

„Agiles Führen: warum Tools nicht reichen“

Dipl.-Ing. Rüdiger Schönbohm (TYSCON), 2.19

Agiles Führen und Agiles Arbeiten gehören untrennbar zusammen. Mischformen von agilen und klassischen (hierarchischen) Führungs- und Arbeitsmodellen gibt es zwar, und sie können auch in bestimmten Situationen durchaus sehr sinnvoll sein, allerdings sind die Change-Aspekte für die Betroffenen ganz erheblich. Denn agile und klassische Führungs- und Arbeitskonzepte sind in vielen Aspekten inkompatibel, nicht zuletzt, weil in traditionellen Unternehmen etablierte Standards und Regelungen, die sog. „Governance“, und damit über Jahre geprägte Verhaltensweisen, die letztendlich eine Firmenkultur entscheidend prägen, oft nicht zu Agilität passen.

Es ist daher wenig überraschend, dass sich viele Organisationen eher schwertun, in einem gewachsenen, traditionellen Führungs- und Arbeitsumfeld echte Agilität nachhaltig zu implementieren. Vereinzelt agile Teams, die mit neuen Methoden und Werkzeugen experimentieren dürfen, gibt es unterdessen durchaus in zahlreichen Unternehmen. Aus den Erfahrungen dieser Teams jedoch *wirklich* zu lernen und dann die Konsequenzen für die restliche Organisation abzuleiten, ist eine ganz andere, wirklich große Herausforderung. Um etwas besser zu verstehen, warum das so ist, werfen wir zunächst einen Blick in die Ursprünge agilen Arbeitens.

Wo kommt „Agiles Arbeiten“ her?

Agilität an sich ist nichts Neues. Immer dort, wo schnelles, flexibles, proaktives und ergebnisorientiertes Handeln gefordert ist, ist ein agiles Vorgehen grundsätzlich sinnvoll. Dies gilt umso mehr, je eher die Situation unübersichtlich, schlecht plan- und vorhersehbar ist. So überrascht es nicht, dass Agilität und agiles Vorgehen in kleinen Teams („Squads“) zunächst im militärischen Kontext erprobt wurde. Auch der Begriff VUCA, der heute oft in einem Atemzug mit Agilität zitiert wird, wurde ursprünglich von den US-Militärs in den 90er Jahren geprägt. US-General Krulak beschrieb 1999 in seinem Beitrag „The Strategic Corporal“ die agile Vorgehensweise als „Operation Absolute Agility“ [Krulak 1999, S. 1]. Interessanterweise stammen diese Ansätze also genau aus dem Umfeld, aus dem historisch auch viele der heutigen traditionellen Begriffe der Unternehmensführung entlehnt sind, siehe den Chief Executive *Officer* oder *Command & Control*. In den folgenden Jahren haben VUCA als zentrale Beschreibung des wirtschaftlichen Umfelds des 21. Jahrhunderts und Agilität als eine mögliche Antwort darauf ihren Weg in die Management-Literatur gefunden.

Warum wird Agilität häufig in Verbindung mit Software genannt? Anders als bei vielen anderen Produktentwicklungen lassen sich Veränderungen in Software technisch verhältnismäßig einfach umsetzen. Diese Veränderungen können zum Beispiel vom Kunden gewünscht sein oder auch einfach nur aus der Beseitigung von Fehlern („bug fixes“) resultieren. Wirklich fehlerfreie Software zu schreiben, ist praktisch unmöglich. Daher kommt der Test- und Erprobungsphase im Projekt eine zentrale Bedeutung zu. Jeder, der ein Smartphone besitzt, weiß unterdessen, wie viele und wie häufig Updates bei Programmen und Apps durchgeführt werden. Der Kunde ist quasi in eine permanente Felderprobung eingebunden, die Entwicklung von Software eigentlich nie zu Ende.

Mit der kontinuierlich zunehmenden Komplexität und Leistungsfähigkeit von Software wurde bereits in den 90er Jahren der Ruf nach einer veränderten Entwicklungssystematik laut, die Erfahrungswissen stärker gewichtet und Entscheidungen auf Basis von Bekanntem fördert. Diese auch als Empirie bezeichnete Vorgehensweise ruht auf drei Säulen, die allen agilen Vorgehensweisen zugrunde liegen: Transparenz, Überprüfung (Inspektion), Anpassung (Adaption).



