



Inkrementelle Verbesserungen in traditionellen Projekt- vorgehensweisen

Was die Wasserfallmethode von agilen Ansätzen lernen kann

CLAUS HÜSSELMANN, MARKUS H. GÖTZ, SCHEER MANAGEMENT

Agile Vorgehensmodelle beinhalten Lessons-Learned-Schleifen als integralen Bestandteil im Rahmen der inkrementellen Vorgehensweise. Dies ist ein wichtiger Bestandteil des Erfolgsrezepts – vielleicht sogar der essenzielle. Die Idee kann genutzt werden, um auch in traditionellen Wasserfallvorgehensmodellen durch eine entsprechende Anreicherung beziehungsweise spezifische Ergänzung des Vorgehensmodells den positiven Nutzenfakt zu erzielen.

1. Herausforderung Projektgeschäft

Der Anteil der betrieblichen Wertschöpfung, der durch Projekte oder projektähnliche Arbeitsweisen erwirtschaftet wird, steigt seit langem und wird in

absehbarer Zukunft signifikant zunehmend, wie unter anderem eine Studie der Deutsche Bank Research für den Betrachtungszeitraum 2007 bis 2020 festgestellt hat [1]. Obwohl sich Referenzmodelle für Projektmanagement-Methoden zunehmend etablieren (zum Beispiel nach PMI [2], IPMA [3] oder PRINCE2 [4]), lässt die Professionalität und die Erfolgsquote von Projekten weiterhin zu wünschen übrig [5]. In der Konsequenz werden neue Wege gesucht, wobei derzeit insbesondere so genannte agile Vorgehensweisen geradezu einen Hype erfahren – zumindest in der Interessenlage sowie im Anwendungsbereich der Softwareentwicklung. Solche agilen Projektmodelle, wie zum Beispiel Scrum [6] [7], erfordern nicht zuletzt eine spezifische Unternehmenskultur (flache Hierar-

chien, Delegation, Teamwork, flexible Prozesse etc.) und sollten und können auch nicht dazu dienen, als Feigenblatt für die Kapitulation vor einem geordneten Vorgehen zu dienen (getreu dem Missverständnis „agil = unstrukturiert“).

Nach agilen Methoden durchgeführte (IT-)Projekte werden vielfach als Paradebeispiele erfolgreicher Projekte aufgeführt [8]. Daher liegt die Idee nahe, sich einmal dediziert mit den Merkmalen und Erfolgsfaktoren agiler Vorgehensweisen zu beschäftigen und die Übertragbarkeit in traditionelle Projektvorgehensweisen – bis hin zu hybriden Projektansätzen [9] – zu untersuchen. Ein wichtiges Element scheint in diesem Sinne der kontinuierliche Verbesserungsprozess zur Projektlaufzeit zu sein, der agilen Vorgehensweise immanent ist und der in traditionellen, wasserfallstrukturierten Vorgehensweisen in der Regel zu kurz kommt.

2. Kritik am Wasserfallmodell

Traditionelle Methoden der Softwareentwicklung (etwa V-Modell XT [10]) oder Standardsoftwareimplementierung (etwa AcceleratedSAP [11]) folgen im Allgemeinen einem wohldefinierten Phasen- beziehungsweise Wasserfallmodell. Dabei werden die definierten Phasen sequentiell abgearbeitet und erreichen vielfach eine Dauer von bis zu einem Jahr, zum Beispiel in einem komplexen SAP-Implementierungsprojekt. Sichtbare Ergebnisse werden in der Regel in der zweiten Projekthälfte produziert, was unter Umständen ein Jahr oder mehr nach Projektstart sein kann. Es wird davon ausgegangen, dass jede Phase mit Erreichen eines Meilensteines abgeschlossen ist – spätere Rücksprünge sind nicht vorgesehen. Damit entfällt die Möglichkeit, Zwischenergebnisse auf der Basis von Feedback-Prozessen anzupassen.

Das strenge Phasen- oder Wasserfallmodell traditioneller Methoden führt zu ...

