

Whitepaper

SCRUM UND ERP – PASST DAS ZUSAMMEN?

Ein ERP-System ist nicht weniger als das Rückgrat eines Unternehmens. Als Abbild der aktuellen Geschäftsprozesse soll es Arbeitsabläufe unterstützen und dabei helfen, sie effizient zu halten. Funktional betrachtet bieten ERP-Systeme unzählige Möglichkeiten. Tatsache ist aber auch, dass sie zu den schwerfälligsten und gleichzeitig kostenintensivsten Architekturen in der IT-Landschaft eines Unternehmens zählen. Will oder muss sich ein Unternehmen den Veränderungen am Markt anpassen, bewegen sich die weit verzweigten ERP-Riesen oft nur langsam: Wird an einer Schraube gedreht, drehen sich unzählige andere mit – Auswirkung ungewiss.

In der Pflicht sind CIOs und IT-Verantwortliche: Eingespannt zwischen Kostenkürzungen, Wachstumsplänen und Innovationsdruck sollen sie ERP-Systeme zu essenziellen Wertschöpfungsfaktoren machen. Viele IT-Verantwortliche sind von agilen Management-Frameworks überzeugt, stellen sich aber die Frage, ob sich damit auch die komplexen Anpassungen von ERP-Systemen bewältigen lassen. Die einfache Antwort lautet: Agile Management-Frameworks und speziell Scrum wurden genau deshalb entwickelt, um große und komplexe Projekte erfolgreich durchführen zu können.



DIE DREI STOLPERSTEINE IN ERP-PROJEKTEN

Es sind nicht nur direkt zurechenbare Kosten, die bei Implementierungen oder Modifikationen eines ERP-Systems entstehen. Jede Veränderung an einem ERP-System ist eine Störung – und das verursacht Kosten. Diese Störung kann sich temporär auf die Phase der Veränderung beschränken (z.B. wenn Mitarbeiter von anderen wichtigen Aufgaben abgehalten werden), schlimmstenfalls wirkt sie aber darüber hinaus – wenn das Ergebnis der Implementierung oder der Anpassung des Systems so gar nicht zur Realität der Nutzer passt. Je öfter sich die Prozesse in einem Unternehmen ändern, desto kostspieliger wird das ERP-System. Welches Ausmaß diese Kosten annehmen, hängt von technischen, menschlichen und organisatorischen Faktoren ab.

Technische Fallen

Ein ERP-System sollte sich dem Business anpassen, nicht das Business dem System. Daher ist es wichtig, die zukünftigen Nutzer so früh wie möglich zu involvieren, um nicht am gelebten Prozess vorbei zu entwickeln. Viele Projektbeteiligte sehen das genauso, trotzdem begegnen uns in der Praxis oft drei Extreme:

- 1. Eine für alle.** Die gleiche – und am einfachsten umzusetzende – Lösung wird allen betroffenen Fachbereichen und Abteilungen übergestülpt, die Realität der Arbeitsabläufe bleibt unberücksichtigt. Ergebnis: Die ERP-Lösung wird zum teuren, aber de facto unnützen Spielzeug.
- 2. Alles für einen.** (Alle) Wünsche der Fachabteilungen werden berücksichtigt, hier und dort wird ganz im Vertrauen mal eine Kleinigkeit geändert – plötzlich funktioniert gar nichts mehr.
- 3. Alles für alle.** Weil es für Standardprodukte unzählige attraktive Add-ons von verschiedenen Anbietern gibt, wird unkoordiniert immer wieder etwas Neues eingeschraubt. Die langfristige Konsequenz: Ein technischer Moloch, der die Kosten explodieren lässt und das System destabilisiert.

Um die Kosten im Griff zu behalten, müssten bei einem so sensiblen Konstrukt die Modifikationen so früh wie möglich integriert werden. Nur so lassen sich die Auswirkungen rechtzeitig erkennen. Dazu ist es aber auch notwendig, den Überblick über sämtliche Änderungen am System zu behalten und Modifikationen zu priorisieren.