Die 3 Säulen der Empirie (Bild: pixabay)

Einer der Meilensteine der Einführung von Empirie in der Software-Entwicklung war sicherlich die Veröffentlichung des „[Agilen Manifests](#)“ im Jahre 2001 durch eine internationale Gruppe von 17 Software-Entwicklern, die sich zum Ziel gesetzt hatten: „Wir erschließen bessere Wege, Software zu entwickeln, indem wir es selbst tun und anderen dabei helfen.“

Das Agile Manifest beschreibt allerdings nicht etwa Werkzeuge und Methoden agilen Arbeitens, sondern fokussiert vor allem auf die „weichen“ Faktoren wie Werte und Prinzipien, die sich als Erkenntnis aus jahrelanger praktischer Entwicklungsarbeit als essenziell und erfolgskritisch ergeben haben:

- [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

Die traditionellen Werte auf der rechten Seite werden dabei nicht etwa aufgegeben, sondern nur niedriger gewichtet als die „agilen“ Werte auf der linken Seite. Um diese Werte mit Leben zu erfüllen, definierte die Gruppe außerdem 12 Prinzipien, die bis heute Grundlage agilen Arbeitens und Führens sind. Nachstehend die Kurzfassung, die originale Formulierung findet sich [hier](#).

1. [Kundenzufriedenheit](#) durch frühe und kontinuierliche Auslieferung hat [höchste Priorität](#).
2. [Änderungen](#), auch spät in der Entwicklung, [sind willkommen](#)
3. [Liefere](#) funktionierende Software [regelmäßig in kurzen Abständen](#) aus
4. Fachexperten und Entwickler [arbeiten eng zusammen](#).
5. Gib [motivierten Individuen](#) das benötigte [Umfeld und volle Unterstützung](#). [Vertraue](#) darauf, dass sie die Aufgabe erledigen.
6. Das [persönliche Gespräch](#) ist die [effizienteste und effektivste](#) Methode des Austauschs.
7. [Funktionierende Software](#) ist das wichtigste [Fortschrittsmaß](#).
8. Agile Prozesse fördern [Nachhaltigkeit](#), wenn alle [gleichmäßiges Tempo unbegrenzt](#) halten.
9. Achte auf [technische Exzellenz](#) und [gutes Design](#).
10. Mache Dinge so [einfach wie möglich](#).
11. Die besten Konzepte entstehen durch [selbstorganisierte Teams](#).

12. Regelmäßige Reflexion und Adaption optimieren die Effektivität des Teams.

Aus der Gruppe der Autoren des Agilen Manifests veröffentlichten Ken Schwaber und Mike Beedle 2001 mit „*Agile Software Development with Scrum*“ das erste Buch über das agile Vorgehensmodell SCRUM, dem 2003 Ken Schwabers „*Agile Project Management with Scrum*“ folgte. 2007 erschien schließlich Ken Schwabers drittes Buch, „*The Enterprise and Scrum*“. Darin geht es um die Ausweitung von SCRUM auf das gesamte Unternehmen [Wikipedia].

