

Das „agile“ Vorgehen: Neuer Wein in alte Schläuche - oder ein „Déjà-vu“?

Hans-Peter Korn

KORN AG
Turnweg 13
CH 5507 Mellingen
contact@korn.ch

Abstract: Nach dem Hinterfragen essentieller Hintergründe des agilen Vorgehens (Planbarkeit, Komplexität, Selbstorganisation, Kommunikation, Vertrauen) werden die unterschiedlichen Sichtweisen von „agil“ dargestellt und heute verbreitete agile Konzepte und Praktiken und deren historische Wurzeln zum Entwickeln von Software, für das Management von Projekten, zur Neu- und Weiterentwicklung und Wartung von Produkten und die insgesamt agile Organisation diskutiert.

1 Einleitung

Agil zu sein ist unbestritten attraktiv: Wer will denn nicht - gemäß dem Duden-Fremdwörterbuch - „behände, flink, gewandt, regsam, geschäftig“ sein? Wer will als das Gegenteil von agil im Sinn seines lateinischen Ursprungs (agilis, abgeleitet von agere) gesehen werden, als nicht handelnd oder unbeweglich? Langsamkeit und Müßiggang (als sprichwörtlicher Anfang aller Laster) sind spätestens seit Martin Luther verpönt. Er schrieb: „*Von Arbeit stirbt kein Mensch, aber von Ledig- und Müßiggehen kommen die Leute um Leib und Leben; denn der Mensch ist zum Arbeiten geboren wie der Vogel zum Fliegen.*“

Wieso aber, obwohl „behände, flink, gewandt, regsam, geschäftig“ immer schon erstrebenswert war, ist Agilität in den letzten rund zehn Jahren zu einem Modebegriff geworden? Was macht „Agilität“ heute noch viel mehr als früher so attraktiv - oder sogar unabdingbar?

Einige Hinweise sind im seit 2009 vom amerikanischen Deloitte's Center for The Edge jährlich herausgegebenen Shift Index nachzulesen [Ha11]. Im Shift Index wird seit 1965 bis heute die Performance von 20.000 US-Unternehmen analysiert.

Der Return on Assets (ROA) sinkt seit Jahrzehnten kontinuierlich und ist heute nur noch knapp über null. Die Volatilität der Aktienkurse nimmt zu. Die topple rate als Maß für den Verlust der Marktführerschaft großer Firmen hat sich mehr als verdoppelt. Die Treue der Kunden ist einer hohen Wahlbereitschaft gewichen. Produktinnovationen sind nicht mehr das Privileg hoch entwickelter Ökonomien, sondern werden mit immer rasan-

terem Tempo im asiatischen und pazifischen Raum entwickelt, zwischen 1995 und 2006 haben die Wechsel der Chief Executive Officers (CEO) um 59% zugenommen, die durch mangelnde Performance begründeten sogar um 318%.

Und ein Drittel der 1970 in Fortune gelisteten 500 umsatzstärksten Unternehmen der Welt existierte 1983 nicht mehr. Von den 1917 von Forbes genannten 100 größten Firmen existierte 2001 nur noch eine, nämlich General Electric.

Deloitte's „Shift Index 2011“ fasst das so zusammen: *„Surveying today's business landscape, perhaps investors intuitively grasp that "normal" is a thing of the past—that we have entered a world that does not stabilize as easily as it once might have.“*

Das „agile“ Vorgehen wird heute - losgelöst von der SW-Entwicklung als solche - als Antwort auf diese Herausforderungen gesehen. Dieser Beitrag beschreibt die unterschiedlichen Sichtweisen dessen, was heute unter „agil“ verstanden wird und widmet sich dann typischen Konzepten und Praktiken des agilen Vorgehens. Dabei wird gezeigt, dass Vieles davon bereits seit Jahrzehnten bekannt ist und auch - allerdings vergleichsweise selten - praktiziert wird.

Im Abschnitt *„Agiles“ Vorgehen als Lösung?* beschäftigt sich der Beitrag zunächst mit einigen essentiellen Hintergründen der Agilität (Planbarkeit, Komplexität, Selbstorganisation, Kommunikation, Vertrauen) und beschreibt danach in vier Gruppen die unterschiedlichen Sichtweisen von „agil“:

- Agil = Flexibilität und Adaptivität mittels „Empirical Process Control“
- Agil = Flexibilität und Adaptivität (Empirical Process Control) plus „Lean Management Principles“
- Agil = Flexibilität und Adaptivität (Empirical Process Control) plus „Lean Management Principles“ verbunden mit einer Sammlung „post-tayloristischer“, hierarchie- und autoritätsfreier, partizipativer, selbststeuernd-kollaborativer, kommunikationsbasierter, systemischer Praktiken und „Glaubenssätze“
- Agil = Aspekte zur Sicherung der Überlebens- und Entwicklungsfähigkeit unabhängig vom Empirical Process Control

Im Abschnitt *Heute verbreitete agile Konzepte und Praktiken* werden Konzepte und Praktiken

- des agilen Entwickelns von Software,
- des agilen Managements von Projekten,
- der agilen Neu- und Weiterentwicklung und Wartung von Produkten

und als Abschluss die insgesamt agile Organisation vorgestellt.

2 Agiles Vorgehen als Lösung?

Wie geht das agile Vorgehen mit den in der Einleitung dargestellten und von der reinen SW-Entwicklung unabhängigen Herausforderungen um?

Im Kern beruht es - in der einen Sichtweise - auf einer im Gegensatz zur uns vertrauten tiefgehend analysierenden, dann in Detail planenden und schlussendlich plangemäß handelnden Herangehensweise auf einem empirischen, betont kommunikations- und kooperationsorientierten und vertrauensbasierten Arbeiten. Es geht dabei darum, die von uns wahrgenommene Komplexität (im Sinn einer grossen Unsicherheit bezüglich der Vorhersehbarkeit zukünftiger Situationen und Ereignisse) zu akzeptieren und mit ihr konstruktiv umzugehen statt sie beherrschen zu wollen.

In einer anderen Sichtweise geht es dabei darum „Selbstorganisation“ zu verstehen als Ergebnis der fortlaufenden, auch das eigene Handeln als Individuum und Team reflektierende, Kommunikation; und darum, Vertrauen als wesentliche Essenz funktionsfähiger sozialer Systeme zu erkennen und aufzubauen.

Vor diesem Hintergrund werden dann die verschiedenen Ausprägungen von „agil“ beschrieben:

- Flexibilität und Adaptivität mittels „Empirical Process Control“ stellt die kurzen Iterationen verbunden mit einem fortlaufenden Anpassen und Lernen ins Zentrum und reicht in der SW-Entwicklung zurück in die 1970er-Jahre.
- Flexibilität und Adaptivität (Empirical Process Control) plus „Lean Management Principles“ ergänzt diese Sichtweise um die vom „Toyota Way“ begründeten Prinzipien des Lean Management.
- Eine weitere Ausprägung umfasst zusätzlich noch etliche Praktiken und „Glaubenssätze“ der Organisationsgestaltung der letzten Jahrzehnte geprägt von post-tayloristischen, hierarchie- und autoritätsfreien, partizipativen, selbststeuernd-kollaborativen und systemischen Konzepten.
- Aspekte zur Sicherung der Überlebens- und Entwicklungsfähigkeit bilden eine weitere Gruppe jener Ausprägungen von „agil“ die unabhängig vom „Empirical Process Control“ sozialen Systemen i.a. dienen können.

Diese Auslegeordnung zeigt, dass der Begriff „agil“ so vieldeutig ist, das er stets einer zum jeweiligen Kontext passenden Präzisierung bedarf. Eigentlich sollte daher auf den Gebrauch dieses Begriffes verzichtet werden und statt dessen von dem gesprochen werden, was jeweils konkret gemeint ist.

2.1 Mit Unberechenbarkeit und Komplexität umgehen statt sie beherrschen zu wollen

Die unbegrenzt erscheinende Menge der sich rasch verändernden und ständig wachsenden Informationen und die uns heute spontan mögliche weltumspannende Vernetzung

nehmen wir als extrem komplex wahr. In immer mehr bislang einigermaßen voraussehbaren Lebensbereichen werden Planungen zunehmend zu unsicheren Prognosen oder gar nur zu Hoffnungen.

Es ist dann wie beim Überqueren eines frisch zugefrorenen und unbekanntes Flusses, um die schemenhaft im Nebel am gegenüberliegenden Ufer erkennbare einfache, aber - vermutlich - gut beheizte Hütte zu erreichen.

Die eine Vorgehensweise ist:

Vorsichtig, Schritt für Schritt, bei jedem Knistern die Richtung ändernd, versuchen wir die Überquerung. Und etwa in der Mitte lichtet sich der Nebel etwas und einige hundert Meter neben der Hütte erscheint ein sicher viel angenehmeres Hotel. Und, vorsichtig, Schritt für Schritt, ändern wir unsere Richtung in Richtung Hotel.

Eine andere Vorgehensweise wäre:

Zunächst mit speziellen Wärmebild-Technologien die Beheizung der Hütte überprüfen. Wenn sie geeignet erscheint, wird sie als verbindliches Ziel definiert. Dann wird anhand der Klimawerte der letzten Wochen und der dokumentierten Erfahrungen der letzten Jahre die Eisdicke berechnet und mit Würfeln von Steinen bekannten Gewichts und berechneter Ballistik stichprobenartig verifiziert. Dann wird der beste Weg über das Eis zur Hütte geplant. Und dann wird dieser Weg, ausgerüstet mit Schwimmweste und einer langen Leiter, gemäß Plan zügig und nach Plan bewältigt.