Organisatorische Fallen

Je nach Unternehmensgröße kann ein ERP-Projekt inklusive Vorbereitungen mehr als ein Jahr dauern – oder es wird zu einer ressourcenintensiven Dauerbaustelle. Eine Ursache: Werden solche Projekte nach traditionellen Vorgehensmodellen abgewickelt, finden erst am Ende aufwändige Integrations- und Abnahmetests statt. Am Ende der Entwicklung türmen sich also die erkannten Fehler und Änderungswünsche und die Lieferung verzögert sich. Wieder entstehen Kosten.

Der menschliche Aspekt

Die Umstellung auf ein ERP-System bzw. die Modifikation von Modulen fordert immer auch die Veränderungsbereitschaft der betroffenen Mitarbeiter. In den meisten Fällen können Kollegen aus den Fachabteilungen anhand der technischen Spezifikationen gar nicht einschätzen, wie sich die vorgeschlagene Lösung auf ihre Arbeit auswirken wird. Im schlechtesten Fall müssen Mitarbeiter von einem Tag auf den anderen mit Software arbeiten, die ihre bisherigen Abläufe völlig über den Haufen wirft – Widerstand ist vorprogrammiert. Umgekehrt verstehen viele Mitarbeiter nicht, welchen Effekt die eigenen, partikularen Wünsche auf das Funktionieren des Ganzen haben. Und so wird politisch versucht, den Willen durchzusetzen.



Die Implementierung oder Adaptierung von ERP-Systemen sollte daher in einem Rahmen stattfinden, der

- ▶ die **Kommunikation und Koordination** zwischen IT-Abteilung, Fachabteilungen, den betroffenen Mitarbeitern und ggf. externen Beratern im Sinne eines gemeinsamen Ziels fördert.
- ▶ **Zwischenprodukte** sichtbar macht. Ob Programmierung oder Customizing: es geht um das Schaffen neuer Funktionalität. Je früher der Kunde bzw. User Ergebnisse sieht und ausprobieren kann, desto brauchbarer wird das Ergebnis durch sein Feedback. Und nicht nur das: Menschen akzeptieren neue Lösungen eher, wenn sie aktiv daran mitwirken und in Entscheidungen eingebunden werden.
- ▶ **Qualität sichert** und damit **Kosten spart**. Ein iterativer Kreislauf aus Programmierung und Test verhindert, dass sich die am Ende des Projekts die entdeckten Fehler aufürmen und kaum noch zu bewältigen sind.

WIE FUNKTIONIERT SCRUM?

Scrum ist ein Management-Rahmenwerk für die iterative und inkrementelle – „agile“ – Entwicklung: Dabei wird ein Produkt in Zyklen von der Vision weg schrittweise verfeinert. Man weiß, was man will, akzeptiert aber die Tatsache, das Produkt nicht im Vorhinein bis ins letzte Detail spezifizieren zu können.

Um Komplexität zu reduzieren, arbeitet ein Scrum-Team in je maximal zwei- bis vierwöchigen Einzelschritten, „Sprints“ genannt. Zu Beginn werden die wesentlichen Funktionalitäten festgelegt und in jedem Sprint auf Basis von Feedback des Kunden bzw. des Anwenders weiterentwickelt oder verworfen. Anders als im klassischen Projektmanagement sieht der Kunde das Ergebnis also nicht erst am Ende der gesamten Entwicklung, sondern ist in jeden Entwicklungsschritt eingebunden. Die Organisationsprinzipien von Scrum:

- ▶ **Kleine, selbstorganisierte und -verantwortliche, cross-funktionale Teams** – bestehend aus ScrumMaster, Product Owner und Entwicklungsteam
- ▶ **Arbeiten nach dem Pull-Prinzip:** Das Entwicklungsteam entscheidet über die Menge der Funktionalitäten (die vom Product Owner nach dem Business Value priorisiert wurden), die es in einem Sprint entwickelt und es entscheidet, wie es diese Aufgaben erledigt.
- ▶ **Klar begrenzte zeitliche Intervalle (Timebox):** Ziel ist es, die gewählten Funktionalitäten tatsächlich innerhalb des Sprints abzuarbeiten.
- ▶ **Nutzbare Business-Funktionalität:** Am Ende jedes Sprints erbringt das Team eine Lieferung, die den Standards, Richtlinien und Vorgaben des Projekts entspricht.

