
Die „agile“ Organisation: Vom Hype zum Daily Business

<i>Warum überhaupt „agil“? Die Herausforderungen</i>	2
<i>„Agile“ Organisation als Lösung?</i>	2
Mit Unberechenbarkeit und Komplexität umgehen statt sie beherrschen zu wollen.....	2
Organisation als stabile (hierarchische) Ordnung ... oder als polymorphes, lebendiges, Kunstwerk?	3
„Selbstorganisation“ als Lösung - oder als Missverständnis?.....	4
Nicht Struktur sondern Kommunikation und Vertrauen bildet soziale Systeme.....	4
Die Vieldeutigkeit des Begriffes „agil“	6
<i>Heute verbreitete agile Konzepte und Praktiken</i>	8
Konzepte und Praktiken des agilen Entwickelns von Software	9
eXtreme Programming (XP).....	9
Konzepte und Praktiken des agilen Managements von Projekten	12
PRINCE 2™.....	13
Dynamic Systems Development Method (DSDM® / DSDM® Atern).....	16
Konzepte und Praktiken der agilen Neu- und Weiterentwicklung und Wartung von Produkten	19
Scrum.....	19
KANBAN / Kanban.....	26
Scaled Agile Framework for Enterprise™ (SAFe)	29
<i>Was kennzeichnet, was leistet eine als Ganzes „agile“ Organisation?</i>	31
Welchen Begriff von „agil“ legen wir zu Grunde?	31
Was ist dann aus der Sicht der Kunden und Mitarbeitenden deutlich anders?.....	32
<i>Wie kann eine Organisation als Ganzes in diesem Sinn „agil“ werden?.....</i>	32
Rahmensetzung als Führungspflicht	33
Change Event	34
Der neue Rahmen des Organisationskreises	34
Wie verstehen wir / was bedeutet für unseren Organisationskreis dieser Rahmen?.....	34
Die Selbstbeschreibung des Organisationskreises	35
Die Identität des Organisationskreises	35
Der Wegweiser für den ab morgen eingeschlagenen Weg	35
Handeln - Überprüfen - Anpassen	36
Dynamische Organisation	37
Organisation als dynamisches Netzwerk	37
Soziokratie und Holacracy™	38
<i>Das neue „Daily Business“</i>	41

Warum überhaupt „agil“? Die Herausforderungen

„Agil“ zu sein ist unbestritten attraktiv: Wer will denn nicht - gemäss dem Duden-Fremdwörterbuch - „behände, flink, gewandt, regsam, geschäftig“ sein? Welche Organisation möchte als das Gegenteil von „agil“ im Sinn seines lateinischen Ursprungs (agilis, abgeleitet von agere) gelten, nämlich als „nicht handeln“ statt als „leicht zu führen, beweglich“? Langsamkeit und Müsiggang (als sprichwörtlicher Anfang aller Laster) ist 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“.

Was also hat sich verändert, dass „Agilität“ in den letzten rund zehn Jahren noch viel attraktiver, ja sogar zu einem der aktuellen „Modebegriffe“, geworden ist?

Einige Hinweise sind im seit 2009 vom amerikanischen Deloitte's Center for The Edge jährlich herausgegebenen Shift Index, einer Analyse der Performance von 20.000 US-Unternehmen seit 1965 bis heute, nachzulesen:

[vgl: http://www.deloitte.com/assets/Dcom-UnitedStates/Local%20Assets/Documents/us_tmt_2011shiftindex_111011.pdf]

Der ROA (Return on Assets) 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 ereignen sich mit immer rasanterem Tempo im asiatischen und pazifischen Raum; zwischen 1995 und 2006 haben die Wechsel der CEO um 59% zugenommen, die durch mangelnde Performance begründeten sogar um 318%.

Und ein Drittel der 1970 in der Fortune 500 gelisteten Unternehmen 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.“

„Agile“ Organisation als Lösung?

Eine „agile“ Organisation wird heute als Antwort auf diese Herausforderungen gesehen. Was aber bedeutet das ganz konkret? Was ist bei einer „agilen“ Organisation anders als bei einer „herkömmlichen“? Hier ein paar zentrale Aspekte:

Mit Unberechenbarkeit und Komplexität umgehen statt sie beherrschen zu wollen

Die unbegrenzt erscheinende Menge der sich rasch verändernden und ständig wachsenden Information und die uns heute spontan mögliche weltumspannende Vernetzung nehmen wir als extrem komplex war. 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 den besten Weg über das Eis zur Hütte planen. Und dann diesen Weg, ausgerüstet mit Schwimmweste und einer langen Leiter, gemäss Plan zügig und nach Plan hinter sich bringen.

Diese zweite Vorgehensweise entspricht einem von der technologischen Analysierbarkeit, Machbarkeit und Planbarkeit bestimmten Denken und ist Basis aller „schwergewichtigen“ und „plangetriebenen“ Vorgehensmethoden der Produktentwicklung und des Projektmanagements auf 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 [Edwin E. Olson and Glenda H. Eoyang: „Facilitating Organization Change: Lessons From Complexity Science. Pfeiffer; 2001]. Geprägt ist diese Vorgehensweise von der Einsicht, dass uns Situationen als „komplex“ dann erscheinen, wenn wir Beziehung zwischen Ursache und Wirkung nur im Nachhinein erkennen können, nicht aber im Voraus. Unser Vorgehen beruht dann auf „Probe - Sense - Respond“ 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 - Wirkung - Beziehungen. [vgl. C. Kurtz, D. Snowden: The New Dynamics of Strategy: Sense-making in a Complex-Complicated World. In: IBM Systems Journal. vol. 42, no. 3 2003, S. 462–83.]

Bei dieser ersten - empirischen - Vorgehensweise verzichten wir bewusst auf eine unangemessen aufwändiges und noch dazu unzuverlässiges „BDUF“. Statt dessen planen wir nur jeweils das für die nächsten Schritte wirklich Erforderliche auf Basis der bisherigen Einsichten. Zu Beginn unseres Vorhabens steht also nur ein „Just Enough Design Up Front“ (JEDUF).

Dieser Wechsel vom BDUF mit seiner „big bang“- Einführung der Lösung zum JEDUF mit seinen häufigen Einführungen von Lösungsincrementen und den die weitere Entwicklung steuernden raschen feedbacks lässt sich insbesondere in der Softwareentwicklung (als wesentlicher Treiber des agilen Vorgehens) in den letzten 30 Jahre exemplarisch verfolgen [vgl: Brian Wernham: Agile Project Management for Government. Maitland and Strong, 2012, Kap. 19]:

James Martin prägte ab 1983 mit seinem Buch „An Informations System Manifesto“ die von sequentiellen und jeweils etliche Monate dauernden Phasen (Analyse, Konzept, Realisierung, Einführung) bestimmten „wasserfallartigen“ und stark arbeitsteiligen Vorgehensweisen. Seit etwa 10 Jahren erleben wir jedoch einen immer deutlicheren Wechseln hin zu flexiblen, schlanken, auf in jeweils wenigen Wochen verfügbare konkret nutzbare und leicht änderbare Teilergebnisse gerichteten stark teamorientierten Entwicklungsmethoden.

Der Wechsel weg vom im Detail plangetriebenen hin zum nur grob im Voraus abgesteckten und emergenten Vorgehen ist natürlich auch eine psychologische Herausforderung: In der Regel ist es vielen von uns wohler, wenn wir einem detaillierten Plan folgen können: [vgl. Dobelli, Rolf: Die Kunst des klugen Handelns. Carl Hanser Verlag, 2012 S. 185]

Lesen Sie doch jetzt nochmals die zwei Vorgehensweisen zum Überqueren des zugefrorenen Flusses. Hand auf's Herz: Bei welcher wäre Ihnen wohler? Welche würden Sie insbesondere im professionellen Kontext, etwa als Anbieter von Winterexpeditionen, bevorzugen?

Organisation als stabile (hierarchische) Ordnung ... oder als polymorphes, lebendiges, Kunstwerk?

Unter „Organisation“ verstehen wir [vgl. <http://wirtschaftslexikon.gabler.de/Definition/organisation.html>] in der Regel das formale Regelwerk eines *arbeitsteiligen*, auf *Spezialisierung* ausgerichteten Systems. Zur Koordination dient eine effiziente Struktur, die, als natürliche Form, als Hierarchie gestaltet ist. Diese Struktur ist ein Abbild der Kompetenzen zur Regelung der *arbeitsteiligen* Erfüllung der *permanenten* Aufgaben. Die Koordinationsleistung der Hierarchie soll bewusst, überlegt und *dauerhaft* erfolgen.

„Die Hierarchie besitzt“ - gemäss wirtschaftslexikon.gabler.de - „Koordinationsfunktion, indem eine übergeordnete Stelle untergeordneten Stellen *Anweisungen* erteilt, die darauf gerichtet sind, den spezialisierten Aufgabenvollzug zielentsprechend *auszurichten*.“

Die hier genutzten Begriffe „Arbeitsteilung“, „Spezialisierung“, „permanent“, „dauerhaft“, „Anweisungen“ und „ausrichten“ zeugen eher von einer von Analysierbarkeit, Machbarkeit und Planbarkeit bestimmten tayloristischen als einer flexiblen, schlanken, auf in jeweils wenigen Wochen verfügbare konkret nutzbare und leicht änderbare Teilergebnisse gerichteten Teamarbeit.

Die Sicht der Organisation als stabile hierarchische Ordnung bezeichnete Prof. Dr. Wilfried Krüger in seiner Abschiedsvorlesung
[http://www.org-portal.org/fileadmin/media/legacy/Organisation_als_Kunstwerk_Abschiedsvorlesung.pdf]
als „Organisation 0.0“ und stellte ihr die „Organisation 2.0“ gegenüber, welche Evolutionsprozesse im Interesse des Wandels als permanente Aufgabe ermöglicht und innerhalb der Organisation kommunikative „Dorfplätze“ wiederbelebt.

„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“. 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 Unternehmen bis hin zu basisdemokratischen und anarchischen Vorstellungen heraus. Also: „Selbstorganisation ist Chiffre für einen Erklärungsnotstand, keine Erklärung.“ [Türcke, Christoph: *Erregte Gesellschaft. Philosophie der Sensation.* München.] 2002, S. 130, Anm. 11]
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.“ [Harald Wolf: *Partizipatives Management - was bleibt? Expertise für die Hans-Böckler-Stiftung.* Soziologisches Forschungsinstitut Universität Göttingen, Juli 2003]

Nicht Struktur sondern Kommunikation und Vertrauen bildet 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 die „Kommunikation“. Niklas Luhmann sagt dazu: „Allein Kommunikation - und nicht Personen! - bilden soziale Systeme. Kommunikation ist die kleinste Einheit eines Systems. Nicht Menschen (psychische Systeme) kommunizieren. Kommunikation kommuniziert und findet nur in sozialen Systemen statt, mögen diese noch so klein oder vorläufig sein.“ [Stefan Nehr Korn, *Systemtheorie: Niklas Luhmann.* 122. Veranstaltung der Humboldt-Gesellschaft am 19.08.2001
<http://www.humboldtgesellschaft.de/inhalt.php?name=luhmann>]

Wenn wir Organisationen als soziale Systeme sehen, dann geht es - nach Luhmann - nicht um deren personelle und funktionale Struktur sondern um die Kommunikation als „die Organisation“ schlechthin. Und diese Kommunikation ist gemäss Luhmann nicht das „Verbreiten von Information“ (z.B. das via Intranet als Videoclip vom CIO verkündete Leitbild) sondern die Einheit der vom Sender gewählten „Information“, deren „Mitteilung“ (als Handlung des Senders) und deren „Verstehen“ (Annahme und Interpretation) beim Empfänger. Erst wenn der Empfänger in einer „Anschlusskommunikation“ zum Sender wird und sein Verstehen als „Information“ „mitteilt“ und der Empfänger (der vorherige Sender) das „verstehet“ entsteht ein soziales System. Erst wenn der CIO sein Leitbild in einer Runde von 10 bis 20 Mitarbeitenden mitteilt und das von den Mitarbeitenden Verstandene als Rückmeldung dem CIO mitgeteilt wird und sich daraus eine

fortlaufende Kommunikation in Form einer Diskussion ergibt, entsteht ein soziales System. Und wenn alle diese Mitarbeiter in jeweils ihren „Kreisen“ das von ihnen Verstandene weiter diskutieren entstehen weitere soziale Systeme und bilden insgesamt kraft dieser Kommunikationen eine „Organisation“, welche sich je nach Verlauf dieser vielschichtig verwobenen Kommunikation dynamisch verändert.

Jedoch: Ein derartig dynamisches Netzwerk stellt sich als ein uns bald überforderndes komplexes Gebilde dar: Welche Informationen sind wichtig? Muss ich jetzt überall alles mitbekommen? Was sind aus der Luft gegriffenen Gerüchte und Hypothesen? Sind die jetzt gerade „aufglühenden“ Kommunikationspfade bloss kurze Strohfeuer oder der Beginn länger dauernder „hot spots“?

Hier komme ich wieder zu Luhmann: 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" [Luhmann, Niklas (2000): Vertrauen. Ein Mechanismus der Reduktion sozialer Komplexität, Stuttgart: Lucius & Lucius.; zitiert aus Dijana Tavra, Vertrauen als Mechanismus der Reduktion von Komplexität; Universität Bern Sozialanthropologisches Institut; <http://tinyurl.com/a62hqaw>]

Und "Vertrauen" wird aufgebaut - ich bleibe hier bei Luhmann - durch "Kommunikation".

Kann es sein, dass all die Diskurse um die - von uns als belastend wahrgenommene - "Komplexität" letzten Endes ein Indiz eines immer grösseren Verlustes von "Vertrauen" und damit auch von "Orientierung" ist?

Kann es sein, dass es nicht so sehr darum geht, "besser" mit dieser sich als Vertrauensverlust zeigenden Komplexität als unverankert in dieser Welt umherschwimmendes Individuum leben zu können sondern eher darum, wie wir wieder durch mehr direkte und persönliche Kommunikation mehr Vertrauen zu Personen, Rollenträgern, Teams, Normen, Organisationen, Strukturen und Prozessen erlangen?

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

Sehr viel: Werfen wir doch einen Blick auf zwei der vier „Werte“ des „Agile Manifest der Software-Entwicklung“: [<http://agilemanifesto.org/>]

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

- Individuen und Interaktionen mehr als Prozesse und Werkzeuge
- Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung

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

Und auf 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.

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

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:

[vgl. Georg Kneer (Hrsg) Markus Schroer (hHg): Soziologische Theorien. VS Verlag für Sozialwissenschaften; 2009]

A: Anpassung (Adaptation) an die Umwelt

G: Zielerreichung (Goal-achievement)

I: Integration als Mechanismus zur Leistungserbringung der Teilsysteme untereinander

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

Im 1997 erschienenen Buch „Komplexität und Agilität: Steckt die Produktion in der Sackgasse?“

beleuchten eine Reihe von Beiträgen „... die Widersprüche zwischen Komplexität im Produktionsumfeld und Agilität im Markt ... und geben 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.“

[Günther Schuh (Hsg), Hans-Peter Wiendahl (Hsg): Komplexität und Agilität: Steckt die Produktion in der Sackgasse? Springer 1997]

Im Februar 2001 trafen sich 17 Vertreter der damals „leichtgewichtig“ genannten Methoden der Softwareentwicklung und formulierten das sie Verbindende als die vier Werte und 12 Prinzipien des „Agilen Manifests der Softwareentwicklung“ [<http://agilemanifesto.org/>] 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 Begriffes „agil“ und seiner Ausdehnung auf andere Bereiche - und auch der Differenzierung seiner Bedeutung.

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

Agil = Flexibilität und Adaptivität sichern mit „Empirical Process Control“ in Form eines Vorgehens in kontinuierlichen kleinen Schritten ausgehend von einem möglichst leichtgewichtigen „Just Enough Design Up Front“: Jeder Schritt liefert ein - wenn immer möglich bereits praktisch nutzbares - Teilergebnis, das Grundlage für das Lernen und für feedbacks zur Gestaltung der nächsten Schritte ist.

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 schreibt: „All other agile tactics are optional details.“ [Brian Wernham, Agile Project Management for Government, Kindle eBook 1 Edition, Maitland and Strong, 2012]

Diese Sicht entspricht auch jener von Volker Nissen. Er unterscheidet dabei jedoch zwischen einer „kapazitiven“ und „funktionalen“ Agilität und ergänzt sie um die „proaktive Agilität“, um unvorhergesehene Änderungen und IT-Innovationen unterstützen bzw. aufspüren zu können. [Volker Nissen, Alexander Mladin: Messung und Management von IT-Agilität. In: Hans-Peter Fröschle (Hrsg.): Wettbewerbsfaktor IT. HMD 269, 46. Jahrgang, Oktober 2009]

Agil = Flexibilität und Adaptivität plus „Lean Management Principles“: Zusätzlich zur obigen Bedeutung werden die Prinzipien des „Lean Managements“ als eigentliche Basis von „agil“ gesehen. Diese werden oft dargestellt als „House of Lean“ ausgehend von Larman und Vodde (2009) und The Toyota Way (2004) [vgl. Dean Leffingwell: Thoughts on Lean Thinking. Blog vom 15. Sept. 2009 in <http://scalingsoftwareagilityblog.com/thoughts-on-lean-thinking/>]

Das Ziel „Wert schaffen“ als Dach des House Of Lean ist gekennzeichnet durch „Sustainable shortest lead time / Best quality and value (to people and society) / Most customer delight, lowest cost, high morale, safety.“ und getragen von den beiden Säulen „Respect for People“ und „Continuous

Improvement“. 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“.[Lexikon IT-Management. Symposion Publishing 2010]

Agil = Flexibilität und Adaptivität 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 Adaption Program“ der „Agile Alliance“ entwickelte Modell ist ein Beispiel für diesen Begriff von „agil“. [Jens Coldewey; Was heisst hier eigentlich "agil"? Kennzeichen agiler Organisationen; in OBJEKTspektrum Ausgabe 05/2012]

Auch das „Agile Manifest der Softwareentwicklung“ [<http://agilemanifesto.org/>] kann dieser Gruppe zugerechnet werden. Seine vier Werte lauten:

„Wir erschließen bessere Wege, Software zu entwickeln, indem wir es selbst tun und anderen dabei helfen. Durch diese Tätigkeit haben wir diese Werte zu schätzen gelernt:

• *Individuen und Interaktionen* mehr als Prozesse und Werkzeuge

- *Funktionierende Software* mehr als umfassende Dokumentation
- *Zusammenarbeit mit dem Kunden* mehr als Vertragsverhandlung
- *Reagieren auf Veränderung* mehr als das Befolgen eines Plans

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

Diese vier Werte und zwölf Prinzipien [<http://agilemanifesto.org/iso/de/principles.html>] sind, dem Charakter eines Manifests entsprechend, als Appell zu verstehen und nicht als „verifizierte Arbeitsregeln“. Sie dogmatisch als Handlungsanweisungen für die Software- oder Produktentwicklung in unterschiedlichsten Situationen zu betrachten wäre eine Fehlnutzung. [vgl: Andrea Janes, Giancarlo Succi: The Dark Side of Agile Software Development. Free University of Bolzano. Online in <http://darkagilemanifesto.org/dark-side-of-agile-janes-succi-splash-2012.pdf>]

Agil = Aspekte zur Sicherung der Überlebens- und Entwicklungsfähigkeit: 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 Departement of Defense:

- *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.

