URL 1: ---

Agilität jenseits der Lagerfeuerromantik

Ein kritischer Blick über den Tellerrand

**Abstract** Agilität hat mittlerweile ihren festen Platz in der IT, kaum noch ein Unternehmen, welches sich nicht mit agilen Methoden beschäftigt. Dennoch bleiben jede Menge Fragen offen und so mancher kritischer Beobachter fragt sich, ob Agilität wirklich all die vielfach idealisiert erscheinenden Versprechen einlösen kann, die ihr zugeschrieben werden. Was ist dran an all den Versprechen? Seriöse Realität oder doch nur Lagerfeuerromantik? Und wie passt das überhaupt in die Realität vieler Unternehmen? Wir wagen einen Blick über den Tellerrand und versuchen eine Betrachtung von einem etwas allgemeineren Standpunkt aus.

## Autorenfotos

--- als Anlage

## Autorenbeschreibungen

Sven Johann ist Software-Entwickler bei der adesso AG. Seit 10 Jahren interessiert er sich für leicht- und schwergewichtige Software-Entwicklungsprozesse und deren Anwendung. Aktuell beschäftigt er sich mit sehr kurzen Releasezyklen durch Continuous Delivery.

Uwe Friedrichsen hat langjährige Erfahrungen als Architekt, Projektleiter und Berater. Aktuell ist er bei der codecentric AG als CTO tätig und beschäftigt sich in dem Kontext insbesondere mit agilen Verfahren und neuen Architekturansätzen und Technologien wie z.B. Cloud Computing.

Artikel (Fließtext)

Vielfach erscheint die Diskussion um Agilität immer noch religiös geprägt: Viele Befürworter vermitteln Agilität eher dogmatisch, man ist entweder „dafür“ oder „dagegen“, alt ist „schlecht“ und neu ist „gut“. Ein Hinterfragen der Werte und Praktiken und ein Abgleich mit anderen Ansätzen ist in einem solchen Umfeld schwierig.

Auf der anderen Seite begegnen einem immer wieder sehr fragwürdige Interpretationen und Umsetzungen von Agilität, die einen erfahrenen IT-ler je nach Situation ein kräftiges Runzeln oder Schweißperlen auf die Stirn treiben. Was ist denn nun richtig, was ist falsch und warum ist es so schwer, das alles vernünftig zu greifen?

Letztlich gibt es bestehende Realitäten in den Unternehmen, gepaart mit den Erkenntnissen aus 40 Jahren Software Engineerings, bei denen sich die Frage stellt, wie Agilität dazu passt. Sind all die bekannten und bewährten Techniken nur noch für den Papierkorb tauglich oder ist eher die Agilität eine Traumwelt ohne Realitätsbezug?

Wir haben diese und weitere Fragen intensiv diskutiert und sind für uns zu einigen „Outside-the-Box“-Antworten gekommen, die zumindest wir in der allgemeinen Diskussion bislang nur selten so gehört haben. Wir denken, dass diese Sichten eine weitere Facette in der Diskussion pro und contra Agilität darstellen können und versuchen deshalb, im weiteren Verlauf dieses Artikels, die wichtigsten Punkte unserer Diskussion nachzuzeichnen.

Um das Ganze für Sie etwas abwechslungsreicher zu gestalten, haben wir es in ein imaginäres Gespräch zwischen zwei Personen gepackt:

* Peter ist eher der konstruktive Zweifler, der Agilität zwar positiv gegenübersteht, aber verschiedenste Aspekte kritisch hinterfragt.
* Paul ist der agile Praktiker, der versucht, Agilität zu leben, dabei aber auch nicht glaubt, dass Agilität nicht die Antwort auf alle Fragen der IT ist.

Aber genug der Vorrede, lassen Sie uns der Diskussion von Peter und Paul zuwenden.

# Agilität falsch verstanden

***Peter***: Hallo Paul! Ich habe nochmal über Agilität nachgedacht: Agile Methoden sind zwar en vogue, werden aber meiner Meinung nach scheitern. Nicht das ich die Ideen falsch finde. Viele Ideen sind ja gar nicht so schlecht. Aber ich bin mir sicher, dass Agilität scheitern wird, denn es ist sehr leicht, das alles falsch zu verstehen.

