

Spotlight

Agil im Großen – Frameworks für den Wandel



Eine themenspezifische Zusammenstellung von Fachartikeln
aus dem Projekt Magazin

www.projektmagazin.de

Mehlbeerenstr. 4, 82024 Taufkirchen

Tel: +49 89 2420798-0

Fax: +49 89 2420798-8

Agil im Großen – Frameworks für den Wandel

Ihre Projekte sind für ein einzelnes Scrum-Team zu groß? Dann benötigen Sie einen geeigneten organisatorischen Rahmen für mehrere, parallel arbeitende Teams: ein Skalierungs-Framework. Welche Frameworks es gibt und wie Sie das passende für Ihre Teams und Anforderungen finden, erfahren Sie in diesem Spotlight. Beiträge zur Produktentwicklung mit dem Large Scale Scrum-Framework (LeSS) und der Kombination aus LeSS und SAFe (Scaled Agile Framework) veranschaulichen detailliert, wie der Einsatz in der Praxis aussieht. Was die Führung zum hierfür notwendigen agilen Wandel beitragen kann, erfahren Sie im abschließenden Kapitel.

Inhalt

Scrum skalieren - das passende Framework finden

1. Scaled Agile Framework – eine Entscheidungshilfe
Agil im Großen: Das richtige Framework für Sie Seite 3
2. Agil Skalieren
Koordination agiler Entwicklungsteams..... Seite 13
3. Teamzuständigkeiten in Scrum – von Komponententeams zu Feature Teams..... Seite 22

Produkte entwickeln mit Large-Scale Scrum

4. Wenn es auf Geschwindigkeit und Flexibilität ankommt
Produkte, Projekte, Portfolios, Experimente – die agile Transformation der Haufe Group Seite 29
5. Mit vielen Teams erfolgreich ein Produkt mit Scrum entwickeln
Teil 1: Rahmenbedingungen und Basismodell..... Seite 46
6. Mit vielen Teams erfolgreich ein Produkt mit Scrum entwickeln
Teil 2: LeSS Hufe-Regeln und Einführungsmaßnahmen Seite 58

Agile Transition: Auswirkungen auf Kultur und Führung

7. Erfahrungsbericht der Fiducia IT AG
Den Kulturwandel zum agilen Unternehmen aktiv gestalten..... Seite 65
8. Agile Transition – Führung im Wandel
Teil 1: Die neue Rolle der Führungskraft..... Seite 79
9. Agile Transition – Führung im Wandel
Teil 2: Die ersten Schritte in die agile Welt..... Seite 88

Scaled Agile Frameworks – eine Entscheidungshilfe

Agil im Großen: Das richtige Framework für Sie

Unter den Frameworks für agile Entwicklung hat sich vor allem Scrum weit verbreitet. Beim Einsatz von Scrum profitieren gerade kleine und mittelständische Unternehmen in technologienahen Märkten von einer verkürzten Time-to-Market und von hoher Flexibilität – gegenüber Kundenanforderungen und der Variabilität des Marktes. Inzwischen versuchen immer mehr etablierte Konzerne, es ihnen gleichzutun. Auch sie wollen durch Scrum mehr Flexibilität und Innovationskraft gewinnen und ihre Marktanteile sichern. Die Herausforderung dabei: Scrum beruht auf der Arbeit eines kleinen Teams und liefert zunächst keine Lösungen für den Einsatz in einer großen Organisation mit einer Vielzahl von Teams.

Autor



Patrick Daut

Dipl.-Wirt.-Inf., seit 2011 als Berater mit Schwerpunkt E-Business bei der Cassini Consulting beschäftigt.

Kontakt: patrick.daut@cassini.de

Mehr Informationen unter:

[projektmagazin.de/autoren](https://www.projektmagazin.de/autoren)

Aus diesem Bedarf heraus sind in den letzten Jahren verschiedene Ansätze zur Skalierung von agilem Vorgehen entstanden. Dazu zählen Frameworks wie das Scaled Agile Framework (SAFe) oder Large-Scale Scrum (LeSS) sowie Ansätze wie der Agile Scaling Cycle oder das populäre Best Practice-Beispiel wie bei Spotify. Auch wenn es wichtige Gemeinsamkeiten gibt, zeigt die nähere Betrachtung doch, dass nicht jedes Skalierungsframework für jeden Bedarf gleich gut geeignet ist. Wenn Sie Entwicklungs- oder Projektleiter sind und vor der Herausforderung stehen, ein passendes Framework zur Skalierung agilen Vorgehens für Ihr Umfeld auszuwählen, finden Sie in diesem Artikel Hinweise, welche Aspekte und Kriterien Sie bei der Auswahl berücksichtigen sollten.

Skalierung als Herausforderung

Aktuell setzen bereits zahlreiche große Organisationen agile Methoden ein. Einige unserer Kunden befinden sich häufig in einer der folgenden Situationen:

- Agile Methoden werden rein auf der Ebene der Entwicklungsteams eingesetzt, andere Bereiche sind nicht angebunden, und der mögliche Nutzen wird nicht voll ausgeschöpft.
- Agiles Vorgehen wurde in Form eines Piloten in einem Projekt oder ein bis drei Entwicklungsteams erprobt. Die Ergebnisse sind überzeugend – aber wie sieht eine Lösung für das gesamte Projektportfolio, die gesamte Produktentwicklung oder die gesamte Unternehmung aus?
- Die Produktentwicklung erfolgt agil – aber ein Produktmanagement ist nicht konsequent angebunden, sondern erfolgt separiert und vorgelagert. Klassische Budgetprozesse schränken die Flexibilität ein.

Gut möglich, dass auch Sie gerade an einem dieser Punkte stehen.

Aus dem praktischen Bedarf heraus sind in den vergangenen Jahren verschiedene methodische Ansätze und Frameworks entstanden, die die Skalierung agilen Vorgehens in großen Organisationen zu lösen versuchen.

Diese Frameworks unterscheiden sich zum Teil erheblich in Ansatz, Ausgestaltung und im Grad der spezifischen Beschreibung. Bemerkenswert ist aber, dass sie alle auf ähnlichen Ideen und Prinzipien beruhen, die letztlich aus dem Bereich des Lean Thinking stammen. Dazu gehören die Fokussierung auf Werthaltigkeit, Wertströme, Flussorientierung, Pull-Prinzip und Qualität – all dies basierend auf einem hohen Stellenwert des Menschen. Zudem weisen selbst ganz verschiedene agile Skalierungsansätze große Ähnlichkeiten bei den grundlegenden Organisationsstrukturen auf, die durch sie definiert werden.

Der Markt der Skalierungsframeworks

Im Folgenden wollen wir vier Ansätze zur Skalierung vorstellen, um anschließend zu diskutieren, für welchen Bedarf sie sich eignen.

Das Scaled Agile Framework (SAFe)

Das Scaled Agile Framework (SAFe) lässt sich als komplexes Framework zur Einbettung agiler Entwicklung in eine große Organisation unter Einbindung der klassischen Unternehmensbereiche beschreiben. SAFe ist komplex in dem Sinne, dass es die folgenden Bestandteile definiert:

- eine Aufbauorganisation in Form von Strukturen, Rollen und deren Abhängigkeiten
- einen Prozess sowie Artefakte, die zu definierten Zeitpunkten erstellt werden
- Methoden und Praktiken.

Den Kern von SAFe bildet eine über die Zeit stabile Organisationsstruktur, die die Arbeit mehrerer an einem (Teil-)Produkt beteiligten Teams zusammenhält und koordinative Rollen definiert. Die Teams selbst folgen einer erweiterten, angepassten Form von Scrum. Das SAFe Framework der Scaled Agile Inc. ist durch mehrere Bücher von Dean Leffingwell sowie eine inhaltsreiche Website (www.scaledagileframework.com) beschrieben, und eine starke Community wird aktiv betreut. Daneben existiert ein Trainings- und Zertifizierungsprogramm. All dies ergibt den Eindruck eines aktiv vermarkteten Produkts.

Das Large-Scale Scrum (LeSS)

Large-Scale Scrum (LeSS) definiert vor allem zwei Aspekte:

- Eine einfache Struktur, die zum größten Teil aus Scrum-Teams und einer Hierarchie aus Product Ownern besteht, wobei der Verantwortungsbereich eines Product Owners auf tieferen Hierarchie-Ebenen kleiner wird.
- Einen einfachen, ebenfalls weitgehend auf Scrum basierenden Ablauf pro Team mit einzelnen Koordinationsmechanismen zwischen den Teams wie Scrum-of-Scrum oder teamübergreifenden Retrospektiven.

LeSS basiert auf der Arbeit von Craig Larman und Bas Vodde, die die beiden Autoren zunächst in Form zweier Bücher publiziert haben. Auch für LeSS sind inzwischen eine inhaltsgetriebene Website (<http://less.works>) sowie ein Trainings- und Zertifizierungsprogramm verfügbar.

Der Agile Scaling Cycle

Der Agile Scaling Cycle definiert im Gegensatz zu SAFe und LeSS weniger ein Framework für die Aufbau- und Ablauforganisation als ein Vorgehen zur Entwicklung einer solchen Organisationsform für die jeweilige Umgebung. Dies ist Kanban, dem Ansatz zur Produktionsprozesssteuerung, nicht unähnlich: auch Kanban beschreibt Schritte zur Definition eines situationsgerechten, individuellen Prozesses, nicht aber den Prozess selbst. Näher ausgeführt wird der Scaling Cycle in einzelnen Blogposts und Vorträgen, unter anderem von Stefan Rook (<https://stefanroock.wordpress.com>).

Die Spotify Engineering Culture

Die Spotify Engineering Culture beschreibt die Organisationsstruktur der Produktentwicklung beim schwedischen Musikstreaming-Dienst Spotify mit einem Fokus auf der Aufbauorganisation. Es lässt sich dabei weniger von einem Framework als vielmehr von einem gelebten Praxisbeispiel sprechen – in einer leichtgewichtigen Organisationsform mit Fokus auf Strukturierung und Koordination von Teams. Auffälliges Merkmal des Best Practice-Beispiels von Spotify ist eine "Tribe" genannte stabile Struktur, die als koordinierende Organisationsform dient. Im Tribe werden mehrere agile Teams – Squads genannt – zusammengefasst. Die vom Berater Henrik Kniberg geprägte Spotify Engineering Culture wird in sehr populären Beiträgen im Spotify Labs Blog (<https://labs.spotify.com>) beschrieben.

Quervergleich der vier Frameworks

Es lohnt sich, einen detaillierteren Quervergleich der vier Frameworks anzustellen. Dazu sollen hier insgesamt sieben Aspekte dienen – vom grundlegenden Ansatz jedes Frameworks bis zum jeweils verfügbaren Verfahren für die Bewältigung von Komplexität.

Ansatz

SAFe bildet recht komplexe Strukturen ab und arbeitet mit einer Vorgehensweise über drei Ebenen hinweg. LeSS stellt dagegen im Wesentlichen eine Erweiterung von Scrum um einzelne Elemente und Koordinationsmechanismen dar – und sammelt dazu einzelne Praktiken. Der Agile Scaling Cycle wiederum will gar kein fertiges Framework sein, sondern skizziert vielmehr ein Vorgehen, mit dem sich eine individuelle Organisationsform entwickeln lässt – ohne diese Organisationsform näher zu beschreiben. Die Spotify Culture schließlich stellt lediglich eine in der Praxis gelebte Organisationsstruktur und -kultur vor.

Basis

Bei SAFe bilden Feature-Teams die Basis, die im Wesentlichen nach Scrum arbeiten. Auch LeSS arbeitet mit Feature-Teams. Der Agile Scaling Cycle spricht von unabhängigen Teams, und die Spotify Engineering Culture kennt sogenannte Squads: unabhängige, cross-funktionale Teams.

Hierarchie/Linie

Während SAFe klar definierte Rollen mit ihren jeweiligen Verantwortungsbereichen kennt, geht LeSS von einer hierarchischen Struktur aus. Diese klare Hierarchie in LeSS besteht aus der Entwicklungsleitung, dem Chief Product Owner, dem Area Product Owner und schließlich aus einem Product Owner. Der Agile Scaling Cycle macht hier – seinem Ansatz gemäß – keine konkreten Vorgaben und empfiehlt lediglich eine Ausgestaltung, die der jeweiligen Organisation entspricht. In der Spotify Engineering Culture bilden die sogenannten Chapters eine hierarchische Struktur.

Aggregationsebene

Wie fassen die Frameworks mehrere Teams zusammen, die an einem gemeinsamen Teilprodukt arbeiten und deren Arbeit somit stark voneinander abhängt? Hier definiert SAFe eine als Agile Release Train bezeichnete Organisationsstruktur. Dieser Agile Release Train ist eine stabile und langfristig angelegte Aggregation von Teams mit 100 bis 120 Personen. Begrenzender Faktor ist hier die sogenannte Dunbar-Zahl, also die aufgrund unserer kognitiven Ausstattung begrenzte Anzahl von Menschen, zu denen eine einzelne Person noch soziale Beziehungen aufrecht erhalten kann. Für LeSS heißt die Aggregationsebene die Area. Diesen Areas liegt in LeSS eine fachliche und/oder architektonische Zerlegung zugrunde. Der Agile Scaling Cycle trifft über die Aggregationsebene keine Aussage. Und die Spotify Engineering Culture arbeitet mit Tribes, denen – der Dunbar-Zahl entsprechend – 100 bis 120 Personen angehören können.

Product Owner

SAFe sieht jeweils ein Teammitglied in der Rolle des Product Owners, hinzu kommt eine Koordinierung der Rollen auf Programm- und Portfolio-Ebene. LeSS geht von einem Team von Product Ownern und ggf. – bei einer großen Anzahl von Teams – auch von einer Hierarchie von Area Product Ownern aus. Auch der Agile Scaling Cycle spricht von einem Product Owner, wobei die Teams selbst die Fähigkeit besitzen, grobe Features zu zerlegen. In der Spotify Engineering Culture schließlich gibt es pro Squad einen Product Owner.

Querschnittsfunktionen

Querschnittsfunktionen sind in SAFe sicherlich am ausführlichsten beschrieben und abgebildet. SAFe definiert Rollen für Funktionen wie Architektur, User Experience etc. Diese Rollen haben in SAFe zwar koordinative und steuernde Funktionen, sie nehmen dabei aber nicht die Verantwortung aus den Teams heraus. LeSS behandelt das Thema Querschnittsfunktionen durch sogenannte Communities of Practice – ein Vorschlag, der sich ähnlich auch im Agile Scaling Cycle wiederfindet. In der Spotify Culture werden Querschnittsfunktionen von den Guilds bzw. den Chapters übernommen. Guilds basieren dabei auf einer ähnlichen Idee wie Communities of Practice, während Chapters formelleren Charakter haben.

Komplexitätsbewältigung

Komplexitätsbewältigung gehört zu den großen Stärken von SAFe. Entsprechend gibt es in SAFe Rollen- und Koordinationsmechanismen zum Managen von Komplexität – und zu deren Verringerung. Als Beispiel sei hier nur die Identifikation von Abhängigkeiten zwischen Teams im gemeinsamen Release Planning angeführt. Zwar gibt

es auch bei LeSS einzelne Mechanismen, die zum Umgang mit bestehenden Abhängigkeiten eingesetzt werden können, aber grundsätzlich will LeSS weniger eine bestehende Komplexität abbilden, als vielmehr Abhängigkeiten reduzieren, um auf diese Weise unabhängige Teams mit der Möglichkeit zur Selbstorganisation zu schaffen.

Auch der Agile Scaling Cycle legt es etwaigen Anwendern nahe, Abhängigkeiten zu vermindern, um so unabhängige, selbstorganisierende Teams zu ermöglichen. Die Spotify Engineering Culture schließlich sagt kaum etwas über die Behandlung von Abhängigkeiten innerhalb eines Tribes aus. Das Ziel der Spotify Culture ist es vielmehr, Abhängigkeiten zwischen verschiedenen Tribes zu minimieren, Mechanismen zur Identifikation solcher Abhängigkeiten zu schaffen und sie zu managen.

Gemeinsamkeiten und Unterschiede

Sicherlich sind im direkten Vergleich der vier Skalierungsansätze grundlegende Unterschiede feststellbar. Schon die Aggregationsebene zur Koordination mehrerer Teams wird in den Frameworks teilweise sehr unterschiedlich gebildet. Auch die Einführung einer hierarchischen Organisationsstruktur, die letztlich dem Zwecke der Delegation dient, lösen die vier Frameworks auf verschiedene Weise. Andererseits finden sich Ideen wie etwa die Communities of Practice, die der fachlichen Koordination über das einzelne Team hinaus dienen sollen, in fast allen Skalierungsansätzen wieder.

Ähnliches gilt für die Bedeutung, die unabhängige Feature-Teams bei der Skalierung haben sollen. Der Hintergrund ist hier die Minimierung von Abhängigkeiten und damit eine Minimierung des Koordinationsbedarfs über Teamgrenzen hinweg. Bemerkenswert ist vielleicht, dass gerade ein mächtiges Framework wie SAFe, das in der Literatur teilweise für die Komplexität seiner Spezifikationen kritisiert wird, im Hinblick auf die definierten Organisationsstrukturen doch große Ähnlichkeit mit dem Modell der Spotify Engineering Culture aufweist.

Zwei Dimensionen zur Auswahl eines Frameworks

Wenn Sie als Projektleiter oder Manager vor der Aufgabe stehen, einen dieser vier Ansätze für die eigene Organisation auszuwählen, erweisen sich letztlich zwei Dimensionen als ausschlaggebend:

- Die Größe der Entwicklungsorganisation
- Die Komplexität von Unternehmen, Produkt und Branche

Diese beiden Dimensionen sind die wesentlichen Einflussfaktoren dafür, welchen Bedarf an Steuerungs- und Kontrollmechanismen es in Aufbau- und Ablauforganisation gibt. Ich empfehle Ihnen also, vor allem darauf zu achten, welche Unterschiede in der Komplexität der Frameworks, im Detailgrad der definierten Strukturen, im Umgang mit Komplexität und in den teamübergreifenden Koordinationsmechanismen bestehen.

Es gibt eine ganze Reihe von Einzelkriterien, die man heranziehen kann, wenn man das für einen spezifischen Anwendungsfall geeignete Framework auswählen möchte. Die Bedeutung jedes einzelnen Kriteriums für den eigenen, individuellen Fall genau zu quantifizieren und zu gewichten, bleibt indes schwierig. Dennoch ist es hilfreich, wenn Sie sich die folgenden Fragen stellen, um die eigenen Ansprüche an ein Skalierungsframework sinnvoll skizzieren zu können.

Die Größe der Entwicklungsorganisation

Wie groß ist die Entwicklungsorganisation?

Die Entwicklungsorganisation in unserem Sinne ist der Teil der Gesamtorganisation, in dem die tatsächliche Entwicklung stattfindet. Hier ist relevant, wie viele Mitarbeiter Ihre Produktentwicklungs- bzw. Produktmanagementorganisation hat und wie viele Teams es in ihr gibt.

Wie viel Erfahrung mit agilen Methoden gibt es bereits?

Ist die agile Entwicklung ein neues Thema für das ganze Unternehmen? Gab es vielleicht fehlgeschlagene Versuche? Oder ist die agile

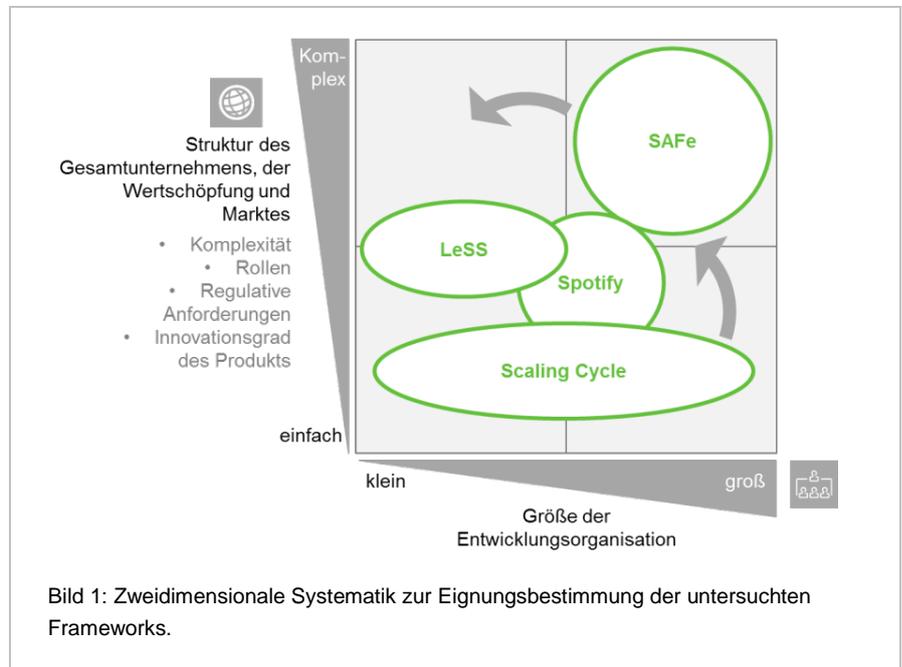
Entwicklung erfolgreich, und es fehlt nur noch an den Schnittstellen? Eine angrenzende Fragestellung, die sich aus meiner Erfahrung als nützlich erwiesen hat: Fragen Sie sich, wie aufwendig es wäre und wie viel Überzeugungsarbeit nötig ist, um in Ihrem Umfeld interdisziplinäre Teams zu bilden, physisch zusammzusetzen und gemeinsam an einem nutzbringenden Produktteil arbeiten zu lassen. Versuchen Sie einmal, die konkreten Teammitglieder zu benennen – oder dies als Auftrag weiterzugeben.

Sind Sie davon gar nicht weit weg? Oder haben Sie Zweifel, ob sie Ihre Mitarbeiter und Kollegen überzeugen können? Welche praktischen Probleme sind zu lösen, und wie stellen Sie sich die Zusammenarbeit der Menschen ganz plakativ vor? Auf diese Weise gewinnen Sie einen Eindruck von der Nähe Ihrer Kultur zur Agilität. In vielen der Projekte, die ich begleiten konnte, war die Zusammenstellung eines ersten Pilotteams ein Knackpunkt. Die Diskussionen und Verhandlungen, die dazu geführt werden, sagen bereits viel über die Kultur aus.

Die Komplexität von Unternehmen, Produkt und Branche

Wie groß ist die Gesamtunternehmung?

Anders gesagt: Wie viele Mitarbeiter hat Ihre Gesamtorganisation? Auch die reine Größe eines Unternehmens ist natürlich bereits ein Indiz für die zu erwartende Komplexität.



Welche Komplexität hat die Aufbauorganisation?

Wie das Unternehmen aufgebaut ist, beeinflusst natürlich die Komplexität einer Skalierung. Gibt es bei Ihnen viele verschiedene Rollen? Oder existiert eine große Zahl interner Abteilungen, die alle Schnittstellen zur Produktentwicklung haben? Wie stark muss die agile Entwicklung in das Gesamtunternehmen und in klassische Unternehmensfunktionen eingebettet werden? Welche Möglichkeiten der Anpassung bestehen? Mit in diese Kategorie fällt die Frage nach räumlicher Nähe – oder Distanz: Bei einem unserer Einführungsprojekte hatte der Kunde Standorte in München, Paris sowie in Osteuropa.

Wie betreibt das Unternehmen seine Wertschöpfung?

Wie komplex sind die Produkte, die Ihr Unternehmen für seine Kunden entwickelt? Was ist die Grundlage der Wertschöpfung, und welche Aufgaben gibt es dabei?

Welche Unsicherheiten gibt es in der Produktentwicklung?

Muss Ihr Unternehmen von gewissen Unwägbarkeiten bei der Entwicklung neuer Produkte ausgehen? Handelt es sich eher um Weiterentwicklungen, die durch Routine zu bewältigen sind, oder geht es um stärker herausfordernde Neuentwicklungen? Wie hoch ist der Innovationsgrad? Sind vielleicht sogar Entwicklungsanstrengungen in einem bislang fachfremden Bereich nötig – vielleicht Produkte mit einem viel höheren Digitalisierungsgrad als bisher? Anders gesagt: Entwickeln Sie ein Elektroauto oder Steuerungssoftware für Komponenten darin, und Ihr Kunde kann nicht genau definieren, was er benötigt? Oder geht es nur um ein Facelift eines schon sehr erfolgreichen Modells?

Wie viele externe Stakeholder wirken auf das Unternehmen ein?

Auch die Rahmenbedingungen spielen für die potenzielle Komplexität in einem Unternehmen eine wichtige Rolle. Wie viele externe Stakeholder gibt es, die Sie einbeziehen müssen?

Wie komplex ist die Branche, welche regulativen Anforderungen existieren?

Umfangreiche regulative Rahmenbedingungen sind ebenfalls ein wichtiger Treiber für Komplexität im Unternehmen. Ein relativ hohes Regulierungsniveau findet sich etwa in Branchen wie Finance oder Automotive. In der Automobilindustrie beispielsweise gibt es mit A-SPICE ein obligatorisches Vorgehensmodell, strikte Anforderungen an Functional Safety und die zu dokumentierende Qualität sowie klare Nachweispflichten, etwa für Audits.

Welches Skalierungsframework für welche Situation?

Spotify oder Scaling Cycle – geringe Komplexität in mittelständischen Unternehmen

Einen Ansatz wie den Agile Scaling Cycle können Sie grundsätzlich unabhängig von der Größe der Entwicklungsorganisation einsetzen – er selbst definiert noch keine feste Struktur und keinen Prozess. Prinzipiell ermöglicht es der sehr offene und unkonkrete Agile Scaling Cycle durchaus, auch für größere Unternehmen funktionierende Lösungen zu entwickeln. Mit zunehmender Komplexität der Gesamtunternehmung, des Produkts und des

Markts haben allerdings diejenigen Ansätze Vorteile, die bereits konkrete Lösungen liefern. Dies gilt zumal dann, wenn die Organisation wenig Erfahrung mit agilem Vorgehen hat.

Eine hohe Komplexität induziert eine Vielzahl an Rollen im Unternehmen, an internen und externen Stakeholdern. Auch etwaige Reglementierungen erfordern konkrete Lösungen. Das Praxisbeispiel von Spotify definiert zwar eine für mittelgroße Unternehmen durchaus geeignete Aufbauorganisation, aber dennoch bringt die Spotify Engineering Culture nur wenig prozessuale Unterstützung mit.

Beispiel: Spotify beim Mittelständler

Die fiktive Ausgangslage: Ein Mittelständler mit 500 Mitarbeitern entwickelt und betreibt eine Webplattform für private Endkunden. Einbezogen sind rund 200 Entwickler, Ingenieure, Qualitätsmanager und Designer. Diese Mitarbeiter sind in etwa 20 Teams organisiert. Schnittstellen zur Entwicklungsorganisation gibt es in den Bereichen Finance und Marketing. Das Produkt selbst ist nicht sonderlich komplex und keinen Regulationen unterworfen. Das Hauptproblem besteht also darin, die Entwicklungsteams zu skalieren. Vor diesem Hintergrund könnte sich das mittelständische Unternehmen recht gut für die Spotify Engineering Culture als agiles Skalierungsframework entscheiden.

LeSS – mittlere Komplexität und nicht zu große Entwicklungsorganisationen

LeSS definiert ein einfaches, kaum von Scrum abweichendes Vorgehen und eine skalierbare Struktur. Der Fokus auf eine Hierarchie aus Chief Product Owner, Area Product Owner sowie Product Owner auf Teamebene wird jedoch ab einer gewissen Größenordnung an Grenzen stoßen: Der Kommunikations- und Koordinationsbedarf wird in dieser Struktur stark ansteigen.

LeSS betont, dass der Schnitt der Anforderungen wie auch der Teams entlang kundenrelevanter Features entscheidend ist – und nicht entlang technischer Module oder Schichten. Die dadurch angestrebte Unabhängigkeit der Anforderungen untereinander und der Teams voneinander ist ab einer gewissen Größe aber nicht mehr ausreichend, um den Koordinationsbedarf zu decken. Product Owner auf den unteren Hierarchie-Ebenen tragen kaum noch Produktverantwortung.

Beispiel: LeSS im großen Verlagshaus

In unserem zweiten fiktiven Beispiel beschäftigen wir uns mit einem großen Verlagshaus – ein Konzern mit 10.000 Mitarbeitern. Dieses Medienunternehmen hat viele unabhängige Produkte. Für jeden seiner Titel entwickelt und betreibt der Verlag eine eigene CMS-Lösung. Es handelt sich um eine unregulierte Branche. Die Produkte mögen inhaltlich komplex sein, aber technisch sind sie nicht übermäßig komplex oder innovativ. Zur Entwicklungsorganisation gehören mehrere Dutzend Teams, die aus Entwicklern, Ingenieuren, Qualitätsmanagern und Designern bestehen. Hier könnte LeSS ein geeignetes Framework darstellen. Denn die zentrale Herausforderung für den Verlag besteht in der Skalierung der Entwicklungsteams selbst.

Eine fachliche und funktionale Trennung in möglichst unabhängige Teilprodukte wird hier schon den Großteil des Koordinationsbedarfs zwischen den Teams decken können. Die Stärke von LeSS in solch einer Konstellation: es ist leichtgewichtiger als das mächtige SAFe. Aber wir bewegen uns hier durchaus an der Grenze. Denn sobald es

zu einem hohen fachlichen Abstimmungsaufwand mit vielen verschiedenen Fachbereichen kommt – wenn die Komplexität also steigt –, bietet LeSS dafür nur noch wenig konkrete Lösungen.

SAFe – hohe Komplexität und große Entwicklungsorganisationen

SAFe ist in seiner definierten Form für kleinere Unternehmen wohl meist zu umfangreich und mächtig. Es bringt aber eine Vielzahl konkreterer Lösungen auf prozessualer Ebene und eine skalierungsfähige Aufbauorganisation mit, die eine hohe Zahl an Teams koordinieren kann und verschiedene Stakeholder und klassische Unternehmensbereiche einzubinden vermag. Für Konzerne dürfte SAFe als zugleich leistungsfähiges, wenngleich auch selbst recht komplexes Skalierungswerkzeug oft die erste Wahl darstellen. Schon bei der Abbildung externer, regulatorischer Anforderungen – von Vorgehensmodellen über Qualitätsdokumentationen bis hin zu Nachweisen für Audits – bietet SAFe umfassende Unterstützung. Die Projekterfahrung zeigt aber, dass sich auf Basis von SAFe auch gut einfachere Strukturen für mittelgroße Organisationen entwickeln lassen.

Beispiel: SAFe im Konzern

In unserem dritten fiktiven Fallbeispiel geht es um einen großen Automobilzulieferer – ein Konzern mit mehr als 10.000 Mitarbeitern. Zum Portfolio gehören verschiedene, weitgehend unabhängige Produkte. Allerdings sind diese Produkte ihrerseits jeweils durch eine hohe technische Komplexität und einen sehr hohen Innovationsgrad gekennzeichnet. Zur Entwicklungsorganisation des Automobilzulieferers zählen vielleicht 15 oder auch weit mehr Teams aus Entwicklern, Ingenieuren, Qualitätsmanagern und Designern.

Angesichts der Komplexität der Entwicklungsaufgaben und der Größe der Entwicklungsorganisation wäre hier SAFe eine nachvollziehbare Wahl. Gerade wenn es um komplexe Produkte geht, für die gegebenenfalls Hard- und Software parallel entwickelt werden müssen, hat SAFe dafür Lösungsvorschläge. Der Agile Release Train in SAFe bietet eine Organisationsstruktur, die es ermöglicht, auch viele Teams, die am selben Produkt arbeiten, miteinander zu koordinieren.

Zudem wird SAFe den Bedingungen eines regulierten Branchenumfelds besser gerecht als die anderen Frameworks. So bietet es mehr Möglichkeiten, Rollen für die Einhaltung von Vorgaben beispielsweise für Functional Safety zu integrieren, etwa analog zur Integration von Querschnittsfunktionen wie Architektur und UX in SAFe.

Fazit

Auch wenn sich alle hier diskutierten Frameworks in Ansatz, Detailgrad und Vollständigkeit unterscheiden, basieren sie doch alle auf gemeinsamen, grundlegenden Prinzipien – und sie haben alle ihre Berechtigung. Welches Framework für eine bestimmte Aufgabenstellung geeignet ist und den konkreten Bedarf Ihrer Organisation für die Skalierung agiler Verfahren am besten erfüllt, hängt von etlichen Faktoren ab. Mitunter können sich auch komplexe Frameworks als besonders geeignet erweisen: Sie geben einen Rahmen vor und bieten so gute Orientierung, lassen aber durchaus viel Raum für Anpassungen.

Wenn Sie bislang in Ihrer Organisation wenig Erfahrung mit Agilität haben, fällt der Einstieg mit einem sehr konkreten Framework oftmals leichter: Fest definierte Vorgehensweisen, Artefakte und Entscheidungspunkte erlauben es Ihnen, nach einem Kurztraining schnell "loszulaufen" und durch ein begleitendes Coaching nachzusteu-

ern. So vermeiden Sie typische Fallen zu Beginn. Hat die Organisation die Arbeit im Framework dann gelernt und verinnerlicht, können Sie freier werden, anpassen oder ggf. sogar ein weniger starres Framework wählen.

Fest steht: Märkte und Kundenanforderungen verändern sich in immer kürzeren Zeiträumen. Mit dem geeigneten agilen Skalierungsrahmen können Sie auch in einer großen Organisation die Flexibilität gewinnen, darauf mit der nötigen Geschwindigkeit zu reagieren.

Agil Skalieren

Koordination agiler Entwicklungsteams

In vielen Unternehmen, die sich mit der Entwicklung von Produkten beschäftigen, arbeiten heutzutage Teams mit agilen Methoden. Häufig benutzen diese Teams Scrum. Scrum gibt mit der Definition von Rollen, Artefakten und Ereignissen einen methodischen Rahmen, mit dessen Hilfe Teams ihre Arbeit agil planen und koordinieren können (siehe hierzu auch [Scrum – eine Einführung](#), Projekt Magazin 21/2009)

Beispiel: Bei dem Unternehmen Robota arbeitet eine Produktabteilung an der Entwicklung eines Roboters, der Pakete in der Umgebung ausliefern kann. Zur Abteilung gehören sechs Teams, die jeweils aus Hardware- und Softwarespezialisten zusammengesetzt sind. Die sechs Teams verwenden seit einem Jahr erfolgreich Scrum und bauen Stück für Stück die Funktionalität des Roboters aus.

Agile Frameworks wie z.B. Scrum sind auf die Teamarbeit zugeschnitten (ca. drei bis neun Personen). Wenn in einem Unternehmen mehrere agile Teams zusammen an einer größeren Produktentwicklung arbeiten, stellt sich die Frage, wie diese gemeinsame Arbeit ebenfalls agil koordiniert werden kann.

Scaled Agile: Agile Zusammenarbeit im großen Maßstab

Eine solche teamübergreifende Zusammenarbeit sollte auch auf den agilen Prinzipien und Techniken aufbauen, damit die Koordination im Großen zu Werten, Kultur und Haltung der Teamarbeit passt. Solche Herangehensweisen mit entsprechenden Rollen, Artefakten und Ereignissen für mehrere agile Teams nennt man "Scaled Agile" oder auch "Agilität im Großen".

In vielen Fällen liegt über den agilen Teams eine historisch gewachsene, hierarchische Managementstruktur. Hier gibt es oft Reibungspunkte, weil zwischen den agilen Teams und dem Management die Vorgehensweisen und Erwartungshaltungen bezogen auf Kommunikation, Reporting, Koordination und auch Entscheidungsfindung nicht zusammenpassen. Diese Hürde gilt es bei der Einführung von agilen Techniken auf der übergeordneten Ebene zu überwinden.

Beispiel: Bei Robota funktioniert das agile Arbeiten in den Teams gut, während es in der Zusammenarbeit der Teams große Probleme gibt. Gemeinsame Probleme der Teams wie eine instabile Entwicklungsumgebung, hohe Testaufwände und fehlende Integration während der Sprints werden zurzeit nicht systematisch adressiert. Den Teams fehlen eine gemeinsame Vision und eine Abstimmung untereinander. Das höhere Management gibt Entwicklungspläne vor, die angesichts des tatsächlichen Entwicklungsfortschritts nicht realistisch sind. Da die

Autoren



Claudia Raak

Geschäftsführerin der wibas GmbH, agiler Coach und Beraterin

sowie Buchautorin

Kontakt: claudia.raak@wibas.de



Malte Foegen

Geschäftsführer der wibas GmbH, Scrum Trainer, Lehrbeauftragter an der

Technischen Universität Darmstadt und der Frankfurt School, Buchautor

Kontakt: malte.foegen@wibas.de

Mehr Informationen unter:

› projektmagazin.de/autoren

Teams mit Scrum gute Erfahrungen gemacht haben, schlagen sie vor, ähnliche Techniken auch zur Koordination der Teams untereinander und zum Management der Anforderungen über alle Teams hinweg zu etablieren.

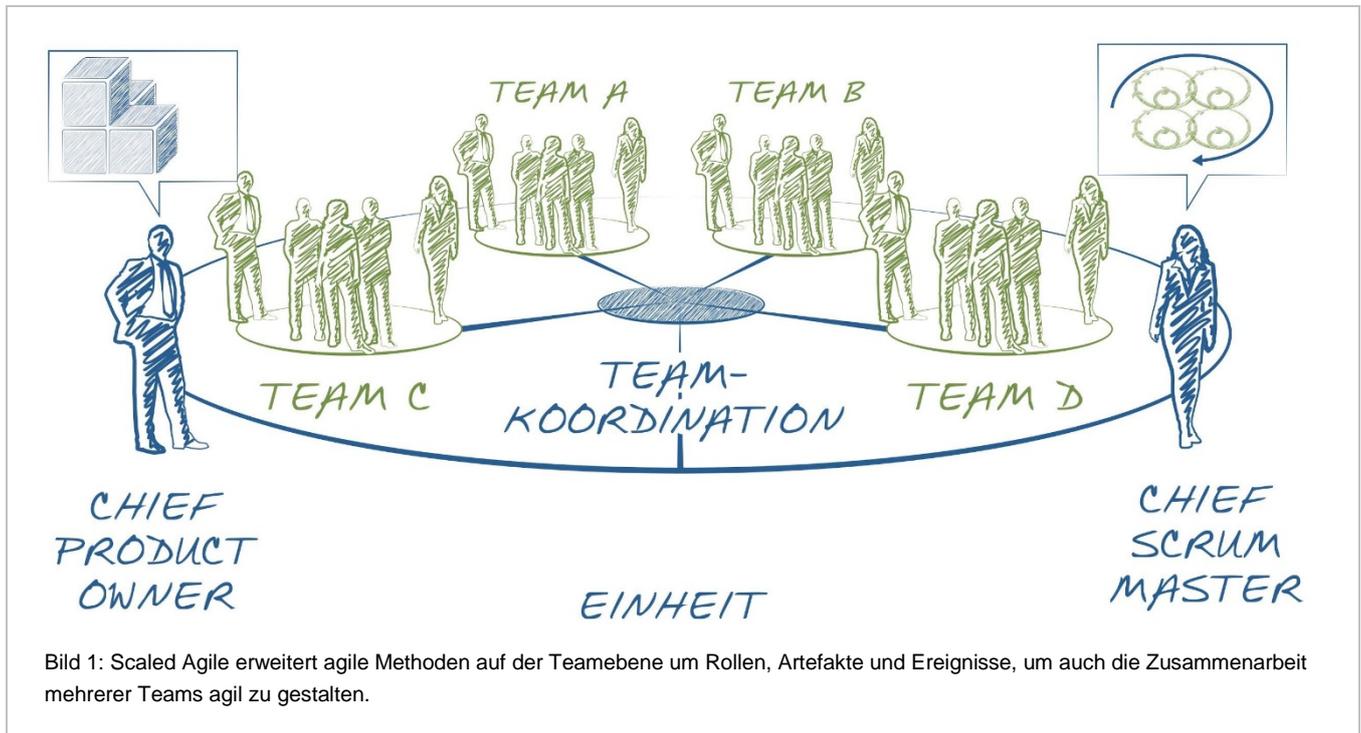
Es beginnt mit der Gewaltenteilung

Ein wichtiges Element von Scrum ist die Gewaltenteilung. Statt eines Projektleiters definiert Scrum drei sorgfältig ausbalancierte Rollen. So ist der Product Owner für ein erfolgreiches Produkt, das Entwicklungsteam für funktionierende Produktinkremente und der Scrum Master für wirksames Arbeiten verantwortlich. Der Vorteil der Gewaltenteilung: Durch die Balance kommt es zu austarierten Entscheidungen, die dem Unternehmen als Ganzes wirtschaftlich besser nutzen. Bei "Scaled Agile" setzen Unternehmen deshalb die Gewaltenteilung auch auf einer höheren Ebene fort, z.B. der Abteilungsebene. Statt eines Abteilungsleiters oder Programm-Managers gibt es einen Chief Product Owner und einen Chief Scrum Master - und natürlich die Teams.