Das agile Vorgehensmodell SCRUM

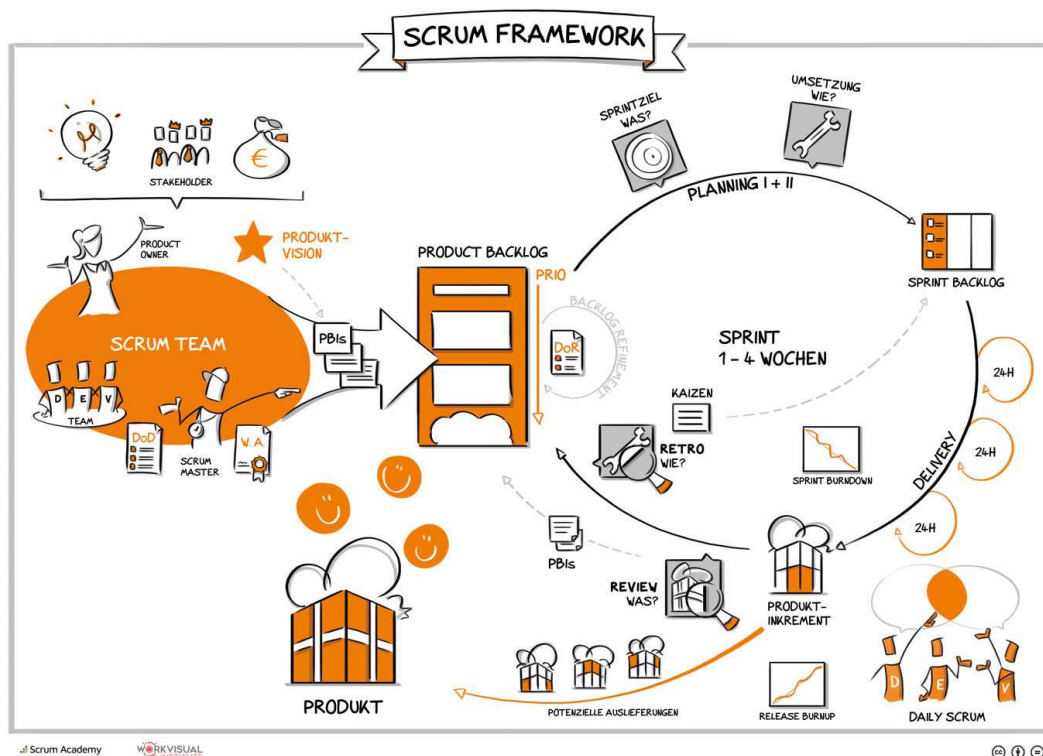
Was genau ist SCRUM? Oft fälschlicherweise als reiner Methodenbaukasten verkannt, ist SCRUM ein zwar gut zu verstehendes, in der Praxis aber meist schwierig umzusetzendes ganzheitliches Vorgehensmodell des Projekt- und Produktmanagements. Richtig angewendet, hilft es Menschen, komplexe adaptive Aufgabenstellungen anzugehen und versetzt sie in die Lage, produktiv und kreativ Produkte mit höchstmöglichem Wert auszuliefern [Auszug Scrum-Guide].

Scrum wird seit den frühen 1990er Jahren als **Prozessrahmenwerk** (*engl. framework*) zum Management der Arbeit an komplexen Produkten verwendet. Es **ist weder ein Prozess noch eine Technik oder vollständige Methode**. Vielmehr gibt es einen Rahmen vor, innerhalb dessen verschiedene Prozesse und Techniken zum Einsatz gebracht werden können. Scrum macht die relative Wirksamkeit des Produktmanagements und der verwendeten Arbeitstechniken (Methoden) sichtbar. Das Scrum-Rahmenwerk besteht aus

- **Scrum-Teams** (agile Teams)

und den dazu gehörigen

- **Rollen** (Product Owner, Scrum Master, Entwicklungsteam)
- **Ereignissen** (Sprint, Sprint-Planung, Daily Scrum, Review und Retrospektive)
- **Artefakten** (Product Backlog, Sprint Backlog, Inkrement)
- **Regeln**.



Jede Komponente innerhalb des Rahmenwerks dient einem bestimmten Zweck und ist unentbehrlich für den Einsatz von Scrum und dessen Erfolg. Durch die Regeln von Scrum werden die Beziehungen und Wechselwirkungen zwischen den Rollen, Ereignissen und Artefakten bestimmt. [Auszug Scrum-Guide].

Scrum wurde ursprünglich dazu entwickelt, um Produkte zu managen und zu entwickeln. Unterdessen wird es weltweit genutzt zur

- Erforschung und Identifizierung rentabler Märkte, Technologien und Produktfähigkeiten
- Entwicklung von Produkten und Erweiterungen
- Auslieferung von Produkten und Erweiterungen, regelmäßig und mehrmals täglich
- Entwicklung und Aufrechterhaltung von Cloud-Umgebungen und anderer Produktivumgebungen
- Erhaltung und Erneuerung von Produkten.

Aus der Software-Entwicklung stammend, werden Scrum und darauf aufbauende Vorgehensmodelle stark zunehmend auch in ganz anderen Feldern eingesetzt: in Regierungs- und Marketingprojekten, zur Verwaltung von Organisationen und generell in der Entwicklung von fast allem, was wir in unserem täglichen Leben als Einzelpersonen oder Gesellschaften verwenden. Für Produkte, Dienstleistungen oder auch das Management von Organisationen.