So war das nämlich schon immer mit Software-Prozessmodellen – sie wurden immer schon falsch verstanden. Der Wasserfall hat ja in der Originalausgabe von Winston Royce [Roy70] auch Feedback-Loops, eben gerade weil man sie benötigt. Trotzdem hat der Wasserfall, entgegen seiner Original-Absicht, eine unbewegliche IT produziert, weil seine Ideen falsch umgesetzt wurden und man auf abgeschlossene Phasen besteht. Der Rational Unified Process (RUP) [Kru03] steht ebenso in der Kritik, aber dort steht nirgendwo geschrieben, dass man nur eine Iteration machen soll und 1000 Artefakte erstellen muss.

Und, ja, viele Scrum-Coaches berichten, dass sie in vielen havarierten Scrum-Projekten unterwegs sind, Scrumerfall oder auch ScrumBut genannt. Es steht Scrum drauf, es ist aber der alte, falsch verstandene Wasserfall drin (siehe z.B. [App], [Sho]). Man macht nun ein Daily Standup, benennt einen Product Owner, lässt sich zertifizieren und dadurch wird das jahrelang praktizierte „Irgendwie“-Vorgehen zu Scrum. Oder man begründet wildes Code-and-Fix mit: ”Wir sind agil”.

***Paul***: Das ist richtig, Peter, aber das ist doch ein Problem, das weit über das Thema Agilität und Software-Prozesse hinausgeht – wobei Agilität natürlich mehr als nur ein Software-Prozess-Thema ist, aber das soll uns hier nicht stören. Aus meinen Erfahrungen heraus wage ich die etwas pessimistische Prognose, dass dieses Problem der Missinterpretation nicht allgemein lösbar ist, auch nicht für Agilität.

Es gibt immer ein paar (wenige) Leute, die die Idee hinter einem nicht-trivialen Thema wie z.B. Agilität verstehen und diese sinnvoll umsetzen. Es gibt aber auch immer viele Leute, die das Thema fehlinterpretieren, bis hin zur kompletten Karikatur seiner selbst, sei es, weil sie das Thema nicht verstanden haben oder sei es, weil sie eigene Interessen verfolgen.

Aber das ist nichts Spezifisches im Bereich der Software-Prozesse. Das liegt im Wesen des Menschen. Vergleichbare Effekte kann man in allen Bereichen des Lebens beobachten. Man muss doch nur einmal in Richtung Politik oder Religion schauen.

Was dabei helfen kann, sind Ausbildung und Training. Beides sind Faktoren, die erwiesenermaßen dazu beitragen, dass Menschen nicht mehr so schnell Opfer verquerer Fehlinterpretationen zu werden. Viel mehr kann man da aber eigentlich nicht tun.

# Agilität im historischen Kontext

***Peter***: Wie ist Agilität eigentlich entstanden? Historisch gesehen gibt es schon seit mehreren Jahrzehnten inkrementelle Entwicklung, kollaborative Programmierung, etc. [Bas03], da sieht Agilität für mich nach Rückbesinnung plus Weiterentwicklung aus.

***Paul***: Damit liegst Du aus meiner Sicht recht gut. Mitte bis Ende der 90er Jahre war der Höhepunkt der großen Vorgehensmodelle, die in ihrer ganzen Pracht selbst DIN A0 Plotter an ihre Grenzen gebracht haben, gepaart mit einen "architekturzentrierten Vorgehen". Es gab Regelwerke, die erst nach einem monatelangen Tayloring halbwegs handhabbar wurden – wenn sich denn jemand daran traute.

Menschen waren darin nur zutiefst misstrauenswürdige Geschöpfe, die den schönen Prozess durch ihre Dummheit gefährden und der Kunde und seine Anforderungen sind ganz aus dem Fokus geraten. Das Pendel war weit über eine sinnvolle Mitte hinaus ausgeschlagen.

Da kam Agilität mit seiner Forderung auf eine Rückbesinnung auf die eigentlichen Werte: Den Kunden, seine Anforderungen, Lösungen schaffen, Menschen und Vertrauen als Motor für erfolgreiche Zusammenarbeit. Das war keine hellseherische Genialität, das war nur ein Erkennen, dass man über das Ziel hinausgeschossen war und man jetzt eine Gegenbewegung benötigte, um wieder zur sinnvollen Mitte zu gelangen.