[vgl.: David S. Alberts, Richard E. Hayes, Wilfried Honekamp (Übersetzer): Power to the Edge. Re Di Roma-Verlag; 2009]

Diese Definition versteht unter „Flexibilität“ im Gegensatz zu allen anderen Sichtweisen von „agil“ nicht nur ein inkrementell-adaptives Vorgehen in kleinen Schritten ausgehend von einem „JEDUF“ sondern lässt auch ein stark plangetriebenes, „BDUF“-basiertes Vorgehen dann zu, wenn es für die aktuelle Situation passender und effizienter ist.

Heute verbreitete agile Konzepte und Praktiken

Heute wird „agil“ und „Scrum“ oft als Synonym benutzt. Blicken wir jedoch nur etwa 10 Jahre zurück. Damals, 2002, war 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“. [Jens Coldewey, „Multi-Kulti: ein Überblick über die agile Entwicklung“, in OBJEKTSpektrum 1-2002]

Heute - rund 10 Jahre 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 in 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, welche recht konkrete Verfahren und Techniken, die dann im Laufe des Projekts angepasst werden können, vorschlagen:

DSDM, eXtreme Programming (XP) und Feature Driven Development (FDD).

Dominant im Jahr 2002 war XP. Heute, zehn Jahre später, dominiert Scrum (allerdings nicht in „Reinkultur“ sondern primär in diversen Abwandlungen) die agile Landschaft. Das zeigen übereinstimmend drei unabhängige Studien aus dem Jahr 2012:

[SwissQ Agile Trends & Benchmarks 2012, online verfügbar in www.swissq.it/resources/detail/8]

[Swiss Agile Study 2012, online verfügbar in www.swissagilestudy.ch/ und erste Ergebnisse in http://www.lean-agile-scrum.ch/wp-content/files/00_Swiss%20Agile%20Study%20Resultate.pdf]

[Status Quo Agile 2012, online verfügbar in <http://www.hs-koblenz.de/Status-Quo-Agile.4782.0.html>]

Gemäss diesen Studien setzten 2012 bei den antwortenden Firmen - je nach Studie - 51 bis 78 Prozent agile Vorgehensweisen (oft parallel zu „klassischen“) ein. Davon nutzen 51 bis 84.5 Prozent Scrum (bzw. ein an Scrum orientiertes Vorgehen). Bei zwei dieser drei Studien stand das vor 10 Jahren noch unbekannte „Kanban“ mit 17 bzw. 43 Prozent an zweiter Stelle, bei der dritten Studie mit 4.5 Prozent an fünfter Stelle. Die Verbindung von Scrum mit Kanban („Scrumban“) hatte eine Verbreitung von immerhin bereits 3 bis 8.5 Prozent. Das vor zehn Jahren dominierende eXtreme Programming (XP) stand mit 5.5 bis 33 Prozent 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 betrachtet werden.

Weitere verbreitete 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 10 Jahren noch recht bekannte Vorgehensweise „Adaptive Software Development“ und die „Crystal-Methodenfamilie“ haben heute kaum noch Bedeutung. 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 RUP und PRINCE2 in den Bereich der agilen Vorgehensweisen vor, indem sie ihre iterativen Elemente verstärken. Auch bislang an ausgeprägten sequentiellen Phasen orientierte „klassische“ Vorgehensweisen wie HERMES und das V-Modell XT öffnen sich für eine Kombination mit agilen Vorgehensweisen zumindest im Sinn eines „hybriden“ Vorgehens.

In den nächsten 10 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 vor 10 Jahren noch niemand in diesem Gebiet an Kanban dachte. Und möglich ist auch, dass in 10 Jahren ein anderes „label“ als „typisch agil“ in aller Munde ist.

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, gefördert u.a. durch Tom Gilb (also noch vor dem Entstehen des stark plangetriebenen Vorgehens mit seinem BDUF in den 1980er-Jahren, welches insbesondere vom James Martin mit seinem „Information Engineering“ vorangetrieben wurde). 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.

Dass ein adaptives Vorgehen mit in jeweils wenigen Wochen verfügbaren Lösungsincrementen in der Software-Entwicklung entstand, hat einen einfachen Grund:

Die Entwicklung von Software ist im wesentlichen die Herstellung einer modellhaften Beschreibung automatisierbarer Funktionen als in einer speziellen Kunstsprache (Java, C#, COBOL,...) formulierter „Text“. Für die Ausführung dieser derart beschriebenen Funktionen ist das ICT-System als bereits vorhandenes physisches System vorbereitet. Es kann diese Funktionen leisten, indem der maschinenlesbare Text der Funktionsbeschreibung (= der Programmcode) das Verhalten des ICT-Systems steuert. Das Entwickeln von Software ist somit vergleichbar mit dem stückweisen Komponieren und Spielen und Überarbeiten eines neuen Musikstücks auf einem bereits vorhanden Instrument ohne das Instrument zu verändern. Physische Produkte hingegen wie etwa eine Autobahnbrücke können nicht oder nur sehr schwer in kleinen und stets bereits nutzbaren Lösungsincrementen realisiert werden. Nur die für die Realisierung nötigen Konstruktions- und Fertigungspläne können inkrementell-adaptiv erstellt werden - und das ist seit langem auch durchaus üblich.

eXtreme Programming (XP)

Charakteristik und Herkunft

[vgl.: <http://www.extremeprogramming.org/donwells.html> und http://en.wikipedia.org/wiki/Extreme_programming]

XP geht davon aus, dass Software nicht auf Basis bereits zu Beginn möglichst vollständig erhobener Anforderungen realisiert und danach eingeführt werden kann sondern davon, 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 - sogar benutzt werden können. Daraus ergeben sich weitere und noch spezifischere Anforderungen. Jedes Software-Inkrement wird dabei innerhalb einer „Iteration“ von einer bis maximal vier Wochen realisiert. Eine Iteration umfasst die vier Basisaktivitäten:

- Zuhören (als Entwickler darauf hören, was der Benutzer haben will)
- Entwerfen (als Entwickler eine übergeordnete Konzeption dazu entwickeln, wie die Software möglichst einfach und daher gut erweiterbar und änderbar strukturiert sein soll)
- Programmieren, eng verzahnt mit
- Testen (wobei idealerweise zunächst die Testfälle erarbeitet und erst dann - mit kontinuierlichem Testen - programmiert wird)

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 beschrieben („User Stories“) und sind nicht zu verstehen als umfangreiche und verbindliche Detailspezifikation sondern als „Diskussionsanstösse“ für den Benutzer und Entwickler dann, wenn die Anforderung unter Mitwirkung des Benutzers konkret realisiert werden soll.

Zu Beginn jeder Iteration klären die Entwickler mit dem Benutzer, welche der ihm wichtigsten Anforderungen („Features“) innerhalb der den Entwicklern verfügbaren Ressourcen realisiert werden sollen. Bei der anschliessenden Konzeption (z.B. anhand von mock ups oder Prototypen) und des Realisierens und Testens bleibt die enge Zusammenarbeit zwischen Benutzer und Entwickler erhalten.

Den Iterationen übergeordnet ist die Planung mehrere Iterationen umfassender Releases.

XP wurde erstmals ab 1996 von Kent Beck, Don Wells, Ward Cunningham und Ron Jeffries zur Erstellung von Software im Bereich der Lohnabrechnung (Comprehensive Compensation System) bei Chrysler entwickelt. 1999 veröffentlichte Don Wells XP als Hypertext [<http://www.extremeprogramming.org/>] und Kent Beck in Buchform [Beck, K. (1999). Extreme Programming Explained: Embrace Change. Addison-Wesley].

Damit begann auch die weltweit rasante Verbreitung von XP.

Es existiert zwar nicht „die eine verbindliche“ Definition von XP, alle seine Beschreibungen beruhen jedoch auf diesen fünf „Werten“:

- Kommunikation
- Einfachheit
- Rückmeldung
- Mut
- Respekt

Je nach Quelle werden dann spezifische „Prinzipien“ und je nach Autor unterschiedlich strukturierte Sammlungen konkreter „Praktiken“, „Techniken“ oder „Regeln“ für die Softwareentwicklung, die Arbeit im Team und für die Benutzer genannt.

Getragen wird XP heute von international vernetzten Softwareentwicklern die sich in vielen Foren und Konferenzen (z.B. dem „XP day Germany“) austauschen. Eine der zentralen Referenzen für all das, was XP aktuell umfasst, sind nach wie vor die Bücher von Kent Beck und das von Don Wells editierte www.extremeprogramming.org. Die Nutzung und Schulung von XP ist an keinerlei Lizenzen gebunden.

Vorgehen und Rollen

Don Wells bezeichnet XP in www.extremeprogramming.org als einen der populären „agilen Prozesse“, dargestellt wird dort in einem Set von Flowcharts jedoch vor allem das Zusammenspiel der „XP-Rules“ und kein umfassender Softwareentwicklungsprozess. Die Flowcharts zeigen:

- Projekt, bestehend aus mehrere Releases, enthaltend u.a.:
- Iteration, enthaltend u.a.:
- Development, enthaltend u.a.:
- Collective Code Ownership

Insgesamt 29 XP-Rules sind in www.extremeprogramming.org beschrieben für die Bereiche:

- Planning
- Managing
- Designing
- Coding
- Testing

[<http://www.extremeprogramming.org/rules.html>]

Ein Beispiel für „Rules“ (oder „Techniken“) im Bereich „Planning“ ist das Beschreiben von Anforderungen in Form von „User Stories“ („Benutzungsgeschichten“): Im Gegensatz zur „üblichen“ Anforderungserarbeitung werden sie nicht von spezialisierten „Business Analysten“ sondern vom Benutzer selber in seiner Sprache ohne jeglichen „EDV-Jargon“ in ungefähr drei kurzen Sätzen formuliert. Erst wenn der Entwickler an dieser „User Story“ zu arbeiten beginnt, beschafft er sich alle weiteren Details in einem persönlichen Gespräch mit dem Benutzer.

Ein weiteres Beispiel aus dem Bereich „Coding“ ist das „Pair Programming“: Zwei SW-Entwickler sitzen dabei nebeneinander an einem Arbeitsplatz mit nur einer Tastatur. Einer codiert, der andere beobachtet das, was als Code entsteht. Die Rollen „Codierer“ und „Beobachter“ wechseln dabei häufig. Die Qualität des Codes (Fehlerfreiheit, Wartbarkeit) wird damit wesentlich verbessert.

Ein anderes Beispiel dieser „Rules“ ist das physische (nicht nur virtuelle) Zusammenarbeiten aller Teammitglieder in einem Teamraum mit genug Pinwänden, Flipcharts und Whiteboards für die spontane Kommunikation und Ko-Kreation.

Eine andere - bei allen „agilen“ Vorgehensweisen übliche - Technik ist das tägliche „Stand Up Meeting“: Alle Teammitglieder treffen sich täglich zur selben Zeit und tauschen im Stehen (damit das Meeting möglichst kurz dauert) aus, was sie seit Gestern erledigten, woran sie derzeit arbeiten, welche Erfahrungen sie machten und wo sie Hilfe von anderen benötigen.

In verschiedenen Veröffentlichungen werden Rollen bei XP wie „Entwickler“ „Kunde“ „Benutzer“ „Manager“ „Teamleiter“ „Tester“ „Projektleiter“ angeführt, in www.extremeprogramming.org werden jedoch keinerlei Rollen festgelegt. Nur innerhalb der Beschreibung einzelner Rules ist von „customer“ und „programmer“ und „team“ die Rede.

Das „Spezielle“ in Hinblick auf den Umgang mit Unberechenbarkeiten und Komplexität

Ganz zentral bei XP ist die möglichst synchrone, persönliche und spontane Kommunikation zwischen Benutzer und Entwickler und im Team. Dahinter steht die Einsicht, dass „Verstehen“ in unvertrauten Situationen am einfachsten und zuverlässigsten im Dialog entsteht und nicht durch das Schreiben, Lesen und Kommentieren umfassender Dokumente.

XP verlangt möglichst einfache Lösungen. Eine Technik dafür ist die fortlaufende Überprüfung und Überarbeitung der bereits realisierten Lösungen („refactoring“): Was ist unnötig? Was ist doppelt? Was kann noch einfacher gelöst werden?

XP beruht auf der Realisierung überprüfbarer Lösungsteile nach längstens 4 Wochen und damit auf entsprechend raschen und häufigen Rückmeldungen der Nutzer. Diese Rückmeldungen sind Anlass zur Anpassung der Release- und Iterationsplanung.

XP fordert eine sehr häufige (= ein- bis mehrmals pro Tag) Integration getesteter Teillösungen ins Gesamtsystem und fortlaufende Tests des Gesamtsystems. Damit sind Fehler rasch erkennbar und lokalisierbar.

Konsequenzen beim Praktizieren innerhalb „üblicher“ Organisationen

Dass Anforderungen zu Beginn der Realisierung nur grob formuliert und erst im Verlauf der schrittweisen Realisierung weiter konkretisiert und durchaus auch verändert werden können steht im Gegensatz zum vertrauten und insbesondere projektorientierten Vorgehen. Das ist zwar typisch für alle agilen Vorgehensweisen, bei XP aber besonders stark ausgeprägt.

Dass Anforderungen nicht zunächst ausformuliert sondern im Verlauf der schrittweisen Lösungsrealisierung vom Benutzer bzw. Kunden zusammen mit den Entwicklern im Dialog erarbeitet werden erfordert die kontinuierliche und rasche Verfügbarkeit des Benutzers bzw. Kunden für die Entwickler. Diese Benutzer bzw. Kunden müssen das nebst - oder anstelle - ihrer üblichen Arbeit organisieren können und zusätzlich müssen sie in der Lage sein, alle Arten von Benutzer bzw. Kunden zu vertreten.

Wenn diese Vertreter der Benutzer bzw. Kunden Anforderungen im direkten Dialog mit den Entwicklern konkretisieren und auch modifizieren führt das zu Abstimmerfordernissen mit übergeordneten Koordinationsinstanzen wie etwa dem Produktmanagement oder Budgetverantwortlichen.

Je stärker die Organisationsstruktur auf einer arbeitsteiligen Spezialisierung beruht umso schwieriger wird es sein, in sich abgeschlossene und funktional überprüfbare Lösungsteile in einer bis vier Wochen zu realisieren. Es muss also quer zur Organisationsstruktur ein Team bestehend aus allen zur Konzeption, Realisierung und zum Testen nötigen Spezialisten gebildet werden. Dieses Team soll physisch in einem Teamraum und nicht nur virtuell und örtlich verteilt arbeiten. Auch all das ist typisch für sämtliche agilen Vorgehensweisen.

Skalierbarkeit / Übertragbarkeit auf ganze Organisationen

XP beschreibt Spielregeln und Techniken, wie 6 bis 10 Entwickler als Team in enger Zusammenarbeit mit dem Benutzer schrittweise funktionierende Software mit hoher Qualität realisieren können. XP beschreibt keine Spielregeln und Techniken zur koordinierten Arbeit mehrerer Teams an einem umfangreichen Softwaresystem. Für derartige Situationen müssen zusätzliche Spielregeln und Techniken des Projektmanagements (z.B. PRINCE2 oder DSDM) bzw. des Programm- und

Portfoliomanagements (z.B. SAFe) angewendet werden. Essentiell ist, dass diese zusätzlichen Spielregeln und Techniken vereinbar sind mit:

- den fünf Werten des XP
- der Realisierung überprüfbarer Softwareinkremente in kurzen Iterationen von max. 4 Wochen
- einem eher groben „JEDUF“
- der kontinuierlichen Konkretisierung und Adaption der Benutzeranforderungen im Verlauf der SW-Entwicklung primär im persönlichen Dialog
- der kontinuierlichen und raschen Verfügbarkeit der Benutzer bzw. Kunden (bzw. deren entscheidungsbefugte Repräsentanten) für die Entwickler

Die von XP vertretenen fünf Werte sind anwendbar auf die ganze Organisation - und sollten es idealerweise ja auch sein. Die meisten Spielregeln und Techniken jedoch beziehen sich auf die Softwareentwicklung. Sie können aber dazu dienen, analoge Spielregeln und Techniken zu finden für Teams in anderen Fachgebieten wie dem Marketing oder dem HR-Management.

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 schrittweise vorantastenden Vorgehen ergab sich mitten auf dem Weg eine neue Richtung hin auf ein bequemeres 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 wir noch auf dem Fluss einige Meter vor dem Ufer angesichts des Hotels die Überquerung stoppen. Wir müssten 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 schlussendlich besser ist als das ursprünglich geplante.

Die Flexibilität bezüglich des Ziels (oder Funktionsumfangs) der Produktentwicklung und allenfalls auch der Anzahl der erforderlichen Produktinkremente ist essentiell für das agile Vorgehen. Das aber passt schlecht zu Projekten im „üblichen“ Sinn. Prof. Dr. Ayelt Komus (Hochschule Koblenz) schreibt dazu:

„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, 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.“

[Prof. Dr. A. Komus: Ergebnisbericht (Langfassung) Studie: Status Quo Agile Verbreitung und Nutzen agiler Methoden. Studie des BPM-Labors der Hochschule Koblenz. Online verfügbar in www.status-quo-agile.de]

Und in einem Interview zu dieser Studie sagt Prof. Komus:

[http://borisgloger.com/newsletter/oktober-2012/quo-vadis-agile/?utm_source=Newsletter&utm_campaign=89cc7711d1-NL201210]

„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 Prof. Konus in der Langfassung der Studienergebnisse, 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.“

PRINCE 2™

Charakteristik und Herkunft:

[Vgl: <http://de.wikipedia.org/wiki/PRINCE2>]

PRINCE2 (ein Akronym für „**P**rojects **i**n **C**ontrolled **E**nvironments“) ist einer der weltweit am meisten verbreiteten und für alle Arten von Projekten einsetzbaren Projektmanagement-Standards. Es ist der de facto Standard für Projekte in Grossbritannien und die Standardmethode für die Projektmanager-Trainings der Vereinten Nationen. Derzeit verbreitet es sich stark auch in ganz Europa, vor allem in Deutschland, Schweiz, Österreich, Italien, Frankreich und Polen. Für seine Nutzung (nicht aber seine Schulung) durch Personen und Firmen ist keinerlei Lizenz oder Zertifizierung erforderlich.

