

# SAFe - Das Scaled Agile Framework

Lean und Agil in großen Unternehmen skalieren

Bearbeitet von  
Christoph Mathis

1. Auflage 2016. Buch. 220 S. Softcover  
ISBN 978 3 86490 228 4  
Format (B x L): 16,7 x 24,1 cm  
Gewicht: 436 g

[Weitere Fachgebiete > EDV, Informatik > Software Engineering > Agile Softwareentwicklung](#)

Zu [Inhaltsverzeichnis](#)

schnell und portofrei erhältlich bei

  
DIE FACHBUCHHANDLUNG

Die Online-Fachbuchhandlung [beck-shop.de](http://beck-shop.de) ist spezialisiert auf Fachbücher, insbesondere Recht, Steuern und Wirtschaft. Im Sortiment finden Sie alle Medien (Bücher, Zeitschriften, CDs, eBooks, etc.) aller Verlage. Ergänzt wird das Programm durch Services wie Neuerscheinungsdienst oder Zusammenstellungen von Büchern zu Sonderpreisen. Der Shop führt mehr als 8 Millionen Produkte.

---

# 1 Einleitung

Viele Darstellungen von Agile wecken die Assoziation an kleine IT-Entwicklungsteams, wenig Disziplin und das Ausrollen einer einfachen Methode.

Stabile selbstorganisierte kleine Teams sind tatsächlich die Grundlage für die Erfolgsgeschichte der Agilität, und in den ersten Jahren haben sich die Agilisten auch auf überschaubare Entwicklungen beschränkt. Mit der Nutzung in großen Organisationen entsteht die Notwendigkeit, das Selbstbild zu erweitern, und es wächst die Dringlichkeit, Missverständnisse auszuräumen.

Agile Entwicklung im Großen braucht eine solide Implementierung der einzelnen Teams, und alle relevanten Ansätze basieren auf einer Skalierung über Strukturen auf Basis von »Teams von Teams«. Sie benötigt auch eine genauere Betrachtung der übergreifenden Prozesse. Daraus ergibt sich die Integration von Lean-Prozessen und Lean Thinking, der wichtigsten geistigen Grundlage der Agilität.

Eine erfolgreiche agile und Lean-Transformation muss auf dem Verständnis gründen, dass Agilität eine geistige Haltung meint und einen Kulturwandel für die ganze Firma darstellt. Sie beeinflusst die Führung und die Entscheidungsbeurteilungen ebenso wie die Gestaltung der Wertströme.

Sie ist ein Startpunkt für einen kontinuierlichen Change-Prozess und bedeutet damit einen tiefen Einschnitt in das Selbstverständnis eines Unternehmens.

## 1.1 Was macht Agilität erfolgreich

Das formulierte Ziel agiler Entwicklung ist die zuverlässige und effektive Lieferung in kurzen Zyklen. Das bedeutet, dass Anforderungen in Features oder »vertikalen Inkrementen« bearbeitet und fertiggestellt werden. Im Fall von Software heißt fertiggestellt, dass die Software durch automatisierte, wiederholbare Tests überprüft ist.

Damit erreicht man eine im Vergleich zu herkömmlicher Softwareentwicklung unvergleichlich bessere Möglichkeit zu priorisieren, d.h. die wichtigsten Features zuerst auszuliefern.

Dieser schnelle Zyklus erfordert eine Autonomie des Bearbeitungsteams über Detailentscheidungen, Arbeitseinteilung im Team und Ähnliches, um die nötige Geschwindigkeit der Kommunikation und Entscheidungsfindung zu erreichen. Dies wiederum erfordert ein stabiles, kleines, engagiertes Team, das in einer transparenten und vertrauensvollen Umgebung arbeitet.

Zentrale Elemente der Effektivität von Agilität hängen also an diesen überschaubaren Teams. Auf der anderen Seite sind manche Vorhaben zu umfangreich oder komplex für ein einzelnes Team. Daher stellt sich die Frage, wie man diese Effektivität in einer Entwicklung in größerem Maßstab erhalten kann.