Natürlich sind dabei auch wieder einige Leute über das Ziel hinausgeschossen. Die agilen Dogmatiker von heute sind die Leute, die das Pendel in das andere Extrem zu zerren versuchen: keine Dokumentation, keine Planung, kein Ausbalancieren von nicht-fachlichen Anforderungen. Die Mitte zu treffen ist halt sehr schwer.

Es ist ja auch viel einfacher, eine extreme Position einzunehmen: Die ist leicht verständlich, man kann klare Abgrenzungen und Feindbilder aufbauen, man kann schön polarisieren, so ähnlich wie die Boulevard-Medien. Mit einer Position einer balancierten Mitte geht das nicht, da leidet die Popularität … ;-)

# Agilität hui, der Rest pfui

***Peter***: Die Berichterstattung ist im Allgemeinen auch (immer noch) sehr einseitig. Agilität ist „gut“, andere Vorgehensweisen stehen für das „Böse“ [Kru07], die schlechte Seite, der Sündenbock für das Scheitern. Aber ganz so trivial kann das doch nicht sein. Obwohl mir persönlich noch kein gescheitertes agiles Projekt begegnet ist, muss es sie doch geben. Es ist unmöglich, dass alle agilen Konzepte in allen Kontexten ohne Adaptierung an die jeweilige Situation funktionieren – sonst wäre es ja die lange ersehnte „Silver Bullet“ [Bro95] – die es nie geben wird, wie wir wissen. Wie kommt es, dass diese Meinung so stark vorherrscht?

***Paul***: Ich kann hier auch nur mutmaßen, aber ich sehe hier aus meiner Sicht vorwiegend zwei Faktoren. Der eine Faktor ist das Evangelistentum und der andere ist Selbstmarketing.

Beginnen wir mit dem Evangelistentum: Auch wenn Agilität eigentlich nicht mehr eine junge Bewegung ist, ist sie erst vor relativ kurzer Zeit in der breiten Masse angekommen. Um überhaupt so weit zu kommen, müssen die dafür notwendigen Umdenkprozesse befeuert werden. Dieses Befeuern übernehmen insbesondere am Anfang einer neuen Idee sogenannte "Evangelisten", Leute die eher eindimensional die Vorteile ihrer Idee anpreisen. Um eine Idee erfolgreich zu machen, sind diese Menschen letztlich unverzichtbar (siehe auch [Man]).

Viel der von Dir erwähnten eindimensionalen Berichterstattung kommt aus dem Umfeld. Ab einem bestimmten Verbreitungsgrad kommt dann natürlich der Zeitpunkt, zu dem sich der Nutzen der Evangelisten erschöpft hat und eher die Realisten gefragt sind, die das Thema differenzierter beleuchten. Diese findet man meiner Meinung nach heute aber immer häufiger und die Zahl der Evangelisten nimmt mit der Verbreitung der Agilität ab.

Selbstmarketing ist der zweite Punkt. Viele der Protagonisten haben sich Agilität auf ihre eigene Fahne geschrieben und da ist es doch nur verständlich, dass Agilität bei ihnen „gut“ sein muss. Wer schreibt sich schon etwas „Böses“ oder auch nur „nicht so Gutes“ auf die Fahne?

# Gründe für Agilität

***Peter***: Das ist alles? Es muss doch noch andere Gründe geben, warum plötzlich sehr viele dieses Vorgehen gut finden!

***Paul***: Die gibt es auch. Agiles Vorgehen ist für viele heutige Projektumfelder eine gute Wahl. Warum? Die meisten heutigen Projektumfelder haben im Sinne der Systemtheorie hohe komplexe Anteile und Agilität bündelt eine ganze Reihe seit vielen Jahren bekannter und erprobter Mittel, um Komplexität handhabbar zu machen.

Das ist übrigens eher ungeplant so entstanden. Initial wurden unter Agilität einfach nur eine Reihe von Werten und Vorgehensweisen gebündelt, von denen ihre Macher wussten, dass sie in ihren Umfeldern gut funktioniert haben. So gesehen ist Agilität eher als empirische Sammlung entstanden. Erst später wurde klar, dass es hier eine enge Beziehung zum Management komplexer Umfelder gibt.