Der Chief Product Owner (CPO) ist für alle Anforderungen eines Produkts verantwortlich. Bei kleinen Einheiten (z.B. mit nur drei Teams) kann er diese Arbeit alleine durchführen - es gibt dann nur einen (Chief) Product Owner für alle Teams. Bei größeren Einheiten (z.B. mit sechs Teams) ist diese Arbeit in der Regel zu umfangreich für eine Person. In diesem Fall hat häufig jedes Team einen "eigenen" Product Owner. Diese arbeiten mit dem Chief Product Owner als Team zusammen. Sie priorisieren die Anforderungen an das Produkt und stellen sicher, dass die Teams immer an den Anforderungen mit der höchsten Priorität arbeiten. Ebenso sorgen sie dafür, dass das Product Backlog zusammen mit Stakeholdern kontinuierlich detailliert und gepflegt wird (Product Backlog Refinements) und jeweils die wichtigsten Anforderungen enthält.

Der Chief Scrum Master (CSM) ist für den Gesamtprozess verantwortlich. Er moderiert insbesondere alle teamübergreifenden Ereignisse wie z.B. eine gemeinsame Planung. Mehrere Scrum Master unterstützen die Entwicklungsteams. Darüber hinaus arbeiten sie zusammen mit dem Chief Scrum Master in einem Team. Gemeinsam identifizieren und beseitigen sie z.B. teamübergreifende Behinderungen, strukturieren den Gesamtprozess (z.B. gemeinsamer Takt), moderieren "große" Ereignisse mit vielen Teilnehmern und bringen die Organisation immer weiter in Richtung einer agilen Organisation voran.

Die Entwicklungsteams schätzen den Umfang der Anforderungen und bestimmen, wie viel sie zusammen in einem Sprint umsetzen können ("Pull"). Sie planen die Sprints. Sie entwickeln, integrieren und liefern das gemeinsame Produktinkrement.



Beispiel: Bei Robota entscheiden die Teams, dass es für jedes Team einen eigenen Product Owner gibt. Die Produktmanagerin Sabine übernimmt die Rolle des Chief Product Owners. Das Product Owner Team beschließt, auf die gleiche Etage zu den Teams zu ziehen. So können sie eng und regelmäßig mit den Teams zusammenarbeiten. Der bisherige Abteilungsleiter Frank entscheidet sich für die Rolle des Chief Scrum Masters. Er hat bisher schon versucht, das Team als "dienende Führungskraft" zu führen und ist froh, sich auf diese Aufgabe konzentrieren zu können (und die Berichte loszuwerden). Auch er zieht in die gleiche Etage wie die Teams. In der Mitte der Etage gibt es einen großen Konferenz- und Kaffeebereich. Diesen richtet Frank als erste Maßnahme so ein, dass sich hier einzelne Teams bis hin zu allen Teammitglieder jederzeit ohne Probleme treffen können. "Community Zone" nennt er das.

Ein gemeinsamer Takt macht die Koordination einfacher

Ein gemeinsamer Takt der Teams vereinfacht die Koordination wesentlich. Deshalb arbeiten alle Teams in synchronen Sprints. Das bedeutet, dass die Sprints der Teams alle zum gleichen Zeitpunkt starten und enden. Der gemeinsame Takt ist die Grundlage dafür, dass Ereignisse je nach Bedarf der Teams gemeinsam durchgeführt werden können. So kann z.B. die Planung am Sprintanfang oder das Review am Ende gemeinsam erfolgen. Alle Scaled Agile Lösungen benutzen daher synchrone Sprints.

Um längerfristig planen zu können sind neben der Planung des nächsten Sprints Prognosen über mehrere Sprints hinweg sinnvoll. Hierbei wird für mehrere Sprints prognostiziert, welche Stories "drankommen". Im klassischen Scrum ist dies die Releaseplanung. Um die Komplexität im Griff zu behalten hat es sich bewährt, eine solche längerfristige Planung auch im Takt zu machen – z.B. alle sechs Sprints. Wir sprechen dann von einer Etappe. Sie ist gewissermaßen ein großer Scrum-Zyklus, der mehrere Sprints umfasst.

Die richtige Flughöhe: Epic – Feature – Story

Bei komplexen Produkten mit vielen Teams gibt es sehr umfangreiche Anforderungen. Um trotzdem das große Ganze im Blick zu behalten, hat es sich bewährt, die Anforderungen auf unterschiedlichen Abstraktionsebenen ("Flughöhen") zu betrachten und schrittweise zu detaillieren.

Anforderungen, die in einem Sprint "passen", werden Stories genannt. Passen sie in ca. zwei bis acht Sprints spricht man von Features. Alles was größer ist bezeichnet man als Epic (wegen ihrer epischen Breite). Anforderungen werden erst detailliert, wenn sie benötigt werden. Ein Epic wird erst dann in Features heruntergebrochen, wenn mit der Arbeit daran begonnen wird ("just in time"). Die Features werden wiederum erst dann in Stories detailliert, wenn die Arbeit daran beginnt. Durch die Abstraktionsebenen und das schrittweise Detaillieren bleiben auch bei komplexen Produkten die Anforderungen handhabbar.

Beispiel: Bei Robota ist allen Teams klar, dass sie einen gemeinsamen Takt benötigen. Deshalb setzen sich Chief Scrum Master, Chief Product Owner, einige Scrum Master und Vertreter der Teams in der neu eingerichteten "Community Zone" zusammen, um den gemeinsamen Takt zu bestimmen. Alle einigen sich schnell auf eine Sprintlänge von zwei Wochen. Diskussionen gibt es darüber, ob eine Etappe sinnvoll ist. Sabine als CPO meint aber, dass bei der Komplexität des Produkts eine regelmäßige gemeinsame Planung auf der nächst höheren Flughöhe von Features wichtig ist. Außerdem würde eine Etappenplanung den Teams regelmäßig eine gemeinsame Ausrichtung geben. Nach einigen Diskussionen einigen sich alle darauf, die Etappenplanung auszuprobieren. Als Etappenlänge legen sie vier Sprints (also acht Wochen) fest.

Die gemeinsame Planung: Vom Push zum Pull

Eine Etappe beginnt mit einer gemeinsamen Planung. Hierbei sind anwesend: alle Teammitglieder, alle Product Owner und alle Scrum Master. Dadurch sind schnelle Entscheidungen möglich. Außerdem entsteht dadurch ein gemeinsames Bild der Etappe, das sonst nur aufwendig transportiert werden kann. Bewährt hat sich auch, hierzu Stakeholder wie Kunden, höheres Management und andere Bereiche einzuladen, so dass sie zu Beginn und zum Abschluss der Planung dabei sind.

Die Etappenplanung dauert in der Regel ein bis zwei Tage. Hierfür bereitet das Product Owner Team die Anforderungen auf der Ebene von Features im gemeinsamen Product Backlog vor. Während der Etappenplanung nehmen die Teams aus dem Product Backlog die Features entsprechend ihrer Priorität heraus ("Pull"), übernehmen diese in ihr Team Product Backlog, brechen die Features in Stories herunter, und prognostizieren, in welchem Sprint welche Story umgesetzt wird.

Für eine gemeinsame Planung ist ein gut vorbereitetes, teamübergreifendes Product Backlog essentiell. Dieses teamübergreifende Product Backlog muss genügend Features für die nächste Etappe enthalten. Alle weiteren Anforderungen können grob als Epics formuliert sein. Der Chief Product Owner ist dafür verantwortlich, dass das Product Backlog ausreichend detailliert ist und dass die Umsetzungscomplexität der Features durch Vertreter der

Teams geschätzt wurde. Außerdem sorgt er dafür, dass es eine "Definition of Done" gibt, die spezifiziert, wann ein Feature von einem Team als "fertig" deklariert werden kann.

Beispiel: Frank eröffnet als CSM die Etappenplanung und stellt den Ablauf der beiden Tage vor. Sabine stellt als CPO die Anforderungen (Features) für die nächste Etappe vor. Die Features wurden vorher durch Vertreter der Teams mit Feature Points geschätzt. Damit sie für alle sichtbar sind, hat Sabine die Anforderungen an einer Wand aufgehängt. Dieser erste Teil der Agenda findet im Plenum mit allen Teams und Stakeholdern statt.

Dann gehen die Teams zusammen mit ihren Product Ownern und Scrum Masters in eigenen Arbeitsbereichen in die Teamarbeit. Die Teams übernehmen jeweils das nächste Feature aus dem gemeinsamen Product Backlog in ihr Team Product Backlog. (Deshalb hat Sabine Karten benutzt - diese können jetzt einfach umgehängt werden.) Die Teams brechen die Features zusammen mit ihrem Product Ownern in Stories herunter. Dabei planen sie die Stories grob in die Sprints der kommenden Etappe ein. Nachdem sie mit einem Feature fertig sind, nehmen sie das nächste aus dem gemeinsamen Product Backlog und verfahren damit genauso. Dies tun sie solange, bis das Team das Gefühl hat, dass es nicht mehr in der Etappe umsetzen kann. Am Ende dieses Teils der Etappenplanung haben alle Teams ihre Team Product Backlogs gefüllt. Die von den Teams übernommenen Features sind die Prognose, was realistischweise in der Etappe umgesetzt werden kann.

Im Verlauf dieser Etappenplanung stimmen die Teams ihre Zwischenstände regelmäßig ab. Hierfür moderiert Frank als CSM stündliche Scrum-of-Scrums, in denen jeweils ein Vertreter aus jedem Team Behinderungen, Abhängigkeiten oder Zusammenarbeitspunkte einbringt und mitnimmt. Damit sehen alle Teams, wann welche Features fertig werden und wovon ggf. einzelne Features abhängig sind. So können die Teams Abhängigkeiten und gemeinsame Aktivitäten abstimmen und in ihre Planung einbeziehen.

Am Ende des Planungsereignisses finden sich nochmal alle zusammen, inklusive Stakeholder, und stellen sich gegenseitig die geplanten Features und Stories für die nächste Etappe vor.

Für den Erfolg der Etappenplanung sind geeignete Räumlichkeiten wichtig. Es wird in der Regel ein größerer Konferenzraum benötigt, in dem alle Teams, Scrum Master und Product Owner sowie alle anwesenden

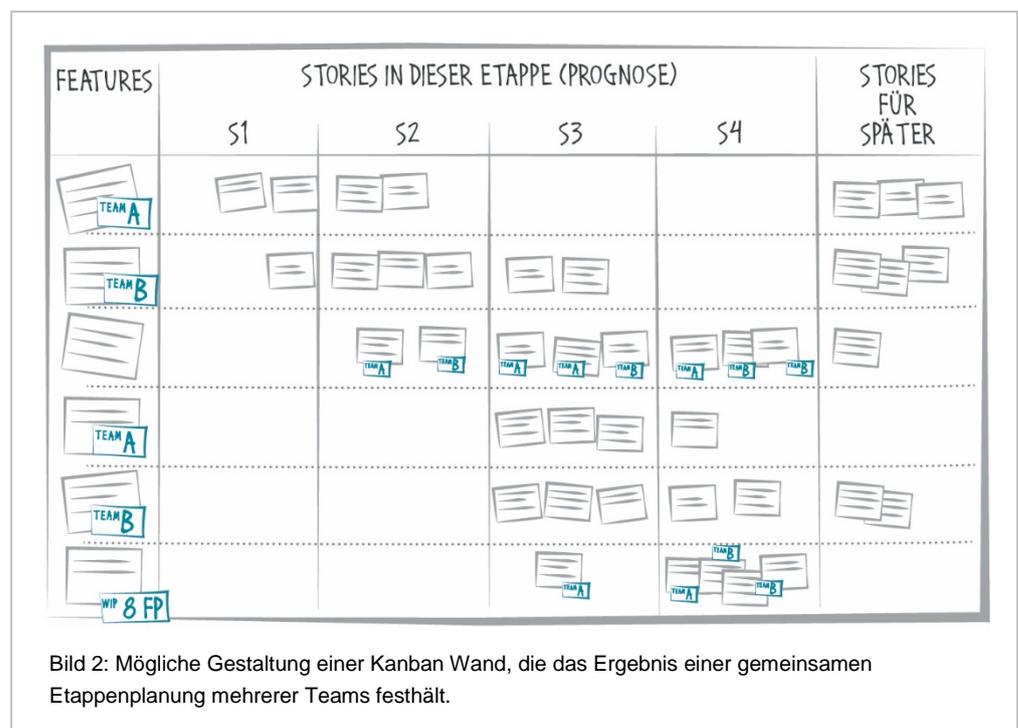


Bild 2: Mögliche Gestaltung einer Kanban Wand, die das Ergebnis einer gemeinsamen Etappenplanung mehrerer Teams festhält.

Stakeholder ausreichend Platz haben und der gemeinsame Start, das gemeinsame Ende sowie die Scrum-of-Scrums stattfinden können. Dann benötigt jedes Team einen Raum, in dem das Team mit seinem Scrum Master und Product Owner ungestört arbeiten kann. Wenn ein Teil der Teams offsite arbeitet, sollten entsprechende Videokonferenz-Möglichkeiten geschaffen werden, damit die Teams an allen gemeinsamen Aktivitäten teilnehmen können.

Im Sprint koordinieren

Nachdem die Etappe gemeinsam geplant wurde, starten die Teams wie gewohnt jeden Sprint mit einer Sprintplanung. Im ersten Teil der Sprintplanung (Bestimmung des Arbeitsumfangs des nächsten Sprints) erfolgt dabei meistens gemeinsam mit allen Teams, um eine schnelle "Wer-macht-was" Abstimmung zu erreichen. Die Detailplanung macht dann jedes Team für sich. Während der Sprints muss vor allem die Koordination der Teams untereinander sichergestellt werden. Hierfür findet typischerweise ein sogenanntes tägliches "Scrum-of-Scrums" statt (SoS) – im Anschluss an das Daily Scrum der Teams. Daran nimmt aus jedem Team jeweils ein Vertreter (nicht Scrum Master) teil. In dem SoS werden teamübergreifende Behinderungen identifiziert und Kooperationspunkte zwischen den Teams vereinbart. Da Punkte nicht gelöst, sondern nur eingebracht und Lösungen koordiniert werden, sollte das SoS nicht länger als 15 Minuten dauern. Als Arbeitsmittel wird oft ein Kanban Board (das sogenannte SoS-Board) verwendet, um Behinderungen und Kooperationspunkte zu verfolgen. (Für weitere Informationen zu Kanban und Kanban-Boards siehe auch: [Den Output des Teams mit Kanban optimieren](#), Projekt Magazin, 21/2012)

Die Scrum Master arbeiten als Team gemeinsam an übergreifenden Behinderungen. Hierzu moderiert der CSM häufig ein regelmäßiges Daily Scrum für das Scrum Master Team. Die Scrum Master können für die anstehenden Punkte eine Kanban Wand benutzen, um die Bearbeitung der Behinderungen zu visualisieren.

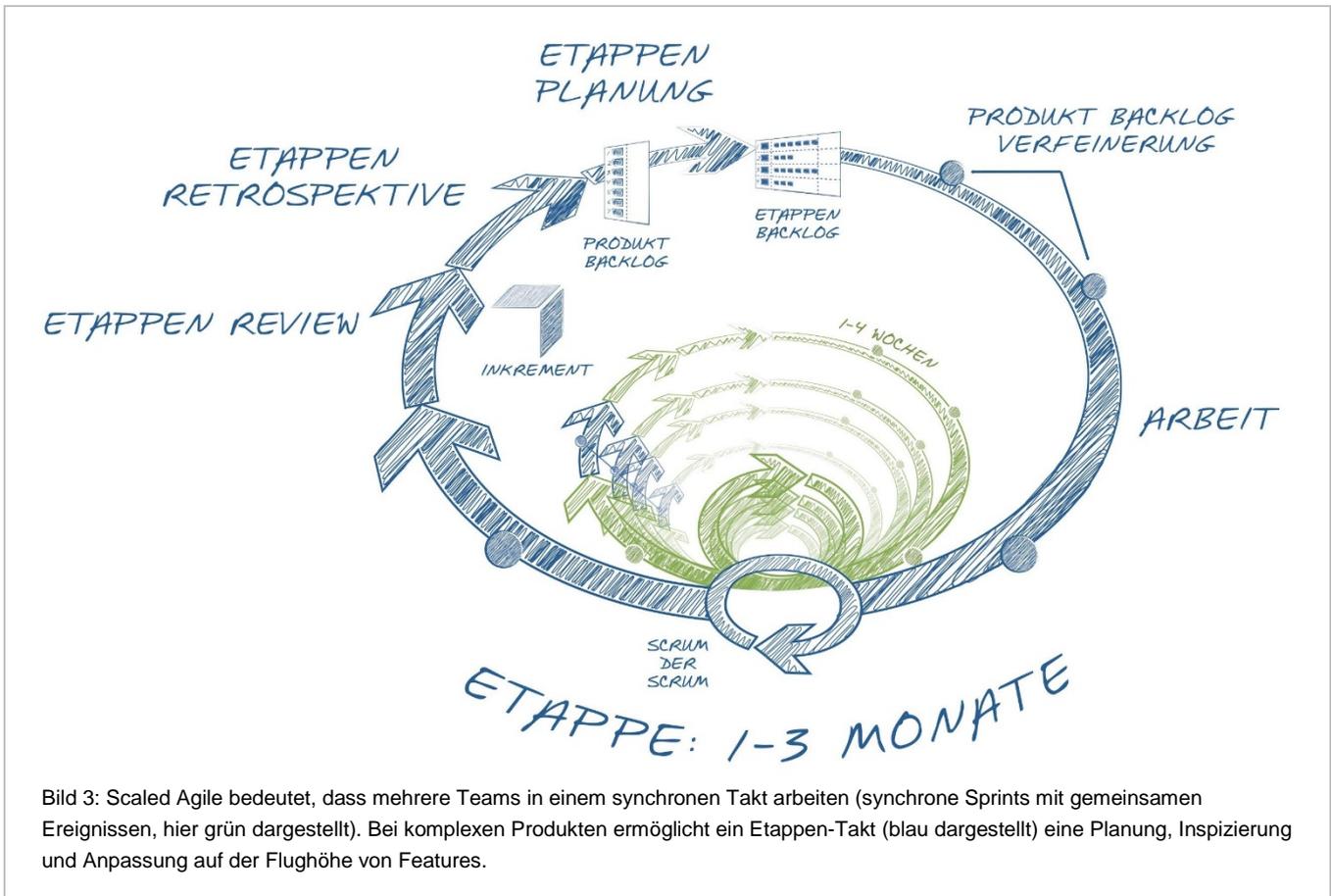
Beispiel: Bei Robota beginnt jedes Team seinen Tag mit einem Daily Scrum um 9 Uhr. Um 9.20 Uhr moderiert Frank als CSM in der Community Zone ein 15-minütiges Scrum-of-Scrums, zu dem die Teams jeweils einen Vertreter schicken.

Gemeinsam inspizieren und anpassen

Teams, die mit Scrum arbeiten, führen in jedem Sprint ein Review und eine Retrospektive durch, um sowohl das Produkt als auch die Arbeitsweise zu verbessern. Wenn mehrere Teams an einem integrierten Produkt arbeiten, ist es notwendig, dass die Teams gemeinsam auf das Produkt und die Arbeitsweise schauen und Verbesserungen identifizieren.

Um gemeinsam auf das Produkt zu schauen, wird jedes Sprint Review gemeinsam mit allen Teams durchgeführt. Dies ist eine logische Konsequenz daraus, dass die Teams ein integriertes potenziell lieferfähiges Ergebnis liefern. Ein Etappenreview am Ende einer Etappe ermöglicht es, auf der Flughöhe der Features das Produktinkrement mit Stakeholdern zu besprechen.

Um die Arbeitsweise zu betrachten, führen die Teams – neben der Retrospektive im Team – eine gemeinsame Retrospektive durch. Wie häufig diese stattfindet, hängt von den Umständen ab. Einige Organisationen führen diese jeden Sprint, manche nur am Ende jeder Etappe durch. Eine Retrospektive aller Teams ist aber wichtig, um das gemeinsame Vorgehen und die Zusammenarbeit und Koordination zwischen den Teams zu verbessern.



Beispiel: Bei Robota haben sie sich entschieden, am Ende jedes Sprints ein gemeinsames Sprint Review durchzuführen. Dabei stellen Vertreter aus jedem Team den Stakeholdern das über alle Teams hinweg integrierte Produktinkrement vor. Weiterhin wurde aufgrund der Vielzahl der umgesetzten Anforderungen beschlossen, dass die Product Owner schon während des Sprints die umgesetzten Anforderungen abnehmen. Sabine als CPO findet das auch deshalb gut, weil so zurückgewiesene Stories im selben Sprint noch korrigiert und fertiggestellt werden können. Außerdem können sie sich im Sprint Review darauf konzentrieren, das Feedback der Stakeholder (bei Robota sind dies Pilotkunden) einzuholen. Das gemeinsame Sprint Review wird durch Frank als CSM moderiert.

Im Anschluss an das Sprint Review führen die Teams zunächst eine eigene Retrospektive, danach eine gemeinsame Retrospektive durch. Franz lädt dazu alle in die Community Zone ein, wo jedes Team (mit Product Owner und Scrum Master) seine wichtigsten Erkenntnisse einbringt. Sie stellen fest, dass sie bei der Entwicklungsgeschwindigkeit eine regelmäßige Integration über alle Teams hinweg und mehr Testautomatisierung benötigen. Sabine und die Teams schreiben dazu ein Feature mit Stories. Sabine als CPO tauscht das neue Verbesserungs-Feature gegen ein anderes im Product Backlog aus, und die Teams nehmen das Feature und die Stories mit in die Sprint Planung. Außerdem einigen sich alle darauf, dass die Integration und die Abnahme durch den Product Owner in die "Definition of Done" aufgenommen wird. Bei der Retrospektive haben die Teams auch ihre Velocity (gelieferte Story-Points pro Sprint) betrachtet und festgestellt, dass sie langsamer als gedacht vorankommen. Sabine notiert schon mal im Kopf, dass in der Etappe wohl weniger Features umgesetzt werden als in der Etappenplanung gedacht. Am Etappen Burndown

schätzt sie ab, dass die letzten zwei Features wohl herausfallen werden. Auch die Teams nehmen die Information in ihre nächste Sprint Planung mit, um die Anzahl der Stories in den nächsten Sprints realistisch anzupassen.

SAFe® oder LeSS – was ist der richtige Skalierungsansatz für mich?

Bei agiler Skalierung haben sich zwei Ansätze etabliert: das Scaled Agile Framework (SAFe) und Large Scale Scrum (LeSS).

LeSS skaliert Scrum, indem es einen Sprint mit mehreren Teams durchführt – im Robota-Beispiel ist dies der gemeinsame Sprint, den die Teams durchführen. LeSS hält sich auch bei mehreren Teams konsequent an Scrum, beschränkt sich auf die Produktentwicklung und ist weniger kompliziert als SAFe. LeSS ist einfach zu verstehen, wenn man Scrum verstanden hat. Wir finden LeSS insbesondere dann sinnvoll, wenn bis zu vier Teams gemeinsam ein Produkt liefern. LeSS ist auch ein guter Start, um mit agiler Skalierung zu beginnen.

SAFe bietet insbesondere Lösungen für sehr große Organisationen. SAFe fügt dem gemeinsamen Sprint eine Etappe hinzu und handhabt Anforderungen auf mehreren Flughöhen (Epic, Feature, Story). Die Etappenplanung bei Robota entstammt SAFe (und wird dort auf Englisch "Program Increment Planning" genannt). Außerdem adressiert SAFe eine agile Projekt-Portfolioplanung und bindet auch Kanban mit ein. Damit ist SAFe eher für komplizierte Organisationen geeignet – aber auch deutlich schwieriger zu verstehen als LeSS. SAFe braucht erfahrene Chief Scrum Master (in SAFe "Release Train Engineer" genannt) und Coaches, um es aufzusetzen.

Welches der Frameworks man wählt – oder ob man Elemente aus verschiedenen Frameworks verwendet wie in unserem Beispiel – hängt von der eigenen Situation ab. Alle Frameworks müssen auf die eigene Situation angepasst werden. Deshalb sprechen wir häufig bei Dingen wie "gemeinsamer Sprint" oder "Etappenplanung" auch von "Mustern". Wichtiger als die Framework-Auswahl ist das Verständnis der agilen Prinzipien. Denn nur mit diesen kommt man zu einer agilen Organisation im Großen – statt zu einem "agilen Theater" im Großen.

Fazit

Scaled Agile schließt die Lücke bei großen Einheiten, die aus mehreren Teams bestehen und die Zusammenarbeit nach agilen Prinzipien gestalten wollen. Eines der wichtigsten Merkmale einer Scaled Agile- Lösung ist, dass auch auf der nächsten Ebene von einem Push-Denken (zum Termin x muss y fertig sein) zu einem Pull-Denken gewechselt wird (die Teams nehmen aus der priorisierten Anforderungsliste das, was in einem Takt umsetzbar ist). Auch die Gewaltenteilung auf Abteilungsebene ist ein weiteres wichtiges Element. Für die Umsetzung von Scaled Agile ist ein gut funktionierendes Scrum auf Teamebene Voraussetzung - im gemeinsamen Takt zeigen sich Schwächen sofort. Die Umsetzung von Agilität auch auf der nächst höheren Ebene hebt die Vorteile von Scrum im Großen. Es deckt Behinderungen, wie z. B. Integrationsschwierigkeiten oder fehlendes automatisiertes Testen frühzeitig auf. Das regelmäßige integrierte Produkt zeigt allen Stakeholdern einen realistischen Fortschritt. Insgesamt wird durch ein solches Vorgehen die Zusammenarbeit der Teams erheblich verbessert, die Geschwindigkeit erhöht und Verschwendung reduziert.

Literatur

- Foegen, Malte u.a: Der Ultimative Scrum Guide 2.0, wibas, 2014
- Foegen, Malte; Kaczmarek Christian: **Organisation in einer digitalen Zeit**, wibas, 2015
- Sutherland, Jeff: Scrum: The Art of Doing Twice the Work in Half the Time, Random House Business, 2015
- Scaled Agile Framework (SAFe), beschrieben in: Leffingwell, Dean: Agile Software Requirements, Addison-Wesley, 2010 und online unter: <http://scaledagileframework.com>
- Large Scale Scrum (LeSS), beschrieben in: Larman, Craig; Vodde, Bas: Scrum erfolgreich skalieren mit LeSS, dpunkt Verlag, 2017, und online unter: <http://less.works>
- Definition von Scrum: <https://scrumguides.org>
- Übersicht über Links und Literatur zu Scaled Agile: <https://www.wibas.com/de/modelle-konzepte/agile-lean/scaling-agile-und-scaling-scrum-links/>
- Kombinierte Übersicht der Methoden: <http://scrumbrowser.com>

Teamzuständigkeiten in Scrum – von Komponententeams zu Feature Teams

Nach der Scrum-Philosophie besteht ein Software-Entwicklungsprojekt idealerweise aus einem einzigen Team mit sieben Personen, die alle im selben Raum arbeiten. In der Praxis sieht es häufig anders aus: Das Projektteam ist so groß, dass eine Unterteilung in mehrere (Sub-)Teams angebracht ist. Darüber hinaus sind die Projektteams nicht selten über verschiedene Standorte verteilt. Die Notwendigkeit für große und verteilte Projektteams ergibt sich aus dem Umfang und der Komplexität der Anforderungen, die in einem engen Zeitrahmen realisiert werden müssen und der Frage, wo im Unternehmensverbund Ressourcen mit passenden Qualifikationen verfügbar sind.

Um die Verantwortlichkeiten der einzelnen Teams festzulegen, existieren verschiedene Ansätze. Im vorliegenden Artikel beschreiben wir unsere Erfahrungen mit zwei Teamformen: Komponententeams und Feature Teams. Unsere Erfahrungen beziehen sich auf mehrere nach Scrum durchgeführte Softwareprojekte mit jeweils bis zu fünf Teams in drei verschiedenen Ländern und einer Projektlaufzeit von bis zu zwei Jahren. Diese Projekte hatten zum Ziel, medizintechnische Anwendungen im Bereich der Strahlentherapie zu realisieren.

Erfahrungen mit Komponententeams

In den ersten Projekten bildeten wir Komponententeams. Dieser Ansatz erscheint logisch, da sich die Teamstruktur einfach an der Struktur der zu entwickelnden Applikation orientiert. Eine Applikation besteht aus einer Menge von interagierenden Komponenten, durch deren dynamisches Zusammenspiel die Anwendung so funktioniert, wie es vorgesehen ist. Bei diesem Ansatz übernimmt jedes Team die Verantwortung für eine oder mehrere Komponenten. Nur das jeweils zuständige Team darf Änderungen in einer Komponente vornehmen.

Diesem Ansatz folgend könnte es beispielsweise ein Datenbank-Team geben, das sich um die Definition des Datenbankschemas und die Programmierung der Datenzugriffsschicht kümmert, ein Geschäftslogik-Team, das die Implementierung der Geschäftslogik übernimmt und ein Front-end-Team, welches die Bedienoberfläche der Applikation bereitstellt.

Mehrere Teams koordinieren

Die meisten Features einer Applikation betreffen allerdings mehrere Komponenten, wobei es häufig vorkommt, dass diese Komponenten unterschiedlichen Teams zugeordnet sind. In diesen Fällen ist eine teamübergreifende Koordination erforderlich: Zunächst müssen sich die beteiligten Teams auf eine gemeinsame technische Lösung verständigen. Dies geschieht häufig durch die Mitwirkung eines übergeordneten Architekten. Als nächstes vereinbaren sie die Realisierungsreihenfolge und Bereitstellungstermine für die Komponenten. Und schließlich müssen geeignete Maßnahmen gefunden und abgestimmt werden, wenn sich die Zulieferung einzelner Komponenten

Autoren



Klaus-Dieter Schmatz

Teamleiter und Program
Manager bei Siemens
Healthcare, PMP, Certified

Scrum Master

Kontakt:

klaus-dieter.schmatz@siemens.com



Ivan Kostial

Seit 2002 bei Siemens
Healthcare tätig, Scrum
Master

Kontakt: ian.kostial@siemens.com

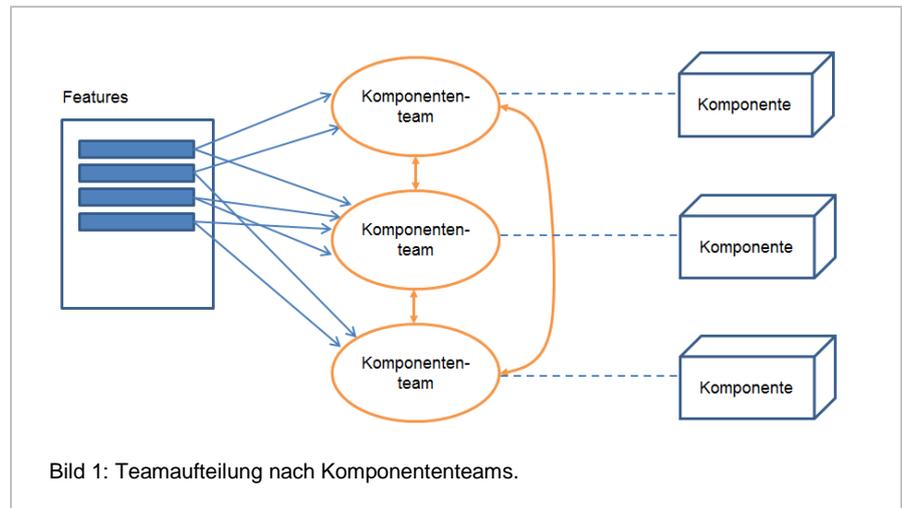
Mehr Informationen unter:

› projektmagazin.de/autoren

verspätet. Da es sich um teamübergreifende Aktivitäten handelt, ist der Projektleiter (oder ein Mitglied des Kernteams) bei der Planung und Steuerung involviert. Bild 1 veranschaulicht diesen Ansatz.

Beispiel

Ein einfaches Feature könnte sein, die Anwendung um ein zusätzliches Informationsfeld zu erweitern. Dafür muss das Datenbank-Team ein Attribut in einer Datenbanktabelle ergänzen und die Datenzugriffsschicht entsprechend erweitern. Das Frontend-Team implementiert den Code, um diese neue Funktionalität der Datenzugriffsschicht aufzurufen und die Daten im User Interface zu repräsentieren. Auch wenn diese Änderung trivial erscheinen mag, so müssen die dafür nötigen Arbeitspakete teamübergreifend geplant werden. Abweichungen vom abgestimmten Vorgehen, etwa durch eine verspätete Bereitstellung einer Schnittstelle, können mitunter drastische Auswirkungen haben: Wenn das Datenbank-Team die Erweiterung der Datenzugriffsschicht nicht fristgerecht abschließen kann, ist das Frontend-Team blockiert und die Fertigstellung des gesamten Features verzögert sich.



Die Projektstruktur überdenken

Generell folgen alle unsere Projekte einem agilen Vorgehensmodell. Die Projektlaufzeit ist in einzelne Iterationen unterteilt, die jeweils das Ziel haben, eine bestimmte Menge an Features in vollem Umfang – Design, Implementierung, Unit Test, Integrationstest und Dokumentation – zu realisieren. Die Iterationen der einzelnen Teams verlaufen synchron, d.h. der Beginn und die Dauer jeder Iteration sind für alle Teams einheitlich. Innerhalb dieses Rahmens machten wir bei den Projekten, die wir mit Komponententeams realisierten, die folgenden Beobachtungen, die uns im Lauf der Zeit zum Nachdenken brachten, ob es nicht einen besseren Ansatz gibt, das Projekt zu strukturieren:

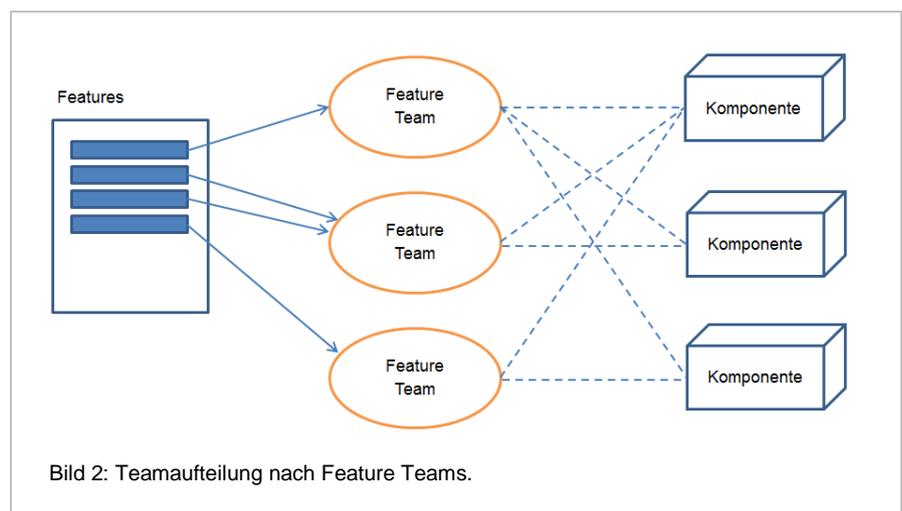
- Wir arbeiteten mit einer relativ langen Iterationsdauer, die je nach Projekt zwischen vier und sechs Wochen lag. Wir hätten uns kürzere Iterationen gewünscht, da der Product Owner dann die Möglichkeit gehabt hätte, schneller und häufiger Feedback zu geben. Eine Verkürzung der Iterationsdauer scheiterte jedoch einerseits daran, dass wir uns im Umfeld der Medizinprodukte bewegen, in dem es strenge Dokumentationsanforderungen gibt. Weiterhin konnten wir die Iterationsdauer nicht verringern, da sich sowohl die technischen Abstimmungen zwischen den Teams am Iterationsbeginn sowie die Integration der einzelnen Komponenten zum Iterationsende als zeitaufwändig erwiesen.
- Um dem hohen Koordinationsbedarf zwischen den Teams gerecht zu werden, richteten wir regelmäßig stattfindende Scrum-of-Scrum-Meetings mit Vertretern aus allen Teams ein. Ein häufiger Diskussionspunkt war dabei, dass fehlende Zulieferungen von anderen Teams die Weiterarbeit blockierten.

- Ein großer Anteil an teamübergreifenden Features konnte nicht innerhalb der ursprünglich geplanten Iteration fertiggestellt werden. In der Regel war zwar die Implementierung vorhanden, aber vielfach fehlten noch die automatisierten Tests und große Teile der Dokumentation. In den meisten Fällen ließ sich dieses Ergebnis darauf zurückführen, dass Schnittstellenimplementierungen erst kurz vor Iterationsende für andere Teams verfügbar wurden, sodass ihnen gerade noch Zeit für die darauf aufbauende Implementierung ihrer eigenen Komponenten blieb. Somit mussten wir diese Features am Iterationsende als unvollständig bewerten und für die nachfolgende Iteration erneut einplanen. Die durchschnittliche Durchlaufzeit für ein Feature betrug 2,5 Iterationen.
- Aufgrund der Erfahrungen mit verspäteten und unvollständigen Zulieferungen durch andere Teams waren die einzelnen Komponententeams verständlicherweise sehr zögerlich, ein echtes Commitment für die in einer Iteration geplanten Features abzugeben. Damit Scrum funktioniert, ist dieses Commitment jedoch sehr wichtig, denn es trägt dazu bei, die Projektmitarbeiter auf das Erreichen der Iterationsziele zu fokussieren.
- Der regelmäßig stattfindende gemeinsame Software-Build scheiterte mit signifikanter Häufigkeit vor allem in den letzten Tagen vor dem Iterationsende. Meist lag es daran, dass zu diesem Zeitpunkt viele Änderungen gleichzeitig abgegeben wurden und die Änderungen von übergreifenden Schnittstellen nicht mit allen betroffenen Teams abgestimmt waren. Die Beseitigung der Build-Probleme erforderte den hartnäckigen Einsatz eines Build Managers, den wir in Vollzeit im Projekt beschäftigten.

Bei der Planung der Nachfolgeprojekte beschlossen wir, Verbesserungen für diese Punkte zu erarbeiten. Inkrementelle Optimierungen erschienen uns als nicht ausreichend, sodass wir uns dafür entschieden, einen neuen Ansatz einzuführen, nämlich den der sog. "Feature Teams" (Larman, Vodde, 2010).

Feature Teams

In diesem Ansatz ist die Zuständigkeit der Teams anders definiert: Die Teams sind nicht mehr für einzelne Komponenten, sondern für ganze Features verantwortlich. Das für ein Feature verantwortliche Team führt sämtliche Änderungen, die zur Realisierung erforderlich sind, eigenständig durch, und zwar an allen betroffenen Komponenten der Applikation. Während einer Iteration bearbeiten die Teams verschiedene Features parallel zueinander (s. Bild 2).



Natürlich bringt diese Vorgehensweise nicht nur Vorteile, sondern auch gewisse Herausforderungen mit sich. Eine wesentliche Herausforderung ergibt sich beispielsweise daraus, dass für die Realisierung unterschiedlicher Features manchmal dieselben Komponenten bearbeitet werden müssen. Somit können sich die Teams bei der Arbeit auf diesen Komponenten gegenseitig in die Quere kommen. Dieses Problem lässt sich durch ständige, sehr häufige gemeinsame Software-Builds entschärfen.

Zu den Vorteilen der Feature Teams gegenüber Komponententeams zählen eine stark verringerte Notwendigkeit von teamübergreifender Planung und Koordination sowie reduzierte Implementierungsabhängigkeiten zwischen den Teams. Um die unterschiedliche Arbeitsweise der Komponententeams und Feature Teams zu verdeutlichen, betrachten wir drei Beispiele aus dem Entwicklungsalltag näher.

Beispiel 1: Spezifikationsänderung

Der Product Owner wünscht nach Gesprächen mit den Anwendern eine Änderung in einem bereits implementierten Feature.

- **Komponententeams:** Vertreter aller Teams treffen sich und besprechen die Auswirkungen der Änderung auf die Komponenten. In den nachfolgenden Iterationen werden die notwendigen Anpassungen in den betroffenen Komponenten implementiert, typischerweise sequentiell nacheinander: zunächst die Datenbankschicht, anschließend die Geschäftslogik und zum Abschluss die Bedienoberfläche. Die gemeinsame Planung kostet Zeit und Aufwand und die sequentielle Implementierung führt typischerweise zu einer langen Durchlaufzeit.
- **Feature Teams:** Ein Team übernimmt die Aufgabe, die Änderung zu implementieren. Das Team diskutiert die Implementierung intern und erledigt die Anpassungen selbstorganisierend. Die Durchlaufzeit wird nicht durch Warten auf Zulieferungen anderer Teams beeinflusst.

Beispiel 2: Dringende Fehlerkorrekturen

Die Anwendung ist im Einsatz und es wurden diverse Fehler gefunden, die möglichst rasch beseitigt werden sollen. Die Entwicklung neuer Features läuft parallel dazu weiter.

- **Komponententeams:** Das Entwicklungstempo (in Scrum-Terminologie Velocity) des Gesamtprojekts wird durch die Geschwindigkeit des langsamsten Komponententeams bestimmt. Ungeplante Tätigkeiten wie dringende Bug Fixes, die kurzfristig neben den geplanten Aufgaben durchgeführt werden müssen, können sich daher stark auf die Gesamtperformance auswirken. Wenn die Fehler alle Teams gleichmäßig betreffen, ist die Auswirkung am geringsten. Wenn allerdings hauptsächlich ein einzelnes Team betroffen ist (zum Beispiel das Front-end-Team, wenn besonders viele Fehler in der Bedienoberfläche gefunden wurden), dann leidet das Gesamtprojekt stark: Obwohl die anderen Teams normal weiterarbeiten können, ist es unmöglich, neue Features fertig zu entwickeln (weil keine Ressourcen für die Implementierung der Bedienoberfläche zur Verfügung stehen). Der Projektleiter muss geeignete strukturelle Gegenmaßnahmen ergreifen, z.B. zusätzliche Mitarbeiter in das Front-end-Team integrieren, um den Engpass aufzulösen.
- **Feature Teams:** Die Fehlerkorrekturen lassen sich gleichmäßig auf alle Teams verteilen. Eine andere Möglichkeit wäre, ein Team aus der laufenden Feature-Entwicklung abzuziehen und exklusiv mit dem Bug Fixing zu beschäftigen, während die verbleibenden Teams weiter an neuen Features arbeiten. In jedem Fall sinkt das Entwicklungstempo des Gesamtprojekts nur proportional zum Aufwand für die Fehlerkorrekturen.

Beispiel 3: Leistungsabfall in einem Team

Die Leistungsfähigkeit eines Teams nimmt plötzlich ab. Mögliche Ursachen sind beispielsweise eine Konzentration krankheitsbedingter Ausfälle oder die Abwerbung von Mitarbeitern durch Konkurrenten.

- **Komponententeams:** Dieses Szenario ist vergleichbar mit dem Fall der ungleichmäßig verteilten Fehlerkorrekturen im vorherigen Beispiel. Die Geschwindigkeit eines Teams bricht stark ein. Der Projektleiter muss schnell reagieren, um die Auswirkungen auf das Gesamtprojekt zu minimieren. Unüberlegte Workarounds wie die kurzfristige Hinzunahme von neuen Mitarbeitern erweisen sich allerdings häufig als kontraproduktiv, weil sie oftmals zu weiteren Störungen führen: Statt sich auf die Arbeit am Produkt zu konzentrieren, wenden die bestehenden Teammitglieder Zeit für die Einarbeitung der neuen Kollegen auf (bei denen man im Interesse der schnellen Verfügbarkeit oft auf eigentlich erforderliche Qualifikationen verzichtet), die ihrerseits erst nach mehreren Wochen oder Monaten einen produktiven Beitrag zum Projekt leisten können.
- **Feature Teams:** Wie im vorherigen Beispiel sind die Auswirkungen auf das Gesamtprojekt begrenzt. Die reduzierte Geschwindigkeit in einem Team stellt keine Behinderung für die anderen Teams dar. Der Projektleiter kann abwarten bis der Normalzustand wieder eintritt (bei temporären Störungen) oder er kann in relativer Ruhe passende Gegenmaßnahmen einleiten (z.B. Suche nach qualifiziertem Ersatz für Mitarbeiter, die das Unternehmen verlassen haben).

Der Veränderungsprozess

Aus jedem unserer Komponententeams sollte also zukünftig ein Feature Team werden. Auch wenn dabei die interne Zusammensetzung der einzelnen Teams unverändert bleiben sollte, war klar, dass die Team-Mitglieder die geplante Umstellung als schwerwiegend empfinden würden, da sich viele der gewohnten Abläufe und Zuständigkeiten ändern würden. Naturgemäß äußerten Teammitglieder verschiedene Vorbehalte gegen die neue Struktur, von denen die nachfolgenden am wichtigsten waren:

- Das erforderliche Know-how für die Realisierung komplexer Features ist über mehrere Teams verteilt. Kein einzelnes Team verfügt über alle Kompetenzen.
- Da die Entwickler nur über ein begrenztes Wissen über Komponenten verfügen, die sie nicht selbst entworfen und selten bearbeitet haben, können ihre Änderungen den Entwurfszielen widersprechen und unbeabsichtigte Nebeneffekte nach sich ziehen. Außerdem besteht die Gefahr, dass sich die Software-Architektur im Lauf der Zeit verschlechtert, wenn Entwickler ohne Kenntnis der ursprünglichen Entwurfsprinzipien Komponenten ändern oder neu hinzufügen.
- Wer kümmert sich um das Refactoring der Komponenten, wenn es keine dedizierten Komponenteneigentümer mehr gibt? Bisher hatte der Komponenteneigentümer das größte Interesse, seine Komponente laufend zu verbessern. Dieses Interesse erlischt, wenn er nicht mehr für die Komponente verantwortlich ist und alle anderen Entwickler Änderungen an der Komponente vornehmen dürfen.

Zwar hatten wir keine vorbereiteten Lösungen für diese Probleme parat, aber wir stellten sicher, dass alle Beteiligten sie im folgenden Veränderungsprozess gemeinsam erarbeiten sollten.

Den Veränderungsprozess gestalten

Wie haben wir den Veränderungsprozess gestaltet? Unser Ziel war es, die Sorgen der Team-Mitglieder abzubauen und die Grundlagen zu schaffen für eine effiziente Zusammenarbeit in dieser neuen Konstellation. Die folgenden Maßnahmen erwiesen sich dabei als die wichtigsten:

1. In unseren Weiterbildungsmeetings (regelmäßige Treffen mit einer Dauer von insgesamt zwei Stunden pro Monat) erläuterten wir allen Team-Mitgliedern die Grundlagen der Warteschlangentheorie, mit der sich die prinzipiellen Probleme des Komponententeam-Ansatzes erklären lassen. Wir besprachen, welche Arten von Warteschlangen in unserem Entwicklungsprozess vorkommen. Beispielsweise bilden sich bei Komponententeams Warteschlangen, wenn mehrere Teams Beiträge zu einem gemeinsamen Feature beisteuern: Das Datenbank-Team implementiert Änderungen in der Datenbankschicht und stellt das Ergebnis in der Warteschlange des Geschäftslogik-Teams zur Verfügung. Dieses Team holt den Beitrag des Datenbank-Teams aus der Warteschlange, führt darauf aufbauend Anpassungen in der Geschäftslogik durch und liefert das Resultat an die Warteschlange des Front-end-Teams, das den letzten Teil des Features realisiert.
2. Mit den Architekten diskutierten wir alle möglichen Aspekte der praktischen Arbeit in Feature Teams, bis wir Lösungsansätze für die wichtigsten potentiellen Probleme gefunden hatten. Einige davon beschreiben wir unten näher. Sie halfen uns in unserem Projektumfeld, aber sie sind keine Patentrezepte und können in einer anderen Organisation ungeeignet sein. Der entscheidende Punkt ist, die Aspekte mit den Betroffenen zu besprechen und gemeinsam geeignete Lösungen zu vereinbaren.
3. Wir beschlossen, die bestehende Spezialisierung der einzelnen Teams nicht zu negieren, sondern zu unserem Vorteil zu nutzen. Zusammen mit den Architekten und den Scrum Mastern erarbeitete der Product Owner eine Untergliederung der Funktionalität der Applikation in inhaltlich verwandte Feature-Gruppen, die sich an dem vorhandenen Know-how der Teams orientierte. So gab es z.B. eine Gruppe von Features aus dem Bereich der Reportgenerierung, für die das frühere Komponententeam, das die wesentlichen Komponenten des Reportgenerators implementiert hatte, prädestiniert war.
4. Vor der ersten Iteration erklärten wir allen Beteiligten den neuen Entwicklungsprozess. Mit dem Beginn der Iteration erfolgte die Umstellung: Es gab keine Komponententeams mehr, nur noch Feature Teams.
5. In den ersten beiden Iterationen gab es viele Rückfragen und Feedback zum Entwicklungsprozess. Wir arbeiteten mit hoher Priorität an diesen Punkten, um kurzfristig Klärungen und Verbesserungen herbeizuführen.

Lösungsansätze

Die folgenden Lösungsansätze für die oben genannten Vorbehalte waren ein wesentliches Ergebnis bei der Umstellung auf Feature Teams:

- Für die Verbreitung des Know-hows über die wichtigsten Komponenten organisierten wir kurze Workshops, in denen die jeweiligen Komponenteneigentümer ausgewählten Vertretern aller Feature Teams einen Überblick über die Komponente gaben sowie die wesentlichen Entwurfsprinzipien und Fallbeispiele für zu erwartende Änderungen erklärten.
- Um die Konsistenz des Designs zu wahren, wenn unerfahrene Entwickler eine Komponente modifizieren müssen, führten wir die so genannten Komponenten-Consultants ein. Diese Rolle übernahmen die Entwickler, die am

besten mit der jeweiligen Komponente vertraut waren. Die Komponenten-Consultants hatten die Aufgabe, am Code Review teilzunehmen sowie Diskussionspartner für andere Entwickler zu sein, die Änderungen an der Komponente vornehmen. Auch wenn es keine Verpflichtung gab, den Komponenten-Consultant in Anspruch zu nehmen, so wurde diese Möglichkeit in den ersten Iterationen intensiv genutzt. Ein positiver Nebeneffekt war übrigens, dass sich durch die gegenseitige Hilfe zwischen den Teams das Vertrauensverhältnis im Gesamtteam verbesserte.

- Die Frage der Zuständigkeit für das Refactoring erwies sich als einigermaßen schwierig zu beantworten. In der Vergangenheit wurden Refactoring-Maßnahmen nicht explizit in der Iterationsplanung berücksichtigt, da sie der Komponenteneigentümer implizit miterledigte. Wir änderten diese Regelung: Jedes Refactoring musste zukünftig in einer Iteration eingeplant werden und erhielt auf diese Weise einen Verantwortlichen. Komponentengreifende Refactoring-Maßnahmen ließen sich auf diese Weise sogar einfacher durchführen als früher.
- Im Lauf der ersten Iterationen zeigte sich ein unerwartetes Problem: Bei technischen Klärungen mittels E-Mail waren zunehmend mehr Beteiligte einbezogen. Das lag daran, dass bei größeren Features typischerweise mehrere Komponenten geändert wurden mussten und der Entwickler des Features einfach alle relevanten Komponenten-Consultants in den E-Mail-Verkehr einband. Um diese relativ ineffiziente Form der Kommunikation zu verbessern, regten wir an, ein regelmäßiges Architekturmeeting zu schaffen. Darin konnten die Fragen direkt mit allen Experten besprochen und zumeist schnell geklärt werden.

Rückblick

Im Rückblick erwiesen sich Feature Teams als signifikante Verbesserung für unsere Projekte. Ein entscheidender Erfolgsfaktor bei der Umstellung von Komponententeams auf Feature Teams war, die Veränderung mit dem gesamten Team zu planen und die Durchführung der geplanten Maßnahmen eng zu verfolgen, sodass auftretende Probleme schnell behoben werden konnten. Nach der Umstellung auf Feature Teams und dem Überwinden anfänglicher Startschwierigkeiten halbierte sich unsere durchschnittliche Feature-Durchlaufzeit von 2,5 auf 1,25 Iterationen. Zwar spielten dabei noch weitere kleine Prozessverbesserungen eine Rolle, aber die geänderte Zuständigkeit der Teams steuerte den größten Anteil bei.

Eine vielleicht noch wichtigere Kennzahl als die Durchlaufzeit ist die Teamzufriedenheit. Wie beurteilten die Projektmitglieder den neuen Entwicklungsprozess nach sechs Monaten? 90% der Projektmitglieder waren sehr zufrieden, 10% gaben ein neutrales Feedback und kein einziger Projektmitarbeiter äußerte sich negativ.

Literatur

- Larman, Craig; Vodde, Bas: Feature Team Primer, 2010, http://featureteamprimer.org/feature_team_primer12.pdf, PDF-Datei, zuletzt eingesehen am 10.06.2013
- Larman, Craig; Vodde, Bas: Scaling Lean and Agile Development: Thinking and Organizational Tools for Large-Scale Scrum, Addison-Wesley Longman, Amsterdam, 2008
- Cohn, Mike: Succeeding with Agile: Software Development Using Scrum, Addison-Wesley Professional, 2009
- Kniberg, Erik: Lean from the Trenches: Managing Large-Scale Projects with Kanban, Pragmatic Programmers, 2012

Wenn es auf Geschwindigkeit und Flexibilität ankommt

Produkte, Projekte, Portfolios, Experimente – die agile Transformation der Haufe Group

Seit der Einführung einer projektorientierten Organisation vor etwa fünf Jahren wenden wir in der Haufe Group in immer mehr Projekten agile Prinzipien an. Vor allem in der Entwicklung von Softwareprodukten setzen sich agile Arbeitsweisen inzwischen durch. Dies gilt insbesondere für Neuentwicklungen. Mittlerweile erfordern die sich immer schneller verändernden Märkte, Kunden-Bedürfnisse und Rahmenbedingungen jedoch auch bei dem Weiterentwickeln von Bestandsprodukten ein Umdenken, um flexibler auf die Veränderungen reagieren zu können. So können wir z.B. dank moderner Endgeräte und neuer Entwicklungstechnologien noch intuitiver bedienbare und intelligenter Konzepte zur Benutzerführung realisieren. Diese wiederum entwickeln sich rasch weiter und befeuern dadurch auch die Erwartungshaltung unserer Kunden. Zusätzlich verstärken der Gesetzgeber und die Behörden den Druck zur Veränderung, indem sie in immer kürzeren Zyklen zunehmend komplizierte Änderungen vornehmen, die wir in unseren Produkten umsetzen müssen.

Wir brauchen kurze Zyklen zur Produktentwicklung!

Immer häufiger machen wir die Erfahrung, dass komplexe Projekte, in deren Verlauf sich schnell und häufig Anforderungen ändern (z.B. aufgrund modifizierter Geschäftsmodelle, aktueller Rechtsprechungen, geänderter Schnittstellenspezifikationen usw.), im Wasserfallprinzip nicht mehr zielführend abgewickelt werden können. Die sequentielle Abarbeitung in hierarchisch organisierten, bereichsspezifischen Spezialisten-Teams, funktioniert in einfachen sowie in komplizierten und gut abschätzbaren Projekten nach wie vor gut, ist jedoch in einem komplexen und sehr dynamischen Umfeld nicht mehr ausreichend. Was fehlt ist vor allem die Möglichkeit, in kurzen Zyklen (von zwei bis vier Wochen) funktionsfähige und auslieferbare Teil-Ergebnisse zu erzielen, Feedback vom Kunden einzuholen, daraus zu lernen und schnell im Projekt zu adaptieren. Agile Arbeitsweisen fördern und fordern genau das.

Umfassender Kulturwandel als Voraussetzung

Die Transformation einer gewachsenen Produktentwicklungslandschaft in eine agile Organisation ist eine Herausforderung für alle Beteiligten, weil nicht nur Arbeitsprozesse und -strukturen, sondern vor allem Denkweisen, Werte und Haltungen sich ändern müssen. Dies gilt sowohl für alle Teammitglieder, als auch für die Manager und die Mitglieder der Geschäftsleitung. Diesen Weg beschreitet die Haufe Group derzeit auch im Bereich der Desktop-Software-Lösungen der Marke Lexware. Er wird nur dann erfolgreich sein, wenn alle Beteiligten im Unternehmen mithelfen und bereit sind, Gewohntes loszulassen, Neues zu wagen und Veränderungen zuzulassen. Dies

Autoren



Michael Baur

Dipl.-Betriebswirt (BA) u. Informationswissenschaftler (Uni), aktuell tätig als

Projektmanager und Product Owner bei der Haufe Group im Geschäftsbereich Projekt- und Requirementsmanagement

Kontakt: michael.baur@haufelexware.com



Steffen Jakob

Dipl.-Wirt.-Ing. leitet seit 2013 den Geschäftsbereich "Projekt- und Requirementsmanagement" der Haufe Gruppe.

In dieser Funktion verantwortet er das Haufe-Projektportfolio

Kontakt: Steffen.jakob@haufelexware.com

Mehr Informationen unter:
> projektmagazin.de/autoren

betrifft die Art der Zusammenarbeit, die Wahrnehmung von Verantwortung, das Treffen von Entscheidungen, die Führung, die Unternehmenskultur und Vieles mehr.

Loslassen – Erfolgsfaktor und zugleich Stolperstein

Wir haben auf unserem bisherigen Weg gelernt, dass ein Erfolgsfaktor einer gelungenen Transition in ein agiles Cluster darin besteht, dass Manager und Geschäftsleitung "wohldosiert loslassen" und gemeinsam mit den Betroffenen an neuen Strukturen und Rahmenbedingungen arbeiten (dazu später mehr). Zudem braucht es mutige Mitstreiter aus der Belegschaft, die sich zutrauen Verantwortung zu übernehmen, Entscheidungen zu treffen und dabei auch Fehler zu machen.

Als ein wesentlicher Stolperstein hat sich bei uns vor allem der Umgang mit Fehlern erwiesen. Dies stellt einen von vielen Handlungsfeldern dar, wo wir als Projektmanager mit gutem Beispiel vorangehen und dazu beitragen können, dass die Transformation ein Erfolg wird. In unserer Rolle als Projektverantwortliche haben wir hierbei die Möglichkeit, das Team entscheiden zu lassen und ihm somit Vertrauen zu schenken. Bei einem Fehler dürfen wir dann keinesfalls nach den Schuldigen suchen und z.B. die Art der Korrektur vorgeben oder gar selbst vornehmen. Stattdessen sollten wir das Team dabei unterstützen, aus dem Fehler zu lernen und selbst eine Lösung zu entwickeln. Im bisherigen Verlauf unserer Transition kamen wir immer wieder an den Punkt, wo wir als Projektmanager sehr weiterhelfen können, ohne selbst steuernd einzugreifen (siehe dazu auch die Artikelserie "[Was zeichnet eine positive Fehlerkultur aus?](#)", ab Projekt Magazin 21/2015).

Die Haufe Group und die Lexware-Produkte

Die Haufe Group mit Hauptsitz in Freiburg im Breisgau hat weltweit insgesamt rund 1.800 Beschäftigte. Sie bietet ihren Kunden integrierte Arbeitsplatz- und Gesamtlösungen zur erfolgreichen Gestaltung ihrer steuerlichen, wirtschaftlichen und rechtlichen Aufgaben an. Sie ist in Deutschland eines der führenden Medien- und Softwarehäuser für Fachinformationen und -portale, (Cloud Computing-)Applikationen, eProcurement, Online-Communitys sowie Personal- und Organisationsentwicklung.

Zu den Hauptzielgruppen der Haufe Group gehören große und mittelständische Unternehmen, Kleinbetriebe und Selbstständige, Steuerberater und Anwälte, der Öffentliche Dienst sowie Immobilienunternehmen und Vereine. Bei ihnen nimmt die Haufe Group mit den Marken Haufe, Haufe Akademie und Lexware eine führende Marktstellung ein. Die Marke Lexware ist führend bei KMU im Bereich kaufmännischer Standard-Software-Lösungen (wie z.B. Lexware financial office), sowie bei privaten Nutzern mit Finanz- sowie Steuersoftware der Produktreihen "TAXMAN" und "Lexware Finanzmanager".

Weitere Informationen unter: www.haufegroup.com

Warum eine agile Transformation?

Die Strukturen und Prozesse in der Desktop-Software-Entwicklung bzgl. der Lexware Standard-Software-Lösungen waren viele Jahre lang stabil und wurden von uns in dieser Zeit nicht grundsätzlich in Frage gestellt. Auf die veränderten Marktanforderungen im Software-Umfeld reagieren wir mit dem verstärkten Ausbau unserer Cloud-Angebote. Wir müssen also die Herausforderung meistern, gleichzeitig neue, qualitativ hochwertige Cloud-Lösungen zu schaffen und gleichzeitig unsere Marktposition sowie die Stärken der Bestandprodukte sicher zu stellen.

Als Familienunternehmen setzen wir dabei auf eine nachhaltige und wirtschaftlich ausgewogene Finanz- und Investitionspolitik. Unsere Portfolio- und Produktstrategie spiegelt dies wider, indem wir bei den Bestandsprodukten auf gesetzliche sowie technische Aktualität und gute Qualität setzen. Bei den neuen Cloud-Produkten investieren wir zudem in die Funktionserweiterung. In beiden Fällen stehen die Kundenbedürfnisse im Vordergrund.

Agile Vorgehensweise für Cloud-Lösungen

Die neuen Cloud-Lösungen entwickeln wir nach agilen Prinzipien: Kleine, funktionsübergreifende und sich selbstorganisierende Teams entwickeln iterativ in kurzen, meist zweiwöchigen Sprints auslieferbare Produktinkremente. In den allermeisten Fällen sind wir dabei in der Lage, kontinuierlich Neuerungen an unsere Cloud-Kunden auszuliefern. Auf diese Art lernen wir direkt von den Nutzern und lassen das Gelernte in die nächste Iteration einfließen.

Was dabei als nächstes gemacht wird, ist im Produkt-Backlog ersichtlich, welches der Product Owner verantwortet. Die Teams arbeiten fast alle nach Scrum, unterstützt von einem Scrum Master. In diesem Projektumfeld benötigen wir keinen "klassischen" Projektmanager mehr. Die meisten seiner Aufgaben wie Steuern, Entscheidungen herbeiführen und den Projektfortschritt reporten, übernehmen Team, Product Owner und Scrum Master. Einige Projektmanager-Aufgaben fallen im agilen Kontext auch weg, wie z.B. das Erstellen eines dedizierten Ablaufplans mit festen Terminen und von Statusreports (siehe dazu den Tipp: "[So etablieren Sie Demos als Ersatz für Statusmeetings](#)", Projekt Magazin 03/2018).

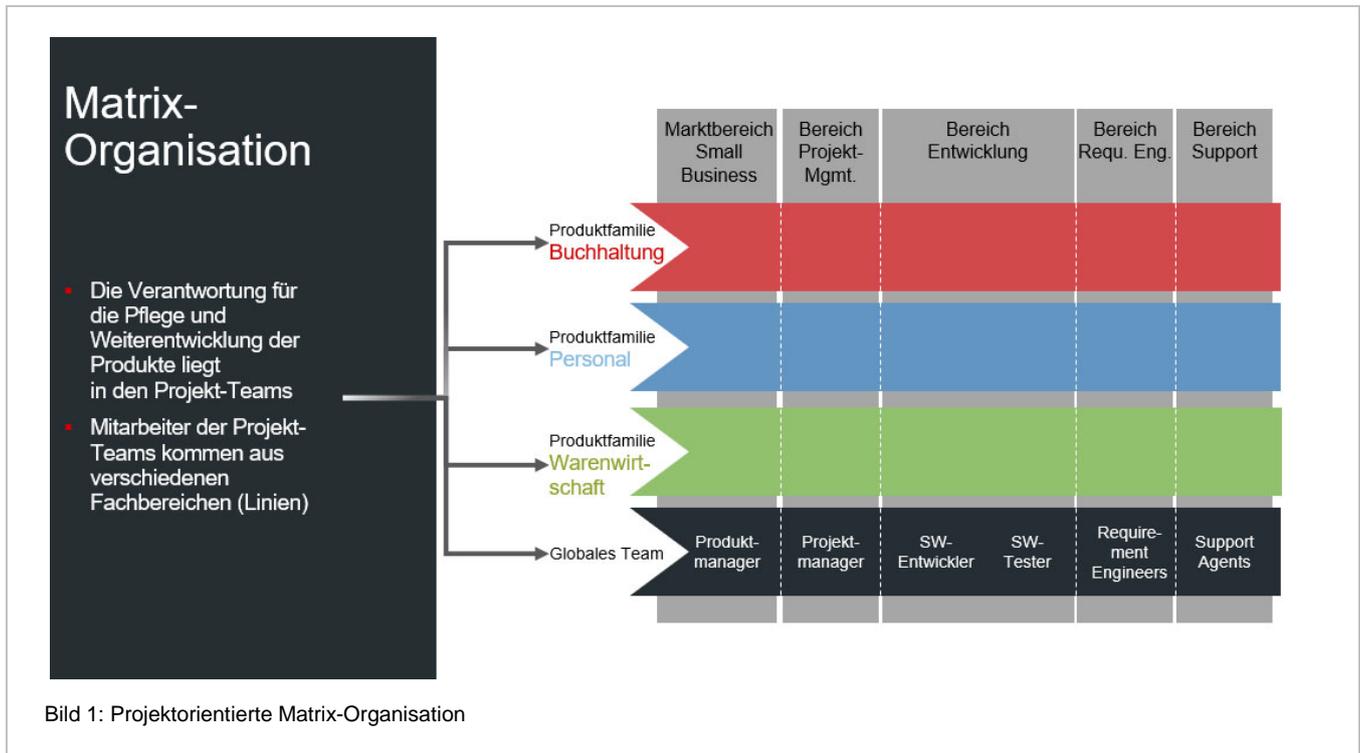
Wasserfallprinzip

In unserem Desktop-Bestandsgeschäft arbeiteten wir viele Jahre mit hierarchisch und nach Disziplinbereichen aufgestellten Teilteams. Die Produktpflege und die Projekte in diesem Umfeld führten wir vor Beginn der Transition sequentiell nach dem Wasserfallprinzip durch: Es wurden spezifische Aufträge formuliert und ein Projektmanager mit der Organisation und Durchführung beauftragt. Die Weiterentwicklung dieser Produkte erfolgte bei uns, seit Einführung einer projektorientierten Matrixorganisation (siehe Glossareintrag "[Starke Matrixorganisation](#)"), in produktspezifischen Teams: Wir als Projektmanager arbeiteten innerhalb dieser Projektteams mit vier Teilteams:

1. Konzeptionsteam, bestehend aus Business-Analysten
2. Entwicklungsteam aus Entwicklern und einem Development-Manager
3. Qualitätssicherungs-Team aus Testern und einem Test-Manager
4. Support-Team aus Support-Spezialisten und einem Support-Manager

Project Governance

Als Projektmanager verantworten wir die termingerechte Lieferung der neuen Releases mit den erforderlichen neuen Features, sowie gesetzlichen und technischen Anpassungen (je nach Produkt bis zu 50 pro Jahr) in hoher Qualität. Die Mitarbeiter der Projektteams gehören je nach fachlicher Ausrichtung zu unterschiedlichen Geschäftsbereichen und Abteilungen, sodass sich hieraus eine Matrixstruktur ergibt.



Ein zentraler Punkt der Project Governance ist, dass Produktmanager, Projektmanager und das Team gemeinsam den kommerziellen Erfolg der Produkte (Umsatz, Ertrag, Kundenzufriedenheit) verantworten. Dies spiegeln auch die persönlichen Zielvereinbarungen der Projektbeteiligten wider.

Neue Funktionen und Änderungen hatten wir zunächst meist vollständig konzipiert, dann entwickelt, danach getestet und erst nach einer Stabilisierungsphase an die Kunden ausgeliefert. Unser Support-Team hat dann die Kunden bei der Anwendung der neuen Funktionen und Änderungen unterstützt.

Die Hauptaufgaben für uns Projektmanager bestanden darin, das Ganze zu steuern, zu koordinieren, zu überwachen und den jeweiligen Projektfortschritt an die internen Auftraggeber zu berichten. Auftraggeber im Desktop-Bestands-Produkt-Umfeld sind bei Haufe-Lexware unsere Produktmanager aus dem zuständigen Zielgruppenbereich, hier "Small-Business".

Auslöser für die agile Transition im Bestandsgeschäft

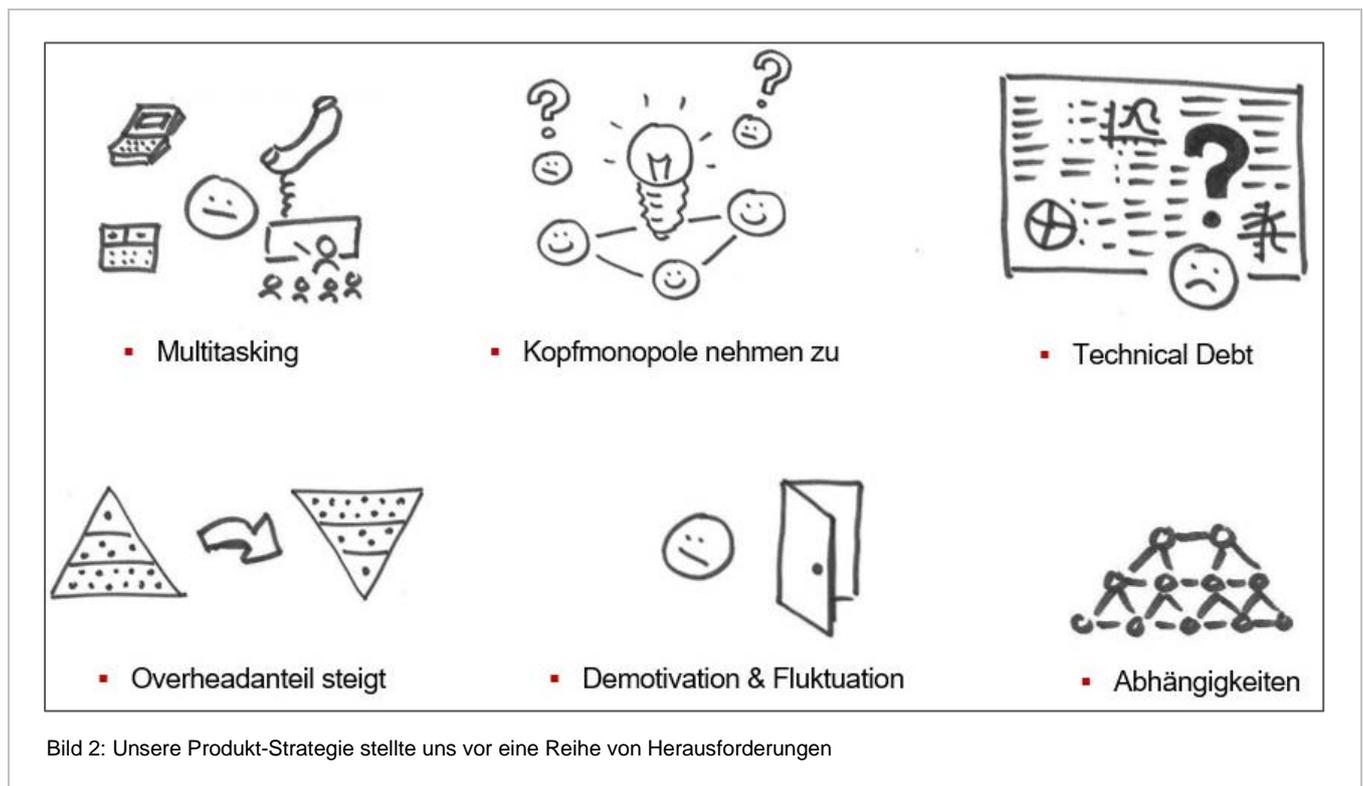
Ausgehend von unserer Unternehmensstrategie (Investition in Cloud, Marginsteigerung im Desktop-Umfeld) kürzten wir im ersten Schritt Budgets und verminderten die funktionale Erweiterung der Bestandsprodukte im Desktop. Aufgrund der vielen gesetzlich und technisch erforderlichen Änderungen, Anpassungen und teils umfangreichen Erweiterungen (vor allem im Bereich der Entgeltabrechnung), erhöhte sich der Druck auf die Desksupteams.

Disruptiver Ansatz als Ausweg

Die gestiegenen Anforderungen an die Teams sollten sich in keinem Fall auf die Produktqualität auswirken. So reifte in der Organisation die Erkenntnis, dass nur ein disruptiver Ansatz, der eine radikale Veränderung der Prozessregeln, der Strukturen und der Arbeitsweisen beinhaltet, einen Ausweg ermöglicht.

Herausforderungen und Handlungsbedarf

Die Kunden schätzen unsere Lexware-Desktop-Produkte vor allem aufgrund der einfachen Bedienbarkeit und für das gute Preis-Leistungsverhältnis. Motivation für den Kauf der Updates oder den Abschluss eines Abonnements, stellen einerseits vor allem die gesetzlichen Änderungen (u.a. neuer Steuertarif, neue Datensätze, neue Bescheinigungen) und andererseits die technischen Änderungen (u.a. Kompatibilität mit den aktuellsten Windows-, Browser- und Office-Versionen) dar. Aus diesem Grund investieren wir vor allem in diesen beiden Bereichen. Zur Reduzierung der Kosten passen wir die Entwicklungsbudgets jährlich nur so weit an, dass wir die technische Nachhaltigkeit und die Stabilität der Produkte nicht gefährden. Solche Veränderungen können verschiedene Konsequenzen für Projektteams nach sich ziehen (siehe auch Bild 2):



- **Multitasking und Abhängigkeiten:** Mitarbeiter mit speziellem Wissen müssen in mehreren Projekten gleichzeitig eingesetzt werden – dies erhöht die Rüstzeiten und senkt deutlich die Effizienz.
- **Kopfmonopole nehmen zu:** Das Wissen über fachliche und technische Zusammenhänge in den Produkten verteilt sich auf weniger Mitarbeiter – die Abhängigkeit von einzelnen Personen nimmt zu.

- **Technical Debt:** Die Investitionen in die Code-Qualität der Desktop-Produkte sind zu Gunsten neuer Produkte rückläufig. Dies birgt bei den Produkten, die stetig umfangreicheren Anforderungen von Seiten des Gesetzgebers erfüllen müssen, die Gefahr steigender technischer Schulden.
- **Overheadanteil steigt:** Durch die Reduzierung der "operativen" Ressourcen in den Projekten verschlechtert sich das Verhältnis von Steuerungsrollen zu operativen Rollen.
- **Demotivation und Fluktuation:** Viele Mitarbeiter wollen ihre Arbeitsweisen und ihre Produkte weiterentwickeln und am technologischen Puls der Zeit arbeiten. Verringerter Innovationsspielraum kann zu Demotivation führen.

Der Ruf nach Veränderung kam aus vielen Richtungen

Der Bedarf und der Wunsch eine Veränderung anzustoßen, wurde in vielen Bereichen und Teams immer lauter. Um diesen Entwicklungen zu begegnen, starteten wir zunächst eine bereichsübergreifende Initiative. Diese setzte es sich zum Ziel, die projektbasierten Strukturen durch ein agiles Cluster zu ersetzen. Sie bildete sich aus Mitgliedern der an der Produktentwicklung beteiligten Bereiche und hatte rund ein Dutzend Mitglieder.

Das Management gab uns das Go

Das Startsignal für die Initiative gaben letztendlich die Geschäftsbereichsleiter für Projektmanagement und Entwicklung. Hierbei führten wir bewusst auch Vertreter unterschiedlicher Hierarchieebenen zusammen. Ergänzt wurde das Team um die bereits in einigen Projektteams vorhandenen Scrum Master und agile Coaches. Als Treiber der Veränderung in den einzelnen Teams waren sie wichtige Multiplikatoren und Seismographen für die Befindlichkeit und die aktuelle Kultur in den Produktteams.

Die Projektmanager spielten eine wichtige Rolle

Gerade wir Projektmanager leisteten zu Beginn einen großen Beitrag, indem wir koordinierten und kommunizierten: Wir brachten inhaltlichen Input und Feedback aus unseren Projektteams mit und sorgten für die Transparenz und für das Verständnis vor allem innerhalb der bestehenden Entwicklungsorganisation. So wurden bestehende Probleme – wie wachsende Schwierigkeiten, die immer häufiger werdenden gesetzlichen Änderungen zeitnah zum Kunden zu bringen – aus den einzelnen Projektteams über die Projektmanager in die Initiative eingebracht. Wir Projektmanager berichteten zudem regelmäßig über die Fortschritte in der Initiative und holten so auch Feedback zu angedachten nächsten Schritten von den Teams ein.

Leitlinie: Mindset over Methodology

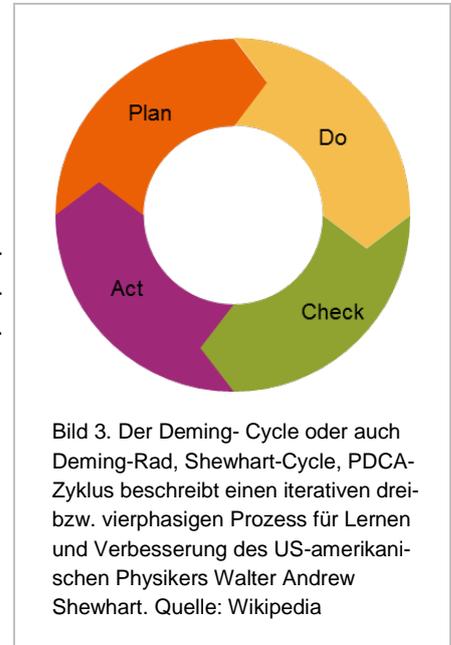
! Unsere zentrale Einsicht zu Beginn war, dass eine agile Transformation zu Selbstverantwortung und Selbststeuerung nicht von oben vorgegeben und "ausgerollt" werden kann. Sie muss durch die Betroffenen selbst gewollt und getrieben werden. Die Initiative stand also vor der Aufgabe, eine Entwicklung in Gang zu bringen, ohne dabei die betroffenen Mitarbeiter zu übergehen oder für sie Entscheidungen zu treffen. Dieses Paradoxon aufzubrechen, war und ist eine echte Herausforderung.

Die Initiative löste dieses Paradoxon auf, indem sie es sich zur Aufgabe machte, Impulse für Veränderungen zu setzen. Sie hob Denkverbote und andere Hemmnisse auf und ermutigte auch die anderen Mitarbeiter dazu. Das umfasste auch und besonders Vorgaben, welche die Teams selbst nicht hinterfragt hätten.

Beispiel: Wir empfahlen einem Team, welches keinen Scrum Master zugeteilt bekommen hatte, einen Teil seines bewilligten Projektbudgets zur Schaffung und Besetzung dieser Rolle einzusetzen.

Wie wir der agilen Philosophie treu blieben

Auch das Vorgehen zur Schaffung dieses agilen Clusters sollte den Grundsätzen von agilem Arbeiten entsprechen. Dies implizierte, dass kein – theoretisch am Reißbrett durchdachtes – Konzept entwickelt wurde, das dann ausgerollt werden würde. Unser Ansinnen war vielmehr, dass die Veränderungen in Form von einzelnen, kleinen Schritten ("Experimenten") erfolgen sollte, die leicht umgesetzt und deren Resultate zeitnah überprüft werden können. Wir haben uns darauf verständigt die Schritte "**PLAN-DO-CHECK-ACT**" (sog. Deming Cycle oder PDCA-Zyklus, siehe Bild 3) dabei möglichst häufig und kurz zu durchlaufen, um basierend auf Erfahrungen zu lernen und weitere Veränderungen und Optimierungen anstoßen zu können.



Dieses iterative Vorgehen war umso wichtiger, als die Veränderungen im laufenden Betrieb durchgeführt werden und eine Störung der Release-Auslieferung nicht tolerierbar ist. Den Beteiligten war darüber hinaus klar und wichtig, dass es bei der Initiative nicht darum ging, den Zustand A in den Zustand B zu überführen, sondern dass damit ein kontinuierlicher Prozess angestoßen wird, der permanente Änderungen und Verbesserungen ermöglicht, ohne jemals zu einem "perfekten" Endzustand zu kommen.

Konsequentes Üben in agilem Arbeiten

Wichtig war, dass wir bereits während der Vorarbeiten in unserer Initiative einige wesentliche agile Praktiken anwendeten, wohlwissend, dass wir aufgrund der Zusammensetzung und Parallelität zum Tagesgeschäft nicht konsequent nach agilen Prinzipien oder innerhalb eines Scrum Frameworks agieren konnten. Uns darin dennoch zu versuchen war wichtig, um einen transparenten und kontinuierlichen Veränderungsprozess anzustoßen, bei dem Veränderung durch Überprüfung und Anpassung entsteht. Entscheidend war, dass wir einige Elemente aus dem agilen Umfeld nutzten um weitere Erfahrungen damit zu sammeln:

- **Erstellen und Führen eines Backlogs:** Im initialen Kick-Off-Termin der Initiative sammelten wir Themen, Maßnahmen und Stories, die wir für das Initialisieren und Vorbereiten einer Transition als relevant ansahen. Diese Sammlung bildete die Basis für das Backlog. Wir formulierten die Themen in Form von User Stories. Für jede Story definierten wir Akzeptanzkriterien um sicherzustellen, dass wir alle das gleiche Verständnis darüber hatten, was mit der jeweiligen Story erreicht werden sollte. Als System zur Dokumentation und Verwaltung der Stories nutzen wir die Kollaborationssoftware "Trello" (www.trello.com) – eine leichtgewichtige, selbsterklärende Cloud-Plattform, die sich sehr bewährte. Trello bietet zwar nicht den Funktionsumfang und die Tiefe gängiger agiler Werkzeuge, wie beispielsweise JIRA, war für unsere Zwecke aber intuitiver anwendbar (siehe dazu auch den Fachbeitrag "[Agiles](#)

"Projektmanagement mit Trello" und "Tricks für die tägliche Projektarbeit mit Trello"). Jedes Mitglied der Initiative konnte das Backlog bearbeiten und erweitern. Es diente als Basis für die Sprint-Backlogs.

- **Durchführung von 4-Wochen-Sprints:** Da die Teilnehmer der Initiative aufgrund anderer Aufgaben nicht in Vollzeit an den Themen arbeiten konnten, wurde ein 4-Wochen-Sprint-Rhythmus eingeführt, in dem die in den Plannings definierten Themen bzw. Stories bearbeitet wurden.
- **Review / Planning / Retro:** Alle vier Wochen fand ein Halbtagsworkshop statt, bei dem wir uns die im Sprint bearbeiteten Themen gegenseitig vorstellten, diskutierten und anhand der Akzeptanzkriterien überprüften. Fertige Stories wurden abgeschlossen, nicht fertiggestellte Stories neu bewertet und entweder auf den nächsten Sprint verschoben oder ins Backlog zurückgelegt. Da es keinen Product Owner gab, erfolgte die Abnahme durch das gesamte "Initialisierungsteam". Im Anschluss daran wurde das Planning durchgeführt. Hier legten wir fest, welche Stories wir aus dem Backlog in der nächsten Iteration umsetzten und wer von uns sich an der Umsetzung jeweils beteiligte. Hierbei überprüften wir gemeinsam die Akzeptanzkriterien der einzelnen Stories und passten diese ggf. an.

Das Lexware Cluster erblickt das Licht der Welt

Intensiv überlegten wir in dieser Zeit, wie und wann alle betroffenen Mitarbeiter involviert werden sollten – bis zur gemeinsamen Entscheidung, die Initiative für alle betroffenen Mitarbeiter (potenziell über 200) zu öffnen und jedem die Mitarbeit zu ermöglichen. Dass es diese Initiative gab, war von Beginn an transparent und wurde auch von der Geschäftsführung bewusst so kommuniziert. Wir Projektmanager bildeten die Schnittstelle in die Projektteams.

Die Initiative öffnet sich

Die "Öffnung" der Initiative bedeutete den Startschuss für die Transition. Hierfür bereiteten wir – ebenfalls über mehrere Sprints – einen gemeinsamen Kick-Off-Termin vor, zu dem wir alle Beteiligten einluden und ihnen die Ausgestaltung und Umsetzung der Transition übertrugen. Die an der Weiterentwicklung der Lexware-Desktop-Produkten Beteiligten kommen aus unterschiedlichen Teams und Bereichen unseres Unternehmens. Es gibt also keine eigene Organisationseinheit für alle hier Beteiligten.

Um klarzustellen, welche Projektteams und Geschäftsbereiche die Transition umfassen wird, nannten wir den Zusammenschluss "Lexware Cluster". Die erste Initiative löste sich nach rund fünf Monaten Vorbereitungszeit zu diesem Zeitpunkt auf und die bisherigen Mitglieder wurden Mitglieder des Clusters (inkl. der beteiligten Geschäftsführer), wo sie ihre Ideen und Vorschläge einbrachten.

Kontinuierliche Veränderung durch (selbstorganisierte) Experimente

Das Cluster leistete von Beginn an einen handfesten Beitrag zur Weiterentwicklung des Unternehmens: Gleich im ersten Treffen äußerten die Teammitglieder bestehende Problemstellungen und Herausforderungen, welche aus ihrer Sicht gelöst oder verbessert werden mussten. Dazu entwickelten wir eine Reihe von Experimenten. Die Ideen dazu kamen von Mitgliedern des Clusters, die wir anschließend gemeinsam priorisierten. Um das agile Arbeiten zu üben, beschlossen wir, dass die Mitarbeiter diese Experimente im folgenden Monat **selbstorganisiert** während der Arbeitszeit durchführten. Erfolgreiche Experimente sollten zur Veränderungen und Weiterentwicklung des Lexware Clusters beitragen.

Reflektion und Weiterentwicklung durch Retrospektive

Seither trifft sich das Lexware Cluster monatlich zur halbtägigen "**Overall-Cluster-Retrospektive**". In dieser Retrospektive werden die Transitionsaktivitäten des Clusters zusammengeführt, entsprechend dem agilen Prinzip des "Inspect and Adapt", also einer kontinuierlichen Reflektion und Anpassung bzw. Weiterentwicklung. Ein über eine Retrospektive initiiertes Experiment wird in der nächsten Retrospektive auf dessen Ergebnis und Wirkung überprüft und falls notwendig angepasst.

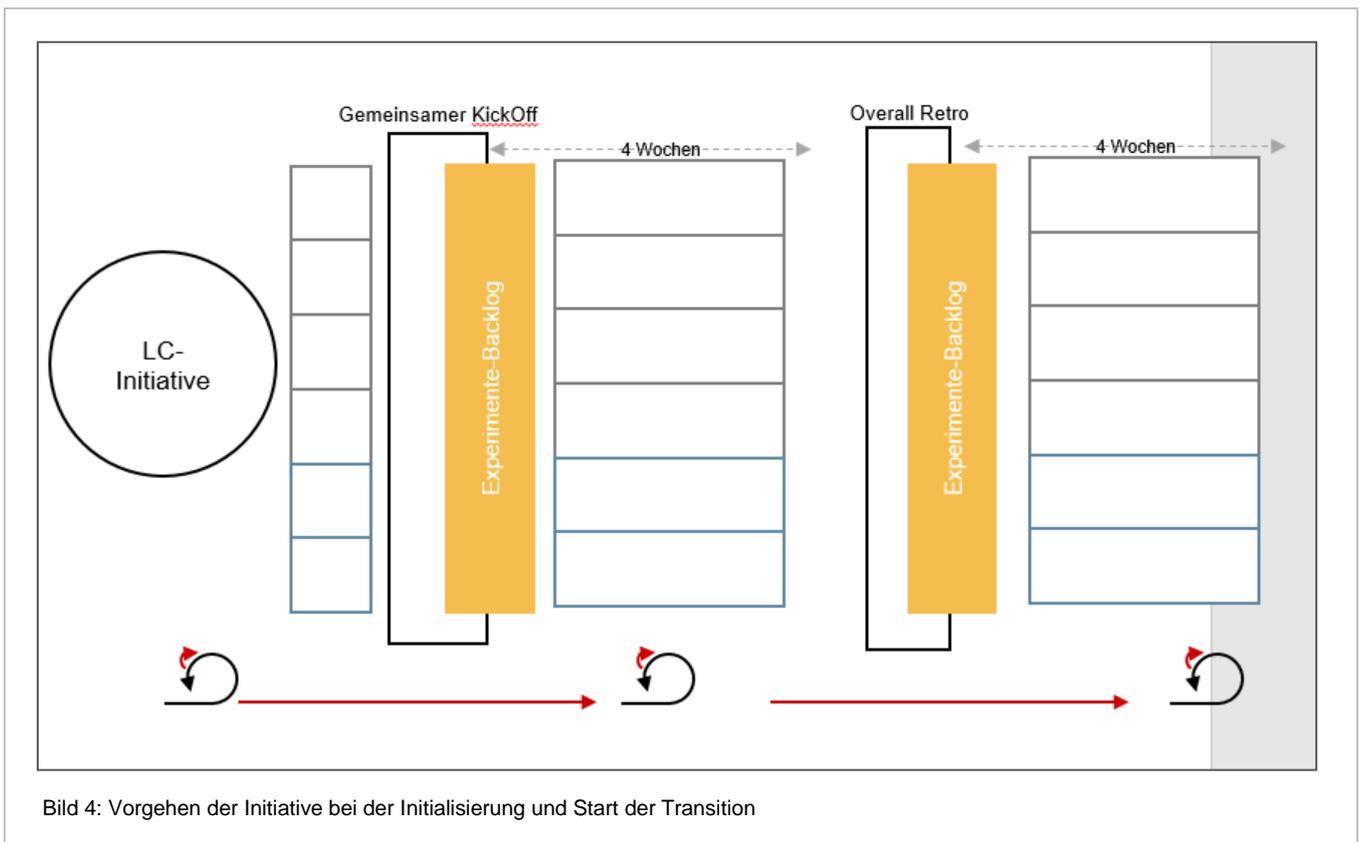


Bild 4: Vorgehen der Initiative bei der Initialisierung und Start der Transition

Transparent dank Wiki

Transparenz stellen wir über ein **Cluster-Wiki** sicher, das wir bereits zum "Kick-Off" einrichteten. Dieses Wiki enthält sämtliche Inhalte zu allen im Cluster laufenden Experimenten. Die Inhalte werden stets aktuell gehalten und jeder Beteiligte im Cluster kann Kommentare hinterlegen, Feedback geben und Verbesserungsvorschläge einbringen.

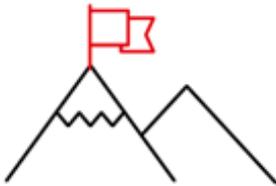
Auf diese Weise begannen wir eine schrittweise Transition in eine neue Struktur und in neue Prozesse, die wir sicherlich nie abschließen werden. Die in der Initialisierungsphase entwickelten Ergebnisse – in der Regel zum Verbessern der Rahmenbedingungen, dazu später mehr –, haben wir in die tägliche Arbeit des Lexware Clusters übernommen.

Ein Zielbild zur Orientierung und Ausrichtung

Das Zielbild für die Transition des Lexware Clusters entstand bereits in einer der ersten Iterationen des Initialisierungsteams. Um ein gemeinsames Verständnis zu erarbeiten, beschäftigten wir uns bereits zu Beginn der

Initiative intensiv mit der Vision und den Zielen. Hier trafen unterschiedliche Vorstellungen aufeinander, was die weitere Arbeit sehr befruchtete. Durch die Auseinandersetzung mit den verschiedenen Aspekten aus unterschiedlichen Blickwinkeln entstand ein ausgewogenes und von Allen getragenes Zielbild. Neben einer übergreifenden Produktvision erstellten wir auch eine Beschreibung von der Struktur und der Arbeitsweise, die uns dabei helfen sollten, diese Vision zu erreichen.

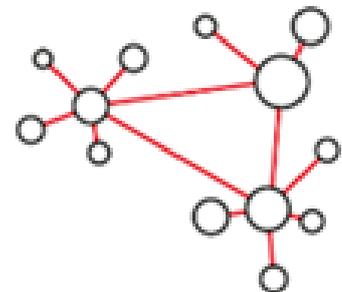
Zielbild und Vision



Das Lexware Cluster ist eine Organisation zur Produkt-Entwicklung, die sich an Markt- und Unternehmenserfordernissen ausrichtet und Standardsoftware herstellt. Ausgehend von differenzierten und regelmäßig überarbeiteten Produktstrategien auf Basis der Unternehmens- und Bereichsstrategie liefert sie Produkte, die ein optimales Verhältnis von Kundennutzen, Gebrauchstauglichkeit, Profitabilität und Nachhaltigkeit aufweisen.

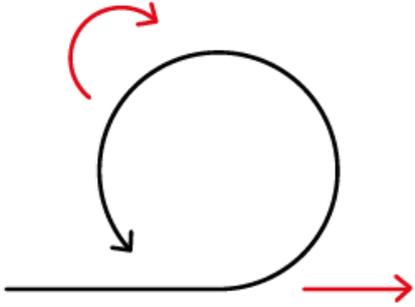
Struktur

- Wir arbeiten in kleinen, selbstorganisierenden Teams an Lexware On-Premise-Produkten.
- Die Linienzuordnung der Mitarbeiter spielt eine untergeordnete Rolle.
- Jeder Mitarbeiter gehört ausschließlich einem Team an.
- Es entsteht eine klare, simple und schlanke Struktur, die sich ihren Aufgaben anpasst.



Arbeitsweise und Kultur

Die Arbeit im Team ist attraktiv dank vielfältiger Aufgaben und Abwechslung. Interdisziplinäres Lernen und Entwickeln ist nicht nur möglich, sondern notwendig, denn so entsteht eine Kultur, die auf den Prinzipien des Agilen Manifests basiert. Transparenz und kontinuierliche Verbesserung sind Teil dieser Kultur.



Durch das Pull-Prinzip entstehen Freiräume für den Einzelnen, es bleibt Zeit für die Pflege sozialer Beziehungen und Innovation.

Die Arbeit wird über ein übergreifendes Cluster-Backlog priorisiert und gesteuert. Eine gemeinsame Backlog-Struktur sowie auch ansonsten gleichlaufende Arbeitsweisen der Teams – etwa in der Ausprägung der Rollen und genutzten Systeme, sowie der Taktung der Sprints –, unterstützen das Alignment der Teams im Hinblick auf das gemeinsame Ziel.

LeSS Hufe als ein mögliches Ziel-Szenario

Als mögliches Ziel-Szenario identifizierten wir das Scrum-Skalierungsmodell "LeSS" bzw. "LeSS Hufe", an welchem sich das Lexware Cluster orientieren und zukünftig annähern wird. Large Scale Scrum (LeSS) ist ein Rahmenwerk für agile Software-Entwicklung nach Scrum-Regeln mit bis zu acht Teams. LeSS Hufe ist die erweiterte Variante für mehr als acht Teams, was bei der Größe des Lexware Clusters das relevante Rahmenwerk wäre. Bild 5 zeigt die Struktur und das übergreifende Prozessmodell von LeSS Hufe. (siehe dazu auch den Fachbeitrag "**Mit vielen Teams erfolgreich ein Produkt mit Scrum entwickeln**", Projekt Magazin 09/2017)

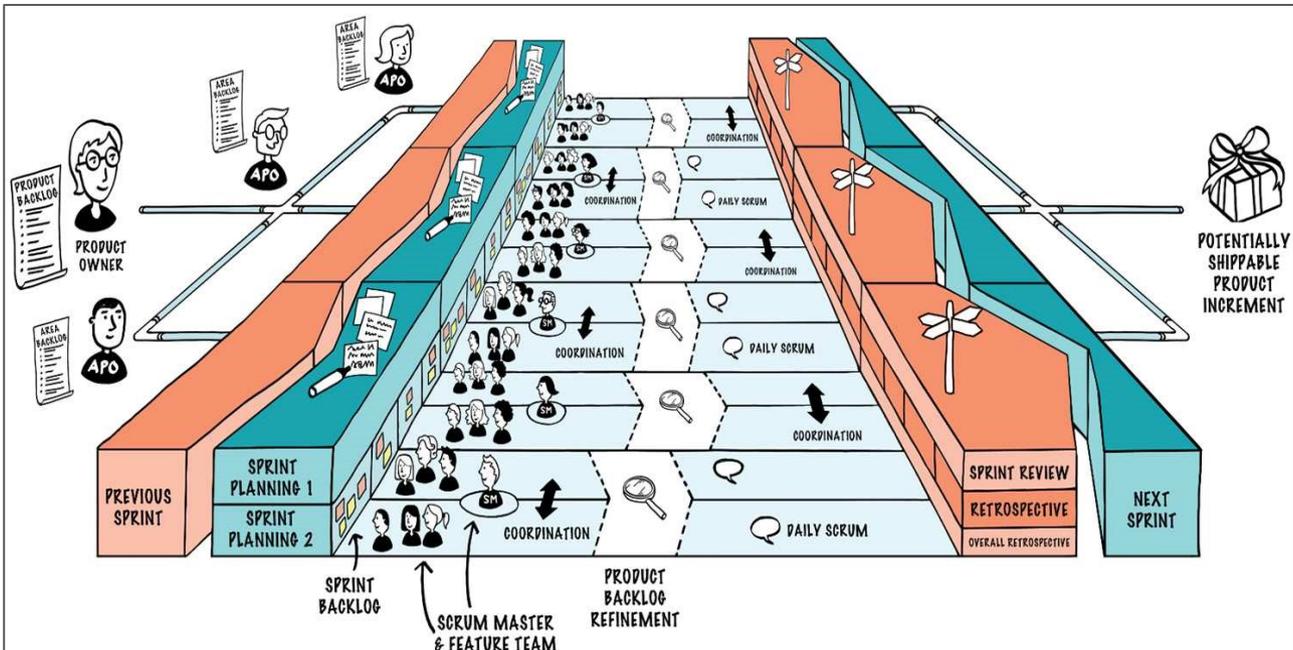


Bild 5: LeSS Hufe (Quelle: <https://less.works/de/less/less-hufe/index.html>)

LeSS baut auf Scrum auf, d.h. um LeSS praktizieren zu können, müssen erst die wesentlichen Elemente von Scrum verstanden und angewandt werden. Im LeSS Framework gibt es die gleichen Rollen wie in Scrum mit dem

Unterschied, dass es neben einem Product Owner, der den Gesamtblick auf das Produkt hat, Area Product Owner gibt, die jeweils für einen Bereich des Produkts verantwortlich sind. Jedem dieser Area Product Owner sind mehrere DEV-Teams zugeordnet, die von Scrum Mastern betreut werden.

In LeSS koordinieren die Teams ihre Aktivitäten untereinander: Alle Teams arbeiten zusammen in einem Sprint, um ein gemeinsames, auslieferbares Produktinkrement herzustellen – und zwar in jedem Sprint. Die üblichen Scrum Ereignisse (Planning, Refinement, Review und Retrospektive) finden teilweise teamintern und teilweise teamübergreifend statt. Ziel ist es, die Aufmerksamkeit aller Teams auf das ganze Produkt zu lenken, anstatt nur auf einen Teil. Damit sollen Probleme wie Doppel-Entwicklungen, negative Auswirkungen bzw. Nebeneffekte auf andere Teile oder unabgestimmte Architekturentscheidungen gelöst werden.

Die Intention von LeSS ist es, die Organisation so einfach wie möglich zu gestalten und die Wertschöpfung zu erhöhen, indem vermeidbare Tätigkeiten wegfallen bzw. in wertschöpfende Aktivitäten umgewandelt werden. Hierzu sind in LeSS zehn Prinzipien und eine Reihe von Regeln bzgl. der Organisation und des Produkts definiert.

Was ist unser gemeinsames Produkt?

LeSS und LeSS Hüge gehen von einem gemeinsamen Produkt aus. Die Klärung, was das gemeinsame Produkt ist, stellt für uns im Lexware Cluster eine der wesentlichen grundlegenden Fragestellungen dar. Je nachdem, wie wir diese Fragestellung im weiteren Verlauf der Transition beantworten, ergäbe sich nach diesem Skalierungsframework für uns eine Struktur nach LeSS Hüge oder eine von mehreren LeSS'en. Aktuell liegt der Fokus vor allem darauf, dass die neu formierten Feature Teams sich kontinuierlich verbessern. Dies gilt natürlich auch für die Scrum Master und Product Owner. Erst wenn die Feature Teams für sich reif genug sind, sollte weiter in die Skalierung investiert werden. Zu viel Veränderung auf einmal wäre an dieser Stelle eher hinderlich bzw. überfordernd.

Was hat sich bislang verändert?

Anpassen von Rahmenbedingungen, Strukturen, Rollen und Prozessen

Schulungen und Trainings zu LeSS, PSM, PSPO, PSD

Es ist essentiell, dass alle im Team von vorne herein die theoretischen Grundlagen erlernen, um von diesem gemeinsamen Wissenstand "abspringen" zu können. Nur so wird es möglich, sich gemeinsam schrittweise in Richtung agiler Arbeitsweisen weiterzuentwickeln.

Um bei allen Beteiligten einen gemeinsamen Wissensstand zu schaffen, etablierten wir mehrere Schulungsformate. Das Schulen übernahmen überwiegend externe Trainer in Inhouse-Schulungen, weil das deutlich effektiver, mit mehr Praxisbezug und kostengünstiger ist, als alle auf eine externe Schulung zu schicken. Zum einen kann dann an konkreten gemeinsamen Praxisbeispielen diskutiert werden. Zum anderen fallen Reisezeiten und Reisekosten weg.

Die Schulungen zu PSM (Professional Scrum Master), PSPO (Professional Scrum Product Owner) und PSD (Professional Scrum Developer) vermittelten den jeweiligen Rollen sowohl das Wissen um Methodiken und Praktiken von Scrum als auch die hinter Scrum liegenden Werte und Prinzipien, die für eine erfolgreiche Implementierung

wichtig sind. Damit konnte ein gemeinsames Verständnis sowohl über die Arbeitsweise als auch über die Kultur der Zusammenarbeit geschaffen werden.

Ziel der LeSS-Schulung war es, bei Product Ownern und Scrum Mastern ein Grundverständnis für die innere Logik und Struktur von LeSS zu schaffen und Fragen zur Machbarkeit zu adressieren. Insbesondere die Koordination zwischen den Feature Teams und den Aufbau des übergreifenden Backlogs und der Area Backlogs wurden dabei adressiert.

Einführen von Scrum-Rollen und -Ereignissen

Um agil arbeiten zu können, bildeten wir zunächst kleinere cross-functional Feature Teams. Zugunsten dieser funktionsübergreifenden Teams lösten wir die bisher bestehenden Teams aus den Fachbereichen auf. Jedes cross-funktional Team bekam einen "Servant Leader" (siehe dazu auch den Glossareintrag zu **Scrum Master**). Für jedes größere Produkt übernahm ein Mitglied aus dem Cluster die Rolle des Product Owners. Ergänzend wurden Product Backlogs aufgebaut, die die bestehenden Lasten- und Pflichtenhefte ablösten. Darüber hinaus führten wir in den Teams Scrum-Ereignisse wie Dailys, Plannings, Refinements, Reviews und Retrospektiven ein.

Da an ein paar wenigen Produkten auch mehrere Teams arbeiten, wurden hier im Verlauf der bisherigen Transition schrittweise übergreifende Scrum-Ereignisse (Overall-Planning, Overall-Retrospektive und Demo-Fair) eingeführt, um die Arbeit an einem gemeinsamen Produkt und damit an einem Product-Backlog zu koordinieren und zu optimieren.

Verschanken von Strukturen und Prozessen

Ein starker Hebel zu mehr Effizienz war die Vereinfachung der Planungs- und Controllingstrukturen. Bisher war die Lexware-Desktop-Produktentwicklung in mehrere Projekte unterteilt, die einzeln geplant und budgetär gesteuert wurden. Wir konnten die Aufwände für Planung und (monatliches) Controlling deutlich reduzieren, indem wir einen Gesamtauftrag für alle Produkte definierten. Gleichzeitig haben wir durch eine produktgenaue Verrechnung der Ist-Aufwände nach wie vor eine hohe Kostentransparenz, auf deren Basis sich Entscheidungen treffen lassen.

Wie viele Unternehmen hatten auch wir ein variables Gehaltsmodell, das auf der Erreichung von individuellen Zielen basiert – und davon ausgeht, dass das Schaffen von monetären Anreizen für persönliche Leistung motivierend wirkt. Heute sind wir davon überzeugt, dass diese Anreize eher demotivieren und haben darum die individuellen Ziele der Mitarbeiter im Cluster durch Unternehmensziele ersetzt. Dies hatte eine wichtige Signalwirkung, unter anderem hinsichtlich der Zusammenarbeit in cross-funktional Teams in einem übergreifenden Cluster. Darüber hinaus sparen wir uns viel Zeit, die uns das Vereinbaren und Messen der diversen Einzelziele kostete.

Ein wichtiger Schritt zu mehr Klarheit und Einfachheit war das Festlegen auf Scrum als verbindliches Framework für alle Teams des Lexware Clusters. Diese Entscheidung erfolgte erst deutlich nach dem Start der Transition. In der Folge werden die unterschiedlichen Steuerungsrollen, die wir in der bisherigen Projektorganisation hatten und aktuell teilweise noch haben, in den nächsten Wochen abgeschafft. Die verbleibenden Aufgaben gehen auf die neuen Rollen und v.a. auf die Teams über. Die bisherige Doppelspitze, bestehend aus Produkt- und Projektmanager, haben wir durch die Rolle des Product Owners ersetzt. Teaminterne Steuerungsrollen wie Development- oder Testmanager werden wir auflösen und damit eine Struktur schaffen, in der es nur noch die in Scrum definierten Rollen des Product Owners, Scrum Masters und des Development Teams gibt.

Als logische Konsequenz werden diese Rollen Anpassungen in der Folge auch in der Aufbauorganisation nachvollzogen, sodass Product Owner, Scrum Master und Mitglieder der Development Teams auch in Linienteams zusammengeführt werden. Damit einher geht auch eine veränderte Aufgabe der Linienführung (Team- bzw. Bereichsleitung): Sie versteht sich zukünftig als People Coach, der/die für die persönlichen Belange und die Weiterentwicklung der Mitarbeiter zuständig ist, hat dabei aber keinerlei fachliche Einflussnahme mehr auf die Arbeit der Mitarbeiter im Cluster.

Bisherige Erfolge und Erfahrungen

Das Ziel ist (auch) der Weg

Die Transformation einer klassisch über "Command & Control" funktionierenden Organisation in ein agiles, selbstorganisiertes Cluster, kann nicht top-down entschieden und vorgegeben werden, will man den Werten treu bleiben, die zu der Transition motivierten.

Nur Werte stiften Sinn

Schwerpunkt und Herausforderung jeder agilen Transition ist nicht die Struktur- oder Prozessveränderung, sondern der Wandel im Denken und in der Haltung der betroffenen Personen. Dies setzt die Bereitschaft voraus, sich mit diesem agilen Wertesystem auseinanderzusetzen und es zu seinem eigenen zu machen – damit kann es nicht von außen vorgegeben, sondern muss von den Betroffenen akzeptiert werden.

Agile Führung vs. Top-Down-Entscheidungen

Gleichzeitig braucht es die Impulse und die Unterstützung der aktuell verantwortlichen Rollen. Dieses Paradoxon auszuhalten und die Balance zwischen "Impulse setzen" und "darauf vertrauen, dass die Betroffenen selbst die wichtigen Punkte angehen", ist eine große Herausforderung für das Management. Zwar dauert dieser Weg mutmaßlich länger als ein klassischer Change Prozess, wir sind jedoch davon überzeugt, dass er nachhaltiger und am Ende wirksamer ist als das Durchdrücken einer Veränderung von oben nach unten.

Hier haben wir – auch durch Fehler – in den vergangenen Monaten viel gelernt: Zu Beginn gaben wir den Teams zu viel Freiraum und zu wenig Struktur. Wir erkannten dies daran, dass die von Teammitgliedern gestarteten Experimente wenig greifbare Ergebnisse einbrachten.

Zur Korrektur griff das Management ein und traf Entscheidungen teilweise ohne die im agilen Kontext ratsame Beteiligung der Mitarbeiter. So wurden z.B. die neuen Product-Owner-Positionen im Management definiert und intern zur Besetzung ausgeschrieben, zunächst ohne eine Rücksprache mit den Personen, die bisher diese Aufgaben innehaben. Auch die organisatorische Ansiedlung der Scrum Master legte das Management fest, ohne Einbeziehung der bisher mit den Servant-Leader-Aufgaben betrauten Personen. Dies führte bei Letzteren zu Irritationen.

Die Kommunikation zwischen Mitarbeitern und Management war aber so offen, dass die Verwirrung über diese Entscheidung konstruktiv besprochen werden konnte. Die Führungskräfte nahmen daraufhin Entscheidungen zurück, um sie im Dialog mit den Mitarbeitern neu zu treffen. Die bewusste Entschuldigung für diesen Fehler stellte Vertrauen und Akzeptanz wieder her und war gleichzeitig ein wichtiger Beitrag für unsere Fehlerkultur.

Veränderung als kontinuierlichen und selbstorganisierenden Prozess begreifen

Während konventionelle Change-Management-Ansätze einen Prozess beschreiben, der vom Zustand A in den (hoffentlich besseren) Zustand B führt, ist unser Transitionsprozess auf Dauer angelegt: Kontinuierliches, durch Experimente validiertes Lernen; kleine, aber durchgehende Veränderungen, die die Betroffenen selbst initiieren, sind dabei wesentliche Elemente. Dies impliziert auch, dass es niemals einen "perfekten" Endzustand geben wird – und keine Stabilität in dem Sinne, dass sich nichts mehr ändern wird.

Gleichlaufende Arbeitsweisen reduzieren die Komplexität

! Wir haben gelernt, dass unterschiedliche Vorgehensweisen, sowohl bzgl. der Zusammenarbeit als auch was den zeitlichen Ablauf angeht, die adressierten Probleme nicht ausreichend lösen, weil so Abhängigkeiten und Kommunikationsverbindungen nicht nachhaltig verringert werden. Die unterschiedlichen Arbeitsweisen erschweren das Zusammenarbeiten der Teams untereinander. Alle Feature Teams müssen sich im selben Sprintrhythmus organisieren und die gleichen agilen Ereignisse durchführen.

Relevanz und Bedeutung der Scrum Master und agile Coaches

Als sehr wichtig und zielführend hat sich die Einbeziehung der bereits vorhandenen Scrum Master und Agile Coaches in die Initiative erwiesen. Diese sind tagtäglich im direkten Kontakt mit den Entwicklungsteams und können so einerseits die vorhandenen Stimmungen und Einstellungen zu einer agilen Transition aufnehmen und andererseits als Coaches und Multiplikatoren, die auf ein gemeinsames Ziel hin ausgerichtet sind, in die Teams hineinwirken. Unterstützt werden muss dies allerdings durch breit angelegte Schulungs- und Trainingsangebote für alle beteiligten Mitarbeiter.

Darüber hinaus mussten wir uns zu Beginn damit behelfen, Scrum Master aus den bestehenden Teams zu entwickeln. Bedingt dadurch, dass diese Mitarbeiter an der inhaltlichen Arbeit des Teams weiterhin Anteil nahmen, – oder sogar daran beteiligt waren, weil sie anfangs meist noch in einer Doppelrolle arbeiteten – konnten sie ihre Scrum-Rolle nur eingeschränkt ausfüllen, was die gesamte Entwicklung zum Scrum-Team bremste. Ideal wäre es, wenn man bereits zu Beginn derartige Doppelrollen vermeiden könnte, indem man die Scrum-Master exklusiv für diese Aufgaben freistellt. Zudem wäre es ratsam, auch erfahrene Scrum Master ins Team aufzunehmen, um hier eine gute Mischung zu haben.

Lessons learned zum Teamsetting

Auch im Hinblick auf das Teamsetting haben wir in mehreren Entwicklungsschritten einiges gelernt und schrittweise optimiert:

- Ein Team sollte nicht zu groß sein, da sich sonst Untergruppen bilden und Entscheidungsprozesse deutlich schwieriger werden. Zur Reduzierung der Kommunikationswege und Komplexität empfiehlt sich eine Team-Größe von fünf bis neun Personen.
- Jedes Team sollte cross-functional zusammengesetzt sein, um den stetigen Austausch und die Zusammenarbeit zu fördern.
- Ein Team an einem Standort findet viel leichter zusammen als ein verteiltes Team. Letzteres muss zusätzliche Hürden überwinden (neben der Notwendigkeit einer stetigen technischen Verbindung): Vor allem Retrospektiven und Teambildungsprozesse werden durch das Fehlen von persönlichem Kontakt (sowohl während der Arbeit in den Sprints, als auch in den Pausen, wie z.B. in der Teeküche oder beim Mittagessen) deutlich erschwert.

Herausforderung der Gleichzeitigkeit

Diese Initiative muss parallel zur laufenden Produktweiterentwicklung betreut werden. Eine Verschlechterung der Qualität, eine Reduzierung des Scopes oder die Verschiebung eines Releases müssen wir unbedingt vermeiden. Dies fordert von allen Beteiligten ein hohes Maß an Einsatz und eine Bereitschaft zu Multitasking. An vielen Stellen mussten wir feststellen, dass Beteiligte aufgrund ihrer hohen Auslastung durch das Tagesgeschäft zu wenig Focus auf die Themen der Initiative legen konnten und ihre kontinuierliche Beteiligung darunter litt. Zur Verbesserung dieser Situation führten wir u.a. ein Limit ein, sowohl für die Anzahl der aktiven Experimente, als auch für die gleichzeitige Teilnahme von Personen an mehreren Experimenten.

Erfolgsfaktor Management

Einer der wichtigsten Erfolgsfaktoren ist die Unterstützung und die Bereitschaft der Geschäftsleitung, der leitenden Personen sowie der Manager, **Verantwortung und Macht abzugeben**. Unabdingbar ist dabei, den Teams das nötige Vertrauen zu schenken, damit diese auch die Verantwortung übernehmen können. Nur so werden die Teams wichtige Entscheidungen treffen und in der Lage sein, aus Fehlern zu lernen und sich stetig zu verbessern.

Balance zwischen Freiraum und Struktur halten

Gleichzeitig ist es wichtig, die Teams nicht zu überfordern, indem man ihnen zu früh zu viel Freiraum gibt. Wir haben die Erfahrung gemacht, dass eine transparente und vor allem vertrauensvolle Zusammenarbeit von Management und Teams hilft, die Balance zwischen Loslassen und der – gemeinsamen – Schaffung neuer Strukturen zu halten. Dies gelingt nur, wenn keine Fronten aufgemacht werden und alle Beteiligten sich an einem gemeinsamen Zielbild orientieren.

Viele Aufgaben werden mittelfristig in die Teams wandern, einige Themen werden Product Owner und Scrum Master übernehmen und andere Dinge werden wegfallen. Für alle diese Veränderungen müssen Geschäftsleitung, Management und Projektmanager bereit sein, auch weil es schlussendlich bedeutet, dass dadurch Management-Aufgaben und damit auch -Positionen wegfallen können. Die Bereitschaft, die eigene Rolle in Frage zu stellen, erfordert eine gehörige Portion Mut, hat aber eine extrem hohe Vorbildfunktion für alle Beteiligten in diesem Veränderungsprozess.

Die agile Reise endet nie, daher sollte man genau klären, warum man sie antritt

Eine derart umfangreiche Transformation anzustoßen und letztlich durchzuführen, darf kein Selbstzweck sein. Es ist nicht ratsam, die Transition einer Produktentwicklungsorganisation anzugehen, nur weil "agil" eben modern ist. Vielmehr muss allen Beteiligten – über sämtliche Hierarchieebenen hinweg – klar sein, aus welchen Gründen und zu welchem Zweck eine Veränderung angestrebt wird.

Entscheidet sich eine Organisation begründet und zielorientiert, zukünftig nach agilen Prinzipien zu arbeiten, bedarf dies einer kontinuierlichen, auf "Inspect & Adapt" beruhenden Weiterentwicklung. Am Beginn einer agilen Transition kann es keinen fest definierten Endzustand geben. Vielmehr ist es wichtig, sich auf die agilen Prinzipien und Werte zu verständigen und sich an einer gemeinsam getragenen Vision auszurichten. All das erfordert Offenheit, Vertrauen, Transparenz, Mut, Zeit und viel Geduld.

Literatur

- Larman, Craig; Vodde, Bas: **Large-Scale Scrum**, Addison Wesley, 2016
- Dies.: LeSS – More with LeSS, offizielle Homepage zum LeSS-Framework: <https://less.works>
- Schwaber, Ken; Sutherland, Jeff: The Scrum Guide, November 2017
online abrufbar unter: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf> (englisch) und <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-German.pdf> (deutsch)

Large-Scale Scrum mit dem LeSS-Framework

Mit vielen Teams erfolgreich ein Produkt mit Scrum entwickeln

Teil 1: Rahmenbedingungen und Basismodell

Large-Scale Scrum (LeSS) ist ein aufstrebendes Modell für die agile Entwicklung von Produkten mit mehreren Teams. Es baut auf den Regeln, Artefakten und Events von Scrum auf. LeSS gibt es als Basismodell (ab zwei Teams) und als LeSS Huge speziell für Skalierungen mit mehr als acht Teams. Sowohl Scrum, als auch Large-Scale Scrum sind für Soft- als auch für Hardware-Produkte und Mischformen geeignet.

Scrum wird in letzter Zeit auch immer mehr für andere Industrien verwendet, z.B. bei Baufirmen oder Finanzdienstleistern. Allerdings eignen sich für Dienstleistungen andere Ansätze besser. Manche Prinzipien und Praktiken können dennoch auch für reine Dienstleistungen verwendet werden. Die Hauptzielgruppe für die Anwendung von Large-Scale Scrum sind Unternehmen, die Produkte in Form von Hard- oder Software entwickeln.

Die drei wichtigsten Treiber für die Skalierungsframeworks LeSS und LeSS Huge sind

- die Organisation zu vereinfachen,
- das Produkt klarer und breiter zu definieren, um Synergien nutzbar zu machen und
- die Planung wertoptimiert zu gestalten.

In diesem Artikel werden die grundlegenden Prinzipien, Regeln, Chancen und Risiken einer Skalierung mit Large-Scale Scrum dargestellt.

Wer LeSS will, muss Scrum können

Wie bereits erwähnt, baut LeSS direkt auf Scrum auf und übernimmt alle Regeln, Richtlinien, Artefakte, Rollen und Events (Meetings) von Scrum. Der Scrum Guide behält seine Gültigkeit. Bevor LeSS angewandt wird, sollten daher Scrum und dessen Elemente bekannt sein. Ebenso wie Scrum erfordert LeSS ein sehr strukturiertes und diszipliniertes Vorgehen, eröffnet damit allerdings neue Freiräume für Kreativität und Innovation.

! Agile Entwicklung mit Scrum benötigt zumeist eine Veränderung der Organisationsstrukturen. Darum sollten weder Scrum noch LeSS lediglich als Verfahren oder Methode verstanden werden. Vielmehr bilden sie eine Grundlage für Organisationsdesign und -entwicklung.

Da LeSS eine skalierte Version von Scrum für einzelne Teams ist, gilt folgendes:

Autor



Dr. Wolfgang Richter
Geschäftsführer von
JIPP.IT und Certified
LeSS Trainer

Kontakt: wolfgang.richter@jipp.it

Mehr Informationen unter:
[› projektmagazin.de/autoren](https://projektmagazin.de/autoren)

Es gibt

- einen einzigen Product Backlog (da er für das Produkt ist und nicht für das Team),
- die "Definition of Done" für alle Teams,
- das potenziell auslieferbare Produktinkrement am Ende jedes Sprints,
- den Product Owner,
- viele funktionsübergreifende ("cross-funktionale") Teams (Teams mit Mehrfachspezialisierung),
- einen gemeinsamen Sprint.

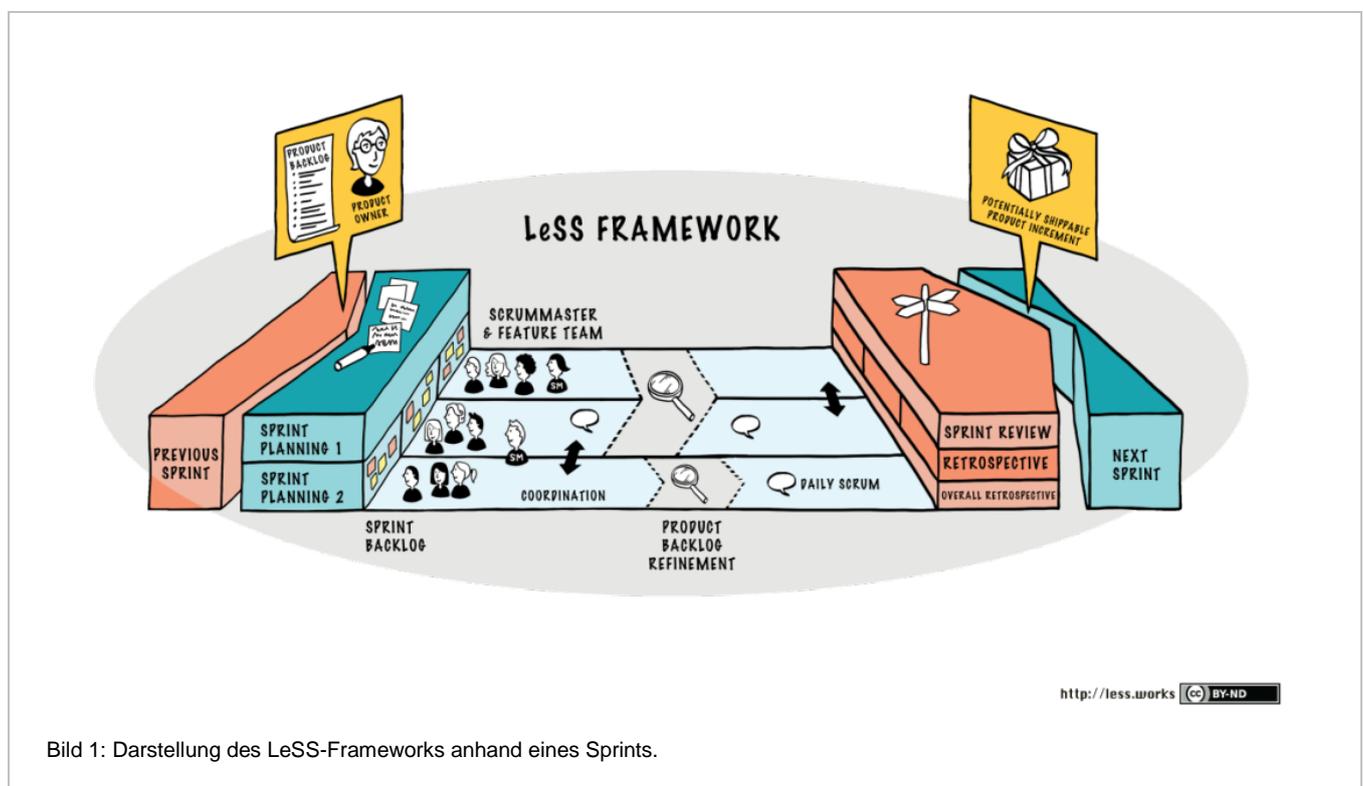


Bild 1: Darstellung des LeSS-Frameworks anhand eines Sprints.

Gemeinsamer teamübergreifender Sprint

In LeSS arbeiten alle Teams in einem gemeinsamen Sprint, um ein gemeinsames, auslieferbares Produktinkrement herzustellen. In jedem Sprint! Dieser Punkt ist bei der Einführung (Adoption) von LeSS mental oft schwer fassbar, da viele Organisationen von Kundenprojekten getrieben sind und oft der Ansatz gewählt wird, für jedes Projekt ein eigenes Projektteam zusammenzustellen. Diese Teams bleiben selten länger als über die Projektdauer zusammen, was dem soziologischen Aspekt von "echten Teams" widerspricht.

Aber auch in Szenarien, in denen Kundenprojekte die treibende Kraft sind, ist es wichtig, die Übersicht zu behalten und Auslieferungen soweit wie möglich aufeinander abgestimmt zu planen. Dadurch können Synergieeffekte

genutzt werden und es kann sich gemeinsames Lernen entwickeln. Die Organisation profitiert dabei von der Kosteneinsparung durch Vereinfachungen von Verwaltungsstrukturen und Prozessen.

Die meisten der Skalierungselemente von Large-Scale Scrum zielen darauf ab, die Aufmerksamkeit aller Teams auf das ganze Produkt zu lenken, anstatt auf "meinen Teil". Bei Skalierungen gilt es, Probleme, die sich aus der Zusammenarbeit von vielen Teams ergeben, zu sehen und zu adressieren. Solche Probleme, z.B. doppeltes Erstellen von Funktionalitäten, unabgestimmte Architekturentscheidungen, Erstellen oder Änderung einer Funktionalität mit Nebeneffekten in nicht sichtbaren Bereichen bei anderen Einheiten usw. sind oft die negativen Folgen, die dadurch entstehen. Aus Produktsicht gilt es, eine wertoptimierte Gesamtlösung zu schaffen, anstatt lokaler Optimierung. Der Fokus muss daher auf das Gesamtbild gelegt werden.

Die Rolle des Product Owners

Der Product Owner ist in Scrum wie auch in LeSS als Geschäftsperson zu sehen, im Sinne eines Produktmanagers, der die Wirtschaftlichkeit der Produktentwicklung und des Produkts verantwortet. Aufgaben wie Business Analysis und die Klärung von Anforderungen oder Splitting von Stories können durch den Product Owner unterstützt werden, sind aber Teamverantwortung. Fertigkeiten wie z.B. Requirements Engineering und Business Analyse sollten daher bei den Teammitgliedern vorhanden sein.

Ein häufig anzutreffendes Missverständnis ist, dass der Product Owner die Anforderungen als User Stories formuliert und diese in das Product Backlog einpflegt. Diese Aufgabe wird dem Team übertragen – wie es einer der Erfinder von Scrum, Ken Schwaber, bereits in der ursprünglichen Definition des Product Owners beschrieben hat.

Ein Backlog für alle Anforderungen

Um die Verantwortung für die Wirtschaftlichkeit übernehmen und somit die wertoptimierte Sortierung von Anforderungen durchführen zu können, müssen die Anforderungen so einfach wie möglich sortierbar sein. Zu diesem Zweck gibt es in LeSS nur ein einziges Product Backlog. Ich werde an dieser Stelle häufig gefragt, wie man mit mehreren Kunden umgeht. Die Antwort darauf ist, die Anforderungen im zentralen Product Backlog mit Kundeninformation zu versehen und somit ein weiteres Sortierkriterium zu haben. Kunden- oder Teambacklogs sollen vermieden werden, da sie die Komplexität der Planung erhöhen. Die einfachste Version in Form einer Tabelle kann z.B. wie in Tabelle 1 gezeigt aussehen:

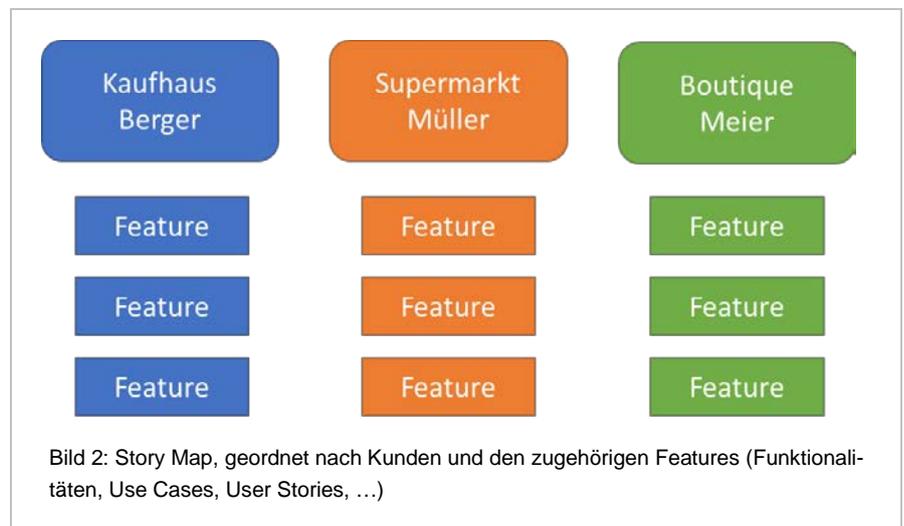
ID	Beschreibung	Komplexität (Story Points / T-Shirt Sizes)	Kunde	Referenzen
1	Als Interessent möchte ich ...	3	Kaufhaus Berger	Mockup 01
5	Als Shop-Administrator möchte ich ...	5	Supermarkt Müller	Mockup 05

7	Als Shop-Betreiber möchte ich ...	1	Supermarkt Müller	
3	Als Stammkunde möchte ich ...	8	Kaufhaus Berger	Gesetzestext ...
14	Als Neukunde möchte ich ...	3	Boutique Meier	
...	...			
...	...			
...	...			
35	Promotionscodeverwaltung	L	Kaufhaus Berger	Konzept P-Codes
46	Personalisierung von Ads	XL	Boutique Meier	Beispiel www.ABC.com

Tabelle 1: Zentrales Backlog mit den Anforderungen aller Kunden.

Um den Backlog zu befüllen, können Story Maps hilfreich sein. Sie sind vor allem in Umgebungen mit sehr vielen Kunden hilfreich, da dieser zusätzliche Visualisierungsschritt eine bessere Sichtbarkeit der anstehenden Aufgaben ermöglicht. Der Nachteil ist, dass es sich um ein zusätzliches Verwaltungsinstrument handelt und somit mehr Aufwand in die Verwaltung fließt. Eine Story Map kann im einfachsten Fall aussehen wie in Bild 2 gezeigt:

Eine andere Gliederung der Story Map ist oftmals hilfreicher. Wenn die treibende Kraft und das Ziel das einheitliche Produkt ist, nehmen Sie Themen als Überschrift und ordnen darunter Features unabhängig vom Kunden an. Produkte wie z.B. Plattformen für Webshops werden im Kern gleich designed für alle Kunden und einzelne Features für Kunden customized, also individuell gestaltet. Das könnte wie in Bild 3 dargestellt aussehen (die Farben der Features indizieren die Kundenzugehörigkeit).



! Bei verteilten Backlogs werden oft zusätzliche Funktionen wie Projektmanager oder Teamleiter eingeführt, die hauptsächlich mit Verwaltungsaufgaben beschäftigt sind. Sofern Verwaltungsaufgaben nur aufgrund einer vermeidbaren Komplexität existieren, stellen sie keine wertschöpfenden Tätigkeiten dar.

Zur Erinnerung: Ziel ist es, die Organisation so einfach wie möglich zu halten, um Zeit und Energie der beteiligten Personen in die wertoptimierte und wertschöpfende Produktentwicklung investieren zu können.

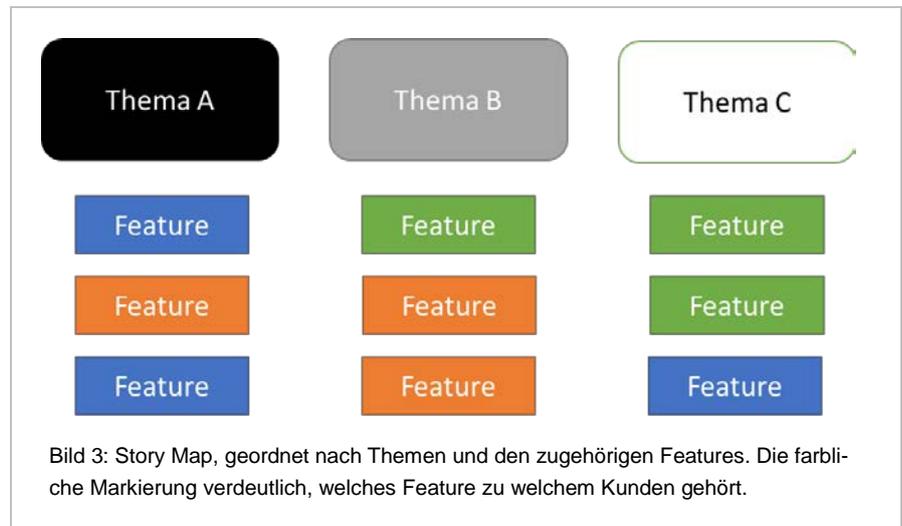


Bild 3: Story Map, geordnet nach Themen und den zugehörigen Features. Die farbliche Markierung verdeutlicht, welches Feature zu welchem Kunden gehört.

Wertschöpfung als übergeordnetes Ziel

LeSS verspricht bei dessen Anwendung "Job Safety", aber keine "Role Safety". Das bedeutet: Es wird jeder gebraucht, muss sich aber unter Umständen verändern. Wertmindernde, bzw. vermeidbare Tätigkeiten werden durch die Adoption und die damit verbundene Analyse der Organisation in wertschöpfende Aktivitäten geändert.

Der Product Owner kann und muss entscheiden, welcher Kunde wie bedient wird. Aussagen wie, "alles ist gleich wichtig" oder "ich kann nicht reihen, wir müssen alle gleichzeitig beliefern" hebeln jede wertoptimierte Sortierung aus und führen in der Regel zu längeren Durchlaufzeiten. Eskalationen können entstehen, der Druck auf die Teams, schneller zu liefern, kann sich erhöhen und zu Kompromissen, wie weniger Testen, führen. Auf Dauer gesehen ist all das ein Automatismus, der die sogenannte "**Technische Schuld**" erhöht.

Grenzen der Wirtschaftlichkeit erkennen

Hat man erst Technische Schulden, muss man in deren Rückzahlung investieren. Schulden verursachen auch Zinsen, welche eine weitere Investition darstellen. Eine Falle, die den Gewinn auf mittlere bis lange Sicht reduziert und im Worst-Case in den Ruin führen kann.

Die Relation zwischen Technischen Schulden und der Produktivität einer Organisation ist im Normalfall indirekt proportional: Je höher die Schulden, desto niedriger die Produktivität. LeSS setzt aus diesem Grund auf die Vereinfachung der Organisation und die Vereinheitlichung des Produkts, um diesen Automatismus so gut wie möglich zu eliminieren.

Zusätzlich fordern Scrum und LeSS "Technical Excellence". Damit ist gemeint, dass in der Umsetzung die bestmöglichen Praktiken, Werkzeuge und Lerneffekte genutzt werden sollen, um qualitativ hochwertigste Produkte zu verwirklichen. Natürlich muss das unter Berücksichtigung der Wirtschaftlichkeit passieren.

Kosten-/Nutzenrechnungen sind essentiell, um die Grenzen der Wirtschaftlichkeit zu erkennen, Technische Schulden möglichst zu vermeiden und Zeit und Energie hauptsächlich in die wertschöpfenden Aktivitäten fließen zu lassen. Dazu zählt in der Agilität auch das Lernen und kontinuierliche Verbessern.

Bedingungen, um LeSS erfolgreich einzuführen

LeSS muss einfach sein

Wenn man skaliert, tendiert man dazu, Rollen, Artefakte, Prozesse, etc. hinzuzufügen. Prozesse sollten sich allerdings empirisch aus der Produktgruppe entwickeln, damit nur Abläufe als Prozess definiert werden, deren Nutzen klar ist. Die meisten Skalierungsframeworks tappen in die Falle der vordefinierten Prozesse. In LeSS soll genau das vermieden werden. Prozesse sind nicht verboten und ein "Starter-Set" an Prozessen ist in Ordnung. Es sollte jedoch ständig verbessert und optimiert werden.

LeSS ist skaliertes Scrum

Statt Scrum als Baustein für ein Skalierungs-Framework zu verwenden, müssen wir Scrum detailliert betrachten und uns für jedes einzelne Element (Rollen, Artefakte, Events) fragen: "Warum ist es hier?", gefolgt von "Wie kann ich denselben Effekt auf einer höheren Ebene erzielen, wenn ich mehr als ein Team habe?". LeSS beantwortet diese Fragen grundlegend, lässt allerdings genug Spielraum, um die Besonderheiten des konkreten Umfelds berücksichtigen zu können. Damit wird erreicht, dass die Vorteile von Scrum weiterhin genutzt werden können.

Es wird hochskaliert statt zurechtgeschnitten

Ein oft angewandtes Prinzip zur Prozessentwicklung ist, ein universelles, allumfassendes Framework zu definieren und es an die Gegebenheiten anzupassen, indem Elemente zurechtgeschnitten oder weggelassen werden. Das funktioniert erfahrungsgemäß nicht besonders gut, da Menschen gerne annehmen, dass alles in ihrem speziellen Umfeld notwendig ist. Zusätzlich kommt oft die Vermutung zum Tragen, dass das eigene Umfeld essentiell anders funktioniert als das von allen anderen. Diese Annahmen führen in der Regel zu aufgeblähten Prozessen oder stark veränderten Modellen. Daher geht LeSS ähnlich dem Prinzip des Minimum Viable Products von der minimal notwendigen Menge an Prozessen zu Beginn aus. Besser schnell starten und lernen, als zu lange vorbereiten, um dann mühsam herauszufinden, was zu viel ist.

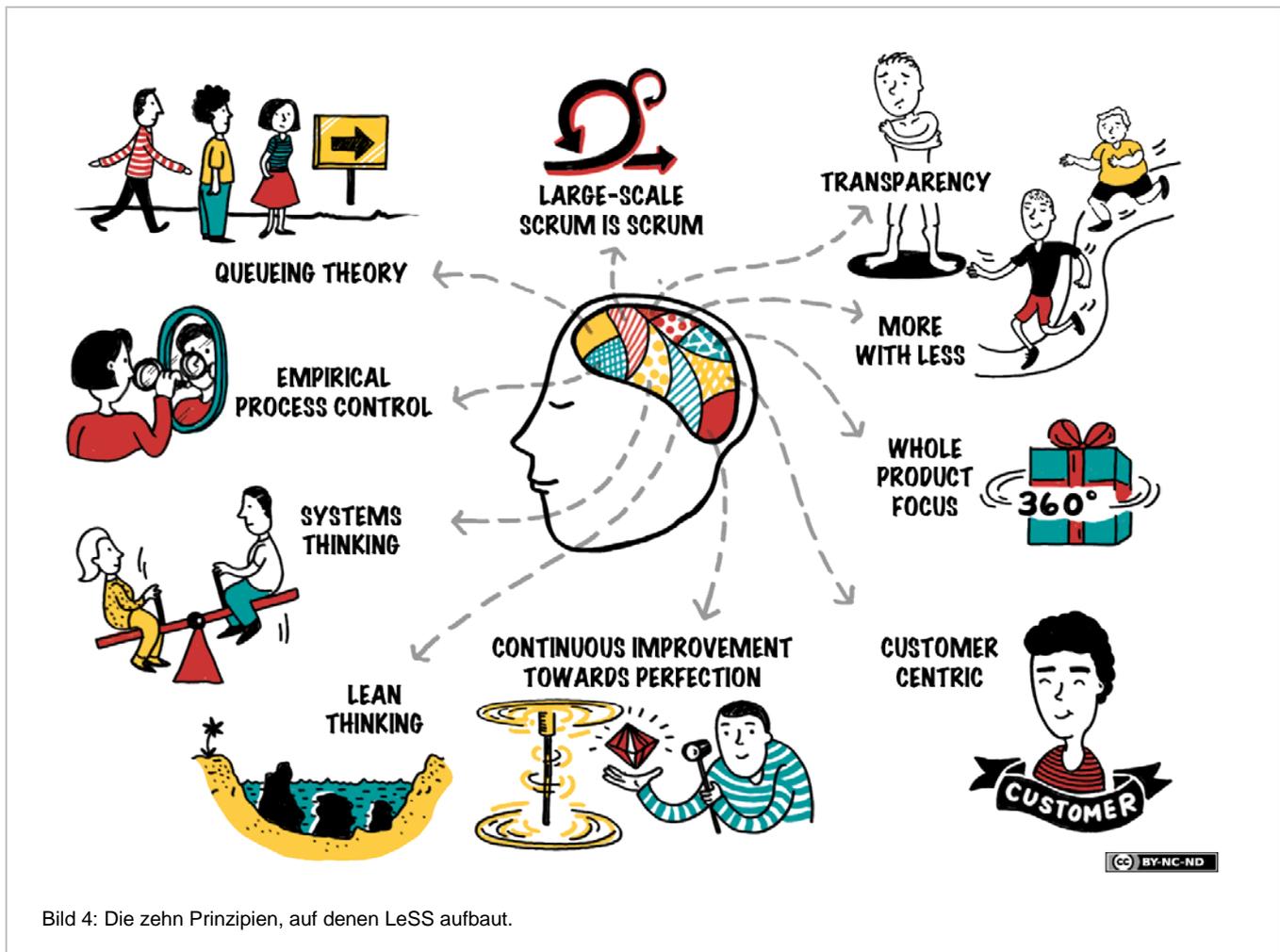
Doch was sind die minimal notwendigen Prozesse, um die Organisation erfolgreich laufen zu lassen? Die Antwort darauf ist, die Regeln von LeSS zu befolgen und diese als Struktur und Prozesse umzusetzen.

Notwendige Prinzipien und Regeln in LeSS

Obwohl Scrum und LeSS weit mehr sind, als das bloße Befolgen von Regeln: Erst ein Grundgerüst an Prinzipien und Regeln ermöglicht ein strukturiertes Vorgehen und schafft damit Freiräume für Kreativität und Innovation.

Zehn Prinzipien schaffen das richtige Umfeld für LeSS

LeSS baut auf zehn Prinzipien auf, die zum Teil selbst als Frameworks verstanden werden können. Es ist nicht notwendig, alle Prinzipien für den Start mit LeSS bis ins Details zu verstehen, allerdings sollten die wichtigsten Erkenntnisse daraus bekannt sein. **Die zehn LeSS Prinzipien:**



1. Large-Scale Scrum ist Scrum:

Es ist kein "neues oder verbessertes Scrum". Bei LeSS geht es darum, die Prinzipien, Elemente und den Effekt von Scrum in einer skalierten Umgebung anzuwenden. Multi-Team Scrum anstatt von mehreren Scrum Teams.

2. Empirische Prozesskontrolle:

Inspektion und Adaption des Produkts, der Prozesse, des Organisationsdesigns sowie der Praktiken, um eine der Situation angepasste Organisation auf der Basis von Scrum zu entwerfen, anstatt einer detaillierten Formel zu folgen. Eine empirische Prozesskontrolle benötigt und kreiert Transparenz. Das betrifft alle vom ersten Tag an und jeder ist gefordert, seine Beobachtungen und Erkenntnisse zu teilen. Folgende Möglichkeiten bieten sich hierzu an:

- Daily Standups, in denen Teams sich synchronisieren und ihre Arbeit verbessern,

- Go-See für Manager, die dorthin gehen, wo die Umsetzung erfolgt, um reale Bilder der Arbeitsumgebungen zu erhalten,
- Scrum Master Communities, die ihre Beobachtungen und Werte aus Retrospektiven und anderen Events teilen,
- Overall Retrospektiven, in denen die verschiedenen Gruppen – von der Geschäftsführung, über den Product Owner, bis zu Team Repräsentanten – zusammen kommen, um gemeinsam Verbesserungen für die Organisation zu erdenken; Zahlen, Daten, Fakten in Kombination mit Gefühlen sind die wichtigsten Größen empirischer Prozesskontrolle. Gemessen werden können z.B. Faktoren wie Idle Time (Wartezeiten), Turn Around Time (Dauer vom Eingang einer Anforderung in der Organisation bis zu deren Einsatz) und viele mehr.

3. Transparenz:

Basierend auf greifbaren, erledigten Product Backlog Elementen, kurzen Zyklen, Zusammenarbeit, gemeinsamen Definitionen und angstfreien Arbeitsumgebungen.

4. Mehr mit weniger (more with less):

(1) Im Zuge der empirischen Prozesskontrolle: Mehr Lerneffekt durch weniger definierte Prozesse. (2) Durch Lean Thinking: Mehr erzeugter Nutzen durch die Vermeidung von "Waste" und "Overhead". (3) In der Skalierung: Mehr Eigenverantwortung, Zweck und Freude durch weniger Rollen, Artefakte und Spezialgruppen.

5. Das Produkt im Vordergrund:

Ein einziges Product Backlog, ein einziger Product Owner, ein potentiell auslieferbares gemeinsames Produktinkrement, ein gemeinsamer Sprint — unabhängig davon, ob es 3 oder 33 Teams gibt. Kunden wollen das Produkt, nicht nur Teile davon.

6. Kundenorientierung:

Identifikation von Wert und "Waste" aus der Sicht des zahlenden Kunden, z.B. Reduktion der Durchlaufzeit aus seiner Perspektive und mehr Feedbackschleifen mit dem echten Kunden. Jeder versteht, wie die tägliche Arbeit direkt mit dem zahlenden Kunden zusammenhängt.

7. Kontinuierliche Verbesserung bis hin zur Perfektion:

Erzeuge und liefere jederzeit ein fehlerfreies Produkt, das den Kunden absolut erfreut und ihm das Leben erleichtert. Führe jeden Sprint einfache und einschneidende Experimente durch, um Perfektion zu erlangen. Einschneidende Experimente sind sehr oft vom Umfeld abhängig. Häufige angewandte sind z.B. Pair Programming (paarweises Arbeiten an einem Computer), Test First (Erstellen von Tests vor dem Erstellen von Funktionalität/Source Code), Verzicht auf Branching (Erzeugen von temporären Versionen des Systems, die wieder zusammengeführt (merged) werden müssen).

8. Systemdenken (Systems Thinking):

Erkenne, verstehe und optimiere das ganze System (nicht einzelne Teile) und erforsche die Systemdynamik. Dazu ist es wichtig, dass z.B. die Scrum Master den Blick übergreifend und "von außen" behalten, um globalere Effekte

zu erkennen. Einfache Beispiele sind Rivalitäten zwischen Teams, unterschiedliche Planungsverhalten von Area Product Owners, Interventionen von Linienmanagern, die unter Umständen den Planungsprozess aushebeln.

Ein Konzept, das häufig in die Irre führt, ist das klassische Reporting, das ausschließlich auf Zahlen basiert. Bei Verwendung der herkömmlichen Formeln, die z.B. in der Earned Value Analysis angewandt werden und ohne Hinweis auf den Kontext können im schlimmsten Fall vollkommen falsche Bilder entstehen. Daher ist es wichtig, das System zu verstehen, die Werkzeuge, wie sie richtig funktionieren und möglicherweise adaptiert werden müssen. Vermeide lokale und Inseleoptimierungen z.B. mit dem irreführenden Fokus auf der "Effizienz" und "Produktivität" von einzelnen Personen oder einzelnen Teams. Für Kunden zählen die gesamte Durchlaufzeit vom Konzept bis zur Inbetriebnahme, sowie der konstante Fluss, nicht die einzelnen Schritte.

9. Lean Thinking:

Erzeuge eine Organisation, in der Manager Systemdenken und Lean Thinking anwenden und lehren, sich selbst ständig verbessern, sowie **Go See** und **Gemba** praktizieren. Füge die beiden Säulen *Respekt für Personen* und *kontinuierliche Verbesserung* hinzu. Alles bis hin zur Perfektion.

10. Warteschlangentheorie (Queuing Theory):

Verstehe, wie sich Systeme mit Warteschlangen in R&D verhalten und nutze diese Erkenntnisse, um Warteschlangengrößen, Work-in-Progress Limits, Multitasking, Arbeitspakete und Variabilität zu managen. Typischerweise wird davon ausgegangen, dass in der Warteschlange "First-Come First-Served" gilt, der Wartezeitraum unendlich ist und dass es auch unendlich viele Elemente gibt, die in das System eintreten können. Wird davon abgewichen, wird dies in der Regel gesondert angegeben. Die wartenden Elemente werden von einer oder mehreren Bedienstationen bearbeitet. Neben der Zahl der Abfertigungsstationen ist die Dauer relevant, die die Abfertigung eines Elements in Anspruch nimmt. Hier kommt die Exponentialverteilung zum Einsatz.

Kurz zusammengefasst beobachte ich häufig, dass umso mehr Elemente in die Warteschlange kommen, umso geringer die Abarbeitungsgeschwindigkeit wird. Es kann zum totalen Stillstand kommen. Für genauere Erklärungen wird auf einschlägige Literatur wie z.B. *Queueing Systems: Volume I: Theory* von Leonard Kleinrock verwiesen.

! Wie erwähnt sind diese zehn Prinzipien für sich selbst schon recht umfangreich, allerdings reicht in den meisten Fällen ein grundlegendes Verständnis dieser Prinzipien. Der Einstieg in eine LeSS Adoption mithilfe dieser Prinzipien ist auch mit wenig Erfahrung möglich. Wer wenig Erfahrung mit den Prinzipien hat, sollte allerdings für geeignete Begleitung sorgen.

Neben den Prinzipien gibt es in LeSS Regeln. Diese betreffen die Struktur, das Produkt und den Sprint. Die folgenden Listen geben einen Überblick darüber.

Regeln in LeSS

Die Regeln in LeSS definieren, was LeSS ist und was nicht.

Regeln für die Organisationstruktur in LeSS

1. Entwerfe Organisationen mit "echten" Teams als fundamentale Bausteine.
2. Jedes Team ist (1) selbst-organisierend, (2) funktionsübergreifend, (3) alle arbeiten im selben Raum und sind (4) langlebig.
3. Die Mehrheit der Teams sind kundenorientierte Feature-Teams.
4. Scrum Master sind für eine gut funktionierende LeSS-Einführung verantwortlich. Ihr Fokus liegt auf den Teams, dem Product Owner, der Organisation und den Entwicklungsmethoden. Scrum Master fokussieren sich nicht nur auf ein Team, sondern auf die gesamte Organisation.
5. Die Rolle des Scrum Masters ist eine dedizierte Vollzeitrolle.
6. Ein Scrum Master kann ein bis drei Teams betreuen.
7. In LeSS sind Manager optional, aber wenn sie existieren, wird sich ihre Rolle sehr wahrscheinlich ändern. Ihr Fokus verschiebt sich vom Managen der täglichen Arbeit hin zur Verbesserung der Organisation und der Entwicklungsmethoden.
8. Die Aufgabe des Managers ist die Verbesserung des Produktentwicklungssystems durch die Anwendung von "Go See" [3] (Beobachte dort, wo die Umsetzung wirklich stattfindet, ohne zu überwachen) sowie der Ansätze "Stop & Fix" (Probleme lösen anstatt diese zu verwalten) und "Experimente über Konformität" ("Responding to Change over Following a plan – Agiles Manifest).
9. Die gesamte LeSS-Struktur muss für die Produktentwicklungsgruppe von Beginn an etabliert werden; das ist für eine LeSS-Einführung essentiell.
10. Für die restliche Organisation, die über die Produktentwicklung hinausgeht, sollte LeSS evolutionär eingeführt werden. Dies kann durch "Go See" erreicht werden, wodurch eine Organisation entsteht, in der Experimente und Verbesserungen zum Alltag gehören.

Regeln für das Produkt in LeSS

1. Es gibt einen einzigen Product Owner und ein einziges Product Backlog für das gesamte auszuliefernde Produkt.
2. Der Product Owner sollte nicht alleine an der Verbesserung des Product Backlogs arbeiten. Er wird dabei von mehreren Teams unterstützt, die direkt mit dem Kunden/den Benutzern und anderen Stakeholdern arbeiten.
3. Alle Priorisierungen nimmt der Product Owner vor, aber Abklärungen finden so viel wie möglich direkt zwischen den Teams und dem Kunden/den Benutzern und anderen Stakeholdern statt.
4. Die Definition des Produkts sollte so generell und endbenutzer-/kundenorientiert wie möglich sein. Im Laufe der Zeit kann die Definition wachsen. Breitere Definitionen sind bevorzugt. "Breiter" ist hier im Gegensatz zu "fein-granular" zu verstehen. Am Beispiel eines Fahrzeugs könnte man so z.B. das Navigationssystem als eigenes Produkt sehen. LeSS geht davon aus, dass es besser ist, das gesamte Fahrzeug als Produkt zu definieren.
5. Es existiert eine "Definition of Done (DoD)" für das gesamte Produkt und sie ist für alle Teams gleich.

6. Jedes Team kann seine eigene, stärkere "Definition of Done" besitzen, indem es die gemeinsame DoD um eigene Punkte erweitert.
7. Ziel ist es, die "Definition of Done" so weit zu verbessern, dass nach jedem Sprint – oder öfters – ein auslieferbares Produkt entsteht.

Regeln für den Sprint in LeSS

1. Es gibt einen gemeinsamen Sprint pro Produkt anstatt eines Sprints für jedes Team. Jedes Team startet und beendet den Sprint zur selben Zeit. Jeder Sprint bringt ein integriertes Gesamt-Produktinkrement hervor.
2. Sprint-Planung besteht aus zwei Teilen: Sprint Planning Teil 1 wird mit allen Teams gemeinsam durchgeführt, während Sprint Planning Teil 2 in der Regel von jedem Team selbstständig abgehalten wird. Für eng verbundene Product-Backlog-Elemente sollte das Sprint Planning Teil 2 für mehrere Teams an einem gemeinsamen Ort stattfinden.
3. Am Sprint Planning Teil 1 nehmen der Product Owner und die Teams oder deren Repräsentanten teil. Zusammen wählen sie die Product-Backlog-Elemente aus, die im kommenden Sprint von den Teams umgesetzt werden sollen. Zudem identifizieren die Teams Möglichkeiten der Zusammenarbeit und noch offen gebliebene Fragen werden geklärt.
4. Jedes Team hat seinen eigenen Sprint Backlog.
5. Während Teil 2 des Sprint Plannings entscheiden die Teams, wie sie die ausgewählten Product-Backlog-Elemente umsetzen werden. In der Regel geht es dabei um das Design und die Erstellung des Sprint Backlogs.
6. Jedes Team hält sein eigenes Daily Scrum ab.
7. Teams koordinieren teamübergreifend. Dezentralisierte und informelle Koordination ist der zentralisierten Koordination vorzuziehen. Lege Wert auf Kommunikation und informelle Netzwerke, wie z.B. Kommunikation über Code (Kommentare, Triggerpunkte und andere Information im Code hinterlegen für weitere Kommunikation), teamübergreifende Besprechungen, Mentoren für Komponenten, Reisende ("travelers", Personen, die keinem Team fix zugeordnet sind, sondern pro Sprint wandern), Kundschafter (Personen, die temporär für einen oder mehrere Sprints bei anderen Teams mitarbeiten, um zu lernen oder Wissen auszutauschen) und Open Spaces.
8. Product Backlog Refinement (PBR) wird von jedem Team für jene Product-Backlog-Elemente durchgeführt, die es wahrscheinlich in der Zukunft umsetzen wird. Multi-Team-PBRs und übergreifendes PBR dienen dazu, das gemeinsame Verständnis zu erhöhen und Möglichkeiten der Zusammenarbeit zu erkennen, wenn es stark zusammenhängende Product-Backlog-Elemente gibt. Zudem helfen Multi-Team-PBRs und teamübergreifende PBRs, wenn es einen Lernbedarf bzw. eine breitere Meinung gibt. Wer sich dabei mit wem trifft, ist stark kontextabhängig und kann von den Teams selbst organisiert werden. Scrum Master können unterstützend als Moderatoren helfen oder dabei unterstützen, geeignete Personen auszuwählen. Auch Stakeholder, Product Owner und andere Personen können an den PBRs ebenfalls teilnehmen, sofern sie einen Beitrag leisten können oder von der Teilnahme profitieren.

9. Es gibt gemeinsam für alle Teams einen **Sprint Review** für das Produkt. Es gilt sicherzustellen, dass die "richtigen" Stakeholder am Review teilnehmen, um die nötige Information für eine effektive Inspektion und Adaption einzubringen. Wer die "richtigen" Stakeholder sind, wird am besten über klassische **Stakeholder Management-Mechanismen** herausgefunden, wie z.B. Einflussmatrizen.
10. Jedes Team hält seine eigene Sprint Retrospektive ab.
11. Eine übergreifende Retrospektive wird im Anschluss an die teambezogenen Retrospektiven durchgeführt, um die teamübergreifenden und organisationsweiten Probleme zu diskutieren und Experimente zur Verbesserung zu erarbeiten. Daran nehmen der Product Owner, die Scrum Master, Teamrepräsentanten sowie Manager teil.

Einige dieser Regeln erscheinen schwierig umsetzbar. Allerdings zeigt sich bei genauerer Betrachtung meist, dass die Schwierigkeit oft nur darin liegt, Gewohnheiten aufzugeben. Dieser soziale Aspekt ist in jeder Adoption wichtig. Veränderungen bedeuten, sich mit Unbekanntem auseinandersetzen zu müssen, Komfortzonen verlassen zu müssen und Ängste zu überwinden. Klar ist auch, dass Ängste nur überwunden werden können, wenn man sich darauf einlässt. Daher muss bei jeder Adoption die Unterstützung in allen Bereichen der Organisation vorhanden sein.

Ausblick

Sie wissen nun, wie das „richtige“ Umfeld für eine erfolgreiche Adoption aussieht und kennen die Regeln für eine die Anwendung von LeSS mit bis zu acht Teams. Im zweiten Teil dieses Beitrags stelle ich Ihnen die Regeln für die größere Variante mit mehr als acht Teams vor und erkläre, wie Sie in sechs Schritten beide Varianten einführen können.

Literatur:

- Schwaber, Ken; Sutherland, Jeff: Scrum Guides, aktuelle Scrum-Guide-Version von Juli 2016 online einsehbar unter: <http://www.scrumguides.org/scrum-guide.html>
- Larman, Craig; Vodde, Bas: Go See
- Larman, Craig; Vodde, Bas: **LeSS – More with LeSS**, offizielle Homepage zum LeSS-Framework
- LeSS Austria Community auf Xing: <https://www.xing.com/communities/groups/less-large-scale-scrum-austria-1091492>
- LeSS Austria Community auf Meetup: <https://www.meetup.com/de-DE/Large-Scale-Scrum-LeSS-Austria/>

Large-Scale Scrum mit dem LeSS-Framework

Mit vielen Teams erfolgreich ein Produkt mit Scrum entwickeln

Teil 2: LeSS Huge-Regeln und Einführungsmaßnahmen

Die Erfinder von LeSS, Craig Larman und Bas Vodde, erkannten rasch, dass es natürliche Grenzen für Skalierungsansätze gibt. Will man einem einzigen Product Owner die Verantwortung für ein umfangreiches Produkt übertragen, so stößt man auf die kognitive Grenze der Übersicht. Das bedeutet, je größer ein Produkt ist und je komplexer die Teile davon sind, desto schwieriger wird es für eine einzelne Person, die Übersicht zu behalten. Somit wird es auch schwieriger, die Sortierung wertoptimiert durchzuführen.

LeSS Huge

LeSS soll das Minimum an Prozess darstellen. Die Koordination von mehr als acht Teams benötigt jedoch auch mehr Prozess. Statt LeSS mit optionalen Elementen zu "verunreinigen", entstand mit LeSS Huge ein zweites Framework für noch größere Vorhaben. LeSS Huge baut auf LeSS auf und definiert weitere Regeln, um sehr große Produktentwicklungsvorhaben agil umsetzen zu können. LeSS Huge wird als passend für Vorhaben mit mehr als acht Teams erachtet, wobei acht keine magische Zahl ist, sondern ein aus vielen Jahren Beobachtung resultierender Erfahrungswert der Autoren von LeSS.

Ebenfalls stößt man bei sehr umfangreichen Produkten an die Grenzen der effektiven Kommunikation. Wird ein Produkt z.B. von 120 Teams gleichzeitig entwickelt, sind gemeinsame Meetings aller Teams kaum mehr möglich und auch logistisch schwer durchführbar.

In der Praxis kommt es aber sehr selten vor, dass Produkte von mehr als 20 bis 30 Teams entwickelt werden. Ist das der Fall, kann über eine Gliederung des Produkts in mehrere Teilprodukte oder Produktlinien nachgedacht werden. Grundsätzlich geht LeSS bei der Produktdefinition von "je breiter desto besser" aus, allerdings muss Rücksicht auf die organisatorische Machbarkeit genommen werden.

Regeln in LeSS Huge

Regeln für Strukturen in LeSS Huge

1. Kundenanforderungen, die aus der Sicht des Kunden eine starke Gemeinsamkeit aufweisen, werden zu Anforderungsbereichen (Requirements Areas) zusammengefasst. Dadurch können Synergien und Fokus entstehen und bei sehr großen Produktentwicklungen die Area Product Owners dem Product Owner dabei helfen, den Überblick zu behalten. Ein Nachteil von Areas ist die lokale Optimierung, die automatisch durch die eingeschränkte Fokussierung eines Area Product Owners und den Feature Teams in der Area basieren.

Autor



Dr. Wolfgang Richter

Geschäftsführer von
JIPP.IT und Certified
LeSS Trainer

Kontakt: wolfgang.richter@jipp.it

Mehr Informationen unter:

› projektmagazin.de/autoren

2. Jedes Team spezialisiert sich in einem Anforderungsbereich. Teams bleiben in diesem Anforderungsbereich für eine lange Zeit. Wenn ein anderer Anforderungsbereich wichtiger wird, können Teams dorthin wechseln.
3. Jeder Anforderungsbereich hat einen Area Product Owner.
4. Jeder Anforderungsbereich hat vier bis acht Teams. Weniger bedeutet zu viel lokale Optimierung, mehr bedeutet möglicherweise einen Verlust des Überblicks durch den Area Product Owner. Vermeide es diese Größenangabe zu verletzen (Ausnahmen sind die Anfangsphasen eines neuen Anforderungsbereichs, wenn sich erst nach und nach Teams dafür bilden oder diese dorthin wandern oder wenn ein Anforderungsbereich das Ende seiner Sinnhaftigkeit erreicht hat und die Teams in andere Areas wandern).
5. LeSS-Huge-Einführungen, inklusive der strukturellen Veränderungen, werden mit einem evolutionären und inkrementellen Ansatz durchgeführt. Inkrementell bedeutet in diesem Fall auf den Erkenntnissen der vorigen Teams aufbauend und nach dem Eintreten von Stabilität in der umgestellten Gruppe.
6. Bedenke: LeSS-Huge-Einführungen benötigen Monate oder Jahre, unendliche Geduld und eine Menge Humor.

Regeln für Produkte in LeSS Huge

1. Jeder Anforderungsbereich hat einen Area Product Owner.
2. Ein (overall) Product Owner ist für die produktweite Priorisierung und die Zuteilung der Teams zu den Anforderungsbereichen verantwortlich. Er arbeitet intensiv mit den Area Product Ownern zusammen.
3. Die Area Product Owner agieren als Product Owner gegenüber ihren Teams.
4. Es gibt nur einen Product Backlog und jedes Product Backlog Element (Eintrag im Product Backlog) gehört zu genau einem Anforderungsbereich.
5. Es gibt einen Area Product Backlog pro Anforderungsbereich. Dieser Backlog ist nur eine detailliertere (gefilterte) Ansicht auf den Product Backlog.

Regeln für Sprints in LeSS Huge

1. Es gibt einen produktweiten Sprint, nicht einen unterschiedlichen Sprint für jeden Anforderungsbereich. Er resultiert in ein integriertes Gesamt-Produktinkrement.
2. Der Product Owner und die Area Product Owner stimmen sich regelmäßig ab. Vor jedem Sprint Planning stellen sie sicher, dass die Teams an den wertvollsten Product Backlog Elementen arbeiten. Nach dem Sprint Review ermöglichen sie weitere Adaptierungen auf Produktebene.

Die Regeln von LeSS Huge ergänzen die Regeln von LeSS. LeSS Huge stellt somit eine Erweiterung von LeSS dar.

Drei Ansätze für eine erfolgreiche Einführung

Um LeSS erfolgreich einzuführen, ist es wichtig, drei Ansätze zu berücksichtigen:

1. Deep and Narrow over Broad and Shallow

2. Top Down und Bottom Up

3. Use Volunteering

Deep and Narrow over Broad and Shallow

Bei diesem Ansatz geht es um die Selektion der Organisationseinheiten und Teams, die für die Adoption vorgesehen sind. LeSS Huge empfiehlt den Schritt-für-Schritt Ansatz im Gegensatz zum "Big Bang", bei dem das gesamte Unternehmen nach einer langen Vorbereitung umgestellt wird. Die Einführung von LeSS mit bis zu acht Teams sollte auf einmal durchgeführt werden, da die Anzahl der beteiligten Personen überschaubar ist.

LeSS Huge Adoptions beginnen mit etwa 50 Personen in der Produktentwicklung. Vorteilhaft ist, dass dabei die Umstellung trotz der Größe des Gesamtvorhabens überschaubar bleibt. Es können Erfahrungswerte für die weiteren Organisationseinheiten genutzt werden. Mögliche Schwierigkeiten können so vor dem nächsten Schritt bereits eliminiert werden.

Es gibt allerdings auch Nachteile dieser Vorgehensweise. Einer davon ist, dass die restliche Organisation nicht eingebunden ist und somit Gerüchte entstehen können. Diesem Nachteil kann man mit Transparenz in der Adoption entgegenwirken.

Ein weiterer Nachteil ist, dass für die Zeit der Adoption parallele Strukturen existieren. Diese können inkompatibel sein und somit Probleme beim Zusammenspiel verursachen. Solche Punkte müssen während der Adoption berücksichtigt werden, damit die Adoption nicht durch vorherrschende Prozesse der noch nicht umgestellten Organisation verhindert wird.

Top Down und Bottom Up

Oftmals geht die Initiative für Adoptions von einzelnen Teams oder Abteilungen aus. Dieser Bottom Up Ansatz ist deshalb riskant, weil die Entscheidungsebenen für Organisationsdesign und -entwicklung, also die höheren Managementebenen (sofern solche existieren), nicht eingebunden sind oder nicht hinter der Veränderung stehen. Eine Adoption bedeutet Veränderungen in allen Ebenen, Strukturen und Prozessen. Findet die Adoption Bottom Up statt, kann es passieren, dass die Management-Ebene blockiert sobald sie von der Veränderung betroffen ist. Im schlimmsten Fall resultiert das in einem undefinierten Zustand der Organisation mit negativen Effekten.

Eine befohlene Adoption, also Top Down, ist ebenso riskant. Oft werden Managemententscheidungen von "unteren" Ebenen skeptisch betrachtet. Die Mitarbeiter folgen zwar den Anweisungen, stehen aber nicht dahinter und entwickeln somit auch nicht die notwendigen Initiativen für kontinuierliche Verbesserung oder ständiges Lernen. Im Englischen spricht man dabei von "owning vs. renting". Die Mitarbeiter "mieten" nur die Auffassung ihrer Managementebenen, das "buy-in", also das Einkaufen und Besitzen findet nicht statt.

Ideal ist aus LeSS-Sicht eine von allen Ebenen unterstützte Adoption. Der dritte Ansatz (Use Volunteering) bei Adoptions hilft dabei.

Use Volunteering

Die Freiwilligkeit kann auf verschiedene Arten und Weisen unterstützt werden. Z.B. kann das Management zwar die Adoption forcieren, aber die Teams selbst entscheiden lassen, wer wann dabei sein will. Die Rahmenbedingungen müssen allerdings klar sein, d.h. wie viele Personen zusammenkommen sollten. Wenn nur ein Team sich freiwillig meldet, wäre die erste Adoption-Phase wenig aussagekräftig.

Eine andere Art der Freiwilligkeit ist, dass die Teams sich selbst finden dürfen. Sogenannte "Feature Team Finding Workshops" können sehr rasch, innerhalb von ein bis zwei Stunden, die neuen Teamstrukturen ergeben. Dabei können die Mitarbeiter selbst herausfinden, welche Skills in einem Team notwendig sind um Features (Funktionalität, Anforderungen) vollständig umsetzen zu können und sich dementsprechend selbst aufstellen.

Maßnahmen, um LeSS und LeSS Hufe einzuführen

Die in LeSS vorgeschlagene Reihenfolge an Maßnahmen zur Einführung sieht folgendes Vorgehen vor:

1. Schule jeden
2. Definiere das Produkt
3. Definiere "done"
4. Habe passend strukturierte Teams
5. Nur der Product Owner vergibt Arbeit für die Teams
6. Halte Projektmanager von Teams fern

Schule jeden

Ein extrem wichtiges Element bei einer Adoption ist Verständnis. Wird von Personen verlangt, dass sie nach LeSS oder Scrum arbeiten, ohne die notwendigen Schulungen erhalten zu haben, wird die Adoption sehr schleppend vorangehen oder scheitern. Daher muss als Vorbereitung für jede Adoption in die Ausbildung der Mitarbeiter investiert werden. Der Return of Invest (ROI) ist hoch, wenn gleich zu Beginn geschult wird. Wird erst spät in die entsprechenden Ausbildungen investiert, reduziert sich der ROI, bzw. kann es dazu kommen, dass oft und viel nachgebessert werden muss.

Definiere das Produkt

LeSS bedingt eine klare Definition des Produkts. Grundsätzlich wäre es ideal, das Produkt so breit wie möglich zu definieren. Allerdings gibt es natürlich Grenzen für die Definition. Das Universum als Produktdefinition würde ein Unternehmen einfach gesagt überfordern und zu abstrakt sein. Daher gibt es in LeSS für das Finden der Produktdefinition sog. "erweiternde und einschränkende Fragen". Diese Fragen sollte sich vor allem der Product Owner stellen. Allerdings ist es gut, wenn er seine Sichtweise durch andere überprüfen lässt bzw. die Definition mit anderen erstellt.

Erweiternde Fragen dienen dazu, um das Produkt so breit wie möglich werden zu lassen. Potenzielle Möglichkeiten, wie der Zukauf von Leistungen von Zulieferfirmen, sind dabei erlaubt. Die Herausforderung bei dieser Übung ist es, sich durch die gewohnte Umgebung und die Angst vor der Größe nicht einschränken zu lassen. Je kreativer die Lösungen sind, desto besser.

Erweiternde Fragen sind:

- Wer sind die eigentlichen Endkunden und was glauben diese, was das Produkt ist?
- Was ist das ursprüngliche Problem, das mit dem Produkt gelöst werden soll?

Zur Beschränkung auf die sinnvoll anwendbare Produktdefinition dienen die "einschränkende Fragen". Diese helfen, die Grenzen zu finden, innerhalb derer für die Organisation noch reelle Steuermöglichkeiten für das Produkt existieren.

Einschränkende Fragen sind:

- Was geschieht innerhalb unserer Organisation?
- Was ist unter unserer Kontrolle?
- Wie sieht die klare Vision für unser Produkt aus?

Warum nicht gleich mit einer bestehenden Produktdefinition beginnen? Aus der Erfahrung heraus ist diese oft zu systemlastig, zu komponentenorientiert oder zu technisch. Oft wird die Systemarchitektur als Grundlage für die Produktdefinition herangezogen, wie z.B. Schnittstellen oder Komponenten, die Regeln verwalten und ähnliches. In LeSS ist es wichtig, die Kundensicht in den Vordergrund zu stellen. Dennoch sollen Synergieeffekte auf Entwicklungsseite genutzt werden, d.h. die Wiederverwendbarkeit von Funktionalität und/oder Source Code über Module und Systemkomponenten, sowie Feature Teams und Requirements Areas hinweg ist gewünscht.

Das Aufbrechen von bestehenden Mustern ist dabei oft notwendig. Es gilt auch, herauszufinden, wer der eigentliche Product Owner ist und dafür muss die Produktdefinition klar sein. "Eigentlich" bedeutet hier, dass oft sogenannte "Proxy Product Owner" als Product Owner gesehen werden, also Personen, die nur in Vertretung für denjenigen agieren, der letztendlich das Hauptinteresse am Produkt hat. Dadurch entsteht das Risiko, dass Entscheidungen durch den eigentlichen Product Owner oftmals spät revidiert werden und Arbeit wiederholt werden muss.

Definiere "done"

Bereits in Scrum ist die Definition von "done" (fertig) essentiell. Die "Definition of Done" (DoD) ist das wichtigste Werkzeug, um Fortschritt erkennen zu können. Nur wirklich fertiggestellte Teile zählen zum Fortschritt. Fast fertige Teile können zwar unter Umständen nützlich sein, beinhalten allerdings Technische Schuld und verringern das Guthaben auf dem Konto ständig. Um das zu vermeiden, sollte die DoD so restriktiv wie möglich sein.

Habe passend strukturierte Teams

Es geht dabei nicht um das Anwerben von neuen Mitarbeitern, sondern um die geeignete Zusammenstellung von Teams. Der Idealzustand aus agiler Sicht ist, dass sich die Teams selbst formen. Gewisse Rahmenbedingungen können dabei natürlich vorgegeben werden. Prinzipiell sollten Teams cross-funktional, heterogen und langlebig sein. Cross-funktional bedeutet, alle Skills sind vorhanden, um Funktionalitäten End-to-End fertigstellen und liefern zu können. Heterogen bedeutet, es gibt verschiedene Charaktere im Team. Langlebig sind Teams, wenn die Personen beständig im Team zusammenbleiben – idealerweise über Jahre hinweg, unabhängig von Projektlaufzeiten.

Um von Teams sprechen zu können und die Vorteile von Teams nutzen zu können, ist Kollokation (das Zusammenarbeiten in einem Raum) notwendig. Eine Wand zwischen Teammitgliedern reicht schon aus, um gewisse Vorteile außer Kraft zu setzen und anstatt eines Teams eine Arbeitsgruppe zu haben. Echte Teams können um ein Vielfaches performanter sein als Arbeitsgruppen.

Nur der Product Owner vergibt Arbeit für die Teams

Diese Regel spricht für sich selbst. Sie wird leider oft als "der Product Owner weist dem Team Arbeit zu" missverstanden. Die Teams können in LeSS wie in Scrum weiterhin selbst die gereihten Backlog Items ziehen, allerdings obliegt die Entscheidung, ob etwas in das Product Backlog kommt und in welcher Reihenfolge, einzig und alleine dem Product Owner.

Halte Projektmanager von Teams fern

Diese kritisch wirkende Regel hängt mit der vorigen zusammen. In traditionellen Organisationen gibt es sehr oft Projektmanager pro Projekt, dazu Teilprojektleiter und vielleicht noch andere Managementfunktionen, die alle- samt wichtige Aufgaben erledigt haben wollen. Die Entscheidung, welche dieser manchmal konkurrierenden Auf- gaben durchgeführt werden und in welcher Reihenfolge, wird häufig auf Entwickler abgewälzt. Häufige Eskalatio- nen an der falschen Stelle sind oft das Ergebnis.

Daher definiert in Scrum und LeSS einzig und alleine der (overall) Product Owner die Reihenfolge. Den Input für seine Sortierung bekommt er von vielen Stellen, unter anderem von Stakeholdern wie z.B. Projektleitern, falls solche existie- ren. LeSS ist für die Produktentwicklung optimiert und sieht Projektleiter daher als nicht notwendig an. Falls in einer Umgebung Projektleiter existieren, so sind diese Stakeholder und den Teams gegenüber nicht direkt weisungsfähig.

Diese sechs Maßnahmen helfen, eine Adoption erfolgreich durchzuführen. Sie stellen keine vollständige Liste dar, sondern dienen als Orientierungshilfe, was in einer LeSS Adoption wichtig ist. Aus diesen Maßnahmen lassen sich weitere Maßnahmen, Regeln, Richtlinien und Strukturen ableiten. Sie helfen auch Freiräume zu definieren, um Selbstorganisation und Selbstmanagement zu ermöglichen.

Fazit

Large-Scale Scrum definiert zwei Frameworks für die Skalierung oder bevorzugter Weise für die Vereinfachung von Organisationen: LeSS und LeSS Huge. LeSS bildet die Basis und ist in der Regel ausreichend für eine Produktentwick- lung mit bis zu acht Teams. Darüber hinaus wird LeSS Huge angewandt, in dem zusätzlichen Regeln definiert sind.

Zehn Prinzipien, wie z.B. Systemdenken, Transparenz und empirische Prozesskontrolle, bilden die Grundlage für beide Frameworks. Experimente und Anleitungen, die für das konkrete Umfeld angewandt werden können, ergänzen diese.

LeSS fordert für die Einführung die Ansätze "Deep and Narrow over Broad and Shallow", "Top Down und Bottom Up" und "Use Volunteering". Nur wenn alle im Boot sind, die Adoption freiwillig unterstützen und die Adoption überschaubar bleibt, kann der Erfolg garantiert werden.

Im Gegensatz zu anderen Skalierungsframeworks startet LeSS mit dem minimalen Satz an Struktur und Prozess und fügt bei Bedarf Weiteres hinzu. Dadurch kann man sehr rasch starten, muss nicht mühsam herausfinden, welche Prozesse zu viel sind und kann reale Erkenntnisse für die ständige Verbesserung einfließen lassen.

Wie in jeder Veränderung stellt auch die Einführung von Large-Scale Scrum eine Investition und Herausforderung dar. Es stellt allerdings auch eine Riesenchance für effizienteres, effektiveres und einfacheres Vorgehen dar.

Immer mehr Case Studies und Erfahrungswerte entstehen international und immer mehr Organisationen nutzen Large-Scale Scrum als das Skalierungsframework ihrer Wahl. Dadurch wird garantiert, dass aus der ständig wachsenden Community heraus Unterstützung generiert wird und die Optimierung von Skalierungen mit Large-Scale Scrum immer weiter voranschreitet.

Literatur

- Schwaber, Ken; Sutherland, Jeff: Scrum Guides, aktuelle Scrum-Guide-Version von Juli 2016 online einsehbar unter: <http://www.scrumguides.org/scrum-guide.html>
- Larman, Craig; Vodde, Bas: Go See
- Larman, Craig; Vodde, Bas: [LeSS – More with LeSS](#), offizielle Homepage zum LeSS-Framework
- [LeSS Austria Community auf Xing](#)
- LeSS Austria Community auf Meetup: <https://www.meetup.com/de-DE/Large-Scale-Scrum-LeSS-Austria/>

Erfahrungsbericht der Fiducia IT AG

Den Kulturwandel zum agilen Unternehmen aktiv gestalten

Seit 2012 setzt die Fiducia IT AG (siehe Infokasten) in der Anwendungsentwicklung vermehrt agiles Vorgehen ein und hat Agilität als Teil Ihrer Unternehmensstrategie verankert. Heute wenden 24 Teams im Bereich Anwendungsentwicklung agile Methoden an, das ist rund ein Viertel aller Entwicklungsteams. In anderen Unternehmensbereichen, wie z.B. dem IT-Betrieb haben sich erste agile Inseln gebildet. Ein "Center of Competence Agile" (CoC Agile) sorgt bereichs- und hierarchieübergreifend für die notwendige Unterstützung aller Stakeholder bei der Umstellung von traditionellen auf agile Methoden, im Folgenden "Agile Transition" genannt.

Autor



Oliver Fischer

Leiter des Center of Competence Agile bei der Fiducia IT AG in München

Kontakt: Oliver.Fischer@fiducia.de

Mehr Informationen unter:
projektmagazin.de/autoren

Agilität ist kein Selbstzweck, zu groß sind die kulturellen Veränderungen die damit einhergehen, als dass man Agilität einfach deshalb einsetzt, weil es gerade "en vogue" ist. Vor allem die kulturellen Veränderungen machen eine agile Transition sehr anspruchsvoll (s.u.) und führen oft zu ihrem Scheitern (VersionOne, 2014). Agilität ist primär eine Kultur und definiert Werte und Prinzipien für eine Zusammenarbeit bei der Erzeugung von Softwareprodukten. Populäre und weit verbreitete Frameworks wie Scrum unterstützen diese Kultur durch Prozesse, Rollen und Artefakte. Eine Einführung von Agilität, z.B. mit Scrum, wird letzten Endes nur dann erfolgreich sein, wenn es auf eine geeignete Unternehmenskultur trifft oder es gelingt, diese entsprechend zu verändern.

Die Fiducia IT AG mit Hauptsitz in Karlsruhe ist Dienstleister für Informationstechnologie in der genossenschaftlichen Finanzgruppe. Über 650 der knapp 1.100 deutschen Volksbanken und Raiffeisenbanken sowie gut 50 Privatbanken setzen auf die IT-Lösungen der Fiducia und vertrauen dem hohen Sicherheitsstandard des Karlsruher Rechenzentrums. Aktuell verwaltet die Fiducia fast 18 Millionen Kontokorrentkonten, verarbeitet jährlich 3,9 Milliarden Buchungsposten, betreut fast 100.000 Bankarbeitsplätze und gewährleistet an ihren 23.000 Selbstbedienungsgeräten auch die reibungslose Versorgung der Bankkunden mit Bargeld. Mit über 2.600 Mitarbeiterinnen und Mitarbeitern erwirtschaftete die Fiducia IT AG in 2013 rund 679 Mio. Euro Umsatz. Weitere Informationen unter www.fiducia.de.

Erste Schritte und Motivation für die Agile Transition

Erste erfolgreiche Ansätze mit agilem Vorgehen gab es bereits 2009 bei der Entwicklung des technischen Frameworks JBF (Java based Banking Framework). JBF ist die technologische Basis für alle Bankarbeitsplätze und Selbstbedienungsgeräte. Dadurch konnte mit Hilfe agilen Vorgehens für diese zentrale Technologie die Time to Market verkürzt, das Knowhow im Team verbreitert und die Code-Qualität verbessert werden. Die Initiative für einen Einsatz von Scrum ging dabei von den Entwicklern und der Projektleitung aus. Berichte über die damit gemachten positiven Erfahrungen, z.B. auf der Inhouse-Konferenz "JBFOne", verhalfen dem Thema Agilität in der Fiducia zu einer verstärkten Aufmerksamkeit.

Agiles Vorgehen als künftiger Standard für Anwendungsentwicklung

Im Jahr 2011 setzte ein Entwicklungsteam bei einem Innovationsprojekt erstmals Scrum auch bei der Umsetzung einer fachlichen, für die Fiducia strategisch wichtigen Anwendung ein. Insbesondere die Zusammenarbeit zwischen Auftraggeber und Entwicklungsteam, die Bereitstellung von Produktinkrementen in kurzen Zyklen und die Möglichkeit, auf Veränderungen und Erfahrungen rasch reagieren zu können, trugen zu einem positiven Projektverlauf bei und unterstützten die "Vermarktung" von Agilität im Hause. Aufgrund der positiven Erfahrungen entschloss sich im Jahr 2012 die Leitung des Bereichs Anwendungsentwicklung, agiles Vorgehen als zukünftigen Standard in der Entwicklung zu definieren.

Die mit der Einführung von Agilität auf Bereichsebene verbundenen Ziele waren die Erhöhung der Software-Qualität, die Verbesserung der Reaktionszeit auf sich ändernde Markt- und Rahmenbedingungen, Steigerung der Mitarbeiterzufriedenheit und eine Erhöhung der Attraktivität als Arbeitgeber. Zu diesem Zweck wurde zeitgleich das Center of Competence für agile Software-Entwicklung gegründet, welches, initial aus vier Mitgliedern bestehend, die Einführung agiler Methoden geordnet begleiten und Teams auf ihrem Weg zur Agilität unterstützen sollte.

Agilität als Bestandteil der Unternehmensstrategie

Im Jahr 2013 ging der Vorstand der Fiducia einen weiteren wichtigen Schritt auf dem agilen Transitionspfad, indem er die Unternehmensstrategie um agile Aspekte erweiterte und in ihr das Ziel einer agilen Organisation verankerte. Die Zielsetzung hierbei ist es, sich als Unternehmen insgesamt flexibler und agiler aufzustellen und so in sich zunehmend rascher ändernden Umfeldern besser reagieren zu können. Dabei ist nicht nur die Time to Market gemeint, auch dem demografischen Wandel soll damit Rechnung getragen werden. Absolventen von Universitäten und Hochschulen suchen vermehrt nach Arbeitsumfeldern, die eigenverantwortliches, selbstorganisiertes und durch wenig Hierarchie geprägtes Arbeiten bieten. Durch diese Anpassungen soll auch zukünftig die gute Wettbewerbsposition gesichert und ausgebaut werden.

Das Vorgehen – den Wandel agil durchführen

Den Wandel hin zu einer agilen Kultur führen wir in der Fiducia ebenfalls mit agilen Mitteln durch, d.h. wir gehen in Iterationen vor und verändern die Organisation auf diese Weise inkrementell. Regelmäßige Reflektionen des sog. Transition-Teams (s.u.) dienen einer Standortbestimmung und Anpassung des Veränderungsprozesses.

Nach unseren bisherigen Erfahrungen ist dies der einzig sinnvolle Weg, eine solche Veränderung durchzuführen. Ansätze, die eine Art Blaupause für eine agile Organisation vorgeben, passen nie wirklich auf das eigene Umfeld, sie müssen in jedem Fall angepasst werden. Es mag verlockend klingen, eine auf dem Markt angebotene Blaupause zu nehmen und einfach über die eigene Organisation zu stülpen. Man hat ein scheinbar klares Zielbild, man glaubt, man brauche nur noch die Prozesse und Rollen einzuführen und fertig wäre die agile Organisation. Agilität ist aber in erster Linie eine Projekt- bzw. Unternehmenskultur, Prozesse und Rollenbeschreibungen sind Ergebnis und nicht Ursache dieser Kultur. Auf diesen Aspekt gehe ich weiter unten im Abschnitt "Kultur" näher ein.

Das Transition-Team

Diesen Wandel nachhaltig zu gestalten und die Akteure dieses Prozesses konkret zu unterstützen ist Aufgabe des sog. Transition-Teams, dem auch das CoC Agile in leitender Funktion angehört. Das Transition-Team umfasst zwischen 15 und 20 Mitglieder, die z.T. in eigenen Teams organisiert sind. Die wichtigsten Rollen und ihre Verantwortungen sind im Folgenden beschrieben.

Product Owner Agile Transition

Der Product Owner der Agile Transition in der Fiducia, wahrgenommen durch den Leiter des CoC Agile, erstellt, pflegt und priorisiert das sog. Transition-Backlog (s.u.). Er sorgt dafür, dass die Inhalte und Priorisierung mit den Stakeholdern der Agile Transition abgestimmt sind und deren Interessen berücksichtigt sind. Gegenüber den Teammitgliedern fungiert er als Auftraggeber. Die Rolle "Scrum Master Agile Transition" wurde dagegen bislang nicht explizit besetzt, typische Aufgaben eines Scrum Masters werden durch sehr erfahrenen Teammitglieder wahrgenommen. Bei Bedarf werden punktuell externe Experten herangezogen, z.B. zur Durchführung von Retrospektiven.

Methoden- und Kernteam

Das Methoden- und Kernteam setzt sich zusammen aus fünf bis sieben erfahrenen, externen, agilen Coaches und internen Mitarbeitern, die ebenfalls schon Erfahrungen mit Agilität gesammelt haben. Aufgabe dieses Kernteams ist die Verbreitung des agilen Gedankenguts und seiner Werte sowie der darauf basierenden Praktiken und Methoden, z.B. Scrum. Das auf die beiden Standorte Karlsruhe und München verteilte Kernteam stimmt sich täglich in einem Online-Meeting ab und trifft sich monatlich persönlich, um Themen gemeinsam zu bearbeiten und Retrospektiven durchzuführen.

Coaches für Agile Software Engineering (ASE)

Ein wesentlicher Erfolgsfaktor für die Einführung agiler Methoden ist die Vermittlung eines geeigneten Handwerkszeugs für den Software-Entwicklungsprozess. Praktiken wie Continuous Integration (CI), Test Driven Development (TDD) und Pair Programming bilden dabei den Kern, der in Form von Trainings und Coachings durch ein Team von vier bis fünf Entwicklern an jedem Standort vermittelt wird. Nachdem in der Fiducia sowohl Java als auch COBOL als Programmiersprachen im Einsatz sind, ist neben einem bereits etablierten ASE-Ausbildungsprogramm (s. u.) für Java auch eines für COBOL in der Entstehung. Analog dem Methoden- und Kernteam treffen sich auch die ASE-Coaches in regelmäßigen Abständen persönlich, um Themen zu vertiefen und stimmen sich einmal pro Woche ab.

Sponsorengremium

Um den agilen Wandel nachhaltig umsetzen zu können, ist die Unterstützung aus dem Management einer der Erfolgsfaktoren. Das Sponsorengremium ist deswegen mit insgesamt vier Führungskräften der ersten und zweiten Führungsebene besetzt. Eine typische Aufgabe des Sponsorengremiums ist z.B. das Beseitigen von organisatorischen Hindernissen, welche nicht oder nur schwer auf Ebene der Einzelteams oder durch das CoC Agile beseitigt werden können.

Weitere, unterstützende Rollen kommen z.B. aus dem Multiprojektmanagement der Anwendungsentwicklung und dem Personalwesen.

Arbeitsweise und Werkzeuge des Transition-Teams

Die "Produkte", die dieses Transition-Team "erzeugt", sind im Wesentlichen Serviceleistungen, z.B. Trainings und Coachings für Einzelpersonen und Teams, das Initiieren und Moderieren von Communities of Practice und das Bereitstellen von Informationen zu agilen Methoden und Praktiken im Wiki oder durch Informationsveranstaltungen. Erzeugt werden aber auch Produkte wie ein agiles Vorgehensmodell, welches in einem Umfeld mit starker gesetzlicher Regulatorik Voraussetzung für einen Einsatz agiler Methoden ist.

Die Gruppen und Mitglieder des Transition-Teams erstellen diese Produkte in einer agilen Vorgehensweise, wie bereits die obigen Rollenbeschreibungen zeigen. Die wesentlichen Elemente dieser Vorgehensweise sind das Transition-Backlog, das Review- und Planning-Meeting sowie die Retrospektive.

Transition-Backlog

Das Transition-Backlog enthält alle für die Agile Transition notwendigen, priorisierten Maßnahmen in Form von Epics und User Storys, wie sie aus Scrum bekannt sind. Eigentümer des Transition-Backlogs ist der Product Owner Agile Transition (s.o.) Dieser ist verantwortlich für die Erstellung der Epics und User Storys und trägt dafür Sorge, dass die Maßnahmen immer aus Sicht eines Anwenders/Kunden der Agile Transition beschrieben und mit Erfolgskriterien versehen werden, um die Erledigung messen zu können.

Die folgenden zwei User Storys des Transition-Backlogs geben ein Beispiel dafür, wie mit diesem Vorgehen effizient der Veränderungsprozess vorangetrieben wird:

User Story: Moderner Führungsstil und Führungsinstrumente

Führungskräfte kennen moderne Führungsstile und haben ein Verständnis der derzeitigen bzw. gewünschten Kultur, um konkrete nächste Schritte ableiten zu können.

Erfolgskriterien:

- Führungsinstrumente aus Lean, Management 3.0, Radical etc. sind im Führungskreis vorgestellt.
- Eine Arbeitsgruppe aus Führungskräften und CoC Agile zur Weiterentwicklung des Führungsstils ist gebildet.

User Story: ASE-Coaching für Projekt XYZ

Mitglieder aus Projekt XYZ können ASE-Coaching abrufen, um gelernte Praktiken zu vertiefen.

Erfolgskriterien:

- Gemeinsame Retrospektive zur Mitte des geplanten Coachings hat stattgefunden.

- Vor und nach dem Coaching wurde eine Umfrage (zum Skill-Level) durchgeführt und ausgewertet.

Review- und Planning-Meeting

Sponsoren und Kernteam treffen sich alle vier Wochen, d.h. nach jedem Sprint. Im ersten rund einstündigen Teil des Meetings (Review) besprechen sie die Ergebnisse der vergangenen vier Wochen. Im zweiten Teil (Planning) planen und beschließen sie die Maßnahmen für den nächsten Sprint. Review und Planning werden unmittelbar hintereinander gehalten, das gesamte Meeting dauert maximal zwei Stunden. Dies ist notwendig, da die beteiligten Führungskräfte nur begrenzt verfügbar sind. Dementsprechend ist eine gute Vorbereitung der Epics und User Storys erforderlich.

Retrospektive

Sponsoren, Kernteam und wichtige unterstützende Rollen, insgesamt ca. 10 bis 15 Personen, treffen sich zweimal im Jahr, um über das Vorgehen und den Status des agilen Wandels im letzten halben Jahr zu reflektieren. Die Retrospektive wird von einem neutralen Moderator geleitet, der von außerhalb des Transition-Teams kommt. Zweck der Retrospektiven ist es, positive Entwicklungen zu erkennen und diese zu stabilisieren sowie Maßnahmen für eine weitere Verbesserung zu identifizieren. Die gemeinsam beschlossenen Maßnahmen werden dann in der Regel als neue User Storys in das Transition-Backlog aufgenommen. Die halbjährliche Retrospektive ist sehr wertvoll, sie

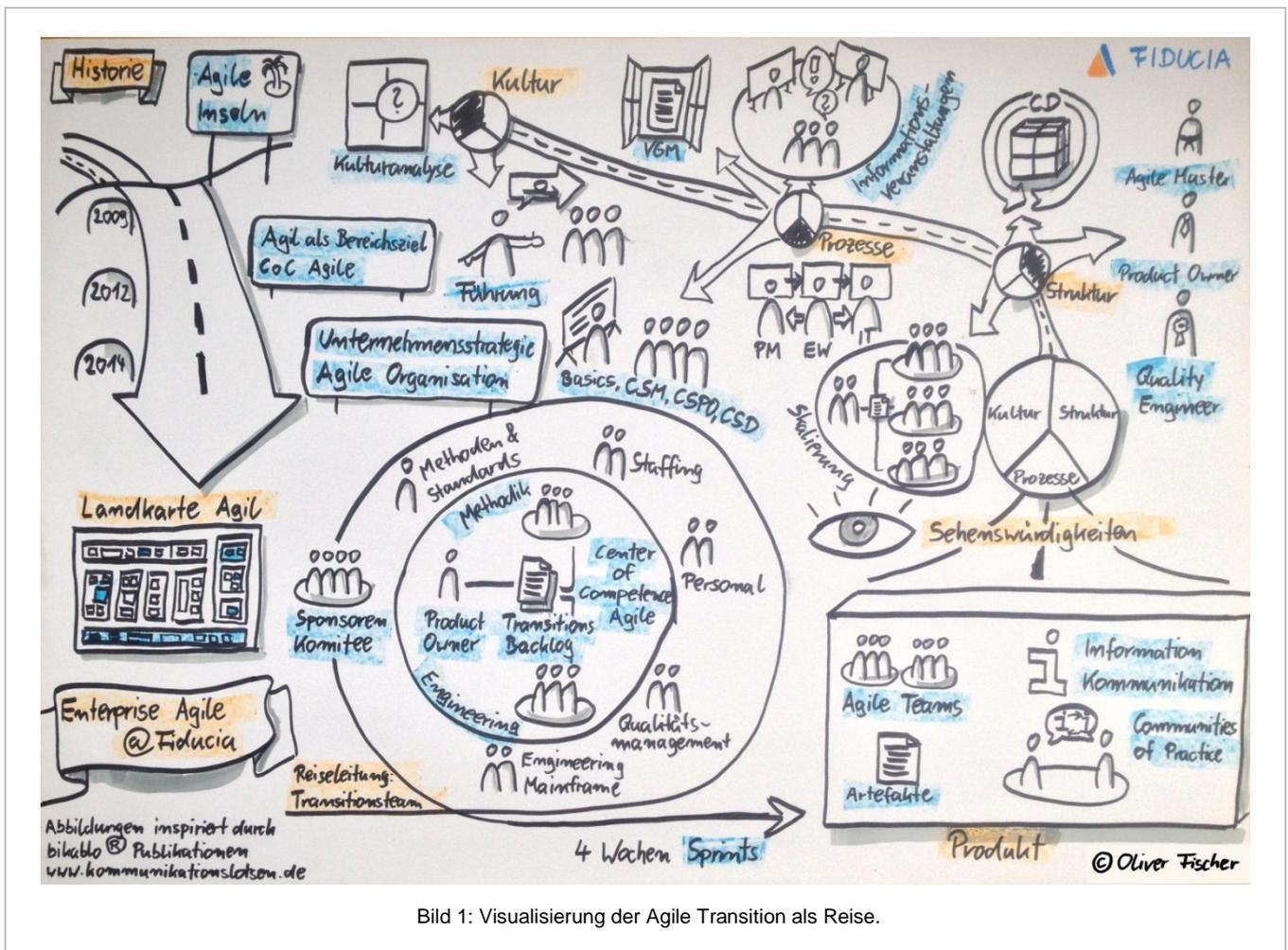


Bild 1: Visualisierung der Agile Transition als Reise.

bietet einen festen Rahmen, um losgelöst von aktuellen Themen den Veränderungsprozess reflektieren zu können.

Wesentliche Ergebnisse des Wandels – ein Zwischenbericht

Die Agile Transition ist noch lange nicht abgeschlossen. Von daher ist es nicht möglich, ein abschließendes Fazit zu ziehen. In gewisser Weise stellt die Agile Transition eine Reise dar, auf der man viele neue Erfahrungen gewinnt. Bild 1 illustriert den hier vorgestellten Prozess in diesem Sinne und soll ein wenig von dem Spirit vermitteln, der mit diesem Wandel verbunden ist.

Die im Folgenden dargestellten Ergebnisse beschreiben deshalb eher einen Zwischenstand mit erfolgreichen Lösungsansätzen aber auch mit nach wie vor offenen Fragen. Von den vielfältigen Aspekten greife ich die meiner Ansicht nach wichtigsten heraus: Agile Struktur, Prozesse und Kultur.

Agile Struktur

Die ersten agilen Teams in der Fiducia benutzten Scrum als agiles Framework. Inzwischen wird auch Kanban verwendet, Scrum hat aber nach wie vor einen sehr großen Anteil. Die mit Scrum kommenden, neuen Rollen richtig zu besetzen und in die Struktur und Prozesse der umgebenden Wasserfallorganisation einzubetten, ist eine der Herausforderungen in einem agilen Wandel.

Scrum Master

Der Scrum Master führt ein Team ohne disziplinarische bzw. fachliche Weisungsbefugnis, seine Rolle ist deswegen eher mit einem Coach zu vergleichen. Für traditionelle Projektleiter erscheint diese Rolle zunächst wenig attraktiv, da sie mit einem gewissen Machtverlust einhergeht. Wenn Projektleiter die Rolle des Scrum Masters besetzen, fordert dies von ihnen deshalb ein Umdenken in der Form der Teamführung. Fachliche Entscheidungen entfallen komplett, Führung durch Anweisung muss durch eine "dienende Führung" ersetzt werden, was gute Projektleiter allerdings oft ohnehin bereits tun.

Da die richtige Besetzung einer der Schlüsselfaktoren für einen erfolgreichen agilen Wandel ist, muss die Rolle des Scrum Masters entsprechend positioniert und bei der Besetzung auf die richtigen Skills geachtet werden. Mindestanforderungen an die Rolle wurden zu diesem Zweck definiert und in die im Haus eingesetzten Rollenbeschreibungen aufgenommen. Entsprechende Ausbildungsbausteine wurden im Seminarkatalog ergänzt. Zusätzlich werden neue Scrum Master intensiv durch einen erfahrenen Coach begleitet.

Ein Erfolgsfaktor ist es, die Bedeutung und Wichtigkeit der Rolle durch intensive Kommunikation mit den im Besetzungsprozess beteiligten Führungskräften entsprechend deutlich zu machen und ihr in der Organisation dadurch den richtigen Stellenwert zu verschaffen. Mit diesen Maßnahmen gelingt es zunehmend, die Rolle des Scrum Masters entsprechend der eines Projektleiters angemessen in der Organisation zu verankern.

Product Owner

Die zweite Schlüsselrolle bei Scrum ist der Product Owner, er führt das Team fachlich, indem er die Anforderungen seiner Stakeholder sammelt, nach ihrem Geschäftsnutzen priorisiert und sie dem Team durch kontinuierliche Kommunikation vermittelt. Er ist für den Produkterfolg verantwortlich. Ein Product Owner benötigt deswegen sowohl PM-Qualifikation (z.B. um seine Stakeholder oder sein Budget zu managen) als auch ein tiefgehendes Produktverständnis.

Bei der Fiducia fanden sich zu Beginn nur sehr wenige Personen, die beide Qualifikationen in ausreichendem Maß mitbrachten. Aus diesem Grund trennten wir den Product Owner in zwei Rollen auf: einerseits in den Chief Product Owner (CPO) mit dem Managementblick und in einer Art Außenministerrolle, andererseits in den Product Owner (PO) mit dem tiefen fachlichen Knowhow und in einer Art Innenministerrolle. Beide agieren als Team, ggf. sogar zusammen mit einem Mitarbeiter aus dem Bereich Produktmanagement.

Die Rolle des CPO wird in der Regel mit dem ehemaligen Projektleiter besetzt, die Rolle des PO mit dem für das jeweilige Produkt zuständigen Produktspezialisten aus der Entwicklung. Die Rollentrennung ist nicht unproblematisch und bringt im Alltag Hürden mit sich, die mehr oder weniger gut bewältigt werden können. Sind CPO und PO unterschiedlicher Meinung und können sich nicht einigen, so ist der Chief Product Owner letztendlich verantwortlich, im Konfliktfall zählt seine Entscheidung. Die Gefahr dabei besteht darin, dass er seine Entscheidungen eher aus einer kurzfristigen Projektsicht trifft, wodurch sich womöglich ein Spannungsfeld zur ganzheitlichen, langfristigen Produktsicht ergibt.

Team

Scrum-Teams sollen funktionsübergreifend besetzt sein, damit sie ihre Arbeit möglichst ohne Abhängigkeit zu anderen Rollen und Teams erledigen können. Sie sind als Team ganzheitlich dafür verantwortlich, dass am Sprint-Ende ein potentiell auslieferbares Produktinkrement bereit steht. Im Prinzip wird das heute im Haus bereits in den Wasserfallprojekten so gelebt, allerdings mit einem gravierenden Unterschied: jede Rolle ist immer nur für ihren Teilbereich verantwortlich, die Gesamtverantwortung wird selten wie beschrieben wahrgenommen. Zudem werden in großen Projekten oft Teams nach den Rollen zusammengestellt, es gibt dann z.B. zwei Entwicklungsteams und ein Testteam.

Bisheriges Rollenverständnis aufbrechen

Um dieses Rollendenken aufzubrechen und die Gesamtverantwortung mehr in der Vordergrund zu stellen, wurden in der Fiducia die einzelnen dedizierten Rollen und deren Titel und Beschreibungen, wie z.B. "Tester", in agilen Teams abgeschafft und alle Mitglieder des Teams als "Software Engineers" bezeichnet. Lediglich der weiter unten beschriebene "Quality Engineer" hat eine herausgehobene Rolle.

Natürlich leben gerade zu Beginn die alten Rollen weiter, eine Umbenennung alleine hilft noch nicht. Nach einiger Zeit war jedoch in jedem Team zu beobachten, dass die Teammitglieder nicht mehr nur ausschließlich in ihren ursprünglichen Rollen verblieben, sondern auch mal andere, rollennahe Aufgaben übernahmen. Typische Beispiele hierfür sind der Fachspezialist, der auch Testfälle beschreibt und testet, oder Entwickler, die durch das Schreiben von automatisierten Testfällen auch Aufgaben der Tester übernehmen. Schwierig bleibt es bei sehr spezialisierten Rollen, die oft nur für kurze Zeit in einem Team tätig sind (z.B. Designer von Benutzeroberflächen), so dass ein entsprechender Knowhow-Transfer nur schwer möglich ist.

Erfolgreiche Teams beibehalten: Die Arbeit zu den Teams bringen

Hier zeigte sich ein Erfolgsfaktor für die Bildung agiler Teams: die Teams bzw. deren Mitglieder müssen über einen ausreichend langen Zeitraum zusammenarbeiten, um aus einer Ansammlung von Spezialisten ein echtes, funktionsübergreifendes Team zu bilden. Neben dem wissensmäßigen Zusammenwachsen findet ebenfalls auf sozialer Ebene ein Zusammenwachsen statt, ein häufiger Wechsel von Teammitgliedern ist kontraproduktiv. Aus diesen Gründen findet in der Fiducia bei der Besetzung von Projektteams derzeit ein Umdenken statt. Anstatt einzelne Personen nach Projektende neuen Projektteams zuzuweisen, wird versucht, bestehende agile Teams stabil zu halten und ihnen weitere und zum Teil auch neue Aufgabengebiete zuzuweisen. Anstatt die "Menschen zur Arbeit zu bringen", versuchen wir somit die "Arbeit zu den Teams zu bringen".

Quality Engineer

Die beständige Steigerung der Software-Qualität ist eines der Ziele, die mit agilem Vorgehen erreicht werden sollen. Qualität am Ende der Wertschöpfungskette "hineinzutesten" gelingt jedoch selten, es ist besser, sie von Anfang an während des gesamten Produktionsprozesses umzusetzen und zu prüfen. Genau diesen Ansatz verfolgen wir mit der Rolle des Quality Engineers (QE). Er ist aber nicht allein verantwortlich dafür, dass Qualität "eingebaut" wird, dies würde dem Gedanken des funktionsübergreifenden, gesamtverantwortlichen Teams widersprechen. Die Verantwortung für ein qualitativ hochwertiges Software-Produkt verbleibt weiterhin vollständig beim Team. Der QE agiert vielmehr als "Anwalt" für qualitatives Software-Engineering im Team und gegenüber dem Product Owner. Die Rolle des Quality Engineers können Entwickler übernehmen, die Experten für Software-Engineering sind, die dafür benötigten Praktiken umfassend kennen, selbst anwenden und anderen vermitteln können.

In der Übergangsphase von traditionell arbeitenden Entwicklergruppen hin zu agilen, funktionsübergreifenden Teams kann die Rolle QE den Teammitgliedern helfen, agile Entwicklungspraktiken zu erlernen, zu vertiefen und auch die ersten Hürden zu überwinden, z.B. beim Aufbau einer Continuous-Integration-Umgebung. In Teams, die diese bereits beherrschen, bringt ein QE sehr wahrscheinlich keinen weiteren Mehrwert.

Continuous Delivery

Eine weitere strukturelle Veränderung auf dem Weg zur agilen Organisation ist das Bereitstellen einer Infrastruktur, die kontinuierliches Bauen (Build), Installieren (Deploy) und Testen von Software ermöglicht. In traditionellen Vorgehensmodellen werden z.B. Testumgebungen erst am Ende des Software-Entwicklungsprozesses benötigt und demzufolge auch erst dort bereitgestellt. Agile Teams benötigen jedoch ab dem ersten Sprint eine Umgebung, in der möglichst produktionsnah getestet werden kann. Ziel ist ja, am Ende eines Sprints ein potentiell auslieferbares Produkt zu haben, folglich muss auch der Test in einer entsprechenden Umgebung ermöglicht werden.

Manuelle Abläufe unterstützen das Bauen, Installieren und Testen nur unzureichend bzw. mit unverhältnismäßig hohem Aufwand, weswegen die Automation dieser Prozesse eine notwendige Voraussetzung für erfolgreiche agile Teams ist. Deshalb begannen wir sehr früh, die bestehenden und mit vielen manuellen Schritten durchgeführten Build-, Test- und Deploymentprozesse zu automatisieren.

Prozesse

Continuous Integration (CI) ist kein Tool, sondern vielmehr ein Prozess. Eine CI-Umgebung alleine reicht deswegen nicht aus, der eigentliche Prozess muss verstanden und angewendet werden, ebenso die weiteren, durch den Quality Engineer vorgelebten Entwicklungspraktiken. Diese zu vermitteln ist Teil des Trainings- und Coaching-Prozesses der Agile Transition.

Trainings- und Coaching-Prozesse

Basis des Trainings für die Agile Transition sind Grundlagenschulungen für alle Mitarbeiter zu Scrum, Kanban und zur Software-Qualität. Zusätzlich gibt es für alle beschriebenen Rollen im agilen Prozess entsprechende Schulungsbausteine, z.B. für die Scrum Master das Training zum Certified Scrum Master oder für den Product Owner zum Certified Scrum Product Owner. Ergänzt werden all die genannten Schulungen durch die Möglichkeit, sich erfahrene "Agilisten" aus dem Kernteam des CoC Agile als Coach zu holen.

Um langfristig erweiterbare und wartungsfähige Software-Produkte zu erzeugen, benötigen agile Teams ein hohes Maß an Software-Engineering-Knowhow. In der Fiducia wurde für die Vermittlung dieses Knowhows ein Weiterbildungs-Programm für Agile Software Engineering (ASE) aufgesetzt. Mit Hilfe eines externen Partners wurden eigene Software-Entwickler zu Coaches ausgebildet, welche zusammen mit den externen Trainern Praktiken wie CI, TDD oder Pair Programming an ganze Engineering-Teams vermitteln. In einer nachgelagerten Coaching-Phase helfen sie den Teams, diese Praktiken im Tagesgeschäft am realen Code anzuwenden. Sie arbeiten dabei sehr eng mit dem Quality Engineer zusammen, denn dessen Aufgabe ist es, die Praktiken nach der Coaching-Phase im Team aufrecht zu halten sowie sie kontinuierlich auszubauen und zu verbessern.

Teams, die auf agiles Vorgehen umstellen, bekommen dadurch einen ganzen Strauß an Maßnahmen geboten, um das notwendige Knowhow aufzubauen und es in der täglichen Praxis unter Anleitung zu vertiefen. Diese Möglichkeit ist ein großer Mehrwert und ein weiterer Schlüsselfaktor für die Agile Transition.

Ausdehnung im Wertschöpfungsprozess

Es sind sehr oft die Entwickler, die gerne mal agile Methodik ausprobieren wollen, ohne dass bereits eine Ausweitung auf das gesamte Unternehmen beabsichtigt ist. Agilität in der Entwicklung zu starten ist ein durchaus gangbares und mit überschaubaren Risiken verbundenes Muster, das man häufig antrifft.

Verbleibt Agilität jedoch rein in der Entwicklung – man spricht hier auch von "Agilität in der Blase" – während z.B. der Rest der Organisation weiterhin hierarchisch im Wasserfallmodell arbeitet, so wird der eigentliche Nutzen von Agilität für das Unternehmen nicht realisiert. Nur durch kurze Iterationen über die komplette Wertschöpfungskette und die frühzeitige Auslieferung von Inkrementen bekommt man schnell genug das notwendige Feedback vom Markt und kann entsprechend reagieren.

In der Fiducia versuchten wir deshalb bereits in den ersten agilen Projekten, das Produktmanagement so oft wie möglich in den agilen Prozess mit einzubeziehen, z.B. durch Aufnahme eines Produktmanagers in das Product-Owner-Team, die Teilnahme von Produktmanagern in Planning- und Review-Meetings. Trotzdem ist dies kein Er-

satz für einen über die komplette Wertschöpfungskette gespannten agilen Ansatz. Derzeit wird deshalb in einigen Pilotprojekten agiles Vorgehen auch bereits auf die frühen Wasserfallphasen ausgedehnt, um hier entsprechende Erfahrungen zu sammeln. Dazu wird in den Pilotprojekten schon vor dem eigentlichen Projektstart durch Produktmanagement und Anwendungsentwicklung gemeinsam ein Product Backlog erarbeitet, priorisiert und geschätzt. Auf diese Weise wollen wir den bestehenden Anforderungs- und Planungsprozess schrittweise flexibilisieren.

Vorgehensmodell

Unternehmen im Finanzdienstleistungssektor wie z.B. Banken und Versicherungen unterliegen starken regulatorischen Vorgaben durch den Gesetzgeber, Stichwort Basel III. Die Fiducia als IT-Dienstleister für Banken muss solche Vorgaben, z.B. der Bundesanstalt für Finanzdienstleistungsaufsicht (BaFin), zwingend erfüllen. Eine dieser Anforderungen ist es, sein Vorgehen für den Entwicklungsprozess zu dokumentieren und auch einzuhalten. Die in der Fiducia gelebten Prozesse sind deswegen in einem Prozess- und Vorgehensmodell beschrieben. Beim Umstieg auf agiles Vorgehen wurde sehr früh begonnen, agiles Vorgehen auch entsprechend zu beschreiben.

Eine der Herausforderungen dabei besteht darin, den Spagat zwischen der Freiheit, die agiles Vorgehen verlangt, und der Regulatorik unter einen Hut zu bringen. Die Beschreibung des agilen Vorgehens im Vorgehensmodell stützt sich deswegen auf wenige Prinzipien und Leitplanken und gibt vor allem die Artefakte vor, die aus regulatorischen Gründen zu erstellen sind. Der Weg zur Erstellung dieser Artefakte ist dadurch weniger eng vorgegeben als für das herkömmliche Vorgehen.

Mit dem Einsatz dieses schlanken Vorgehensmodells in der Fiducia beobachteten wir ein interessantes Phänomen: Die Mitarbeiter riefen nach mehr Vorgaben und Regelungen. Neue Freiheiten müssen auch verstanden und langsam wieder erlernt werden, eine Aufgabe die vom CoC Agile aktiv z.B. mit Informationsveranstaltungen oder im direkten Dialog angegangen wird.

Informationsprozesse

Die Zielsetzung und Bedeutung von Agilität zu kommunizieren und den Menschen im Unternehmen näher zu bringen ist eine ganz wesentliche Aufgabe bei einem agilen Wandel. Im CoC Agile werden ca. 30% der Kapazitäten für dieses firmeninterne Marketing verwendet. Dabei kommen verschiedene Formate und Kanäle zum Einsatz: Artikel in einem firmeninternen Wiki, regelmäßige Blogs, Informationsveranstaltungen mit interaktiven Bestandteilen, Vorträge auf Abteilungs- und Bereichsebene und auf dem hauseigenen IT-Kongress "JBFOne". Bestehende Gremien, vor allem der Betriebsrat, werden regelmäßig über neueste Entwicklungen informiert, Fehlinformation aufgrund von Gerüchten oder Unwissen kann so reduziert werden.

Um auch für die neu geschaffenen Rollen eine Austauschplattform zu bieten, etablierten wir verschiedene Communities of Practice. Hier treffen sich z.B. die Scrum Master mit dem Kernteam des CoC Agile und tauschen neueste Erkenntnisse aus. Bei Bedarf beschließen sie auch Maßnahmen für eine Verbreiterung des agilen Gedankenguts.

Um das Ziel "Erhöhung der Attraktivität als Arbeitgeber" zu erreichen, ist die Fiducia mit dem Thema Agilität auch auf öffentlichen Konferenzen präsent. Dabei bietet sich wiederum die Möglichkeit, mit anderen Firmen, die sich

gerade in einem ähnlichen Veränderungsprozess befinden, in Kontakt zu kommen und sich über verschiedene Lösungsansätze in einem agilen Wandel auszutauschen.

Ein agiler Wandel lebt, wie auch Agilität an sich, vom kontinuierlichen und intensiven persönlichen Austausch und der Interaktion der Betroffenen. Deshalb wurden im Jahr 2013 weit über 500 Mitarbeiter in mehr als elf Veranstaltungen über Agilität informiert. In der Annahme, alle Betroffenen nun ausreichend informiert zu haben, fuhren wir im Jahr darauf diese breit angelegten Veranstaltungen zurück. Plötzlich kam jedoch das Gerücht auf, das agile Vorgehen in der Fiducia sei eingestellt worden. Wir lernten daraus, dass zu wenig Information und Austausch schnell zu Falschannahmen führen kann.

Solche unbeabsichtigten Reaktionen, negativ wie positiv, lassen sich im Laufe des Veränderungsprozesses oft beobachten. Sie sind ein Zeichen für die Komplexität der Kulturveränderung und lassen sich nicht im Voraus planen. Die Aufgabe eines Transition-Teams besteht darin, den Veränderungsprozess aufmerksam zu beobachten und bei Bedarf diesen schnell und flexibel anzupassen.

Kultur

Agilität bedeutet eine neue Kultur im Unternehmen zu etablieren: eine tayloristische, auf Hierarchie, Steuerung und Kontrolle abgestützte Kultur wird ersetzt durch eine kooperative, selbstbestimmte und auf gegenseitigem Vertrauen basierende Zusammenarbeitsform. Das Verständnis für diese veränderte Kultur überhaupt herzustellen und Agilität nicht nur als einen Prozess mit anderen Rollen und Aufgaben anzusehen, ist die Kernaufgabe in einem agilen Wandel. In der Fiducia hat dabei das Schneider-Kulturmodell (siehe Infokasten) geholfen.

Schneider Kulturmodell

William E. Schneider beschreibt vier verschiedene Kulturtypen, von denen typischerweise eine primäre und eine sekundäre in jedem Unternehmen besonders stark ausgeprägt sind (Schneider, 1999). Diese vier Kulturtypen gemäß Schneider sind:

1. die Steuerungskultur, in der Unternehmen erfolgreich sind, indem sie Kontrolle und Steuerung erlangen und behalten
2. die Kompetenzkultur, in der ein Unternehmen erfolgreich ist weil es Mitarbeiter hat, die die Besten sind in dem was sie tun
3. die Wachstums- oder Kultivierungskultur, in welcher Lernen, persönliches Wachsen und die Sinnhaftigkeit im Vordergrund stehen
4. die Zusammenarbeitskultur, die durch Zusammenarbeit und Vertrauen den Unternehmenserfolg sicherstellt.

Keine der vier Kulturen ist besser oder schlechter als die anderen, vielmehr haben sie alle ihre Berechtigung für verschiedene Zielsetzungen. Agilität stimmt mit vielen Merkmalen der Zusammenarbeits- und Kultivierungskultur überein (Sahota, 2012), passt also besser zu Unternehmen deren primäre Kultur auch dort anzusiedeln ist.

Wie lässt sich die eigene Kultur nun ermitteln? Schneider hat dazu einen Fragebogen entworfen, anhand dessen man sehr einfach die Ausprägung der vier verschiedenen Kulturen im eigenen Unternehmen oder auch nur einem Bereich ermitteln kann. Wir führten diese Kulturanalyse im Kreis der Führungskräfte in der Anwendungsentwicklung durch, um in einem ersten Schritt zu erkennen, wo wir heute stehen und wo wir langfristig hin wollen. Die

eigene Kultur zu kennen ist eine wichtige Voraussetzung, um sie in Richtung einer agilen Kultur entwickeln zu können. Ein nächster Schritt kann deswegen sein, gezielt mit entsprechenden Maßnahmen die Kultur in einzelnen Bereichen zu verändern (z.B. beim Treffen von Entscheidungen). Zusätzlich ist das Vorleben der durch Agilität proklamierten Werte und Prinzipien durch Führungskräfte essentiell für den Kulturwandel.

Bedeutung von Führung im agilen Wandel

Zusammen mit einer Gruppe von Führungskräften begann das CoC Agile im Bereich Anwendungsentwicklung Prinzipien und Verhaltensweisen zu erarbeiten, die diesen Kulturwandel befördern und unterstützen. Tabelle 1 zeigt beispielhaft einige dieser ausformulierten Prinzipien und Verhaltensweisen. So wie sich Agilität in der Wertschöpfungskette nach vorne und hinten ausdehnen muss, um ihren wirklichen Nutzen zu entfalten, so gilt dies in gleichem Maße für die verschiedenen Hierarchiestufen in einem Unternehmen. Agilität nur auf Teamebene entfaltet sein Potential nicht wirklich, ein kultureller Wandel kann nicht eintreten, da spätestens im Umfeld des Teams die alte Kultur bestehen bleibt. In bestimmten Situationen widersprechen sich dann bestimmte Verhaltensmuster der beiden Kulturen und zielen vielleicht sogar in unterschiedliche Richtungen, wodurch Konflikte vorprogrammiert sind. Aussagen wie "Agil funktioniert bei uns nicht" sind die Folge und klare Anzeichen für den noch nicht vollzogenen Kulturwandel.

Da sich Kultur nicht durch Anordnung oder ein paar neue Rollen und einen veränderten Prozess ändern lässt, sondern nur durch Vorleben der mit der Kultur einhergehenden Werte und Prinzipien durch die Führung, ist bei einem agilen Wandel viel Geduld gefragt. Je weiter Ist- und Zielkultur auseinander liegen, desto länger wird auch der Übergang von der einen zur anderen Kultur dauern.

Prinzipien	Verhaltensweisen
Als Führungskraft ist mir klar, dass in einem agilen Unternehmen Führung Aufgabe jedes Mitarbeiters ist. Ich unterstütze die Selbstorganisation meiner Mitarbeiter und räume ihnen die notwendigen Freiräume ein.	Als Führungskraft lasse ich Mitarbeiter die Entscheidungen selbst treffen, über die sie besser informiert sind als ich. Wo das nicht möglich ist, binde ich Mitarbeiter in Entscheidungen mit ein. Damit die Mitarbeiter wissen, welche Entscheidungen sie selbst treffen können, habe ich ein Delegation Board (vgl. Apello, 2011) veröffentlicht. Als Führungskraft spreche ich meinen Mitarbeitern, auch und gerade in schwierigen Situationen, das Vertrauen aus.
Als Führungskraft lebe ich eine offene Fehlerkultur. Ich finde die Ursachen von Fehlern und verändere die Rahmenbedingungen so, dass sich die Eintrittswahrscheinlichkeit verringert.	Als Führungskraft mache ich meine Fehler und die Veränderungen, die sich daraus ergeben haben, öffentlich. Ich versuche, meinen Mitarbeitern die Angst vor einem Fehlschlag zu nehmen, indem ich Ihnen deutlich mache, dass Fehler zu einem Lernprozess dazugehören. Wenn mir Probleme zu Ohren kommen, informiere ich mich erst einmal in einem persönlichen Gespräch bei den direkt Betroffenen und biete gegebenenfalls meine Unterstützung an.
Als Führungskraft habe ich für mich und meine Mitarbeiter "Kontrolle" neu definiert.	Als Führungskraft habe ich mit meinen Mitarbeitern besprochen, in welchem Rahmen sie Entscheidungen selbst treffen können, und wann sie mich einbeziehen müssen. Als Führungskraft hole ich mir Informationen aus dem Tagesgeschäft bei meinen Mitarbeitern ab, damit diese die Informationen nicht mehr zu mir bringen müssen. Ergebnisse und Probleme aus den Projekten bekomme ich mit, weil ich regelmäßig an den Reviews teilnehme. Automatisch ermittelte KPIs sind für mich ein Anlass, um mit meinen Mitarbeitern in ein persönliches Gespräch über Verbesserungsmöglichkeiten einzusteigen. Dabei bin ich bereit, die Aussagekraft und den

	Nutzen dieser KPIs zu hinterfragen.
Als Führungskraft ist mir bewusst, dass Experimente und veränderte Verhaltensweisen auch zu Fehlern führen können.	Ich ermuntere meine Mitarbeiter, neue Dinge auszuprobieren oder die Dinge anders anzugehen. Dabei erwarte ich nicht, dass alles auf Anhieb funktioniert, sondern ich begleite und unterstütze meine Mitarbeiter bei dieser Veränderung.

Tabelle 1: Beispiele für Agile Führungsprinzipien und Verhaltensweisen.

Erfolge und Erfahrungen

Die Agile Transition in der Fiducia ist noch lange nicht zu Ende, vermutlich sind gerade einmal die ersten 20% geschafft. Deshalb ist auch noch keine abschließende Bewertung möglich. Doch bereits heute sind erste Erfolge in den Teams sichtbar:

- Die Transparenz ist gestiegen, auftretende Hindernisse können dadurch früher beseitigt und Missverständnisse in der Umsetzung von Anforderungen rechtzeitig korrigiert werden.
- Wenn die Teams tatsächlich selbstbestimmt ihre Arbeitsaufträge aus dem Backlog ziehen können und mit agilen Entwicklungspraktiken arbeiten, erhöht sich die innere und äußere Qualität der Software und auch die Zufriedenheit der Mitarbeiter verbessert sich.
- Die Produktivität nimmt ebenfalls dort zu, wo Teams sehr eng mit dem Auftraggeber zusammenarbeiten und das richtige Produkt selbstbestimmt richtig bauen können.
- Echte Teamarbeit nimmt zu, Wissensinseln werden dadurch verkleinert.
- Die Reaktionszeit innerhalb des Projektverlaufs verbesserte sich, gewonnene Erkenntnisse oder geänderte Anforderungen können schneller aufgenommen werden. Eine Verbesserung in der gesamten Wertschöpfungskette wird sich wie weiter oben erwähnt erst erzielen lassen, wenn Agilität auch dorthin entsprechend ausgedehnt wird.
- Ob sich die Attraktivität als Arbeitgeber verbesserte, lässt sich im Moment noch nicht beurteilen.

Neben den bereits heute spürbaren positiven Veränderungen lassen sich zum Abschluss fünf wesentliche Erfahrungen aus der Agile Transition zusammenfassen:

Agilität schafft Transparenz

Mit Agilität wird erst mal nichts "von alleine" besser, Agilität macht die bestehenden Probleme und Unzulänglichkeiten in der Organisation nur früher und deutlicher sichtbar. Diese Sichtbarkeit nicht dem veränderten Vorgehen zuzuschreiben oder sogar Schuldige zu suchen, sondern sie als Auftakt für kontinuierliche Verbesserungen zu sehen, ist Teil des notwendigen Kulturwandels.

Agilität ist eine Kultur

Die Einführung von Agilität bedeutet, sich in einen kulturellen Veränderungsprozess zu begeben. Die eigene, bestehende Kultur dabei zu kennen und gezielt Schritt für Schritt zu verändern, ist der zentrale Erfolgsfaktor eines agilen Wandels. Vorleben der neuen Kultur durch die Führungskräfte ist von entscheidender Bedeutung.

Es gibt keine Blaupause

Menschen und Unternehmen sind einzigartig, für einen agilen Wandel gibt es deswegen keine Blaupause. Modelle, die eine agile Organisation "out of the box" versprechen, können höchstens ein Ausgangspunkt für eine Veränderung sein. Seinen eigenen Weg Schritt für Schritt zu finden, dabei auch Rückschläge hinzunehmen und daraus zu lernen ist nach der bisherigen Erfahrung der einzige Weg zur agilen Organisation.

Ein agiler Wandel ist ein komplexes System

Ein kultureller Veränderungsprozess lässt sich nicht bis ins Detail im Voraus planen. Etliche der Veränderungen zeigen oft erst mal keine oder nur geringe Wirkung, an anderen Stellen zeigen sich dafür nicht beabsichtigte Effekte. Hier hilft iteratives, inkrementelles Vorgehen mit regelmäßigen Retrospektiven, um auf solche Effekte reagieren zu können.

Geduld, Geduld, Geduld!

Je weiter Ist- und Zielkultur auseinander liegen, je geringer eine Veränderungsnotwendigkeit im Unternehmen spürbar ist, desto länger wird der Wandel dauern. Geduld und ein dickes Fell, um Rückschläge auszuhalten, gehören zu den Eigenschaften, die alle an einem agilen Wandel Beteiligten mitbringen sollten.

Literatur

- VersionOne: 8th Annual State of Agile Survey, 2014, <http://www.versionone.com/pdf/2013-state-of-agile-survey.pdf>, zuletzt aufgerufen im April 2015
- Schneider, William E.: The Reengineering Alternative – A Plan for Making Your Current Culture Work, McGraw Hill, 2000
- Sahota, Michael: An Agile Adoption And Transformation Survival Guide, lulu.com, 2012
- Apello, Jurgen: Management 3.0: Leading Agile Developers, Developing Agile Leaders, 2011, ISBN-13: 978-0321712479

Erfahrungsbericht der Deutschen Telekom

Agile Transition – Führung im Wandel

Teil 1: Die neue Rolle der Führungskraft

Immer mehr Unternehmen setzen auf agile Methoden, wenn es darum geht, Software zu entwickeln oder komplexe Projekte zu managen. Dies stellt sowohl Führungskräfte als auch Mitarbeiter vor neue Herausforderungen. Verantwortung verteilt sich anders, die Arbeitsweise verändert sich grundlegend, alte Hierarchien und Prozesse greifen nicht mehr. Kurzum: die Einführung agiler Arbeitsweisen verlangt einen umfangreichen Change bei den Mitarbeitern, den Führungskräften, der Kultur und letztendlich der gesamten Organisation.

Doch während bei einer Agilen Transition der Fokus häufig auf den Mitarbeitern und den neuen Arbeitsmethoden liegt, kann die neue Rolle der Führungskräfte schnell aus dem Blickfeld geraten. Dabei müssen die Change-Verantwortlichen auch dem sensiblen Thema "Führung" von Beginn an viel Raum geben, da die Einführung agiler Arbeitsweisen oft eine erhebliche Beschneidung bisheriger Kompetenz- und Einflussbereiche der Führungskräfte bedeutet.

Dieser zweiteilige Beitrag beschreibt, wie bei der agilen Transition im Bereich "Digital Business Unit" der Deutschen Telekom AG auch diesem neuen Führungsverständnis Raum gegeben wurde. Der Artikel gibt Anregungen und Beispiele, wie agile Führung sich verändern muss, um die Transition hin zu einem agilen Unternehmen zu bewältigen und Sie erfahren, welche Maßnahmen zum Einsatz kamen sowie welchen Herausforderungen sich das Change-Team stellen musste.

Von hierarchischer Zusammenarbeit hin zu dynamischer Verlinkung

Wir sind es gewohnt, gerade in Konzernen und großen Unternehmen, dass Organisation hierarchisch ist. Selbst in moderneren Matrixorganisationen sind eine klare Zuordnung der Verantwortlichkeiten, strikte Berichtswege und definierte Prozesse Bestandteil des Organisationsverständnisses.

Nur: In einem agilen Arbeitsumfeld sind solche organisatorischen Vorgaben hinderlich und ersticken sämtliche agilen Ambitionen sofort im Keim. Es ist also wesentliche Aufgabe einer jeden agilen Transition, die bisherigen Organisationselemente dem agilen Arbeitsverständnis anzupassen. Doch wie dazu vorgehen?

Autoren



Andrea Maier

Dipl.-Betriebswirtin (FH),
Certified Scrum Master und
Product Owner,

Tätigkeitsschwerpunkte in agiler
Transition und agilen Methoden

Kontakt: a.maier@telekom.de



Bernd Klumpp

Dipl.-Ingenieur Nachrichtentechnik (FH), Certified
Scrum Master, Product

Owner und Scrum Professional, Leiter
des Bereichs "Agile Capability" bei der
Deutschen Telekom

Kontakt: bernd.klumpp@telekom.de

Mehr Informationen unter:
[projektmagazin.de/autoren](https://www.projektmagazin.de/autoren)

Anstatt nun teure Umorganisationen anzustoßen, die aufgrund der Dynamik des Marktes innerhalb weniger Monate wieder obsolet sind, gibt es auch die Möglichkeit, die Organisation eher als eine Art "Strukturrahmen" zu begreifen, deren Hauptaufgabe es ist, eine Basis für die agile Zusammenarbeit zu schaffen und Orientierung zu geben.

Eine solche Organisation lebt davon, sich zu aktuellen Fokusthemen dynamisch zu verlinken, um Ziele effizienter zu erreichen – also beispielsweise die Mitarbeiter über Silos hinweg in cross-funktionale, eigenverantwortliche Teams zusammenzubringen, damit Produkte und Projekte umgesetzt werden können. Ist dies erreicht, sind wir bei einem zentralen Kernpunkt des agilen Arbeitens: Aufbrechen von Hierarchien, Fokus auf die Bedürfnisse des Kunden und aktive Unterstützung und Förderung der Arbeit in eigenverantwortlichen Teams.

Wir müssen also die Menschen, die die Produkte und Services kreieren, in direkten Kontakt mit den Kunden und Geschäftsverantwortlichen bringen, egal wo sie in der Hierarchie angesiedelt sind. Das ist der Kern für die Entwicklung von Produkten und Lösungen, die den Kunden begeistern.

Kultur und Führungsverständnis müssen sich ändern

So weit, so gut: Doch damit dies überhaupt möglich wird, benötigt es auf der einen Seite ein Umdenken der Führungskräfte und auf der anderen Seite eine Unternehmenskultur, die eine solche Arbeitsform trägt. Führung in diesem Kontext verlangt die Offenheit der Führungskraft, dass sie "ihren" Mitarbeiter über die gesamte Wertschöpfungskette bei seiner Arbeit unterstützt. Sie muss ihn bei seiner Arbeit in diesen Teams fördern und dabei helfen, diese Verantwortung anzunehmen und zu leben. Bereichsdenken und wasserfallartiges "Über-den-Zaun-schmeißen" von Teilverantwortung muss dynamischer cross-funktionaler Kollaboration weichen. Und wenn dies vom Unternehmen gelebt wird, muss eine Reaktion auf Marktveränderung nicht zwangsläufig jedes Mal die nächste Re-Organisation bedeuten.

Bevor wir uns nun anschauen, wie wir Führungskräfte zu dieser neuen kollaborativen Grundhaltung bewegen können, betrachten wir zunächst, wie Kultur und Führungsarbeit im agilen Kontext überhaupt aussehen sollten.

Kulturwandel als unverzichtbares Fundament für agile Arbeitsweisen

Agiles Arbeiten ist sehr viel mehr als nur ein neues Prozess-Regelwerk oder ein neues Zusammenarbeitsmodell. Agiles Arbeiten erfordert insbesondere, agile Werte konsequent umzusetzen und zu leben. Dazu gehört u.a., dass

- offen und transparent kommuniziert wird,
- die Teams mehr Verantwortung übernehmen,
- Führungskräfte lernen loszulassen und zu vertrauen und
- das Management bereit ist, bestehende Prozesse zu ändern oder sogar abzuschaffen.

Deshalb müssen mit der Einführung agiler Arbeitsweisen auch das Bewusstsein und die Bereitschaft vorhanden sein, Rollen anzupassen und Verhalten zu ändern. Das funktioniert nur bei einer offenen, aufgeschlossenen Kul-

tur, die solche Veränderungen zulässt und fördert. Ist diese nicht vorhanden, ist eine Umstellung auf agile Arbeitsweisen – wenn überhaupt – nur sehr eingeschränkt und mit geringem Nutzen möglich.

Die Führungskräfte bei der agilen Transition abholen

Dieser notwendige Kulturwandel bedeutet eine große Herausforderung. Alte Denkmuster müssen aufgebrochen werden und Führungskräfte ihren Platz in der neu geschaffenen Organisationsstruktur finden. Da agiles Arbeiten auch bedeutet, dass Mitarbeiter bzw. Teams mehr Befugnisse erhalten, fallen diese zwangsläufig an anderer Stelle weg. Die Beschneidung bisheriger Kompetenzbereiche rüttelt aber bei vielen Führungskräften an ihrem Selbstverständnis.

Aus diesem Grund ist es zwingend erforderlich, im Zuge der Einführung agiler Arbeitsweisen auch entsprechende Initiativen zu starten, bei denen es um das neue Rollenverständnis geht und um die Art der künftigen Zusammenarbeit. Denn es fallen nicht nur Befugnisse und Aufgaben der Führungskraft weg, es kommen auch neue herausfordernde Anforderungen auf die Führungskraft zu. Um die ergriffenen Maßnahmen bei der Agilen Transition besser zu verstehen, betrachten wir zunächst, wie sich die Rolle der Führungskraft konkret verändert.

Agile Transition – wie sich die Rolle der Führungskraft verändert

Mit der Einführung agiler Arbeitsweisen ändert sich die Rolle der Führungskraft und "Führung" teilt sich auf verschiedene Rollen auf.

Disziplinarische Führungskraft

Die disziplinarische Führungskraft fokussiert sich darauf, dem Mitarbeiter bzw. dem Team die notwendigen Rahmenbedingungen für eine bestmögliche Wertschöpfung zur Verfügung zu stellen. Sie unterstützt das Team mit der Konzeption und Vermittlung der entsprechenden Ziele und der Strategie für die Aufgabe. Aber sie vertraut dem Team auch, dass es die besten Wege für gute Lösungen selbst findet. Sie unterstützt bei der Beseitigung von Hindernissen, wie z.B. fehlende Ausstattung, oder sorgt für eine Veränderung der Unternehmensprozesse bzw. -vorschriften, die im Weg stehen.

Der Idealzustand im agilen Kontext ist, dass die Führungskraft eine treibende Rolle innehat, mit der sie das richtige Kulturmilieu für agiles Arbeiten schafft, die agilen Werte schätzt und lebt sowie eine entsprechende Lernkultur fordert und fördert. Und, ganz wichtig, sie ist für die individuelle Weiterentwicklung des Mitarbeiters verantwortlich. Das heißt aber auch, dass sie den Kontakt durch regelmäßige Gespräche, durch Besuch von Meetings wie das Daily-Scrum-Meeting oder Review-Meeting suchen muss. Ein- oder zweimal im Jahr ein Mitarbeiter-Jahresgespräch zu führen, reicht dabei nicht mehr aus.

Was die Führungskraft nicht mehr machen soll, ist "Mikromanagement", d.h. sie soll sich aus dem Tagesgeschäft heraushalten und dem Team vertrauen, denn dort liegt die Kompetenz für die Lösung der Aufgaben.

Fünf neue Führungsprinzipien

Im Rahmen der Einführung agiler Arbeitsweisen im Bereich der "Digital Business Unit" der Deutschen Telekom AG entstand deshalb sehr schnell eine Initiative, bei der Führungskräfte aller Ebenen gemeinsam mit Mitarbeitern und dem Betriebsrat in mehreren Workshops herausarbeiteten, welches Führungsverhalten zukünftig weniger gewünscht und welches mehr gewünscht und erforderlich ist. Daraus resultierten fünf Führungsprinzipien, die mit Unterstützung der Geschäftsleitung unseres Bereichs als Rahmen für die zukünftige Führungsarbeit vereinbart wurden (s. Kasten und Bild 1).

Die fünf neuen Führungsprinzipien der "Digital Business Unit"

1. Vertrauen schenken: Meine Tür steht grundsätzlich offen. Ich besuche meine Mitarbeiter regelmäßig (mind. alle 14 Tage) an ihrem Arbeitsplatz und frage sie, wie ich sie unterstützen kann.
2. Transparenz herstellen: Informationen gebe ich zeitnah weiter und stelle die Bedeutung für meine Mitarbeiter heraus. Ich stelle regelmäßige Austauschmöglichkeiten sicher.
3. Verantwortung leben: Ich setze strategische Leitplanken und gebe Orientierung durch klare Ziele. Ich stelle sicher, dass jeder im Team seine Rolle und Verantwortung kennt und übernimmt. Dies erfolgt bei Bedarf, mind. jedoch halbjährlich.
4. Kunde zuerst: Ich stelle sicher, dass mein Team und ich uns regelmäßig mit Kundenbedürfnissen und Feedback – intern oder extern – beschäftigen und dieses berücksichtigen (direkten Kundenkontakt herstellen, andere Feedback-Kanäle auswerten usw.)
5. Machen – validieren – besser machen: Ich stelle sicher, dass meine Mitarbeiter über die Geschäftsergebnisse und die Performance meines Bereichs informiert sind.

Die Führungsprinzipien sollen helfen, das eigene Führungsverhalten zu reflektieren und daran ausgerichtet weiter zu entwickeln. Um die neuen Führungsprinzipien für alle öffentlich zugänglich zu machen, wurden Poster erstellt, die die Führungskräfte für alle sichtbar in ihren Räumen platzieren konnten (Bild 1). Und auf einem Riesenposter, das prominent in einer zentralen Kaffecke in unserem Haus hängt, konnten auch alle Führungskräfte und Mitarbeiter ihre Zustimmung zu den Führungsprinzipien durch eine Unterschrift dokumentieren – mehrere hundert Signaturen wurden somit verewigt.

Die weiteren Rollen

Was passiert aber mit anderen Führungsaspekten? Diese teilen sich im agilen Kontext auf weitere Rollen auf, nachfolgend dargestellt am Beispiel der bekanntesten agilen Methode "Scrum":

Der **Product Owner** gibt die Produktvision vor, mit der er dem Team ein attraktives Ziel aufzeigt und den Rahmen für die Arbeit definiert. Er führt das Team fachlich, indem er dem Team die Bedeutung des Marktes für das Produkt und umgekehrt die Bedeutung des Produkts für den Markt vermittelt. Er hat ein Augenmerk darauf, wie sich das Team im Hinblick auf fachliche Kompetenzen und Leistungsfähigkeit entwickelt, Er gibt vor, "was" entwickelt werden soll, vertraut jedoch auf die Lösungskompetenz im Team, die das "wie" entscheiden.

Der **Scrum Master** achtet auf Team-Prozesse, fördert die Selbstorganisation im Team und führt das Team lateral zu hoher Performance. Er beseitigt Hindernisse und sorgt dafür, dass das Team die optimalen Rahmenbedingungen zur Erfüllung seiner Aufgabe hat.

Und nicht zuletzt das **Team** bzw. jeder **Einzelne** übernimmt Verantwortung für sich selbst und die Arbeit, die er leistet. Der Grad der Selbstorganisation und Selbstverantwortung ist in einem agilen Kontext sehr viel höher. Erst das Zusammenspiel dieser verschiedenen Rollen führt zu guten Ergebnissen.

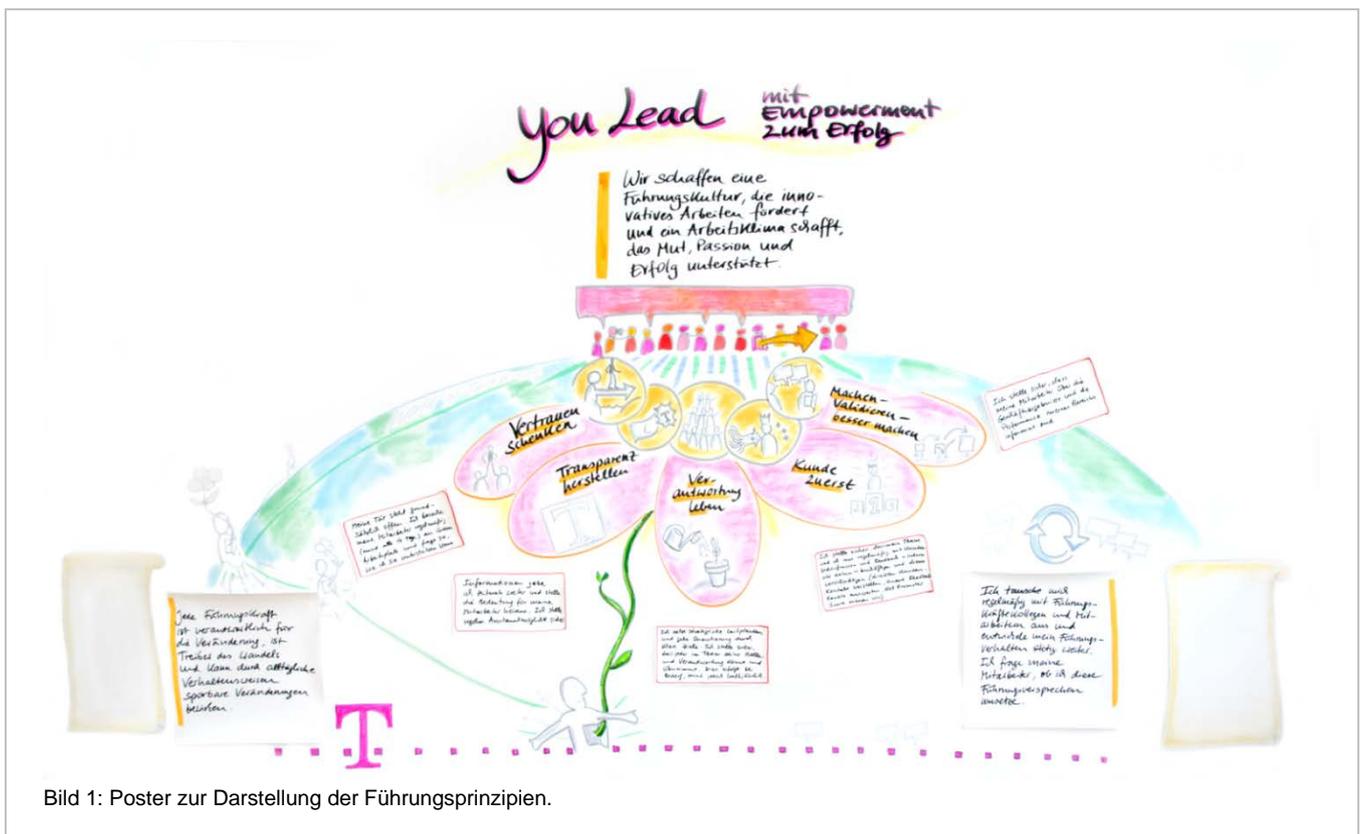


Bild 1: Poster zur Darstellung der Führungsprinzipien.

Wie spielen diese Rollen nun zusammen?

Die Führungskraft übernimmt zur Unterstützung dieser Rollen die Aufgabe des Coachs und Helfer bei Problemen. Ist die Führungskraft z.B. auch verantwortlich für die Geschäftsziele, unterstützt sie den Product Owner mit entsprechender Produktstrategie und gibt Feedback und Impulse für seine Arbeit.

Der Scrum Master, der auch für die Beseitigung von Hindernissen zuständig ist, braucht häufig die Hilfe von Führungskräften, um diese zu lösen. Bei einer disziplinarischen Führungsbeziehung kann die Führungskraft zudem den Scrum Master auch bei Methoden und Tools zur Förderung von Selbstorganisation und Teamdynamik beraten und coachen und ihn dabei in seinen Fähigkeiten entwickeln.

Gleiches gilt für die Führungskraft, wenn sie der disziplinarische Vorgesetzte eines oder mehrerer Teammitglieder ist. Denn Selbstverantwortung und Selbstorganisation stellen hohe Anforderungen an die Fähigkeiten von Mitar-

beitern, die bislang einen solchen Freiheitsgrad, und die damit einhergehende Verantwortung, vielleicht noch gar nicht erleben konnten. Die Verteilung der Führungsaufgaben am Beispiel von Scrum ist in Bild 2 zu sehen.

	Führungskraft <ul style="list-style-type: none">▪ Disziplinarische Führung▪ Mitarbeiterentwicklung und Coach▪ Ziele und Strategie▪ Schaffung der Rahmenbedingungen und Kultur		Product Owner <ul style="list-style-type: none">▪ fachliche Führung▪ Vermittlung der Marktrelevanz für das Produkt▪ „Anwalt“ des Kunden▪ unternehmerische Verantwortung
	Scrum Master <ul style="list-style-type: none">▪ laterale Führung▪ begleitet und unterstützt das Team bei seiner Entwicklung▪ fördert Selbstorganisation und kontinuierliche Verbesserung im Team zur Steigerung der Performanz		Team <ul style="list-style-type: none">▪ Selbstorganisiert und selbstkontrolliert▪ Übernimmt Verantwortung für die Umsetzung der besten Lösung für das Unternehmen▪ Eigenverantwortung der Entwicklung der Fähigkeiten im Team und steigert Teamperformanz

Bild 2: Verteilung der Führungsaufgaben am Beispiel Scrum.

Vorgehen bei der Agilen Transition

Im Folgenden beschreiben wir mit Fokus auf die Führungsarbeit, welche Erfahrungen wir bei der agilen Transition gemacht haben, auf welche Herausforderungen wir gestoßen sind und was bei der Transition gut lief bzw. was wir anders machen würden.

Ohne Rückendeckung von ganz oben läuft nichts

Erste agile Initiativen, die sich in den Entwicklungsteams "Bottom Up" in den Jahren 2008 und 2009 entwickelten, stießen schnell an Grenzen und versackten oft mit nur geringer Wirkung. Zwar ist es auf der einen Seite unserer Erfahrung nach relativ einfach, in kleinen Teams agile Methoden einzuführen. Schnell erkennen die Mitarbeiter die Vorteile, die ihnen agiles Arbeiten bringt, wie z.B. mehr Einfluss auf das eigene Tun, mehr Transparenz zu den eigentlichen Problemen, mehr Schlagkraft in der Umsetzung durch Cross-Funktionalität und den Freiraum, Dinge selbst anzupacken. Das motiviert!

Kommen jedoch auf der anderen Seite Hindernisse auf, die die Teams nicht selbst lösen können, sondern deren Entstehen außerhalb ihres Einflussbereichs liegt, wie z.B. unpassende Prozesse und Anforderungen oder man-

gelnde Unterstützung aus dem Rest des Unternehmens, das agiles Arbeiten nicht kennt, dann verebbt das Engagement schnell und es tritt Hilflosigkeit ein.

Erst das Umschwenken an der Spitze unseres Geschäftssegments und die klare Botschaft, dass wir in den relevanten Anwendungsfeldern agile Arbeitsweisen dringend benötigen, um unsere Geschäfte auch in Zukunft erfolgreich bewältigen zu können, gab dem Ganzen den notwendigen Schub und das erforderliche Management-Commitment von "oben".

Das Agile Transition Team

Durch Mandat der obersten Führungsebene wurde deshalb das "Agile Transition Team" gegründet. Dies setzte sich cross-funktional aus Führungskräften und erfahrenen Senior-Managern aus verschiedensten Bereichen zusammen. Die Kernaufgabe bestand darin, den Teams den Rücken freizuhalten und für sie Hindernisse aus dem Weg zu räumen, die durch das nicht-agile Umfeld entstehen. Also für die notwendige Veränderung der Rahmenbedingungen und Prozesse zu sorgen und somit die Transition voranzutreiben.

Wie hat sich das "Agile Transition Team" zusammengesetzt?

Das Management-Team der "Digital Business Unit", also die oberste Führungsebene, übertrug das Mandat zur Umsetzung der agilen Transition an einen Vice President, der bereits aus seiner vorherigen Tätigkeit Erfahrungen mit agilem Arbeiten und der damit verbundenen Transition mitbrachte. Er übernahm die Rolle des "Chief Product Owners" für das "Agile Transition Team". Unter seiner Federführung wurden weitere Führungskräfte und Senior-Manager aus allen Bereichen ausgewählt, die sich freiwillig gemeldet hatten, um an dem Thema mit 40% bis 100% ihrer Ressourcen mitzuarbeiten.

Bei der Auswahl ging es weniger um Erfahrung mit agilem Arbeiten, denn diese hatten wir zu der Zeit vor gut vier Jahren kaum. Es ging vielmehr darum, ein diverses, cross-funktionales Team mit Kollegen zusammenzusetzen, die gut vernetzt waren, ein hohes Ansehen im Haus genossen und die das Unternehmen gut kannten. Wir wollten eine kraftvolle Koalition ("Powerful Coalition") nach Dr. John P. Kotters "Leading Change" zusammenstellen, die um Experten, Trainer und Coaches aus externen agilen Beratungsfirmen ergänzt wurde.

Wie hat das Team gearbeitet?

Und von Anfang an war klar: Das "Agile Transition Team" arbeitet auch mit agilen Methoden, nämlich mit Scrum. Dies hat dazu beigetragen, dass die internen Führungskräfte und Senior-Manager mit Hilfe der externen Berater sehr schnell nicht nur den theoretischen Ansatz von Scrum gelernt haben, sondern ihn durch konkrete Anwendung erst so richtig erlernen und verinnerlichen konnten. Ganz nach dem Motto: "Eat Your Own Dog Food!"

Unter dem "Chief Product Owner" bildeten sich zwei Scrum-Teams. Ein Team hatte den Fokus auf der Wissensvermittlung zu agilen Methoden, also Training und Coaching – das "**Agile Coaching Team**". Das andere Team hatte den Fokus auf dem Treiben der agilen Transition selbst, also dem beseitigen von Hindernissen aus der Organisation heraus, dem Gestalten des Umfelds und der Prozesse, so dass agiles Arbeiten überhaupt möglich wird, und der Etablierung einer Unternehmens- und Führungskultur, die agiles Arbeiten nachhaltig unterstützt – dieses Team firmierte wei-

terhin unter dem Namen "**Agile Transition Team**". Mit der Etablierung der beiden Teams wurden dann auch die ersten agilen Projekte in einzelnen Geschäftseinheiten gestartet (s. Teil 2: Die ersten Schritte in die agile Welt).

Beide Teams arbeiteten mithilfe von Scrum entsprechend cross-funktional, transparent und kollaborativ zusammen. Es gab ein gemeinsames Backlog und die Sprints der Teams waren synchronisiert, um sicherzustellen, dass entsprechende Synergien optimal genutzt werden konnten. Und beide Teams arbeiteten auch gemeinsam an den Themen, die bei den agil arbeitenden Produktteams, wie z.B. bereits im Bereich "Mediencenter", gerade die höchste Priorität hatten in Bezug auf Wissensvermittlung und Beseitigung von Hindernissen.

Die Realität bestimmt den Plan

Das heißt, die Arbeit der agilen Produktteams aus den Geschäftseinheiten hatte das Backlog des "Agile Transition Teams" mit entsprechenden Anforderungen gefüttert. Oder anders ausgedrückt: Es gab keinen theoretischen "Change-Plan" mit Konzepten zur Umsetzung, möglichen Risikobetrachtungen und Meilensteinplanungen. Sondern das, was tatsächlich bei der Einführung von agilen Methoden an Themen und Problemen aus den Teams heraus gemeldet wurde, definierte die Priorität und den Inhalt der Arbeit des "Agile Transition Teams".

Es gab auch keine "klassische" Programmorganisation mit Steering Board, Gremienstruktur oder ähnlichem. Vielmehr wurden, ganz nach Scrum, Reviews der Arbeitsergebnisse in Review-Meetings vorgestellt, bei denen alle interessierten Kollegen und Führungskräfte aus der Organisation teilnehmen und Feedback geben konnten. So wurden der Arbeitsfortschritt und die Ergebnisse transparent für alle Mitarbeiter, Verbesserungsvorschläge konnten kurzfristig aufgenommen werden, Ergebnisse zur agilen Transition (z.B. Veränderung von Prozessen) konnten in kurzen Iterationen erzielt werden – Effekte, wie man sie aus der Software-Entwicklung mit agilen Methoden kennt, sind mit der Anwendung von Scrum als Management-Methode auch im Projekt der agilen Transition eingetreten.

Daneben war sichergestellt, dass der "Chief Product Owner" bzw. die Product Owner der Scrum-Teams im "Agile Transition Team" auch schnellen und direkten Zugang zum Management-Team hatten, sollten Probleme, Hindernisse oder Fragestellungen auftreten, die das "Agile Transition Team" nicht alleine lösen konnte.

Erfolgsfaktoren

Rückblickend können wir sagen, dass diese Herangehensweise ein voller Erfolg war. Ausschlaggebend dafür waren die entsprechende Offenheit aller Beteiligten, diese neue Herangehensweise mit Scrum zu respektieren und zu unterstützen sowie die Priorisierung des Themas als solches beim Management-Team.

Unterm Strich mündete dieses Vorgehen darin, dass nach gut zwei Jahren aus dem Projektansatz eine eigene Linienabteilung hervorgegangen ist, die jetzt mit Unterstützung der gesamten Organisation die agile Transition weiter treibt – namentlich das "Agile Capability Management", in dem wir, die Autoren, auch jetzt noch als Führungskräfte arbeiten.

Zusammenfassung

- Ein Commitment von "oben" ist wichtig – bottom up ist ein guter Anfang, reicht aber nicht aus für eine nachhaltige Umsetzung.

- Begleitung durch ein "Agile Transition Team" ist essentiell – es braucht Zeit für die Veränderung und die Teams können dies nicht alleine schaffen.

Ausblick

Das Management stand hinter dem Vorhaben und das Change-Team war aufgestellt... somit waren wir bereit, die ersten Schritte in die agile Welt zu wagen. Dass dabei natürlich nicht alles reibungslos verlaufen würde, war uns von vornherein klar – mit welchen konkreten Herausforderungen wir uns konfrontiert sahen und wie wir diese bewältigten, erfahren Sie im zweiten und abschließenden Teil.

Erfahrungsbericht der Deutschen Telekom

Agile Transition – Führung im Wandel

Teil 2: Die ersten Schritte in die agile Welt

Im ersten Teil haben wir gesehen, dass am Anfang eines agilen Veränderungsprozesses zum einen das Commitment von Seiten des Managements vorhanden sein sollte und zum anderen die Agile Transition mit einem kompetenten Team steht und fällt, das den Wandel von Anfang an begleitet. Ist das Team aufgestellt, kann es nun endlich losgehen – nur wie anfangen?

Statt viel Zeit mit der Erstellung eines theoretischen Konzepts und eines Projektplans für die Agile Transition zu verbringen (was ja eigentlich üblich ist, wenn man einen "Top-Management-Auftrag" erhält), hat man sich früh im Bereich "Digital Business Unit" dafür entschieden, "einfach mal anzufangen".

Die Umsetzung erfolgte entlang eines ersten Piloten in einem Produktteam aus dem Bereich "Mediencenter", einer Cloud-Lösung für Privatkunden der Deutschen Telekom zur Speicherung und Verwaltung von elektronischen Dateien wie Fotos und Videos. Die Kollegen wurden in Scrum geschult, es wurde ein Coaching-Pair aus einem internen Kollegen des "Agile Transition Teams" und einem externen agilen Coach gebildet. Und man hat einfach angefangen, agil zu arbeiten.

Probleme identifizieren und lösen

Sehr schnell tauchten die ersten Hindernisse auf. Einkaufsprozesse, die nicht zu agilem Arbeiten passten, Teamräume, die nicht zur Verfügung standen, Management-Reportings aus der klassischen Projektwelt, die nicht in Einklang zu bringen waren mit der agilen Vorgehensweise und vieles mehr.

Das "Agile Transition Team" hat sich darauf konzentriert, diese Probleme sukzessive abzarbeiten. Für ganz konkrete Fälle wurden Lösungen gefunden, die häufig der "Blueprint" für eine veränderte Arbeitsweise und Prozesswelt waren. Noch heute arbeiten wir mit dieser Vorgehensweise, um die agile Transition effizient und nachhaltig zu verankern.

So war beispielsweise die Projektsteuerung und Ressourcenallokation für unsere Produkt-Roadmap auf klassisches Projektmanagement nach Wasserfall ausgerichtet und zwar für Projekte, die Monate dauern konnten. Es gab eine klare Unterscheidung der Prozessschritte in

1. Definition der Anforderungen,
2. Prüfen auf Machbarkeit und Schätzen der Realisierungsaufwände,

Autoren



Andrea Maier

Dipl.-Betriebswirtin (FH),
Certified Scrum Master und
Product Owner,

Tätigkeitsschwerpunkte in agiler
Transition und agilen Methoden

Kontakt: a.maier@telekom.de



Bernd Klumpp

Dipl.-Ingenieur Nachrichtentechnik (FH), Certified
Scrum Master, Product

Owner und Scrum Professional, Leiter
des Bereichs "Agile Capability" bei der
Deutschen Telekom

Kontakt: bernd.klumpp@telekom.de

Mehr Informationen unter:

projektmagazin.de/autoren

3. Realisierung der Anforderungen und
4. Testen und Rollout.

Zu jedem dieser Prozessschritte gibt es eigene Anforderungen, die erfüllt sein müssen, und in entsprechenden Entscheidungsvorlagen offiziell genehmigt werden müssen. Also ein klassischer "Stage-Gate-Prozess".

Neue Wege finden und beschreiten

Dieses Vorgehen funktioniert allerdings in Scrum mit seinen kurzen Sprints nicht mehr, an deren Ende nach wenigen Wochen ein lauffähiges Produktinkrement steht, das beim Kunden ausgerollt werden kann – und danach wiederholt sich das Ganze mit neuen Anforderungen.

Genauso gibt es auch kein Lastenheft für die Anforderungen mehr, sondern ein Product Backlog, in dem nur die am höchsten priorisierten Anforderungen genauer beschrieben sind. Kurzum: der bislang genutzte Prozess passt hinten und vorne nicht mehr.

Gemeinsam mit dem Scrum-Team des Medienzentrums, dem "Agile Transition Team" und den Prozessverantwortlichen wurden Wege zur Lösung gesucht. Dazu musste bei den Prozessverantwortlichen zuerst einmal ein Verständnis für die Arbeit nach Scrum geschaffen werden. Die Kollegen zeigten dabei eine große Hilfsbereitschaft und Kreativität in der Lösungsfindung. Beispielsweise wurde das Scrum-Artefakt "Backlog" anstatt eines klassischen Lastenheftes akzeptiert. Und für die eigentliche Umsetzungsphase, die in mehreren Sprints über mehrere Monate laufen kann, musste nur noch eine initiale Gate-Freigabe erfolgen. Man vertraute darauf, dass während der Realisierung im Sinne der Produktvision und auf Basis des Product Backlogs das Team bestmögliche Leistungen bringen wird, die nicht kleinteilig kontrolliert werden müssen.

Und das Team hat das in sie gesetzte Vertrauen voll erfüllt. Die Entwicklungsergebnisse kamen schneller und in einer besseren Qualität. Damit war der Weg frei für weitere Teams, den für agiles Arbeiten angepassten "State-Gate-Prozess" zu nutzen.

Zusammenfassung

- Nicht lange planen, einfach anfangen – agil löst nicht die Probleme in unserem Unternehmen, aber agiles Arbeiten macht sie transparent und man weiß somit ganz genau, woran man am besten arbeiten sollte.

Welchen neuen Platz haben die alten Führungskräfte?

Wir konnten gut verfolgen, wie unsere Führungskräfte am Anfang auf "die agile Welle" reagiert haben. Neugier, Skepsis, aktives Interesse, totale Verdrängung, alles war zu sehen. Aber fast alle hatten eines gemeinsam: Den Führungskräften inklusive des Top-Managements und allen weiteren Beteiligten war am Anfang nicht bewusst, wie gravierend die Auswirkungen auf die Rolle der Führungskraft sein werden.

Man ging davon aus, "dass die Teams halt anders arbeiten". Doch agiles Arbeiten erfordert eine starke Veränderung der eigenen Arbeitsweise; Rollen und Aufgabeninhalte verändern sich. Auf einmal sollte über Abteilungsgrenzen hinweg gearbeitet und dabei ein völlig anderer Grad an Verantwortung übernommen werden.

Alles in allem also ein klassischer Change-Prozess, bei dem das Thema "Führung" enorm wichtig war – aber wir uns dessen noch gar nicht wirklich bewusst waren. Dieses Problem wurde jedoch schnell transparent. Wir erkannten, dass neben der reinen "Scrum-Framework-Schulung" für Vorgesetzte und Mitarbeiter auch Informationen und Workshops erforderlich sind, die explizit auf die neue Rolle der Führungskräfte abzielten.

Spezielles Training

Wir setzten deshalb ein speziell zugeschnittenes Training "Certified Scrum Master for Leaders" auf mit dem Ziel, den Schwerpunkt auf die Rolle der Führung und auf die Auswirkungen auf die Führungskräfte im agilen Kontext zu setzen. Alle Führungskräfte der ersten beiden Ebenen waren angehalten, den Workshop zu besuchen.

Sie durchliefen ein 2-tägiges Scrum-Master-Training, das alle relevanten Inhalte zum Arbeiten nach Scrum enthielt. Die Führungskräfte sollten genau verstehen, wie ihre Mitarbeiter jetzt in den Teams arbeiten und wie sich die Rollen und Aufgaben dabei verändern. Also genau die Inhalte, die auch in einem regulären "Certified Scrum Master Training" zu finden sind.

Ergänzt wurden diese Inhalte allerdings um spezielles Wissen zu ihrer veränderten Rolle als Führungskraft. Wie verteilen sich Führungsaufgaben, wenn nach Scrum gearbeitet wird? Welche neuen Skills und Tools benötigen sie? Und wie können sie ihre tägliche Führungsarbeit verändern, um die Teams zu unterstützen. In einer Mischung aus Methodenvermittlung, Reflektion der eigenen Arbeit und interaktivem Austausch untereinander sollten sie die für sie relevanten zukünftigen Inhalte und Aufgaben einer veränderten Führung erarbeiten.

Unsere Erfahrung: Was die persönliche Teilnahme an den Seminaren anging, stießen wir immer wieder auch auf Zurückhaltung. Nach den zwei Tagen gaben die Führungskräfte jedoch zum Abschluss ein positives Feedback, dass sie erst jetzt besser verstanden haben, was agiles Arbeiten und agile Werte bedeuten und wie sie selbst die Transition unterstützen können.

! Auch bei diesen Trainings gingen wir nach bewährter Manier im "Pair" vor: d.h., ein externer Experte und zertifizierter Scrum-Trainer hat gemeinsam mit einem internen Kollegen aus dem Agile Transition Team das Training für die Führungskräfte durchgeführt. Das stellte sicher, dass das vermittelte Wissen zu agilem Arbeiten, insbesondere Scrum, immer eng an unseren unternehmens-eigenen Fragestellungen und Herausforderungen gespiegelt werden konnte und die Führungskräfte konkrete Maßnahmen und neue Vorgehensweisen für sich ableiten konnten.

Zusammenfassung

- Spezielle Trainingsangebote für alle Führungskräfte auf allen Ebenen mit dem Fokus auf die Rolle der Führungskraft

Unser Führungsverständnis muss sich also ändern – aber wie?

Was jetzt noch fehlte, war ein gemeinsames Verständnis, was denn nun die Anforderungen sind, die an eine Führungskraft im agilen Kontext gestellt wird. Daraufhin wurde die im ersten Teil beschriebene Initiative zur Überarbeitung unserer Führungsprinzipien mit Führungskräften aller Hierarchieebenen, Mitarbeitern und Vertretern des Betriebsrates gestartet (s. Teil 1, Bild 1).

Diese neuen Führungsprinzipien sollten als Rahmen für die Führungsmannschaft dienen, an der sich eigenes Führungsverhalten spiegeln und weiterentwickeln ließ. Hierbei setzten wir beim "Ausrollen" in unserem Bereich neben der "traditionellen" Top-Down-Kommunikation auch auf begleitende Maßnahmen wie Communitys of Practice für Führungskräfte, spezielle Workshop-Angebote zur Vertiefung einzelner Aspekte und die Nutzung einer internen Social-Media-Plattform für den schnellen Austausch.

Mitarbeiterbefragung als Kompass

Außerdem war uns wichtig, dass wir dieses Führungsverständnis nachhaltig in unserer Kultur verankern und dieses auch aktiv leben. Nur wie wissen wir, ob wir auf dem richtigen Weg sind? Wir haben dazu unser Instrument der Mitarbeiterbefragung genutzt, welche zwei Mal im Jahr stattfindet. Dort stellten wir sieben Fragen zum Führungsverständnis, anhand derer die Mitarbeiter ihre Führungskraft bewerten konnten:

1. Meine Führungskraft erkennt meine individuellen Bedürfnisse, Fähigkeiten und Ziele.
2. Meine Führungskraft schenkt Vertrauen und fördert Selbstorganisation.
3. Meine Führungskraft kommuniziert direkt und stellt Transparenz her.
4. Meine Führungskraft stellt sicher, dass jeder im Team seine Rolle und Verantwortung kennt und übernimmt.
5. Meine Führungskraft stellt sicher, dass Arbeitsergebnisse evaluiert werden und daraus gelernt wird.
6. Meine Führungskraft macht deutlich, dass "Kunden begeistern" im Zentrum unserer Arbeit steht.
7. Wenn ich an all die erbrachten Leistungen und Anstrengungen denke, halte ich die erfahrene Anerkennung für angemessen.

Für uns ist dies ein weiteres Feedbackinstrument, das uns zeigt, wo wir noch besser werden müssen. Und zusätzlich haben wir uns quantitative Ziele gesetzt, die wir erreichen wollen – und die auch in die Zielvereinbarung der Führungskräfte einfließen.

Zusammenfassung

- Definition eines gemeinsamen Führungsverständnisses unter hoher Beteiligung der Führungskräfte aller Ebenen und Mitarbeitern.
- Verankerung des Führungsverständnisses in der Mitarbeiterbefragung als wichtiges Messinstrument.

Weitere Maßnahmen auf dem Weg zum "agilen Manager"

Jetzt hatten wir für uns beispielsweise als Führungsprinzip definiert: "Vertrauen schenken". Was heißt das aber für die einzelne Führungskraft, die bisher lieber die Kontrolle behielt und sich dazu Mechanismen aufgebaut hatte? Was kann ich als Führungskraft tun, um "Vertrauen zu schenken"? Hierzu schlossen sich zwei weitere Aktivitäten an.

Individuelle Workshops zur praktischen Umsetzung

Zum einen entwickelten wir ein individuelles Workshop-Angebot für Führungsteams, denen wir neben vertiefender Theorie zu agilen Methoden auch Tools für ihre tägliche Führungsarbeit und zur Umsetzung der Führungsprinzipien angeboten haben. So eignet sich z.B. "Delegation Poker" aus Management 3.0 sehr gut, um über konkrete Fragestellungen zu Aufgaben und Verantwortungen in den Teams ein gemeinsames Verständnis zum Delegationslevel zu erzielen. Also wie viel Vertrauen und Empowerment möchten wir als Führungskräfte unseren Teams konkret geben?

Interessant in den Übungen war, dass die Führungskräfte selbst sehr unterschiedliche Meinungen hatten und über Austausch und Diskussion zu neuen Erkenntnissen kamen, wie sie gemeinsam mit ihren Teams Themen in Zukunft angehen wollen. Hierzu gehörte z.B., dass die Urlaubsplanung künftig am besten direkt im Team entschieden und die Führungskraft nur noch informiert wird.

Sich dessen bewusst zu sein, welche Entscheidungsbefugnis man den Teams geben möchte und dieses transparent zu machen, führt dazu, dass "Vertrauen schenken" auf einmal "anfassbar" wird. Das führte dazu, dass die Führungsteams direkt aus dem Workshop wertvolle Ergebnisse für ihr eigenes "Führungs-Tagesgeschäft" mitnehmen konnten.

Auseinandersetzung mit den Führungsprinzipien

Die zweite Aktivität, die umgesetzt wurde, war ein offenes Trainingsangebot, bei dem sich acht bis zwölf Führungskräfte einen Tag lang mit den Führungsprinzipien beschäftigten, in dem sie Einzelarbeit, Zweier-Arbeiten oder Gruppenaufgaben gemeinsam bearbeiteten. Ziel war es, sich Zeit zu nehmen für die Selbstreflektion, sich mit anderen Führungskräften auszutauschen und am Ende des Tages mit einer individuellen Maßnahme aus dem Workshop rauszugehen, die jeder am nächsten Tag umsetzen und für seine persönliche Weiterentwicklung nutzen konnte.

Der Workshop war deshalb ein Mix aus Input-Teilen und direkt anschließender Reflexion der Führungskräfte. Eine solche individuelle Maßnahme konnte z.B. sein, dass die Führungskraft sich bewusst Zeit nimmt, um selbst einige Stunden in der Woche im Teamraum zu arbeiten und dadurch ansprechbarer zu sein. Oder klassische schriftliche Statusreports ablöst durch direkte Kommunikation und Interaktion mit dem Team, z.B. regelmäßiger Besuch der Daily-Stand-Up-Meetings oder der Review-Meetings.

Ein weiteres Beispiel: Motivierte Mitarbeiter sind Stützpfiler für jedes Unternehmen. Soweit klar. Aber was motiviert Mitarbeiter eigentlich und was kann ich als Führungskraft tun, um sie zu motivieren? Für die Fragestellung zeigten wir ein Video von Dan Pinks Rede zu intrinsischer Motivation, welche auf seinem Buch "The Surprising Truth of What Really Motivates Us" basiert. Er zeigt darin auf, was die Kernelemente der intrinsischen Motivation sind:

1. "Autonomy" – autonomes Arbeiten und die Möglichkeit, selbst zu bestimmen, was ich tue, um meine Ziele zu erreichen.

2. "Mastery" – die Möglichkeit, meine Fähigkeiten auszubauen, dazuzulernen, immer besser zu werden und dadurch mehr Zufriedenheit und Anerkennung bei meiner Arbeit zu bekommen.
3. "Purpose" – der eigentliche Sinn und Zweck meiner Arbeit, z.B. eben nicht nur dafür zu sorgen, dass wir als Unternehmen Geld verdienen, sondern durch innovative, aber einfach zu handhabende Produkte unsere Kunden zu begeistern und dabei auch zu erleben, welchen Beitrag ich dabei geleistet habe.

Wir ließen die Führungskräfte dann reflektieren, ob und wie sie das schon in ihrem eigenen Bereich umsetzen. Danach tauschten sie sich in einer kleineren Gruppe aus und notierten ihre "Aha-Erkenntnisse". Diese wiederum wurden dann im Plenum vorgestellt.

Damit erreichten wir sowohl Selbstreflexion als auch Vernetzungspunkte unter den Teilnehmern. Die Führungskräfte griffen gute Ideen ihrer Kollegen auf, sodass ein aktives "Voneinander-Lernen" initiiert wurde. Am Ende des Workshops war dann jeder Teilnehmer aufgefordert auf Basis der Gespräche und Impulse des Tages sich eine individuelle Aktivität vorzunehmen, und diese mit einem Partner zu besprechen. In vielen Fällen resultierten daraus gegenseitige Absprachen, die nach wie vor helfen, eine nachhaltige Umsetzung des Vorhabens zu erreichen.

Individuelle Beispiele, die aus diesen Workshops entstanden, waren die Abschaffung von wenig effizienten Status-Meetings, die Einführung von kurzen Gesprächsrunden zur schnellen Informationsweitergabe oder kollegiales Coaching zur Besprechung schwieriger Führungssituationen.

Zusammenfassung

- Trainings- und Workshop-Angebote bereitstellen für die Selbstreflexion und zur Erarbeitung konkreter Maßnahmen mit Bezug auf die individuellen Bedarfe.

Was waren und sind die Herausforderungen?

Führungsarbeit im agilen Kontext bedeutet zunächst einmal, dass dies einen Prozess darstellt, der "kein Ende" hat. Man muss sich mit der Führungsarbeit, mit seiner Rolle und der persönlichen Weiterentwicklung permanent beschäftigen. Wir hörten zwar oft, "das mach ich doch schon". Wenn man aber genauer hinsah, geschah es bei einigen dann eben doch nicht.

Feedback annehmen

Im Grunde sind die erarbeiteten neuen Führungsprinzipien einfach; aber die Umsetzung ist in der Praxis nicht trivial und erfordert mitunter auch die Veränderung des eigenen Verhaltens. Das funktioniert allerdings nur, wenn ich ehrlich zu mir bin, wenn ich Feedback meiner Mitarbeiter oder Peers annehme und dann auch damit etwas mache.

Es gibt Menschen, die diese Arbeits- und Verhaltensweise begrüßen. Andere fühlen sich unwohl und schaffen die Veränderung nicht. Auch hier muss man offen damit umgehen und ggf. andere Arbeitsfelder suchen – anstatt jemand in ein neues "Schema" pressen zu wollen.

"Dafür hab ich keine Zeit"

Eine weitere große Herausforderung war: "Ich würde ja gerne, aber ich habe dafür keine Zeit!" (z.B. für die Trainings oder die tägliche Führungsarbeit). Es ist eine Frage der Priorisierung und auch der Steuerung durch den eigenen Vorgesetzten, der durch eigenes Verhalten zeigen sollte, wie wichtig ihm Führungsarbeit ist. Nimmt mein Vorgesetzter sich Zeit für den Austausch und die Reflektion zu Führungsthemen mit mir, und macht er deutlich, dass er mich dabei unterstützt, damit auch ich mir genug Zeit für Führungsarbeit nehmen kann, erhält dieses Thema eine ganz andere Priorität. Ohne dass ich mir Zeit fürs Führen nehme, kann ich nicht effizient führen.

Das gleiche gilt auch für das Thema Kultur. Viele Führungskräfte sprechen gerne über Kultur, und wie wichtig diese für den Erfolg ist. Aber Kultur wird häufig "mystifiziert" als etwas, das nicht anfassbar, sondern einfach da ist. Dabei kann ich als Führungskraft direkt Einfluss nehmen auf die Kultur, wenn ich mir auch die Zeit dafür nehme. Statt mich mit Zahlen und Projektreportings zu beschäftigen, kann ich auch zu den Menschen gehen, die konkret an den Themen und Produkten arbeiten.

Die Kultur beeinflussen

Ist die Kultur von Kontrolle und Mikromanagement geprägt, kommt dies bei den Mitarbeitern natürlich auch schnell genauso an. Lege ich in meiner Führungsarbeit aber Wert darauf, zu zeigen, dass ich Vertrauen gebe, Probleme verstehen will, um zu helfen und meinen Mitarbeitern direkte Wertschätzung für ihre Arbeit gebe, dann schaffe ich damit eine Kultur, die durch Offenheit, Interesse an Transparenz und einem Dialog auf Augenhöhe geprägt ist.

Das geht nicht über Nacht und erfordert Zeit und beiderseitiges Vertrauen. Aber wenn ich es schaffe, durch meine tägliche Arbeit als Führungskraft (und nicht als Zahlenmanager) diese Kultur herzustellen, gewinne ich wertvolle Handlungsmechanismen, um gemeinsam mit meinen Mitarbeitern daran zu arbeiten, Kunden zu begeistern und unternehmerischen Wert zu schaffen. Dieses Vorgehen ist auch unter dem Begriff "Gemba-Walk" bekannt, eine Methode aus dem Lean Manufacturing mit der Idee, dass alle Probleme letzten Endes direkt beobachtbar sind, und die besten Optimierungsansätze somit direkt vor Ort – der Gemba – entwickelt werden können.

Herausforderung Fluktuation

Um noch einen letzten, vierten Punkt zu nennen: Gerade in der Einführungsphase können Fluktuation und personelle Veränderung auf Führungsebene problematisch sein. Wenn es im agilen Transitionsprozess gelungen ist, ein erstes gemeinsames Verständnis zwischen den Beteiligten aufzubauen, und genau jetzt kommt eine neue Führungskraft hinzu, die das Ganze noch nicht kennt oder dem skeptisch gegenüber steht, ist wiederum von neuem viel Arbeit und Zeit notwendig, um das Zusammenspiel zu finden. Diesen Punkt sollte man nicht unterschätzen.

Führungskräfte in der Agilen Transition – Zusammenfassung der wesentlichen Schritte

1. Ein Commitment von "oben" ist wichtig – bottom up ist ein guter Anfang, reicht aber nicht aus für eine nachhaltige Umsetzung.
2. Begleitung durch ein "Agile Transition Team" ist essentiell – es braucht Zeit für die Veränderung und die Teams können dies nicht alleine schaffen.
3. Nicht lange planen, einfach anfangen – agil löst nicht die Probleme in unserem Unternehmen, aber agiles Arbeiten macht sie transparent und man weiß somit ganz genau, woran man am besten arbeiten sollte.
4. Spezielle Trainingsangebote für alle Führungskräfte auf allen Ebenen mit dem Fokus auf die Rolle der Führungskraft.
5. Definition eines gemeinsamen Führungsverständnisses unter hoher Beteiligung der Führungskräfte aller Ebenen und Mitarbeitern.
6. Verankerung des Führungsverständnisses in der Mitarbeiterbefragung als wichtiges Messinstrument.
7. Trainings- und Workshop-Angebote bereitstellen für die Selbstreflektion und zur Erarbeitung konkreter Maßnahmen mit Bezug auf die individuellen Bedarfe.

Fazit

Es lohnt sich, beziehungsweise ist zwingend notwendig, dem Thema "Führung" bei der Einführung von agilen Methoden von Beginn an Raum zu geben und sich bewusst zu machen, dass nur mit einem passenden Führungsverständnis und der Bereitschaft für Veränderung eine erfolgreiche Nutzung der agilen Arbeitsweise möglich ist.

Insgesamt setzen agiles Arbeiten und agile Werte auf ein "positives" Menschenbild, das davon ausgeht, dass Menschen Know-how haben, Leistung bringen wollen und sich einbringen. In solch einem Raum können die Mitarbeiter kreativ sein und sich engagieren.

Das fällt aber nicht vom Himmel, sondern ist ein Lern- und Veränderungsprozess, dem sich insbesondere die Führungskräfte stellen müssen. Denn dann versprechen agile Arbeitsweisen erfolgreiche Ergebnisse und auch Spaß im täglichen miteinander.