

# The Scrum@Scale Guide

The Definitive Guide to Scrum@Scale:  
The Rules of the Game

Draft Version 0.93 – January 2018

## Table of Contents

<b>Purpose of the Scrum@Scale Guide .....</b>	<b>3</b>
<i>Why Scrum@Scale? .....</i>	3
<i>Definition of Scrum@Scale.....</i>	3
<i>Values-Driven Culture .....</i>	5
<i>Getting Started With Scrum@Scale.....</i>	6
<b>The Scrum Master Cycle.....</b>	<b>6</b>
<i>Team-Level Process .....</i>	6
Coordinating the “How” – The Scrum of Scrums.....	6
The Scrum of Scrums Master (SoSM) .....	8
Scaling the SoS .....	8
<i>The Executive Action Team.....</i>	9
The EAT’s Backlog & Responsibilities .....	10
Outputs/Outcomes of the Scrum Master Organization .....	11
<b>Product Owner Cycle .....</b>	<b>11</b>
<i>Coordinating the “What” – The MetaScrum.....</i>	11
The Chief Product Owner (CPO) .....	12
Scaling the MetaScrum .....	12
<i>The Executive MetaScrum (EMS).....</i>	13
Outputs/Outcomes of the Product Owner Organization.....	14
<i>Understanding Feedback .....</i>	15
<i>Metrics &amp; Transparency.....</i>	15
Some notes on Organizational Design.....	16
<i>Acknowledgements .....</i>	17

# Purpose of the Scrum@Scale Guide

Scrum, as originally outlined in the Scrum Guide, is a framework for developing, delivering, and sustaining complex products by a single team. Since its inception, it's usage has extended to the creation of products, processes, services, and systems that require the efforts of multiple teams. Scrum@Scale was created to efficiently coordinate this new ecosystem of teams. It achieves this goal through setting up a “minimum viable bureaucracy” via a “scale-free” architecture. This guide contains the definitions of the components that make up the Scrum@Scale framework, including its scaled roles, scaled events, and enterprise artifacts, as well as the rules that bind them together.

Dr. Jeff Sutherland developed Scrum@Scale based on the fundamental principles behind Scrum, Complex Adaptive Systems theory, game theory, and object-oriented technology. Combined with the results of field work with dozens of companies from startups to those in the Fortune 100, this guide was developed with the input of many experienced Scrum practitioners with the goal of the reader being able to implement Scrum@Scale on their own.

## Why Scrum@Scale?

Scrum was designed for a single team to be able to work at its optimal capacity at a sustainable pace. In the field, it was found that as the number of Scrum teams within an organization grew, the optimal output (working product) and velocity of those teams began to fall (due to issues like cross-team dependencies and duplication of work). It became obvious that a framework for effectively coordinating those teams was needed in order to enable linear scalability. Scrum@Scale is designed to accomplish this goal. The main features that ensure this outcome are its “scale-free” architecture.

By utilizing a “scale-free” architecture, the organization is not constrained to grow in any particular way decided upon by a set of arbitrary rules; rather it can grow organically based on its unique needs and at a sustainable pace of change that can be accepted by the groups of individuals that make up the organization.

Scrum@Scale is designed to scale across the organization as a whole: all departments, products and services. It can be applied in all types of organizations in industry, government, or academia.

## Definition of Scrum@Scale

Scrum (n): A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.

The Scrum Guide is the minimal feature set that allows inspection and adaptability via radical transparency to drive innovation, performance, and team happiness.

Scrum@Scale (n): A framework within which networks of Scrum teams operating consistently with the Scrum Guide can address complex adaptive problems, while creatively delivering products of the highest possible value.

**NOTE:** These “products” may be hardware, software, complex integrated systems, processes, services, etc., depending upon the domain of the Scrum teams.

Scrum@Scale is:

- Lightweight – the minimum viable bureaucracy
- Simple to understand – consists of only Scrum teams
- Difficult to master – requires implementing a new operating model

Scrum@Scale is a non-prescriptive meta-model for scaling Scrum. It radically simplifies scaling by using Scrum to scale Scrum. It consists only of Scrum teams coordinated via Scrum of Scrums and Meta Scrums.

