

# Scrum@Scaleの理論と実装

- 組織をリファクタリングしながらスケールする

Chatwork株式会社 粕谷 大輔 (daiksy)

Regional Scrum Gathering Tokyo 2022 (2022-01-05 Room C 14:00-14:45)

粕谷 大輔(@daiksy / だいくしー)

Chatwork株式会社 エンジニアリングマネージャー

エンジニア組織開発支援

アジャイルコーチ

チームファシリテーション

Scrum@Scale



SNSではこんなアイコンをよく使っています

「かすやばねえ天才」の略だろ？

KPTって



ニア組織開発支援

アジャイルコーチ

ムファシリテーション

Scrum@Scale



**⚠️注意: スクラムのスケーリングについて**

# **⚠️注意: スクラムのスケーリングについて**

- **スクラムのスケーリングは基本的にオススメしません**
- **スケーリングを検討する前に、単一の熟練したスクラムチームを育てましょう**
- **スケーリングを検討する前に、スコープを減らせないのか。1つのチームだけで本当にやれないのかを入念に検討しましょう。場合によってはプロのコーチなどの手を借りましょう**
- **とはいえ、事業の成長と組織のスケールは避けて通れないので、本当にどうしてもスケーリングの必要がある場合は丁寧に進めましょう（ただし単一の成熟したチームがまずはできあがっていることが前提です）**

# 今日お話しすること

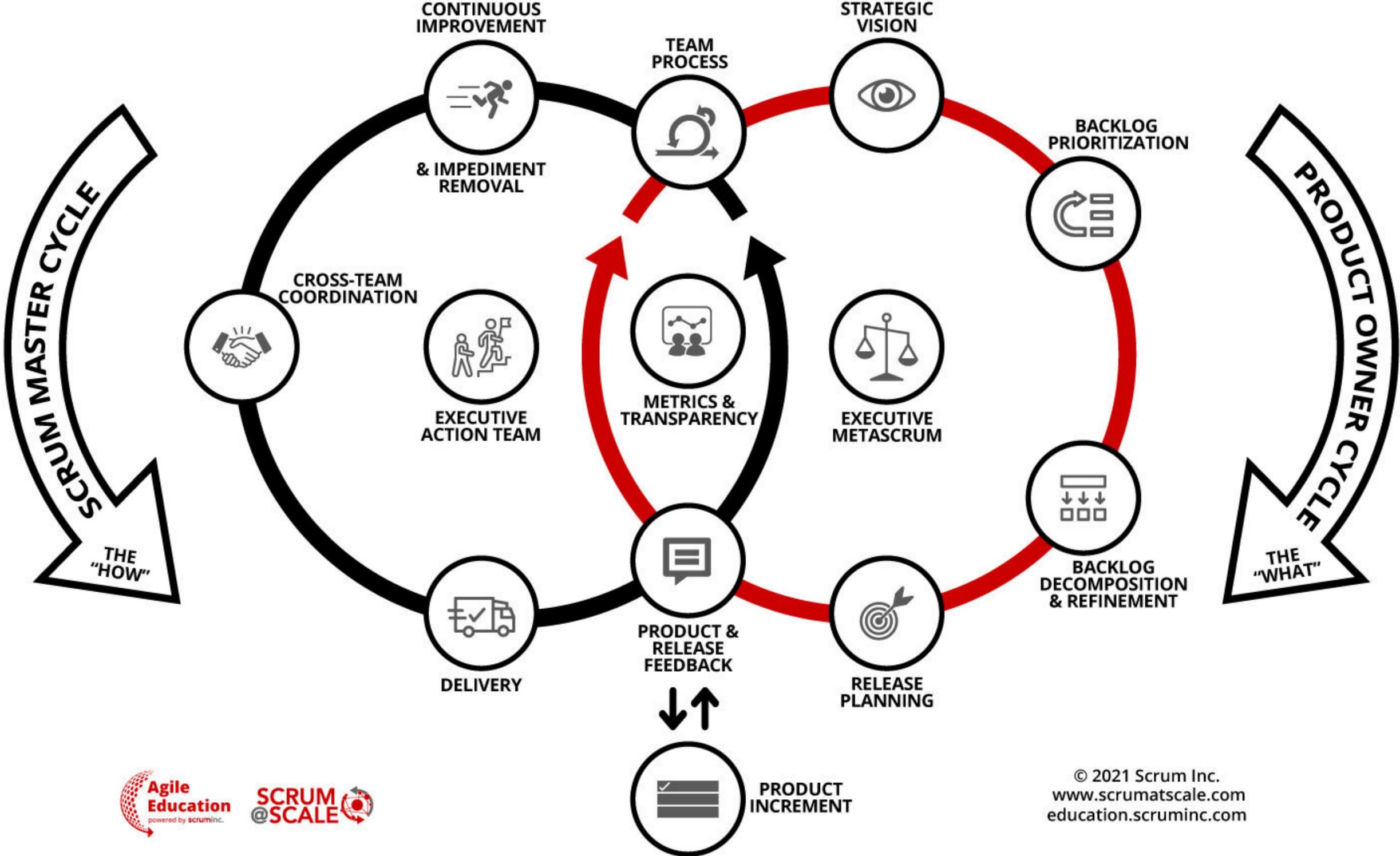
- 前半: Scrum@Scaleの解説
- 後半: Chatworkにおける実例

# Scrum@Scaleの解説

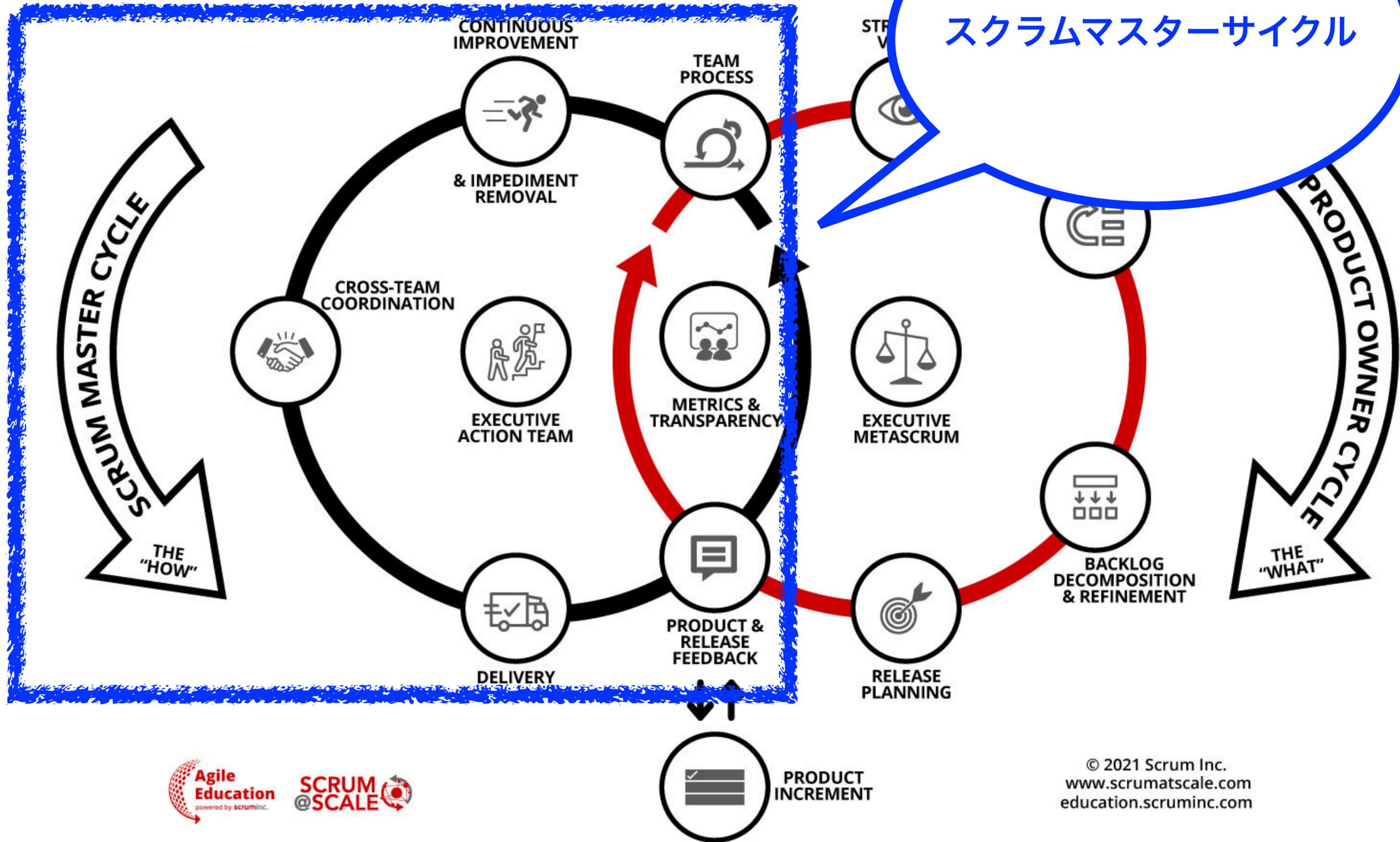
# Scrum@Scaleとは？

- スクラムの共同考案者の1人であるJeff Sutherland博士を中心に確立
- 複数のスクラムチームによるフラクタル構造
- 実用最小限の官僚機構
- HowとWhatそれぞれの取り組みが定義されている（スクラムマスターサイクルとプロダクトオーナーサイクル）
- 個人的にはプロダクトオーナーの活動についての定義があるのが気に入っています