## 1.2 Warum skalieren

Komplexe Vorhaben brauchen viele Mitwirkende und erfordern eine verstärkte Arbeitsteilung. Das scheint zunächst ein Widerspruch zu den beschriebenen Erfolgsbedingungen von Agilität zu sein.

Wenn man sich diesen Widerspruch genauer ansieht, bemerkt man, dass vor 15 Jahren vor allem kleinere Projekte bevorzugt mit agilen Methoden angegangen wurden. Langsam hat sich mehr Wissen darüber angesammelt, wie Agilität funktioniert, und die Vorteile haben sich auch außerhalb dieses Bereichs herumgesprochen.

Agile Methoden haben im Verlauf dieser Zeit viel im Kern der Softwareentwicklung geändert und werden jetzt an umfassenderen Herausforderungen gemessen:

- Große, komplexe Entwicklungsvorhaben realisieren
- Den gesamten Entwicklungszyklus betrachten
- Die Organisation als Ganzes einbeziehen
- Große und etablierte Organisationen unterstützen
- Den Produktentwicklungszyklus insgesamt zu agilisieren und die schnelle Anpassung von Produkten in Interaktionen mit dem Markt einbeziehen

### Muss man wirklich skalieren?

Komplexität ist die Ursache für eine skalierte Umsetzung und die Beteiligung vieler Personen an einer Lösung.

Komplexität hat mehrere Aspekte, die gleichzeitig vorhanden und unterschiedlich ausgeprägt sein können:

- Essenzielle Komplexität:  
Das zu lösende Problem ist tatsächlich so umfangreich und benötigt die Betrachtung vieler Einzelaspekte.

- Zufällige Komplexität (accidental complexity):  
Die Komplexität entsteht durch eine suboptimale Herangehensweise und ein unzureichendes Verständnis der Problemstellung.

Vor einer Skalierung der Entwicklungsmannschaft steht daher die Frage, welcher Anteil der Komplexität wirklich essenziell ist. Wenn der Großteil der Komplexität zufällig ist, ist unter Umständen eine Skalierung vermeidbar.

### Teamgrößen

Teams funktionieren am besten in einer Größe von vier bis zu ca. zehn Mitgliedern.<sup>1</sup> Bei dieser Größe hat jedes Mitglied die »mentale Bandbreite«, um mit jedem anderen Teammitglied zu kommunizieren und ein Gesamtbild zu überblicken. Noch kleinere Gruppen tendieren dazu, nicht die nötigen Rollen in der Interaktion abzudecken und weniger kreativ zu sein.

Wenn das Team weiter wächst, wird die gleichzeitige Kommunikation zunehmend schwierig und das Team strukturiert sich in Untergruppen oder tendiert dazu, Konflikte zu entwickeln.

Das führt bei größeren Vorhaben allerdings zu neuen Anforderungen und Fragen:

- Nach welchen Kriterien teilt man die Teams ein (Komponententeams versus Feature-Teams)?
- Wie kommunizieren die Teams untereinander?
- Wie behandelt man Abhängigkeiten?
- Wie kooperieren die Teams bei gemeinsamen Lieferungen?

### Die Dunbar-Zahl

Eine zweite Grenze definiert der Kognitionswissenschaftler Robin Dunbar mit der »Dunbar-Zahl«.<sup>2</sup> Er nennt 150 als die Zahl der Menschen, mit denen ein Einzelner soziale Beziehungen unterhalten kann. Das wird als sinnvolles Limit für größere Arbeitsgruppen gesehen. Bis zu dieser Grenze lassen sich agile Praktiken wie Face-to-Face-Kommunikation übertragen bzw. anpassen.

Bei noch wesentlich größeren Mannschaften versagen auch diese Methoden.

