

Sind SAP-Activate-Projekte der Tod der Festpreisprojekte?

Management Summary

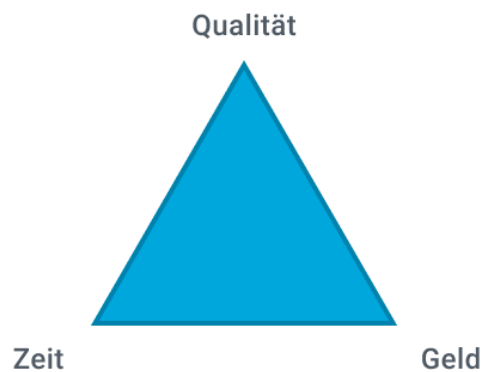
SAP Implementierungsprojekte sind seit jeher eine Herausforderung, sowohl aus der Sicht des Unternehmens, das diese Software zum Einsatz bringen möchte (Kunde), als auch aus Sicht der Unternehmensberatung, die dieses Unternehmen unterstützt.

Mit SAP Activate stellt die SAP eine neue Methode zur Implementierung ihrer Software vor. Diese greift dabei vorzugsweise auf solche Instrumente zurück, die aus der agilen Softwareentwicklung bekannt sein mögen. Erfahren Sie in diesem Whitepaper, wie SAP Activate mit agilen Werkzeugen eine Implementierung unterstützt. Nach dem Lesen werden Sie erste Erkenntnisse gewonnen haben, ob diese Methode für Ihr Unternehmen oder als Unternehmensberatung Ihren Kunden geeignet ist und was sich auch in der Zusammenarbeit zwischen dem implementierenden Unternehmen und dem Beratungspartner ändern wird: nicht nur das Vorgehen, sondern auch die Preisvereinbarung zwischen den Partnern unterliegt einem Wandel.

Zielkonflikte in Projekten

Im Rahmen einer Softwareimplementierung stehen die beiden Partner (Kunde und Beratungspartner) zunächst einmal den grundlegenden Anforderungen von Projekten gegenüber. Es geht um Zielkonflikte, die häufig in Form des nachstehend abgebildeten Dreiecks dargestellt werden. Es gilt, die Zielstellungen Zeit, Geld oder Budget und Qualität zu einem gelungenen Interessenausgleich zu kombinieren. Bereits die drei Ecken des Dreiecks beinhalten Potentiale für Konflikte:

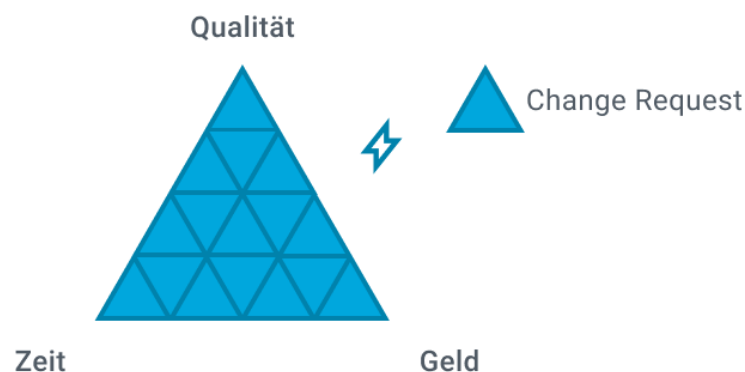
- Während der Kunde regelmäßig versuchen wird, mit einem möglichst schlanken Budget zur gewünschten Qualität zu gelangen, hat das Beratungsunternehmen prinzipiell Gewinnoptimierungsabsichten und freut sich insofern über jede Erweiterung des Budgets zu seinen Gunsten.
- Die Beratungsunternehmen bevorzugen im Allgemeinen Implementierungen, die dicht am Standard der SAP-Prozesse bleiben, wohingegen die Kunden am liebsten alle historisch liebgewonnenen Prozessbesonderheiten beibehalten möchten, oder sogar neue Wünsche abseits der Standardprozesse im Rahmen des Projektes formulieren. So kollidieren Best-Practise-Vorstellungen mit Customizing- und Programmierwünschen.
- Die Zeit steht dabei im abgebildeten Projektdreieck gleich für mehrere potentielle Zielkonflikte. Nahezu immer hat das Management des Kunden ein bestimmtes Datum für den Systemwechsel im Hinterkopf. Gleichzeitig soll die Belastung der Mitarbeiter durch das Projekt so gering wie möglich ausfallen, um den Anforderungen aus der Linienfunktion so gut, wie es geht, gerecht zu werden. Die Beratungsunternehmen wünschen sich für ihre Mitarbeiter hingegen eine möglichst konstante und gut planbare Auslastung.