Das beantwortet aber vielleicht auch ein Stück weit Deine Frage, wo Agilität nichts hilft (Stichwort: Keine „Silver Bullet“), nämlich überall dort, wo wir keine Komplexität im Sinne der Systemtheorie haben. Eine nicht komplexe Aufgabenstellung agil zu verwalten, ist eher kontraproduktiv. Habe ich z.B. eine stabile Umgebung mit stabilen Anforderungen, dann gibt es keinen Grund für ein agiles Vorgehen [Kru08]. Das führt dann meist zu Situationen, in denen die Aufwände letztlich deutlich höher werden als beim Einsatz klassischer Vorgehensweisen.

Man muss sich dabei aber vor Augen halten, dass die meisten heutigen IT-Projekte ganz klar komplex im Sinne der Systemtheorie sind: Unklare Anforderungen, dynamische Anforderungsänderungen, komplexe Beziehungen allerorten, usw. Deshalb macht ein agiles Vorgehen im Großen eigentlich fast immer Sinn. Das bedeutet aber nicht, dass man jede Detailaufgabe wie etwa das Aufsetzen einer Entwicklungs- und Build-Umgebung für ein Projekt automatisch agil gestalten muss. Auf der Ebene macht es dann immer noch Sinn zu differenzieren, um welche Art von Aufgabenstellung es sich handelt.

# Einführung von Agilität

***Peter***: Okay, aber wie soll ich da am besten beginnen? Was sind die Voraussetzungen? Da Agilität ja eine Art “Mindset” ist, kann man mit Sicherheit nicht Ruck-Zuck Erfolg damit haben, oder? Das Problem ist doch, dass man statt vorgegebenen Prozessvorschriften zu folgen seinen eigenen Weg finden muss.

***Paul***: Aus meiner Erfahrung sollte man sich nur nicht der Illusion hingeben, dass man sofort perfekt agil sein wird. Man muss Agilität lernen, und Lernen bedeutet, Fehler zu machen. Es wird also eine Reihe Irrwege und "blutige Nasen" geben, bis man das Thema wirklich verstanden hat und man nachhaltig agil wird.

Ein weiterer Faktor ist die Art des Lernens. Man muss gerade auch im agilen Umfeld bereit sein, Dinge lange genug zu erproben, bis man ihre Effekte wirklich verstanden hat, da sich viele Effekte im agilen Umfeld erst mit deutlicher Verzögerung zeigen und es vielfach keine unmittelbaren Ursache-Wirkung-Zusammenhänge gibt.

Das ergibt eine schwierige Balance zwischen dem Vermeiden vorschnellen Änderns passender und dem zu langen Beibehalten unpassender Verfahren. Viele agile Prinzipien werden vorschnell von Teams verworfen, weil sie sich nicht genug Zeit geben zu lernen. Sie denken nach einer kurzen Zeit, sie hätten es verstanden, sehen nicht ad hoc den gewünschten Effekt und verwerfen das Prinzip oder ändern es ab:

Pair Programming? Warum soll ich zwei Entwickler an eine Aufgabe setzen, die auch einer alleine erledigen kann? Weg damit. Daily Scrum? Wir sitzen doch eh schon zusammen. Retrospektiven: Wieso? Wir hätten nichts besser machen können. Und so weiter. Da gehen dann eine Menge Wissen und möglicher Erfolg vorschnell verloren

Ein externer Experte kann helfen, eine Reihe von Irrwegen zu vermeiden, weil er sich schon seine "blutigen Nasen" geholt hat und diese Erfahrungen einbringen kann. Aber eine Erfolgsgarantie ist er auch nicht, was man ja auch in der Praxis beobachten kann: Es gibt genug gescheiterte agile Projekte, die brav nach Lehrbuch von einem agilen Experten begleitet worden sind.

# Agilität in großen Projekten

***Peter***: Hm, naja, blutige Nasen, Irrwege, zugeben, dass man was falsch gemacht hat… Wenn ich mir meine Projekte in diversen Großunternehmen anschaue, dann kann ich nicht glauben, dass so was funktioniert. Denn in großen bis sehr großen Projekten gibt es doch immer Interessengruppen, die so etwas nicht zulassen werden. Mache ich was falsch und gebe es zu, wird das gegen mich verwendet. Ist doch so!