- Wissensaufbau auf Halde: In jeder Phase wird eine große Menge detailliertes Wissen erworben und dokumentiert. Der Einsatz des Wissens erfolgt zu einem großen Teil zeitlich entkoppelt in nachgelagerten Phasen.
- Intransparenz bezüglich des Projektfortschritts: Da erworbenes Wissen häufig erst sehr viel später umgesetzt wird, kann keine unmittelbare Verifika-

tion der Korrektheit durch Anwendung („Proof of Concept“) erfolgen.

- Vernachlässigung des sich ändernden Projektumfelds: Auf Veränderungen der externen und internen Projektparameter (zum Beispiel Gesetzgebung, neue Anforderungen, ...) sowie Erkenntnisse bezüglich getroffener Annahmen kann nicht oder nur erschwert reagiert werden.

Untersuchungen ergeben regelmäßig, dass bei mehr als der Hälfte aller ERP-Implementierungen geplante Zeit- und/oder Kostenziele nicht erreicht werden [12].

3. Merkmale von agilen Vorgehensmodellen

Ziel der agilen Softwareentwicklung ist es, den Prozess der Softwareentwicklung im Vergleich zu klassischen Vorgehensmodellen wie dem V-Modell XT flexibler und schlanker zu machen, da diese oft als schwergewichtig und bürokratisch angesehen werden. Im Fokus sollen die zu erreichenden Ziele und die technischen und sozialen Probleme stehen. Im Februar 2001 wurde das Agile Manifest formuliert. Dabei wurden vier wesentliche Werte als Basis definiert [13]:

KURZ UND BÜNDIG

Zurzeit ist agiles Projektmanagement in aller Munde. Allenthalben stellen sich Organisationen, Projektleiter und Teams die Frage, ob nicht mit agilen Methoden bessere Erfolge erzielt werden können. Entsprechende Vorgehensmodelle, zum Beispiel Scrum, beinhalten Lessons-Learned-Schleifen als integralen Bestandteil im Rahmen der inkrementelle Vorgehensweise. Systematische Lessons Learned, welche insbesondere zur Projektlaufzeit für anstehende Projektphasen genutzt werden, sollten auch in Wasserfall-Vorgehensmodellen zur Erhöhung der Erfolgsquote und der Projekteffizienz integriert werden.

Stichworte: Inkrementelle Vorgehensweise, Lessons Learned, Timeboxing, Rollout-Projekt, Agilität und Wasserfall, Projektvorgehensmodell

ID	Identifikator der Position/Ifd. Nummer
Themenbereich	Thematische Klassifizierung der Position zur gezielteren Bearbeitung und Zuordnung, z.B. Teambuilding
Faktor	Benennung der Position
Kurzbeschreibung	Beschreibung der Position
Kategorie	Charakterisierung der Position als Erfolgs- oder Störfaktor
Einfluss	Höhe der Relevanz der Position: hoch – mittel – gering
Maßnahme	Definierte Aktivität zu Behebung (Störfaktor) oder Stärkung (Erfolgsfaktor) der Position
Kurzbeschreibung Maßnahme	Beschreibung der Maßnahme
Zuständigkeit	Zuordnung der Maßnahme zu Person(en) ("Responsible" nach RACI)
Datum	Geplantes Erledigungsdatum
Bemerkung	Ergänzende Anmerkung/Erläuterung

Abbildung 1: Template zum Lessons-Learned-Workshop

- Individuen und Interaktionen sind wichtiger als Prozesse und Werkzeuge. Zwar sind wohldefinierte Entwicklungsprozesse und Entwicklungs werkzeuge wichtig, wesentlicher sind jedoch die Qualifikation der Mitarbeitenden und eine effiziente Kommunikation zwischen ihnen.
- Funktionierende Programme sind wichtiger als ausführliche Dokumentation. Gut geschriebene und ausführliche Dokumentation kann zwar hilfreich sein, das eigentliche Ziel der Entwicklung ist jedoch die fertige Software.
- Die stetige Abstimmung mit dem Kunden ist wichtiger als die ursprüngliche Leistungsbeschreibung in Verträgen. Statt sich an ursprünglich formulierten und mittlerweile veralteten Leistungsbeschreibungen in Verträgen festzuhalten, steht vielmehr die fortwährende konstruktive und vertrauensvolle Abstimmung mit dem Kunden im Mittelpunkt.
- Der Mut und die Offenheit für Änderungen stehen über dem Befolgen eines festgelegten Plans. Im Verlauf eines Entwicklungsprojektes ändern sich viele Anforderungen und Randbedingungen ebenso wie das Verständnis des Problemfeldes. Das Team muss darauf schnell reagieren können.

