

Erfahrungsbericht Agile/Scrum in ERP-Projekten

Dirk Blaurock
Dirk Blaurock IT Consulting
Pinneberg

Schlüsselworte

ERP, Oracle E-Business Suite, Agile, Scrum, Einführung

Einleitung

Das Thema Agilität ist schon seit einigen Jahren gängige Praxis. Nicht nur in der Softwareentwicklung finden Agile Methoden Anwendung, sondern auch in der normalen Produktentwicklung, wo häufig das Ergebnis innovativ ist. Kernelement agiler Methoden ist das Lösen von klassischen Vorgehensmodellen über eine Idee, Design, Detaillierung und Umsetzung. Ideen werden in selbstorientierten Cross funktionalen Teams eigenständig bearbeitet und in kurzen Zeiträumen umgesetzt. Dabei ist häufig nicht Ergebnis häufig nicht die 100% Lösung, sondern das erstellte Produkt wird immer weiter verfeinert. Das wesentliche dieser Methode ist das zielgerichtet arbeiten innerhalb von Teams. Wie gestaltet sich dieses aber bei der Einführung bzw. Weiterpflege eines Enterprise-Resource-Planning System wie die Oracle E-Business Suite. Aufgrund der bereichsübergreifenden Prozesse innerhalb eines ERP System kann zwar nicht 1:1 der klassische Agile Ansatz wie Scrum angewendet werden, er kann aber hilfreich sein, komplexe ERP Projekte in kleiner Bausteine zu zerlegen und diese zielgerichtet umzusetzen. Es gibt Besonderheiten zu beachten, vor allem in Hinsicht das es bei ERP Projekten nicht auf einzelne kleine Features ankommt sondern auf das Zusammenspiel gesamtheitlicher Prozesse inklusiv externe Einflüsse wie gesetzliche oder von externen Partnern (Kunden, Lieferanten),

Agile Methode Scrum

Bei der agilen Methode Scrum erfolgt die Umsetzung von Features für ein „Produkt“ in kleinen Iterationen, Sprints genannt. Ein Sprint sollte eine Länge von 2 bis maximal 4 Wochen haben. Ausgangslage des Sprint's ist ein definierter Scope, welcher zur Umsetzung durch ein selbstorientiertes Team in dem Sprint vorgesehen ist. Das Team wählt aus einer Art Wunschliste (= Produkt Backlog) die Arbeitspakete (=Items) aus, welche es meint mit den bestehenden Teamkapazitäten abzuwickeln zu können. Das Backlog wird durch einen sogenannten Produktowner verwaltet, welcher zudem die verschiedenen Items auch priorisiert. Das Team arbeitet nun während eines Sprints gemeinschaftlich an den Themen und setzt diese um, weitgehend durch parallele Arbeit. So werden schon in einem Design Prototypen erstellt, der Tester macht sich Gedanken um die Testfälle, Testdaten werden schon vorbereitet, letztendlich alles mit dem Ziel in kurzer Zeit Ergebnisse zu erzielen. Das Kernziel von Scrum ist, lieber viele kleine Schritte in kurzer Zeit als einen großen Schritt in langer Zeit.

Nach der Abwicklung eines Sprints werden Items, welche nicht vollständig umgesetzt sind, wieder in das Backlog aufgenommen. Items welche umgesetzt sind, werden in das „Produkt“ übernommen. Anschließend startet der neue Sprint.