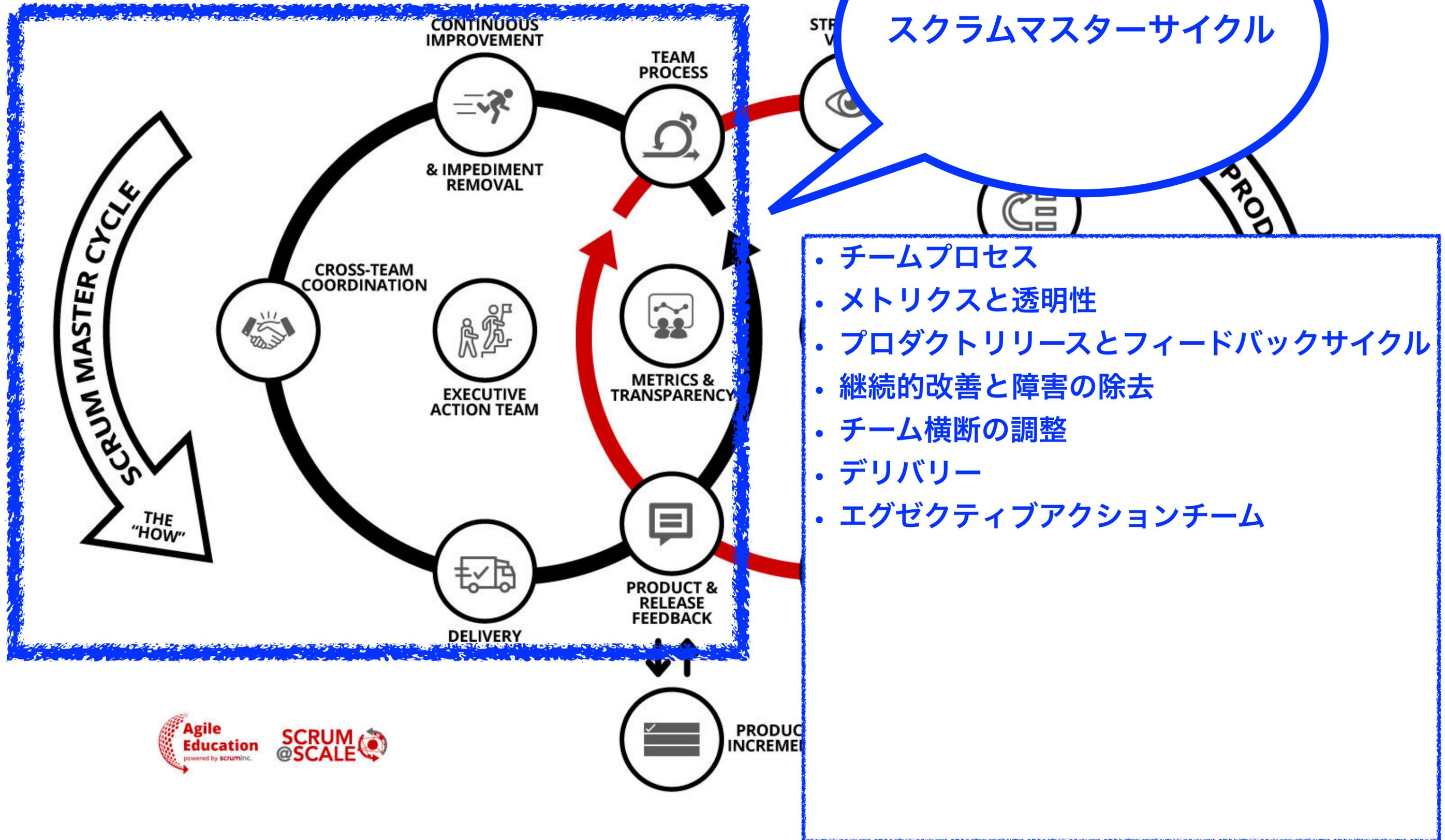
# Scrum@Scale Framework



# Scrum@Scale Framework

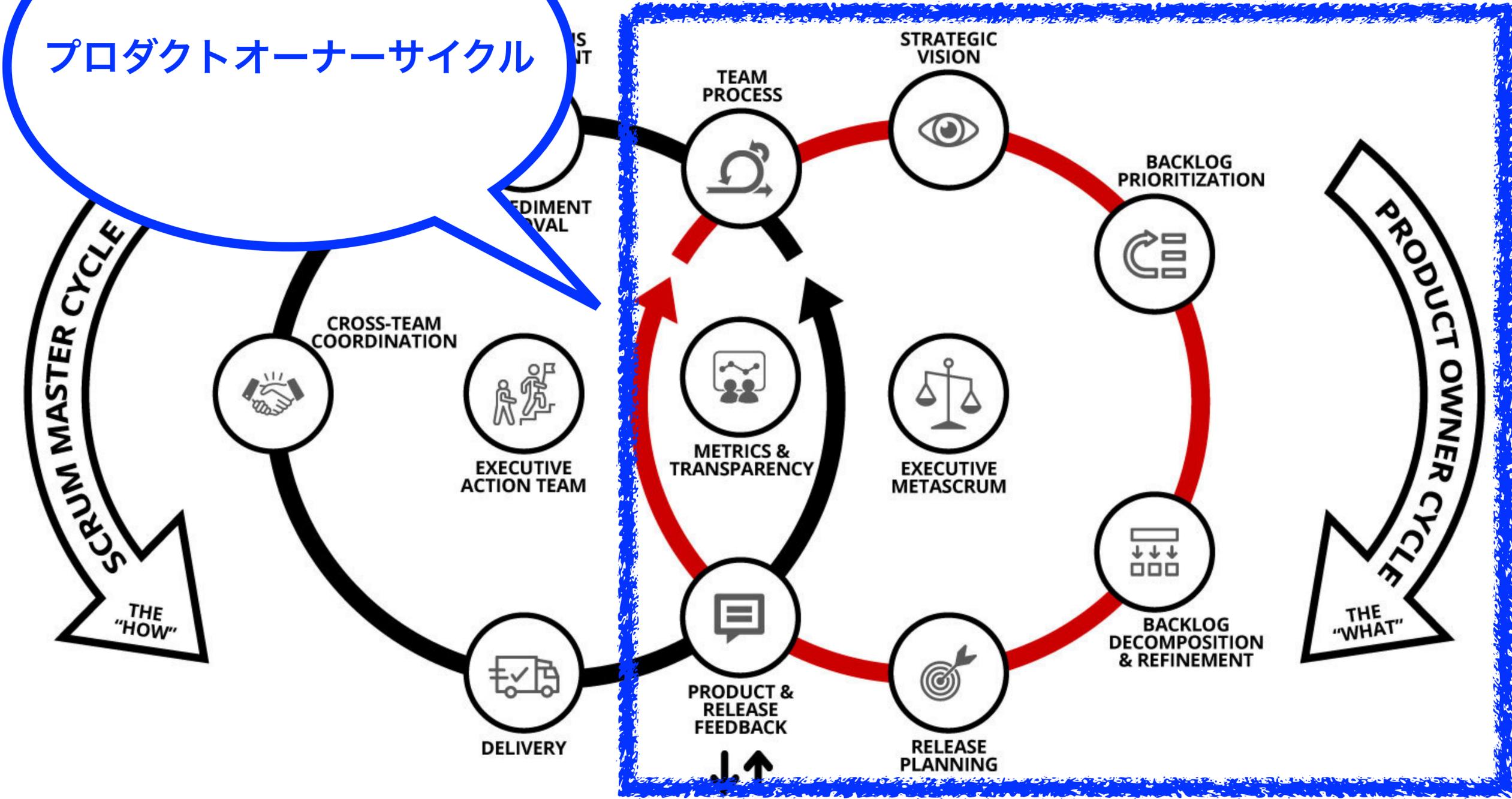


# Scrum@Scale Framework



# Scrum@Scale Framework

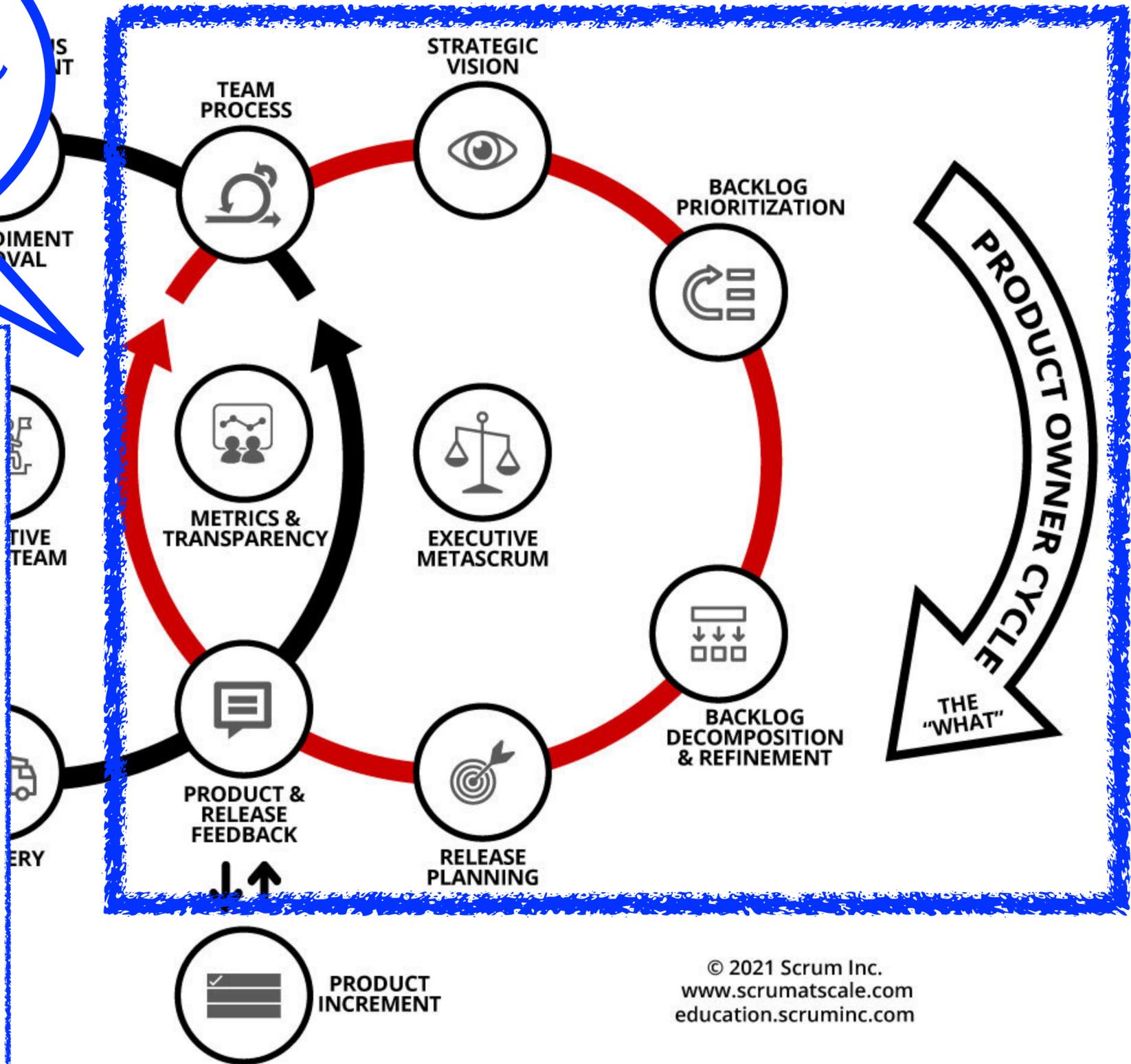