Wesentlicher Erfolgsfaktor ist die intensive Kommunikation: Die Teammitglieder stimmen sich in täglichen Kurzmeetings (Daily Scrum) über den aktuellen Stand der Arbeit ab. In Estimation Meetings (Backlog Grooming) arbeiten sie am gemeinsamen Verständnis über die zu entwickelnden Funktionalitäten und in den Sprint Plannings 1+2 wird konkretisiert, was wann und wie entwickelt wird. In den Reviews am Ende eines Sprints spricht das Team mit dem Kunden über die entwickelten Produktinkremente und in der Retrospektive hinterfragt das Team seine Arbeit, um sie ggf. zu verbessern.



AGILE ANTWORTEN AUF DIE HÄUFIGSTEN BEDENKEN IM ERP-UMFELD

Status Quo Agile 2014 ¹
www.status-quo-agile.de

Die eingangs erwähnten Herausforderungen betreffen ERP-Projekte, die wir selbst erlebt haben oder die uns von Kunden und Kollegen geschildert wurden. Agile Management-Frameworks versprechen, diese Herausforderungen zu lösen – dennoch ist gerade der ERP-Bereich nicht so stark von agilen Methoden durchdrungen wie die klassische Softwareentwicklung.¹ Die Ursache dafür erkennen wir in der Beratungspraxis sehr deutlich: Nach wie vor hält sich hartnäckig die Meinung, dass ERP-Systeme wegen ihrer Sonderstellung im Unternehmen nicht Gegenstand agiler Entwicklung sein könnten. Auf die häufigsten Vorbehalte geben wir hier Antworten aus der agilen Perspektive.

„Wieso sollten bei standardmäßigen ERP-Einführungen agile Methoden verwendet werden? Es wird doch meistens nichts entwickelt, sondern die Software nur an die bestehenden Prozesse angepasst.“

ERP-Einführungen greifen in die Arbeitsweise vieler Mitarbeiter ein, und ob Standard oder eigene Entwicklung: Veränderungen bewirken Unsicherheit. Seien Sie sich bewusst, dass die Mitarbeiter bisher eine gewisse Arbeitsweise gewohnt waren – und diese Arbeitsweise wird nun möglicherweise verworfen oder stark verändert. Wundern Sie sich also nicht, wenn Sie auf – aktiven oder passiven – Widerstand treffen.

Die Verantwortlichen für eine ERP-Einführung sollten also rücksichtsvoll mit den Ängsten, Anliegen und Arbeitsweisen der Betroffenen umgehen. Die Ängste vor dem Neuen werden kleiner oder verschwinden ganz, wenn Mitarbeiter ab dem ersten Tag Teil des Projekts sind und mitbestimmen können, wie der Weg zum Ziel aussehen soll. In agilen Management-Frameworks wie Scrum bekommt die Kommunikation mit dem Anwender durch regelmäßige Reviews viel Raum. Die Anwender probieren die Produktinkremente aus und melden zurück, was bereits gut passt und was nicht.

„In ERP-Systemen kann nur schwer agil entwickelt werden, da alle Anforderungen am Anfang bekannt sein müssen. Darauf aufbauend können das Datenmodell und das Fachkonzept erstellt werden.“

Im Grunde ist es doch so: Konzepte sollen beruhigen. Mit einer festgelegten Wegbeschreibung fühlen sich vor allem Projektmanager sicherer. Kunden oder User winken diese Konzepte hingegen oft nur durch, ohne sie auch nur ansatzweise verstanden zu haben. Der Grund: Überforderung. Ein Dokument mit Datenmodellen und bestenfalls grafischen Vorschlägen für das User Interface liefert dem zukünftigen Anwender keine Anhaltspunkte dafür, ob das Produkt tatsächlich seine Arbeit unterstützen wird. Zusätzlich haben User meistens eine konkrete Vorstellung von dem, was sie gerne hätten – was aber nicht immer das ist, was sie tatsächlich brauchen. Was ihm wirklich hilft, erkennt ein Anwender nur durch Ausprobieren.