Agile und Scrum – Sprints

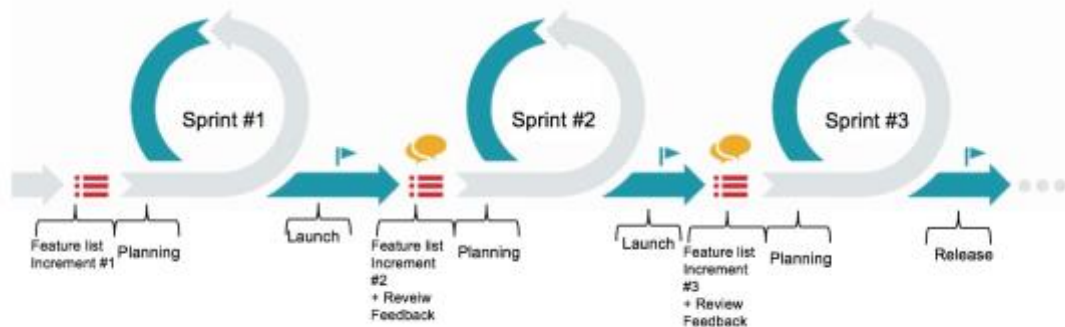


Abb. 1: Sprints beim Scrum

ERP Projekte

Bei ERP Projekten muss man unterscheiden zwischen der Einführung und der Weiterpflege des ERP System. Die Weiterpflege eines ERP System kann in der Regel sehr gut mittels Scrum umgesetzt werden, ist hier aufgrund der schnelleren Time to Delivery auch anzuraten. Die Teams sind meist überschaubare und es werden häufig Add Ons wie ein Report oder Konfigurationsänderungen vorgenommen. Aufgrund Scrum hat man hier eine gute Priorisierung und durch die kleineren Anforderungen auch die Chance, etwas in 2-4 Wochen umzusetzen.

Anders sieht dieses jedoch in komplexen ERP Einführungen oder einer Migration mit Strukturveränderungen aus. Hier gibt es eine Vielzahl von Stakholder, hier wird sich ein Unternehmen erst mal schwermachen, einen zentralen Produktowner dafür zu etablieren.

ERP: Einflüsse

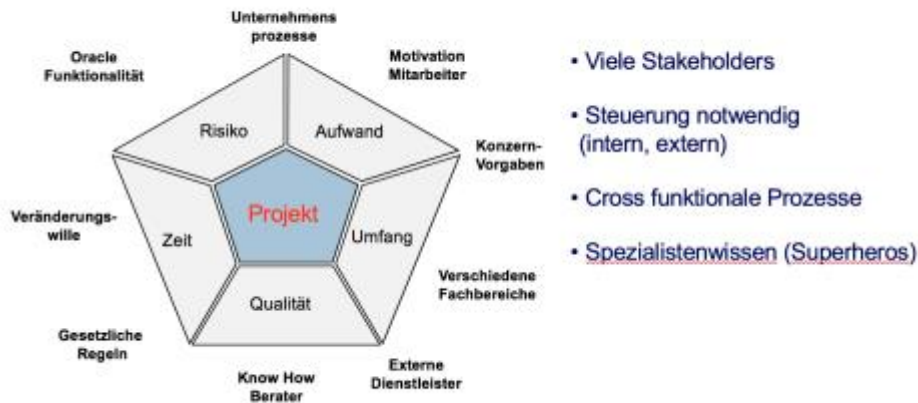


Abb. 2: Sprints beim Scrum

Auch wird häufig Expertenwissen (Superheros) in solchen Projekten benötigt, welche meist von externen Dienstleistern kommen. Das Team wird für eine Zeitlang durch außenstehenden Ressourcen aufgestockt. Und letztendlich sind die Kennzeichen von solchen Projekten, dass umfangreiche Geschäftsprozesse quer durch das Unternehmen etabliert werden, welche nicht mal eben E2E in 2-4 Wochen umgesetzt werden können.

ERP Projekte werden meistens noch im klassischen Wasserfallmodell umgesetzt.