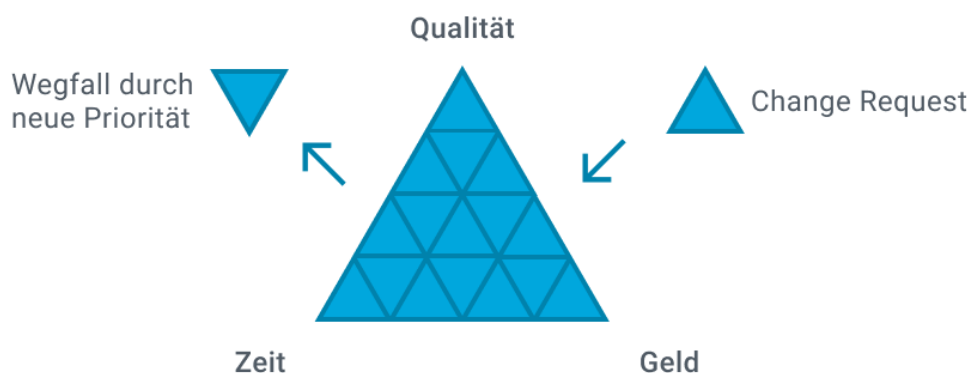
Selbstverständlich spitzt die Grafik die Interessenkonflikte zu. Und natürlich gab es auch in der Vergangenheit Projekte, in denen Beratungsunternehmen und ihre Kunden sehr harmonisch miteinander die Projektaufgabenstellung bewältigt haben. In vielen Projekten ist es allerdings erforderlich, im Projektverlauf Änderungen zu berücksichtigen. Ich möchte das wie folgt darstellen:

Nimmt man einmal an, dass die Grundfläche des Zielkonfliktdreiecks die Summe aller Aufgaben des Projekts in einer bestimmten Qualität und zu einem bestimmten Budget beschreibt, so kann das Dreieck unterteilt werden, zum Beispiel in kleinere Dreiecke. Das große Dreieck beschreibt also den Projektumfang (Scope) des Projektes und die kleineren Dreiecke sind Aufgabenpakete, die sich aus den Fachkonzepten ergeben. Gewissermaßen handelt es sich also um den sogenannten Business Blueprint (= Sollkonzept), angereichert um das Budget und Termine.

Ergeben sich jetzt im Projektverlauf Veränderungen an dem ursprünglichen Projektplan und unterstellt man vereinfachend, dass immerhin hierüber Einigkeit zwischen dem Kunden und dem Beratungspartner besteht, so wurde die neue Veränderung als Change Request (CR) für das Projekt beschrieben. Geschieht dies ordentlich, so wird der CR genauso beschrieben, dass wir erneut die Aussagen hinsichtlich Zeit, Qualität und Budget treffen. Der Versuch, nun allerdings den Change Request auf der Fläche des bisher beschriebenen Projektdreiecks zu platzieren, führt zu der Feststellung, dass hierfür keine Freifläche vorgesehen ist. Wird angenommen, dass es nicht so ohne weiteres möglich ist, zusätzliche Arbeitskräfte in das Projekt zu nehmen, da diese angelernt werden müssten, so ist der Change Request mindestens eine zeitliche Herausforderung, selbst wenn dessen Qualität und das Budget exakt beschrieben sind.



Diesbezüglich kam es in vergangenen Projekten häufig zu Konflikten zwischen den Beratungsunternehmen und den Kunden geführt haben, versuchen moderne Projektvorgehensmodelle anzusetzen. Die Sau, die hier durchs Dorf getrieben wird, heißt Agilität. Bedauerlicherweise wird diese Agilität häufig falsch verstanden. Zwar ist es dem Kunden jederzeit möglich, Änderungswünsche in das Projekt einzubringen, jedoch führt dies in der Regel dazu, dass er andere Dinge, die bisher zum Scope gehörten, in der Priorität so herabsetzen muss, dass sie aus dem Scope fallen, damit die Grundfläche des Dreiecks unverändert bleibt, sprich die Zeit und das Budget bleiben konstant, aber die Qualität wird an die geänderten Anforderungen angepasst. Die Verantwortung für derartige Änderungen liegt im agilen Projektvorgehensmodell beim sogenannten Product Owner¹. Diese Funktion wird stets vom Kunden besetzt. Damit hat per Definition der Kunde selbst die Verantwortung für die Qualität des Implementierungsprojektes.



¹ Der Product Owner ist eine zentrale vom Kunden selbst zu besetzende Rolle innerhalb des SAP-Activate-Projektes. Ihm obliegt die Aufgabe der Priorisierung der im Projekt umzusetzenden Anforderungen.