Diese zweite Vorgehensweise entspricht einem von der technologischen Analysierbarkeit, Machbarkeit und Planbarkeit bestimmten Denken und ist Basis aller „plangetriebenen“ Vorgehensmethoden der Produktentwicklung und des Projektmanagements auf der Grundlage eines möglichst frühen Big Design Up Front (BDUF).

Die erste Vorgehensweise hingegen entspricht den Erkenntnissen im Umgang mit Complex Adaptive Systems, wie sie u. a. von Edwin E. Olson und Glenda H. Eoyang beschrieben werden [OEG01]. Geprägt ist diese Vorgehensweise von der Einsicht, dass uns Situationen als komplex dann erscheinen, wenn wir Beziehungen zwischen Ursache und Wirkung nur im Nachhinein erkennen können, nicht aber im Voraus. Unser Vorgehen beruht dann auf Handeln-Beobachten-Reagieren und emergenten Praktiken. Im Gegensatz dazu stehen die uns kompliziert erscheinenden Situationen mit - wenn auch sehr aufwendig - analysierbaren und voraussagbaren Ursache-Wirkungs-Beziehungen oder den von uns als simpel betrachteten Situationen mit für alle offensichtlich erscheinenden Ursache-Wirkungs-Beziehungen [KS03].

Bei dieser ersten - empirischen - Vorgehensweise verzichten wir bewusst auf eine unangemessen aufwendiges und noch dazu unzuverlässiges BDUF. Stattdessen planen wir nur jeweils das für die nächsten Schritte wirklich Erforderliche auf Basis der bisherigen Einsichten. Zu Beginn unseres Vorhabens gehen wir von einem eher groben Just Enough Design Up Front (JEDUF) aus.

Dieser Wechsel vom BDUF mit seiner Paukenschlag-Einführung (big bang) der kompletten Lösung zum JEDUF mit seinen häufigen Einführungen von aufeinander aufbauenden Lösungsteilen (Lösungsinkrementen) und den die weitere Entwicklung steuernden raschen Feedbacks zeigt sich exemplarisch in der Softwareentwicklung (als wesentlichem Treiber des agilen Vorgehens) in den letzten 30 Jahren, wie sie etwa von Brian Wernham im Kapitel 19 „The Lure of Big Design Up Front“ seines Buches Agile Project Management For Government [We12] gut beschrieben ist.

James Martin prägte ab dem Ende der 1980er-Jahre mit seinen Büchern [Ma90] die von sequenziellen und jeweils etliche Monate dauernden Phasen (Analyse, Konzept, Realisierung, Einführung) bestimmten „wasserfallartigen“, betont arbeitsteiligen und stark werkzeuggestützten Vorgehensweisen. Allerdings wies bereits 1970 Winston W. Royce im Artikel „Management of Large Software Systems“ in Proceedings IEEE WESCON August 1970 pp. 1 - 9 darauf hin, dass das in seinem Artikel in Fig.2 dargestellte „wasserfallartige“ Vorgehen ungeeignet sei und empfiehlt statt dessen in Fig.10 ein iteratives Vorgehen mit sehr vielen Rückkopplungsschleifen unter Einbezug auch des Benutzers. Dennoch haben sich seit den 1980er-Jahren in Anlehnung an die Prozesse zur Fertigung physischer Produkte Vorgehensweisen auf Basis streng getrennter sequentieller Phasen i.S. von Analyse, Konzept, Realisierung, Einführung etabliert und gipfeln in der Idee der „Software Factory“, also Software unter Außerachtlassen des fundamentalen Unterschieds zwischen „Design“ und „Make“ auf Basis industrieller Fertigungsprozesse zu entwickeln. [Po02]

Spätestens seit der Jahrtausendwende erleben wir jedoch einen immer deutlicheren Wechsel hin zu flexiblen, schlanken und stark teamorientierten Entwicklungsmethoden, die in jeweils wenigen Wochen verfügbare und konkret nutzbare Teilergebnisse liefern. Dabei wird im Bereich von Architektur und Design auf den intensiven Einsatz von Modellierungswerkzeugen verzichtet. Stattdessen erfolgt die Software-Realisierung (das Programmieren und Testen) zu einem möglichst hohen Grad toolunterstützt und - beim Testen - automatisiert.

Der Wechsel weg vom ab Beginn im Detail ingenieurmäßig analysierten, modellierten und durchgeplanten Vorgehen hin zum nur grob im Voraus abgesteckten und danach schrittweise angepassten inkrementell-adaptiven Vorgehen ist natürlich auch eine psychologische Herausforderung: In der Regel ist es vielen von uns wohler, wenn wir einem detaillierten Plan folgen können [Do12].

2.2 „Selbstorganisation“ als Lösung - oder als Missverständnis?

„Selbstorganisation“ gilt schlechthin als „die“ Alternative zu einer von „command and control“ geprägten, stark arbeitsteiligen, unflexiblen und hierarchischen „Organisation 0.0“ [Kr09]. Beim Hinterfragen der Bedeutung von „Selbstorganisation“ jedoch stellt sich diese bald als recht buntes Gemisch aus abstrakten Prinzipien der Systemtheorie, aus chaotisch und eigendynamisch gesehenen Naturprozessen, aus Evolutionsphänomenen bei Organismen, aus Mustern der Emergenz sozialer Strukturen, aus Übertragungen neoliberaler Mechanismen des freien Markts auf die Organisationseinheiten im Unter-

nehmen bis hin zu basisdemokratischen und anarchischen Vorstellungen heraus. Also: „Selbstorganisation ist Chiffre für einen Erklärungsnotstand, keine Erklärung.“ [Tu02]

Die Forderung nach Selbstorganisation kann überdies zu einem leistungspolitischen „double bind“ führen: „*Einerseits gehören nunmehr Selbstkoordination und kreative Problemlösung zum offiziellen Aufgabenkanon der Gruppe, andererseits fehlt Zeit und Personal, um diese Aufgaben angemessen erfüllen zu können.*“ [Wo03] Andere Stimmen sehen in der Förderung der Selbstorganisation im Unternehmen ein Mittel der Selbstdisziplinierung der Mitarbeitenden im Interesse der Profitmaximierung der Kapitalgeber und des Top-Managements: „*Management expects individuals in post-Fordist capitalism to be flexible, innovative, motivated, dynamic, modern, young, and agile, and it wants them to identify with the corporation and to have fun at work. Strategies of participatory management aim at the ideological integration of laborers into corporations. This is a new quality of the disciplinary regime that aims at a rise of profits by an increase in productivity and cost reductions achieved by the workers' permanent self-discipline*“ [Fu08]

2.3 Kommunikation und Vertrauen bilden soziale Systeme

Auf einem festeren Boden stehen wir, wenn wir den Begriff „Selbstorganisation“ vermeiden und uns statt dessen auf die Erkenntnisse der Theorie komplexer adaptiver Systeme, insbesondere sozialer Systeme, abstützen. Auf diesem Feld gewinnt dann ein anderer Begriff weitaus größere Bedeutung als jener der Selbstorganisation, nämlich „*Kommunikation*“. Gemäß Niklas Luhmann bildet allein Kommunikation - und nicht Personen - Systeme. Kommunikation versteht er als die kleinste Einheit eines Systems, wobei nicht Menschen (oder psychische Systeme) kommunizieren sondern Kommunikation kommuniziert und nur in sozialen Systemen stattfindet. [Lu84]. Ein großes Unternehmen stellt somit ein sehr komplexes Gebilde „kommunizierender Kommunikation“ dar. Damit wir Menschen diese umfassende Komplexität bewältigen können, so Luhmann, müssen wir sie reduzierend betrachten. Gemäss seiner Systemtheorie ist die Reduktion von Komplexität die Hauptaufgabe von sozialen Systemen (also auch von Organisationen). Nur so können wir Menschen überhaupt überleben. Ein wesentliches Element der Komplexitätsreduktion ist - gemäss Luhmann - das Vertrauen: *„Vertrauen ist stets in die Zukunft gerichtet. ... im Akt des Vertrauens (wird) die Komplexität der zukünftigen Welt reduziert... Vertrauen erschließt durch die Reduktion von Komplexität Handlungsmöglichkeiten, die ohne Vertrauen - nach Luhmann - unwahrscheinlich und unattraktiv geblieben und somit nicht zum Zuge gekommen wären“* [Ta09] [Lu00]

Ausgehend von Luhmann geht es also nicht so sehr darum, "besser" mit dieser sich durch den Verlust an Vertrauen verstärkt - oder gar als überfordernd - wahrgenommenen Komplexität so umzugehen, dass wir jederzeit flexibel und mit kurzfristig wechselnden Zielen in unserer Welt leben können sondern eher darum, wie wir wieder mehr Vertrauen zu Personen, Rollenträgern, Teams, Normen, Organisationen, Strukturen und Prozessen erlangen. Und Vertrauen erlangen wir durch direkte und persönliche Kommunikation und Kooperation.

Viel zu theoretisch und - was hat das mit „agil“ zu tun?

Sehr viel: Werfen wir doch einen Blick auf den ersten und dritten der vier „Werte“ des „Agile Manifest der Software-Entwicklung“: [Be01]

„... wir ((haben)) diese Werte zu schätzen gelernt:

1. *Individuen und Interaktionen mehr als Prozesse und Werkzeuge*
2. *Funktionierende Software mehr als umfassende Dokumentation*
3. *Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung*
4. *Reagieren auf Veränderung mehr als das Befolgen eines Plans*

Obwohl wir die Werte auf der rechten Seite wichtig finden, schätzen wir die Werte auf der linken Seite höher ein.“

Und beachten wir auch diese drei seiner zwölf „Prinzipien“:

- *Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten.*
- *Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteam zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.*
- *In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend an.*

All das bedingt und fördert Kommunikation. Und funktioniert nur auf Basis von „Vertrauen“ statt auf Basis einer alle Beteiligten überfordernden und noch dazu trägen granularen Steuerung via „Auftragserteilung von oben“.