***Paul***: Ja, das ist ein sehr wichtiger Punkt: um erfolgreich ein agiles Projekt durchzuführen, ist es besonders wichtig, in einem vertrauensvollen Umfeld zu arbeiten. In Umfeldern, in denen Vertrauen sehr schwer bis unmöglich ist, ist ein vernünftiges agiles Arbeiten nicht möglich.

Misstrauensbasierte Umfelder basieren auf einer CYA-Strategie ("Cover your ass"), d.h. das Ziel, sich nur keine Blöße zu geben und sich gegen jeden möglichen Angriff abzusichern ist wesentlich wichtiger, als ein gutes und erfolgreiches Ergebnis zu produzieren.

Das ist aber genau das, was Agilität ja nicht will. Man will ein möglichst gutes Produkt abliefern, man möchte keine endlosen Verträge und Spezifikationen, man möchte seine Energie in die Erschaffung einer Lösung stecken und nicht in die Absicherung jedweder Eventualitäten.

Diese Nichtablenkung von der Lösungsherstellung funktioniert aber nur in einer auf Vertrauen basierenden Zusammenarbeit. Ich kann es mir nur leisten, nicht CYA zu machen, wenn ich mir sicher bin, dass mein Gegenüber mich nicht bei der nächstbesten Gelegenheit in die Pfanne haut. Ich kann z.B. nur auf epische Spezifikationen verzichten, wenn ich darauf vertrauen kann, dass mein Gegenüber die daraus resultierenden Lücken nicht gnadenlos zu seinem Vorteil ausnutzt.

Ist das nicht gegeben, ist echte Agilität nicht möglich. Punkt. Natürlich kann man trotzdem iterativ arbeiten, alle 4 Wochen dem Kunden etwas Lauffähiges vorführen oder ein tägliches Standup-Meeting machen, aber richtig agil ist das nicht. Hast Du ein von Politik und Misstrauen durchsetztes Projekt, wird das mit Agilität nichts.

***Peter***: Sag ich doch, bei Großprojekten geht das nicht!

***Paul***: Du spielst auf die in Großprojekten typischerweise sehr ausgeprägte politische Dimension an. Die macht es in der Tat sehr schwer. Es gibt aber auch noch andere Besonderheiten, die Großprojekte sehr schwer machen und häufig scheitern lassen. So neigen Großprojekte dazu, riesige Wasserfall-Projekte mit einem ausgeprägten Phasendenken zu sein. Wenn das schon bei deutlich überschaubareren Projekten aufgrund der Komplexität nicht funktioniert, dann ist ein solches Vorgehen gerade bei Großprojekten erst recht zum Scheitern verurteilt.

Hier kann einem eine agile Vorgehensweise gute Dienste leisten. Anstatt zu versuchen, zu Anfang alles festzuschreiben, was erwiesenermaßen für die meisten Systeme nicht möglich ist (siehe z.B. [Weg95], [Weg97]), fange ich besser mit etwas Kleinem an und ergänze das sukzessive in vielen überschaubaren Iterationen, wobei ich immer darauf achte, aus dem bisher Gemachten zu lernen und die nächsten Schritte daran anzupassen.

Großprojekte sind letztlich immer schwierig, nicht nur aufgrund der typischen Projektpolitik. Einige erfolgreiche große Projekte, die mit agilen Vorgehensweisen umgesetzt worden sind, zeigen einem aber, dass sich Agilität und Großprojekte nicht ausschließen.

# Was kommt nach Agilität …

***Peter***: Na schön, aber alles in allem hört sich das sehr schwer an. Da muss man erfahren sein. Du erwähnst die Theorie komplexer Systeme, ist da die nächste Stufe der Prozessmodelle am Horizont?

***Paul***: Ich weiß auch nicht, was die nächste Stufe ist, die ein Prozessmodell erreichen wird. Persönlich glaube ich nicht, dass es irgendwann *das* Prozessmodell geben wird. Eigentlich ist weitestgehend alles da, was wir benötigen. Die Kunst ist nur, die der konkreten Situation angemessenen Bausteine auszusuchen und sie angemessen zu kombinieren. Das hat aber aus meiner Sicht sehr viel mit persönlicher Kompetenz, auch "Können" genannt zu tun.