Vor dem Hintergrund einer starken Zunahme der Technologie-, Markt- und Umweltkomplexitäten (VUCA) erweist sich Scrum vor allem bei dem iterativen und inkrementellen Wissenstransfer als besonders effektiv.

Scrum ist vollständig im [Scrum Guide](#) beschrieben, der von Ken Schwaber und Jeff Sutherland in den späten 90er Jahren entwickelt wurde und seitdem kontinuierlich aktualisiert wird.

Unterdessen gibt es viele Erweiterungen des ursprünglichen Scrum-Rahmenwerks, vor allem in Bezug auf Werkzeuge und Methoden sowie Fragen der Skalierung. Denn Scrum in der Reinform bezieht sich immer auf kleine Teams von nicht mehr als ~ 10 Mitarbeitern.

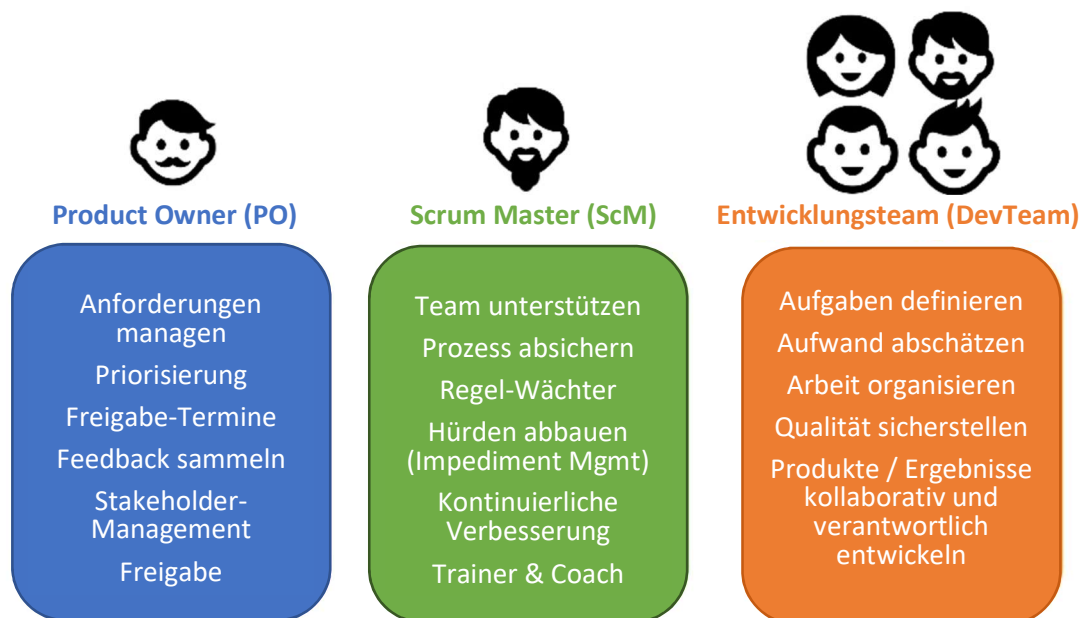
Größere Projekte, die mehr Kapazität erfordern, lassen sich damit nicht mehr abwickeln. Neuere Konzepte wie LeSS (Large Scale Scrum), SAFe (Scaled Agile Framework) oder Scrum of Scrums zielen daher darauf ab, die bekannten Vorteile kleiner Teams zu verbinden mit der Notwendigkeit größerer Einheiten und Kapazitäten. Dabei wird unterschieden zwischen horizontaler Skalierung, also der „Verbreiterung“ auf einer Ebene, z.B. über Unterteilung in Teilprojekte, die gleichberechtigt nebeneinanderstehen, und der vertikalen Skalierung über mehrere Hierarchiestufen, z.B. zur Überführung ganzer Organisationseinheiten in agile Strukturen. Die resultierenden Organisationsentwicklungen sind allerdings meist ziemlich komplex, weshalb man sie bisher noch selten findet.

Im Folgenden wird daher nur die grundlegende Form des einfachen Scrum-Teams und das Zusammenspiel seiner Elemente eingegangen.

Product Owner, Scrum Master und Team – der magische Dreiklang

Es ist kein Zufall, dass in Scrum nur diese Rollen beschrieben werden, denn sie sind fein aufeinander abgestimmt und voneinander abhängig. Die „Magie“ des Zusammenspiels entfaltet sich daher auch erst, wenn alle Beteiligten das „richtige“ Rollenverständnis haben und danach „leben“. Werden die Rollen nicht richtig verstanden und/oder unzureichend umgesetzt, besteht leicht die Gefahr von Reibung, Verwirrung und Blindleistung; die erhoffte Verbesserung bleibt dann aus oder kehrt sich um ins Gegenteil.