Wenn auch all das unter „agil“ subsumiert wird - was genau bedeutet dann „agil“?

Und: Ist „agil“ im Rahmen der etablierter Führungsstrukturen und Vorgehensweisen nicht wie „neuer, noch gärender, Wein in alte Schläuche“? Oder ist es gar nicht so „neu“ - sondern ein „Déjà-vu“?

2.4 Die Vieldeutigkeit des Begriffes „agil“

AGIL erscheint in den 1950er-Jahren beim amerikanischen Soziologen Talcott Parsons als Akronym für vier überlebenswichtige Funktionen lebender und somit auch sozialer Systeme [Pa12]:

A: Anpassung (Adaptation) an die Umwelt

G: Zielerreichung (Goal Attainment)

I: Integration als Mechanismus zur Leistungserbringung der Teilsysteme untereinander

L: Strukturhaltung oder Latenz als Mechanismus zur Erhaltung der Identität des Systems, obwohl alles stetig im Wandel ist.

Im 1997 erschienenen Buch „Komplexität und Agilität: Steckt die Produktion in der Sackgasse?“ beleuchtet eine Reihe von Beiträgen „... . die Widersprüche zwischen Komplexität im Produktionsumfeld und Agilität im Markt.. . und [gibt] Perspektiven für deren Auflösung. Insbesondere wird den Fragen nachgegangen, was Management heute und in Zukunft bedeutet, wohin sich die Märkte bewegen werden und wie man ihnen folgt, wie die technischen Innovationen aussehen und was sie bewirken werden und wie sich die industrielle Produktion durch neue Technologien und Organisationsprinzipien ändern wird.“ [SW97]

Im Februar 2001 trafen sich 17 Vertreter der damals „leichtgewichtig“ genannten Methoden der Softwareentwicklung und formulierten das sie Verbindende als die vier Werte und zwölf Prinzipien des Agilen Manifests der Softwareentwicklung [Be01], auf das sich auch heute noch alle agilen Vorgehensweisen der Softwareentwicklung implizit oder explizit berufen.

Das war auch der Ausgangspunkt für die rasche Verbreitung des Begriffs „agil“ und seine Ausdehnung auf andere Bereiche außerhalb der SW-Entwicklung - und auch der Differenzierung seiner Bedeutung.

Heute können im Kontext der Produktentwicklung und des Projektmanagements, des Managements und der Organisationsgestaltung folgende vier Gruppen der Bedeutung von agil unterschieden werden:

2.4.1 Agil = Flexibilität und Adaptivität mittels „Empirical Process Control“:

Kennzeichnend dafür ist ein Vorgehen in kleinen, stets gleich langen Schritten, ausgehend von einem möglichst leichtgewichtigen Just Enough Design Up Front. Jeder Schritt liefert dabei ein - wenn immer möglich bereits praktisch nutzbares - Teilergebnis, das Grundlage für Feedbacks zur Gestaltung des nächsten Schritts ist. Das Vorgehen in jedem Schritt wird im Team reflektiert, um kontinuierliche Verbesserung, also Lernen, zu erreichen.

Für Tom Gilb, der bereits in den 1970er-Jahren für ein evolutionäres IT-Projektmanagement eintrat, besteht die zentrale Bedeutung von „agil“ in diesem inkrementell-adaptiven Vorgehen und kontinuierlichem Lernen. Und er meint, dass alle anderen agilen Taktiken optionale Details seien [Po02].

Diese Sicht entspricht auch der von Volker Nissen. Er unterscheidet dabei zwischen einer „kapazitiven Agilität“ als Fähigkeit der IT, auf schwankende mengenmäßige Anforderungen schnell antworten zu können (Skalierbarkeit, Performanz), und einer „funktionalen Agilität“, die sich auf die funktionalen Anforderungen bezieht. Bei vorhersehbaren Anforderungsänderungen spricht er von „reaktiver Agilität“. Wenn die IT jedoch eine aktive Rolle für das fachliche Geschäft übernimmt und unvorhergesehene Änderungen und IT- Innovationen aufspürt, spricht er von „proaktiver Agilität“ [NM09].

2.4.2 Agil = Flexibilität und Adaptivität (Empirical Process Control) plus „Lean Management Principles“:

Zusätzlich zur obigen Bedeutung werden hier die Prinzipien des Lean Management als eigentliche Basis von „agil“ gesehen, oft dargestellt als House of Lean, ausgehend von Larman und Vodde (2009) und The Toyota Way (2004) [Le09].

Das Ziel „Wert schaffen“ als Dach des House of Lean ist gekennzeichnet durch:

- Kurze Durchlaufzeiten
- Beste Qualität und höchsten Wert
- Höchste Kundenzufriedenheit
- Niedrigste Kosten
- Hohe Moral
- Sicherheit

und wird getragen von den zwei Säulen:

- Respekt für alle Personen
- Kontinuierliche Verbesserung

Erreicht wird das Ziel durch spezifische Entwicklungsmethoden, basierend auf 14 Lean Principles.

Dieser Begriff von agil wird vertreten auch in der Definition von Agilität gemäß Lexikon IT-Management [Lex10].

2.4.3 Agil = Flexibilität und Adaptivität (Empirical Process Control) plus „Lean Management Principles“ verbunden mit einer Sammlung „post-tayloristischer“, hierarchie- und autoritätsfreier, partizipativer, selbststeuernd-kollaborativer, kommunikationsbasierter, systemischer Praktiken und „Glaubenssätze“:

Dieser Begriff von agil erscheint auch im IT-unabhängigen Kontext des Managements und der Organisationsgestaltung. Das im Agile Enterprise Adaptation Program der Agile Alliance entwickelte Modell ist ein Beispiel für diesen Begriff von agil [Co12].

Auch das bereits weiter oben erwähnte Agile Manifest der Softwareentwicklung [Be01] kann dieser Gruppe zugerechnet werden. Seine vier Werte und die weiteren zwölf Prinzipien sind, dem Charakter eines Manifests entsprechend, jedoch als Appell zu verstehen und nicht als verifizierte Arbeitsregeln. Sie dogmatisch als wortwörtlich zu erfüllende Handlungsanweisungen für die Software- oder Produktentwicklung in unterschiedlichsten Situationen zu betrachten wäre eine Fehlnutzung [JS12].

2.4.4 Agil = Aspekte zur Sicherung der Überlebens- und Entwicklungsfähigkeit unabhängig vom Empirical Process Control:

Eine ganz andere, nicht auf die Software- und Produktentwicklung, sondern die Führung sozialer Systeme in risikoreichen und komplexen Missionen ausgerichtete Sicht von agil findet sich in „Power to the Edge“, einer Publikation des Command and Control Research Program des US Department of Defense [AH09]:

- *Robustheit*: die Fähigkeit, aufgaben-, situations- und bedingungsübergreifend effektiv zu bleiben;
- *Belastbarkeit*: die Fähigkeit, sich von Unglücksfällen, Schäden oder einer destabilisierenden Störung der Umgebung zu erholen oder sich darauf einzustellen;
- *Reaktionsfähigkeit*: die Fähigkeit, auf eine Veränderung der Umgebung rechtzeitig zu reagieren;
- *Flexibilität*: die Fähigkeit, mehrere Lösungsmöglichkeiten einzusetzen und nahtlos von einer zur anderen überzugehen;
- *Innovationsfähigkeit*: die Fähigkeit, neue Dinge zu tun und die Fähigkeit, alte Dinge auf eine neue Art und Weise zu tun;
- *Anpassungsfähigkeit*: die Fähigkeit, Arbeitsprozesse zu ändern und die Fähigkeit, die Organisation zu ändern.

Diese Definition versteht unter Flexibilität im Gegensatz zu anderen Sichtweisen von agil nicht nur ein inkrementell-adaptives Vorgehen in kleinen Schritten, ausgehend von einem JEDUF, sondern lässt etwa auch ein stark plangetriebenes, BDUF-basiertes Vorgehen dann zu, wenn es für die aktuelle Situation passender und effizienter ist. Entscheidend dabei sind jedoch die Fähigkeiten, mehrere Lösungsmöglichkeiten einzusetzen und nahtlos von einer zur anderen überzugehen, auf eine Veränderung der Umgebung rechtzeitig zu reagieren und Arbeitsprozesse und die Organisation zu ändern.

Zu dieser Bedeutungsgruppe gehört auch das eingangs erwähnte Akronym „AGIL“ von Talcott Parsons.

3 Heute verbreitete agile Konzepte und Praktiken

Wenn heute von „agil“ gesprochen wird, wird es oft mit Scrum gleichgesetzt. Blicken wir jedoch rund ein Jahrzehnt zurück.

Damals, 2002, war in einem Artikel von Jens Coldewey das zu lesen: „*Noch immer wird agile Entwicklung in weiten Kreisen gleichgesetzt mit eXtreme Programming. Dadurch wird die Chance vergeben, auch von den anderen Verfahren zu lernen*“ [Co02]. Heute - rund ein Jahrzehnt später - wäre in diesem Artikel „eXtreme Programming“ durch „Scrum“ zu ersetzen. Und dadurch wird - auch heute - die Chance vergeben, von den anderen derzeit verbreiteten weiteren agilen Verfahren zu lernen.