Diese Prinzipien sind Leitsätze für die agile Arbeit, wie sie in entsprechenden Vorgehensmodellen (zum Beispiel Scrum) umgesetzt werden. Doch was ist daran eigentlich „agil“ im engeren Sinne, drückt

also die besondere Flexibilität dieser Ansätze aus? Aus unserer Sicht ist dies insbesondere die Bereitschaft, die Anforderungen (Requirements) sowie den Umfang (Scope) des Projektes zur Laufzeit aufgrund gewonnener Erkenntnisse zu verändern. Diese Erkenntnisse werden durch systematische Prozesse der Rückkopplung mit den Stakeholdern (Anwendern, Auftraggebern) gewonnen. Im Scrum-Vorgehen [6] wird dies beispielsweise manifestiert durch das Element „Sprint Review“, bei dem das Entwicklerteam nach der relativ kurzen Phase eines so genannten Sprints eine Rückmeldung zu den vorliegenden Ergebnissen mit der Option modifizierter Anforderungen für den nächsten Entwicklungszyklus (Sprint) zu bekommen. Diese und gegebenenfalls auch andere Einflüsse veranlassen den Product Owner, den Arbeitsvorrat der umzusetzenden Anforderungen (Product Backlog) bei Bedarf zur Projektlaufzeit anzupassen. In diesem Punkt grenzt sich das agile Vorgehen grundlegend von traditionellen, wasserfallorientierten Vorgehensmodellen ab, die im Allgemeinen darauf fußen, einen festgeschriebenen Leistungsumfang sequenziell abzuarbeiten.

Andere Charakteristika agiler Methoden, wie zum Beispiel die intensive Kommunikation nicht zuletzt zur Verbesserung der Arbeitsweisen im Projekt, sind keine Alleinstellungsmerkmale von Agilität und können beziehungsweise sollten auch in anderen Vorgehensmodellen Eingang finden.

4. Best-of-Breed-Ansatz

4.1 Inkrementelle Verbesserung durch Lessons Learned

Im agilen Vorgehensmodell Scrum findet die Reflektion der Arbeitsweise im Projekt (Zusammenarbeit, angewendete Methoden, Standards und Techniken, Hindernisse etc.) in den Elementen „Daily Scrum“ sowie insbesondere der sogenannten „Sprint Retroperspective“ statt. Letztere stellt einen systematischen Prozess des Lessons Learned dar, wobei mit Blick auf den nächsten Entwicklungszyklus Verbesserungspotenziale identifiziert und entsprechende Konsequenzen gezogen werden. Auf diese Weise gelingt die gewünschte Lernkurve zur Projektlaufzeit.

Diese konsequente und systemimmanente Durchführung von Lessons-Learned-Aktivitäten ist unseres Erachtens ein entscheidender Schlüsselfaktor

für den Erfolg agiler Methoden. Dabei wird insbesondere aus den Fehlern und Problemen gelernt, um in der nächsten Phase des Projektes bereits eine Verbesserung zu erzielen. Es geht also keinesfalls (nur) darum, für das nächste Projekt Erkenntnisse zu gewinnen, was in vielen Organisationen selbst mit ansprechendem Reifegrad des Projektmanagements als Ziel entsprechender Aktivitäten der Projektabschlussphase etabliert sein mag.

Unser Postulat lautet daher, diesen Erfolgsfaktor agiler Methoden auch in Wasserfall-Projektmodellen systematisch zur Anwendung zu bringen! So fordert beispielsweise der sogenannte Project Management Body of Knowledge (PMBOK) des Project Management Institutes (PMI [2]) mit der Prozessgruppe Projektabschluss (Project Closure) die regelmäßige Durchführung phasenabschließender Elemente. Diese sollten – neben den üblichen Abnahme- und Freigabe-bezogenen Aspekten – nicht zuletzt das Thema Lessons Learned fokussieren.