In der agilen Entwicklung probiert der User regelmäßig aus, weil regelmäßig neue Teile des Produkts geliefert werden. Die Erfahrungen des Users beim Ausprobieren helfen den Entwicklern wiederum, das schlussendlich „richtige“ Produkt zu liefern. Statt in den engen – und möglicherweise unpassenden – Vorgaben eines unumstößlichen Konzepts stecken zu bleiben, fließt die Realität in die weitere Entwicklung ein. Das kontinuierliche Evaluieren hat aber noch einen weiteren Vorteil: Anpassungen am System passieren so effizient wie möglich. So wird auf diese Art und Weise deutlich, ob das reine Customizing der Standardmodule schon genügt, oder ob selbst geschriebener Code (z.B. in ABAP) den Geschäftsprozess besser abbilden würde. Wenn die Anwender bereits in der ersten Iteration den Standard ausprobieren können, werden teure Adaptionen oft obsolet.



„Von kleinen Anpassungen abgesehen kann eine End-to-End-Funktionalität selten in zwei Wochen entwickelt werden. Moderne Geschäftsprozesse sind zu komplex, um in so kurzer Zeit etwas Sichtbares entwickeln zu können.“

Robinson, Frank: ²
Minimum Viable Product
<http://bit.ly/1AiAo5Q>

In der agilen Softwareentwicklung werden Anforderungen in möglichst kleine Stücke geteilt. Diese Teile bieten einen Mehrwert für den Kunden und können in einem Sprint fertiggestellt werden. Nach diesem Grundprinzip kann auch in ERP-Projekten ein sogenanntes Minimum Viable Product konstruiert werden²: zum Beispiel ein Kernprozess, eine Basisfunktionalität, eine Haupttransaktion oder schlicht der „80%-Fall“ des operativen Geschäfts.

Dieses Minimum Viable Product kann meist mit relativ wenig Aufwand entwickelt werden und daher innerhalb weniger Tage bis Wochen auch End-to-End implementiert werden. Im weiteren Verlauf wird Schritt für Schritt der Prozess modelliert und verfeinert, um den Anwender so zu unterstützen, wie er es braucht. Auch bei der Implementierung von Schnittstellen zwischen verschiedenen Systemen oder ERP-Modulen kann mit einem Minimalset an Datenobjekten schnell ein erster Durchstich erzielt werden. In den folgenden Sprints wird dann die vollständige Datenstruktur der Schnittstelle entworfen und implementiert. Wichtig ist in allen Varianten: Die echten Anwender verifizieren den gewünschten Prozess so schnell wie möglich. Darauf basierend wird er kontinuierlich angepasst.

„In ERP-Systemen ist es nur beschränkt möglich, alle zwei Wochen fertige Software mit zusätzlicher Funktionalität zu liefern und produktiv zu setzen. Es besteht die Gefahr, dass bestehende Funktionalität nicht mehr einwandfrei funktioniert und operative Prozesse blockiert werden.“

IDC: Maintaining ERP Systems – ³
The Cost of Change
<http://bit.ly/1AiB4lq>

Unternehmen stehen heute oft vor dem Problem eines durch unzählige und manchmal wahllose Anforderungen gewachsenen, instabilen Systemgefüges, das die Kosten für die Wartung in die Höhe treibt.³ Auch bei der agilen Entwicklung im ERP-Umfeld sind die passenden Entwicklungspraktiken das Um und Auf. Es entsteht kein System-Moloch, wenn man konsequent auf die Qualität des Quellcodes achtet. Mit automatisierten Tests kann kontinuierlich ausgeliefert und gleichzeitig die bestehende Funktionalität abgesichert werden. Für diese, nach jeder Änderung stattfindenden Tests werden im agilen Umfeld Continuous Integration und Delivery Systeme eingesetzt. Intelligente und optimierte Entwicklungsumgebungen (wie ABAP Eclipse Unterstützung), Testautomatisationsuites (eCATT und CBTA in SAP), Versions-/Buildverwaltungen und Mocking Frameworks (MockA) sind auch im ERP-Bereich auf dem Vormarsch und unterstützen die Business-Anforderungen einer laufend produktiv gesetzten Software.

