

# Vier Reifestufen einer resilient-agilen Organisation

Quandte / Mnich / Nowotny

2023

ISBN 978-3-8006-7092-5

Vahlen

schnell und portofrei erhältlich bei  
[beck-shop.de](https://beck-shop.de)

Die Online-Fachbuchhandlung [beck-shop.de](https://beck-shop.de) steht für Kompetenz aus Tradition. Sie gründet auf über 250 Jahre juristische Fachbuch-Erfahrung durch die Verlage C.H.BECK und Franz Vahlen.

[beck-shop.de](https://beck-shop.de) hält Fachinformationen in allen gängigen Medienformaten bereit: über 12 Millionen Bücher, eBooks, Loseblattwerke, Zeitschriften, DVDs, Online-Datenbanken und Seminare. Besonders geschätzt wird [beck-shop.de](https://beck-shop.de) für sein

umfassendes Spezialsortiment im Bereich Recht, Steuern und Wirtschaft mit rund 700.000 lieferbaren Fachbuchtiteln.

### Die zeitlichen Dimensionen der vier Reifestufen zur resilient-agilen Organisation

- **Gemeinsames Mindset** – Alles beginnt im Kopf: Sekunden und Minuten entscheiden über das Verständnis zur Organisationalen Resilienz und Agilität! Aber in Summe kann eine Organisation sehr lange auf dieser Stufe verharren, da jeder und jede ja auch eine andere Geschwindigkeit hat und von außen aufoktroierte Haltungsänderungen nicht funktionieren.
- **Handlungen und Praktiken** – Hier sind eher Stunden und Tage erforderlich, um die handwerkliche Seite passender Praktiken nachzuvollziehen! Auch für die gemeinsame Auswertung und Verbesserung braucht man Zeit.
- **Frameworks** – Aufbau resilient-agiler Strukturen, Strategien und Prozesse: Wochen und Monate gehen ins Land, bis die Frameworks mit Leben gefüllt und bis die Rollen definiert und besetzt sind, bis die Möglichkeiten der Frameworks erkannt und gemeinsam ausgeforscht wurden! Bei Organisationen, die den berüchtigten „Zombi-Scrum“ praktizieren, wird der Zustand einer befriedigenden Einbettung z.B. von agilen Frameworks in die tatsächliche Rollen- und Prozesslandschaft eines Unternehmens nie wirklich abgeschlossen.
- **Resilienz und Agilität auf Enterprise-Niveau** – Jahre können ins Land gehen, bis die ganze Organisation eine für sich passende Verankerung und umfassende Integration umgesetzt hat.

beck-shop.de  
DIE FACHBUCHHANDLUNG  
Welcher Rahmen soll es sein?

Wenn agile Transformationen scheitern, kann es daran liegen, dass die strukturellen Herausforderungen nicht angegangen wurden. Das ist jedoch zwingend notwendig, um die Triebwerke für diese Stufe in voller Leistungsstärke zu nutzen. In der Raketentechnologie bei Space X ist hierfür der „Vakuum-optimierte RVac mit einer größeren Düselglocke“<sup>53</sup> zuständig, da außerhalb der Atmosphäre – wie auch im Weltall – kein Sauerstoff mehr für die Verbrennung zur Verfügung steht. RVac steht dabei für „Raptor Vacuum Engine“, der es erlaubt, frei im Raum zu navigieren.<sup>54</sup> Es geht uns bei dieser Stufe sehr viel mehr um eine intelligente Aussteuerung, einen vereinbarten Prozess bei der Einführung und das Austesten resilient-agiler Methoden und Strukturen, und weniger um die Bündelung verfügbarer Kräfte. Die Gravitation der Erde wird uns dennoch beschäftigen, auch wenn nun bereits einige kritische Phasen überstanden sind. Hierzu kann ein umfassendes „Meta-Framework“ dienen.