Diese Lessons-Learned-Aktivitäten werden in Form von Workshops mit dem Projektteam sowie gegebenenfalls weiteren dafür relevanten Stakeholdern durchgeführt, deren Umfang sich an der Dauer der zurückliegenden Projektphase orientiert. Hier lassen sich wiederum Anleihen an Scrum vornehmen, wobei die erwähnte Sprint-Retroperspektive nach einem vierwöchigen Entwicklungszyklus circa zwei Stunden in Anspruch nehmen sollte [14].

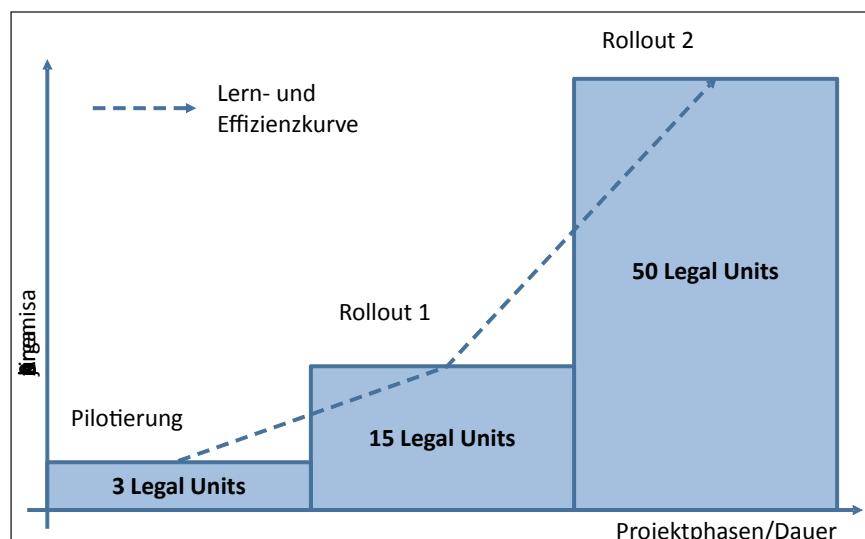


Abbildung 2: Phasenmodell Rollout-Projekt

Bei den Lessons-Learned-Erhebungen sollten Erfolgsfaktoren und Störfaktoren identifiziert werden: „Was hat den Ablauf des Projektes gestört?“ beziehungsweise „Was war der Grund, warum Dinge gut gelungen sind?“. Letztere werden leider vielfach unter den Tisch gekehrt, da die Teilnehmer oftmals dazu neigen, nur die Probleme zu diskutieren. Es ist aber neben der Beseitigung von Störfaktoren ebenso wichtig, erkannte Erfolgsfaktoren für die kommenden Arbeiten „hinüber zu retten“. Persönliche Schuldzuweisungen der Teilnehmer untereinander müssen durch eine geschickte (An-)Moderation vermieden werden; auch dazu trägt die Identifikation von Erfolgsfaktoren psychologisch bei.

Nicht zuletzt sind Lessons Learned nur etwas wert, wenn daraus Maßnahmen abgeleitet werden, die in der Folge konsequent – und mit Priorisierung – abgearbeitet werden. Die Abbildung 1 zeigt ein einfaches, aber effizientes Instrument zur Dokumentation eines Lessons-Learned-Workshop auf:

Mit den phasenbezogenen Lessons Learned lässt sich der größte Nutzen für das Projekt erzielen. Die entstehenden Lernkurven ermöglichen gerade zum entscheidenden Abschluss eines Projektes hin ein eingespieltes Arbeiten mit hoher Effizienz.

4.2 Lerneffekte am Beispiel eines Rollout-Projekts

Projekte sind betriebliche Organisationsformen, die bedingt durch ihre (nur) temporäre Ausprägung darauf angewiesen sind, dass die Arbeitsformen möglichst schnell in einen „eingeschwungenen“ Zustand kommen. Die Zeit der Linienorganisation hierfür haben Projekte nicht.