Preisgestaltungskonflikt der Vergangenheit

Bei näherer Betrachtung lässt sich konstatieren, dass im Grunde genommen alle Projekte der Vergangenheit irgendwelchen Änderungen im Projektverlauf unterlegen waren. Faktisch ist es demnach so, dass so gut wie nie diejenige Qualität in einem Produktivsystem live gesetzt worden ist, welche ursprünglich im Sollkonzept beschrieben war. Diese Tatsache akzeptierend, bedeutet dies, dass die Qualität in jedem Projekt volatil ist, also niemals konstant. Gleichzeitig versuchen allerdings sehr viele Kunden, ihre Projekte an die Beratungsunternehmen zu einem Festpreis zu verkaufen (für die öffentliche Hand und eine Vielzahl von Unternehmen ist dies über Ausschreibungen sogar teilweise zwingend vorgeschrieben (Ausschreibungspflicht)). Die Beratungsunternehmen dagegen, versuchen stets auf ein Vertragsmodell zu drängen, in dem nach Zeit und Aufwand abgerechnet wird. Verantwortung für die Qualität möchten sie nicht übernehmen. Das deutsche Recht unterscheidet hier zwischen dem Werkvertrag, für dessen Erfüllung eine konkret beschriebene Leistung – eben ein bestimmtes Werk – erbracht werden muss, und einem Dienstvertrag, bei dem lediglich ein Dienst angeboten werden muss, nicht aber ein bestimmtes Ergebnis geschuldet wird. Die Beratungspartner schulden hier nur die Arbeitszeit.

Projektwahrheiten

Zwei weitere Faktoren sind regelmäßig in Projekten anzutreffen und werden dennoch gerne verschwiegen oder übersehen:

1. Der Beratungspartner kennt das Leistungspotential der zu implementierenden Software im Normalfall besser als der Kunde, der die Software implementieren möchte. Dieser wiederum verfügt über das Wissen der derzeit gelebten Prozesse in seinem Unternehmen. Auch kann nur er exakt beschreiben, welche Prozesse für den Unternehmenserfolg von besonderer Bedeutung sind. Der Versuch Klarheit darüber zu erlangen, wie die IT-gestützten Prozesse der Zukunft aussehen sollen, kann deshalb erst dann gut funktionieren, wenn das Wissen beider Vertragspartner an einer Stelle zusammengefügt wird. Denn nur, wer beide Sichten kennt, kann letztendlich fundiert darüber eine Entscheidung treffen, ob ein Best-Practise-Prozess in der einzuführenden Standardsoftware vom Kunden durch Customizing adaptiert werden kann, oder ob es angezeigt ist, umfangreichere Anpassungen (Entwicklungen/Programmierung) an der Software vorzunehmen, um den besonderen Anforderungen des Kunden gerecht zu werden. Im Allgemeinen reift diese Erkenntnis auf beiden Seiten im Verlauf des Projektes; allein dies führt schon zu Veränderungen am Sollkonzept nach dem Motto: „Ja, wenn wir das gewusst hätten, dann ...“
2. Kommt es nun im Projektverlauf zu Veränderungen an den Anforderungen, also an der gewünschten Qualität, dann ist die Veränderung der Priorität, und damit der Entfall von ursprünglich geplanten Funktionen oder Leistungen, nur die halbe Wahrheit: Insbesondere in großen Projekten sind für bestimmte Aufgabenpakete Experten mit bestimmten Kenntnissen oder Fähigkeiten erforderlich. Wenn dieses Können nicht mehr gefordert ist, sondern eine vollkommen andere Expertise benötigt wird, so ist diese allerdings nicht immer ad hoc abrufbar. Kurzum, die Prioritätenänderung kann bedeuten: „Lass Deinen Berater für das Vertriebsmodul zu Hause und schicke mir bitte einen Experten für das Modul Finanzwesen!“. Dies mag auf dem Papier gut aussehen, ist in der Praxis aber nicht immer realisierbar.

Genau aus diesen Herausforderungen gilt es nun einen guten Interessenausgleich zwischen Beratungsunternehmen und Kunden zu finden. Wie kann dies gelingen?

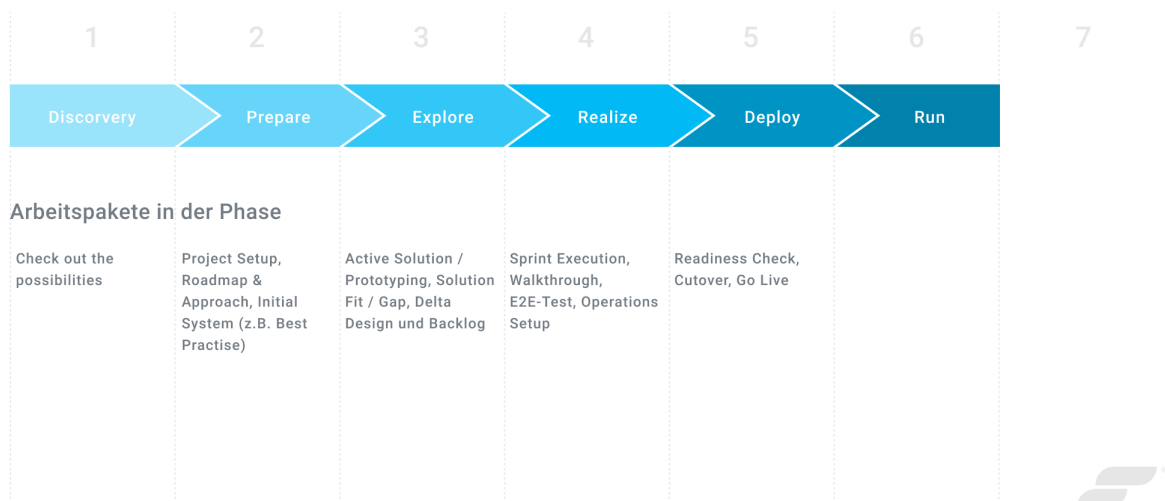
Wasserfall (ASAP) oder Agil (Activate)