Ich definiere Können als eine Kombination aus Wissen, Erfahrung und Talent. In meiner Definition (die letztlich auch nicht von mir stammt, sondern die ich mal irgendwo gelesen hatte und als besonders gut treffend erachtet hatte) wird der eigentliche Knackpunkt sehr gut sichtbar: Das Talent.

Du kannst so viel wissen, wie Du willst und üben bis zum Umfallen, wenn Du kein Talent hast, wirst Du nie wirklich gut sein. Das gilt nicht nur für Fußball spielen (viele hunderttausende Jugendliche wissen praktisch alles über Fußball, trainieren ganz viel und landen trotzdem nicht in der Bundesliga), sondern auch für die IT und im Speziellen auch für die sinnvolle Anwendung für Prozessmodellen.

Das "Gemeine" an der Sache: Talent ist nicht übertragbar, das ist personengebunden. Und damit sind wir wieder zurück am Anfang: Diejenigen, die Agilität richtig verstehen und umsetzen, sind immer „Könner“, die anderen sind häufig die, die noch nicht „können“, sei es, dass ihnen Wissen, Training oder Talent fehlt.

# … und was bleibt

***Peter***: Was bleibt uns neben all der (völlig normalen) Verwirrung und den Missverständnissen an der Agilität erhalten?

***Paul***: Ich denke, egal wie es mit der Agilität weitergehen wird, sie hat wichtige Impulse gebracht, die überleben werden: Sie hat z.B. eine Re-Fokussierung auf die Wertschöpfung anstatt auf die Methode gebracht und hat die Möglichkeit einer vollständigen à priori Planung selbst mäßig komplexer Vorhaben in Frage gestellt.

So macht Agilität klar, dass alle Beteiligten während eines Projektes dazulernen werden und es keinen Sinn macht, einen armen Menschen am Anfang irgendwo hinzusetzen, damit er alle Anforderungen vorab als Hochzeitsfrage "... der spreche jetzt oder schweige für immer!" behandelt, insbesondere da das ja in der Regel unmöglich ist, wie bereits zuvor beschrieben.

Als Gegenentwurf dazu hat Agilität den möglichen Produktivitätsgewinn durch kurze Feedbackzyklen mit viel Kommunikation gezeigt. Man tut das, wovon man zum aktuellen Zeitpunkt weiß, dass es das wichtigste und richtigste im gegebenen Kontext ist, schaut sich das Ergebnis an, lernt daraus und macht den nächsten Schritt.

Agilität hat auch sinnvollere Preismodelle salonfähig gemacht. Klassischerweise sehen viele Preismodelle so aus: ich mache eine Ausschreibung, in die ich hineinschreibe, dass der Preis zu einem großen Teil den Zuschlag bestimmt.

So hat sich ein System etabliert, in dem jeder weiß, dass man nur durch Change Requests Marge machen kann. Also bietet man zum Dumping-Preis an und setzt dann einen abgebrühten Projektmanager auf das Projekt, der sich jede Kleinigkeit - von denen es erfahrungsgemäß im Laufe des Projekts jede Menge geben wird - in Form von saftigen Change Requests versilbern lässt.

Agile Projekte versuchen hingegen, den Mehrwert für den Kunden in das Zentrum ihrer Betrachtungen zu stellen. So wird dann beispielsweise statt exakter Anforderungen ein "Leistungsversprechen" formuliert. Damit kann man auch weiterhin alle Aspekte festmachen, die einem Kunden wichtig sind: Zeit, Budget, Qualität und Scope. Nur ist der Scope dann kein dickes Pflichtenheft mehr, sondern ein Versprechen des Dienstleisters, dass er in der Zeit eine bestimmte Arbeit erbringt mit dem Ziel, für den Kunden den maximalen Mehrwert zu erzeugen.

Womit diese Arbeit im Detail ausgefüllt wird, kann der Kunde steuern. Durch kurze Iterationen sieht der Kunde zum einen schnell, ob der Dienstleister sein Leistungsversprechen auch hält und er hat zusätzlich noch die Möglichkeit, neue oder veränderte Anforderungen, z.B. auf Basis von Erkenntnissen aus der letzten Iteration einzubringen. Der Kunde hat wie zuvor Sicherheit in allen Dimensionen und erhält den maximalen Mehrwert für sein eingesetztes Geld, wenn denn Kunde und Dienstleister ehrlich sind und sich gegenseitig vertrauen … aber das hatten wir schon.