Besonders kritisch ist die Rolle des **Product Owners (PO)** und die Tatsache, dass es in Scrum aus gutem Grund keinen Projektleiter gibt. Vielmehr ist das Scrum-Team gesamtheitlich für den Erfolg des Projekts und das Ergebnis verantwortlich. Besonders in traditionell hierarchisch geprägten Unternehmen wird häufig der PO gleichgesetzt mit einem quasi „agilen Projektleiter“, was aber nicht nur ungenau, sondern schlicht und ergreifend falsch ist. Noch schlimmer wird es, wenn Product Owner und Scrum Master (ScM) sich in einer Person finden, oder der Product Owner aktiver Teil des Entwicklungsteams ist, der auch noch selbst entwickelt.



Aus dem Scrum-Guide:

„Der PO ist dafür verantwortlich, den Wert des Produktes zu maximieren, das aus der Arbeit des Entwicklungsteams entsteht. Wie dies geschieht, kann je nach Organisation, Scrum-Team und Einzelpersonen stark variieren.“

Der PO ist die einzige Person, die für das Management des Product Backlogs verantwortlich ist. *(Zur Erklärung: das Product Backlog enthält eine priorisierte, stets aktuelle gehaltene Liste aller Anforderungen an das Produkt/Projektergebnis).*

Das Management des Product Backlogs umfasst:

- Die Product-Backlog-Einträge klar zu formulieren
- Die Einträge im Product Backlog so zu sortieren, dass Ziele und Missionen optimal erreicht werden können;
- Den Wert der Arbeit zu optimieren, die das Entwicklungsteam erledigt;
- Sicherzustellen, dass das Product Backlog sichtbar, transparent und für alle klar ist, sowie zeigt, woran das Scrum-Team als nächstes arbeiten wird;
- Sicherzustellen, dass das Entwicklungsteam die Product-Backlog-Einträge im erforderlichen Maße versteht.

Der PO kann die oben genannten Arbeiten selbst durchführen oder sie durch das Entwicklungsteam erledigen lassen. Der PO bleibt jedoch immer rechenschaftspflichtig“.

Der Product Owner führt über Anforderungen und (Kunden-)Erwartungen, also vor allem über Inhalte. Er ist so etwas wie der „Sprecher“ des Kunden und anderer Stakeholder sowie Nutzer (User). Als Coach sorgt er dafür, dass sein Entwicklungsteam die bestmöglichen Rahmenbedingungen vorfindet, um den Wert des Produktes zu maximieren. Er definiert das **WAS** und hilft, die Arbeit des Teams zu optimieren. Umgekehrt ist der PO auch Sprecher seines Teams und hat die Aufgabe, dessen Interessen gegenüber Dritten zu vertreten. Eine höchst anspruchsvolle Aufgabe also.

Die Frage, **WIE** die Projektziele erreicht werden, liegt dagegen federführend beim **Entwicklungsteam (DevTeam)**. Auch hierzu wieder ein Auszug aus dem Scrum-Guide:

„...Entwicklungsteams sind von der Organisation so strukturiert und befähigt, dass sie ihre eigene Arbeit selbst organisieren und managen. Die daraus resultierende Synergie optimiert die Gesamteffizienz und -effektivität des Entwicklungsteams.

Entwicklungsteams weisen die folgenden Eigenschaften auf:

- Sie sind selbstorganisierend. Niemand sagt dem Entwicklungsteam, wie es aus dem Product Backlog potenziell auslieferbare Funktionalität erzeugen soll.
- Entwicklungsteams sind interdisziplinär. Sie haben als Team alle Fähigkeiten, die notwendig sind...
- Scrum kennt für Mitglieder des Entwicklungsteams keine Titel. Dies ist unabhängig von der Arbeit, die diese Personen erledigen.
- Scrum kennt keine weiteren Unterteilungen innerhalb des Entwicklungsteams, ungeachtet der verschiedenen Themenfelder, mit denen das Team sich befasst...
- Einzelne Mitglieder des Entwicklungsteams können zwar spezialisierte Fähigkeiten oder Spezialgebiete haben, aber die Rechenschaftspflicht obliegt dem Team als Ganzem.“