Um diese Frage zu beantworten, gilt es zunächst zu verstehen, wie Implementierungsprojekte funktionieren. Dazu möchte ich die Projektphasen der SAP aus dem ehemaligen Vorgehensmodell ASAP und der neuen SAP-Activate Methode (hier der agile Ansatz) sowie die üblichen Werkzeuge und wesentlichen Inhalte der Phasen nebeneinanderstellen.

SAP® ASAP

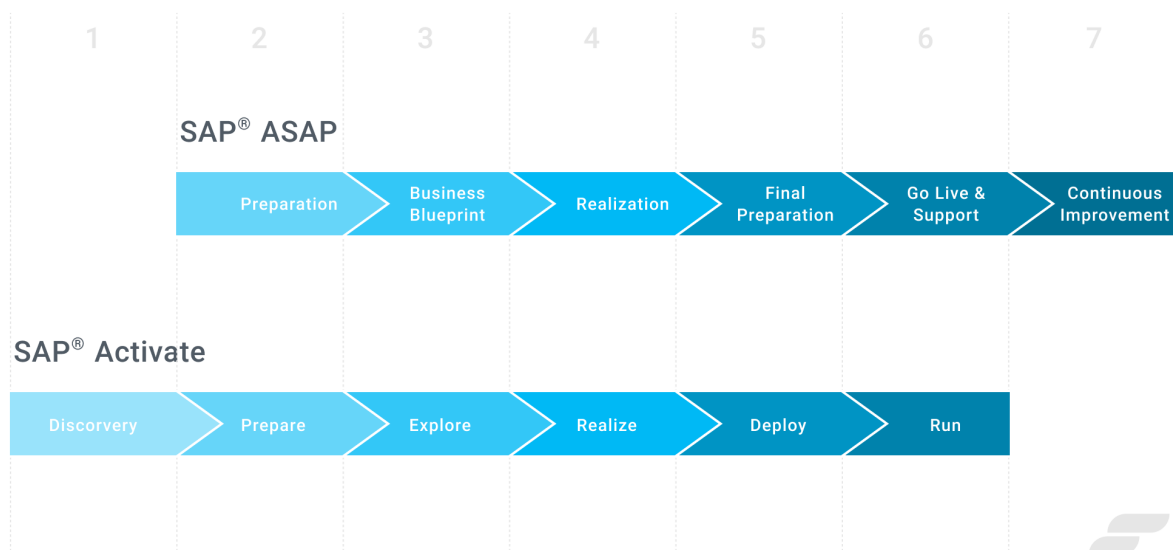


SAP® Activate



Sieht man einmal davon ab, dass die Phasen z. T. neu bezeichnet wurden, so scheinen die Änderungen auf den ersten Blick nicht so ungeheuer groß zu sein. Dennoch gibt es drei wesentliche Veränderungen.

1. ASAP war immer ein Wasserfall-basiertes Vorgehensmodell. SAP Activate ist heute üblicherweise agil (auch wenn es für On-Premise-Installationen noch eine Wasserfall-basierte Variante des Modells gibt).²
2. SAP Activate stellt die SAP-Systeme sofort bei Projektbeginn bereit. Dazu wird ein Standardsystem eingerichtet (z. B. aus Best-Practise-Prozessen der SAP). Damit hat der Kunde sofort zum Zeitpunkt des Projektstarts Zugriff auf sein System, kann daran wachsen und Erfahrungen sammeln. Auch erhält der Implementierungspartner dadurch sehr früh Rückmeldungen zum System. Das ehemalige ASAP-Modell (früheres Vorgehensmodell der SAP) hat den Kunden häufig erst spät auf das System gelassen, was dann auch zu spätem Feedback zur Funktionalität des Systems führte (mit dem Risiko der späten Enttäuschung oder des späten Scheiterns).
3. Wann immer es möglich ist, wird das aufwendige Abfassen eines Sollkonzeptes durch die Erstellung eines Delta-Backlogs³ ersetzt.