„Wie können agile Methoden einen Überblick über die zahlreichen kleinen Anforderungen der vielen verschiedenen Fachbereiche schaffen? Reichen nicht einfache To-Do- und Abarbeitungslisten?“

Transparenz und Visualisierung gehören zu den wichtigsten Hebeln der agilen Softwareentwicklung. Für das Anforderungsmanagement – gerade in komplexen Bereichen wie ERP-Systemen – ist das eine enorme Hilfe: Die Anforderungen werden in einem Product-, Company- und/oder Team-Backlog klar dargestellt und so behalten alle Verantwortlichen den Überblick. Sie können sich auf Basis der geforderten Funktionalitäten austauschen, sich auf eine übergreifende Priorisierung einigen, doppelgleisige Implementierungen frühzeitig erkennen und die damit verbundenen Interessenkonflikte auflösen.



Noch wichtiger und wirksamer ist diese visuelle Unterstützung in großen Projekten, in denen viele verschiedene Fachbereiche in die abzubildenden Geschäftsprozesse einbezogen werden müssen. Verfolgt jeder Fachbereich seine eigenen Interessen, kann das schnell zu unkontrollierten Änderungen führen – der Scope wächst kontinuierlich an. Wenn hingegen Product Owner die Backlogs verantworten und priorisieren, werden die wirklich wichtigen Interessen der verschiedenen Stakeholder gezielt berücksichtigt. Es wird Funktionalität für Funktionalität implementiert und sofort produktiv gesetzt, um dem Kunden Mehrwert zu liefern und Feedback zu erhalten.

„Um die vielfältigen und komplexen ERP-Komponenten konfigurieren und anpassen zu können, brauchen wir spezialisierte Experten. Das erschwert die Arbeit in cross-funktionalen Teams.“

Auch in ERP-Projekten hat es sich bewährt, unterschiedliche Skills in einem Team zu vereinen: ERP-Berater arbeiten mit ERP-Entwicklern, CRM-Experten mit MM-Experten, Business Analysten oder System-Architekten bereichern solche Teams ebenfalls. Die Anforderungen werden auf diese Weise aus verschiedenen Perspektiven betrachtet und der Austausch im Team fördert Aspekte zutage, die der Einzelne nicht entdeckt hätte – das stellt also wieder sicher, dass das „richtige“ Produkt geliefert wird. Intensive Zusammenarbeit – etwa durch Pair Programming – fördert außerdem den Wissensaustausch im Team und im Unternehmen. Mitarbeiter entwickeln dadurch ein T-Shape-Profil: Sie besitzen ihr Spezialwissen, erweitern es aber um ein breites Basiswissen in anderen Disziplinen. Der/die Einzelne entwickelt sich persönlich weiter und kann bei vielfältigeren Aufgaben unterstützen.

3 gute Gründe, warum Scrum und ERP zusammenpassen

1. Treffsichere Erfüllung der Kundenbedürfnisse

Durch das kontinuierliche Feedback wird im Laufe des Projekts die für den Anwender richtige Software entwickelt. Je besser das System zum Geschäftsprozess passt, desto besser können Marktchancen genutzt werden.

2. Schnelle Lieferung

Mit Scrum und agilen Entwicklungspraktiken können komplexe Anforderungen in kurzen Zeitintervallen implementiert werden. Das Projekt wird in seinen Lieferungen für alle Beteiligten – vom Entwicklungsteam bis zum Management – vollkommen transparent. Änderungen fließen während der Projektlaufzeit ein, und die Störungen halten sich für alle Beteiligten in Grenzen.

3. Motivierte Mitarbeiter

Cross-funktionale Teams helfen Mitarbeitern, sich außerhalb ihres Spezialwissens zu entwickeln und mit anderen Experten an Lösungen zu arbeiten. Das gemeinsame Ziel fördert die Zusammenarbeit, die deutlich geringeren Handover sparen wertvolle Zeit.

Aus der Praxis: Systemübergreifendes SAP-Projekt mit cross-funktionalen Teams

Wie können komplexe Geschäftsprozesse über mehrere SAP-Systeme hinweg modelliert und gleichzeitig unzählige Fachbereiche in die Modellierung eingebunden werden? Diese Frage beschäftigte den Kunden in einem unserer Projekte. Ein operativer Geschäftsprozess, der von mehreren hundert Mitarbeitern unterschiedlichster Fachabteilungen genutzt wird, musste über zwei SAP-Systeme, verschiedene Module und eine Drittsoftware hinweg abgebildet werden.