Besonders hohe Anforderung an effizientes Arbeiten stellen Rollout-Projekte (beispielsweise eines ERP-Systems). Diese sind charakterisiert durch eine Vielzahl involvierter beteiligter Personen beziehungsweise Organisationseinheiten (Niederlassungen, Gesellschaften etc.) sowie sich im Grundsatz wiederholender Aufgabenstellungen. Es ist daher in der Regel erfolgskritisch, effiziente Arbeitsweisen im Projekt zu entwickeln und in diesem Sinne eine adäquate Lernkurve zu erreichen.

Das folgende Beispiel zeigt ein Projektvorgehensmodell auf, das durch eine spezifische Einführungsstrat

egie (Konzeption der Projektphasen) sowie zielgerichtete Projektprozesse diesen Gedanken erfolgreich umgesetzt hat (siehe auch [15]). Es ist damit gelungen, einen SAP-Rollout für rund 65 Legal Units (Behörden) in einem mehrjährigen Projekt termingenau erfolgreich durchzuführen.

Ausgangsbasis ist die sogenannte Einführungsstrategie, bestehend aus Pilotierung (natürlich auch eine klassische Technik der Erkenntnisgewinnung), Rollout 1 und Rollout 2, wie in Abbildung 2 schematisch dargestellt:

Dabei wurde die Pilotierung anhand dreier repräsentativ ausgewählter Units durchgeführt, die den Großteil der restlichen Units fachlich und organisatorisch nahekommen – und bereit waren, den noch geringen Reifegrad des Projektes in Kauf zu nehmen. Auf diese Weise konnte der Rollout 1 mit weiteren 15 Legal Units auf Basis einem fachlich reifen SAP-Templates durchgeführt werden. Das Vorgehen zum Rollout selber jedoch wurde in dieser Konstellation naturgemäß erstmals durchgeführt, also zunächst nicht im eingeschwungenen Zustand.

Die Dauer der Projektphase und die Anzahl der betroffenen Unit erforderte bereits hier ein stark parallelisiertes Vorgehen. Hier wurde ein Timeboxing-Ansatz (auch ein Element von agilen Modellen wie Scrum) mit fest vorgeschriebenen, wiederkehrenden Blöcken