プロダクトオーナーサイクル



# Scrum@Scale Framework

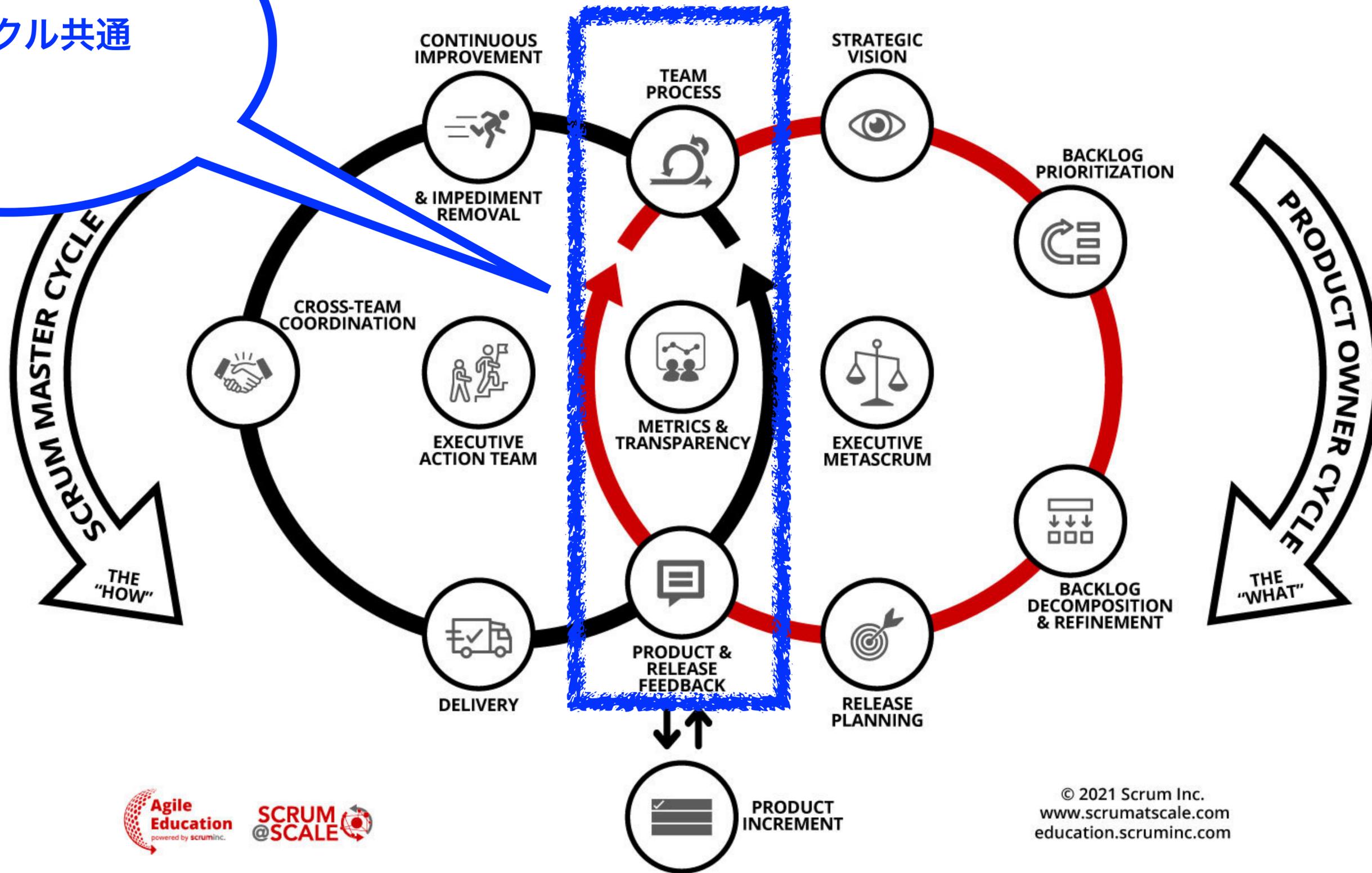
プロダクトオーナーサイクル

- チームプロセス
- メトリクスと透明性
- プロダクトリリースとフィードバックサイクル
- 戦略的ビジョン
- バックログの優先順位付け
- バックログの分割とリファインメント
- リリースプランニング
- エグゼクティブメタスクラム