Speziell an PRINCE2

- ist seine konsequente und im Projektverlauf immer wieder überprüfte Ausrichtung auf die geschäftliche Rechtfertigung, dargestellt als „Business Case“, der fortlaufend aktualisiert wird
- ist die Strukturierung seines Ablaufs nach überprüfbaren auf einander aufbauenden Teilergebnissen („Produkten“)
- die Gliederung in die „Managementphasen“: Initiierungsphase und eine oder mehrere Durchführungsphasen (welche jeweils die Produktinkremente liefern)
- ist, dass am Ende jeder Phase und nicht nur am Projektende die gemachten Erfahrungen gesammelt und der „Business Case“ überprüft und aktualisiert werden
- ist, dass es ab 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
- ist, dass es stets an die jeweilige Grösse, Art und Umgebung des Projekts anzupassen und nicht sein umfangreicher Maximalumfang 1:1 zu nutzen ist
- ist, dass seine Rollenträger möglichst autonom innerhalb eines definierten Rahmen agieren.

Damit kommt PRINCE2 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 müssen spezifische Methodenrahmen wie DSDM oder Scrum oder XP als Ergänzung angeschlossen werden.

Eigentümer von PRINCE2 ist das UK Office of Government Commerce, seit 2010 integriert in die Efficiency and Reform Group innerhalb des UK Cabinet Office. Zertifizierungen auf zwei Stufen bietet APMG an. [<http://www.apmg-international.com/en/qualifications/prince2/prince2.aspx#.UT42w1cwKxk>]

Der Ursprung von PRINCE2 ist die in den 70er-Jahren von der britischen Regierung initiierte PROMPT-Methode, einer Sammlung der Best Practices der IT-Entwicklung. Daraus wurde 1989 als UK-Regierungsstandard für das IT-Projektmanagement PRINCE. 1996 entstand daraus PRINCE2 als allgemeine PM-Methode. Nach einer grundlegenden Überarbeitung von 2006 bis 2009 liegt es derzeit als PRINCE2:2009 vor.

Vorgehen und Rollen:

[Überblick: <http://de.wikipedia.org/wiki/PRINCE2>]

[Kurzbeschreibung: Guido Hoffmann; James Mitchell Lee: Kontrolle ohne Bürokratie: Projektflexibilität durch PRINCE2. In: Führung im IT-Projekt. Hrsg.: Sebastian Kammerer, Michael Amberg, Michael Lang. Symposium Publishing 2012]

[Detail: UK Office of Government Commerce: Erfolgreiche Projekte managen mit PRINCE2. The Stationery Office Ltd 2009]

Basis des Vorgehens sind diese 7 Prinzipien:

1. Fortlaufende geschäftliche Rechtfertigung anhand eines - fortlaufend zu aktualisierenden - „Business Case“
2. Lernen aus Erfahrungen am Ende jeder Phase
3. Definierte Rollen und Verantwortlichkeiten
4. Steuern über Managementphasen
5. Steuern nach dem Ausnahmeprinzip: Erst beim Überschreiten des Rahmens des autonomen Agierens wird die übergeordnete Steuerungsebene aktiv
6. Produktorientierung: Relevant sind die realisierten Ergebnisse, nicht die ausgeübten Aktivitäten
7. Anpassen an die Projektumgebung („tailoring“)

Diese Prinzipien werden verwirklicht im Rahmen von 7 Prozessen, jeder davon ist in weitere „Aktivitäten“ aufgeteilt:

1. Vorbereiten eines Projekts
2. Lenken eines Projekts
3. Initiieren eines Projekts
4. Steuern einer Phase
5. Managen der Produktlieferung
6. Managen eines Phasenübergangs
7. Abschließen eines Projekts

Ausgeführt werden diese Prozesse in den „Managementphasen“ des Projekts:

- Die „Initiierungsphase“ eines Projekts besteht aus dem 1. und 3. Prozess
- die Produkte liefernden mehreren iterativen „Durchführungsphasen“ enthalten die Prozesse 4 bis 7 (letzterer nur bei der letzten Durchführungsphase).

Die in jeder der „Durchführungsphasen“ realisierten Produkte sind einerseits die von den Spezialisten in den Teams entwickelten Fachergebnisse (funktionsfähige oder gar bereits nutzbare Teilprodukte, Prototypen, Modelle, aber auch Architekturkonzepte, Auslegungsrechnungen, Konstruktionsunterlagen etc) einerseits und „Managementprodukte“ genannte Projektmanagementdokumente (Arbeitspaketbeschreibung, Erfahrungsprotokoll, Projekttagbuch, Risikomanagement-Strategie etc) andererseits. PRINCE2 definiert als Maximum 26 solcher „Managementprodukte“, die aber bei kleinen Projekten zu nur vier „Dokumentensätzen“ zusammengefasst werden können und auch tw. mündlich berichtet werden können.

Bei der Arbeit an den einzelnen Prozessen im Rahmen der Managementphasen werden diese 7 von PRINCE2 im Detail beschriebenen „Themen“ bearbeitet:

1. Business Case
2. Organisation
3. Qualität
4. Pläne
5. Risiken
6. Änderungen
7. Fortschritt

PRINCE2 unterscheidet diese Rollen:

- Projekt-Lenkungsausschuss: führt den Prozess „Lenken des Projekts“ aus, bestehend aus den drei gleichwertigen Mitgliedern:
 - Auftraggebervertreter (= Eigentümer des Business Case)
 - Benutzervertreter
 - Vertreter der das Produkt entwickelnden Bereiche oder externen Firmen

- Projektmanager: verantwortlich für das Tagesgeschäft innerhalb der jeweils laufenden „Durchführungsphase“ innerhalb des von Lenkungsausschuss festgelegten Spielraums; beauftragt die Teammanager.
- Projektunterstützungsteam des Projektmanagers (optional, bei eher grossen Projekten)
- Projektsicherung: überprüft im Auftrag und im Interesse des Lenkungsausschusses die Projektergebnisse und den Projektstatus
- Teammanager: Verantwortlich für die Herstellung der Produkte gemäss Produktbeschreibungen
- Änderungsausschuss (optional): Handling von Änderungsanträgen und Spezifikationsabweichungen

Das „Spezielle“ in Hinblick auf den Umgang mit Unberechenbarkeiten und Komplexität:

PRINCE2 - als universelle PM-Methode - erlaubt sowohl ein betont inkrementell-adaptives als auch ein stark plangetriebenes Vorgehen. Das spezifische Vorgehen wird im projektspezifischen „tailoring“ festgelegt. Es findet Ausdruck insbesondere in der Art der Gestaltung der „Durchführungsphasen“:

Bei einem betont inkrementell-adaptiven Vorgehen werden die Durchführungsphasen dazu dienen, konkrete und vom Benutzer testbare oder gar produktiv nutzbare Teilergebnisse zu liefern, etwa:

- Spielfeld mit Containern als Umkleidekabinen
- Zuschauertribüne ohne Dach
- Tribünenüberdachung und Umkleideräume mit Duschen
- Flutlichtanlage
- Kiosk und Bar

Bei einem stark plangetriebenes Vorgehen wird es Durchführungsphasen dieser Art geben:

- Konzeption der gesamten Sportanlage
- Detaildesign der gesamten Sportanlage
- Bau der gesamten Sportanlage
- Abnahme der gesamten Sportanlage
- Einweisung der Betreiber der gesamten Sportanlage
- Freigabe der gesamten Sportanlage zur Nutzung

In beiden Fällen beruht PRINCE2 zur Reduktion der Komplexität auf dem Vertrauen in das autonome und sich selbst steuernde Agieren der Rollenträger innerhalb des definierten Rahmens und der übergeordneten Steuerung nach dem Ausnahmeprinzip nur beim Überschreiten dieses Rahmens.

Konsequenzen beim Praktizieren innerhalb „üblicher“ Organisationen:

Die deutsche Beschreibung von PRINCE2 (Erfolgreiche Projekte managen mit PRINCE2. The Stationery Office Ltd 2009) umfasst 377 Seiten. Das verleitet „übliche“ Organisationen mit ihrer traditionellen Regelungs- und Absicherungstradition dazu, beim „tailoring“ eher zu viel dieses Maximalumfangs stehen zu lassen als „radikal“ das nur unbedingt Nötige zu nehmen.

Gegen diesen „Hang zur Bürokratisierung“ anzukämpfen ist nicht einfach.

Zusätzlich muss vermieden werden, dass - weiterhin - möglichst früh allzu umfassende und detaillierte Konzeptionen („BDUF“) verlangt werden um „zuverlässige“ Schätzungen und Planungen zu erhalten. PRINCE2 zwingt nicht dazu (es will ja den „JEDUF“ fördern), es kann jedoch dazu missbraucht werden.

Für Organisationen mit einem ausgeprägten Bedürfnis nach granularer Kontrolle bedeutet das Steuern nach dem Ausnahmeprinzip und das autonome Agieren der Rollenträger einen erheblichen Kulturwandel.

Skalierbarkeit / Übertragbarkeit auf ganze Organisationen:

PRINCE2 ist anwendbar auf einzelne Projekte jeglicher Art und Grösse, muss aber für jedes Projekt entsprechend angepasst werden. Das Management mehrerer Projekte innerhalb eines Programms und

auch das Portfoliomanagement behandelt PRINCE2 allerdings nur am Rande und bietet dazu keine spezifischen Prozesse oder Methoden an.

PRINCE2 ist eine Projektmanagement-Methode, kein Modell zur Gestaltung von Organisationen.

Dynamic Systems Development Method (DSDM[®] / DSDM[®] Atern)

Charakteristik und Herkunft:

[vgl.: Matthew Caine: DSDM Atern im Überblick.

Online in: <http://www.netzwoche.ch/de-CH/News/2011/10/19/DSDM-Atern-im-Ueberblick.aspx>]

DSDM ist einerseits ein für sich alleinstehendes agiles „Framework“ zur projektmässigen Entwicklung dynamischer, also nicht genau vorausplanbarer, Systeme, einschliesslich eines Rahmens für das Projektmanagement. Andererseits fokussiert DSDM insbesondere auf die Steuerung der teambasierten Entwicklungsarbeit. Dieser Teil des DSDM kann auch im Rahmen anderer übergeordneter Projektmanagementmethoden wie PRINCE2[™] [siehe <http://www.infoq.com/news/XP-DSDM-Prince2>] oder PMI eingesetzt werden. DSDM ist mit seinen Phasen, Prozessen und Outputs aus „traditioneller“ Sicht gut verständlich, andererseits ist es durch essentielle agile Elemente gekennzeichnet: Es beruht auf einem iterativ-adaptiven und inkrementellen Entwicklungsvorgehen und dem fortlaufenden Einbezug der Nutzer und Kunden in den Entwicklungsprozess. Es unterstützt seine Ausrichtung auf klar definierte strategische Ziele und auf die frühe Lieferung echter Geschäftsvorteile. Deren Realisierungsreihenfolge folgt einer „Prioritized Requirement List“ (PRL) basierend auf den „MoSCoW-Regeln“. In den frühen Phasen wird gemäss JEDUF nur das für den jeweils nächsten iterativen Realisierungsschritt wirklich Erforderliche (und Machbare) konzipiert und geplant.

DSDM wurde 1997 erstmals publiziert, damals jedoch beschränkt auf eine reine Software-Entwicklungsmethode mit dem Ziel, im „Rapid Application Development (RAD)“ etwas mehr Disziplin zu schaffen. Im Jahr 2007 wandelte es sich unter dem Namen „DSDM Atern“ zu einer generischen Vorgehensweise des Projektmanagements und zur agilen Entwicklung von Lösungen.

(„Atern“ ist die Abkürzung von „Arctic Tern“, der Küstenseeschwalbe, einer Vogelart, die im Schwarm extrem lange Strecken dank einem hohen Mass an Kollaboration zurücklegen kann)

Auch das dem „DSDM Atern“ vorangehende und immer noch weit verbreitete DSDM 4.2 (aus dem Jahr 2003) integrierte bereits eXtreme Programming für die agile Softwareentwicklung.

DSDM wurde entwickelt und wird weiterentwickelt vom 1994 gegründeten „DSDM Consortium“. Es versteht sich als eine herstellerunabhängige non-profit-Organisation.

Das DSDM-Consortium wird finanziert durch jährliche Mitgliederbeiträge (dreistellige Eurobeträge) der das DSDM nutzenden Firmen. DSDM ist das einzige als ISO9001-verträglich akkreditierte agile Verfahren. Alle zur Nutzung von DSDM und DSDM Atern erforderlichen Beschreibungen und Vorlagen sind online entweder frei und kostenlos verfügbar oder zum Preis üblicher Fachbücher zu erwerben.

DSDM ist bis jetzt ausserhalb Grossbritannien kaum bekannt, wird dort aber insbesondere auch für grosse Projekte - auch im Bereich der öffentlichen Verwaltung - als eines der Standards (nebst oder auch in Verbindung mit PRINCE2) genutzt. [Siehe einige Fallbeispiele in [Brian Wernham, Agile Project Management for Government. Maitland and Strong, 2012]

Vorgehen und Rollen:

[Details: Kompakte Darstellung von DSDM Atern in Deutsch:

http://www.dsdm.org/Ebooks/DSDM_Atern_Taschenbuch/index.html# und

<http://www.netzwoche.ch/de-CH/News/2011/10/19/DSDM-Atern-im-Ueberblick.aspx>

Etwas umfangreicher in Englisch: <http://www.dsdm.org/atern-handbook/flash.html>]

DSDM Atern-Projekte bestehen aus *sieben Phasen*. Die Ergebnisse und Erkenntnisse jeder Phase können zur Überarbeitung der Ergebnisse der vorangehenden Phasen führen.

- Projektvorbereitung: Eröffnungsphase des Projekts, Festlegung der Rahmenbedingungen
- Machbarkeitsprüfung: Begründung des Projekts

- Rahmenplanung / Grundlagen: Iterativ-inkrementelle Phase zur Klärung (i.S. eines JEDUF) von Umfang, technischen Rahmenbedingungen und Standards
- Feinplanung / Forschung: Iterativ-inkrementelle Entwicklungsphase zur Demonstration (z.B. mittels Prototypen) der Funktionalität der Lösung
- Technische Entwicklung: Iterativ-inkrementelle Entwicklungsphase zur Realisierung der Lösung
- Bereitstellung / Auslieferung: Jedes einzelne Lösungsincrement für sich oder Pakete einzelner oder aller Inkremente werden dem Kunden zur Verfügung gestellt.
- Projekt-Nachbereitung: Abschliessende Auswertung der erreichten Nutzenmaximierung

Einzelne diese Phasen können je nach Projekt und Risiko zusammengelegt werden oder zeitversetzt überlappend ablaufen oder zu Phasengruppen (z.B. pro Lösungsincrement) zusammengefasst und pro Lösungsincrement iterativ durchlaufen werden, z.B. so:
(Forschung + Entwicklung > Auslieferung) → (Forschung > Entwicklung > Entwicklung > Entwicklung > Auslieferung) → (Forschung > Entwicklung > Auslieferung) → etc.

Jede einzelne Phase dient bei allen Ablaufvarianten der Herstellung phasenspezifisch definierter Ergebnisse, deren Form und Inhalt vom jeweiligen Projekt abhängen. Jede Phase dient dazu, konkrete Ergebnisse zu liefern und ist nicht nur eine Zusammenfassung von Aktivitäten.

Taktgebend für die Projektarbeit ist das „*Timeboxing*“:

- Das Projekt als Ganzes ist eine Timebox (= fester Endtermin)
- Das gesamte Projekt umfasst eine oder mehrere Inkrement-Timeboxen mit jeweils festen Endterminen zur Auslieferung von Lösungs-Inkrementen.
- Jede Inkrement-Timebox besteht aus einer Feinplanungs / Forschungs - Timebox, einer oder mehreren Entwicklungs-Timeboxes (10 bis 30 Tage) und einer Bereitstellungs / Auslieferungs - Timebox.
- Jede Timebox aller Arten und Ebenen beginnt mit einem „kick off“ gefolgt von einer „Untersuchungs-“, einer „Verfeinerungs-“ und einer „Konsolidierungs-Iteration“. Jede Iteration startet mit einer Planung und endet mit einem Review.
- Innerhalb jeder dieser Iterationen findet ein tägliches „stand-up-Meeting“ statt.

DSDM Atern liegen *acht Arbeitsprinzipien* zugrunde:

- Konzentration auf Kundenbedürfnisse
- Pünktliche Lieferung
- Zusammenarbeit
- Keine Kompromisse in Bezug auf Qualität
- Eine solide Grundlage schaffen und schrittweise darauf aufbauen
- Iterative Entwicklung
- Kontinuierliche und klare Kommunikation
- Kontrolle demonstrieren

Rollen im Projekt

- Business Sponsor: Hauptverantwortlich für die Entwicklung des Geschäftsplans; sichert Finanzierung und Ressourcen.
- Projektmanager: Verantwortlich für die Projektleitung: Gesamtplanung; überwacht Fortschritte, Verfügbarkeit von Ressourcen und Projektkonfiguration; Risikomanagement und Unterstützung der Teams.
- Business Visionary: Hauptverantwortlich für die Entwicklung und Beachtung der Business Vision. Mitwirkung bei der Ausarbeitung der wesentlichen Anforderungen und des Designs
- Technischer Koordinator: Abstimmung und Kontrolle der technischen Architektur und technischen Risiken

- Teamleiter der Lösungsentwicklung: Fördert die Beteiligung des Gesamtteams; Detailplanung von Timebox-Inhalten sowie täglichen Arbeitsabläufen und –inhalten

Rollen in der Lösungsentwicklung

- Teamleiter der Lösungsentwicklung: wie oben beschrieben
- Business-Repräsentant: Repräsentiert die Sicht des Kunden bei täglichen Entscheidungen; beschreibt Business-Szenarios
- Lösungs-Entwickler: Entwicklung der Lösung und Mitwirkung bei deren Qualitätssicherung.
- Lösungs-Tester: Entwickelt Test-Szenarios; Durchführung umfassender Tests
- Wirtschaftsanalyst: Förderung der Kommunikation zwischen Business-Repräsentanten und den technischen Mitarbeitern des Teams; Gewährleistet, dass die betrieblichen Konsequenzen von ad-hoc-Entscheidungen ausreichend durchdacht werden
- Unternehmensberater: Bringt Fachwissen aus der Branche des Kunden ein; meist ist es ein künftigen Benutzer

Das „Spezielle“ in Hinblick auf den Umgang mit Unberechenbarkeiten und Komplexität

- Das inhaltliche Projektergebnis wird anpassbar gehalten, Kosten, Termine und Qualitätsanforderungen 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)

Konsequenzen beim Praktizieren innerhalb „üblicher“ Organisationen

DSDM Altern definiert - im Unterschied zu Scrum - keine „ungewöhnlichen“ und neuen Rollen und Verantwortlichkeiten sondern solche, die auf die im Unternehmen bereits bestehenden Rollen und Positionen übertragbar sind.