The component-based nature of Scrum@Scale allows an organization to customize their transformational strategy and implementation. It gives them the ability to target their transformation efforts in the area(s) they deem most valuable or most in need of change and then progress on to others.

In Scrum, care is taken to separate accountability of the “what” from the “how”. The same care is taken in Scrum@Scale so that jurisdiction and accountability are expressly understood in order to eliminate wasteful organizational politicking that keep teams from achieving their optimal productivity.

In separating these two jurisdictions, Scrum@Scale contains two cycles: the Scrum Master Cycle (the “how”) and the Product Owner Cycle (the “what”), each touching the other at two points. Taken together, these cycles produce a powerful framework for coordinating the efforts of multiple teams along a single path.



## Getting Started With Scrum@Scale

When implementing large networks of teams, it is critical to develop a scalable **Reference Model** for a small set of teams. Any deficiencies in a Scrum team implementation will be magnified when multiple teams are deployed.

Therefore, the first leadership challenge is create a small set of teams that implements Scrum well. This set of teams works through organizational issues that block agility and creates a **Reference Model** for Scrum that is known to work in the organization and can be used as a pattern for creating Scrum teams across the organization.

As soon as a set of teams accelerates, blocks, impediments, and bottlenecks appear in the organization that delay delivery across a value stream. The most effective way to eliminate these problems is to spread Scrum across the organization so that the entire value stream is optimized. Scrum@Scale enables the saturation of an organization with Scrum.

Finally, as the number of Scrum teams increases, Scrum@Scale can enable linear scalability through effective communication, coordination, and escalation of impediments as opposed to other approaches that produce increasingly less output as number of teams increases.

Achieving optimal results by scaling velocity, saturating the organization with Scrum, distributing velocity and quality, and enabling linear scalability are dependent on an organization's specific strategy, product and services.

## The Scrum Master Cycle

### Team-Level Process

The **Team-Level Process** is laid out clearly in the Scrum Guide. It is composed the three artifacts (Product Backlog, Scrum Backlog, and Potentially Shippable Increment of Product), five events (Product Backlog Refinement, Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective), and three roles (Product Owner, Scrum Master, Team).

The goals of the team level process are to:

- Maximize the flow of completed and quality tested work
- Attempt to increase velocity a little each sprint
- Operate in a way that is sustainable and enriching for the team

### Coordinating the “How” – The Scrum of Scrums

A set of the teams that have a need to coordinate comprise a “**Scrum of Scrums**” (**SoS**). The SoS is a “team of teams” (McChrystal, 2015), which hold a **Scaled Daily Scrum (SDS)** event during which the teams each send a representative (usually the

teams of scrum master, although any person or number of people may attend). The SDS exists to coordinate teams and remove impediments to delivering value.

The Scaled Daily Scrum event mirrors the Daily Scrum in that it optimizes the collaboration and performance of the network of teams. Additionally, the SDS:

- it is time-boxed to 15 minutes or less
- a representative of each team must attend and contribute
- the representative answers 3 simple questions
  - What impediments does my team have that will prevent them from accomplishing their Sprint Goal (or impact the upcoming release)?
  - Is my team doing anything that will prevent another team from accomplishing their Sprint Goal (or impact their upcoming release)?
  - Have we discovered any new dependencies between the teams or discovered a way to resolve an existing dependency?

This team of Scrum Masters is a Scrum Team unto itself which is responsible for a fully integrated set of potentially shippable increments of product at the end of every sprint from all participating teams. A SoS functions as a Release Team and must be able to directly deliver value to customers. To do so effectively, it needs to be consistent with the Scrum Guide; that is, have its own roles, artifacts, and events. It needs all the skills necessary to deliver a fully integrated potentially shippable product at the end of every Sprint. It has Product Owner representation to resolve prioritization issues. It may need experienced architects, QA Leaders, and other operational skill sets.<sup>1</sup>

The main artifact of the SoS is the Chief Product Owners backlog which is distributed across teams. The Scrum of Scrums Master (SoSM) as delivery manager must assure that any impediments to completing a fully integrated potentially shippable product at the end of every Sprint is removed.