Jens Coldewey behandelte 2002 im oben zitierten Artikel sechs agile Konzepte und Praktiken:

Drei Meta-Prozesse, wie das Team zu einem individuellen Prozess kommt:

- Adaptive Software Development,
- die Crystal-Methodenfamilie und
- Scrum.

und drei Konzepte, die recht konkrete Verfahren und Techniken, die dann im Laufe des Projekts angepasst werden können, vorschlagen:

- Dynamic Systems Development Method (DSDM),
- eXtreme Programming (XP) und
- Feature Driven Development (FDD).

Dominant im Jahr 2002 war XP. Heute, ein Jahrzehnt später, dominiert Scrum (allerdings zum größten Teil in diversen Abwandlungen die - gemäß „Scrum Guide“ [SS] - eigentlich nicht „Scrum“ genannt werden sollten) die agile Landschaft. Das zeigen übereinstimmend drei unabhängige Studien aus dem Jahr 2012 [Sw12] [KM12] [Ko12]

Gemäß diesen Studien setzten 2012 bei den antwortenden Firmen - je nach Studie - 51 bis 78 % agile Vorgehensweisen (oft parallel zu klassischen) ein. Davon nutzen 51 bis 84,5 % Scrum (bzw. ein an Scrum orientiertes Vorgehen); es steht damit in allen diesen Studien an erster Stelle. Bei zwei dieser drei Studien stand das vor zehn Jahren noch unbekannte Kanban mit 17 bzw. 43 % an zweiter Stelle, bei der dritten Studie mit 4,5 % an fünfter Stelle. Die Verbindung von Scrum mit Kanban (Scrumban) hatte eine Verbreitung von immerhin bereits 3 bis 8,5 %. Das vor zehn Jahren dominierende eXtreme Programming (XP) stand mit 5,5 bis 33 % Verbreitung in allen Studien nur noch an vierter Stelle, wobei allerdings viele der ursprünglichen XP-Praktiken (wie z. B. User Stories) heute als integraler Bestandteil von Scrum gesehen und praktiziert werden.

Weitere verbreitete - von den Befragten als „agil“ betrachtete - Vorgehensweisen waren 2012 u. a. IBM Rational Unified Process (RUP), Agile Unified Process, Open Unified Process, Feature Driven Development, Usability Driven Development und unterschiedliche hybride Vorgehensweisen.

Die vor rund zehn Jahren noch recht bekannte Vorgehensweise Adaptive Software Development und die Crystal-Methodenfamilie haben heute kaum noch und das Feature Driven Development (FDD) hat nur noch eine begrenzte Verbreitung und die Dynamic Systems Development Method (DSDM/DSDM Atern) ist nach wie vor in Großbritannien - auch bei Großprojekten von Behörden - sehr bekannt. Hingegen stoßen heute weitere etablierte Vorgehensweisen mit immer schon iterativen Elementen wie der Rational Unified Process (RUP) und PRINCE2 (Projects in Controlled Environments) in den Bereich der agilen Vorgehensweisen vor, indem sie ihre iterativen Elemente verstärken. Auch bislang an ausgeprägten sequenziellen Phasen orientierte klassische Vorgehens-

weisen wie das Schweizer HERMES und das V-Modell XT öffnen sich für eine Kombination mit inkrementell-adaptiven Vorgehensweisen zumindest im Sinn eines hybriden Vorgehens.

Interessant sind die in der Swiss Agile Study 2012 erhobenen Auswirkungen. Die Ergebnisse auf die Frage „How has agile software development influenced the following aspects?“ sind in [KM12] auf der Seite 27 dargestellt. Wenn man die Prozent-Werte für „much worse“ „worse“ und „unchanged“ zusammenzählt ergibt sich dieses Bild:

	% much worse + worse + unchanged	
Ability to manage changing priorities	10.0	
Development process	19.0	
Time to market	22.5	
Alignment between IT & business objectives	26.0	
Project visibility	26.5	
Team morale	29.5	
Requirements management	31.5	
Productivity	35.0	
Risk management	37.5	
Engineering discipline	45.5	
Management of distributed teams	46.5	28.5% Don't know
Software quality	47.0	
Software maintainability / extensibility capability	62.0	
Development Cost	65.2	

Demnach blieben aus Sicht der Mehrheit der Antwortenden bei der agilen SW-Entwicklung die Merkmale „Software maintainability / extensibility capability“ und „Development Cost“ unverändert oder haben sich sogar verschlechtert oder sehr verschlechtert. Deutlich verbessert haben sich hingegen die Merkmale: Ability to manage changing priorities, Development process, Time to market, Alignment between IT & business objectives, Project visibility, Team morale, Requirements management, Productivity, Risk management.

In den nächsten zehn Jahren wird sich die agile Methodenlandschaft sicher auch weiterhin erheblich verändern. Wohin die Reise geht, ist schwer abzuschätzen. Möglicherweise wird es dann agile Vorgehensweisen geben, an die wir heute noch gar nicht denken . . . wie im Jahr 2002 in der Softwareentwicklung noch niemand an Kanban dachte. Oder vielleicht wird statt „agil“ ein anderer Begriff in aller Munde sein.

Im Folgenden werden heute aktuelle Konzepte und Praktiken des agilen Entwickelns von Software (mit eXtreme Programming als typischen Vertreter), des agilen Managements von Projekten (mit PRINCE2 und DSDM Atern als typische Vertreter) und der agilen Neu- und Weiterentwicklung und Wartung von Produkten (mit Scrum, Kanban und SAFe als typische Vertreter) vorgestellt. Den Abschluss bilden einige Gedanken zur insgesamt agilen Organisation.

3.1 Konzepte und Praktiken des agilen Entwickelns von Software

Begonnen hat das inkrementell-adaptive Vorgehen zur Lieferung praktisch nutzbarer Software-Lösungen im Abstand von nur wenigen Wochen bereits in den 1970er-Jahren, gefordert u. a. durch Tom Gilb. Also noch vor dem Entstehen des stark plangetriebenen Vorgehens mit seinem BDUF in den 1980er-Jahren. Diese frühen inkrementell-adaptiven Vorgehensweisen wie etwa das Rapid Application Development (RAD) wurden jedoch als eher chaotisch empfunden und die 1997 erstmals publizierte Dynamic Systems Development Method (DSDM) - damals noch eine reine Software-Entwicklungsmethode - hatte das Ziel, hier etwas mehr Disziplin zu schaffen.

Typisches Beispiel für das agile Entwickeln von Software ist eXtreme Programming (XP), siehe Kap. 3.1.1 in [Ko] und [Wd09a] [Wd09b] [Be99].

XP geht davon aus, dass Software nicht auf der Basis bereits zu Beginn möglichst vollständig erhobener Anforderungen realisiert und danach eingeführt werden kann, sondern dass sich die Anforderungen kontinuierlich und differenzierter erst im Verlauf der schrittweisen Realisierung einzelner Softwareteile (Inkremete) ergeben. Diese Software-Inkremete müssen vom Benutzer getestet oder - besser - praktisch genutzt werden können. Daraus ergeben sich dann weitere und noch spezifischere Anforderungen. Jedes Software-Inkrement wird dabei innerhalb einer Iteration von einer bis maximal vier Wochen realisiert. Eine möglichst enge Zusammenarbeit der Benutzer und der Software-Entwickler und eine direkte Kommunikation sind dabei unabdingbar. So etwa werden die Anforderungen aus Benutzersicht nur als grobe Anwendungsfälle mit wenigen Sätzen als User Stories beschrieben. Sie sind keine umfangreichen und verbindlichen Detailspezifikationen, sondern dienen als Diskussionsanstöße für den Benutzer und Entwickler dann, wenn die Anforderung unter Mitwirkung des Benutzers konkret realisiert werden soll.

3.2 Konzepte und Praktiken des agilen Managements von Projekten

Gibt es überhaupt ein „agiles“ Management von Projekten?

Im Abschnitt *„Mit Unberechenbarkeit und Komplexität umgehen statt sie beherrschen zu wollen“* wurden die beiden gegensätzlichen Vorgehensweisen beim Überqueren eines zugefrorenen Flusses beschrieben. Beim sich schrittweise vorantastenden Vorgehen ergab sich mitten auf dem Weg eine neue Richtung hin auf ein bequemerer Hotel, statt wie ursprünglich geplant zur Hütte. Wäre das Überqueren des Flusses ein Projekt, dann hätte sich das ursprüngliche Ziel deutlich verändert und vermutlich auch die Zeit für die Überquerung des Flusses. Stabil wären nur die eingesetzten Ressourcen (die Anzahl der mitwandernden Personen) und die Qualität geblieben (den Fluss sicher überqueren und eine möglichst angenehme Stube finden). Würden wir auch die verfügbare Zeit begrenzen, dann könnte es sein, dass noch auf dem Fluss einige Meter vor dem Ufer angesichts des Hotels die Überquerung zum Stillstand käme. Wir müssen also noch ein paar Weginkremente (also zusätzliche Zeit für die mitwandernden Personen) investieren, um dieses neue, aber bessere Ziel zu erreichen. Aus klassischer Projektsicht wäre das ein typisches aus dem Ruder gelaufenes Projekt, obwohl das Ergebnis schließlich besser ist als das ursprünglich geplante.