Wie alle agilen Vorgehensweisen bleibt auch bei DSDM Altern der Projektscope flexibel, nur Kosten, Termin und Qualitätsanforderungen werden fixiert. Das, und das Prinzip „JEDUF statt BDUF“ erfordern ein grundlegendes Umdenken und führt zu weitreichenden Konsequenzen für das Controlling, Vertragswesen und die langfristigen Portfolio-Roadmaps.

All das erfordert eine stark vertrauens- statt nur vertragsbasierte Zusammenarbeit mit den Kunden und Lösungsnutzern.

Das modellierungs- und prototyporientierte Arbeiten ist nur möglich mit einem intensiven Einbezug der späteren Lösungsnutzer. Das hat oft erhebliche Konsequenzen auf deren Termin- und Ressourcenplanung und die geographische Verfügbarkeit in Konkurrenz zur üblichen Arbeit.

Das Projektteam muss ermächtigt sein, auch wichtige Entscheide zu treffen oder rasch zur Diskussion zu stellen, ohne umfangreiche formale Anträge an das höhere Management schreiben zu müssen.

Skalierbarkeit / Übertragbarkeit auf ganze Organisationen

DSDM / DSDM Altern entspricht mit seinen Phasen und Rollen dem, was bei grossen Projekten üblich ist und erwartet wird - auch aus Sicht des „traditionellen“ Projektmanagements. Damit ermöglicht es insbesondere grossen Organisationen einen anschlussfähigen und evolutionären Übergang in Richtung eines inkrementell-adaptiven Vorgehens im jeweils machbaren und nötigen Ausmass. Mit seinen fünf Rollen im Bereich Projektmanagement und seinen fünf Rollen im Bereich der Lösungsentwicklung bietet es ausreichende und nicht allzu differenzierte Rollen auch für grosse

Projekte. Die sehr flexible Handhabbarkeit und Kombinierbarkeit seiner sieben Phasen macht es anpassbar an verschiedenste Arten und Grössen von Projekten.

DSDM / DSDM Altern bietet jedoch keine Mechanismen der Steuerung von Multiprojekt-Landschaften auf Programm- und Portfolioebene. Dafür muss es mit anderen Frameworks des Programm- und Portfoliomanagements erweitert werden, etwa mit dem „Scaled Agile Framework for Enterprise“ (SAFe) [<http://scaledagileframework.com/>]

DSDM / DSDM Altern versteht sich als agiles Framework zur Realisierung von Projekten und nimmt für sich nicht in Anspruch, ein Framework auch der agilen Unternehmensführung zu sein.

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

Wie weiter oben 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 [<http://www.scrum.org/Scrum-Guides>] zu lesen: „So lange 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.“

Scrum

Charakteristik und Herkunft

Scrum beschreibt kein Projektvorgehen von der Projektinitialisierung bis zum Projektabschluss sondern ist ein Framework zum Vorgehen bei der Neu- und Weiterentwicklung (Detailanalyse, Design, Implementation) komplexer, im Voraus also nicht genau definierbarer und planbarer, Produkte auf 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 erst dort, wo das Projekt bereits bewilligt, seine Finanzierung gesichert und der Projektplan und das Entwicklerteam schon vorhanden waren. [Ken Schwaber: Agiles Projektmanagement mit Scrum. Microsoft Press Deutschland, 2007. S. 72]

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 potentiell auslieferbares Produktinkrement herstellen kann. 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 wird - unter Einbezug relevanter teamexterner Stakeholder - das effektiv Geleistete und die Arbeitsweise während des Sprints überprüft.

Das - und nicht mehr oder weniger - beschreiben die Regeln von Scrum. Sie beschreiben z.B. nicht, wie die relativ detaillierten 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 etliche bis mehrere Dutzend Teams parallel an verschiedenen Produkten und Projekten arbeiten können. Scrum beschreibt auch kein 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 Design, der Modellierung, der Konstruktion / Realisierung, der Tests und der Qualitätssicherungsmassnahmen und der Inbetriebnahme und fortlaufenden Produktbetreuung muss ergänzend erarbeitet und vereinbart werden.

Scrum will alle diese spezifischen Aspekte auch nicht regeln sondern ist [gemäss Scrum Guide <http://www.scrum.org/Scrum-Guides>] „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.“

Das ist einer der Hintergründe auch für diese Aussage im Scrum Guide:

„Scrum ist:

- Leichtgewichtig
- Einfach zu verstehen
- Extrem schwer zu meistern“

So gesehen ist Scrum als solches ein allgemeines Vorgehens-Framework für jene Art der Teamarbeit, bei der:

- In kurzen Abständen (d.h. pro Sprint von max. 4 Wochen) nutzbare und aufeinander aufbauende Teile irgend eines Produkts von einem Entwicklerteam für teamexterne Nutzer hergestellt werden können.
- Mindestens drei bis maximal neun Entwickler zu 100% nur in diesem einen Team sehr eng und sich selbst organisierend zusammenarbeiten.
- Die Erfahrungen mit den bereits realisierten Produktinkrementen in die Realisierung der künftigen so rasch als möglich einfließen und sich das Produkt somit fortlaufend verändern kann.
- Als Basis der Teamarbeit eine von einem und nur einem unternehmensweit anerkannten „Product Owner“ fortlaufend aktualisierte Liste („Product Backlog“) mit all dem existiert, was im Produkt künftig benötigt werden könnte.

Für solche Produkte (oder Arbeiten i.a.) passt die Teamarbeit gemäss Scrum daher nicht:

- Es sind keine „vernünftigen“ pro Sprint (alle 2 bis 4 Wochen) verfügbaren und zumindest „potentiell“ von teamexternen Stakeholdern nutzbaren Teilergebnisse (Produktinkremente) denkbar
- Es gibt keine pro Sprintergebnis alle 2 bis 4 Wochen überprüfungs- und nutzungsbereiten repräsentativen Stakeholder.
- Das Produkt ist von Anfang an im Detail definiert. Veränderungen sind nur ausnahmsweise möglich.
- Ein von einem und nur einem Product Owner verantwortetes und veränderliches Product Backlog ist nicht denkbar.
- Die Entwickler arbeiten gleichzeitig und mit immer wieder anderen Entwicklern an diversen Produkten.

Jeff Sutherland schuf das Scrum-Vorgehen 1993. Mit dem Wort "Scrum" bezog er sich auf eine 1986 im Harvard Business Review publizierte Studie von Takeuchi und Nonaka, in der "high-performing, cross-functional teams" mit der Scrum-Formation im Rugby verglichen wurden. Ken Schwaber formalisierte den Prozess für die Software-Industrie in einem Artikel für die OOPSLA 1995. Jeff Sutherland und Ken Schwaber beschreiben im „Scrum Guide“ [<http://www.scrum.org/Scrum-Guides>] als „gültigen Leitfaden für Scrum“ die verbindlichen „Spielregeln“, die „untastbar“ sind: „Obwohl es möglich ist, nur Teile von Scrum zu implementieren, ist das Ergebnis nicht Scrum. Scrum existiert nur in seiner Gesamtheit“. Dieses kompromisslose Einhalten der Scrum-Regeln vergleichen Jeff Sutherland und Ken Schwaber mit den Regeln des Schach, die ebenfalls kompromisslos einzuhalten sind, wenn das Spiel „Schach“ genannt wird.

[<http://www.scrum.org/Portals/0/Documents/Scrum%20Updates/Scrum%20Update%202011.pdf>]

Ken Schwaber ist jedoch kein „Scrum-Dogmatiker“ sondern schreibt ausdrücklich:

Scrum can be replaced or superseded by anything that also supports its underlying principles:
1. Self-organization – people doing complex work are much more effective organizing themselves and the work than someone who isn't doing the work.

2. *Bottom-up intelligence* – figuring out how to do work is a management activity best performed by the people doing the work, since the work is unpredictable, with many twists and turns.

3. *Empiricism* – it is hard to plan what you don't know, so we instead see what has been accomplished, and then figure out what to do next. We do this frequently to control risk and determine the best path to our goal.

4. *Transparency* – we periodically have to know what is actually happening to make effective empirical decisions.

[<http://kenschwaber.wordpress.com/2012/10/05/what-comes-after-scrum/>]

Scrum kann ohne Lizenzen von allen Personen (auch ohne Zertifizierung) und Firmen auf Basis des frei verfügbaren „Scrum Guide“ [<http://www.scrum.org/Scrum-Guides>] und einer fast unüberblickbaren Literatur zu Scrum eingesetzt und geschult werden.

Sowohl die „Scrum Alliance“ als auch Ken Schwaber's „Scrum.org“ und deren Scrum-Trainer und Scrum-Coaches bieten weltweit kommerzielle Trainings und diverse „Zertifizierungen“ (Scrum Alliance) bzw. „Assessments“ (Scrum.org) für verschiedene Scrum-Rollen und Niveaus an.

Vorgehen und Rollen

[Details: Einen raschen Überblick bieten <http://www.scrum.org/Resources/What-is-Scrum> und

http://www.scrumalliance.org/pages/what_is_scrum

Alle Regeln zu Scrum in Deutsch auf nur 15 Textseiten:

<http://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum%20Guide%20-%20DE.pdf>]

Das „*Scrum Team*“ besteht aus diesen Rollen:

- *Product Owner*: Verantwortlich für den Mehrwert des Produkts und der vom Team gelieferten Produktinkremente.
- *Entwicklungs-Team*: Erstellt bis zum Ende eines jeden Sprints ein fertiges und potentiell auslieferbares Produktinkrement.
- *Scrum Master*: Verantwortlich, dass Scrum verstanden und korrekt angewandt wird und das Team effizient und mit bestmöglicher Qualität arbeiten kann.

Scrum umfasst diese „Ereignisse“:

- *Sprint*: Festes Zeitfenster von 2 - 4 Wochen zur Erstellung eines fertigen Produktinkrements
- *Sprint Planning*: Planung der im nächsten Sprint zu erledigenden Arbeit
- *Daily Scrum*: Tägliches 15-minütiges „standup“ des Scrum Teams.
- *Sprint Review*: Scrum Team und Stakeholder überprüfen gemeinsam, was im soeben abgeschlossenen Sprint fertig gestellt wurde.
- *Sprint Retrospective*: Zum Sprintende sucht das Scrum Team nach weiteren Verbesserungen seiner Entwicklungsarbeit.

Scrum basiert auf zwei Artefakten:

- *Product Backlog*: Vom Product Owner laufend aktualisierte geordnete Liste mit allem, was in dem Produkt künftig benötigt werden könnte.
- *Sprint Backlog*: Die zur Erfüllung des Sprintziels für den Sprint ausgewählten Product Backlog Einträge sowie der vom Scrum Teams für das nächste Produktinkrement prognostizierte Lieferumfang.

Die Produktneu- und -weiterentwicklung verläuft bei Scrum in einer Abfolge von stets gleich langen Iterationen, **Sprints** genannt, mit einer Dauer von maximal vier Wochen. Scrum beschreibt keine den Sprints übergeordneten Phasen, Milestones oder Releases, verbietet sie aber auch nicht.

Die Aufgaben für einen solchen Sprint werden aus einem **Product Backlog** genährt. Der Product Backlog ist eine nach Erledigungsreihenfolge geordnete Liste von Aufgaben, welche im Rahmen des Gesamtvorhabens erledigt werden müssen. Die Liste enthält z.B. Spezifikationen von zu realisierenden Komponenten, Fehlerbehebungen, notwendige Infrastrukturarbeiten und Installationen oder Ähnliches.

Der Product Backlog wird ausschliesslich vom **Product Owner** verwaltet und gepflegt. Der Product Owner allein als eine und nur eine Person (ev. als entscheidungsbefugter Repräsentant eines Gremiums) ist verantwortlich gegenüber dem *gesamten* das Produkt liefernden Unternehmen (und nicht nur gegenüber seinem Team), alle notwendigen Anforderungen aufzunehmen, zu ordnen, allfällig aufzuteilen und in den Product Backlog einzutragen. Er ist (Zitat aus dem Scrum Guide) „für die Maximierung des Wertes des Produkts und der Arbeit, die das Entwicklungs-Team verrichtet, verantwortlich.“ Er sollte daher - mit Bezug auf den zweiten Halbsatz - dem Produktlieferanten und nicht dem Kunden angehören. Er vertritt jedoch innerhalb des Lieferanten die Interessen des Kunden, abgestimmt auf die Unternehmensinteressen des Lieferanten.

Das **Entwicklungsteam** erhält die Vorgaben, was es zu realisieren hat, von einem und nur einem einzigen Product Owner. „Niemand darf das Entwicklungs-Team anweisen, mit anderen Anforderungen als den im Product Backlog festgelegten zu arbeiten und dem Entwicklungs-Team ist es nicht erlaubt, Arbeitsanweisungen von anderen Personen als dem Product Owner anzunehmen.“ (Zitat aus Scrum Guide). Es besteht aus nur als „Entwickler“ bezeichneten Personen (alle für die Entwicklung notwendigen Spezialisten, Tester, Administratoren, etc) als sich selbst organisierendes Team. „Nur die Mitglieder des Entwicklungs-Teams erstellen dieses (Produkt-) Inkrement.“ (Zitat aus Scrum Guide).

Jeder Sprint liefert ein - zumindest potentiell - nutzbares weiteres „**Produktinkrement**“, welches aus Nutzersicht überprüft oder - noch besser - von den Nutzern verwendet werden kann. Diese Überprüfung aus Nutzersicht bzw. die konkrete Nutzungserfahrung der Teillösung wird für möglichst rasche Anpassungen des Product Backlogs genutzt. Sprints, die keine solchen von teamexternen Stakeholdern (insbes. Kunden oder Nutzer) validierbaren oder nutzbaren Produktteile liefern, entsprechen nicht den Scrum-Regeln.

Zu Beginn eines Sprints wird zwischen dem Entwicklungsteam und dem Product Owner im sogenannten **Sprint Planning Meeting** das Sprint-Ziel formuliert und eine Prognose gemacht, was während des folgenden Sprints zur Erreichung des Ziels umgesetzt werden kann. Basis der Prognose sind der geschätzte Umfang der anstehenden Aufgaben bezogen auf jenen Arbeitsumfang, der bisher pro Sprint geleistet werden konnte. Diese vom Team prognostizierten Aufgaben bilden den **Sprint Backlog**. Während des Sprints verfolgt das ganze Scrum Team täglich im maximal 15-minütigen **Daily Scrum** die gestern erledigte und die für heute geplante Arbeit, teilt Erkenntnisse und identifiziert Verbesserungsnotwendigkeiten. Am Ende des Sprints demonstriert das Entwicklungsteam im **Sprint Review** den teamexternen Stakeholdern (und nicht nur dem Product Owner!) das gemäss Sprint-Ziel fertiggestellte Produktinkrement möglichst konkret und holt Feedbacks dazu ein. Dabei werden auch neue oder geänderte Anforderungen für den nächsten oder für einen der nächsten Sprints formuliert. In der darauf folgenden **Sprint Retrospektive** werden im ganzen Scrum Team (nicht nur im Kreis der Entwickler) das Vorgehen, die Arbeitsmethoden und die Zusammenarbeit des abgeschlossenen Sprints besprochen, das Funktionierende hervorgehoben und Verbesserungen vereinbart.

Der **Scrum Master** agiert bei all dem als „servant leader“ des Teams. Er ist (Zitat aus Scrum Guide) „dafür verantwortlich, dass Scrum verstanden und korrekt angewandt wird. Dies erreichen Scrum Master durch die Sicherstellung, dass das Scrum Team sich konform zur Scrum Theorie verhält und die gültigen Praktiken und Regeln einhält... Der Scrum Master unterstützt die Akteure außerhalb des Scrum Teams dabei, zu verstehen, welche ihrer Verhaltensweisen in Bezug auf das Scrum Team hilfreich sind und welche nicht. Der Scrum Master hilft außerdem, diese Verhaltensweisen so zu verändern, dass das Scrum Team einen maximalen Geschäftswert erzeugen kann.“

Scrum legt weder Organisationsform noch Führungsrollen fest. Obschon viele den Scrum Master als Projektleiter missverstehen, kennt Scrum, da es ein Framework für die (kontinuierliche) PRODUKTentwicklung und keines für das PROJEKTmanagement ist, die Rolle des klassischen Projektleiters als Rolle innerhalb des Scrum-Teams nicht.

Andere Rollen neben den Scrum-Rollen sind ausserhalb des Scrum-Teams aber weiterhin möglich. So etwa kann ein Produktmanager oder ein Gesamtprojektleiter auf der den Scrum-Teams übergeordneten

Ebene (etwa des Programm-Managements) angesiedelt sein und dort mit den Product Ownern eng zusammenarbeiten.

Missverständnisse und Zuschreibungen

Scrum ist als Vorgehens-Framework der Teamarbeit im Rahmen der Produktentwicklung überall dort sehr erfolgreich, wo die vorhin genannten Voraussetzungen gegeben sind. Das zeigt die nahezu unüberblickbare Menge publizierter Beispiele. Andererseits bringt auch die Nutzung nur einiger weniger agiler Elemente, die mit „Scrum“ assoziiert werden, wie das „Daily Standup Meeting“ oder ein kurzer „sprintartiger“ Arbeitszyklus auch ohne potentiell nutzbare Produktinkremente aber mit sichtbaren Zwischenresultaten, schon grosse Vorteile. Das allein wird - leider - sehr oft schon als „Scrum-Vorgehen“ bezeichnet.