ERP: "Wasserfall" Modell

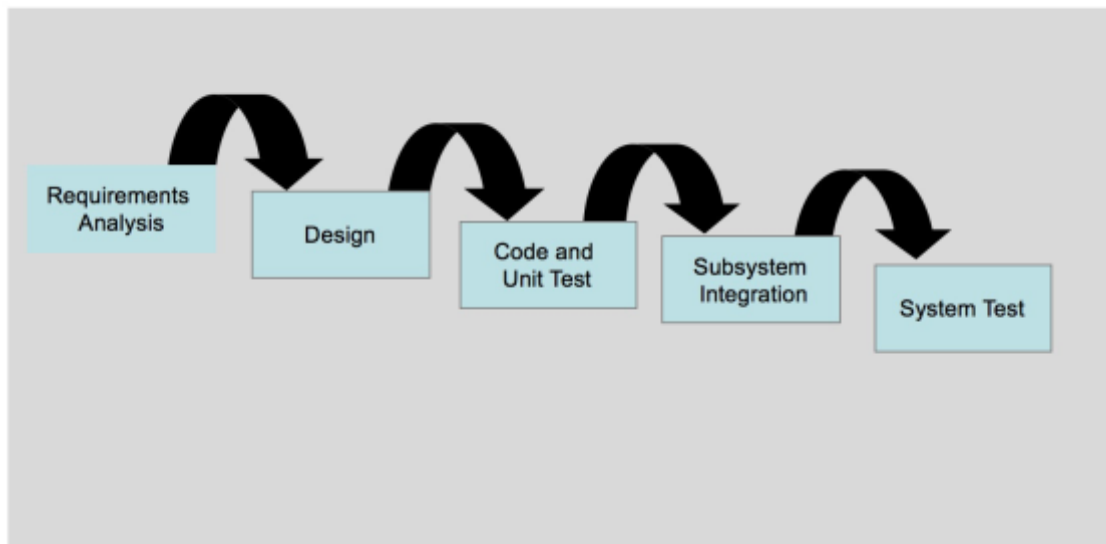


Abb. 3: ERP Wasserfall Modell

Der Requirmentaufnahme folgt dann ein Detail Design, worauf aufbauend die Konfiguration und das Coding durchgeführt wird. Über einen Integrationstest und dem Systemtest erfolgt dann letztendlich die Produktivnahme. Aufgrund dieser aufeinander aufbauenden Arbeitspakete können so sehr gut Prozesse über alle Unternehmensbereiche dokumentiert und umgesetzt werden, jedoch kann auf Veränderungen sehr schwer eingegangen werden. Dieses ist meistens im ERP Bereich nicht so kritisch, da Veränderungen dort nichts so dynamisch sind. Jedoch ist das Kritische, dass es greifbares Feedback von den Stakeholdern erst recht spät gibt. Daraus ergeben sich die in ERP Projekten gängige Change Request Listen über welche noch kurz vor einen Go Live Termin ausgiebig diskutiert wird.

Um diesem entgegenzuwirken werden teilweise ERP Projekte in kleiner Iterationen durchgeführt, um in kürzerer Zeit (vielleicht 3 Monate) schon ein erstes Feedback der Stakeholder zu erhalten.