Im Scaled Agile Framework (SAFe) werden spezifische Organisationsformen, Planungsverfahren und Artefakte entwickelt, die diesen Erkenntnissen Rechnung tragen:

---

1. Das ist bewährte Praxis in Scrum-Projekten und so in der Scrum-Referenzliteratur beschrieben.  
2. Siehe [http://en.wikipedia.org/wiki/Dunbar's\\_number](http://en.wikipedia.org/wiki/Dunbar's_number).

- Die **Teamebene** sind (möglichst) Feature-Teams, die mit Scrum und XP als zugrunde liegende Methoden arbeiten. Alternativ zu Scrum kann auch Kanban verwendet werden, dann nehmen die Teams aber ebenso am Planungstakt der Programmebene teil und haben ebenso wie Scrum-Teams einen Product Owner und einen Scrum Master.
- Die **Programmebene** ist eine Gruppe solcher Teams, die zusammen 50–125 Personen umfassen. Dazu kommen einige zusätzliche Rollen bzw. Funktionen zur Unterstützung der größeren Struktur.

## 1.3 Die neuen Herausforderungen an Agilität

### Große Vorhaben unterstützen

*»Scaling Agile methods is the last thing you should do.«*

*Martin Fowler, 2003*

In der Anfangszeit der agilen Softwareentwicklung haben sich die Pioniere vornehmlich kleinere Entwicklungen vorgenommen. Es gab wenig Erfahrung mit Agilität insgesamt und kaum Expertise in komplexen (und teuren) Projekten. Erst allmählich und mit mehr Erfahrung versuchte man sich an größeren Projekten und traf damit auch auf deren spezifische Probleme.

In letzter Zeit ist das Interesse an großen Entwicklungen erheblich gestiegen und damit werden auch die neuen Herausforderungen deutlich.

### Der ganze Wertstrom

Wenn man die gesamte Lieferkette betrachtet, wirkt sich auf die Zykluszeit von Produkten – und damit auf die Reaktionsmöglichkeit der Organisation – nicht nur die reine Softwareentwicklung aus. Auch die vorgelagerten Bereiche wie Produktentwicklung, Spezifikation und die nachgelagerten Bereiche wie Abnahme, Systemtest und Auslieferung, also die Agilität einer Organisation, wird durch den gesamten Wertstrom bestimmt.

Nachdem also der Kernbereich der Softwareentwicklung »agilisiert« worden ist, rücken diese vor- und nachgelagerten Bereiche als Flaschenhälse in den Fokus.

Zur Integration von Betrieb und der Auslieferung mit der Entwicklung gibt es eine lebhafte Debatte und auch schon viele Ergebnisse unter dem Stichwort »DevOps«.

### **Die ganze Organisation und Hierarchie**

Die Bereiche Produktentwicklung, Marketing und Vertrieb haben eine solche Veränderung oft noch gar nicht auf dem Radar, obwohl ein agiles Umfeld tiefgreifende Veränderungen für sie bedeutet.

Das Gleiche gilt für das Management. Die Produktteams und Produktentwicklungsmannschaften haben eine neue Perspektive und neue Erwartungen an ihr Management. Sie erwarten andere Führungsprinzipien und benötigen andere Unterstützung.

### **Große und etablierte Entwicklungen**

Doch nicht nur die Unternehmen müssen ihre Strukturen anpassen, auch bei den Methoden der Agilisten sind Anpassungen notwendig. In der agilen Welt hat sich das Prinzip bewährt, im Kleinen anzufangen, Wissen aufzubauen, mit einem Team an die Arbeit zu gehen und erst dann organisch zu wachsen. Das hat den Vorteil, dass sich eine agile Kultur sehr wirkungsvoll verankern kann und sich das Alignment quasi im Selbstlauf ergibt.

Leider steht dieser Weg einer großen bestehenden Organisation, die agil werden will, in der Regel nicht offen. Sie kann nicht einfach den Betrieb einstellen und auf der grünen Wiese neu starten.