The SoS team needs to be responsive in real-time to impediments raised by the team. The Scrum Master for the SoS team should endeavor to make an impediment backlog visible to the organization, both in terms of its contents and the progress being made on each impediment. Besides the team's SDS, they should also hold a scaled version of each Scrum event (Sprint Planning, Sprint Review, & Retrospective) and have some type of Backlog Refinement session wherein they decide when the impediments are "ready" to be removed, how best to remove them, and how the team will know it is "done". Particular attention should be paid to the SoS Retrospective in which the teams' representatives share any learnings or process improvements (aka. Kaizens) that their individual teams have succeeded with in order to standardize those practices across the SoS teams.

---

<sup>1</sup> When starting Scrum@Scale the teams may not have an infrastructure that supports continuous deployment. This will force the SoS to set up an "integration team" or "release team" that generates the extra work required to overcome engineering deficiencies. For example, Amazon has 1000 Scrum teams that deliver a new feature to production every second without an integration team. The ROI of investment in automation to enable continuous deployment is on the order of 72000% (Rico, 2017). This astronomical ROI will eventually force every company to automate to be competitive.

## The Scrum of Scrums Master (SoSM)

The Scrum of Scrums Master (SoSM) is essentially responsible for the release of the joint teams' effort. As a servant-leader of this team of teams, they accomplish this by assuring that the Chief Product Owner's backlog is properly distributed across teams and any impediments raised by the teams which they cannot handle are resolved in priority order, particularly cross-team dependencies. The SoSM acts as a "coach of coaches" and bears responsibility for improving the performance of the Scrum of Scrums, working closely with the Chief Product Owner, and delivering a potentially shippable product every sprint or more often.

The SoSM serves the teams and the organization in several ways:

- Ensures that the SoS teams' increments can be integrated each Sprint
- Prioritizes the backlog of impediments
- Is accountable for eliminating impediments
- Can be one of the team SM's or someone external to the teams
- Works closely with the Product Owners to coordinate the team's Deployment with their Release Plans (see pages 10 & )

## Scaling the SoS

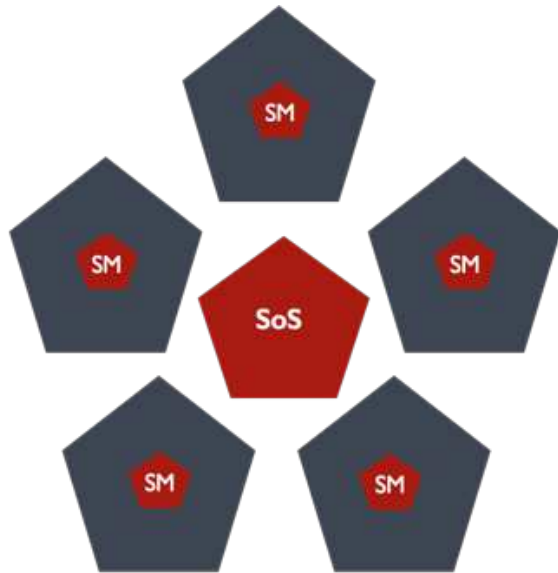
Depending upon the size of the organization or implementation, more than one SoS may be needed to deliver a very complex product. In those cases, a **Scrum of Scrum of Scrums (SoSoS)** can be created out of multiple Scrums of Scrums.

Scaling the SoS reduces the number of communication pathways within the organization so that complexity is encapsulated. A SoS looks like a single team to a SoSoS which can assess the delivery of a value stream to the customer that looks like a single product even though multiple SoS are required to create it.

This can be repeated with as many teams as needed in an organic pattern so that a richness of communication is achieved with a minimal number of communication pathways. These SoSoS should also have SoSoSM's and scaled versions of each artifact & event.