<sup>53</sup> [https://de.wikipedia.org/wiki/Raptor\\_\(Raketentriebwerk\)](https://de.wikipedia.org/wiki/Raptor_(Raketentriebwerk))

<sup>54</sup> <https://interestingengineering.com/spacex-fired-its-starship-raptor-vacuum-engine-twice-in-a-single-hour>

## Ein Plädoyer für ein Meta-Framework

Es gibt viele gute Metaphern und Bilder, die Agilität auf unterschiedlichen Ebenen beschreiben. So kann man sich „das agile Universum“ (Berg 2018, S. 121) wie ein metaphorisches Schalen- oder Zwiebelmodell der Erde vorstellen. Das agile Mindset befindet sich nach Bergs Vorstellung ganz in der Mitte, also genau dort, wo die Erde ihren heißen flüssigen Kern hat. Dicht an diesem Kern angesiedelt sind agile Axiome, agile Werte und die agilen Prinzipien. In diesem Buch ist dies in Summe alles Teil unserer ersten Stufe. Näher an der Erdkruste befinden sich nach Berg (2018) die agilen Praktiken, die sich in unserem Bild in Stufe 2 befinden. Die sichtbaren agilen Modelle bzw. Rahmenwerke wie Scrum, Kanban oder Design Thinking befinden sich bei Berg direkt auf der Erdoberfläche und sind damit der Betrachtung leichter zugänglich. In unserem Modell bilden diese die Stufe 3.

Eine weitergehende Skalierung auf gesamter Organisationaler Ebene ist im Berg'schen Modell nicht vorgesehen. In welchem methodischen Rahmen sich dies abspielt, ist hingegen auch nicht entscheidend, denn sowohl Frameworks wie beispielsweise Kanban oder Scrum lassen sich sehr gut einzeln oder auch als integrierte Elemente eines Systems skalieren. Und auch andere wichtige Frameworks wie Design Thinking, Lean Startup und Vorgehensweisen wie der Einsatz von Objectives and Key Results (OKR) lassen sich in die skalierten resilient-agilen Organisationswelten sinnvoll integrieren. Gleichermaßen finden dort oftmals „resilient-agile Werte und Prinzipien“ (Stufe 1) sowie „wirksame Praktiken und Tools“ (Stufe 2) ihren Platz.

Nun wenden wir uns verschiedenen, bekannten Frameworks agilen Arbeitens zu, explorieren die Besonderheiten und Herausforderungen und werden hoffentlich auch in Deinem Sinne das weiter oben beschriebene Cherry-Picking betreiben. Unser Ziel ist es, daraus ein Vorgehen zur Entwicklung Deines eigenen Meta-Frameworks zu skizzieren.

Dazu ist es zunächst einmal wichtig zu verstehen, welche Fragestellungen durch die verfügbaren Skalierungs-Frameworks eigentlich konkret „geregelt“ werden. Es sind aus unserer Sicht im Wesentlichen die folgenden zehn Fragen bzw. Empfehlungsfelder:<sup>55</sup>

1. Wie funktionieren Planung und Strategiefindung?
2. Wie arbeiten viele verschiedene agile Teams zusammen?
3. Wie wird die Abstimmung zwischen den Teams organisiert?
4. Was sind Aufgabenzuschnitt und Selbstverständnis eines Product-Managers/Product-Owners?
5. Was sind Aufgabenzuschnitt und Selbstverständnis eines Scrum Masters/Agile Coachs?
6. Was sind Aufgabenzuschnitt und Selbstverständnis eines Release Engineers/Group Managers?

<sup>55</sup> <https://echometerapp.com/en/agile-at-scale> bzw. <https://www.atlassian.com/agile/agile-at-scale>

7. Welche Häufigkeiten und Zusammensetzungen von Teams werden für agile Praktiken empfohlen?
8. Welcher Ablauf und welche Teamzusammensetzungen werden für Demos/ Reviews empfohlen?
9. In welcher Häufigkeit und Teamzusammensetzung sollten die Retros erfolgen?
10. Gibt es Empfehlungen zur Kundenorientierung, Wertefokus, Management von Abhängigkeiten, Strategietransparenz, Portfolio Management, Releases on Demand, Risk Management sowie DevOps?

Die Bezeichnungen für den dritten Punkt („Abstimmung zwischen den Teams“) sind in aller Regel unterschiedlich:

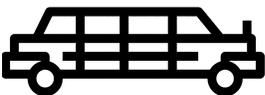
Bezeichnungen für Abstimmung zwischen den Teams
<ul style="list-style-type: none"> <li>■ SAFe → Agile Release Train (ART)</li> <li>■ LeSS → Area</li> <li>■ DA → Team of Teams</li> <li>■ Spotify → Tribes</li> <li>■ Scrum@Scale → Scrum of Scrums</li> <li>■ SPS → Integrationsteam</li> </ul>

Neben den sechs klassischen Skalierungsframeworks SAFe, LeSS, DA, Spotify, Scrum@Scale und SPS wollen wir noch drei weitere wertvolle Frameworks einordnen:

- Design Thinking → Rahmen für einen nutzerzentrierten Innovationsprozess
- Kanban Flight Levels → Skaliertes Arbeiten mit Kanban auf drei Levels
- UnFIX Modell → Vielfältige Anpassungen des Organisationsdesigns

Diese *neun* Frameworks werden wir nun kurz beschreiben und ihren Nutzen für ein CRAF (Corporate Resilient Agile Framework) im Sinne eines Meta-Frameworks herausarbeiten. Starten wollen wir mit einer kurzen Beschreibung der sechs bekanntesten und nah an Scrum orientierten Skalierungsmodelle:

## SAFe (Scaled Agile Framework) – Die komfortable Limousine



SAFe ist die „komfortable Limousine“ unter den agilen Skalierungs-Frameworks und nicht nur bei der Deutschen Bahn sehr populär. Dieses Framework ist hochwertig, wunderbar ausgearbeitet und bietet einen umfassenden Leistungskatalog. Es bietet in vielerlei Hinsicht „Komfort“, beispielsweise, indem viele Antworten geliefert werden, die sonst zeitaufwändig selbst durchdacht und erarbeitet werden müssten. Ein Umstand, der aus Sicht mancher für agile Transformationen auch erforderlich wäre, um tatsächlich die Selbstorganisation zu stärken.

## Grundgedanken

Die genaue Bezeichnung lautet: „Scaled Agile Framework“ (kurz „SAFe“). Die theoretischen Grundlagen stammen aus einer Publikation von Dean Leffingwell aus dem Jahre 2011. Die heutige Verbreitung liegt laut dem 12th Annual State of Agile Report vom 9. April 2018 bei rund 29% unter den rund 1.500 befragten Organisationen, welche zum einen agile Frameworks nutzen und sich andererseits an der Studie beteiligt haben (vgl. Collabonet, 2018).

Das Modell bildet dabei im Kern drei Ebenen ab: Teamebene, Programmebene sowie Portfolioebene. Auf der Teamebene scheint SAFe sehr an Scrum angelehnt zu sein. Allerdings gibt es Einschränkungen: Zum Beispiel ist es nicht so, dass in jedem Sprint ein potenziell auslieferbares Produkt erstellt wird. Das würde bei SAFe typischerweise nur in Zusammenhang mit einem sogenannten „hardening sprint“ passieren. Der „hardening sprint“ ist aus Sicht der Scrum-Erfinder Jeff Sutherland und Ken Schwaber übrigens ein Unwort, da es aus deren Sicht nichts Lehrreicherer gibt, als tatsächlich regelmäßig Inkremente zu liefern (was bei den Sutherland und Schwaber zuordenbaren Skalierungs-Frameworks Scrum@Scale und Nexus auch geschieht). Bei SAFe ist das anders: Hier wird immer dann geliefert, wenn der „Agile Release Train“ kommt und dieser bestückt wird, also frühestens alle drei Monate. Eine Zeitspanne, die in etwa zwischen Scrum-Sprints (1 bis 4 Wochen) und dem Planungshorizont eines klassischen Großprojekts liegt. Somit ist SAFe ein Kompromiss von klassischer und agiler Vorgehensweise, die allein schon deswegen für Großkonzerne attraktiv und für agile Fundamentalisten des Teufels ist.

## Besonderheiten

Alles beginnt mit dem „PI-Planning“, ein Face-to-Face-Großgruppenformat, bei dem sich alles um die Planung des „product increments“ dreht. Für dieses Großgruppenformat gibt es in SAFe eine „time box“ von ein bis zwei Tagen. Auch ohne SAFe kann diese Vorgehensweise im Sinne eines „big room planning“ bei Großprojekten eingesetzt werden. In dieser „time box“ treffen sich alle Teams und planen die nächsten acht bis zehn Wochen gemeinsam.

Bei der Planung des „program increments“ geht es vor allem darum, welches Team was umsetzt und wie die Abhängigkeiten zwischen den Teams sind. Das Ergebnis wird am „program board“ visualisiert. Die zu leistende Arbeit wird dabei in „value streams“ aufgeteilt. Damit entsteht ein Zielbild für die nächsten drei Monate.

Die Sprint Backlog Items werden dann in drei Kategorien eingeteilt: (1) Fehlerbehebung (wird vor allem anderen priorisiert), (2) Items vom „big room planning“, (3) Themen, die ungeplant zusätzlich umzusetzen sind. Den Erkenntnissen aus Netzwerktheorie und Zeitmanagement folgend, werden bei SAFe die Sprints nur zu 30 bis 70% befüllt, um Raum für Fehlerbehebung und Ungeplantes zu lassen.