ERP: "Iteratives" Modell

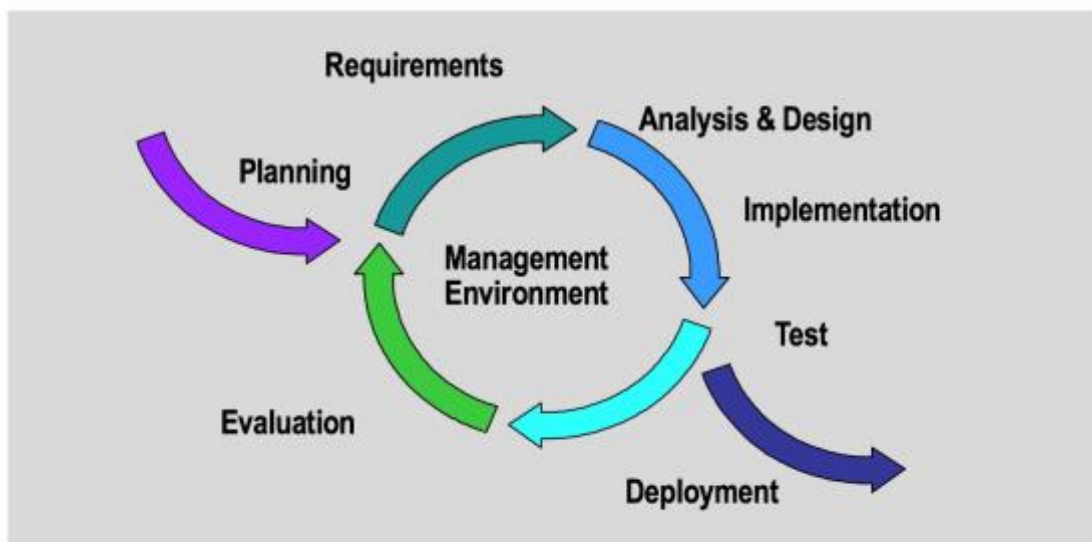


Abb. 4: ERP Iteratives Model

Aber auch dieses Modell verhindert nicht das ERP Projekte umfangreiche Ressourcen bindet, ein großer Fokus auf die Dokumentation gelegt wird und das Projekt sehr abstrakt ist. Daraus folgendes die in solchen Projekten häufig anzufinden Missverständnisse.

ERP Projekte im Agile Mode

Bekanntlich gibt es in ERP Projekten eine Vielzahl von Problemen. Nachfolgend sind die Kernprobleme in ERP Projekten aufgelistet und wo Agile Methoden wie Scrum helfen können.

Probleme im Projekt	Ursache	Scrum kann helfen
Zeitplan und Budget werden nicht eingehalten	<ul style="list-style-type: none"> ■ Schlechte Planungsqualität ■ Ungenaue Verträge ■ Ungenaue Vorbereitungsphase 	Nein
Inflation der "Changes"	<ul style="list-style-type: none"> ■ Vorgaben waren nicht fixiert ■ Kein Veränderungswille bei den Anwendern ■ Falsches Change Management 	teilweise
Unterschiedliche Auslegung der Leistungsinhalte und Mitwirkungspflichten	<ul style="list-style-type: none"> ■ Rollen nicht klar definiert oder abgegrenzt ■ Versprochen Mitwirkungsleistung nicht machbar 	Ja
Fehlender Durchblick bei Projektfortschritt und Kostenentwicklung	<ul style="list-style-type: none"> ■ Transparenz fehlt (wo stehen wir) ■ Steuerung des Dienstleisters mangelhaft 	teilweise
Die Projektkultur lässt zu wünschen übrig	<ul style="list-style-type: none"> ■ Kein Teamgedanke ■ Kein Leben von Soft Skills 	Ja
Sitzungen fallen aus / Mitarbeiter sind nicht verfügbar	<ul style="list-style-type: none"> ■ Probleme vor sich herschieben ■ Ressourceneinsatz nicht realistisch 	Ja
Fehlende Mitarbeitermotivation	<ul style="list-style-type: none"> ■ Mangelndes Team- und Change Management ■ Keine Zieldefinition ■ Zu wenig miteinander „schwätzen“ 	Ja

Disharmonien oder Verbrüderung der Projektleiter	<ul style="list-style-type: none"> ■ Nicht mehr sachlich agieren ■ Nicht die Auseinandersetzung suchen (zu freundlich) 	Teilweise
Schwächen in der Dokumentation und Planung	<ul style="list-style-type: none"> ■ Falsches Projektvorgehen ■ Falsche technische Infrastruktur ■ Vielleicht auch zu viel Dokumentation 	Nein
Software läuft nicht zufrieden stellend	<ul style="list-style-type: none"> ■ Softwareauswahl richtig ■ Hat Key Player das richtige Know How 	teilweise

Wie man gut aus der Aufstellung erkennen kann, Scrum hilft bei den Bereichen, wo „softskills“ gefragt sind, wo das Miteinander in einem Projekt wichtig ist, das Arbeiten auf Vertrauensbasis und das hinarbeiten auf ein gemeinsames Ziel. Und dieses wird unterstützt durch den wesentlichen Fokus der agilen Methoden auf das Team. Ein Team liefert End to End eine definierte Funktionalität, arbeitet Cross Funktional und verantwortet eine Lösung von Anfang bis Ende. Es ist nicht so das bei Scrum ein Analyst etwas in ein Designdokument definiert, anschließend wird dieses übergeben und der Analyst hört nie wieder etwas. Das Team erarbeitet ein Design zusammen, wirkt an Testfällen, führt diese aus und verantwortet zusammen die Lösung gegenüber den Stakeholder.