Die sich daraus ergebende Popularität von „Scrum“ ist jedoch, bezogen auf „Scrum als solches“ (also gemäss Scrum Guide) Quelle etlicher Missverständnisse und Zuschreibungen:

- Scrum ist keine schlanke Entwicklungsmethode. Es ist ein allgemeines Vorgehens-Framework und ein „verfügbarer leerer Container“ für etablierte spezifische Methoden der professionellen Produktentwicklung die es zusätzlich zu beachten gilt.
- Scrum führt nicht zu einer massiven Erhöhung der Effizienz: Es unterstützt die Adaptivität, Qualität und Effektivität. Insgesamt (bezogen auf ein gesamtes „fertiges“ Projekt) wird es mit Scrum nicht eindeutig schneller oder billiger. [Mark Harwardt: Wasserfallmodell versus Scrum - Ist der gute Ruf der agilen Methode gerechtfertigt?. Masterarbeit. Hagen: Fernuniversität Hagen Lehrgebiet Programmiersysteme, 2008]
- Scrum erfordert die strikte Professionalität erfahrener Entwickler. Welche spezifischen Methoden und Techniken erforderlich sind, beschreibt Scrum als „Framework“ und „leerer Container“ nicht sondern setzt ihre Beherrschung stillschweigend voraus.
- Scrum legalisiert kein „Arbeiten auf Zuruf“ sondern setzt ein agiles und professionelles Anforderungs-Engineering 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 teamexternen Spezialisten) so neu- oder weiterentwickelt entwickelt wird, dass LIEFERBARE TEILE davon nach jeweils bereits 2 - 4 Wochen vorhanden sind. Und zwar so, dass die RASCHEN Rückmeldungen ihrer Nutzer UNMITTELBAR zur Steuerung der weiteren inhaltlichen Entwicklung dienen.

Nicht angemessen ist das z.B. bei:

- technologisch sehr heterogenen Produkten mit langwierigen Integrationsprozessen
- Vorhaben, deren Anforderungen und Lösungskonzeptionen bereits zu Beginn gut modellierbar und stabil sind
- im Voraus genehmigungspflichtigen Produkten ohne fortlaufende Adaptionenmöglichkeit
- physischen Produkten, die nur als Ganzes funktionieren
- Änderungen des schon Gelieferten, die lange dauern und sehr teuer sind
- Organisationsentwicklungen u.ä. Vorhaben ohne Produktcharakter
- Selbstorganisation braucht Führung: Das Team als Ganze ist selbstverantwortlich - jedoch innerhalb klar vorgegebener Spielregeln und Rahmenbedingungen. Das, WAS zu erledigen ist, legt der Product Owner fest, für den Rahmen des effizienten und qualitätsbewussten WIE ist der „servant leader“ des Teams (= der Scrum Master) verantwortlich. Es hat sich jedoch die Ansicht verbreitet, dass der ScrumMaster nur als „Coach“ ohne Führungskompetenz agiert. Das aber widerspricht seiner im Scrum Guide klar formulierten Verantwortung. Und es entsteht in Unternehmen eine erhebliche „Führungslücke“. [vgl. Hans-Peter Korn: Redesigned Agile in grossen Unternehmen. In: Hans-Peter Korn (Hsg), Jean Pierre Berchez (Hsg): Agiles IT-Management in großen Unternehmen. Symposium Publishing, 2013] Die Behauptung, dass ein Teamleiter mit Führungsverantwortung die Selbstorganisation des Teams verhindert ist Ausdruck einer spezifischen auf „hire and fire“ und einer von detailliertem „command and control“ geprägten Führungskultur. Sie ist sicher nicht für alle Unternehmen generalisierbar.

Das „Spezielle“ in Hinblick auf den Umgang mit Unberechenbarkeiten und Komplexität

Zentral bei Scrum sind [vgl. den Scrum Guide] die drei Säulen: **Transparenz - Überprüfung - Anpassung**. Die wesentlichen Aspekte des Prozesses (und nur das und nicht alle Vorkommnisse z.B. innerhalb des Teams) müssen für diejenigen Personen *transparent* sein, die für das Prozessergebnis verantwortlich sind. Transparenz setzt einen dafür definierten gemeinsamen Standard voraus. Produkt- und Sprint Backlogs und der Fortschritt auf dem Weg zur Erreichung des Ziels müssen ständig durch sachkundige Personen am Ort des Geschehens *überprüft* werden, um unerwünschte Abweichungen rasch zu entdecken. Das geschieht via „Statusberichte“ sondern täglich (beim Daily Scrum) und bei jedem Sprint Review. Wird eine Abweichung festgestellt, dann muss so rasch als möglich - auch innerhalb des Sprints - das Vorgehen oder der Arbeitsgegenstand *angepasst* werden.

Überprüft werden nur konkrete, „fertige“ und zumindest potentiell nutzbare Produktinkremente und z.B. keine blossen Beschreibungen von Konzeptionen.

Das initiale erstellte Produkt Backlog kann im weiteren Verlauf jederzeit - aber nur vom Product Owner - verändert werden.

Zur besseren Handhabung der Komplexität ist ein und nur ein Product Owner für das WAS (das Produkt Backlog), nicht aber für das WIE, verantwortlich. Das WIE - und nicht das WAS - verantwortet das Entwicklerteam.

Zur Vereinfachung und Beschleunigung der Kommunikation und der internen Organisation arbeiten die Entwickler vollamtlich und sehr eng als ein interdisziplinäres und sich selbst steuerndes Team, mit Vorteil in einem gemeinsamen Teamraum.

Der Fortschritt wird in erster Linie durch das Scrum Team als Ganzes selber überwacht, nicht durch teamexterne Kontrollinstanzen. Die Fortschrittskontrolle erfolgt mit möglichst einfachen, wenn immer möglich visuellen und manuell aktualisierten Mitteln. Der Fortschritt wird nicht mittels „geplante Stunden / verbrauchte Stunden“ sondern nur mittels „insgesamt nötige Arbeit / noch zu realisierte Ergebnisse“ gemessen. (Analogie: Bei einer Hausmauer ist nicht wichtig, wie viele Stunden bereits daran gebaut wurde sondern wie viel Prozent der Ziegel noch zu verbauen sind und wie viel Prozent der Bauzeit noch bevorstehen).

Konsequenzen beim Praktizieren innerhalb „üblicher“ Organisationen

➔ „Knackpunkt“ Product Owner:

In grossen Unternehmen führt die gemäss Scrum Guide 2011 definierte Rolle des Product-Owner als alleiniger und umfassender Repräsentant des „WAS“ zu Überschneidungen und Konflikten mit anderen Rollen. Diese müssen im Rahmen der Abstimmung des agilen Vorgehens mit der teamübergordneten Produktentwicklung und dem Portfolio-, Programm- und Release-Management und der teamübergreifende Koordination pro Produkt und Release bereinigt werden.

➔ „Knackpunkt“ Entwicklungsteam:

Für die teamarbeitsbasierte Produktentwicklung gemäss Scrum ist es in der Regel nötig, die bisherigen Teams neu zu strukturieren:

- Jedes Team ist interdisziplinär besetzt, verfügt also über permanente, stets im Team voll aufgelastete und daher ungeteilte Mitglieder / Spezialisten die insgesamt zum Definieren / Realisieren / Testen des vom Team entwickelten Produkts nötig sind.
- Teams müssen abgeschlossene Produktinkremente und nicht nur Teile davon, die erst zusammen mit Teilen anderer Teams funktionsfähig sind, liefern können.

➔ „Knackpunkt“ mittel- und längerfristige Planung und vorausgehende Basisarbeiten

Der Planungshorizont von Scrum bezüglich „Was ist wann mit welchen Ressourcen (Personal) vorhanden“ umfasst nur jeweils einen „Sprint“ (= max. 4 Wochen). Alle darüber hinaus gehenden Planungen benötigen zusätzliche Methoden des Release- und Portfoliomanagements ausserhalb von Scrum. Auch das Schaffen aller Voraussetzungen, damit die eigentliche Produktentwicklung gemäss Scrum beginnen kann, erfordert zusätzliche Massnahmen ausserhalb von Scrum.

→ „Knackpunkt“ Controlling

Die teambasierte Fortschrittskontrolle mit möglichst einfachen, visuellen und manuell aktualisierten Mitteln erschwert eine teamübergreifende und automatisierte Gesamtübersicht. An ihre Stelle tritt die fortlaufende Kommunikation mit den Product Ownern.

Die Fortschrittskontrolle anhand des Erledigungsstands der konkreten Arbeitsergebnisse anstatt auf Basis von „Personentagen“ erfordert ein Umdenken und Neugestalten vieler etablierter Instrumente.

Skalierbarkeit / Übertragbarkeit auf ganze Organisationen

Scrum als Framework der PRODUKTentwicklung kann nicht für JEDE Art von Vorhaben eingesetzt werden sondern ist dann geeignet, wenn ein Produkt so neu- oder weiterentwickelt entwickelt wird, dass LIEFERBARE TEILE davon nach jeweils 2 - 4 Wochen (zumindest potentiell) nutzbar vorhanden sind. Es ist also kein Framework für sämtliche Aufgabenstellungen innerhalb einer Organisation.

Wie aber passt Scrum für jene Aufgaben in einem grossen Unternehmen, die der Neu- und Weiterentwicklung umfassender Produktsuiten durch mehreren Dutzend Teams dienen?

Das Scrum-Framework beschreibt nur diese Situation:

- EIN Scrum-Team allein
- arbeitet an nur EINEM Produkt
- das von EINEM Produkt Owner als „Kundenproxi“ vertreten wird.

Es gibt im Scrum Guide nur diesen einen Hinweis auf „mehrere Teams arbeiten an EINEM Produkt“: „Häufig arbeiten mehrere Scrum Teams zusammen am gleichen Produkt. Ein einziges Product Backlog wird zur Beschreibung der anstehenden Arbeit an dem Produkt verwendet. Dazu wird ein Attribut zur Gruppierung der Product Backlog Einträge eingeführt.“

Für solche Situationen:

- Mehrere Teams arbeiten an mehreren Produkten
- Pro Produkt oder Produktgruppe gibt es Releases mit dazugehörigen teamübergreifenden Planungen und Reviews
- Koordination der „Team-Backlog Owner“ (sie vertreten gegenüber ihrem Team die Anforderungen mehrere Produkte) und der „eigentlichen“ Produkt Owner (bzw. der Produkt Manager)

müssen die Scrum-Spielregeln entsprechend erweitert werden. Etwa auf Basis des „Scaled Agile Framework for Enterprise“ [<http://scaledagileframework.com>] oder - bezogen auf grosse Projekte - mit DSDM oder PRINCE2 als übergeordneten Rahmen.

Sehr oft folgt in grösseren Unternehmen die Produktentwicklung insgesamt einem sequentiellen Phasenmodell basierend auf „Konzept - Realisierung - Einführung“. Ein inkrementell-adaptives Vorgehen i.S. des „Scrum Guide“ ist dann nur begrenzt und modifiziert als „Teamarbeitstechnik“ primär innerhalb der Realisierungsphase möglich. [vgl. Studie HERMES und Agilität, Eidgenössisches

Finanzdepartement EFD Informatikstrategieorgan Bund ISB, S. 12. Online in:

http://www.hermes.admin.ch/dienstleistungen/hilfsmittel/projektmanagement-generell/hermes-und-agilitat-am-beispiel-von-scrum/hermes-und-agilitat-am-beispiel-von-scrum/at_download/file]

Bei diesen „hybriden“ Vorgehensweisen werden die Phasen „Konzept - Realisierung - Einführung“ jeweils in Form von Sprints erledigt, diese Sprints liefern jedoch in der Phase „Konzept“ vor allem „Papier“ (funktionale und nichtfunktionale Spezifikationen, Architekturvorgaben, etc) und kaum konkrete (potentiell) nutzbare Produktinkremente. Und sehr oft werden die in den späteren Sprints realisierten Produktinkremente zunächst nur gesammelt und erst zum Schluss - durchaus auch in nur einer Lieferung - den Nutzern übergeben. „Inspect - adapt“ als Kern des agilen Prinzips ist damit wesentlich reduziert.

KANBAN / Kanban

Charakteristik und Herkunft

KANBAN (in Grossbuchstaben) ist eine 2007 veröffentlichte Methode zur „evolutionären Verbesserung“ des Prozesses der Softwareentwicklung auf den Grundlagen des „Lean Managements“ mit dem Ziel, den Arbeitsfluss sichtbar zu machen und zu beschleunigen.

„Kanban“ (in Kleinbuchstaben) ist ein für die Softwareentwicklung stark abgewandeltes dezentrales Selbststeuerungssystem auf der Grundlage der bei der Toyota Motor Corporation entwickelten Methode der Produktionsablaufsteuerung auf Basis des „Pull-Prinzips. Unter „Pull-Prinzip“ wird in diesem Kontext verstanden, dass bei einer Abfolge von Bearbeitungsschritten jeder Schritt die nächste Aufgabe vom vorgelagerten Schritt sich dann selber „holt“, wenn bei ihm Kapazität dafür vorhanden ist und sein Output vom nachfolgenden Bearbeitungsschritt benötigt wird. Das steht im Gegensatz zu „push“, bei dem ein Schritt seine Ergebnisse zur weiteren Bearbeitung an den nächsten Schritt „unbefragt“ weiterleitet oder eine zentrale Steuerungsinstanz die Arbeiten verteilt.

Im Ursprung ist Kanban ein in der Logistik und Produktion etabliertes und 1947 von Taiichi Ohno in der japanischen Toyota Motor Corporation entwickeltes dezentrales Produktions-Steuerungssystem. David J. Anderson suchte Ende des letzten Jahrtausends als Manager der SW-Entwicklung in verschiedenen Firmen Wege, um die etablierten, aber suboptimalen, Vorgehensweisen in Entwicklungsteams zu verbessern. Dabei erkannte er, dass jede Teamsituation spezifisch ist und es keine generell passenden Prozesse gibt. Bei seiner Arbeit am Buch „Agile Management for Software Engineering“ [David Anderson: Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results. Prentice Hall International, 2003] suchte er einen neuen Lösungsansatz für seine Anliegen auf Basis der „Engpassstheorie“ [Eliyahu M. Goldratt: What Is This Thing Called Theory of Constraints. North River Pr Inc, 1990]. Seine Lösungsidee war, einen Prozess zu beschreiben, mit dem jedes Team selber den seinen kontinuierlichen Arbeitsfluss begrenzenden Engpass finden und erweitern kann. Daraus entwickelte Anderson einen einfachen auf dem „Pull-Prinzip“ basierenden „Drum-Buffer-Rope-Ansatz“ für die Softwareentwicklung. Diesen Ansatz konnte er 2004 bei Microsoft erfolgreich und ohne grosse Widerstände umsetzen und berichtete 2005 und 2006 an Konferenzen darüber. Donald G. Reinertsen [Donald G. Reinertsen: The Principles of Product Development Flow: Second Generation Lean Product Development. Celeritas Publishing, 2009] machte Anderson daraufhin aufmerksam, dass er all das als komplettes Kanban-System gestalten könne. 2006, als Entwicklungsleiter bei Corbis, führte er dort mit Erfolg Kanban in der SW-Entwicklung ein und schuf etliche Erweiterungen dafür. 2010 erschien sein auch heute noch grundlegendes Buch „Kanban: Successful Evolutionary Change for Your Technology Business“ [Deutsche Übersetzung: David J. Anderson: Kanban: Evolutionäres Change Management für IT-Organisationen. Dpunkt Verlag; 2011] in dem er einerseits das von ihm entwickelte „Kanban“ (in Kleinbuchstaben) zur operativen dezentralen (Selbst-)Steuerung der SW-Entwicklungsarbeit beschreibt und andererseits das „KANBAN“ (in Grossbuchstaben), welches der evolutionären Prozessverbesserung der SW-Entwicklung dient.

Vorgehen und Rollen

[Details: David J. Anderson: Kanban: Evolutionäres Change Management für IT-Organisationen. Dpunkt Verlag; 2011]

„Kanban“ (in Kleinbuchstaben) zur operativen dezentralen (Selbst-)Steuerung der Arbeit (vereinfachte Beschreibung):

Kanban gibt keinerlei Rollen vor sondern beschränkt sich auf diese Vorgehensregeln:

Die einzelnen - sequentiellen - Schritte eines Arbeitsflusses, wie er aktuell praktiziert wird, werden auf einer grossen Tafel als Spalten dargestellt. Zum Beispiel:

Analyse und Spezifikation - Programmierung und Unit-Test - Integration - Integrationstest. Jede dieser Spalten ist unterteilt in zwei weitere Spalten: „in Arbeit“ und „fertig“.

Ganz links gibt es zusätzlich eine Spalte „anstehende Arbeiten“ und ganz rechts eine Spalte „Bereit zur Abnahme“

Für jede Spalte (und Unterspalte) ist festgelegt, wie viel „Arbeitspakete“ maximal in dieser Spalte liegen dürfen. Das ist das „Work In Progress Limit“ (WIP-Limit) der Spalte.

In einem - z.B. wöchentlichen - „Nachschubmeeting“ werden von einer teamexternen auftraggebenden Instanz (Person oder Gremium) so viele Arbeitspakete (dargestellt in Stichworten auf Zettel) in die ganz linke Spalte „anstehende Arbeiten“ platziert bis ihr - im Sinn kurzer Warteschlangen möglichst tiefes - Limit erreicht ist.

Das Entwicklungsteam (oder die für den ersten Schritt der Entwicklungsarbeit zuständige Person oder Gruppe) zieht aus der Spalte „anstehende Arbeiten“ den Zettel eines dieser Arbeitspakete in die Spalte des ersten Bearbeitungsschrittes (in unserem Beispiel „Analyse und Spezifikation / in Arbeit“), aber nur dann, wenn dort das WIP-Limit der Arbeiten noch nicht erreicht ist.

Wenn die „Analyse und Spezifikation“ für dieses Arbeitspaket erledigt ist, wandert der Zettel in die Spalte „Analyse und Spezifikation / fertig“, aber nur dann, wenn dort das - möglichst tiefe - WIP-Limit der fertigen Arbeiten noch nicht erreicht ist (damit nicht allzu sehr „auf Halde“ gearbeitet wird).

Das Entwicklungsteam (oder die für den nächsten Schritt der Entwicklungsarbeit zuständige Person) zieht aus der Spalte „Analyse und Spezifikation / fertig“ einen Zettel eines dort vorhandenen Arbeitspakets in die Spalte des nächsten Bearbeitungsschrittes (in unserem Beispiel „Programmierung und Unit-Test / in Arbeit“), aber nur dann, wenn dort das WIP-Limit der Arbeiten noch nicht erreicht ist.

Das wird so lange fortgesetzt bis der Zettel eines Arbeitspakets in der Spalte „Bereit zur Abnahme“ platziert werden kann und von dort von der auftraggebenden Instanz oder einer von ihr beauftragten Abnahmeinstanz nach der Überprüfung entfernt wird.