Die Flexibilität bezüglich des Ziels (oder Funktionsumfangs) der Produktentwicklung und allenfalls auch der Anzahl der erforderlichen Produktinkremente (die gesamte Durchlaufzeit) ist essentiell für das agile Vorgehen. Das „eiserne Dreieck“ des klassischen Projektmanagements (fixer Umfang der Ergebnisse, fixe Kosten, fixe Zeit) wird damit ersetzt durch seine inkrementell-adaptive Variante (fixe Qualitätsansprüche, fixe Kosten, fixe Zeit, flexibel daran angepasste Ergebnisse). Auf vertraglicher Ebene passt zu dieser Variante der vielerorts beschriebene „Agile Festpreis“.

All das entspricht jedoch schlecht einem Projekt im üblichen Sinn. Ayelt Komus schreibt dazu [Ko12]:

„Agile Methoden wie Scrum sind keine Projektmanagementmethoden im eigentlichen Sinne. Ein Projekt, bspw. nach DIN 69901, ist durch seine „Einmaligkeit der Bedingungen in ihrer Gesamtheit“ gekennzeichnet. Weiterhin werden Projekten klare Ziele und begrenzte zeitliche und finanzielle Ressourcen zugeschrieben. Scrum als besonders populäre agile Methode hingegen zeichnet sich eben dadurch aus, dass es einen festen Rhythmus gleicher Dauer und gleicher Personalressourcen anstrebt. Die Ziele werden laufend weiterentwickelt ... Im Gegensatz zum klassischen Projektmanagement werden die jeweils im Fokus befindlichen Ziele an den Ressourcen ausgerichtet und nicht umgekehrt.“

Und in einem Interview zu dieser Studie sagt er: [Om12]:

„Interessanterweise werden oft Aufgabenstellungen als Projekte verstanden, die im eigentlichen Sinne gar keine sind. Schließlich ist mit der Einführung eines Softwareproduktes, eines Geschäftsprozesses, eines neuen Autos oder einer neuen SAP-Lösung die Arbeit nicht abgeschlossen. Vielmehr geht es um kontinuierliche Verbesserung der Lösungen und der Nutzung derselben. Wahrscheinlich ist das eines der Hauptprobleme klassischer Projektmanagementmethoden. Sie beruhen auf dem Gedanken: ‚Es gibt eine Fertigstellung und dann ist die Arbeit getan.‘ Außerdem verleitet diese Denkweise dazu, viel zu große Aufgabenstellungen in einem Stück zu bearbeiten.“

Dennoch, so schreibt Ayelt Konus in der Langfassung der Studienergebnisse [Ko12], finden „*agile Methoden Eingang in das Projektmanagement - oft auch als Ergänzung oder Erweiterung in Form eines sogenannten hybriden Ansatzes', also einer vermischten bzw. kombinierten Form agiler und klassischer Methoden. An vielen Stellen werden eigentlich kontinuierliche Prozesse, deren Ziele und Ressourcen eben nicht zu Beginn feststanden, als Projekte gemanagt bzw. bezeichnet.*“

Typische Beispiele für das agile Management von Projekten sind

- PRINCE2™, siehe Kap. 3.2.1 in [Ko] und [HL12] [UK09]
- und Dynamic Systems Development Method (DSDM® / DSDM® Atern), siehe Kap. 3.2.2 in [Ko] und [Ca11] [DS08]

Sowohl PRINCE2 als auch DSDM Atern werden oft als „Wasserfallmethoden“ gesehen. Das entspricht - im Gegensatz zur ihrer häufigen praktischen Anwendung - jedoch nicht ihrer Intention:

Speziell nämlich an PRINCE2 ist

- seine konsequente und im Projektverlauf immer wieder überprüfte Ausrichtung auf die geschäftliche Rechtfertigung, dargestellt als Business Case, der fortlaufend aktualisiert wird;
- die Strukturierung seines Ablaufs nach überprüfbaren aufeinander aufbauenden Teilergebnissen (Produkten);
- die Gliederung in die Managementphasen: Initiierungsphase und danach eine oder mehrere Durchführungsphasen (die jeweils die Produktinkremente liefern);
- dass am Ende jeder Phase und nicht nur am Projektende die gemachten Erfahrungen gesammelt werden (das Projekt als lernende Organisation) und der Business Case überprüft und aktualisiert wird;
- dass es zu Beginn über das gesamte Projekt hinweg nur eine grobe Konzeption und Planung gibt, die Details aber jeweils erst zu Beginn einer Durchführungsphase nur für diese Phase erarbeitet werden;
- dass es stets an die jeweilige Größe, Art und Umgebung des Projekts anzupassen ist und nicht sein umfangreicher Maximalumfang 1: 1 genutzt werden soll (Tailoring). Im Minimum umfasst es nur diese vier Dokumentensätze: Projektleitdokumentation - Projektstatusberichte (auch mündlich möglich) - Projekttagbuch - Projektabschlussbericht;
- dass seine Rollenträger möglichst autonom innerhalb eines definierten Rahmens agieren.

Und DSDM Atern ist dadurch gekennzeichnet:

- Das inhaltliche Projektergebnis wird anpassbar gehalten, Kosten, Termine und Qualitätsanforderungen hingegen sind fixiert.

- Einzelne Phasen oder Phasengruppen können iterativ durchlaufen werden.
- Die Lösung kann schrittweise in einzelnen Inkrementen ausgeliefert werden.
- MoSCoW-Priorisierung (Must Have/Should Have/Could Have/Won't Have this Time) der Teilresultate: Pro Timebox werden etwa 60% Must Have, 20% Should Have, 20% Could Have geplant.
- Förderung von Modellen und Prototypen.
- Kein BDUF, aber ein bewusstes JEDUF - mit allen Stakeholdern vereinbart - als Mittel gegen schleichende Funktionserweiterungen.
- Frühzeitige Risikoanalyse (in den Phasen Machbarkeitsprüfung und Rahmenplanung/Grundlagen).

PRINCE2 ist der De-facto-Standard für Projekte in Großbritannien und die Standardmethode für die Projektmanager-Trainings der Vereinten Nationen. PRINCE2 kommt zentralen Elementen des inkrementell-adaptiven Vorgehens im Projektmanagement entgegen, erzwingt sie aber nicht. Es befasst sich jedoch nur mit den Arbeitsprozessen und Rollen bis zum Teammanager, nicht mit jenen auf der Teamebene selbst. Dort können spezifische Methodenrahmen wie Scrum, XP oder DSDM als Ergänzung angeschlossen werden. [Pr06]

DSDM ist einerseits ein für sich allein stehendes Framework zur projektmäßigen Entwicklung dynamischer, also nicht genau vorausplanbarer Systeme einschließlich eines Rahmens für das Projektmanagement. Andererseits fokussiert DSDM (wie Scrum) insbesondere auch auf die teambasierte Entwicklungsarbeit. Dieser Teil des DSDM kann, wie vorhin bereits erwähnt, auch im Rahmen anderer übergeordneter Projektmanagementmethoden wie PRINCE2 eingesetzt werden [Pr06]. DSDM ist mit seinen Phasen, Prozessen und Outputs aus traditioneller Sicht gut verständlich, andererseits ist es durch essenzielle agile Elemente gekennzeichnet: Es ermöglicht, bei entsprechender Verschmelzung seiner eher sequenziellen Phasen, ein iterativ-adaptives und inkrementelles Entwicklungsvorgehen. Es fordert den fortlaufenden Einbezug der Nutzer und Kunden in den Entwicklungsprozess. Es unterstützt die Ausrichtung auf klar definierte strategische Ziele und auf die frühe Lieferung echter Geschäftsvorteile. DSDM ist das einzige als ISO9001-verträglich akkreditierte agile Verfahren.

3.3 Konzepte und Praktiken der agilen Neu- und Weiterentwicklung und Wartung von Produkten

Wie vorhin bereits erläutert, passt die Flexibilität bezüglich des Ziels (oder Funktionsumfangs) und allenfalls auch der investierten Anzahl von Produktinkrementen besser zu einer den gesamten Produktlebenszyklus betrachtenden Vorgehensweise als zu einem bezüglich Zeit, Kosten und Inhalt begrenzten Projekt. So etwa ist im Scrum Guide [SS] zu lesen: *„Solange ein Produkt existiert, existiert auch ein Product Backlog.“* Das Wort Projekt kommt im Scrum Guide nur an einer einzigen Stelle vor: *„Jeder Sprint kann als Projekt verstanden werden, für das ein zeitlicher Rahmen von maximal einem Monat zur Verfügung steht.“*

Deshalb ist Scrum keine Projektmanagement-Methode (auch wenn sie oft so bezeichnet wird), sondern beschreibt generische Vorgehensregeln einer inkrementell-adaptiven Produktentwicklung, beginnend mit der Realisierung des ersten Produktinkrements über alle weiteren Inkremente der Weiterentwicklung des Produkts hinweg bis zu seinem Lebensende.

Auf den gesamten Lebenszyklus von Produkten, nicht nur auf die Phasen seiner projektmäßigen Bearbeitung (Erstentwicklung und umfassende Überarbeitungen) beziehen sich auch das Scaled Agile Framework (SAFe) und Kanban. Wobei Kanban nicht nur das Entwicklungsvorgehen beschreibt, sondern für alle Arten sequenziell aufeinander folgender Arbeitsschritte nutzbar ist.