Sample Diagrams:





SoS perspective of 5 Teams



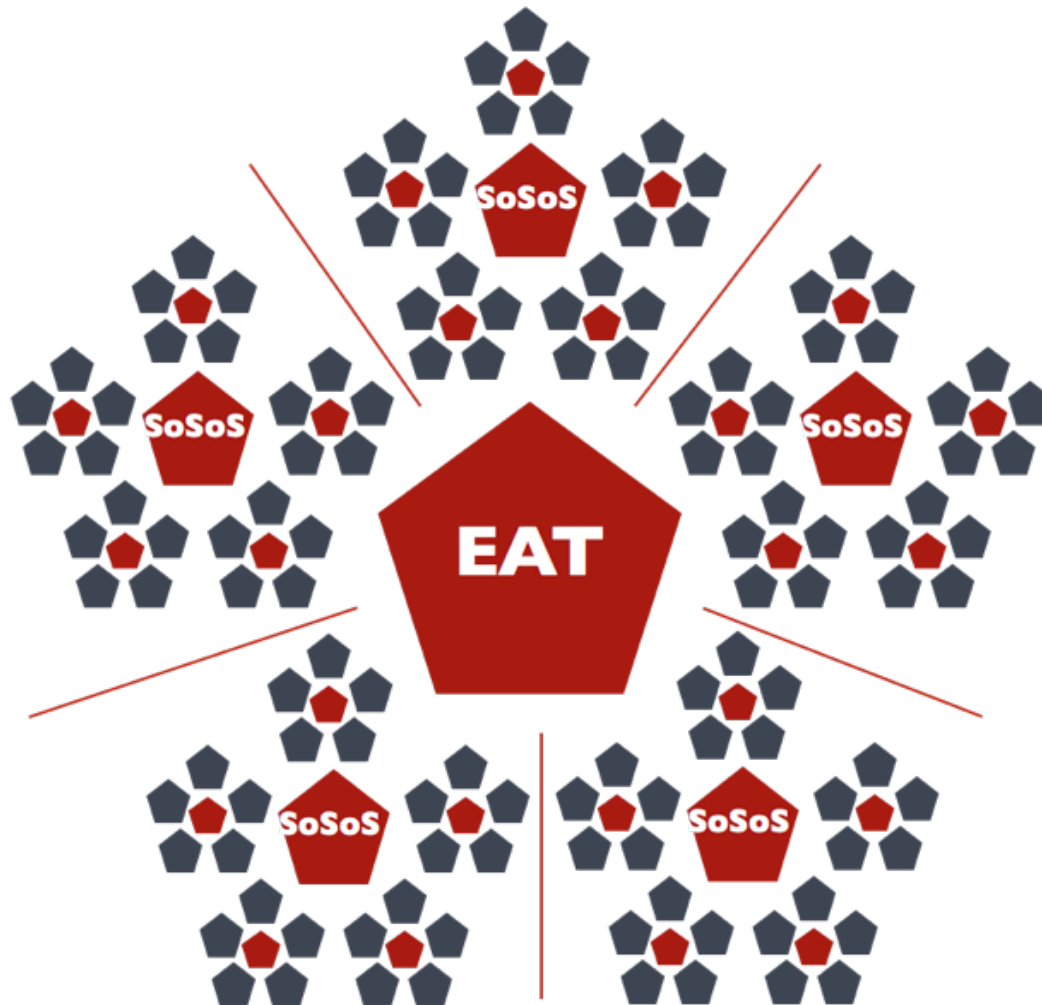
SoSoS perspective of 25 Teams

**NOTE:** While the Scrum Guide defines the optimal team size as being 3 to 9 people, Harvard research determined that optimal team size is 4.6 people. (Hackman 2002). Experiments with high performing Scrum teams has repeatedly shown that 4 or 5 people who are doing the work is the optimal size. It is essential to linear scalability that this pattern be the same for number of teams in a SoS. Therefore, in the above and following diagrams, pentagons were chosen to represent a team of 5. These diagrams are meant to be examples only, your organizational diagram may differ greatly.

## The Executive Action Team

The Scrum of Scrums enables a network design of Scrum teams which is infinitely scalable. The Scrum of Scrums for the entire agile organization is called the **Executive Action Team (EAT)**. The EAT is the final stop for impediments that cannot be removed by the SoS's that feed it. Therefore, it must be comprised of individuals who are empowered, politically and financially, to remove them. The function of the EAT is to coordinate multiple SoS's (or SoSoS's). As with any Scrum team, a PO and SM need to be chosen from its component members. They must meet at least once per sprint and have a transparent backlog.