Daraus ergibt sich die Frage, wie agile Prinzipien so aufbereitet werden können, dass sie für die Transformation einer solchen reifen, gewachsenen Organisation nutzbar gemacht werden können.

Wenn man allgemeine Erfahrungen aus Change-Prozessen auswertet, ergibt sich der starke Hinweis, dass Organisationen der »early majority« ein höheres Sicherheitsbedürfnis haben und mehr Führung und detailliertere Startvorgaben wollen oder brauchen als die Innovatoren, die zu Agilisten der ersten Zeit wurden.

Die Herausforderung besteht darin, die damit verbundenen Fallen zu vermeiden. Außerdem gilt es zu verhindern, dass Agile und Lean nur als marginal neue Vorgehensweisen wahrgenommen und ohne wirklich substantielle Veränderungen aufgesogen werden.

Agile und Lean sind fundamental anders als die vorherrschende tayloristische Organisation, die Einführung von agilen Methoden ist daher ein disruptives Element, das die Firma und ihre Kultur tiefgreifend verändert.

### **Der Zyklus von Produkt und Markt**

Nach der Veränderung der internen Struktur und Kultur einer agil arbeitenden Firma wandelt sich die Herangehensweise an Veränderungen stark: Diese werden als Möglichkeiten und Chancen wahrgenommen statt als zu bewältigende Probleme.

Das hat auch Folgen für die Leichtigkeit, mit der Veränderung integriert wird: Änderungen der gesamten Wertschöpfungskette werden einfacher, billiger, schneller.

Damit ist der Weg offen für den nächsten Schritt: Kunden extrem frühzeitig einzubeziehen und ihr Feedback zum integrierten Teil der Produkt- und Strategieentwicklung zu machen. Eric Ries hat in seinem Buch »Lean Startup« [Ries 2011] den Code dafür geknackt, wie dies effektiv umgesetzt wird. Mittlere und große Firmen beginnen, dies zu erkennen und verwenden zunehmend diese Methoden, um ihre Geschäftsprozesse weiterzuentwickeln.

### **Kultur oder wie Agile aus dem Radarschatten kam**

*»The only thing that scales is values.«*

*Ray Arrell, Intel*

Die »Agilisierung« einzelner Teams kann mancherorts noch unbemerkt unter dem Radarschatten einer Organisation vorgehen: Ein agiler Coach schult die Entwickler und hilft dem Team bei den Anfangsproblemen mit der Vorgehensweise. Das Team wird produktiver, und der Vorgesetzte lässt es gewähren.

Diese Einführung über einen Regelverstoß kann ein erster Schritt sein – allerdings ist er immer gefährdet und er skaliert nicht auf größere Gruppen oder gar die Gesamtorganisation.

In den Beziehungen eines Teams mit seiner Umgebung müssen irgendwann die Erwartungen abgestimmt werden, etwa die Rollenbilder und Karrierewege der Entwickler, Scrum Master und Product Owner.

Es ist ebenfalls nicht realistisch, dass Vorgesetzte eine größere Gruppe ihrer Mitarbeiter autonom agil arbeiten lassen – sie würden ihre Führungsverantwortung vernachlässigen. Im Gegenteil: Vorgesetzte müssen einen wesentlichen Beitrag zur Entwicklung der Mitarbeiter leisten und aktiv die Grenzen der Autonomie und Initiative verteidigen helfen. Die Erwartungen der Organisation und ihre Kultur bilden eine der wesentlichen Rahmenbedingungen für erfolgreiche agile Teams. Mit einem Wort: Eine skalierte agile Einführung als »Untergrundprojekt«, d.h. ohne Beteiligung des Managements, funktioniert nicht. Den Stand dieser sich entwickelnden Diskussion zu Führungsrollen und Führungsstilen beschreibt Kapitel 9.