# Fazit

Hier möchten wir die Diskussion von Peter und Paul verlassen. Sie haben viele der genannten Aspekte noch wesentlich intensiver diskutiert und auch noch diverse weitere Aspekte wie z.B. das Zusammenspiel von Architektur und Agilität (Stichwort: „Emergenz“). Das würde dann aber den Rahmen dieses Artikels bei weitem sprengen.

Wir hoffen, wir konnten Ihnen für Ihre Überlegungen und Diskussionen rund um das Thema Agilität ein paar neue Ideen und Anregungen jenseits der ausgetretenen Diskussionspfade liefern und Ihnen helfen, das Thema in einem etwas größeren Kontext zu verankern.

Natürlich waren es nur einige ausgewählte Aspekte, die wir hier diskutiert haben und es gibt noch viel, viel mehr dazu zu sagen, aber wir hoffen, dass die wichtigsten Ideen herübergekommen sind:

* Agilität ist keine „Silver Bullet“, aber sehr gut für den Umgang mit komplexen Situationen geeignet, wie man sie in heutigen IT-Projekten typischerweise antrifft.
* Agilität in Großprojekten ist problematisch, insbesondere wegen der typischerweise sehr ausgeprägten politischen Dimension in Großprojekten, die Agilität aufgrund einer fehlenden Vertrauenskultur unmöglich macht.
* Es gibt nach wie vor in vielen Projekten Probleme mit dem Verständnis oder der Anwendung von Agilität. Das liegt dann in der Regel aber nicht an der Agilität an sich, sondern an viel allgemeineren Problemen, die uns häufig schon die gesamte IT- bzw. Menschheitsgeschichte begleiten.
* Viel wertvolles und essentielles Wissen geht verloren, wenn man sich nicht selbst intensiv mit dem Thema auseinandersetzt und sich stattdessen auf Meinungen und Zusammenfassungen anderer verlässt, wie man schön am Schicksal des Wasserfall-Modells sehen kann: In seiner Originalfassung hatte er viele interessante Aspekte wie Feedback-Zyklen und den involvierten Kunden. All dieses wertvolle Wissen ist aber verlorengegangen und am Ende blieb das übrig, was heute so häufig als Feindbild der Agilität herhalten muss.

Wenn wir Ihnen also nur einen Rat im Zusammenhang mit Agilität geben können: Vertrauen Sie niemandem blind (auch uns nicht), sondern machen Sie sich Ihr eigenes Bild. Agilität hat so viele Facetten, die passen nicht in einen Artikel, eine Präsentation und schon gar nicht auf einen Bierdeckel … ;-)

# Literatur & Links

**[App]** J. Appelo, „ScrumButs Are the Best Part of Scrum“, siehe <http://www.noop.nl/2009/09/scrumbuts-are-the-best-part-of-scrum.html>

**[Bas03]** V. Basili, C.Larman, Iterative and Incremental Development – A Brief History, IEEE Software, 2003

**[Bro95]** F. Brooks, The Mythical Man Month, Addison Wesley 1995

**[Gal02]** J. Gall, Systemantics – The Systems Bible, 3rd Edition, General Systematics Press, 2002

**[Kru03]** P. Kruchten, The Rational Unified Process, Addison Wesley, 2003

**[Kru07]** P. Kruchten, Voyage in the Agile Memeplex, ACM Queue, 2007

**[Kru08]** P. Kruchten, Podcast LPZ IT-Radar, Universität Leipzig, 2008

**[Man]** M.L. Manns, L. Rising, Fearless Change: Patterns for Introducing New Ideas, siehe http://www.cs.unca.edu/~manns/intropatterns.html

**[Roy70]** Winston Royce, Managing the Development of Large Software Systems, 1970, IEEE WESCON (a.k.a Wasserfall)

**[Sho]** James Shore, The Decline and Fall of Agile, siehe <http://jamesshore.com/Blog/The-Decline-and-Fall-of-Agile.html>

**[Weg95]** P. Wegner, Interactive Foundations of Object-Based Programming, IEEE Computer 28(10): 70-72, 1995

**[Weg97]** P. Wegner, Why Interaction Is More Powerful Than Algorithms, Communications of the ACM 40(5): 80-91, 1997