SAFe arbeitet auf drei unterschiedlichen Ebenen: „Portfolio Level“ (Rollen: Portfolio Manager, Enterprise Architect, Epic Owner), darunter das „Program Level“ mit den „program increments“ (Rollen: System Architect, Product Manager, Release Train

Engineer) und schließlich das „Team Level“ mit den einzelnen Scrum-Teams (Product Owner, Scrum Master, Entwickler).

Der „Release Train Engineer“, der das „program increment“ unterstützt, agiert als agiler Coach. In der abgespeckten Version „Essential SAFe“ konzentriert sich alles auf einen einzelnen „Agile Release Train“ (ART). Im Zentrum soll so agiles Produzieren stehen. Die Erweiterungen folgen dann später, etwa wenn die Organisation wächst bzw. weitere Teile zum agilen Arbeiten übergehen.

Über sogenannte Business Epics (kundenorientiert) und Enabler Epics (technische Lösungen) werden bei SAFe die Zielvorgaben konkretisiert, denen ein übergeordnetes Thema, also ein Investment über eine Zeit von sechs bis zwölf Monaten zugrunde liegt. Die einzelnen Epics werden dann in die Value-Stream-Ebene heruntergereicht und dort in Capabilities und anschließend in Features aufgeteilt. So lassen sich fast klassisch Arbeitspakete schnüren, die entwickelt und dann später über den „Agile Release Train“ ausgeliefert werden können. Aus klassischer Management-Sicht können auf diese Weise große Arbeitspakete und entsprechend die Budgets gut portioniert und geplant werden.

Über SAFe ist alles verfügbar, was für eine luxuriöse Reise benötigt wird, inklusive einer in vielen Punkten bereits festgelegten Reiseroute. Das Ziel lässt sich so recht sicher ansteuern und das plastische Bild des „Agile Release Trains“ ist vermutlich ein Grund, warum sich SAFe bei Konzernen wie der Deutschen Bahn einer gewissen Popularität erfreut.

### Kritische Würdigung

Die langen Sprints und Zeitspannen können zu „politischem Handeln“ führen, wie man das aus dem klassischen Projektmanagement kennt. Die Teams schaffen sich ihre Sicherheiten und bauen taktische Puffer ein, denn erst nach drei Monaten muss ja geliefert werden. Damit greift eine allgemeine Projektmanagement-Regel: „Puffer, die da sind, werden auch verbraucht!“

In SAFe werden viele Rollen, Levels und Events unterschieden. Das erhöht das Risiko, dass dieses Framework sehr unflexibel und schwerfällig wird, ähnlich klassischen Prozessen. Dem agilen Prinzip No.10 „Simplicity“ aus dem „Agilen Manifest“ wird diese Struktur nicht gerecht.

Wer mit SAFe einen Change voranbringen will, der schaut vor allem nach oben, kümmert sich zuerst um die Portfolio- und Programmebene, und weiter unten wird dann die eigentliche Arbeit verrichtet. Insofern ist das zentrale Schaubild von SAFe rein optisch ein Abbild der klassischen Welt: Es sieht ein bisschen wie ein Wasserfall aus, auch wenn ursprünglich eine agile Intention grundlegend für die Entwicklung war.

Unten auf dem großen SAFe-Bild sind zentrale Werte (Alignment, Built-in Quality, Transparency etc.), das Lean-Agile Mindset, WiP-Limits und Ähnliches verborgen. Dies liegt dem Framework SAFe zugrunde, allerdings gehen diese für das agile Arbeiten zentralen Werte aufgrund der sehr umfassenden mehrstufigen Strukturen zuweilen unter.

Die insgesamt komplizierte Struktur von SAFe geht mit einem hohen initialen Schulungsaufwand und damit hohen Kosten einher. Die Frage, die wir uns stellen: Wird durch die Struktur nicht der Blick auf das Wesentliche verstellt, da in SAFe praktisch alles vorgegeben ist? Das beginnt mit strukturgebenden Elementen wie dem „Agile Release Train“, der nicht sehr viel Raum für Selbstorganisation lässt. Auf der anderen Seite sorgt es jedoch für Stabilität und Berechenbarkeit, was viele Großunternehmen von ihrem Selbstverständnis her benötigen (vgl. Vollmer 2017).