Chancen und Risiken

Genau in den drei vorgenannten Veränderungen liegen die größten Chancen und die größten Risiken. Viele Business-Blueprint-Dokumente, Sollkonzepte, Fachkonzepte, Lastenhefte, oder wie auch immer sie genannt wurden, hatten häufig nur einen nennenswerten Zweck: Sie sollten klarstellen, wann ein Change Request vorliegt und wann nicht. Oft wurde auch diese Aufgabe von den Konzepten nur mäßig erfüllt. Der Grund liegt darin, dass die Anforderungen ohne Ansehen der Standardsoftware niedergeschrieben wurden. Bei Ausschreibungen besteht sogar die Praxis, dass jene Unternehmen,

² Wasserfall-basierte Projekte haben einen sequenziellen Verlauf, agiles Vorgehen ist inkrementell. Mehr dazu siehe: SAP Activate – Agilität in SAP Implementierungsprojekten, Seite 51., Abb. 5.2, Martin Kipka, erschienen im Dez. 2020, Espresso Tutorials, ISBN 9783960124719

³ Ein Backlog ist eine priorisierte Liste mit textlich formalisierten Anforderungen des Kunden an die Software. Der Delta-Backlog beschreibt dabei nicht alle Anforderungen, sondern nur diejenigen, die vom Standard der auszuliefernden Software abweichen. Die Projektpartner sollten diesbezüglich in Abhängigkeit vom gewählten Ausgangssystem, an dem das Delta ermittelt wird, gemeinsam Kriterien erarbeiten, was als Delta und was als Standard betrachtet wird. So könnte z. B. alles, was vor den Workshops zur Ermittlung des Deltas an Einstellungen vorgenommen wurde, als Standard eingestuft werden, auch wenn hierfür bereits Customizing-Tätigkeiten erforderlich waren.

die an der Erstellung der Ausschreibungsunterlagen beteiligt waren, vom Hauptprojekt zwingend auszuschließen sind. Entsprechend haben diese Unternehmen häufig keine Idee vom Standard der später zu implementierenden Software. Kein Mensch käme auf die Idee, zunächst auszuführen: „Ich beschreibe mal vollkommen losgelöst von den Verkehrsmitteln, wie man von hier nach Kalkutta gelangt.“, um dann später zu sagen: „Ich kaufe oder miete ein Flugzeug, aber ich sollte es mit Pedalen vorwärtsbewegen können, weil ich in der Vergangenheit alle meine Reisen mit einem Fahrrad gemacht habe und dies mein geübter Fortbewegungsprozess ist.“. Klar hinken solche Vergleiche, aber genau solchen Unsinn liest man schon mal, wenn die Konzepte bzw. Pflichtenhefte durch fachfremde (= die zu implementierende Software ist unbekannt) Unternehmen erstellt werden.

Dieses Risiko wird im agilen Vorgehen dadurch umgangen, indem sofort bei Projektbeginn die Standardsoftware in Form einer Modellcompany der SAP oder ein durch Best Practise Prozesse ausgeprägtes System oder ein vom Implementierungspartner ausgeprägtes Baseline System o. ä. bereitgestellt wird, um es gegen die geplanten Prozesse zu verproben. Der Kunde schaut sich also den Standard der Software an und entscheidet dann anhand der werttreibenden Prozesse und Produkte seines Unternehmens, ob er diesen Standard übernehmen kann und will oder ob es noch einiger Anpassungen bedarf, die seinen Besonderheiten entsprechen. Diese Besonderheiten und Abweichungen vom Standard werden dann im Delta-Backlog formalisiert erfasst. Dies bedeutet, dass der qualitative Scope des oben erwähnten Projektdreiecks durch den Standard der Software, ergänzt um die Angaben des Delta-Backlogs, beschrieben wird.

Das ist vielversprechend – insbesondere für diejenigen Kunden, die dicht am Standard bleiben möchten, was regelmäßig die Wartung der Software vereinfacht und eben das aufwendige Erstellen der Sollkonzepte erspart. Es ergeben sich daraus aber sofort zwei Herausforderungen:

1. Wer Standardprozesse adaptiert, muss in der Regel seine Aufbau- und Ablauforganisation anpassen. Dies stellt das Organisational Change Management (OCM) vor große Aufgaben. Nicht selten ergeben sich hierbei sogar Auswirkungen auf die bestehenden Arbeitsverträge (Stellenbeschreibungen, Tarife, ...). Diese Veränderungsprozesse bedürfen also sehr hoher Aufmerksamkeit.
2. Das agile Vorgehensmodell erlaubt dem Kunden jederzeit den Backlog zu modifizieren, also qualitative Veränderungen in das Projekt einzubringen. Die Ausschreibungsunterlagen sicherten dem Kunden in der Vergangenheit einen bestimmten Leistungsumfang zu, den Beratungspartnern eine Beschreibung der zu erledigenden Aufgaben. Entfallen die Ausschreibungsunterlagen und werden auch die Sollkonzepte nur noch durch ein Delta-Backlog ersetzt, so kann kein Beratungsunternehmen ein seriöses Festpreisangebot unterbreiten. Allerdings waren diese Angebote in der Vergangenheit häufig auch nur Papiertiger, die dann durch Change Requests und Nachtragsangebote ausgehebelt wurden. Insofern ist das in vielen Fällen verschmerzbar, solange nicht aus anderen Gründen eine Festpreisvereinbarung zwingend geboten ist.