Jeder Arbeitsschritt holt (Pull-Prinzip) vom vorgelagerten nur dann ein neues Arbeitspakete, wenn er dazu in der Lage ist (= sein WIP-Limit ist nicht ausgeschöpft). Immer dann, wenn der Arbeitsfluss infolge des bereits erreichten WIP-Limits einer Spalte stockt oder in einem Arbeitsschritt das WIP-Limit chronisch unterschritten wird, sind die Mitglieder des Entwicklungsteams oder die für die angrenzenden Spalten Zuständigen verpflichtet mitzuhelfen, den Stau zu beseitigen - oder zu entscheiden, das WIP-Limit zu erhöhen (was jedoch zu „Halden“ und einem neuen Stau bei einer der folgenden Spalten führen kann). Das wiederholte Anpassen führt mit der Zeit zu einer Optimierung des Arbeitsflusses.

„KANBAN“ (in Grussbuchstaben) zur evolutionären Prozessverbesserung:

KANBAN anerkennt vorbehaltlos die aktuell praktizierten Vorgehensweisen, Rollen und weitere Organisationsaspekte, macht sie aber sicht- und messbar und schafft damit objektive Voraussetzungen für schrittweise Verbesserungen auf Basis von 3 Grundprinzipien und 5 Kerneigenschaften:

3 Grundprinzipien:

- Beginne dort, wo du jetzt bist
- Respektiere bestehende Prozesse, Rollen, Verantwortlichkeiten
- Vereinbare mit den anderen ein fortlaufendes (nie abgeschlossenes), inkrementelles, evolutionäres Veränderungsvorgehen

5 Kerneigenschaften:

- Fluss der Arbeit („workflow“) visualisieren
- Work In Progress begrenzen
- Arbeitsfluss messen und kontrollieren
- Regeln des Prozesses explizit machen
- Modelle als Mittel der Prozessverbesserung nutzen (ToC, System Thinking, Deming, Verschwendung reduzieren, etc)

Daraus resultierende Prozessverbesserungen sind z.B.

- Unterschiedliche und für jedes Projekt / Team / Produkt optimale spezifische Prozesse.
- An den Arbeitsschritten können ein Team als Ganzes oder einzelne Spezialisten oder - auch verteilte - Teams kombiniert mit Einzelpersonen transparent arbeiten.
- Sowohl ein kontinuierlicher Arbeitsfluss als auch ein natürlicher Arbeitsrhythmus ohne „künstlich“ aufgeprägte Timeboxes, Sprints oder Iterationen sind möglich.
- Die Reihenfolge der Arbeitspaket-Erledigung ist kurzfristig festlegbar

- Die ausnahmsweise Verletzung der WIP-Limiten (z.B. bei Notsituationen) ist als regulärer Prozess möglich.
- „Swimmlanes“ für verschiedene Serviceklassen erlauben spezifische Kapazitätszuordnungen und Durchlaufgeschwindigkeiten oder auch termingebundene Arbeiten.
- Methoden der quantitativen statistischen Prozesskontrolle (Durchlaufzeiten, Verweilzeiten etc) zur Prozessoptimierung sind anwendbar und erlauben ein gezieltes Controlling.
- Experimente zur Prozessoptimierung sind erlaubt.

Das „Spezielle“ in Hinblick auf den Umgang mit Unberechenbarkeiten und Komplexität

„Kanban“ (in Kleinbuchstaben) zur operativen dezentralen (Selbst-)Steuerung der Arbeit ist extrem flexibel und rasch an spezielle, auch nicht vorausgesehene, Situationen anpassbar. Die Festlegung der Spalten für die Arbeitsschritte und deren WIP-Limits kann im Sinn der „empirischen Prozesskontrolle“ rasch und auch experimentell verändert werden.

Die Häufigkeit der „Nachschubmeetings“ zum Auffüllen der Spalte „anstehende Arbeiten“ ist beliebig festlegbar. Es kann auch mehrmals pro Tag stattfinden. Abarbeitungsreihenfolgen können daher sehr kurzfristig geändert werden.

All das, was planbar ist oder geplant werden muss kann mit speziellen „Swimmlanes“ gesteuert und vom nicht planbaren oder nicht genau zu planenden getrennt werden.

Unplanbare Notsituationen können im Prozess geordnet berücksichtigt werden.

Die Kanban-Tafel zeigt jederzeit den aktuellen Arbeitsstand, Unterlast und Engstellen im Arbeitsfluss und erlaubt sofortige Interventionen.

„KANBAN“ (in Grussbuchstaben) zur evolutionären Prozessverbesserung der Arbeit macht die aktuell praktizierten Arbeitsweisen sicht- und messbar (quantifiziert und statistisch auswertbar) und schafft damit objektive Voraussetzungen für schrittweise, auch experimentelle, Verbesserungen.

Konsequenzen beim Praktizieren innerhalb „üblicher“ Organisationen und Skalierbarkeit / Übertragbarkeit auf ganze Organisationen

„Kanban“ (in Kleinbuchstaben) zur operativen dezentralen Steuerung auf Basis des „Pull-Prinzips“ kann in Organisationen mit einem ausgeprägten top-down Steuerungs- und Kontrollbedürfnis auch auf Detailebene 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-Prinzip termingerechten „just in time“ - Lieferungen vermag diese Skepsis ab- und Vertrauen aufzubauen. Für Manager können die unzähligen und bereits jahrzehntelangen positiven Erfahrungen mit solchen Systemen in der Produktion und Logistik vertrauensbildend sein.

Es kann zur operativen dezentralen Steuerung auf Basis des „Pull-Prinzips“ unter Beachtung von „WIP-Limiten“ für alle Arten von - in Schritten zerlegbaren - Arbeitsflüssen auf allen Stufen einer Organisation und in allen Bereichen praktiziert werden.

So etwa wird es im Rahmen des „Scaled Agile Framework for Enterprise“ (SAFe - siehe folgend) als Steuerungsmethode im strategischen Portfoliomangement vorgeschlagen.

„KANBAN“ (in Grussbuchstaben) zur evolutionären Prozessverbesserung 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 ist. Das reduziert den Widerstand gegen Veränderungen erheblich.

Andererseits macht es das Existierende und Etablierte sehr transparent - und auch messbar. Bisher unentdeckte (und ev. bewusst verschleierte) Schwachstellen und „Verschwendungen“ werden damit sichtbar. Das kann zu erheblichen Irritationen führen.

Es kann ebenfalls für alle Arten von - in Schritten zerlegbaren - Arbeitsflüssen auf allen Stufen einer Organisation und in allen Bereichen praktiziert werden.

Scaled Agile Framework for Enterprise™ (SAFe)

Charakteristik und Herkunft

Die Grundprinzipien agiler Vorgehensweisen beziehen sich in der Regel entweder auf nur ein Projekt oder - wie bei Scrum - auf ein Team, das ein Produkt entwickelt. Und es wird stillschweigend eine relativ homogene technologische Landschaft angenommen, in der einzelne Produktinkremente nach jeweils einer Iteration von 2 bis 4 Wochen in Produktion gehen und - zumindest potentiell - genutzt werden können.

Die Skalierbarkeit von agilen Vorgehensweisen auf Multi-Team- bzw. Multi-Produkt-Situationen bezogen auf den gesamten Lebenszyklus der Produkte in großen Unternehmen mit einer hohen technologischen Vielfalt wird bei diesen Vorgehensweisen kaum thematisiert.

Das agile Portfolio- und Programm-Management in großen Unternehmen, die Koordination Dutzender von Teams und technologische Aspekte wie die teamübergreifende Continuous Integration bleiben offen und führen zu erheblichen Problemen.

Vor diesem Hintergrund hat Dean Leffingwell - ein in seinen Rolle als Top-Manager grosser IT-Firmen davon selbst Betroffener - nach Lösungen gesucht und sie in seinem Büchern beschrieben.

[Dean Leffingwell: Scaling Software Agility: Best Practices for Large Enterprises. Addison-Wesley Longman, 2007] und [Dean Leffingwell: Agile Software Requirements: Lean Requirements for Teams Programs and the Enterprise. Addison Wesley, 2010].

Das Ergebnis nennt er „Scaled Agile Framework for Enterprise (SAFe)“. Das SAFe hat sich auf Basis von Einsätzen in Programmen mit nur 50-100 Beteiligten aber auch in grossen Unternehmen mit mehreren Tausend Softwareentwicklern entwickelt. Seine jeweils aktuellste Version ist als umfassende Dokumentation in <http://scaledagileframework.com/> frei verfügbar.

SAFe offeriert umfassende Beschreibungen der individuellen Rollen, Teams, Tätigkeiten und Artefakte, die zur Skalierung der Agilität von der Team- zur Unternehmensebene nötig sind. Zentrale Aspekte, gemäss Leffingwell, sind:

- *Scaling Value: Not everything is a User Story*
- *Scaling Team and Timebox: No team is an island*
- *Scaling Design: Emergent design meets intentional architecture*
- *Scaling Portfolio Management: Addressing legacy mindsets*
- *Scaling Leadership: Your enterprise can be no leaner than your executives thinking*

Das SAFe versteht sich im Gegensatz zu einem Projektmanagement-Framework als agiles Framework zur kontinuierlichen Lieferung produktiv nutzbarer Lösungen in „Releases“ im Rahmen der Neu- und Weiterentwicklung und der Wartung von Produkten längs ihrer gesamten Lebenszeit.

SAFe bietet ein Angebot an guten Praktiken, die, obwohl aufeinander abgestimmt, auch nur teilweise genutzt werden können. SAFe verlangt nicht, dass es vollumfänglich zu verwenden ist.

Vorgehen und Rollen

[vgl.: Dean Leffingwell: Agil bis zur Unternehmensstrategie - ein „Big Picture“. In: Hans-Peter Korn (Hsg), Jean Pierre Berchez (Hsg.): Agiles IT-Management in großen Unternehmen. Symposion Publishing 2013

Weitere Details: <http://scaledagileframework.com/>]

Das SAFe beschreibt Rollen, Artefakte und Vorgehensweisen insbesondere des Anforderungsmanagements auf Team-, Programm- und Portfolioebene.

Die Teamebene

Auf der Eben der Agilen Teams kann Scrum wie im Scrum Guide beschrieben oder ein an Scrum angelehntes oder ein anderes agiles Vorgehen praktiziert werden.

Agile Teams mit 7 ± 2 Mitgliedern definieren, bauen und testen *Stories* in eine Abfolge von *Iterationen* und *Releases*. („Stories“ sind im SAFe kleine, eigenständige Funktionalitäten, die zur Gänze in einem Produktinkrement geliefert werden und für den Nutzer in irgend einer Form wertvoll sind.)

In sehr kleinen Unternehmen gibt es allenfalls nur wenige solcher Teams. In großen Unternehmen hingegen arbeiten jeweils mehrere agile Teams „gemeinsam in einem Boot sitzend“ an der Realisierung umfassender Funktionalitäten für komplette Produkte, Features, Komponenten, Subsysteme etc. Der jeweilige *Product Owner* dieser Teams ist verantwortlich für das Managen des *Team-Backlogs* bestehend aus Stories, die das Team zu erledigen hat.

Die Programmebene

Auf der Programmebene erfolgt die teamübergreifende Koordination der Funktionalitäten umfangreicher Systeme auf Basis eines synchronisierten *Agile Release Train (ART)*. Ergebnisse des ART sind *Potentially Shippable Increments (PSIs)* zu terminfixierten *Releases* alle 60 bis 120 Tage. Diese überprüfbaren Inkremente können (müssen aber nicht) dem Kunden ausgeliefert werden. Die effektive Auslieferung ist abhängig einerseits davon, ob der Kunde die Kapazität hat, das neue Produkt entgegenzunehmen, und andererseits von externen Ereignissen, die das Timing bestimmen. Der Inhalt jedes Releases eines Agile Release Trains wird zusammen mit allen Mitgliedern (nicht nur einiger Vertreter) aller dazu beitragenden Teams in einem 2-tägigen *Release-Planungsworkshop* ausgehend vom „*Program Backlog*“ erarbeitet. SAFe benutzt die allgemein gebräuchliche Bezeichnung *Product Manager* für jene Personen, die auf dieser Ebene die Features des Systems im Program Backlog festlegen, wobei es auch viele andere Bezeichnungen für diese Rolle geben kann. Die Unterscheidung zwischen dem Product Owner auf Teamebene und dem Product Manager auf Programmebene entschärft die in grossen Unternehmen möglichen Rollenkonflikte zwischen dem Product Owner i.S. von Scrum und den anderen Rollenträgern im Bereich der Produktgestaltung.

Auf der Programmebene wird in Kooperation der einzelnen Teams auch die übergreifende Architektur der Produkte und der Produktbasis vorausschauend vereinbart und weiterentwickelt.

Weitere Rollen auf der Programm-Ebene sind das *Release-Management*, das *System Team* und diverse den Teams bei Bedarf verfügbare Spezialisten, die als „*shared resources*“ zur Verfügung stehen.

Die Portfolioebene

Unter „Portfolio“ versteht SAFe die Gesamtheit jener *Investment Themes*, die die Investitionsprioritäten auf Unternehmensebene steuern. Das *Portfoliomanagement* umfasst jene Personen, Teams und Organisationsteile, die sich darauf konzentrieren, die Investitionen des Unternehmens mit seiner Geschäftsstrategie in Einklang zu bringen. Die Investitionsthemen sind Treiber der *Portfolio Vision*, dargestellt als eine Zusammenstellung umfangreicherer Vorhaben in Form von *Business- und Architecture-Epics*. Deren Erarbeitung wird mit jeweils einem Kanban-System so gesteuert, dass die Arbeit laufend priorisiert und deren Umfang begrenzt wird. Die Epics bilden den *Portfolio Backlog* und fließen im Lauf der Zeit in verschiedene *Agile Release Trains* und deren Program Backlogs ein. Nebst diesen agil vorangetriebenen Programmen sind aber auch solche mit einem sequentiellen „Stage - Gate“ - Vorgehen („Wasserfallvorgehen“) weiterhin möglich. [siehe Abb. 2 in <http://scaledagileframework.com/program-portfolio-management/>]

Das „Spezielle“ in Hinblick auf den Umgang mit Unberechenbarkeiten und Komplexität

SAFe nutzt auf Teamebene alle iterativen, inkrementellen und adaptiven Elemente insbesondere von Scrum, so auch kurze Iterationen, umrahmt sie aber mit langsamer getakteten „Releases“ als Lieferzeitpunkte für die eigentlichen „Potential Shipable Increments“. Das erleichtert umfangreiche Integrationsarbeiten einerseits und trägt der in der Regel auf Nutzerseite begrenzten Fähigkeit (und Bereitschaft) der Entgegennahme neuer Produktinkremente in sehr kurzen Abständen Rechnung.

Auf Programm- und Portfolioebene wird ein rollender „JEDUF“ praktiziert.

Im Rahmen eines ART wird der Umfang nur des jeweils nächsten Releases verbindlich festgelegt. Für die späteren Releases enthält die „Roadmap“ nur grobe und unverbindliche Prognosen. (Kein eisernes Dreieck)

Auf Portfolioebene gibt es statt der festen Jahresplanung eine rollende Aktualisierung von Release zu Release.

Die mit Kanban erreichte Mengenbegrenzung der auf Portfolioebene gleichzeitig bearbeiteten Epics erhöht die Flexibilität.

Konsequenzen beim Praktizieren innerhalb „üblicher“ Organisationen

SAFe orientiert sich an den üblichen Planungs- und Aktionsebenen grösserer Unternehmen und schafft dort eine deutlich verbesserte Transparenz und Traceability. Allerdings wird die „Mikropolitik“ damit erschwert bzw. offensichtlicher.

Schwierig ist sicher der Wechsel von den etablierten und starren Jahresplanungen und der Abschied von verbindlichen Mehrjahresprogrammen.

SAFe sieht die Manager aller Ebenen als „Botschafter“ des „Lean Management“, welche die Basis des „House of Lean“ darstellen. [vgl Dean Leffingwell: Thoughts on Lean Thinking. Blog vom 15. Sept. 2009 in <http://scalingsoftwareagilityblog.com/thoughts-on-lean-thinking/>]

Skalierbarkeit / Übertragbarkeit auf ganze Organisationen

Der Kerngedanke von SAFe ist es, eine agile Produktentwicklung beginnend auf der Portfolioebene auch sehr grosser Unternehmen zu ermöglichen. Andere Funktionsbereiche neben der Produktentwicklung betrachtet das SAFe nicht.

Was kennzeichnet, was leistet eine als Ganzes „agile“ Organisation?

Das kann nur beantwortet werden, wenn zunächst das geklärt ist:

Welchen Begriff von „agil“ legen wir zu Grunde?

Diese Alternativen - siehe den Abschnitt *Die Vieldeutigkeit des Begriffes „agil“* zu Beginn - stehen zur Auswahl:

- Agil = Flexibilität und Adaptivität
- Agil = Flexibilität und Adaptivität plus „Lean Management Principles“
- Agil = Flexibilität und Adaptivität plus „Lean Management Principles“ verbunden mit einer Sammlung „post-tayloristischer“, hierarchie- und autoritätsfreier, partizipativer, selbststeuernd-kollaborativer, kommunikationsbasierter, systemischer Praktiken und „Glaubenssätze“ (dem entspricht u.a. das "Agile Manifest")
- Agil = Aspekte zur Sicherung der Überlebens- und Entwicklungsfähigkeit (Robustheit - Belastbarkeit - Reaktionsfähigkeit - Flexibilität - Innovationsfähigkeit - Anpassungsfähigkeit)

Wenn wir eine Bedeutung wählen, welche in *jedem* Fall Flexibilität und Adaptivität verlangt und damit die Abkehr vom umfassenden Abklärungen, Entwürfen und Planungen vor der Realisierung, also in jedem Fall das „Just Enough Design Up Front“(JEDUP) dem „Big Design Up Front“ (BDUF) vorzieht und damit alle Vorhaben als grundsätzlich mittelfristig unkalkulierbar und unplanbar betrachtet müssen wir als Konsequenz stets ein sich vorwärts tastendes Vorgehen in möglichst kleinen Realisierungsschritten und einem Höchstmass an Rückkopplung wählen. Und damit auf eine Vielzahl vertrauter Planungs- und Kontrollinstrumente verzichten und uns andere noch ungewohnte Methoden aneignen.

Genau das bildet den Hintergrund des (vorhin beschriebenen) „radikalen“ eXtreme Programmings und von Scrum.

Wenn wir hingegen die Angemessenheit und Machbarkeit des „Just Enough Design Up Front“(JEDUP) und des „Big Design Up Front“ (BDUF) nicht als binäre Entscheidung sondern als die Eckwerte eines Kontinuums sehen, in dem es alle Arten von Mischungen beider Ausprägungen gibt, dann sollten wir „agil“ allgemeiner als Sicherung der Überlebens- und Entwicklungsfähigkeit (Robustheit - Belastbarkeit - Reaktionsfähigkeit - Flexibilität - Innovationsfähigkeit -

Anpassungsfähigkeit) verstehen. *Flexibilität* wäre dann die Fähigkeit, mehrere Lösungsmöglichkeiten einzusetzen und nahtlos von einer zur anderen überzugehen. Eine der Lösungsmöglichkeiten kann immer wieder mal auch ein sequentielles „Stage - Gate“ - Vorgehen sein.