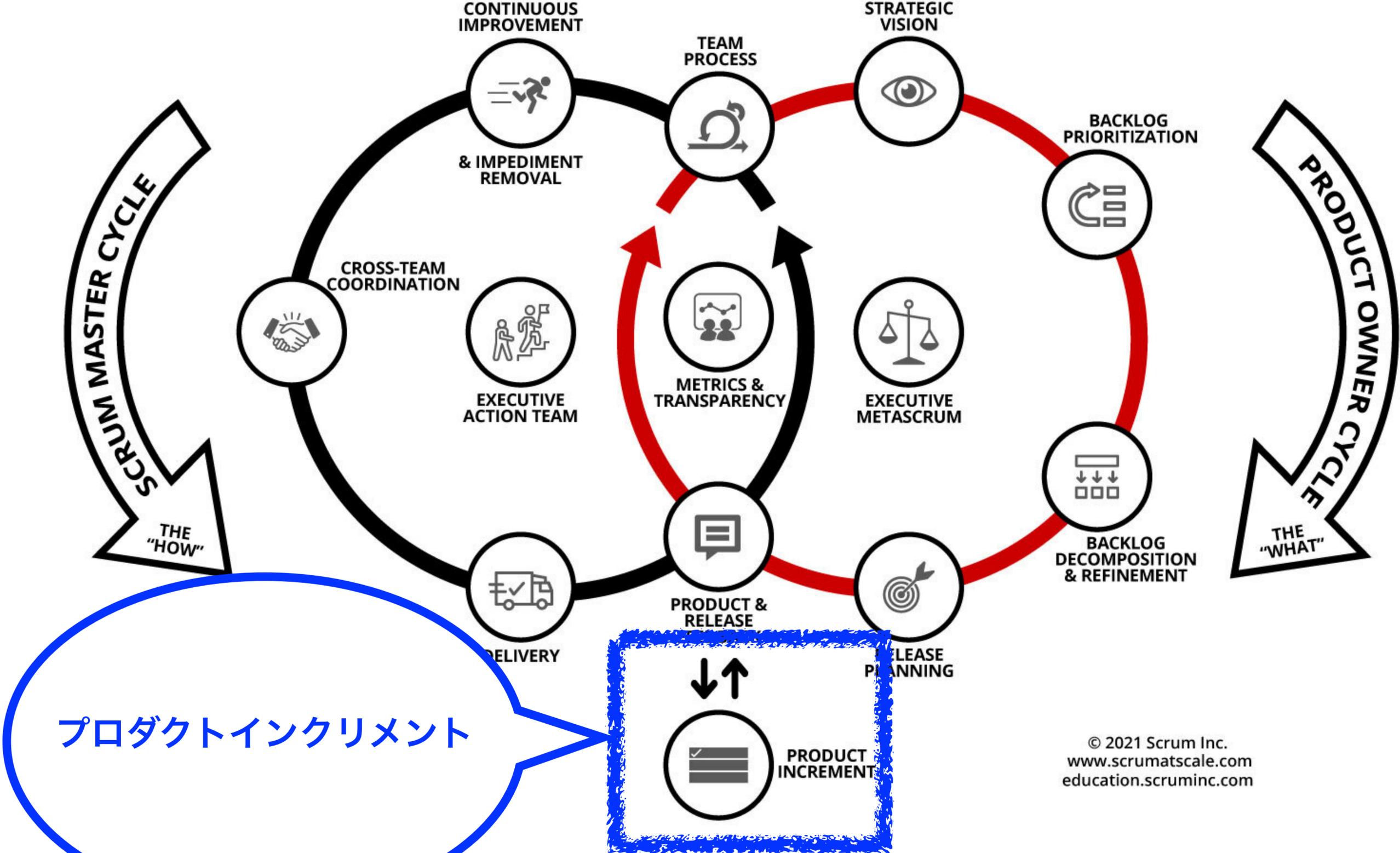


# Scrum@Scale Framework

両サイクル共通

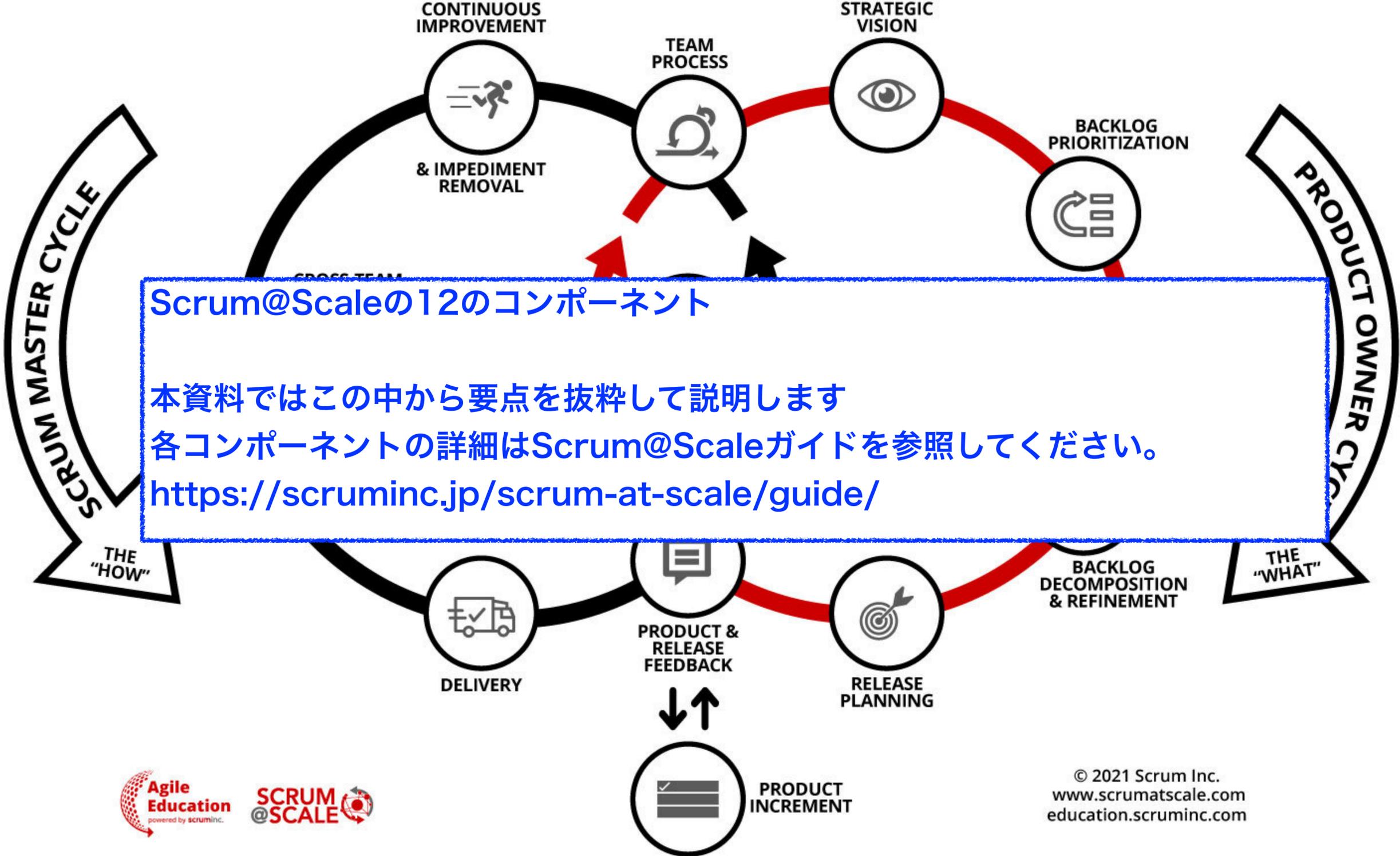


# Scrum@Scale Framework



プロダクトインクリメント

# Scrum@Scale Framework

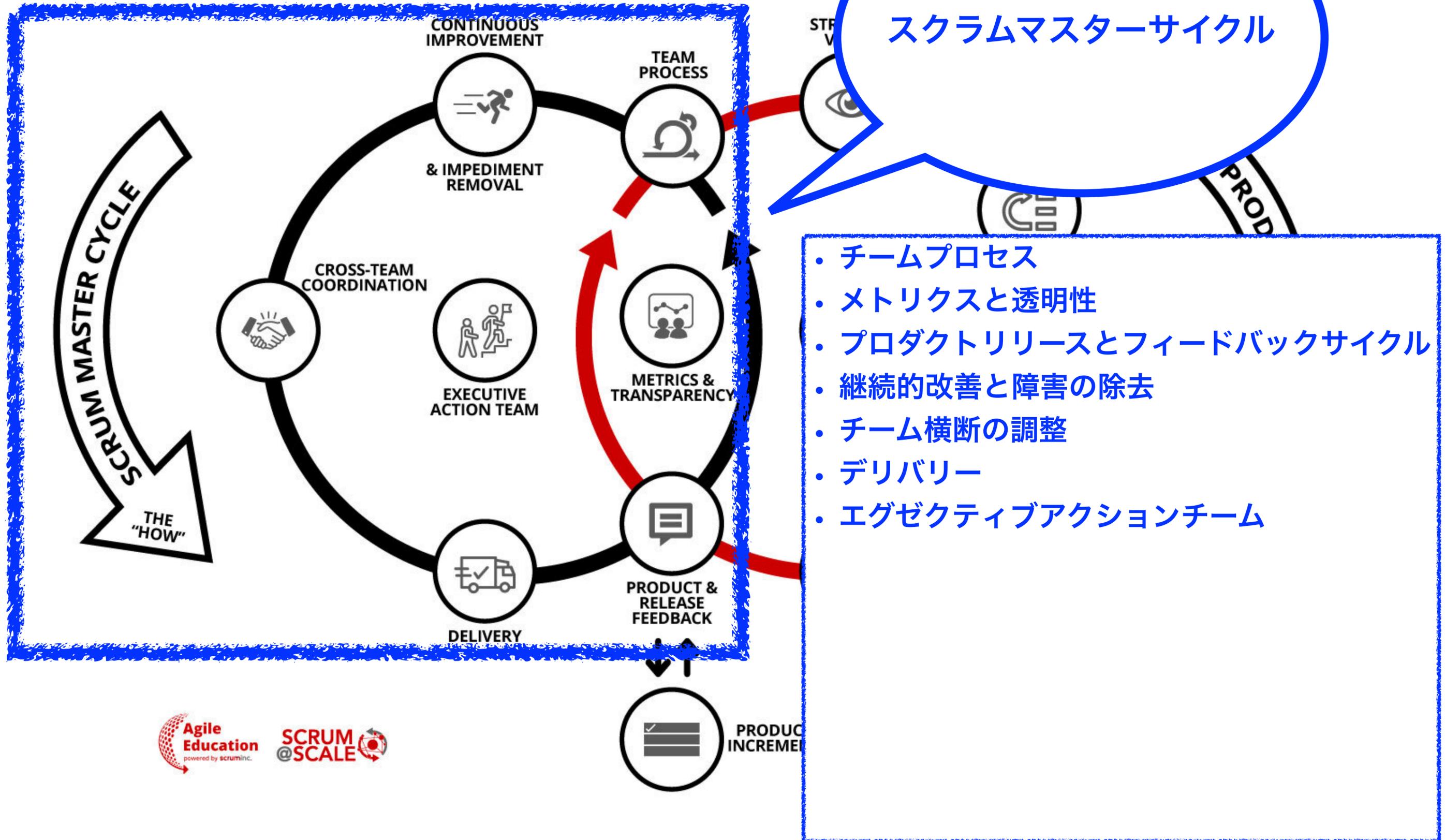


## Scrum@Scaleの12のコンポーネント

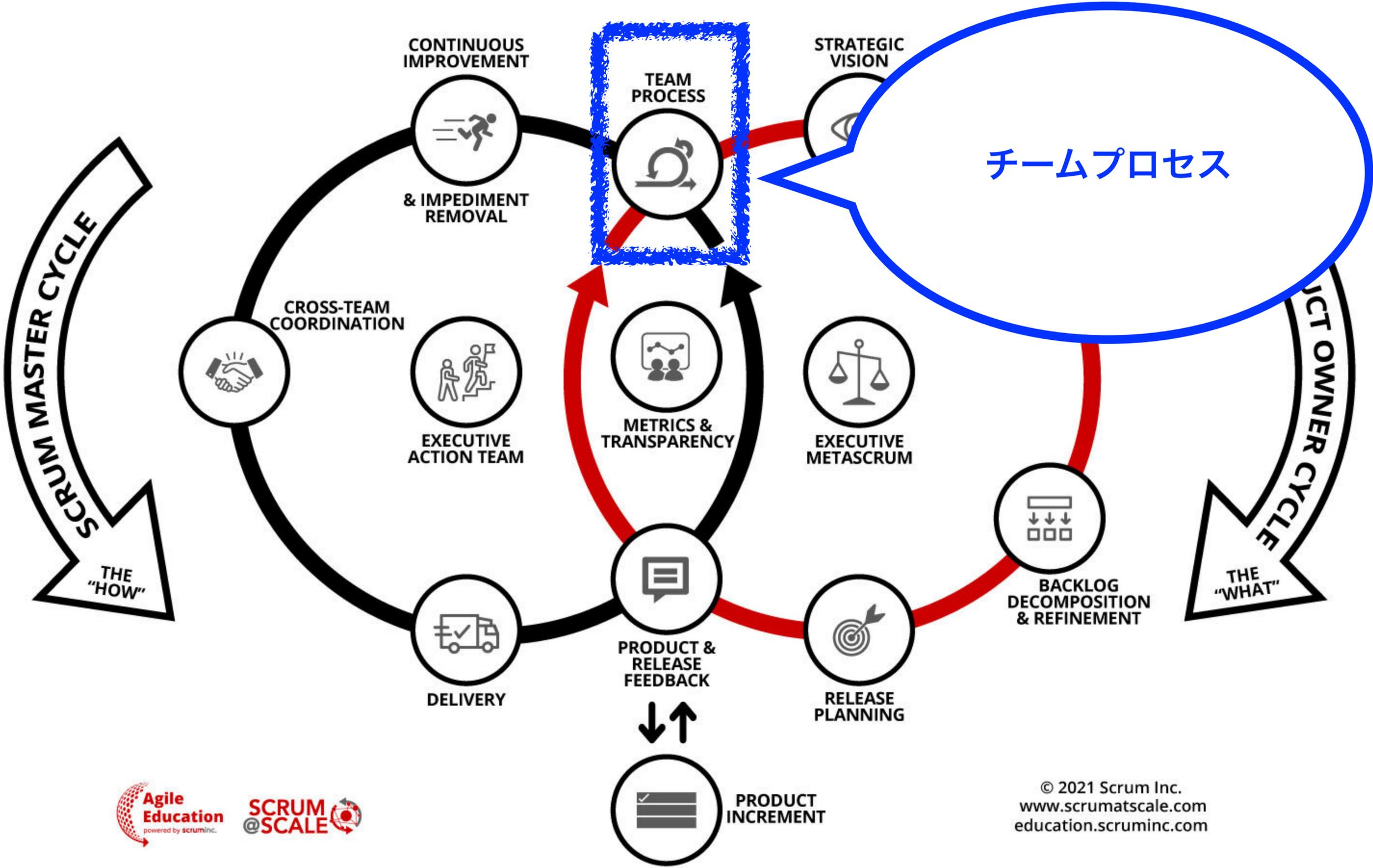
本資料ではこの中から要点を抜粋して説明します  
各コンポーネントの詳細はScrum@Scaleガイドを参照してください。  
<https://scruminc.jp/scrum-at-scale/guide/>

スクラムマスターサイクル