Alles Wissen im Team. Um diese Ausgangssituation optimal zu adressieren, wurden zwei eng gekoppelte, cross-funktionale Scrum-Teams aufgesetzt. Jedes Team umfasste sowohl ERP-Entwickler als auch SAP-Berater beider SAP-Systeme, jeweils einen Vertreter des Herstellers der Drittsoftware und einen Tester mit Know-how in eCATT. So konnte jedes Scrum-Team funktionale User Stories end-to-end implementieren und mit automatisierten Tests absichern.

Frühe Integration. Mit dem Fokus auf das Minimum Viable Product wurde die frühzeitige Integration aller Systeme möglich. Bereits im ersten Sprint wurde die Schnittstelle zwischen den beiden SAP-Systemen mit einem Minimum-Set an Daten implementiert. In den kommenden Sprints wurde die Schnittstelle Schritt für Schritt erweitert und die Drittsoftware eingebunden.

Kommunikation. Um die vielen, vor allem operativ betroffenen Personen einzubeziehen, gab es für das Management der Fachbereiche regelmäßige Austauschtermine in Form von Backlog Groomings mit den Product Ownern der beiden Teams. Um die Anwender zu integrieren, wurden „Key-User-Gruppen“ gebildet, die einen repräsentativen Querschnitt der Anwender abdeckten. Die Key-User nahmen nicht nur an den Sprint Reviews teil, sondern arbeiteten während der Sprints auch regelmäßig mit den Scrum-Teams zusammen.

Die Ergebnisse. Der Budget- und Zeitrahmen wurde in diesem Projekt perfekt eingehalten. Aber noch viel wichtiger: Durch das iterative Vorgehen und Ausprobieren erkannten alle Beteiligten, was sie tatsächlich brauchten. Viele ursprünglich genannte Anforderungen verschwanden völlig von der „Wunschliste“ oder wurden durch Neues ersetzt. Durch das frühzeitige Integrieren der Module sank außerdem das Risiko einer inkompatiblen Schnittstelle drastisch. Der für das Projekt verantwortliche Fachbereich war so überrascht und begeistert von den schnellen und „richtigen“ Resultaten, dass er in Zukunft alle seine beauftragten Projekte mit Scrum durchführen will.

DARAUF SOLLTEN SIE BEIM START MIT AGILEN METHODEN ACHTEN

Viele Unternehmen konzentrieren sich bei der Einführung von Scrum auf die bloße Implementierung des „Scrum-Handwerks“ in den Entwicklungsteams. Wenn Sie dauerhaft mit Scrum erfolgreich sein wollen, brauchen Sie eine breitere Perspektive. Behalten Sie bei der Einführung von Scrum daher vor allem die folgenden drei Punkte im Blick:

- ◆ **Einbeziehen des Umfelds.** Scrum ist nicht nur eine Sache des Entwicklungsteams. „Agilität“ ist kein Produkt, sondern eine Einstellung, die von allen mitgetragen werden muss. Daher sollten auch das Management und die Kunden (in Form der Fachbereiche) beim Verinnerlichen von agilen Methoden und Werten begleitet werden. Nur so kann eine konstruktive Atmosphäre für das Entwickeln von Produkten und Lösungen entstehen.
- ◆ **Moderne Entwicklungspraktiken.** Testgetriebene Entwicklung und Continuous Integration sind heute unerlässlich, um die Komplexität von Systemen in kleinen Iterationen zu bewältigen. Entwicklungsteams brauchen dafür die nötigen technischen Infrastrukturen und Möglichkeiten – und vor allem das Wissen.
- ◆ **Selbstorganisation braucht Führung.** Agilität ist ein Wechselspiel aus rahmengebender Führung und Selbstorganisation. Laterale Führungskräfte wie ScrumMaster, aber auch Product Owner und Manager brauchen Unterstützung, um ein neues Selbstverständnis entwickeln zu können. Und sie müssen wissen, wie sie mit sozialen Dynamiken am besten umgehen.