Im weiteren Verlauf dieses Artikels beziehen wir uns auf diesen allgemeineren Begriff von „agil“:

- *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.

Was ist dann aus der Sicht der Kunden und Mitarbeitenden deutlich anders?

- Die Organisationsstruktur wird nicht nur durch stabile Hierarchien bestimmt sondern passt sich dynamisch an die aktuellen und voraussehbaren Wertschöpfungsprozesse an.
- Die Organisation versteht sich als Produkt fortwährender Kommunikation, nicht als Funktions- und Kompetenzraster.
- Mit den üblichen Mitteln der Planung und Kontrolle wird je nach Aufgabenbereich das effektiv planbare und kontrollierbare geplant und kontrolliert und nicht mehr - aber auch nicht weniger.
- Der primäre Fokus liegt auf der kontinuierlichen und nachhaltigen Unterstützung ganzer Produktlebenszyklen.
- Projekte gibt es nur dort, wo es sich effektiv um Projekte handelt (Umfang / Termin / Kosten sind planbar). Es gibt keine Grossprojekte sondern mehr oder weniger ergebnisoffene Programme. Sie bestehen aus einzelnen „echten“ Projekten einerseits und aus nicht projektmässig planbaren „forschungsartigen“ Entwicklungsvorhaben andererseits.
- Light-Tight Control Model: Auf Portfolioebene gibt es für die aktuell auf Programm- und Teamebene bearbeiteten Initiativen klare Entscheide und Vorgaben. Sie bilden den zuverlässigen Rahmen für die auf Teamebene realisierten flexiblen, inkrementellen und adaptiven Ausgestaltungen. Die Steuerung auf Portfolioebene bezieht sich auf diesen Rahmen, nicht auf weitere Details. [vgl. Brian Wernham: Agile Project Management for Government. Maitland and Strong, 2012. S. 225]

Wie kann eine Organisation als Ganzes in diesem Sinn „agil“ werden?

Wenn wir „Organisation“ nicht als formales Regelwerk eines arbeitsteiligen, auf Spezialisierung ausgerichteten Systems verstehen, also nicht als Funktions- und Kompetenzraster, sondern als ein durch Kommunikation gebildetes soziales System, dann kann diese Frage so umformuliert werden: „Mit welcher Kommunikation wird die von ihr gebildete Organisation agil?“

Sonja Radatz postuliert im „Relationsmodell“, dass jeder „Kreis“ innerhalb einer Organisation, der weitere Kreise umfasst und jeder von ihnen wieder weitere Kreise (z.B. Bereich - Abteilungen - Gruppen) sich mit diesen Komponenten darstellen lässt:

[Sonja Radatz: Veränderung verändern - Das Relationale Veränderungsmanagement - Die zukunftsweisende 4. Schule des Veränderungsmanagements. Verlag Systemisches Management, 2009]

- Selbstbeschreibung
- Identität
- Ziele und Strategien
- Prozesse / Handlungsmuster
- Kommunikationsstrukturen
- Operative Handlungen

Wenn also Kommunikation soziale Systeme bildet, dann wird sie zur Bildung eines Organisationskreises diese sechs Komponenten thematisieren. Wenn ein Organisationskreis „agil“ werden soll, dann wird sich die Kommunikation damit beschäftigen, was denn „agil“ für diese sechs Komponenten bedeutet, wie sie sich damit verändern. Diese Kommunikation verläuft jedoch nicht völlig frei sondern innerhalb eines gegebenen thematischen Rahmens. Diesen Rahmen zu setzen ist die Pflicht des „leitenden“ Mitglieds des Organisationskreises. Dieser Rahmen schafft den Wirkungs- und Kommunikationsraum für die nicht leitenden Mitglieder.

Rahmensetzung als Führungspflicht

Wie beginnt in einem Organisationskreis die Kommunikation darüber, was denn „agil werden“ bedeutet?

Einmal dann, wenn „von oben“ etwas wie: „Unser Unternehmen muss agiler werden. Deshalb haben wir uns entschlossen zwei kleinere Projekte versuchsweise mit Scrum abzuwickeln.“ Oder dann, wenn jemand im Organisationskreis erzählt: „Ich habe an einem Vortrag gehört, dass „agile Entwicklung“ eine tolle Sache ist. Das wäre etwas für uns!“

Folge davon ist ein eher abstrakter Diskurs darüber, was „agil“ ist und warum es passt und nicht passt. Es wird also recht hypothetisch über eine im ersten Fall aufgezwungene und im zweiten Fall faszinierend erscheinende Lösung diskutiert - und nicht darüber, welche der aktuellen Herausforderungen denn anders als bisher bearbeitet werden sollten.

Eine pragmatischere Kommunikation entsteht dann, wenn die Führungskraft des Kreises bewusst macht, was die konkreten Rahmenbedingungen des Organisationskreises sind. Das können Vorgaben sein wie:

- *„Ab nächsten Monat stehen max. 40% der Kapazität für längerfristig planbare Arbeiten zur Verfügung“. Der Rest muss kurzfristig flexibel einsetzbar sein.*
- *„Das Produkt FOD wird in sechs Monaten nicht mehr weiterentwickelt und wechselt in den reinen Unterhaltsstatus. Bis dann ist es in Hinblick auf diesen Status zu stabilisieren.“*
- *„Für KGH-7-RS gibt es Ende Juni einen dem Kunden demonstrierbaren Prototypen zum Nachweis der technischen, insbesondere thermischen, Machbarkeit. Vorversionen davon kann der Kunde jeweils alle sechs Wochen überprüfen.“*
- *„Anhand der Entwicklung von KGH-7-RS wird parallel zum derzeit genutzten Model-Kit-85 das Konkurrenzsystem mOdelCenter bis Ende Mai getestet, aber nicht produktiv genutzt. Alle zwei Wochen wird dem ganzen Team demonstriert, wie dieselben Aufgaben mit Model-Kit-85 einerseits und mOdelCenter andererseits gelöst werden.“*
- *„Eine Entwicklungsaufgabe ist erst dann bereit zur Abnahme, wenn ein Review mit einem anderen, daran unbeteiligten, Teammitglied stattgefunden hat“*

Der Rahmen lautet also nicht „wir werden agil“ sondern legt möglichst konkret das fest, was der Organisationskreis zu leisten hat und gibt die „Erlaubnis“ zu flexiblen Arbeitsweisen.

Bei der Vorgabe des Rahmens sollte auf „modische“ und breit auslegungsfähige Begriffe und Einengungen der möglichen Lösungen verzichtet werden. Worte wie „agil“ oder „Scrum“ oder „DSDM“ oder „Sprint“ oder „Kanban“ oder „Product Owner“ oder „Shipable Product Increment“ sollten besser nicht vorkommen.

„Wir werden agil“ und das, was „agil“ in dem Zusammenhang überhaupt bedeutet, ergibt sich dann als Lösungsmöglichkeit, wie mit diesem Rahmen umgegangen werden kann. Diese Lösungsmöglichkeit zu entwickeln erfordert Kommunikation zu den o.e. sechs Komponenten. Die Kommunikation dient dazu, den Rahmen zu klären, ein neues Selbstverständnis des Organisationskreises zu entwickeln und neue „Wegweiser“ für das tägliche Handeln aufzustellen. Diese Kommunikation darf sich jedoch nicht über Wochen und Monate neben der ja alles dominierenden laufenden Arbeit hinziehen sondern erfordert die aktive Beteiligung *aller* Personen des Organisationskreises in einem kompakten „Change Event“ (ein halber bis ein Tag). Unmittelbar danach, also ab dem ersten Tag nach dem Change Event - und nicht erst in vielen über etliche Wochen verteilten Einzelmaßnahmen - agieren alle so, wie es im Change Event erarbeitet wurde.

Change Event

Der Ablauf des Change Events orientiert sich an den von Sonja Radatz beschriebenen Beispielen in Kapitel 10 in [Sonja Radatz: Veränderung verändern - Das Relationale Veränderungsmanagement - Die zukunftsweisende 4. Schule des Veränderungsmanagements. Verlag Systemisches Management, 2009]

Der neue Rahmen des Organisationskreises

Der Event beginnt mit einer kurzen Reflexion der Erwartungen und Ziele der Mitglieder des Organisationskreises:

- Was hat uns heute hierher geführt?
- Wohin geht unsere gemeinsame Reise?
- Warum lohnt es sich, dabei zu sein?
- Was muss heute besprochen werden, damit wir auch in den kommenden Jahren ausgezeichnete Erfolge erzielen?

Danach stellt die Führungsperson des Kreises den - wie oben erläuterten - Rahmen vor:

- Neue Aufgaben und Herausforderungen
- Das bleibt unverändert
- Zukünftige Trends der quantitativen Vorgaben

Wie verstehen wir / was bedeutet für unseren Organisationskreis dieser Rahmen?

In selbstmoderierten Maxi-Mix-Gruppen (jeweils 3 bis 4 Personen aus unterschiedlichen Aufgabenbereichen / mit unterschiedlichen Rollen) wird das besprochen:

- 1) Was haben wir Relevantes gehört?
- 2) Was bedeutet das Gehörte für uns?
 - Was vom Bestehenden & Funktionierenden passt gut dazu / unterstützt uns dabei
 - Was brauchen wir zusätzlich oder anders?
- 3) Welche Fragen haben wir noch?

Die Führungsperson beantwortet danach diese von den Maxi-Mix-Gruppen in 3) formulierten Fragen.

Achtung: Es geht hier *nicht* darum, den Rahmen in Frage zu stellen oder Änderungsvorschläge zu erarbeiten sondern darum, den Rahmen zu verstehen und Auswirkungen zu erkennen. Aus Sicht einer idealistischen Vorstellung von „egalitärer“ und „basisdemokratischer“ Führung, die eher einer Verneinung von Führung (auch seitens der zur Führung Verpflichteten selbst) gleicht, ist eine solche Aussage irritierend. Führung jedoch zu verstehen - und zu akzeptieren - als Setzen eines klaren und nicht verhandelbaren Rahmens (nicht aber als Beschreibung des „Bildes im Rahmen“) entspricht dem im Abschnitt *Was ist dann aus der Sicht der Kunden und Mitarbeitenden deutlich anders?* genannten „Light-Tight Control Model“. Sonja Radatz schreibt dazu:

„Ist es nicht unmenschlich, von Mitarbeitern zu verlangen, dass sie den Rahmen akzeptieren?“ Oder: „Was tun Sie, wenn die Mitarbeiter einfach nicht mitmachen?“

Meine Antwort ist einfach: Ich verlange nicht, dass jemand den Rahmen akzeptiert. Aber ich verlange, dass er sich für oder gegen ein Akzeptieren des Rahmens entscheidet, sonst wird er niemals darin erfolgreich und zufriedenstellend tätig sein. Und dann läuft die Zusammenarbeit nicht mehr.“

Die Selbstbeschreibung des Organisationskreises

Die selbstmoderierten Maxi-Mix-Gruppe erarbeiten nun die „Selbstbeschreibung“ des Organisationskreises:

- 1) Wie wollen wir uns auf Basis dieses Rahmens als Organisationskreis ab sofort selbst beschreiben?
- 2) Wie wollen wir von anderen beschrieben werden
 - a) von den Nutzern unserer Ergebnisse?
 - b) von Mitgliedern anderer Organisationskreise unseres Unternehmens?
 - c) von unternehmensexternen Entscheidungsträgern?
 - d) vom Wettbewerb?

Die Gruppen dokumentieren ihre Ergebnisse pro Punkt in ein bis zwei Sätzen auf je einem Flipchart. Danach bewerten alle Gruppen alle Statements. Daraus ergibt sich, hinter welchen Statements alle Beteiligten zu 100% stehen.

Die Identität des Organisationskreises

Welche Identität brauchen wir in Zukunft, um die Selbstbeschreibung durchgängig zu leben?

Alle Teilnehmenden (als Brainstorming) notieren pro Identitätsaspekt ihre Ideen auf Klebezettel:

- **Vision:** Was soll langfristig gelebt werden?
- **Mission:** Wofür sollen wir die Selbstbeschreibung leben?
- **Grundannahmen:** Von welchen Grundannahmen soll jeder ausgehen (dürfen/ müssen), wenn die Selbstbeschreibung durchgängig gelebt wird?
- **Leitlinien des Handelns:** Welche Leitlinien des Handelns müssen wir alle dem Tun als roter Faden zugrunde legen, um die Selbstbeschreibung zu leben?
- **Kernkompetenzen:** Welche Kernkompetenzen braucht unser Organisationskreis, um diese Selbstbeschreibung zu leben?
- **Geschichten:** Welche „Geschichten“ dürfen wir uns nicht mehr / welche müssen wir uns ab sofort durchgängig erzählen, um die Selbstbeschreibung zu leben?
- **Do's/Don'ts:** Was sind absolute Do's/Don'ts ab sofort und für jeden, wenn wir diese Selbstbeschreibung leben?
- **Werte:** Was muss jedem etwas wert sein, wenn wir die Selbstbeschreibung leben?

Danach gruppieren alle die Notizen zu „Faktoren“ und bewerten sie. Daraus ergibt sich hinter welchen dieser „Faktoren“ alle Beteiligten zu 100% stehen.

Der Wegweiser für den ab morgen eingeschlagenen Weg

Die Ziele unseres Organisationskreises

In selbstmoderierten Maxi-Mix-Gruppen werden Antworten dazu erarbeitet:

„Um diese Selbstbeschreibung und Identität zu leben: Welche bereichübergreifenden und konkreten Ziele ("Als kann ich / habe ich damit") verfolgen wir?"

Danach bewerten alle Gruppen alle Ziele. Daraus ergibt sich hinter welchen Zielen alle Beteiligten zu 100% stehen.

Unsere Strategie

Alle Teilnehmer einigen sich gemeinsam auf ein bis drei Sätze dazu:
„Diese Strategie liegt diesen Zielen zugrunde:“

Spezifische Umsetzungsmassnahmen:

Alle Teilnehmer organisieren sich in themen- oder aufgabenspezifische Kleingruppen. Jede Gruppe gestaltet ein attraktives Poster mit diesen Inhalten:

- Motto (Kurztext und/oder Symbol für alle Maßnahmen)
- Spezifische Umsetzungsmassnahmen ("Als kann ich / habe ich damit")
- Beteiligte und deren Beiträge
- Betroffene und deren Nutzen
- Rahmenbedingungen, Voraussetzungen

Die Gruppen stellen ihre Poster vor und erhalten Rückmeldungen dazu.

Danach:

- Überarbeitung der Poster in den Kleingruppen
- Planung möglichst konkreter erster Schritte („ab morgen machen wir“) zur Umsetzung dieser Massnahmen (was, wann, wer mit wem, wie) und deren zukünftiges "Controlling" (Rhythmus, durch wem / in welchem Rahmen)

Diese Planungen stellen die Gruppen einander vor und ziehen die Rückmeldungen danach in ihre Pläne ein.

Damit ist der Change Event abgeschlossen.

Vermissen Sie in diesem Ablauf Worte wie „agil“ oder „Scrum“ oder „Sprint“ oder „Kanban“ oder „Product Owner“ oder „Shipable Product Increment“?

Dass diese und ähnliche Begriffe im Layout des Change Events nicht vorkommen ist Absicht: Es liegt an den Beteiligten des Events selber zu entscheiden, bei welchen Aufgaben und Herausforderungen welche Vorgehensweisen angemessen sind.

Das ist Ausdruck einer „agilen Agilität“, also einer „Agilität 2. Ordnung“ im Gegensatz zu einer als Doktrin verstandenen Form der Agilität.

Handeln - Überprüfen - Anpassen

Bereits am ersten dem Change Event folgenden Tag beginnt die Umsetzung der im Event formulierten ersten Schritte und alle orientieren sich sofort an der vereinbarten Selbstbeschreibung, Identität, den Zielen und der Strategie. Die am Event dazu produzierten Artefakte (z.B. Flipcharts) sind an einem zentralen Ort (z.B. bei der Kaffecke) weiterhin sichtbar.

In den zu den konkreten Massnahmen vereinbarten Controllings wird regelmässig das damit Bewirkte überprüft. Darauf basierend werden weitere kurzfristig wirksame Schritte entwickeln und vereinbart.

Wenn irgend möglich sollte der aktuelle Arbeitsstand aller dieser Massnahmen und ihrer relevanten Teilschritte allen Mitgliedern der Organisationseinheit permanent sichtbar sein, z.B. angelehnt an simple Aufgabenlisten (noch zu tun / erledigt), Kanban oder Netzpläne oder Balkendiagramme.

Das hier beschriebene „Rahmensetzen - Change Event als Kommunikationsgefäss - Handeln/Überprüfen/Anpassen“ ist das Gestaltungsprinzip aller Organisationskreise auf allen Ebenen. Bei einem Organisationskreis, der aus weiteren Kreisen besteht, sind die in diesem Kreis erarbeiteten Ziele und Massnahmen die Rahmensetzungen für die darin enthaltenen weiteren Kreise.

Wie aber können derartige Organisationskreise eine insgesamt *dynamische* Organisation so bilden, dass sie der aktuellen und kurzfristig zu erwartenden Art der Wertschöpfungsprozesse, also dem eigentlichen Zweck der Organisation, entspricht und nicht erstarrten Gepflogenheiten?

Dynamische Organisation

„Dynamisch“ steht für beweglich, tatkräftig, energiegeladen im Gegensatz zu „statisch“. Bei sich ändernden Rahmenbedingungen und einem schwankenden Boden ist ein statisches Verhalten bestenfalls eine Option im Sinn von „Aussitzen und warten bis es wieder ruhiger wird“. Ruhiger wird es für Unternehmen heute jedoch immer seltener.

Dennoch: Nach wie vor sind Unternehmen in der Regel von einer, wie eingangs beschriebenen, stabilen und hierarchischen Organisationsstruktur geprägt. Und das, obwohl es seit weit mehr als zwei Jahrzehnten konkrete Beispiele für Alternativen gibt.