Sample Diagram showing an EAT coordinating 5 groupings of 125 teams:



## The EAT's Backlog & Responsibilities

The first part of the EAT's backlog are the impediments that flow into it. Via the SoS's, the entire SM organization reports into the EAT and as such it is responsible for supporting and enhancing the overall agility of the organization.

Since Scrum is an agile operating system that is different from traditional project management. It has different rules of operation, different APIs, and different procedure calls to get work done. The Executive Action Team is responsible for implementing this agile operating system by establishing, maintaining, and enhancing the agile implementation in the organization. The EAT's role is to create an Organizational Transformation Backlog (a prioritized list of the agile initiatives that need to be accomplished) and see that it is carried out.

The EAT's responsibilities include but are not limited to:

- Creating an agile operating system for the Scrum part of the organization
- Being accountable for the quality of Scrum in the organization
- Setting and monitoring the metrics that will determine that quality

- Owning the training and coaching competency within the organization
- Creating a center for continuous learning for Scrum professionals
- Creating and supporting a Product Owner structure within the organization
- Changing the corporate operational rules, procedures, and guidelines to enable agility within the Scrum part of the organization

This final bullet point is a major responsibility of the EAT. The goal is to set up and support a corresponding Product Owner organization through associations of PO's that mirror the SoS's and scale their PO functions. These teams of PO's are known as **MetaScrums** (see page 9).

## Outputs/Outcomes of the Scrum Master Organization

The SM organization (SoS, SoSoS, and EAT) work as a whole to complete the other components of the Scrum Master Cycle: **Continuous Improvement and Impediment Removal, Cross-Team Coordination, and Deployment.**

The goals of Cross-Team Coordination are:

- Coordinate similar processes across multiple related teams
- Manage cross-team dependencies to ensure they don't become impediments
- Maintain alignment of team norms and guidelines for consistent output

The goals of Continuous Improvement and Impediment Removal are:

- Identify impediments and reframe them as opportunities
- Maintain a safe and structured environment for prioritizing and removing impediments, and then verifying the resulting improvement
- Ensure visibility in the organization to effect change

Since the goal of the SoS is to function as a release team, the deployment of product falls under their scope, while what is contained in any release falls under the scope of the Product Owners. Therefore, the goals of the Deployment are:

- Deliver a consistent flow of valuable finished product to customers
- Integrate the work of different teams into one seamless product
- Ensure high quality of the customer experience

# Product Owner Cycle

## Coordinating the “What” – The MetaScrum

Groups of Product Owners who need to coordinate a single backlog form teams called **MetaScrums**. The MetaScrum aligns the teams' priorities along a single path so that they can coordinate their backlogs. MetaScrums hold a scaled version of Backlog Refinement.

- Each team PO (or proxy) must attend

The forum for Leadership, Stakeholders, or other Customers to express their preferences

- The meeting is time-boxed to 2 hours or less per week of Sprint length
- They may meet as often as needed, but usually once per Sprint

The main functions of the MetaScrum are to:

- create an overarching vision for the product & make it visible to the organization
- generate a single, prioritized backlog; ensuring that duplication of work is avoided

Create a uniform “Definition of Done” that applies to all teams

- eliminate dependencies raised by the SoS’s
- generate a coordinated Release Plan
- deciding upon and monitoring metrics that give insight into the product

MetaScrums, just like SoS’s, function as Scrum teams on their own. As such, they need to have someone who acts as SM and keeps the team on track in discussions, and a single person who is responsible for coordinating the generation of a single overall Product Backlog for all of the teams covered by the MetaScrum. This person is designated as the **Chief Product Owner**.