Die Vermutung „Wenn ihr SAFe macht, seid ihr safe!“ könnte jedoch mitunter trügerisch sein. Es kann sich in eine Art „Scheinagilität“ ausprägen, bei der die Teams vorab definierte Themen umzusetzen hätten, eine Selbstorganisation aber eigentlich nicht erforderlich ist. Ohne Selbstorganisation jedoch ist jedes agile Vorgehen zum Scheitern verurteilt, ob nun skaliert oder nicht. Das Gegenargument hierzu ist, dass jede selbstorganisierte Einheit in einer größeren Organisation auch eine Art „Silo“ darstellen kann, welches sich vom Rest der Organisation entfernt, und somit auch hier Gefahren lauern.

Jurgen Appelo bläst in ein ähnliches Horn und verpackt seine Kritik in die folgenden Worte: „Someone asked about the difference between SAFe and Agile. That’s like asking about the difference between apple pie and fruit“. Ist SAFe also als eine Art weich gekochte Form des ursprünglich knackig-agilen Vorgehens?

Positiv ist das sehr umfangreiche, im Detail ausgearbeitete, agile Framework, dass auf drei unterschiedlichen Ebenen handelt, bei dem Kunden nichts „Halbgares“ vorgesetzt werden soll.

Sicherheit für Großprojekte kann SAFe nicht garantieren, es ist eher als Einstiegslevel mit eingeschränkter Flexibilität zu verstehen. Es umfasst alle Ebenen einer Organisation, verfügt über hilfreiche Instruktionen, unterstützt die Zusammenarbeit zwischen Teams und sorgt durch den Fokus dafür, dass sich alle auf die Abgabe konzentrieren. Weitere Details finden sich in dem Buch: „Agile Skalierung: ein professioneller Leitfadener“ (Nowotny, 2018).<sup>56</sup>

## LeSS Large Scale Scrum – Der Rennwagen

„LeSS is not less save than SAFe“ ist eine Überzeugung, die gelegentlich anzutreffen ist. Das Framework „LeSS“ (Large-Scale Scrum) wurde von Craig Larman und Bas Vodde entwickelt und hatte seine ersten Einsatzfelder im



Bereich der Finanz- und Telekommunikationsbranche gefunden. Es basiert auf der Grundidee, die Prozesse so einfach und klein wie möglich zu halten, also mit minimalem Aufwand mehrere Teams „zum Laufen zu bekommen“. Das passende Bild ist der Rennwagen, der mit weniger Ballast und einer optimalen Trimmung

<sup>56</sup> <https://www.amazon.de/Agile-Skalierung-professioneller-Leitfaden-Organisation/dp/1792868553>

eine sehr hohe Leistungsfähigkeit erreicht. Dazu bedarf es dann lediglich einer Handvoll an Regeln. Laut Craig Larman und Bass Vodde gibt es sogar nur eine primäre Regel für alle scaled frameworks: „Lassen Sie die Finger davon! Und zwar dann, wenn es auch ohne Skalierung geht! Anders gesagt: Gehen Sie öfter zu Fuß und nutzen Sie Transportmittel nur dann, wenn es sich nicht vermeiden lässt. So lebt es sich gesünder!“

## Grundgedanken

Die Grundidee: Was muss ich zu Scrum noch dazugeben, damit es mit mehreren Teams funktioniert? Ein wichtiger Grundgedanke ist, bereits funktionierende Scrum-Teams nicht nach Projektende aufzulösen, sondern als langfristig agierende „scrum facility“ zu etablieren. Diese können in verschiedenen Projekten tätig sein. Es gibt mehrere Scrum Teams mit nur einem Product Owner und einem gemeinsamen Backlog. Die Sprints laufen parallel mit dem Ziel, alle Ergebnisse in ein einziges, gemeinsames PSPI (Potentially Shippable Product Increment) zu implementieren. Die Scrum-Events „Sprint Planning“, „Sprint Review“ und „Sprint-Retrospektive“ finden in allen Teams parallel statt.

Der Prozess ist sehr stark an Scrum angelehnt. Ähnlich dem Scrum Guide handelt es sich bei LeSS um ein sehr reduziertes, man könnte auch sagen kondensiertes Konzept. Der Body of Knowledge, der dann in der Umsetzung von allen Beteiligten einzubringen ist, ist dabei wesentlich größer. LeSS kann in der Basisvariante für bis zu acht Teams skaliert werden.