Organisation als dynamisches Netzwerk

So etwa die oft zitierte brasilianische Firma Semco mit ihrer dank des Geschäftsführers und Mehrheitseigners Ricardo Semler radikalen Demokratisierung [Ricardo Semler: Das Semco System - Management ohne Manager - Das neue revolutionäre Führungsmodell. Heyne, 1995 (nur noch antiquarisch erhältlich)] und [http://de.wikipedia.org/wiki/Das_Semco_System] und [<http://www.semco.com.br/en/mapa.asp>]. Kennzeichen sind Partizipation statt Hierarchien, Vertrauen statt Kontrolle, Mitbestimmung statt autoritärer Führung und Entbürokratisierung. Die Rolle als Führungskraft übernimmt eine Person aufgrund der Achtung der Geführten und es existiert kein Organigramm des Unternehmens. Statt der vorherigen 12 Hierarchieebenen stellte Semler (als Sohn des Firmengründers) in den 1980er-Jahren die Organisation auf hierarchiefreie konzentrische Kreise um. An die Stelle der hierarchischen Steuerung und Kontrolle tritt sie Selbstorganisation der Gruppen.

Helmut Reinhardt [http://wiki.nuevalandia.net/Selbstorganisation_bei_SEMCO] beschreibt das so: „Wichtig ist, dass das Ziel der Gruppe erreicht wird. Hat irgendein Mensch des Unternehmens eine Idee oder ein Konzept, so lädt er über das Intranet der Firma per Rundmail ausnahmslos alle anderen Menschen zu einem Treffen ein. Kommen kann, wer will und interessiert ist. ... Bildet sich eine Gruppe für das Projekt, wird es in Angriff genommen und die Aufgaben werden entsprechend den Wünschen der teilnehmenden Menschen verteilt. Entscheidungen werden immer im Teamprozess getroffen. Vorgesetzte Gruppenleitersmenschen und Entscheidungsträgermenschen, die auch bei Semco nötig sind, werden von den Gruppenmitgliedermenschen selbst gewählt und regelmäßig (alle sechs Monate) bewertet. Erfüllt der Gruppenleitersmensch seine Aufgaben nicht zur Zufriedenheit der anderen, wird ihm das offen mitgeteilt und in der Regel gibt dieser die Gruppenleitung dann selbst wieder ab. ... Die Aufgabenteilung wird innerhalb der Teams abgesprochen. ... Wichtig ist nicht, ob jemand 8-12 Stunden täglich in der Firma arbeitet und dort anwesend ist. Wichtig ist einzig und allein, dass das Ziel der Gruppe erreicht wird, ohne dass einem Gruppenmitglied vorgegeben wird, wieviel Zeit es dafür aufbringen muss.“

Oder als flexibles Netzwerk kooperierende kaum hierarchisch strukturierte kleine und mittlere Unternehmen (oft „start up“-Firmen). Oder „heterarchische“ Organisationen insbesondere in wissensintensiven Bereichen mit hohem Innovationsdruck, wo eine die Balance zwischen schöpferischer Zerstörung und Kreation unterstützende Organisation erforderlich ist. [Merkus Reihlen, Annette Rohden: Das heterarchische Unternehmen - Ein flexibles Organisationsmodell. In: Sonja Radatz (Hsg.): Evolutionäres Management - Antworten auf die Management- und Führungsherausforderungen im 21. Jahrhundert. Verlag Systemisches Management; 2003. Seite 78 - 91]

Bedeutet „dynamische Organisation“ also zwangsläufig den Wechsel von einer hierarchischen Struktur zu einem Netzwerk mit sich je nach Aufgabe immer wieder selbständig zu Kooperationen zusammenfindenden Personen und Teams? Oder bieten vorgezeichnete - durchaus auch hierarchische - Strukturen nicht doch auch eine für die meisten von uns wichtige Orientierung, Vorhersehbarkeit und Stabilität? Schafft das nicht auch „Vertrauen“ als überlebenswichtige Reduktion einer uns sonst überfordernden Komplexität?

Welche Möglichkeiten also gibt es, hierarchische Strukturen dynamischer zu gestalten? Wie können innerhalb solcher Strukturen vertrauensbasierte Autonomie, Partizipation und Beweglichkeit gelebt werden? Hier Optionen dazu:

Soziokratie und Holacracy™

Auguste Comte, (Schöpfer der Begriffs „Soziologie“) prägte 1851 das Wort „sociocratie“ [vgl. <http://de.wikipedia.org/wiki/Soziokratie>]. Der US-amerikanischen Soziologe Lester Frank Ward benutzte den Begriff in einem 1881 verfassten Artikel. Cornelis „Kees“ Boeke, ein niederländischer Reformpädagoge, entwickelte das Konzept der Soziokratie Mitte des 20. Jahrhunderts weiter. Es wird auch heute noch in den „Iederwijs-Schulen“ verwendet. [<http://de.wikipedia.org/wiki/Iederwijs-Schule>]. Gerard Endenburg, ein Schüler Boekes, adaptierte dessen Ideen 1970 als Organisationsmethode für das von seinen Eltern übernommene elektrotechnische Unternehmen. Diese Adaption nannte er „Sociocratische Kringorganisatiemethode“ (Soziokratische Kreisorganisationsmethode). 1976 half ihm das, eine Unternehmenskrise zu überwinden. Damit fand die „Soziokratie“ auch weltweit Beachtung. Weiter gepflegt und verbreitet wird die Soziokratie heute vom Sociocratic Center [<http://www.sociocratie.nl/global/>], mit einer Organisation auch in Deutschland [<http://www.soziokratie.com/>]

Im Jahr 2001 gründete Brian J. Robertson zusammen mit zwei Kollegen die rasch wachsende Startup-Firma Ternary Software, Inc. und leitete sie sieben Jahre als CEO. Beim Aufbau dieser Firma entwickelte er das Organisationsprinzip „Holacracy™“ dem er sich heute vollamtlich widmet. Er ist Mitbegründer von „HolacracyOne“ [<http://holacracy.org/>], einer Organisation mit dem Ziel, diese Methode weiterzuentwickeln, kommerziell zu verbreiten und zu schulen und Firmen bei der Einführung von Holacracy zu beraten. Von HolacracyOne lizenzierte Organisationen bieten Services auf Basis von Holacracy an. Viele Elemente der Holacracy sind ähnlich jenen der - frei verfügbaren und älteren - „Soziokratie“. Die von aussen betrachteten nicht immer offensichtlichen Unterschiede zwischen Soziokratie und Holacracy beleuchten Bernard Marie Chiquet (Holacracy) und Christian Rüter (Soziokratie) in einem gemeinsamen Dokument: [HolacracyOne Community of Practice Call - Sociocracy - Holacracy™ - December 7th, 2011. Online in <http://tinyurl.com/bttu6bf>]

Eines der Anwendungsbeispiele für die Holacracy ist Adscale Laboratories Ltd [<http://youtu.be/ReXNOoDbdik>]

Die vier Grundprinzipien der Soziokratie, die auch dem Kern der Holacracy entsprechen, sind:

1. Eine Organisation besteht aus „Kreisen“, die nebengeordnet und unabhängig sind oder aus Kreisen mit ihnen hierarchisch übergeordneten Kreisen. Jeder Kreis trifft innerhalb des definierten Rahmens (er wird vom übergeordneten Kreis festgelegt) autonome Entscheidungen.
2. Zwischen den hierarchisch verbundenen Kreisen gibt es eine doppelte Verknüpfung: In jedem Kreis gibt es einen auch dem übergeordneten Kreis angehörender Kreis-Leiter, der die Beachtung des für den Kreis übergeordnet festgelegten Rahmens sicherstellt, sowie (mindestens) einen Delegierten des Kreises in den übergeordneten Kreis.
3. Beschlüsse beruhen auf „Konsent“.
4. Die Kreise wählen die Rollenträger im Konsent nach einer offenen Diskussion.

Unter „Konsent“ im Gegensatz zu „Konsens“ wird verstanden, dass statt der Frage: „Bist du dafür?“ die Frage „Hast du einen schwerwiegenden, auf die Sache bezogenen Einwand?“ gestellt wird. Ein Konsent liegt dann vor, wenn niemand einen schwerwiegenden Einwand hat. Erreicht wird er durch Kommunikation, nicht durch einen Mehrheitsentscheid oder dadurch, dass - wie beim Konsens - alle damit voll einverstanden sind oder - wie beim Kompromiss - eine Sache so lange und oft bis zur Profillosigkeit modifiziert wird, bis alle „ja“ dazu sagen können.

[vgl.: <http://soziokratie.blogspot.ch/2009/08/konsent.html>]

Hans-Peter Korn

Bei der Soziokratie werden beide Verknüpfungen (Kreis-Leiter und Kreis-Delegierter) vom Kreis selbst gewählt. Bei der Holacracy ist das nur beim Kreis-Delegierten („Up-Link“) der Fall. Der Kreis-Leiter („Down-Link“) wird vom übergeordneten Kreis bestimmt.

Innerhalb der Kreise erfolgt die autonome Steuerung im Rahmen von Meetings. Bei der Holacracy sind das:

- Regelmässiges Steuerungs-Meeting: Es dient der Optimierung der Vorgehensmethoden im Kreis. Nebst dem Mitgliedern des Kreises nehmen der Kreis-Leiter und alle Delegierten der Unter-Kreise daran teil.
- Wöchentliches Arbeits-Meeting: Stark strukturiertes und zeitlich begrenztes Forum zur Ermittlung des Arbeitsstands (auch anhand spezifischer Metriken), Synchronisation und Koordination und zum Erkennen kurzfristig nötiger Verbesserungsmaßnahmen.
- Tägliches 15-minütiges Stand-Up Meeting: Zu den aktuellen Arbeiten werden diese Informationen ausgetauscht: Das wurde seit dem gestrigen Meeting gemacht / das ist bis zum morgigen geplant / das ist dafür als wichtige Verbesserung oder noch nicht gegebene Voraussetzung nötig.

Überlegen Sie sich bitte jetzt für Ihre Organisation:

Was alles könnte sich dort verändern wenn nur diese drei Elemente eingeführt werden:

1. Die „Boards“ jeder Ebene (also etwa das Board aller Gruppenleiter unter Führung des Abteilungsleiters) sehen sich nicht primär als „Arena der Einzelinteressen“ sondern als „wertschaffende Kreise“ und praktizieren die o.e. drei Arten von Meetings.
2. Entscheidungen werden auf Basis des Konsent gefällt.
3. An allen drei Arten der Meetings der Boards nehmen nebst den Leitern der dort vertretenen Organisationseinheiten auch je eine von den untergeordneten Einheiten delegierte Person teil (also nicht nur die „von oben“ eingesetzten Gruppenleiter sondern auch ein weiterer Gruppenvertreter, der von der Gruppe gewählt wird)

Der erste Punkt ist noch einigermaßen realistisch. Und bereits damit könnte schon viel an Klarheit und Reaktionsvermögen gewonnen werden. Und die Gruppenleiter ändern damit ihre Optik: Sie sind nicht mehr nur die Verantwortlichen für ihre Gruppe, die sich einmal pro Woche treffen und jeweils die Interessen ihrer Gruppe „verteidigen“ sondern sie bilden gemeinsam mit dem Abteilungsleiter einen „Kreis“, der einem gemeinsamen unternehmerischen Zweck dient.

Der zweite Punkt ist - je nach bisheriger Entscheidungskultur - schon etwas schwieriger oder aber dem entsprechend, was de facto schon praktiziert wird, damit alle ohne blossen „Dienst nach Vorschrift“ in die selbe Richtung ziehen.

Der dritte Punkt jedoch kann missverstanden werden: Wie etwa werden dann in so einem Board sensitive Personaldiskussionen besprochen? Oder sehr vertrauliche firmenpolitische Weichenstellungen? Werden diese Diskussionen dann viel intensiver als bisher von einzelnen Keyplayern und Machträgern im versteckten Rahmen geführt? Nein, diese Gefahr besteht nicht. In den Meetings werden nämlich keine Personen oder vertrauliche Firmenstrategien u.ä. behandelt. Das geschieht weiterhin in der bisherigen Art. In den Meetings diskutiert wird über Prozesse und Vorgehensweisen, nicht über davon betroffene - insbesondere sensible - „Dinge“. Dazu schreibt Brian J. Robertson: „To use a metaphor, circle meetings *don't deal with instances of objects*, they define and refine the class structure and interfaces of the organization based on data from “running the program”.“
[Brian J. Robertson: Holacracy- A Complete System for Agile Organizational Governance and Steering. In AGILE PROJECT MANAGEMENT ADVISORY SERVICE, VOL. 7, NO. 7, Cutter Consortium 2006. Online in: <http://tinyurl.com/cetkdwj>]

Weitere Details zur Soziokratie finden sich etwa hier: [Christian Rütter: Soziokratie - Ein Organisationsmodell - Grundlagen, Methoden und Praxis. Seminarunterlage und Einführungstext, 2010. Online in: <http://soziokratie.org/wp-content/uploads/2011/06/soziokratie-skript2.7.pdf>] und zur Holacracy hier: [Brian J. Robertson: Organisation weiterentwickelt - Einführung in Holacracy™. Online in http://holacracy.org/sites/default/files/resources/organization_evolved_german.pdf] und

ausführlicher [Brian J. Robertson: Leading-Edge Organisation: Einführung in Holacracy™. Online in <http://holacracy.org/sites/default/files/resources/LeadingEdgeOrganisation-deutsch.pdf>]

Eine Darstellung des Zusammenhangs zwischen einer agilen Organisation und der Holacracy findet sich hier: [Brian J. Robertson: Holacracy- A Complete System for Agile Organizational Governance and Steering. In AGILE PROJECT MANAGEMENT ADVISORY SERVICE, VOL. 7, NO. 7, Cutter Consortium 2006. Online in: <http://tinyurl.com/cetkdwj>]

Soziokratie und Holacracy sind Möglichkeiten, eine auf stufenweise Verfeinerung bzw. Verdichtung beruhende Koordination und Steuerung (also eine hierarchische Struktur) mit einem hohen Mass an Autonomie und Partizipation zu verbinden. Ein erster Schritt in diese Richtung ist kein revolutionärer Umbau der bestehenden hierarchischen Organisation sondern ihre „Umdeutung“: Auf jeder Ebene werden „Der Leiter von XYZ und die ihm direkt Unterstellten“ als ein „Kreis“ gesehen, der die oben beschriebenen drei Arten von Meetings etabliert.

Als nächstes gilt es, als viel schwierigeren Schritt, die Zweckorientierung jedes dieser Kreise zu optimieren: Welchem für die Gesamtorganisation essentiellen Zweck dient dieser Kreis? Folge davon wird sein, dass sich Kreise erweitern, verändern, wegfallen oder neue Kreise entstehen. Natürlich ist all das verbunden mit den bekannten Ängsten, die mit der bisherigen Position verbundenen Anerkennung und Macht zu verlieren.

Die Umgestaltung der bisherigen Hierarchie zu ineinander geschachtelten Kreisen löst ein Problem jedoch nicht: Wie können verschiedene Koordinations- und Steuerungsansprüche gleichzeitig erfüllt werden? Wie also lösen wir das Problem der - oft mehr als nur 2-dimentionalen - Matrix?

Prinzipiell wäre das so möglich:

Angenommen, ein Team z.B. der Produktentwicklung bearbeitet einen Teil eines Gesamtproduktes. Dabei sind spezifische Anforderungen der Qualitätssicherung zu erfüllen, die Entwicklungsmethodik muss den Firmenstandards entsprechen und das Team muss im Rahmen der ihm zugesprochenen Ressourcen (Personal und Infrastruktur) arbeiten.

- Der Teamleiter als Verantwortlicher für die Ressourcen ist - zusammen mit einem weiteren Delegierten des Teams - Mitglied des übergeordneten Teamleiter-Kreises.
- Der Chefentwickler - zusammen mit einem weiteren Delegierten des Teams - nimmt am übergeordneten Kreis „Entwicklungsstandards“ teil.
- Der QS-Beauftragte im Team - zusammen mit einem weiteren Delegierten des Teams - nimmt am „Qualitätssicherungskreis“ statt.
- Der Produktverantwortliche im Team - zusammen mit einem weiteren Delegierten des Teams - nimmt am Kreis der „Produkt Owner“ teil, der Leiter dieses übergeordneten Kreises ist Mitglied des weiter übergeordneten „Produktgruppen-Kreises“.
- Und es kann sein, dass im Kreis der „Product Owner“ der Produktverantwortliche unseres Teams auch noch als weiterer Delegierter in den „Produktgruppen-Kreises“ gewählt wird.

Das kann rasch zu einer markanten Belastung durch diverse - unbestreitbar wichtige - Meetings führen. Diese Belastung gibt es heute bereits in genau solchen „mehrdimensionalen“ Situationen. Mit der Regel, dass jeder Kreis durch zwei Personen in den übergeordneten koordinierenden Kreisen vertreten ist, wird sie jedoch deutlich vergrößert.

Es wird also auch hier weiterhin nötig sein, zwischen einem und nur einem „primären“ übergeordneten Kreis und weiteren „sekundären“ zu unterscheiden. Und nur im primären Kreis sollte es eine Doppelvertretung geben. Diese eine Vertretung eines Kreises (z.B. der QS-Vetreter des Teams) im übergeordneten Kreis (z.B. der Qualitätssicherungskreis) muss dann jedoch vom Kreis selbst bestimmt werden und nicht „von oben“ definiert sein. Bereits das macht einen relevanten Unterschied gegenüber den heute verbreiteten Praktiken.

Das neue „Daily Business“

So also präsentiert sich von aussen sichtbar eine in diesem Sinn „agile“ Organisation:

- Sie besteht aus ineinander verschachtelten „Kreisen“
- Die Kreise der primären Abstimmung sind mittels „Doppelvertretung“ verbunden
- Für jeden Kreis gibt es einen von seinen übergeordneten Kreisen festgelegten Rahmen
- Jeder Kreis ist autonom innerhalb dieses Rahmens
- Jeder Kreis erarbeitet seine mittelfristige strategische Ausrichtung und seine Ziele und Massnahmen in einem mindestens halbjährlichen „Change Event“ und zwischendurch bei grossen Veränderungen. Dabei steht auch diese Frage im Zentrum: „Welchem für die Gesamtorganisation essentiellen Zweck dient unser Kreis, welchen Zwecken dienen unsere Unterkreise?“
- Jeder Kreis steuert seine Arbeit mit monatlichen Steuerungs-Meetings, wöchentlichen Arbeits-Meetings und täglichen Stand Ups.

Die „agile“ Organisation ist dann kein punktueller (und vorübergehender) „Hype“ nur in einigen Teilen der Organisation sondern unternehmensweites „daily business“. Damit das nicht nur eine formale Fassade bleibt ist ein dem „Lean Thinking“ [<http://scalingsoftwareagilityblog.com/thoughts-on-lean-thinking>] entsprechendes und insbesondere von den Personen in Leitungsrollen vorgelebtes Denken und Handeln erforderlich.