# Scrum@Scale Framework



# Scrum@Scale Framework



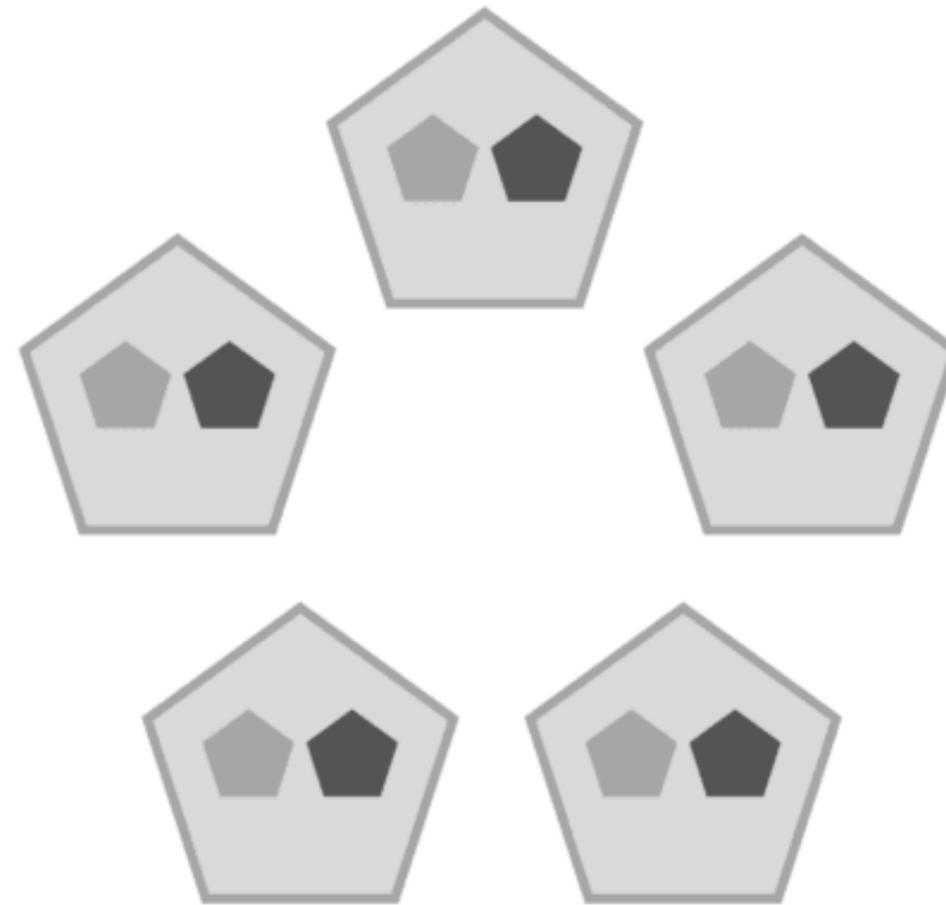
# チームプロセス

- スクラムガイドで規定されているスクラムを行うこと
- スクラムマスターサイクルとプロダクトオーナーサイクルの最初の交差点
- チームプロセスのゴール
  - 完成の定義を満たす作業のフローを最大化する
  - チームのパフォーマンスを時間の経過とともに向上させる
  - 持続可能で、チームをより豊かにする方法で運営する
  - 顧客フィードバックループを加速させる

# スクラムチームのスケール



**A Scrum Team**

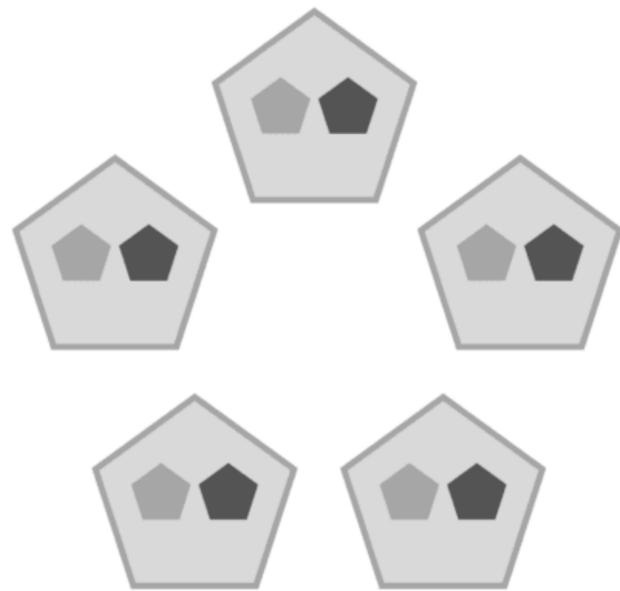


**A Scrum of Scrums Team**

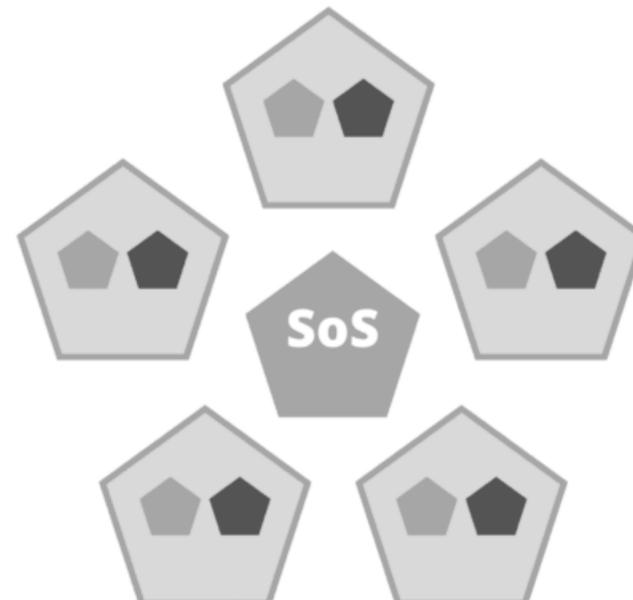
# スクラムチームのスケール(フラクタル)