Damit stellt sich die Frage: „Wie kann eine Preisvereinbarung getroffen werden, die dem Kunden einerseits eine Budgetierung des Projektes erlaubt, die aber andererseits dem Beratungsunternehmen die Chance gibt, ohne ständige Nachverhandlungen eine zügige und erfolgreiche Projektumsetzung zu verfolgen?“

Dazu bedarf es einiger Grundvoraussetzungen:

1. Der Standard der Software sollte bekannt sein. Der Kunde muss schlichtweg vor Aufnahme der Arbeit wissen, welche Software er implementieren möchte. Genau aus diesem Grund bietet SAP die Discover-Phase an, in der das System zum Kennenlernen bereitgestellt wird und sich somit auch mit den Standards anderer Hersteller vergleichen lässt.
2. Der Kunde sollte seinem Beratungspartner trauen (können) und ihm die Einführung zutrauen. Umgekehrt sollte der Beratungspartner dem Kunden trauen können.

Wenn die Beratungspartner nach der Discover-Phase gemeinsam das Projekt starten, entscheidet der Kunde bzw. dessen Product Owner, *was* gemacht wird und die Implementierungsteams (zusammengesetzt aus Menschen unterschiedlicher Disziplinen des Kunden und des Beratungsunternehmens) entscheiden, *wie* es umgesetzt werden soll. Dabei entwickelt sich im Lauf der Zeit eine Projektfortschrittsgeschwindigkeit (Velocity), die bereits nach kurzer Zeit Aussagen zulässt, welche der im Delta-Backlog festgehaltenen Anforderungen sich bis wann umsetzen lassen. Damit können von den Vertragspartnern mehrere Ranges bestimmt werden. Es werden Worst-Case-Annahmen und Best-Case-Annahmen getroffen. Mit Blick auf das Projektdreieck kann man stark vereinfacht gegenüberstellen:

Projektzeitraum: frühestes mögliches Abschlussdatum vs. spätestes zulässiges Abschlussdatum

Budget: Budget-Untergrenze (Minimalvergütung) vs. Budget-Obergrenze

Qualität: Must-Have-Anforderungen vs. vollausgeprägtes Idealsystem

Damit ist das Festpreisprojekt zwar Geschichte, aber beide Implementierungspartner haben ein gutes Gefühl: Der Kunde weiß, was er mindestens bis wann bekommt und was er dann dafür zahlen muss. Die Beratungspartner bekommen die Chance auf ein veritables Projekt, wenn sie außergewöhnlich gut performen. Damit ist dann beiden Seiten gedient.

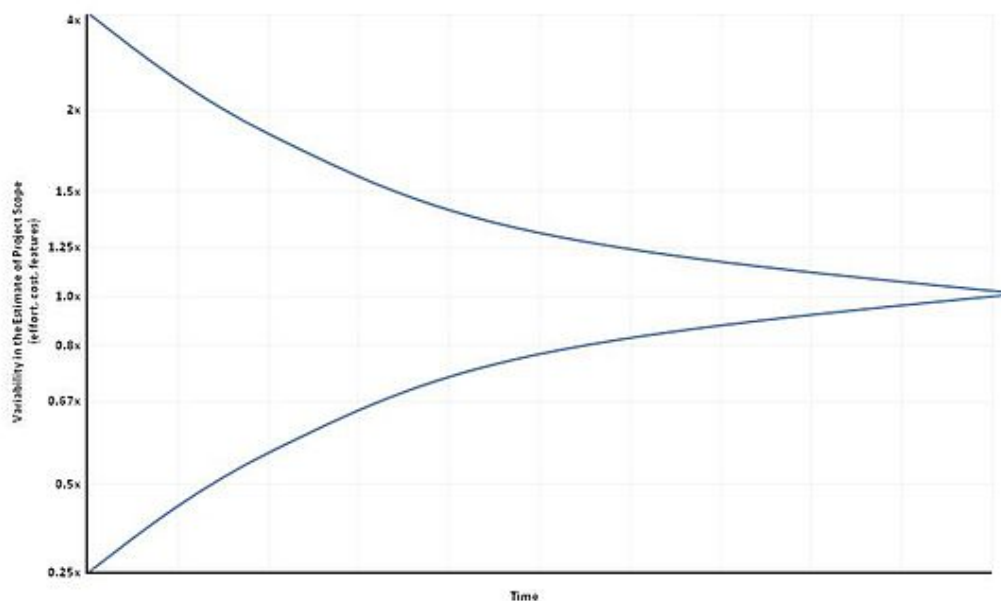
Entscheidend ist, dass die Partner durch ständiges Überprüfen der Priorisierung der Anforderungen zu dem System gelangen, das die wertschöpfenden Prozesse des Kunden am besten unterstützt. Dies kann folgendermaßen passieren: Der in der Explore-Phase erarbeitete Delta-Backlog enthält, wie oben ausgeführt, alle Anforderungen aus Kundensicht, die als Delta zu einem Standardsystem erkannt wurden. Das Implementierungsteam bewertet die Komplexität der Anforderung in Form von sogenannten Story Points. Je mehr Story Points eine Anforderung hat, desto komplexer wurde sie vom Team eingeschätzt. Die Anforderungen des Delta-Backlogs werden jetzt in zeitlich genau fixierten Umsetzungsphasen (Sprints) vom Team in der Software umgesetzt.