Der Markt für die Begleitung von Organisationen bei der Einführung von agilen Methoden ist noch sehr jung. Er wächst aber kontinuierlich, da sich die Vorteile weit über die Softwareentwicklung hinaus herumgesprochen haben. Im deutschsprachigen Raum gibt es daher erst wenige spezialisierte Anbieter, die sowohl auf Teamebene als auch in skalierten Umgebungen mehr als fünf Jahre profunde Erfahrung mit agilen Methoden vorweisen können. Noch weniger Erfahrung gibt es im ERP-Umfeld, da sich agile Management-Frameworks in diesem Bereich gerade erst durchsetzen. Seien Sie daher kritisch bei der Wahl Ihrer Partner für diesen Weg.

WIE ARBEITET BORIS GLOGER CONSULTING?

Agile Produktentwicklung kann Ihr Unternehmen und dessen ERP-Systemlandschaft für die Dynamik des Marktes bestens rüsten. Unser Ziel ist es daher, uns überflüssig machen: Sie selbst und Ihre Mitarbeiter müssen Agilität leben – das können wir Ihnen nicht abnehmen. Aber wir bauen dafür das nötige Fundament: Von Anfang an schaffen wir unter allen Beteiligten – von den Entwicklungsteams bis zum Management – mit einfachen Bildern, Methoden und Verfahren Klarheit und ein gemeinsames Verständnis dafür, was agile Methoden für die Organisation bedeuten. Auf diesem gemeinsamen Verständnis etablieren wir Pilotgruppen, die entscheiden, wie sie mit den für die Implementierung notwendigen Aufgaben beginnen und wie sie diese lösen wollen. Dazu gehört auch das Bilden eines ersten Scrum-Teams: Dieses Team durchläuft einige Sprints und erkennt dabei, was wir gemeinsam noch verbessern können.

Das Management beziehen wir von Anfang an ein und es wirkt aktiv mit. Es trifft notwendige Entscheidungen und lernt selbst, was es bedeutet, mit agilen Methoden zu arbeiten. Diese Einbindung ist uns wichtig, da sich das Vertrauen in die neuen Methoden auf allen Ebenen bilden muss. Schrittweise binden wir während der Arbeit mit agilen Methoden die passenden Entwicklungspraktiken ein, wir integrieren Ihre externen Partner und die gesamte Organisation beginnt, sich mit dem neuen Arbeiten zu beschäftigen.

Das ist **ein** möglicher Weg. Den genau für Ihr Unternehmen sinnvollen Weg entwerfen wir gemeinsam – passend zu Ihrer Ausgangssituation und Ihren Zielen. Während des gesamten Umsetzungsprozesses sind wir für Sie da.

Reden wir über Ihre Herausforderung!

Gegenseitiges Vertrauen ist für uns die unbedingte Voraussetzung in einer Zusammenarbeit. In einem unverbindlichen Gespräch zeigen wir Ihnen, wer wir sind, wie wir arbeiten und was wir für Sie tun können. Rufen Sie uns einfach an oder schreiben Sie uns!

borisgloger consulting GmbH
Lichtentaler Straße 7
D-76530 Baden-Baden
T +49 (0) 72 21.398 737-0
M +49 (0) 151.54 40 12 61
F +49 (0) 72 21.398 737-10
office@borisgloger.com
www.borisgloger.com



Über die borisgloger consulting GmbH

Die borisgloger consulting GmbH mit Sitz in Baden-Baden und Wien wurde 2008 gegründet, beschäftigt 20 Mitarbeiter und gehört in der DACH-Region zu den führenden Managementberatungen im Bereich des agilen Change-Managements und der agilen Produktentwicklung. Der Schwerpunkt liegt auf dem Management-Framework Scrum. borisgloger consulting bietet darüber hinaus Training und Consulting für Fach- und Führungskräfte im Bereich des agilen Managements an. Zu den Kunden zählen unter anderem die Scout-Gruppe, Roche PVT, otto.de, Deutsche Post und die Ergo Direkt Versicherung.

Mehr Informationen unter www.borisgloger.com