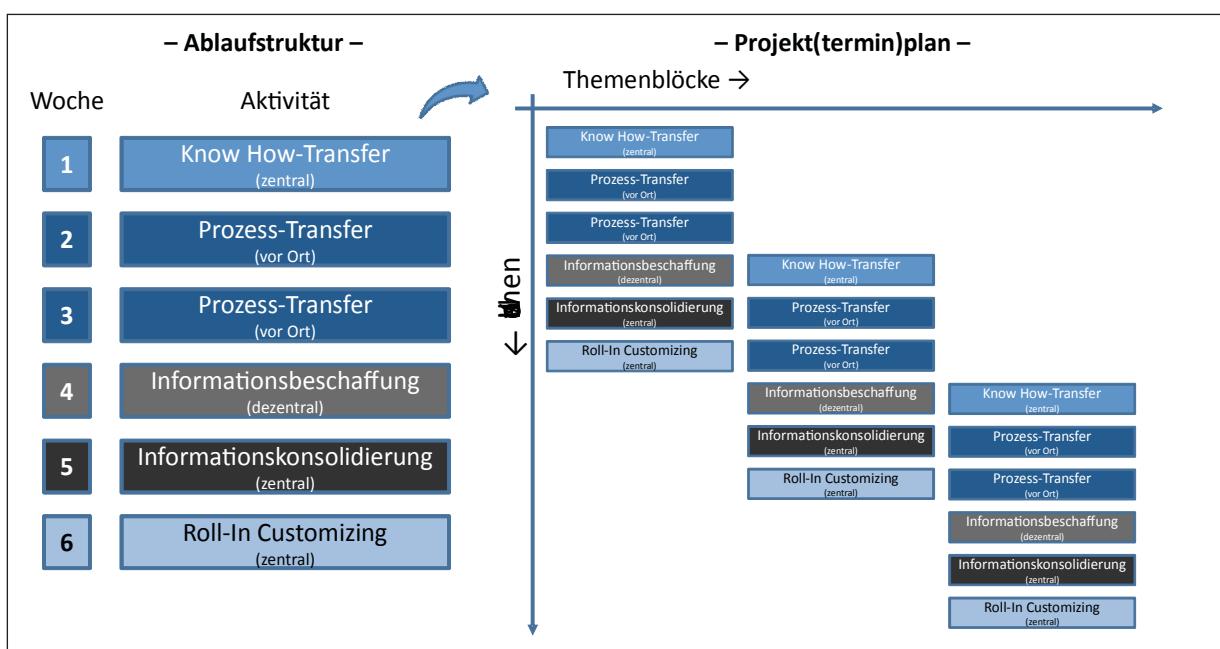


Abbildung 3: Phasenmodell Rollout-Projekt

gewählt, der es ermöglichte, eine maximale Effizienz hinsichtlich der personellen Kapazitäten, terminlichen Planungssicherheit, Projektfortschritt und nicht zuletzt Lernkurve zu erreichen. Die Abbildung 3 zeigt das entsprechende Planungsschema, bestehend aus zeitlich versetzt durchgeführten Wochen-Sequenzen mit den Blöcken „Know-how-Transfer“, „Prozess-Transfer“, „Informationsbeschaffung“, „Informationskonsolidierung“ und „Roll-In Customizing“.

Der so genannte Know-how-Transfer spielt dabei die entscheidende Rolle im Sinne eines Lessons Learned. In diesem Block wurde in Workshop-Form die folgende Detailphase fachlich vorbereitet und nicht zuletzt die vorherige Detailphase reflektiert, um auf diese Weise das Vorgehen Schritt für Schritt und insbesondere mit Blick auf den anstehenden Rollout 2 zu optimieren. Die Reflektion umfasste dabei fachliche, aber auch prozedurale und kommunikative Aspekte. Das Timeboxing-Postulat sorgt hierbei zudem für Effizienz: In Anwendung des Pareto-Prinzips werden auch bei einem gegebenenfalls notwendigen „Timeout“ 80% der Erkenntnisse, also des Nutzens, gewonnen. Durch die Wiederkehr des Ablaufs im Folgezyklus wird in Summe zudem jeder wichtige Aspekt identifiziert.

Der so durchgeführte Rollout 1 führte zu einer signifikanten Lernkurve, was zum Erfolgsgaranten für den Rollout 2 mit ca. 50 Legal Units im gleichen Projektzeitraum (1 Jahr) wurde. Im Rollout 2 wurde geschilderte Planungsschema weiter verwendet – allerdings mit jeweils verringertem Zeitbedarf pro Workshop (anders gesprochen: mit gewonnenem zeitlichen Puffer, was der schlussendlich erreichten Termintreue zu Gute kam).

5. Fazit

Zurzeit ist agiles Projektvorgehen in aller Munde. Allenthalben stellen sich Organisationen, Projektleiter und Teams die Frage, ob nicht mit agilen Methoden bessere Erfolge erzielt werden können. Dem Ursprung dieser Paradigmas entsprechend werden dabei (noch?) insbesondere IT-Entwicklungsprojekte fokussiert.

Ein Schwarz-weiß-Denken ist dabei jedoch nicht angesagt. Vielmehr sollte im Sinne eines Best-of-Breed-Ansatzes das Beste aus beiden Welten – dem traditionellen, phasenorientierten Wasserfallmodell und der Agilität – genutzt werden. Denn schließlich ist nicht nur die Flexibilität im Requirements Management („Rollierender Scope“) ein agiler Wettbewerbsvorteil, sondern gerade auch Elemente wie kleine Teams, Teamwork, offene Kommunikation und nicht zuletzt die systematische zeitnahe und wiederholte Rückkopplung und Reflektion von Projektergebnissen und -prozessen. Vielleicht sind das sogar die entscheidenden Faktoren.

Mit den phasenbezogenen Lessons Learned lässt sich der größte Nutzen für das Projekt erzielen.

Dieser Erkenntnis folgend, sollte in jedem Projekt, gleich welchen Vorgehensmodells, systematische Lessons-Learned-Sequenzen integriert werden, welche nicht nur zum Projektende für die „Projektnachwelt“ erarbeitet werden, sondern insbesondere und gerade zur Projektlaufzeit für anstehende Projektphasen und damit den Erfolg des Projektes selber genutzt werden. Die Quote von Projektmisserfolgen lässt sich auf diese Weise mit Sicherheit organisationsweit signifikant verbessern.