Typische Beispiele für die agile Neu- und Weiterentwicklung und Wartung von Produkten sind

- auf Teamebene:
 - Scrum, siehe Kap. 3.3.1 in [Ko] und [Sc] [SAa] [SS] [SAb]
- für alle Arten kontinuierlicher Arbeitsflüsse auf Basis sequentieller Bearbeitungsschritte:
 - KANBAN (in Großbuchstaben), eine 2007 veröffentlichte Methode zur evolutionären Verbesserung des Prozesses der Softwareentwicklung und Kanban (in Kleinbuchstaben), ein für die Softwareentwicklung stark abgewandeltes dezentrales Selbststeuerungssystem auf der Grundlage der bei der Toyota Motor Corporation entwickelten Methode der Produktionsablaufsteuerung auf der Basis des Pull-Prinzips, siehe Kap. 3.3.2 in [Ko] und [An11]
- für kontinuierliche Lieferungen von Produktinkrementen in Multiteam / Multiprodukt-Umgebungen:
 - Scaled Agile Framework for Enterprise™ (SAFe) [Le13] [Le] [Le10]

Scrum beschreibt kein Projektvorgehen von der Projektinitialisierung bis zum Projektabschluss, sondern ist ein Framework zum Vorgehen bei der Neu- und Weiterentwicklung komplexer, im Voraus also nicht genau definierbarer und planbarer Produkte auf der Basis empirischer Prozesskontrolle. Das illustriert z. B. auch Ken Schwaber in einem Fallbeispiel, bei dem eine existierende Großrechneranwendung auf eine Webanwendung umgestellt werden sollte. Die Fallbeschreibung beginnt dort, wo das Projekt bereits bewilligt, seine Finanzierung gesichert und der Projektplan und das Entwicklerteam schon vorhanden waren [Sc07]. Scrum beschreibt, unabhängig von der Art des Produkts, wie das Scrum-Team innerhalb einer immer gleich langen Timebox (genannt Sprint) von zwei bis maximal vier Wochen ein fertiges, nutzbares und potenziell auslieferbares Produktinkrement herstellen kann. Das Scrum-Team besteht aus dem Product Owner, den Entwicklern (als Entwicklerteam) und dem Scrum Master (der dem Team dienenden Führungskraft). Zu Beginn jedes Sprints wird im Scrum-Team vereinbart, was für das nächste Produktinkrement zu leisten ist. Basis dafür ist eine fortlaufend aktualisierte und nach Erledigungsreihenfolge geordneten Liste (genannt Product Backlog) von all dem,

was im Produkt künftig benötigt werden könnte. Am Ende des Sprints werden - unter Einbezug relevanter teamexterner Stakeholder - das effektiv Geleistete (Sprint-Review) und die Arbeitsweise während des Sprints (Sprint-Retrospektive) überprüft. Diese Überprüfung dient der fortlaufenden Verbesserung sowohl des Produkts (Review) als auch des Prozesses der Produktentwicklung (Retrospektive). Das - und nicht mehr oder weniger - beschreiben die Regeln von Scrum [SS]. Sie beschreiben z. B. nicht, wie die Elemente des Product Backlogs vor dem ersten Sprint ermittelt, beschrieben, bezüglich Machbarkeit und Risiko beurteilt und mit den Backlogs anderer Produkte koordiniert werden und wie die fortlaufende Aktualisierung und Abstimmung mit anderen Teams und Produkten erfolgt. Scrum gibt auch keine Hinweise, wie eine Handvoll bis mehrere Dutzend Teams gleichzeitig an verschiedenen Produkten und Projekten arbeiten können. Scrum beschreibt auch kein spezielles Vorgehen, wie während eines Sprints ungeplante und sehr dringende Fehler bereits produktiver Produktinkremente behoben werden können. All das und die je nach Produktart ganz unterschiedlichen Methoden und Techniken der Analyse, des Designs, der Modellierung, der Konstruktion/Realisierung, der Tests und der Qualitätssicherungsmaßnahmen und der Inbetriebnahme und fortlaufenden Produktbetreuung müssen ergänzend erarbeitet und vereinbart werden. Scrum will alle diese spezifischen Aspekte auch nicht regeln, sondern ist *„weder ein Prozess noch eine Technik zur Erstellung von Produkten; es ist vielmehr als Framework zu verstehen, innerhalb dessen verschiedene Prozesse und Techniken zum Einsatz gebracht werden können.“* [SS]. Die auf den ersten Blick einfach erscheinenden Regeln von Scrum führen allerdings auch zu ungerechtfertigten Erwartungen und damit verbundenen Fehlnutzungen, siehe siehe Kap. 3.3.1 in [Ko] und [Ko13]:

- Scrum ist keine schlanke Entwicklungsmethode. Es ist ein allgemeines Vorgehens-Framework und ein verfügbarer leerer Container für etablierte spezifische Methoden der professionellen Produktentwicklung.
- Scrum führt nicht zu einer massiven Erhöhung der Effizienz: Es unterstützt die Adaptivität, Qualität und Effektivität. Insgesamt (bezogen auf ein gesamtes fertiges Produkt) wird es mit Scrum nicht eindeutig schneller oder billiger [Ha08].
- Scrum erfordert - wie auch andere agile Vorgehensweisen - die strikte Professionalität erfahrener Entwickler [BT03].
- Scrum legalisiert kein Arbeiten auf Zuruf, sondern setzt ein agiles und professionelles Anforderungsengineering und kontinuierlich adaptierte Design- und Architekturrahmen voraus.
- Scrum kann nicht für JEDE Art von Vorhaben eingesetzt werden, sondern ist dann geeignet, wenn EIN Produkt von einem oder wenigen ausschließlich dafür arbeitenden interdisziplinären Teams allein (also ohne Abhängigkeit von team-externen Spezialisten) so neu- oder weiterentwickelt wird, dass lieferbare Teile davon nach jeweils bereits zwei bis vier Wochen vorhanden sind. Und zwar so, dass die raschen Rückmeldungen ihrer Nutzer unmittelbar zur Steuerung der weiteren inhaltlichen Entwicklung dienen.

- Auch Selbstorganisation braucht Führung: Das Team als Ganzes ist selbstverantwortlich - jedoch innerhalb klar vorgegebener Spielregeln und Rahmenbedingungen, vgl. S. 77 - 81 in [Ko13]

KANBAN (in Großbuchstaben) dient der evolutionären Prozessverbesserung indem es die aktuell praktizierten Arbeitsweisen sicht- und messbar (quantifiziert und statistisch auswertbar) macht und damit objektive Voraussetzungen für schrittweise, auch experimentelle, Verbesserungen schafft. Es geht vom Existierenden und Etablierten aus ohne Anspruch, es sofort verändern zu wollen, und erlaubt evolutionäre Veränderungen in jener Art und Geschwindigkeit, wie sie der Organisation angemessen sind. Das reduziert den Widerstand gegen Veränderungen erheblich.

Kanban (in Kleinbuchstaben) dient der operativen dezentralen (Selbst-) Steuerung der Arbeit auf Basis des Pull-Prinzips. Es ist extrem flexibel und rasch an spezielle, auch nicht vorausgesehene Situationen anpassbar. Die Arbeitsschritte - die Spalten der Kanban-Tafel - und deren Work-In-Progress-Limits (WIP-Limits) können im Sinn der empirischen Prozesskontrolle rasch und auch experimentell verändert werden. In Organisationen mit einem ausgeprägten Top-down-Steuerungs- und Kontrollbedürfnis auch auf Detailebene kann es jedoch zum Gefühl des Kontrollverlusts führen und auch zu Unsicherheiten bezüglich der längerfristigen Planbarkeit und Zuverlässigkeit der Fertigstellungstermine der Arbeitspakete. Erst eine Reihe positiver Erfahrungen mit „trotz“ Pull-Prinzips termingerechten Just-in-time-Lieferungen vermag diese Skepsis ab- und Vertrauen aufzubauen.

Die Grundprinzipien agiler Vorgehensweisen beziehen sich in der Regel auf nur EIN Produkt/Projekt, dessen Inkremente von EINEM Team in einer relativ homogenen Systemlandschaft entwickelt und rasch integriert werden. Das „Scaled Agile Framework for Enterprise (SAFe)“ beschreibt, wie agile Vorgehensweisen für Multi-Team- bzw. Multi-Produkt-Situationen bezogen auf den gesamten Lebenszyklus der Produkte in großen Unternehmen mit einer hohen technologischen Vielfalt bis auf die Ebene des Portfolio-Managements skaliert werden können. Das SAFe offeriert umfassende Beschreibungen der individuellen Rollen, Teams, Tätigkeiten und Artefakte zum agilen Arbeiten auf den Ebenen

- Teams (dort wird als Vorgehen Scrum in Verbindung mit XP empfohlen),
- Programm/Agile Release Trains (diese liefern alle etwa zehn Wochen = fünf Scrum-Sprints in die gesamte IT- Systemlandschaft integrierte Potential Shippable Increments, die aus Beiträgen mehrerer Teams und IT-Plattformen bestehen),
- Portfolio (dort wird ein an Kanban angelehntes Vorgehen empfohlen).

Zentrale Aspekte des SAFe sind:

- Skalierung der zu schaffenden Werte: Nicht alles ist eine User-Story.
- Skalierung von Team und Timebox: Kein Team ist eine Insel.
- Skalierung des Design: Emergentes Design verbindet sich mit bewusst vorausgedachter Architektur.

- Skalierung des Portfoliomanagements: Das althergebrachte Denken in Jahresbudgets hinterfragen und ändern.
- Skalierung der Führerschaft: Das Unternehmen kann nur so „lean“ wie das Denken der Geschäftsführung sein.