Dabei werden also in jedem Sprint bestimmte Anzahlen von Story Points abgearbeitet. Diese Abarbeitungsgeschwindigkeit von Story Points beschreibt dann folglich die Velocity, z. B.: Das Team schafft in einem Sprint 20 Story Points. Unterstellt man nun, dass ein Sprint immer zwei Wochen dauert, und misst man die Velocity der ersten drei Sprints, so kann man normalerweise schon nach sechs Wochen recht zuverlässig Aussagen darüber treffen, wie viele Story Points bis zu welchem Datum erledigt werden können. Dies hilft dem Product Owner bei der Priorisierung.

Unabhängig vom gewählten Vorgehensmodell sind Projekte stets mit Risiken behaftet. Dieser Herausforderung wurde in der Vergangenheit eben oft dadurch begegnet, dass man über die Festpreisvereinbarung mehr oder weniger erfolgreich wenigstens das finanzielle Risiko zu minimieren versuchte. Tatsache ist jedoch, dass das Risiko des Scheiterns für komplexe Projekte das Kernrisiko

ist. Die Wahrscheinlichkeit, dass nicht alles nach Plan läuft, nimmt dabei mit der Größe und der Komplexität des Projektes zu. Deshalb ist es viel geschickter, diesem Risiko dadurch zu begegnen, dass ein Scheitern möglichst früh erkannt wird, um ggf. noch abrechnen zu können, anstatt lange das Scheitern bewusst oder unbewusst auszublenden und um sich dann mit viel Energie und Budget am Ende über Change Requests, Terminverschiebungen oder Scope-Anpassungen – irgendwie gesichtswahrend – „durchzumogeln“. Daher geht es darum, die Risiken des Projektes und die damit verbundene Unsicherheit des Projektverlaufs mit jedem Projekttag zu verringern:

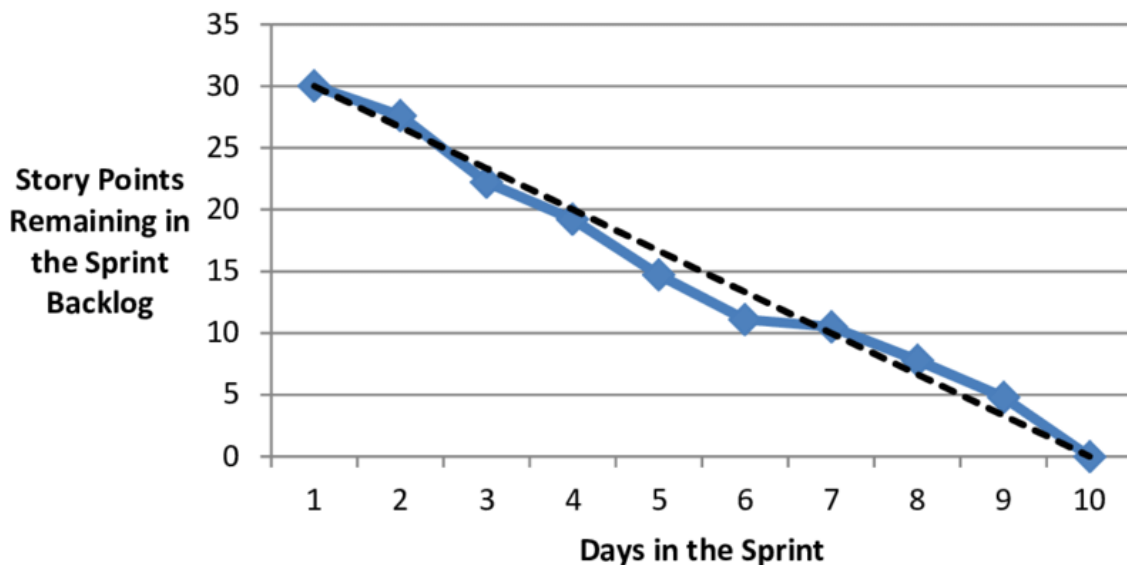
The Cone of Uncertainty in Project Management



“Fail often and fail early!”, ist das Prinzip. Natürlich geht es nicht darum, möglichst viele Fehler zu machen, sondern es geht darum, besser früh viele kleine, aber korrigierbare Fehler zu erkennen, diese auszumerzen und somit zu einem erfolgreichen Projektende zu gelangen, statt am Projektende einmal das Scheitern eingestehen zu müssen.