## The Chief Product Owner (CPO)

Through the MetaScrums, Chief Product Owners drive production in an aligned manner. Chief Product Owners coordinate priorities among Product Owners who work with individual teams. They align backlog priorities with Stakeholder/Customer needs. Just like a SoSM, they may be an individual team PO who chooses to play this role as well, or they may be a person specifically dedicated to this role. Their main responsibilities are the same as regular PO’s, but at scale:

- Setting a strategic vision for the whole product
- Creating a single, prioritized backlog of value to be delivered by all of the teams
- these items would be of a greater granularity than that for a team PO
- Work closely with the SoSM so the Release Plan that the MetaScrum team generates can be deployed efficiently
- Monitor incoming product feedback and adjust the backlog accordingly

## Scaling the MetaScrum

Just as SoS’s can grow into SoSoS’s, MetaScrums can also be multiplied by the same mechanism. There is no specific term associated with these multiplied units, nor do the CPO’s of them have specific expanded titles. We encourage each organization to develop their own. For the following diagrams, we have chosen to add an additional “Chief” to the title of those PO’s as they magnify out.

Some sample diagrams:



MetaScrum perspective of 5 Teams

Teams

ADD LUMPY DIAGRAM HERE



MetaScrum perspective of 25

## The Executive MetaScrum (EMS)

The MetaScrums enables a network design of Product Owners which is infinitely scalable alongside their associated SoS's. The MetaScrum for the entire agile organization is the **Executive MetaScrum**. The EMS owns the organizational vision and sets the strategic priorities for the whole company, aligning all the teams around common goals.

Sample diagram showing an EMS coordinating 5 groupings of 125 teams:



## Outputs/Outcomes of the Product Owner Organization

The PO organization (various MetaScrums, the CPO's & the Executive MetaScrum) work as a whole to satisfy the components of the Product Owner Cycle: **Strategic Vision, Backlog Prioritization, Backlog Decomposition & Refinement, Release Planning**.

The goals of setting a Strategic Vision are:

- Clearly align the entire organization along a shared path forward
- Compellingly articulate why the organization exists
- Describe what the organization will do to leverage key assets in support of its mission
- Update continuously based on feedback to outmaneuver the competition

The goals of Backlog Prioritization are:

- Identify a clear ordering for products, features, and services to be delivered
- Reflect value creation, risk mitigation and internal dependencies in ordering of the backlog

The goals of Backlog Decomposition & Refinement are:

- Break complex projects and products into manageable independent functional elements that can be completed by one team in one sprint
- Capture and distill emerging requirements and customer feedback
- Ensure all backlog items are truly “Ready” when they reach sprint backlog
- Parse backlog to individual teams

The goals of Release Planning are:

- Forecast delivery of key features and capabilities
- Communicate snapshot of delivery expectations to stakeholders
- Inform updated prioritization, as needed, based on stakeholder input

## Understanding Feedback

The **Feedback** component is the second point where the PO & SM Cycles touch. Product feedback drives continuous improvement through adjusting the Product Backlog while Release feedback drives continuous improvement through adjusting the Deployment mechanisms. The goals of obtaining and analyzing Feedback are:

- Understand how customers actually use and interact with the product
- Define improvements to existing functionality
- Distill actionable changes in direction from the noise of all responses
- Capture ideas for new features and functionality not previously identified
- Update progress towards product/project completion to refine release planning and stakeholder alignment
- Define improvements to deployment methods and mechanisms

## Metrics & Transparency

The final component deals with **Metrics & Transparency**. Radical transparency is essential for Scrum to function optimally. It gives the organization the ability to honestly assess its progress and to inspect and adapt its products and processes. This is the foundation of the empirical nature of Scrum as laid out in the Scrum Guide.

To ensure optimal activity, both the SM & PO Cycles require metrics that will be decided upon by the separate SM and PO organizations. Metrics are unique to both specific organizations as well as to specific functions within those organizations. Scrum@Scale does not require any specific set of metrics, but it does suggest that at a bare minimum, the organization should track some metric of:

- Productivity – ex. change in amount of Working Product delivered per Sprint
- Value Delivery – ex. business value per unit of team effort
- Quality – ex. defect rate or service downtime
- Sustainability – ex. a team “happiness” metric

The goals of having Metrics and Transparency are to:

- Provide all decision makers including team members with appropriate context to make good decisions
- Shorten feedback cycles as much as possible to avoid over-correction
- Accomplish all of this with minimal additional effort by teams, stakeholders or leadership



## Some notes on Organizational Design

Scrum@Scale is designed to scale productivity, to get the entire organization doing twice the work in half the time with much higher quality and in a significantly improved work environment. Large organizations that can do this well can cut the cost of their products and services while improving quality and innovation.