**A Scrum Team**



**A Scrum of Scrums Team**

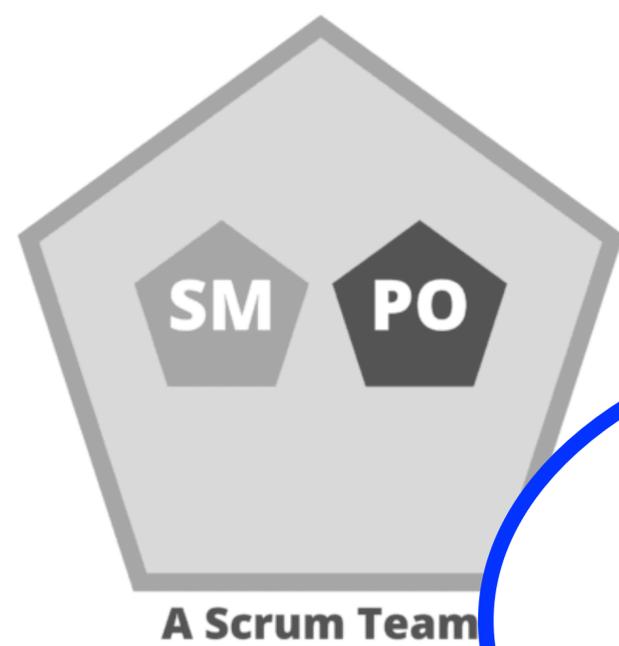


**SoS of 5 Teams**

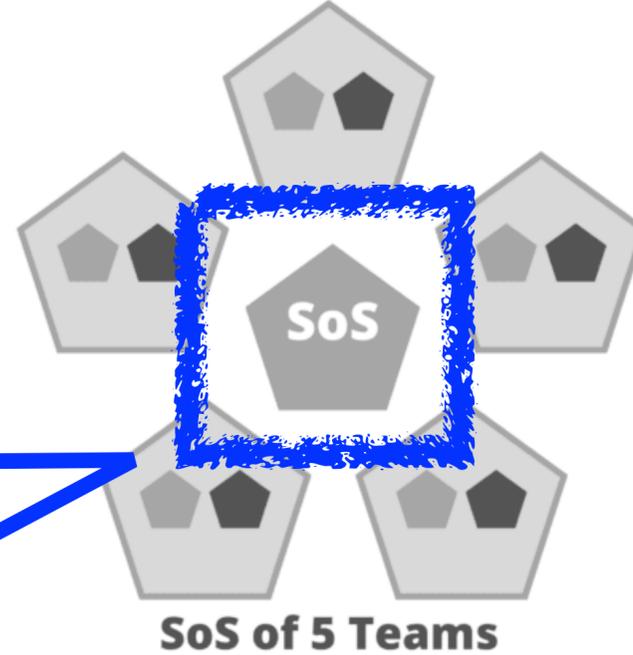
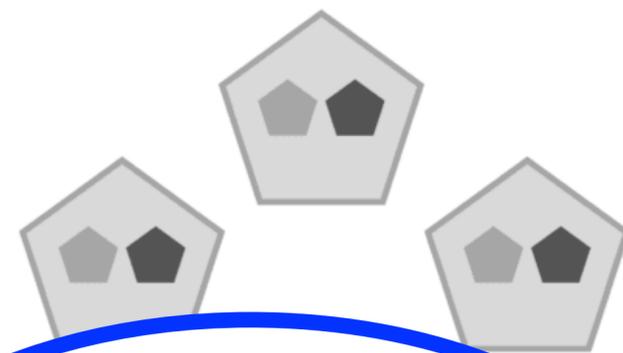


**SoSoS of 25 Teams**

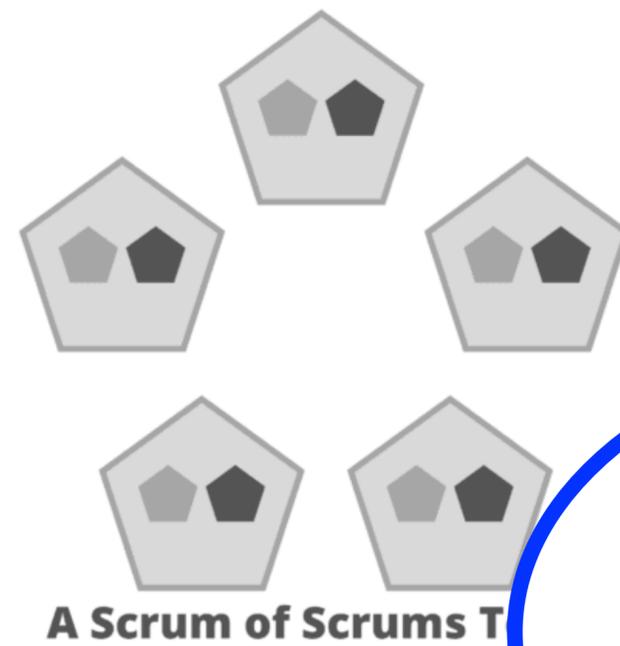
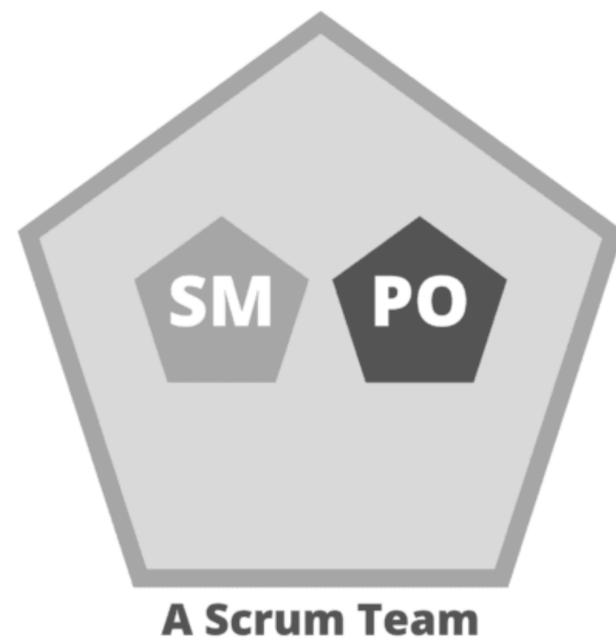
# スクラムチームのスケール(フラクタル)



Scrum of Scrum (SoS)



# スクラムチームのスケール(フラクタル)



Scrum of Scrum of  
Scrum  
(SoSoS)



# Scrum of Scrum (SoS)

- 各スクラムチームを接続するバーチャルなチーム
- 1つのSoSは4~5チームまでが最適な数
- 各チームからの代表者が集まり、スクラムイベントを実施していく
  - スケールドデイリースクラム(SDS)
  - スケールドレトロスペクティブ
- SoSのスクラムマスターはスクラムオブスクラムマスターと呼ぶ

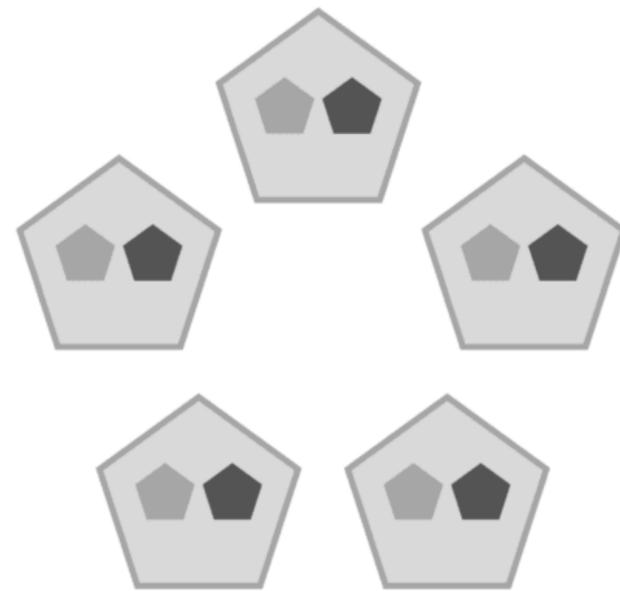
# Scrum of Scrum of Scrum (SoSoS)

- 各SoSチームを接続するバーチャルなチーム
- 1つのSoSoSは4~5SoSチームまでが最適な数
- 各チームからの代表者が集まり、スクラムイベントを実施していく
  - スケールドデイリースクラム (SDS)
  - スケールドレトロスペクティブ

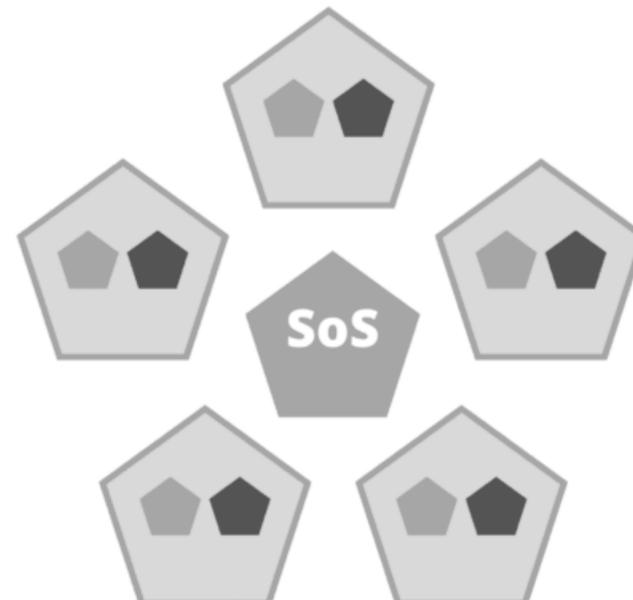
# スクラムチームのスケール(フラクタル)



