

# Scaling Agile!

## Agile Frameworks auf Unternehmensebene

Urs Böhm, Noser Engineering

### Agile Teams

Mit der Jahrtausendwende setzte sich das agile Manifest und in dessen Folge agile Methoden wie Scrum und Kanban in der Softwareentwicklung durch. Immer mehr werden traditionelle Methoden wie Wasserfall und V-Modell in den Hintergrund und in Nischen getrieben. Neue junge gut ausgebildete Entwickler und Manager erwarten agile Methoden für Projekte und Linienaufgaben der Entwicklung.

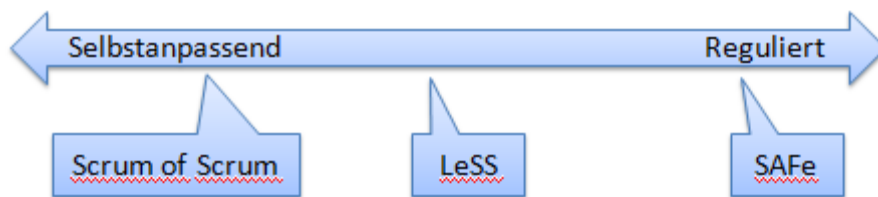
Scrum geht explizit auf die 7 plusminus 2 Teamgrösse ein. Diese hat sich seit der Zeit der römischen Legionen bewährt hat. Hier bildeten auch 8 Legionäre eine "contubernium" (Zeltgemeinschaft) welche verschworen miteinander arbeitete und lebte.

Bei grösseren Software-Vorhaben ist aber ein Team oft zu klein oder kann nicht alle benötigten Fähigkeiten abbilden. Kanban-Entwicklergruppen können etwas grösser werden, aber man verliert dann Teameffekte und ist auch hier irgendwie limitiert. Hier kommt die Skalierung ins Spiel. Was tun wenn man Systeme baut welche dutzende, hunderte oder mehr Personen benötigen, dennoch die Vorteile einzelner Teams nutzen will?

### Man muss Skalieren!

Auf drei verschiedene bewährte Frameworks werde ich eingehen:

- SAFe <http://www.scaledagileframework.com/#>
- LeSS <https://less.works/>
- Scrum of Scrum (Diversität der Webseiten)



Man benötigt Management Support zur Transformation.

Hinter allen dieser Frameworks steht eine aktive Community welche Hilfestellung gibt und die Frameworks anpasst und weiterentwickelt. Das Material ist weitgehend Copyright geschützt und auf den Seiten im Internet frei verfügbar. Zu allen diesen Frameworks gibt es auch unzählige Videos im Internet.

### Vom Projektmanagement zum Flowmanagement

Ein Kollege und SW-Architekt hat mir die Augen geöffnet mit den Worten: "Software wird nie fertig, man entwickelt sie einfach immer weiter..."

Im Bereich der skalierten agilen Frameworks tendiert Software dazu als eher im Flow, als im Projekt entwickelt zu werden. Dies muss man analysieren, verstehen und anwenden. Wenn ich eine Flow basierte Aufgabe habe muss ich mit anderen Methoden vorgehen, als bei einer Projekt basierten Vorgehensweise.

## **Risiken, Kosten, Planung**

1. Projekt, Produkt wird nicht fertiggestellt (gecanceled)
2. Produkt wird nicht genutzt (released)
3. Time to market. Die Features welche die höchste Wertschöpfung haben, sollten zuerst zur Verfügung stehen.

Hier kommt der wichtige Bereich der Budgetierung und Priorisierung zur Geltung. Dies wird bei grossen Projekten, die entsprechend viele Mitarbeiter benötigen, auf Unternehmensebene getroffen und müssen ebenfalls agil sein. Es ist ein hohes Risiko die Budgetierung und Planung auf Unternehmensebene klassisch zu machen, aber das Produkt agil entwickeln zu wollen. Bei kleinen Projekten kann man es im Griff haben, bei skalierten meiner Meinung nach nicht!

## **Fazit**

Der Consultant sagt oft: "Es kommt darauf an." Es gibt keine "Silver-Bullet" oder "One Size fit's all" Lösung. Die aktuelle Situation, die Herausforderung bestimmen die optimale Lösung. Alle drei Frameworks funktionieren bei korrekter Umsetzung in einer geeigneten Umgebung. Bevor Sie umsteigen analysieren und prüfen Sie bitte Ihre Situation. Tauschen Sie sich mit Fachleuten aus. Stellen Sie sicher, dass Sie in einer geeigneten Umgebung sind. Falls nicht, müssen Sie die Umgebung schaffen, was aufwendig und kräftezehrend sein kann.

Halten Sie bei radikalen Umstellungen Reserven vorrätig.

Möglicherweise ist Ihre Software - gerade im Embedded Bereich - gar nicht so gross, dass diese Frameworks angemessen sind. Prüfen Sie bitte ob eine geeignetere Zuschneidung der Gesamtaufgabe in kleinere Teilaufgaben mit definierten System- und Projektgrenzen nicht zielführender ist.

## **Links:**

LeSS vs SAFE Vis a vis Update

<https://www.youtube.com/watch?v=d5NtO56XX2E>

The Universe of "Agile" Methods

<https://www.youtube.com/watch?v=jngGYqvpnVs>

GOTO 2015 Why Scaling Agile Doesn't Work

<https://www.youtube.com/watch?v=2zYxWEZ0gYg>

GOTO 2015 Agile is Dead Pragmatic Dave Thomas

<https://www.youtube.com/watch?v=a-BOSpxYJ9M>

## **Autor**



Der Referent **Urs Böhm** hat an der Universität des Saarlandes Elektrotechnik studiert. Er ist seit über 15 Jahren in Embedded Soft-, Firm- und Hardware Projekten als Projektleiter, Berater und Entwickler tätig. Er bildet sich im Projektmanagement kontinuierlich weiter und hält u.a. Zertifikate als Projektingenieur VDI, Certified ScrumMaster und Projektleiter IPMA Level C. Er hielt bereits Vorträge am Embedded Software Engineering Kongress und weiteren Konferenzen.