LITERATUR

- [1] Deutsche Bank Research: Deutschland im Jahr 2020, Neue Herausforderungen für ein Land auf Expedition, 2007
- [2] Project Management Institute: A Guide to the Project Management Body of Knowledge (PMBOK® Guide), 3. Ausgabe, Pennsylvania 2004
- [3] GPM Deutsche Gesellschaft für Projektmanagement e.V.: ICB - IPMA Competence Baseline, Version 3.0, Nürnberg 2008
- [4] Ebel, Nadin: PRINCE2 – Projektmanagement mit Methode, Addison-Wesley, München 2007
- [5] Saïd Business School working papers, August 2011, University of Oxford, online 24.05.2012
- [6] Schwaber, Ken, Sutherland, Jeff: The Scrum Guide, The Definitive Guide to Scrum: The Rules of the Game; 2011
- [7] Schwaber, Ken: Agile Project Management with Scrum, Microsoft Press, 2004
- [8] Schwaber, Ken, live in: Google Tech Talks, 5. September 2006, online 28.12.2011
- [9] Hüsselmann, Claus: Agilität im Auftraggeber-/Auftragnehmer-Spannungsfeld, Mit hybrider Projekt-Ansatz zur Win-Win-Situation, PM Summit 2012
- [10] Bundesministerium des Innern: V-Modell® XT, Version 1.2.1, 2006
- [11] <http://wiki.sdn.sap.com/wiki/display/SM/ASAP+Methodology>, online 03.10.2012
- [12] 2010 ERP Report, Panorama Consulting Group, 2010, online 28.12.2011
- [13] <http://de.Wikipedia.org>, online 27.12.2011
- [14] Pichler, Roman: Scrum – Agiles Projektmanagement erfolgreich umsetzen, dpunkt.verlag, Heidelberg 2009
- [15] Bürmann, Richard, Hüsselmann, Claus: ERP-gestützte Personalwirtschaft in der Bundesverwaltung, in Innovative Verwaltung 5/2008, S. 39 – 41

SUMMARY

Incremental improvements in traditional project procedure models – what the waterfall method can learn from agile approaches

At present, agile project management is widely talked about. Everywhere organizations, project managers and teams are wondering, whether they could be more successful using agile methods. Appropriate project procedure models, like scrum, contain lessons learned loops in terms of an incremental proceeding. Systematic lessons learned, especially executed during project life cycle with respect to the upcoming project phases, should as well be integrated within waterfall-like project approaches to improve the success quota and the project efficiency.

Keywords: Incremental approach, lessons learned, timeboxing, rollout project, agility and waterfall, project procedure model

SERVICE

AUTOREN



Dipl.-Math. techn. Dr. rer. oec.
Claus Hüsselmann
Scheer Management GmbH
Associate Partner Project
Performance Management
Consulting

Claus Hüsselmann ist als Partner bei Scheer Management zuständig für das Beratungsfeld Projekt-, Programm- und Portfolio Management. Nach Studium der Technomathematik wirkte er zunächst als leitender Entwickler in einem SAP-Systemhaus für kommunale Anwendungen. Nach seinem Wechsel zur IDS Scheer AG – jetzt Software AG – verantwortete er mehrere Beratungseinheiten, (Groß-)Projekte sowie zuletzt den Project Operations & Risk Control-Bereich für das gesamte Consulting-Geschäft.



Dipl.-Ing. Markus H. Götz
Scheer Management GmbH
Principal Consultant Project
Performance Management

Markus H. Götz ist im Bereich Project Performance Management Principal Consultant für Projekt-/Programmmanagement mit den Schwerpunkten Organizational Change Management und Wissensmanagement. Seine Erfahrungen im Projekt- und Multiprojektmanagement sammelte er im nationalen und internationalen Umfeld und anschließend in den Bereichen Weiterentwicklung, Rüstung sowie im Controlling.

KONTAKT

claus.huesselmann@scheer-management.com
markus.goetz@scheer-management.com

Scheer Management GmbH
Science Park 2
66123 Saarbrücken
Tel.: +49 (0)172 / 4580 266
Fax: +49 (0)681 / 935 1100
www.scheer-management.com