Wie zuvor geschildert, werden in SAP Activate die Anforderungen des Kunden in der Realize-Phase im Rahmen von Sprints umgesetzt. Ein Sprint beschreibt dabei einen genau definierten Zeitraum von wenigen Wochen. Dabei folgt ein Sprint immer unmittelbar auf den nächsten und alle Sprints sind stets gleich lang. Die ermittelte Velocity beschreibt in Story Points, die in einem fest definierten Zeitraum vom Team umgesetzt werden, somit eine Maßgröße, die den Projektpartnern nicht nur einen sehr guten Überblick über den Projektfortschritt verschafft, sondern die auch als Anhaltspunkt dafür dienen kann, was als außerordentlich gute und was als weniger gute Performance zu bewerten ist. So erhalten die Partner ein Bild davon, wie die geplante Velocity aussieht und wie im Vergleich dazu das Projekt tatsächlich vorankommt.

Sprint Burn-Down Chart



Diese Abweichungen können mit Penalties (Strafen) belegt werden, wenn eine bestimmte Velocity unterschritten wird oder auch mit Zusatzvergütungen, wenn die Erledigung der Aufgabenpakete, und damit der Story Points, deutlich über dem Durchschnitt liegt. So entsteht ein Bonus-/Malus-System, das beide Seiten gemeinschaftlich mit Risiken, aber auch mit Chancen ausstattet. Performt das Team sehr gut, so bekommt der Kunde eine größere Anzahl von Anforderungen früher als geplant umgesetzt, muss dafür aber tiefer in die Tasche greifen. Bleiben die Teams hinter den Erwartungen zurück, so wird die Vergütung für den Implementierungspartner gekürzt. Eine schlechte Performance führt zu Projektverzögerungen, aber eben auch zur Senkung der Projektkosten.


Trifft ein Kunde die Entscheidung für ein solches Bonus-Malus-System, ist es natürlich wichtig, dass die Partner seriös und nachvollziehbar miteinander umgehen. Das künstliche Verlangsamens der Velocity durch den Implementierungspartner in den ersten Sprints während der Ermittlung der Vergleichsgrößen ist natürlich unfair. Genauso unfair ist es aber auch, wenn der Kunde dann die Velocity später absichtlich verlangsamt, um Einsparungen zu erzielen. Ist das System damit zum Scheitern verurteilt? Nun, das Risiko des Scheiterns ist latent und permanent jedem Projekt innewohnend. Unfairer Umgang der Projektpartner lässt sich selbstverständlich über ein Vertragswerk ein Stück weit regeln, ist aber aus meiner Sicht ein schlechter Ansatz für ein partnerschaftlich durchgeführtes Projekt.

SAP Activate macht – sofern es gut aufgesetzt wird – Spaß und ist ein erfolgversprechendes Vorgehensmodell. Wie immer, liegen die Hürden an anderer Stelle. Missgunst und Misstrauen, schlechte Kommunikationskultur und mangelnder Teamgeist sind die Hauptursachen für das Scheitern von Projekten. Aber das war auch in der Vergangenheit schon so.

Wir haben in den vergangenen Jahren sehr gute Erfahrungen mit SAP Activate gemacht. Wenn Sie also einen Partner suchen, der Ihnen dabei hilft, dieses Vorgehensmodell zur Anwendung zu bringen, oder Sie ganz einfach Ihre Mitarbeiter im agilen Projektvorgehen in S/4HANA-Implementierungsprojekten oder in anderen Projekten fit machen möchten, dann nehmen Sie Kontakt zu uns auf! Wir bieten Ihnen:

- Agiles Vorgehen in SAP-Projekten (nach SAP Activate)
- Teilprojektleitungen und Gesamtprojektleitungen
- Projektbegleitendes Qualifizierungsmanagement (auch mit SAP Enable Now)

 info@sollistico.de

 +49 3328 338177 0

Rufen Sie doch einfach einmal bei mir oder meinem Team an und vereinbaren Sie einen vollkommen unverbindlichen Termin!

Fazit

Das Projektvorgehensmodell SAP Activate erfordert von allen Projektbeteiligten ein Umdenken. Sowohl der Projektverlauf als auch die Projektsteuerung erfolgen über andere Instrumente als bei wasserfallbasierten Vorgehensmodellen der Vergangenheit. Neu ist aber auch die Preisgestaltung, die unbedingt dem neuen Vorgehen angepasst werden sollte. Dadurch ist zwar das „gefühlte“ Projektrisiko auf beiden Seiten zu Beginn des Projektes groß, gleichzeitig nimmt aber das Risiko des Scheiterns mit dem Projektfortschritt ab. Anders als in der Vergangenheit wird man also tendenziell früher scheitern und nicht erst spät im Projekt. Das ungute Gefühl nimmt also ab und nicht zu – wie so oft in vergangenen Projekten.

Dieses Whitepaper wurde Ihnen von der sollistico GmbH bereitgestellt. Der Verfasser ist Martin Kipka. Weitere Informationen vom Autor zum gleichen Thema finden Sie auch hier:



BUCHEMPFEHLUNG

SAP® Activate

Agilität in SAP S/4HANA-Implementierungsprojekten

Autor: Martin Kipka | Veröffentlichung: 30.12.2020 | Herausgeber: Espresso Tutorials GmbH

Sichern auch Sie sich ihr Exemplar:

