

Fachbeitrag

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.

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.

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

ähnliche Artikel

[Teamzuständigkeiten in Scrum – von Komponententeams zu Feature Teams](#)

sowie in den Rubriken:

[Agiles Projektmanagement](#)

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://stefanrook.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

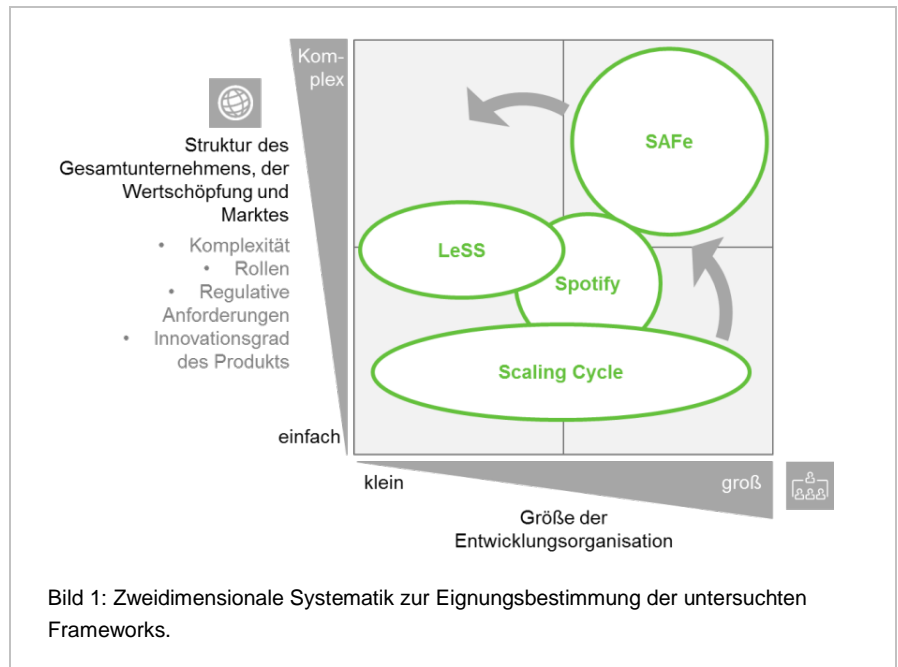
Entwicklung erfolgreich, und es fehlt nur noch an den Schnittstellen? Eine angrenzende Fragestellung, die sich aus meiner Erfahrung als nützlich erwiesen hat: Fragen Sie sich, wie aufwendig es wäre und wie viel Überzeugungsarbeit nötig ist, um in Ihrem Umfeld interdisziplinäre Teams zu bilden, physisch zusammensetzen 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.

Hat Ihnen dieser Artikel gefallen?

Bewerten Sie ihn im Projekt Magazin online und teilen Sie so Ihre Meinung anderen Lesern mit. Wählen Sie dazu den Artikel im Internet unter <http://www.projektmagazin.de/ausgaben/2016> oder klicken Sie [hier](#), um direkt zum Artikel zu gelangen.