In der Praxis verhandeln PO und DevTeam die für den jeweils nächsten Entwicklungsschritt (*Iteration*) geplanten Ergebnisse (*Inkremete*). Das Team committet sich zur Umsetzung, kann im Gegenzug dann aber weitestgehend frei entscheiden, WIE diese Umsetzung erfolgt, sprich mit welchen Methoden, Werkzeugen, Organisation, etc. Natürlich gibt es auch hier Grenzen, z.B. gesetzliche oder vom Unternehmen vorgegebene (Governance) und Regeln (z.B. aus Scrum).

Das Entscheidende ist jedoch, dass sich der PO aus dem eigentlichen (Entwicklungs-)prozess weitgehend raushält und erst wieder ins Spiel kommt, wenn die Ergebnisse im Rahmen des nächsten Reviews, am Ende des Entwicklungsschritts, vorliegen. Der PO hat eine Rolle sehr ähnlich der eines Fußball-Trainers, der zwar sehr wohl weiß, was die Zuschauer erwarten, was seine Mannschaft abzuliefern hat und in welchem Rahmen, sie daraufhin trainiert, bestmöglich vorbereitet und qualifiziert, aber natürlich nicht selbst auf dem Feld mitspielt.

Bleibt noch der **Scrum Master (ScM)**: er ist der Scrum-Experte und Prozesswächter, der Schiedsrichter und Sparrings-Partner, der genau darauf achtet, dass Regeln, Prozesse und Vereinbarungen im Sinne Scrum eingehalten werden. Er fungiert als neutraler Ansprechpartner für das gesamte Team und ggf. weitere Stakeholder (z.B. Produkt-Nutzer oder Führungskräfte und Experten aus der Linie). Bei Unternehmen oder Teams, die noch wenig Erfahrung mit Agilität haben, kann dem ScM auch noch die Rolle des Agilen Coaches für das Team und weitere Teile der verbleibenden Organisation zufallen.

„Der Scrum Master ist ein Servant Leader“

Zitieren wir wieder aus dem Scrum Guide:

„Der Scrum Master ist dafür verantwortlich, Scrum entsprechend des Scrum Guides zu fördern und zu unterstützen. Scrum Master tun dies, indem sie allen Beteiligten helfen, die ...Theorie, Praktiken, Regeln und Werte zu verstehen. Der Scrum Master ist ein „Servant Leader“ für das Scrum-Team. Der Scrum Master hilft denjenigen, die kein Teil des Scrum-Teams sind, zu verstehen, welche ihrer Interaktionen mit dem Team sich hilfreich auswirken und welche nicht. Der Scrum Master hilft dabei, die Zusammenarbeit so zu optimieren, dass der durch das Scrum-Team generierte Wert maximiert wird.“

Für Unternehmen, die sich erst seit kurzer Zeit mit Agilität beschäftigen, sind kompetente Scrum Master, die sowohl das Scrum-Framework als auch das operative Geschäft gut verstehen, von zentraler Bedeutung. Keinesfalls sollten die Scrum Master- und Product Owner-Rollen in einer Person gebündelt werden, da dies sonst die notwendige Neutralität und damit Akzeptanz des Scrum Masters untergräbt.

Oft ist es sinnvoll, wenn sich Unternehmen schon sehr früh darum bemühen, in der eigenen Organisation ausgewählte Mitarbeiter zu Scrum Mastern auszubilden, die dann als Dienstleister der verschiedenen Scrum Teams zur Verfügung stehen. Oft entstehen daraus virtuelle ScM-Pools, in denen das (Erfahrungs-)wissen gebündelt und weiterentwickelt wird. In der Anfangsphase können auch qualifizierte Externe als Interims-Manager diese Aufgabe gut übernehmen, um die Lernkurve für das Unternehmen möglichst steil zu halten und schnell zu Ergebnissen zu kommen.

Herausforderungen in der Umsetzung

Neue Methoden und Werkzeuge (Tools) zu erlernen und einzuführen ist zwar manchmal auch anspruchsvoll, gelingt jedoch mit Engagement, Ausdauer und gutem Willen meist recht gut. Wesentlich schwieriger ist eine *nachhaltige* Veränderung von menschlichem Verhalten, zumal dann, wenn das bisherige Verhalten in einer gewachsenen (Firmen-) Kultur erfolgreich war.