A Scrum Team



A Scrum of Scrums Team



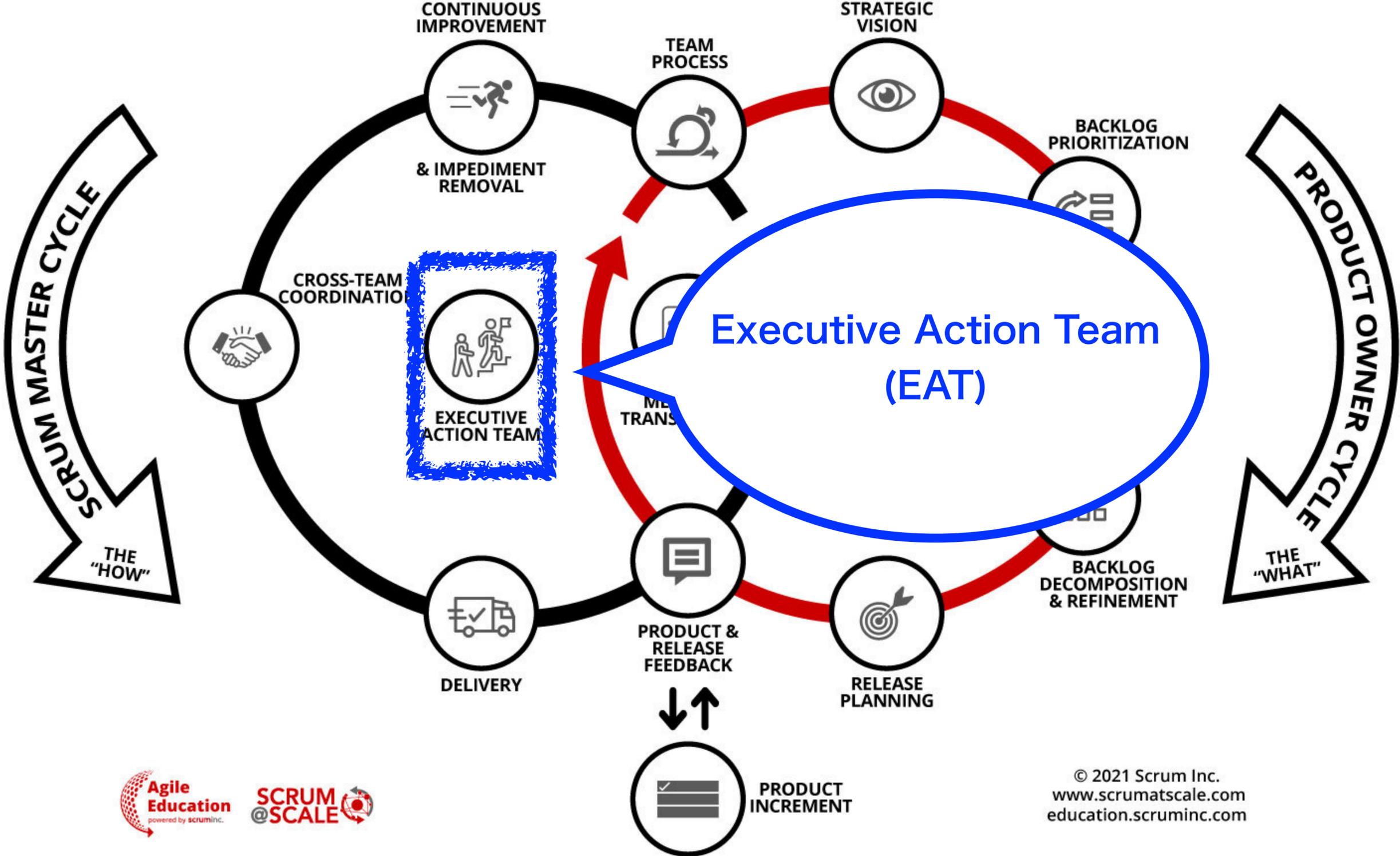
SoS of 5 Teams



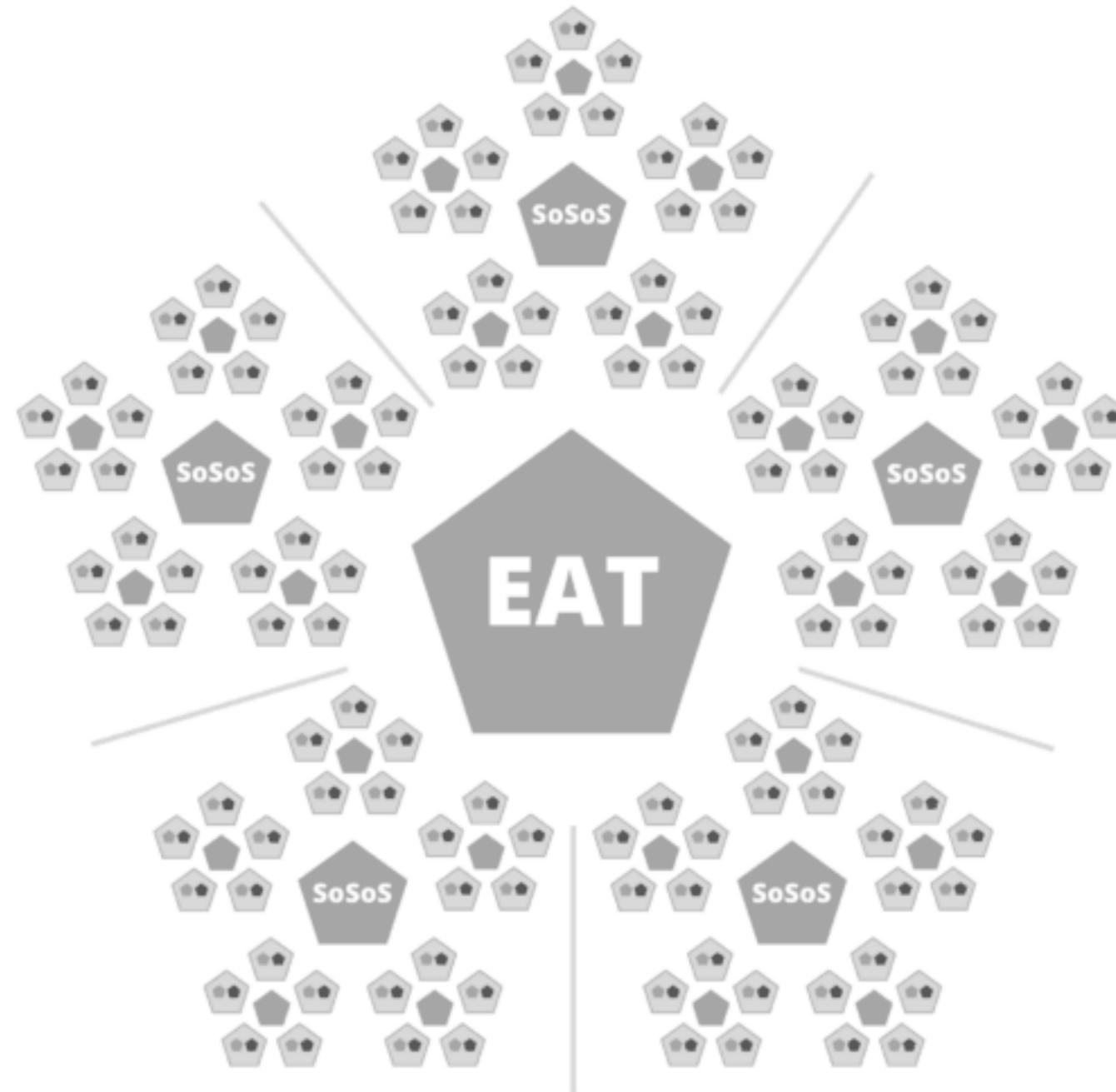
SoSoS of 25 Teams

SoS -> SoSoS -> SoSoSoS と  
理屈の上では無限にスケールさせられる…

# Scrum@Scale Framework



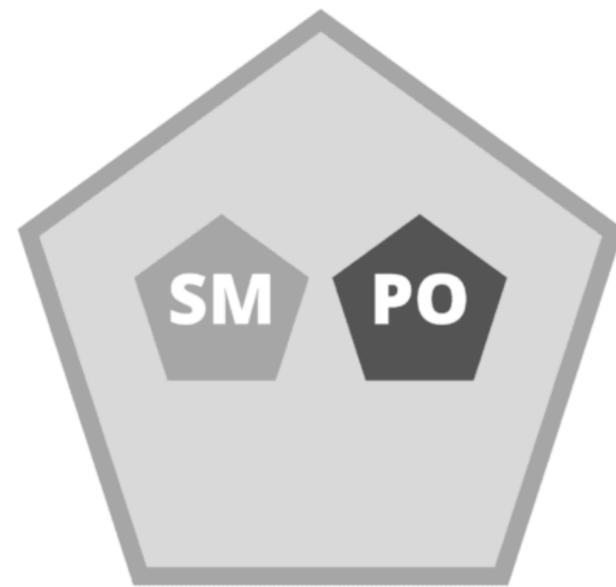
# Executive Action Team(EAT)



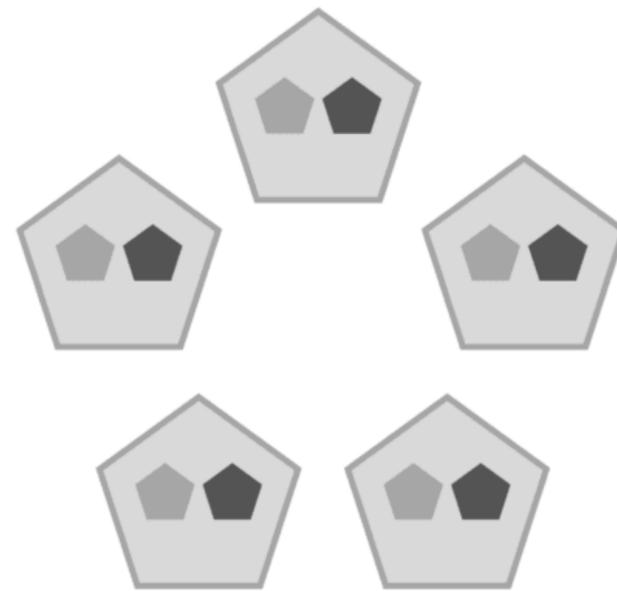
# Executive Action Team(EAT)

- スクラムマスターサイクルにおける最上位の決定機関
- スクラムオブスクラムでは取り除けない障害を取り除くために必要な権限をもったメンバーが参画する必要がある
  - CEO, CTO, 人事の責任者, 決裁権者など
- メンバーの選定条件は、「プロダクトの最終的なすべてをEATで解決できること」
  - EATの集まりで「あ。それは〇〇さんに聞かないと」となれば、〇〇さんをEATに入れるべき
- エグゼクティブなメンバーをスクラムのフローの中に組み込む