Diese Sichtweise stellt jedoch auch die gesamte Projektorganisation vor neuen Herausforderungen. Anstatt nun die einzelnen Arbeitspakete zu budgetieren (und gegen die Actuals zu tracken) müssen bei Scrum Kapazitäten betrachtet und budgetiert werden. Es wird grober betreffs der finanziellen Umsetzungskosten geplant, dafür wird zielgerichteter und freier gearbeitet.

Jedoch macht sicherlich Scrum längst nicht in allen Bereichen Sinn, denn teilweise sind die ERP Anforderungen zu übergreifend und komplex, um diese in einzelnen kurzen Zeitrahmen umzusetzen. Dieses soll die nachfolgende Stacey Matrix darstellen.

Stacey Matrix ERP

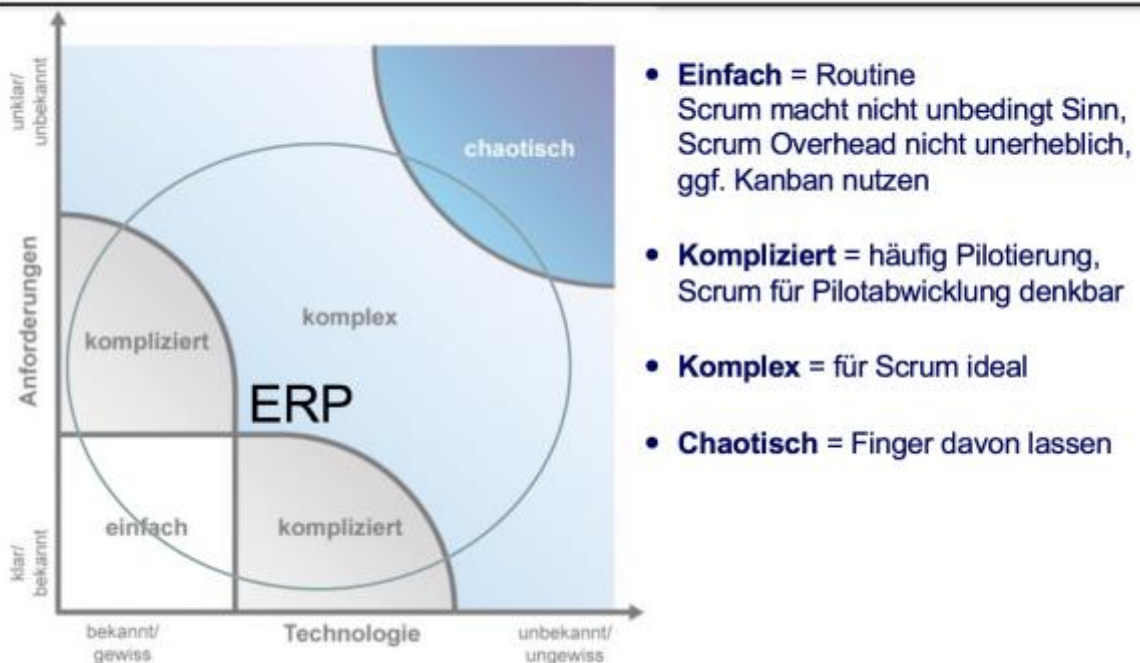


Abb. 5: Stacey Matrix

Auf der horizontalen ist die Komplexität der „Technologie“, also die Umsetzungsidee, aufgeführt. Auf der vertikalen die Konkretheit der Anforderung der Stakeholder.