Agilität im Unternehmen einzuführen ist daher auch weniger eine Frage der dazu notwendigen Tools als des dafür erforderlichen Mindsets und Verhalten der Betroffenen. Sowohl Führungskräften als auch Mitarbeitern fällt es häufig schwer, die hier beschriebenen Aufgaben und Verantwortlichkeiten im agilen Umfeld wahrzunehmen und tatsächlich auszufüllen. Während Führungskräfte meist nur ungern „loslassen“, um sich in die Rolle des Kunden bzw. Anforderungsgebers (PO) zu begeben, zaudern Mitarbeiter vielfach, die neu gewonnene Freiheit und damit einhergehende Verantwortlichkeiten auch wirklich anzunehmen. In diesem Paradigmenwechsel liegt die eigentliche Herausforderung in der Agilisierung.

Hinzu kommt, dass agiles Arbeiten und Führen in aller Regel nur bestimmte Teile einer Organisation betreffen, während der Rest weiter arbeitet wie bisher. Kunden-/marktnahe Bereiche und solche, in denen ein hohes Maß an Kreativität die Arbeit bestimmt, eignen sich eher für Agilität als Einheiten, deren Arbeit von repetitiven, auf Effizienz getrimmten Prozessen und Standards dominiert wird.

An den Schnittstellen zwischen diesen unterschiedlichen Gruppen kommt es dann häufig zu Spannungen. Ungeregelter „Zugriff“ auf Experten aus der Linienorganisation, für agile Teams unpassende Regeln und Standards, Linien-Führungskräfte, die von der „Seite“ in die agilen Teams hineinregieren, Nichtanerkennen der Autorität des Product Owners, unzureichendes Verständnis zu Aufgabenstellungen, die für agiles Vorgehen geeignet sind... das sind nur einige typische Probleme, die sehr schnell auftreten.

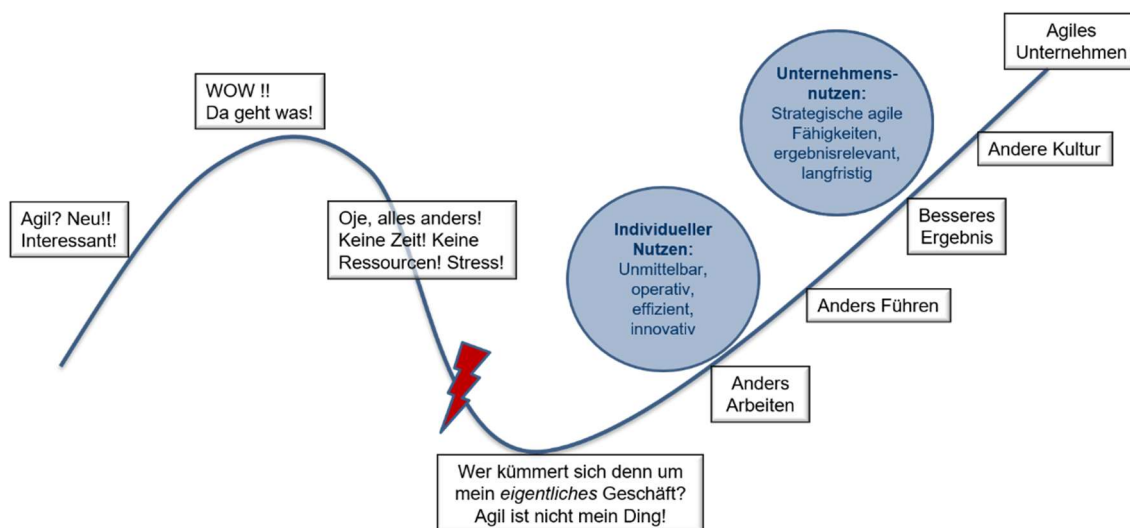
Die Erfahrung zeigt, dass die Weiterentwicklung einer Firmenkultur in Richtung mehr Agilität nur dauerhaft gelingen kann, wenn sie top-down entfaltet und vorgelebt wird. Das bedeutet aber nicht, dass Initiativen für agiles Arbeiten und Führen nicht aus der Mitte einer Organisation oder gar von der Arbeitsebene kommen sollten. Im Gegenteil: es braucht unbedingt die Experten auf der Arbeitsebene, die sich gut mit agilen Tools und Vorgehensweisen auskennen und verstehen, wie eine agile Vorgehensweise ihren Arbeitsalltag und ihr jeweiliges Geschäft verbessern könnte. Solche Impulse sollten Führungskräfte unbedingt ernst nehmen und aktiv fördern. In vielen Firmen passiert das ja durchaus auch. Die zahlreichen kleineren agilen Projekte, Initiativen, Inkubatoren, etc., die man in vielen Organisationen antrifft, zeigen dies ja deutlich.