Darüber hinaus gibt es noch „LeSS Huge“. Hier können mehr als acht Teams skaliert werden. Wenn mehr als acht Teams an einem Product Backlog (BP) arbeiten, wird dieser in mehrere Area-Product- Backlogs (A-PBs) unterteilt. Jedem A-PB werden vier bis acht Teams zugeordnet, und jede Area bekommt dann einen eigenen Area Product Owner (A-PO).

## Besonderheiten

Anders als SAFe verfolgt LeSS die Philosophie, dass so wenig wie möglich im Vorhinein festgelegt wird. Weniger ist also mehr, wie der Name hier schon andeutet. Die wichtigsten Prinzipien lauten:<sup>57</sup>

1. **Large-Scale Scrum ist Scrum** → Um LeSS zu verstehen und erfolgreich anzuwenden, muss man Scrum verstehen, denn LeSS setzt auf Scrum Konzepte auf und erweitert diese, um Scrum erfolgreich zu skalieren.
2. **Transparenz** → Transparenz zeigt sich durch konkrete Inkremente, kurze Feedback-Zyklen sowie in einer Zusammenarbeit auf Augenhöhe und Offenheit aller Beteiligten.

<sup>57</sup> <https://digitalneuordnung.de/blog/less-framework/>

3. **More with LeSS** → Ein Wortspiel, das es in sich hat. Statt mehr Rollen, mehr Meetings, mehr Artefakte, stärkt LeSS die Verantwortung der umsetzenden Teams, um mehr durch weniger zu erreichen.
4. **Whole Product Focus** → Strikte Fokussierung auf ganzheitliches Produkt, statt einzelner technischer Komponenten, was wiederum weitreichende Auswirkungen hat und sich in einem Backlog, einem Product Owner und einem Sprint für alle Teams manifestiert.
5. **Kundenzentrierung** → Dieses Prinzip stellt den Kunden ins Zentrum und alle Teams und handelnden Personen sind aufgefordert, den tatsächlichen Wert für den Kunden im Auge zu behalten.
6. **Kontinuierliche Verbesserung** → Permanentes Lernen und Streben nach Perfektion, sowohl Lernen als auch die kontinuierliche Verbesserung jedes Einzelnen, der Teams, sowie der gesamten Organisation.
7. **Lean Thinking** → Am wirklichen Ort des Geschehens und nicht aus der Führungsetage heraus zu entscheiden (sogenannter „Gemba-Walk“), das dreistufige Lernkonzept ShuHaRi (Shu = nachahmen, Ha = variieren, Ri = eigene Regeln definieren) sowie Respekt gegenüber Menschen.
8. **Systems Thinking** → Das ganze System, nicht einzelne Teile stehen im Vordergrund, d.h. keine lokalen Optimierungen, sondern das große Ganze und globale Optimierungen.
9. **Empirische Prozesskontrolle** → kontinuierliches Inspizieren und Adaptieren des Produkts und des Arbeitsprozesses sowie Anpassung des weiteren Vorgehens aufgrund echter empirischer Erfahrungen.
10. **Queuing-Theorie** → Verständnis für die Effekte von Warteschlangen auf Systeme verbunden mit der Aufforderung WIP-Limits (also die Menge gleichzeitig zu bearbeitender Arbeiten) zu definieren sowie Multitasking und große Arbeitspakete zu vermeiden.

Letzteres bedeutet, dass der Product Owner auch Dinge wegwerfen soll, wenn die Schlange zu lang wird, um so den Durchsatz zu erhöhen. Jedes Team arbeitet also erst mal mit „echtem Scrum“, wobei im Sprint Planning und bei der Retrospektive die folgenden Abweichungen vorgesehen sind:

Das Sprint Planning wird in zwei Teile aufgesplittet. Im ersten Teil treffen sich Repräsentanten aus den Teams zu einer gemeinsamen Sitzung, in der geplant wird, welche Product Backlog Items (kurz „PBIs“) aus dem Backlog in den Sprints umgesetzt werden sollen. Im zweiten Teil beschließen dann die Teams intern, wie die PBIs im kommenden Sprint umgesetzt werden, und bauen dann ihr eigenes teamspezifisches Sprint Backlog auf.

Auch die Retrospektive wird in zwei Teile aufgeteilt. Der erste Teil ist die interne Retrospektive jedes Scrum-Teams. Im zweiten Teil treffen sich dann die Repräsentanten aus jedem Team, um gemeinsam eine Retrospektive abzuhalten, um Punkte zu identifizieren, die nicht vom Team selbst gelöst werden können.