Ist die Komplexität der Umsetzungsidee gering und die Anforderung ist völlig klar, so kann man hier von Routinetätigkeiten sprechen, Beispielweise hat man für sein ERP System ein standardisiertes Reportingmodul und die Fachbereiche fordern einen simplen Report. Hier wäre Scrum als agile Methode überdimensioniert, Anforderung (=User Story) ist klar, Umsetzung ist kein Problem, es muss nur jemand tun. Hier wäre ggf. die Umsetzung mittels der agilen Methode Kanban ein besseres Instrument, zielgerichtet zu einem Ergebnis zu kommen.

Andererseits sind total undefinierte Anforderung in Kombination mit keiner Lösungsidee auch nicht für Scrum geeignet, da dabei in den kurzen Sprint Intervallen keine Ergebnisse erzielt werden. Hier muss man an mindestens einer der Variablen arbeiten und diese erst mal konkretisieren, ggf. diese Konkretisierung mittels eines Sprints versuchen.

Ideal für Scrum sind Bereiche, wo die Anforderungen einigermaßen klar sind und es auch erste Lösungsansätze gibt. Hier kann das Cross funktionale Team Lösungen erarbeiten und in kurzen Feedbackschleifen mit den Fachabteilungen zu einem Ergebnis kommen.

Scrum kann sicherlich helfen eine effiziente Einführung oder Pflege des ERP Systems zu erreichen. Es stellt jedoch auch die Organisation vor Herausforderungen. Einfach zu sagen, so liebes Team, ab heute arbeiten wir agil, ist nicht die Lösung. Der Teamgedanken muss verankert werden, das Line Management muss sich bei den Cross Funktional Teams

etwas zurückhalten. Die Budgets müssen anders betrachtet werden, weg von einem reinen Projektbudget zu einem Teambudget.

Weiterhin kann nicht einfach 1:1 der Scrum Prozess bzw. Bestandteile des agilen Manifest über ein ERP gestülpt werden. ERP hat auch unter Scrum die Besonderheit, dass manche Dinge vielleicht Monate dauern, bis sie letztendlich nutzbar sind und es wird selten einen Produkt Owner für die Priorisierung geben.

Folgende Punkte sollte man bei der Scrum Nutzung in ERP Projekten betrachten:

- **Kleine Deliveries/Timebox**
Scrum basiert auf vielen kleinen Deliveries, welche in ERP schwierig zu bewältigen sind -> ggf. Sprint Dauer erhöhen
- **Budget Driven Organisationen**
Scrum basiert auf bestehende Kapazitäten, welche entsprechend Prioritäten eingesetzt werden. Bei Budget orientierteren Organisationen kann diese Budget Sichtweise zu einem totalen Chaos führen, welches gesteuert werden muss.
- **Multiple Produkt Owner**
Konzept vom Scrum ist ein Produkt Owner, welcher priorisiert. Dieses ist aufgrund der zusammenhängenden ERP Funktionalität häufig nicht umsetzbar. Mehrere PO müssen sich zusammentun, abstimmen und um die Team Kapazitäten kämpfen.
- **Splitting User Stories**
Um 2/3 Wochen deliveries zu erreichen, verfällt man leicht in das Zersplitten echter Userstories in „Micky Maus“ Task. Das verkompliziert die Abstimmung und die Transparenz eines Business Deliveries.

Letztendlich kann Scrum helfen auch in ERP Projekten die Kommunikation zu erhöhen und somit eine zielgerichtete Umsetzung voranzutreiben. Das klassische Problem, jemand denkt sich was aus, welches dann doch anders oder auch nicht gebraucht wird, soll durch diese Kommunikation minimiert werden und somit ein ERP Projekt zu einem höheren Erfolg machen.

Typische Szenen, wie es in ERP Projekten läuft



Und festgestellt wird dieses häufig nach Monaten

Abb. 6: Typische ERP Szenen

Kontaktadresse:

Dirk Blaurock

Dirk Blaurock IT Consulting

Datumer Chaussee 186a

D-25421 Pinneberg

Telefon: +49 (0) 171-4923557

E-Mail: Dirk.Blaurock@Dirk-Blaurock.de

Internet: www.Dirk-Blaurock.de