Erfolgskritisch und in vielen Fällen eine große Hürde ist allerdings der nächste Schritt: das Hochskalieren von Agilität auf größere Bereiche des Unternehmens. Das gelingt ohne Einbindung und aktive Mitarbeit des Top-Managements einfach nicht. Die Gründe dafür liegen auf der Hand:

Aus der beschriebenen Inkompatibilität der dominierenden Prinzipien zwischen agilem (ergebnis- und kundenorientierten) und klassischem (prozess- und synergieorientierten) Arbeiten resultiert automatisch größerer Veränderungsbedarf am gesamten System. Angefangen von den Arbeitsprozessen über Fragen der Organisationsstruktur, Rollen und Verantwortlichkeiten von Führungskräften und Mitarbeitern bis hin zu einem angepassten Regelwerk gilt es, eine Richtung und einen Weg zu finden, der für das *gesamte* Unternehmen gangbar ist.

Dies kann nur Erfolg haben, wenn eine Agile Transformation auch ganzheitlich (systemisch) angegangen wird. Das Top-Management sowie die nachgelagerten Führungskreise müssen über ausreichend Wissen und Verständnis zu Agilität verfügen. Neben einer grundsätzlichen Überzeugung zum Mehrwert von Agilität für ihr eigenes Geschäft – sie müssen daran glauben – sollte eine klare, gemeinschaftliche Einschätzung im Führungskreis bestehen, was bis wann erreicht werden soll, was dafür notwendig ist und welche möglichen Konsequenzen resultieren.

Leider wird immer noch vielfach unterschätzt, wie viel Zeit und Kraft solche Veränderungen erfordern und dass sich signifikanter Mehrwert meist erst mittelfristig zeigt. Eine Folge ist, dass oft zu früh die Energie „von oben“ nachlässt und die Organisation dann wieder in den alten Arbeitsmodus zurückfällt. Dieses Phänomen ist charakteristisch für viele Veränderungsthemen und sicherlich ein Grund, warum die meisten Veränderungsinitiativen scheitern.



Veränderungskurve nach Kübler-Ross

Ist man als Organisation mal in das „Tal der Tränen“ gestürzt und findet nicht zügig wieder raus, dann ist das Veränderungsthema meist „verbrannt“. Diejenigen, die früh als Unterstützer und Promotoren in Erscheinung getreten sind, sind zutiefst enttäuscht, weil sich ihre Anstrengung und Mehrbelastung, die eigentlich ja immer entsteht, nicht gelohnt hat. Auf der anderen Seite ist es Wasser auf die Mühlen derjenigen, die „es ja schon immer wussten“ und einfach nur abgewartet haben, bis auch diese „Sau durchs Dorf getrieben“ ist. Eine Organisation, die einmal diese negative Erfahrung gemacht hat, ist kurzfristig kaum mehr fürs gleiche Thema zu mobilisieren. Hierin liegt die eigentliche Gefahr.

Wie kann man dem vorbeugen? Eine agile Transformation gehört schon allein aus Gründen der Budgetierung und Ressourcen zwingend in die Unternehmensstrategie. Dort ist sie in Bezug zum jetzigen Geschäft einzuordnen und es sind grundlegende Fragestellungen in Bezug auf Einsatzgebiet, Priorisierung, Geschäftsmodell, und ähnliches zu beantworten. Dabei sollte auch eine glaubhafte Change Story entwickelt werden, die dem Unternehmen mittelfristig Orientierung gibt. Außerdem hat es sich bewährt, die Veränderungen iterativ in überschaubaren Schritten anzugehen und dabei immer wieder zu prüfen, ob man noch auf dem richtigen Weg ist. Womit die Umsetzung der agilen Transformation selbst zur *agilen Weiterentwicklung der Organisation* wird.