Das SAFe versteht sich im Gegensatz zu einem Projektmanagement-Framework als agiles Framework zur kontinuierlichen Lieferung produktiv nutzbarer Lösungen in (internen) Releases im Rahmen der Neu- und Weiterentwicklung und der Wartung von Produkten längs ihrer gesamten Lebenszeit. Angesichts seiner im Internet [Le] frei verfügbaren sehr umfangreichen Beschreibung der Rollen und Praktiken wird das SAFe von einigen Agilisten als „schwergewichtig“ und „wasserfallorientiert“ gesehen, von anderen dennoch als unbestreitbar agil. [Ru12] Dabei zu berücksichtigen ist, dass die vom SAFe angebotenen Praktiken, obwohl aufeinander abgestimmt, auch nur teilweise genutzt werden können. SAFe verlangt nicht, alle seine Praktiken vollumfänglich zu verwenden. Und es bleibt auch zu bedenken, dass andere unbestritten als „schlank“ gesehene agile Rahmenwerke - wie etwa Scrum - als solche allein nicht nutzbar sind sondern zusätzlich eine Vielzahl spezifischer Methoden der professionellen Produktentwicklung erfordern.

3.4 Die insgesamt agile Organisation

Seit einigen Jahren, inspiriert von der Verbreitung des agilen Vorgehens in der SW-Entwicklung, werden die im Abschnitt *Die Vieldeutigkeit des Begriffes „agil“* genannten Aspekte auch als Prinzipien „agiler“ Organisationen und Unternehmen insgesamt empfohlen. Da jedoch Unternehmen nebst der - allenfalls inkrementell-adaptiv gestaltbaren - Entwicklung und Realisierung von Produkten auch viele Prozesse ohne Produktcharakter (wie etwa das Personalwesen) umfassen, werden unter agil dann nicht primär das inkrementell-adaptive Vorgehen sondern diverse post-tayloristische, hierarchie- und autoritätsfreie und selbststeuernd-kollaborative Prinzipien und Überzeugungen verstanden. Oder die in 2.4.4 genannten allgemeinen Aspekte zur Sicherung der Überlebens- und Entwicklungsfähigkeit.

All das sind jedoch in der Organisationsentwicklung schon lange bekannte und diskutierte Ideen. Einige davon wurden - allerdings in vergleichsweise wenigen Fällen - erfolgreich umgesetzt. Vielfach zitierte Beispiele sind u. a. SEMCO [Se95] [St12] und die Morning Star Company Inc. [Gi12]. Weitere Fälle, die man heute als agil bezeichnen könnte, finden sich in [Ra03].

Der Versuch, diese heute unter „agil“ subsumierten Prinzipien im Rahmen der mehrheitlich etablierten Führungsstrukturen und der heute üblichen Vorgehensweisen im Gesamtunternehmen einzuführen ist tatsächlich wie „neuer, noch gärender, Wein in alte Schläuche“, braucht also „neue Schläuche“, also ein im Unternehmenskontext grundsätzlich verändertes Denken und Handeln. Diese Forderung nach „neuen Schläuchen“, also nach einem neuen Denken und Handeln, wird jedoch bereits seit Jahrzehnten erhoben, ist also ein „Déjà-vu“. Allerdings immer noch ein selten erfülltes.

Dieses veränderte Denken und Handeln setzt voraus, dass beginnend bei der Unternehmensspitze

- das „alles im Griff haben“ ersetzt ist durch ein „vertrauensvolles Loslassen“. Also: „Kontrolle ist gut - Vertrauen ist besser“
- die Überzeugung vorherrscht, dass ein Unternehmen erst durch die fortlaufende Konversation aller das Unternehmen umfassenden Mitarbeitenden und Stakeholder entsteht, nicht durch von oben verkündete Visionen oder verordnete Strukturen.

Diese Voraussetzungen zu erfüllen ist jedoch sehr anspruchsvoll.

Oft wird vorgeschlagen, diese Veränderung „von unten nach oben“ voranzutreiben. Ein soziales System „bottom up“ zu gestalten ist eine bereits Mitte des 19. Jahrhunderts von Michail Alexandrowitsch Bakunin, dem Mitbegründer des kollektivistischen Anarchismus, formulierte Vorstellung: Er lehnt jede Form der von oben nach unten wirkenden institutionalisierten und zentralisierten Autorität ab und akzeptiert statt dessen nur die Autorität von Spezialisten, weil sie nicht aufgezwungen sondern aus eigener Einsicht anerkannt wird. Deshalb fordert er auch den Aufbau der Gesellschaft von unten nach oben auf Basis freier Zusammenschlüsse und Interessensgemeinschaften der Menschen um eine möglichst große Autonomie sicherzustellen. [Ba05]

Aktuelle Erkenntnisse der Organisationsentwicklung widerlegen jedoch die Möglichkeit, ein Unternehmen (als soziales System) ausschließlich „bottom up“ verändern zu können. Essenziell nämlich ist die Unterstützung durch die Entscheidungsträger – bis hinauf zum Top Management auf Geschäftsleitungsebene als „institutionell vorgesezte“ und nicht „von unten legitimierte“ Autoritäten. Simone Inversini schreibt dazu: *„Es erweist sich als von zentraler Bedeutung, die Entscheidungsträgerinnen und -träger der Organisation verbindlich in die Veränderung einzubinden. Diese Ergebnisse widersprechen klar der von frühen Vertreterinnen und Vertretern der OE (Organisationsentwicklung) aus der gruppensdynamischen Tradition geübten Kritik an der Hierarchie an sich, mit welcher rein partizipative Ansätze legitimiert wurden. Die Ergebnisse veranschaulichen, dass eine Veränderung mit mangelhaft agierenden bzw. ohne Machtpromotorinnen und -promotoren wenig Chancen hat.“* [In05]

4 Schlussfolgerungen

In dieser Arbeit wurde gezeigt, dass der Begriff „agil“ ein sehr breites Spektrum an Bedeutungen hat - insbesondere dann, wenn die Domäne der SW-Entwicklung verlassen wird. Die Aussage „wir arbeiten agil“ ist daher eher missverständlich als klärend. Dargestellt wurden auch die heute verbreiteten agilen Konzepte und Praktiken und deren historische Wurzeln und typische Vertreter dieser Praktiken zum Entwickeln von Software, für das Management von Projekten und zur Neu- und Weiterentwicklung und Wartung von Produkten. Dargestellt wurde auch, dass die Gleichsetzung agil = Scrum viel zu kurz greift und damit die Chance vergeben wird, von den anderen derzeit verbreiteten weiteren agilen Verfahren zu lernen und das Risiko besteht eine der gerade populären statt die

im jeweiligen Kontext am besten passende Vorgehensweise zu nutzen. Den Abschluss bildete die kritische Würdigung der Idee einer insgesamt agilen Organisation mit Verweis auf die bereits lange Ideengeschichte in diesem Bereich und mit der Darlegung der Schwierigkeiten der Umsetzung dieser Ideen, die beginnend bei der Unternehmensspitze eine gegenüber dem „Üblichen“ grundlegend veränderten Art des Denkens und Handelns erfordert.

All jene, die vor der Frage stehen, ob und wie das „agile“ Vorgehen ihre aktuelle Berufspraxis verbessern könnte haben in dieser Arbeit Hinweise erhalten, dass diese Frage viel differenzierter zu stellen ist: Was von den unterschiedlichen Bedeutung von „agil“ ist in erster Linie für die Verbesserung der heutigen Situation relevant? Geht es primär um ein inkrementell-adaptives Vorgehen oder primär darum, mehrere Lösungsmöglichkeiten einzusetzen und nahtlos von einer zur anderen überzugehen, auf eine Veränderung der Umgebung rechtzeitig zu reagieren und Arbeitsprozesse und die Organisation zu ändern? Geht es um grundlegende Verbesserungen im Bereich der SW-Entwicklung oder des Projektmanagements oder der kontinuierlichen Weiterentwicklung von Produkten längs ihres gesamten Lebenszyklus? Oder geht es darum, das Unternehmen als Ganzes neu zu gestalten? Die in dieser Arbeit angebotenen zahlreichen Referenzen erlauben dabei das gezielte weitere Eintauchen in spezifische Themenfelder.

Diese Aspekte sollten in naher Zukunft vertieft werden:

- Wie können die heute in der Regel getrennt weiterentwickelten und angewendeten Praktiken kooperieren? Etwa so, wie SAFe auf Teamebene Scrum und XP nutzt oder so, dass PRINCE2 einen PM-Rahmen für mehrere Scrum-Teams bildet.
- Wie kann ein inkrementell-adaptives Vorgehen bei der Produktentwicklung in sehr großen Unternehmen so funktionieren, dass viele Produkte / Produktfamilien und mehrere tausend Entwickler weltweit in hunderten von Teams koordiniert zusammenarbeiten? Wie kann hier eine weitestgehende Selbststeuerung der Teams funktionieren?
- Wie kann es gelingen, dass beginnend bei der Unternehmensspitze das „vertrauensvolle Loslassen“ zum Normalfall wird und „Kontrolle ist gut - Vertrauen ist besser“ zum Leitspruch?