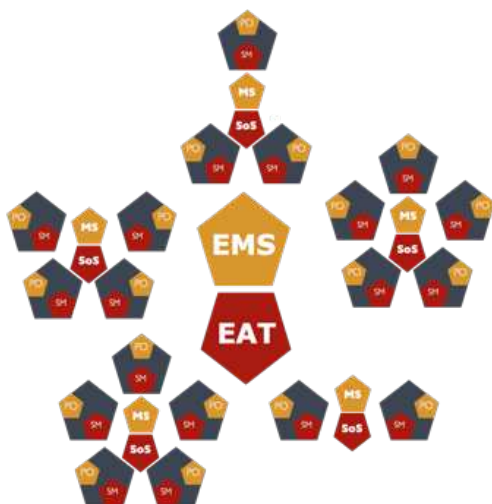
Scrum@Scale is designed to saturate an organization. All teams including Leadership, Human Resources, Legal, Consulting and Training, in addition to product and service teams implementing the same style of Scrum while streamlining and enhancing an organization.

The “scale-free” nature of Scrum@Scale allows the design of the organization to be component-based, just like the framework itself. This permits for the rebalancing or refactoring of teams in response to the market and the need for new products as well as support linear scalability both with size and global distribution as documented in multiple studies (Sutherland, Schoonheim et al. 2007, Sutherland, Viktorov et al. 2007, Jakobsen and Sutherland 2009).

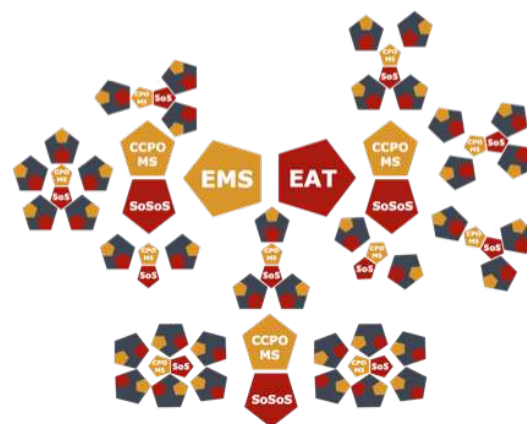
As an organization grows larger, capturing the benefits of distributed teams may be important. Rarely do organizations achieve the supposed cost benefits of lower hourly rates. However, some organizations capture talent otherwise unavailable and are able to expand and contract the organization as needed through outsourced development. Scrum@Scale shows how to do this while avoiding long lag times, compromised communications, and inferior quality.

Well implemented Scrum can run an entire organization.

Some sample diagrams:

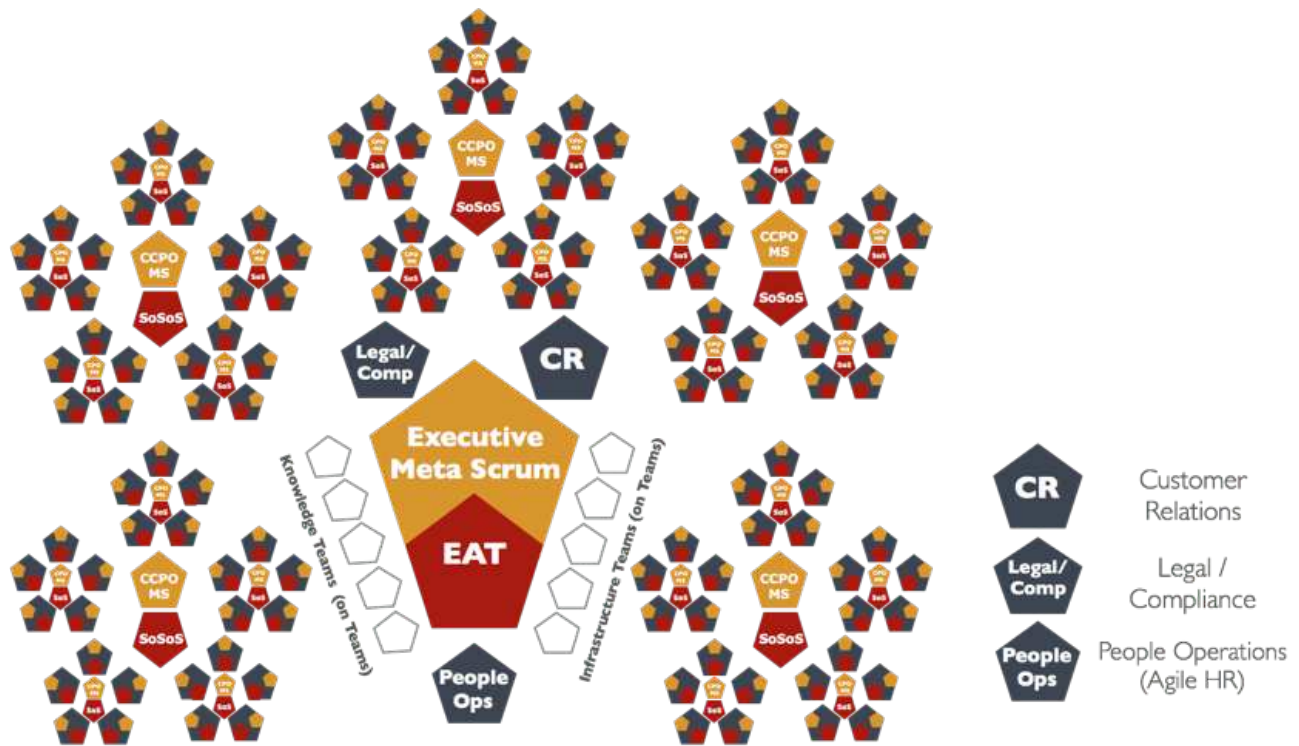


4 SoS's with 2, 3, & 2x5 Teams



3 SoSoS's with 10, 13, & 15 teams





In this organizational diagram, the **Knowledge & Infrastructure Teams** represent virtual teams of specialists of which there are too few to staff each team with. They coordinate with the Scrum teams as a group via service-level agreements where requests flow through a PO for each specialty who creates a transparent ordered backlog of said requests. An important note is that these teams are NOT silos of individuals who sit together (this is why they are represented as hollow pentagons); their component team members sit on the actual Scrum teams, but they make up this virtual Scrum of their own for the purpose of backlog dissemination and process improvement.

**Customer Relations, Legal / Compliance, and People Operations** are included here as well as they are necessary parts of organizations and will exist as independent Scrum teams on their own who all the others may rely upon.

A final note on the representation of the EAT & EMS: in this diagram, they are shown as overlapping since 2 members sit on both of the teams. In very small organizations or implementations, the EAT & EMS may consist of the entirely the same team members.

## Acknowledgements

We acknowledge IDX for the creation of the Scrum of Scrums which first allowed Scrum to scale to 100s of teams (Sutherland 2001), PatientKeeper for the creation of the Metascrum (Sutherland 2005) which enabled rapid deployment of innovative product,

and OpenView Venture Partners for scaling Scrum to the entire organization (Sutherland and Altman 2009. We value input from Intel with over 25000 people doing Scrum who taught us “nothing scales” except a scale-free architecture and SAP with the largest Scrum team product organization who taught us management involvement in the MetaScrum is essential to get 2000 Scrum teams to work together.

The agile coaches and trainers implementing these concepts at Toyota, GE, 3M, Amazon, Spotify, and many other companies working with Jeff Sutherland have been helpful in testing these concepts across a wide range of companies in different domains.

And finally Avi Schneier and Alex Sutherland have been invaluable in formulating and editing this document.

## References

Sutherland, J. (2001). "Agile Can Scale: Inventing and Reinventing Scrum in Five Companies." Cutter IT Journal **14**(12): 5-11.

Sutherland, J. (2005). Future of Scrum: Parallel Pipelining of Sprints in Complex Projects. AGILE 2005 Conference. Denver, CO, IEEE.

Sutherland, J. and I. Altman (2009). Take No Prisoners: How a Venture Capital Group Does Scrum. Agile 2009, Chicago, IEEE.