Anpassung erfahren und so kontinuierlich den Wertbeitrag des Produkts über den Lebenszyklus sichern und an die Anforderungen anpassen. In diesem Verständnis ist ALM mit einem Produktlebenszyklus vergleichbar. Insbesondere ist hierdurch, ähnlich dem Mehrwert eines Produkts für den Anwender oder Käufer, das ALM nicht projektzentrisch als Folge von isolierten (Teil-)Projekten, sondern produktzentrisch am Nutzen und der Wertschöpfung orientiert. Hierdurch müssen IT und Fachbereich als Partner mit einer gemeinsamen Verantwortung ALM als kontinuierliche Produktentwicklung begreifen.

Die folgende Abbildung stellt die wesentlichen Unterschiede der Betrachtungsweisen gegenüber und zeigt die organisatorischen Implikationen der unterschiedlichen Sichten auf. Von der auf Effizienz ausgerichteten Sicht auf das ALM erfolgt ein Übergang zur Steuerung der Effektivität der gemeinsamen Lösung. Das ist mehr als ein „Alignment", eine „Ausrichtung an" oder eine „Angleichung an": Hier übernimmt man gemeinsam Verantwortung für ein Produkt.

Abbildung: Produktorientierung verzahnt IT und Fachbereiche

Beide Seiten arbeiten gemeinschaftlich sinnvolle Anforderungen aus und bewerten diese. Der Anstoß dazu kann von der IT oder dem Fachbereich kommen. Einzig der Wertbeitrag entscheidet über Budget oder Realisierung und dient damit auch als Maßzahl für den Erfolg.

Fazit

Bereits kleinere Schritte in Richtung einer produktzentrischen Sicht auf das ALM weichen die Schranken auf zwischen Fachbereich und Entwicklung auf der einen Seite und Entwicklung und Betrieb auf der anderen Seite. Dies führt zu einer höheren Effektivität der (Software-)Produkte und zu mehr Innovation. Und ganz nebenbei und kostenlos „gewinnt" man die Identifikation von Fachbereich und IT mit einem gemeinsam geschaffenen Produkt.

Quellen

David Chappel: „What ist Application Lifecycle Management?", http://www.davidchappell.com/whatisalm--chappell.pdf – 2010.

Linda Ballas: „Application Lifecycle Management", http://itproforum.org/archive/201003_ballas.pdf – 2008.

Peter F. Drucker: "The Five Most Important Questions You Will Ever Ask Your Organization", San Francisco, Jossey-Bass – 2008.