Literaturverzeichnis

- [AH09] Alberts, D. S.; Hayes, R. E.; Honekamp, Wilfried (Übersetzer): Power to the Edge. Re Di Roma Verlag, 2009.
- [An11] Anderson, D. J.: Kanban: Evolutionäres Change Management für IT-Organisationen. Dpunkt Verlag, 2011.
- [Ba05] Bakunin, M.: Die revolutionäre Frage. Föderalismus, Sozialismus, Antitheologismus. Münster, 2005.
- [Be01] Beck et al.: Manifesto for Agile Software Development. 2001. Siehe: <http://agilemanifesto.org/>

- [Be99] Beck, K.: Extreme Programming Explained: Embrace Change. Addison-Wesley, 1999.
- [BT03] Boehm, B.; Turner, R.: Using Risk to Balance Agile and Plan-Driven Methods. In: IEE Computer. IIEEE, June 2003; S. 57–66
- [Ca11] Caine, M.: DSDM Atern im Überblick. 2011. Siehe: <http://www.netzwoche.ch/de-CH/News/2011/10/19/DSDM-Atern-im-Ueberblick.aspx>
- [Co02] Coldewey, Jens: Multi-Kulti: ein Überblick über die agile Entwicklung. In: OBJEKTspektrum Ausgabe 01/2002.
- [Co12] Coldewey, J.: Was heißt hier eigentlich „agil“? Kennzeichen agiler Organisationen. In: OBJEKTspektrum Ausgabe 05/2012.
- [Do12] Dobelli, R.: Die Kunst des klugen Handelns. Carl Hanser Verlag, 2012; S. 185
- [DS08] DSDM Consortium: DSDM Atern Handbook. Whitehorse Press Limited, 2008. Siehe: <http://www.dsdm.org/content/dsdm-atern-handbook>
- [Fu08] Fuchs, C.: Internet and Society: Social Theory in the Information Age. Routledge, Reprint edition, 2008.
- [Gi12] Gillies, C.: Cheflose Unternehmen: Ein Hauch von Anarchie. Handelszeitung 01.06.2012, siehe: <http://www.handelszeitung.ch/management/cheflose-unternehmen-ein-hauch-von-anarchie>
- [Ha08] Harwardt, Mark: Wasserfallmodell versus Scrum – Ist der gute Ruf der agilen Methode gerechtfertigt? Masterarbeit. Lehrgebiet Programmiersysteme, Fernuniv. Hagen, 2008.
- [Ha11] Hagel III, J. et al.: The 2011 Shift Index. Deloitte Center for the Edge, http://www.deloitte.com/assets/Dcom-UnitedStates/Local20Assets/Documents/TMT_us_tmt/us_tmt_shiftindex_revised_120512.pdf
- [HL12] Hoffmann, G.; Lee, J. M.: Kontrolle ohne Bürokratie: Projektflexibilität durch PRINCE2. In (Hg.: Kammerer, S.; Amberg, M.; Lang, M. Hrsg.): Führung im IT-Projekt.. Symposion Publishing. 2012.
- [In05] Inversini, S.: Wirkungsvolles Change Management in Abhängigkeit von situativen Anforderungen. Humanwissenschaftlichen Fakultät der Universität Potsdam, 2005. Siehe: <http://opus.kobv.de/ubp/volltexte/2005/549/pdf/inversini.pdf>; Kurzfassung: http://www.iafob.ch/deu/10j/downloads/Teil_3_6.pdf
- [JS12] Janes, Andrea; Succi, Giancarlo: The Dark Side of Agile Software Development. In: Proceedings of the ACM international symposium on New ideas, new paradigms, and reflections on programming and software. ACM, 2012. Siehe: <http://darkagilemanifesto.org/dark-side-of-agile-janes-succi-splash-2012.pdf>
- [KM12] Kropp, M.; Meier, A.: Swiss Agile Study 2012. Fachhochschulen Zürich und Nordwestschweiz, 2012. Siehe: www.swissagilestudy.ch
- [Ko] Korn, H.-P.: Agile Konzepte. In (Bartsch, O.; Lindinger, M. Hrsg.): IT-Servicemanagement. TÜVRheinland. Siehe: <http://www.tuev-media.de/produkte/91154-it-servicemanagement.php>
- [Ko12] Komus, A.: Ergebnisbericht (Langfassung) Studie: Status Quo Agile. Verbreitung und Nutzen agiler Methoden. Studie des BPM-Labors der Hochschule Koblenz, 2012. Siehe: www.status-quo-agile.de
- [Ko13] Korn, H.-P.: Redesigned Agile in großen Unternehmen. In (Korn, H.-P.; Berchez, J. P. Hrsg.): Agiles IT-Management in großen Unternehmen. Symposion Publishing, 2013.
- [Kr09] Krüger, W.: Organisation als Kunstwerk. Abschiedsvorlesung Justus-Liebig-Universität Gießen, Lehrstuhl für Unternehmensführung und Organisation, 13.02.2009.
- [KS03] Kurtz, C.; Snowden, D.: The New Dynamics of Strategy: Sense-making in a Complex-Complicated World. In: IBM Systems Journal, vol. 42, no. 3. 2003; S. 462-483
- [Le] Leffingwell, D.: Scaled Agile Framework. Siehe: <http://scaledagileframework.com/>
- [Le09] Leffingwell, D.: Thoughts on Lean Thinking. Blog vom 15. Sept. 2009, scalingsoftwareagilityblog.com/thoughts-on-lean-thinking/
- [Le10] Leffingwell, D.: Agile Software Requirements: Lean Requirements Practices for Teams, Programs and the Enterprise. Addison-Wesley, 2010.

- [Le13] Leffingwell, D.: Agil bis zur Unternehmensstrategie - ein „Big Picture“. In (Korn, H.-P.; Berchez, J. P. Hrsg.): Agiles IT-Management in großen Unternehmen. Symposion Publishing, 2013.
- [Lex10] Lexikon IT-Management. Symposion Publishing, 2010.
- [Lu00] Luhmann, N.: Vertrauen: Ein Mechanismus der Reduktion sozialer Komplexität. UTB, Stuttgart, 2000.
- [Lu84] Luhmann, N.: Soziale Systeme. Suhrkamp, 1984.
- [Ma90] Martin, J.: Information Engineering, Book I/II/III. Prentice Hall, 1990.
- [NM09] Nissen, V.; Mladin, A.: Messung und Management von IT-Agilität. In (Fröschle, H-P Hrsg.): Wettbewerbsfaktor IT. HMD 269, 46. Jahrgang, 2009.
- [OEG01] Olson, E. E.; Eoyang, G. H.: Facilitating Organization Change: Lessons From Complexity Science. Pfeiffer, 2001.
- [Om12] Omann, D.: Quo vadis, Agile? borisgloger newsletter, Oktober 2012. Siehe: http://borisgloger.com/newsletter/oktober-2012/quo-vadis-agile/?utm_source=Newsletter&utm_campaign=89cc7711d1-NL201210
- [Pa12] Parsons, T.: The social system. University of California Libraries, 1951 und Quid Pro, LLC, 2012.
- [Po02] Poppendieck, M.: Lean Design. March 18, 2002, <http://www.leanessays.com/2010/11/lean-design.html>
- [Pr06] Hartmann Preuss, D.: Case Study: DSDM Bridges the Gap Between PRINCE 2 and XP. InfoQ News Oct 06, 2006. Siehe: <http://www.infoq.com/news/XP-DSDM-Prince2>
- [Ra03] Radatz, S.: Evolutionäres Management: Antworten auf die Management- und Führungsherausforderungen im 21. Jahrhundert. Verlag Systemisches Management, 2003.
- [Ru12] Rüssel, F.: Is SAFE unsafe – My Thoughts. Blog vom 7. Aug. 2012 in <http://www.agilerescue.de/is-safe-unsafe-my-thoughts/>
- [SAa] Scrum Alliance: Why Scrum? Siehe: <http://www.scrumalliance.org/why-scrum>
- [SAb] Scrum Alliance: Core Scrum. Agile Atlas. Siehe: <http://agileatlas.org/atlas/scrum>
- [Sc] Schwaber, K.: What is Scrum? Siehe: <http://www.scrum.org/Resources/What-is-Scrum>
- [Sc07] Schwaber, K.: Agiles Projektmanagement mit Scrum. Microsoft Press Deutschland, 2007. S. 72
- [Se95] Semler, R.: Maverick: The Success Story Behind the World's Most Unusual Workplace. Grand Central Publishing, Reprint edition, 1995.
- [SS] Schwaber, K.; Sutherland, J.: The Scrum Guide. Siehe: <http://www.scrumguides.org/>
- [St12] Stacho, P.: Das beste Managementbuch aller Zeiten: Zusammenfassung Semco-System von Ricardo Semler (Demokratisches Managementmodell). Blog vom 2. Juli 2012, Siehe: <http://geschaeftsmann20.com/2012/07/02/semcosystem/>
- [Sw12] SwissQ: Agile Trends & Benchmarks 2012. Siehe: www.swissq.it/swissq-agile-trends-benchmarks-2012-3, weitere Studien in www.swissq.it/ressourcen
- [SW97] Schuh, G.; Wiendahl, H.-P. (Hrsg.): Komplexität und Agilität: Steckt die Produktion in der Sackgasse? Springer, 1997.
- [Ta09] Tavra, D.: Vertrauen als Mechanismus der Reduktion von Komplexität; Universität Bern Sozialanthropologisches Institut, 2009. Siehe: <http://tinyurl.com/a62hqaw>
- [Tu02] Türcke, C.: Erregte Gesellschaft. Philosophie der Sensation. C.H. Beck, 2002; S. 130, Anm. 11
- [UK09] UK Office of Government Commerce: Erfolgreiche Projekte managen mit PRINCE2. The Stationery Office Ltd, 2009.
- [Wd09a] Wells, D.: Einführung in das Extreme Programming. 2009. Siehe: www.extremeprogramming.org/donwells.html
- [Wd09b] Wells, D.: Extreme Programming. Siehe: www.extremeprogramming.org
- [We12] Wernham, B.: Agile Project Management For Government. Maitland and Strong, 2012.
- [Wo03] Wolf, H.: Partizipatives Management - was bleibt? Expertise für die Hans-Böckler-Stiftung. Soziologisches Forschungsinstitut Universität Göttingen, Juli 2003.