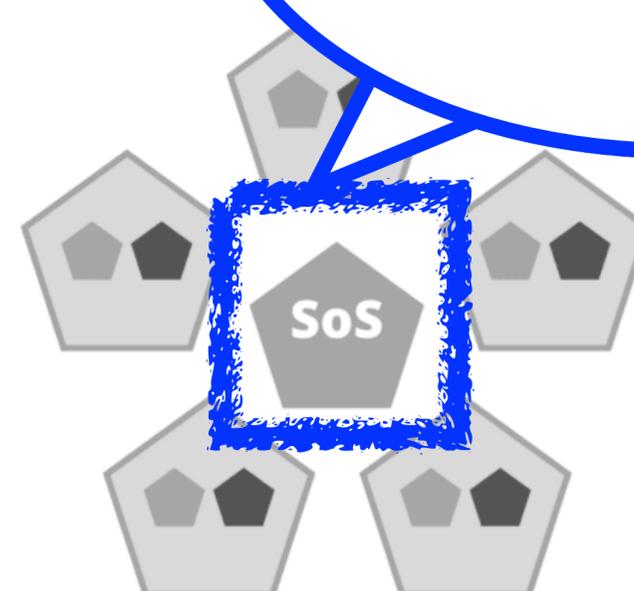
# Executive Action Team(EAT)



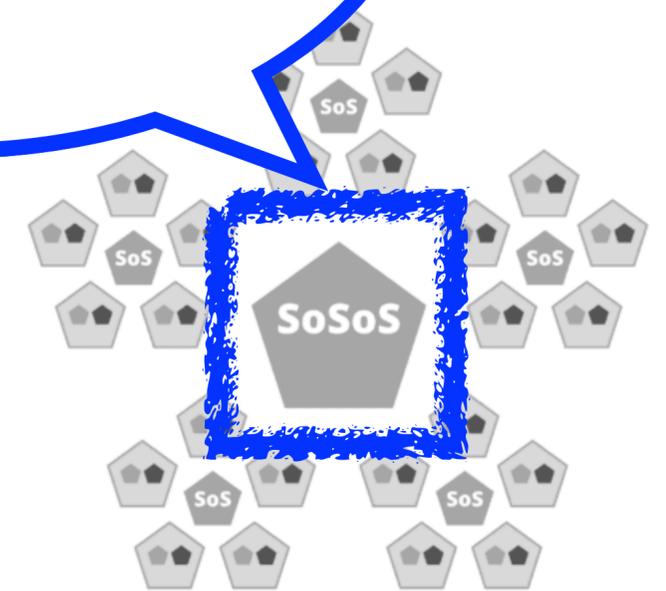
A Scrum Team



A Scrum of Scrums Team



SoS of 5 Teams



SoSoS of 25 Teams

チーム規模によっては  
ここがEATになっていても良い

# スケール化されたスクラムの情報の流れの例

- 10:00 - 10:15 各チームのデイリースクラム
- 10:15 - 10:30 SoSのデイリースクラム
- 10:30 - 10:45 EATのデイリースクラム

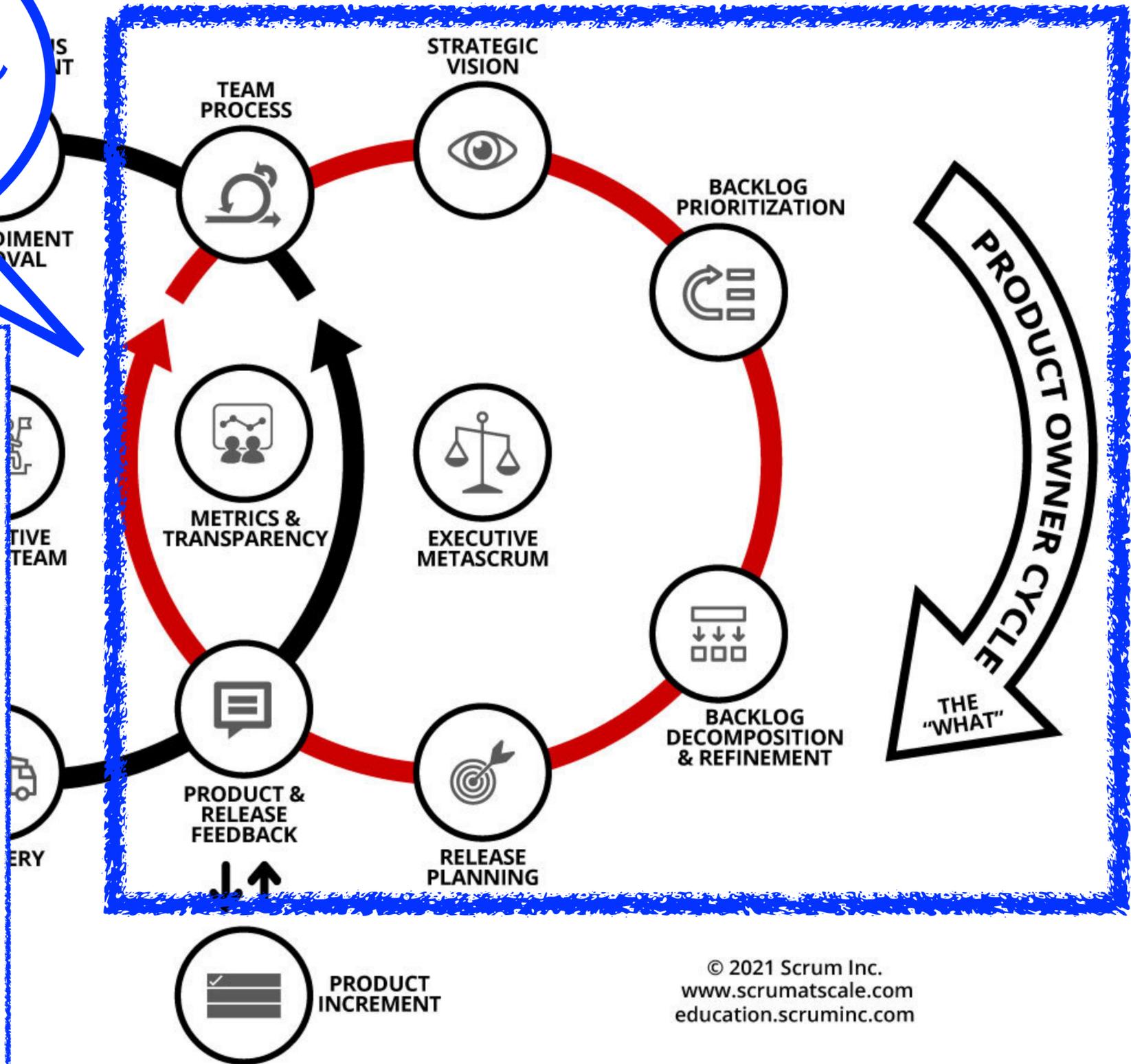
各チームからの課題が、毎日30分という短い時間で最上位の意思決定者まで届く  
ここでEATが課題の解決を先送りしてしまうと、EATが仕事のフローのボトルネックになってしまうので、EATには課題解決を素早く行える権限をもった人がいる必要がある。

プロダクトオーナーサイクル

# Scrum@Scale Framework

プロダクトオーナーサイクル

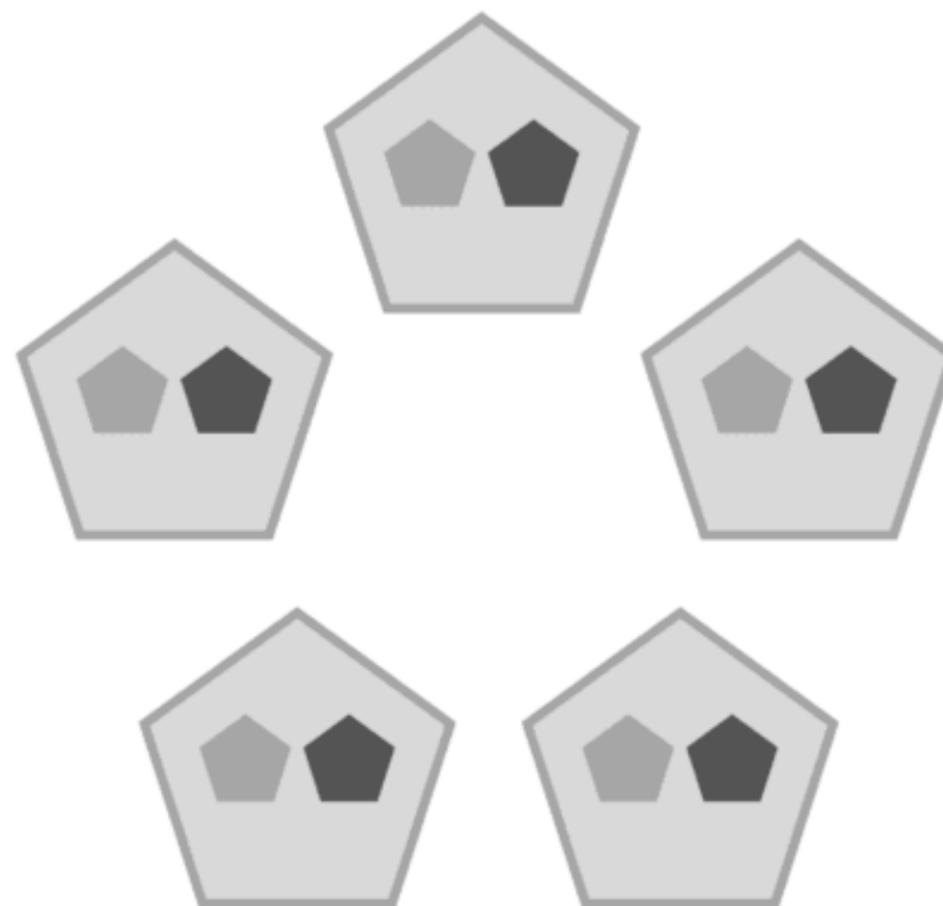
- チームプロセス
- メトリクスと透明性
- プロダクトリリースとフィードバックサイクル
- 戦略的ビジョン
- バックログの優先順位付け
- バックログの分割とリファインメント
- リリースプランニング
- エグゼクティブメタスクラム



# プロダクトオーナーのスケール



**A Scrum Team**

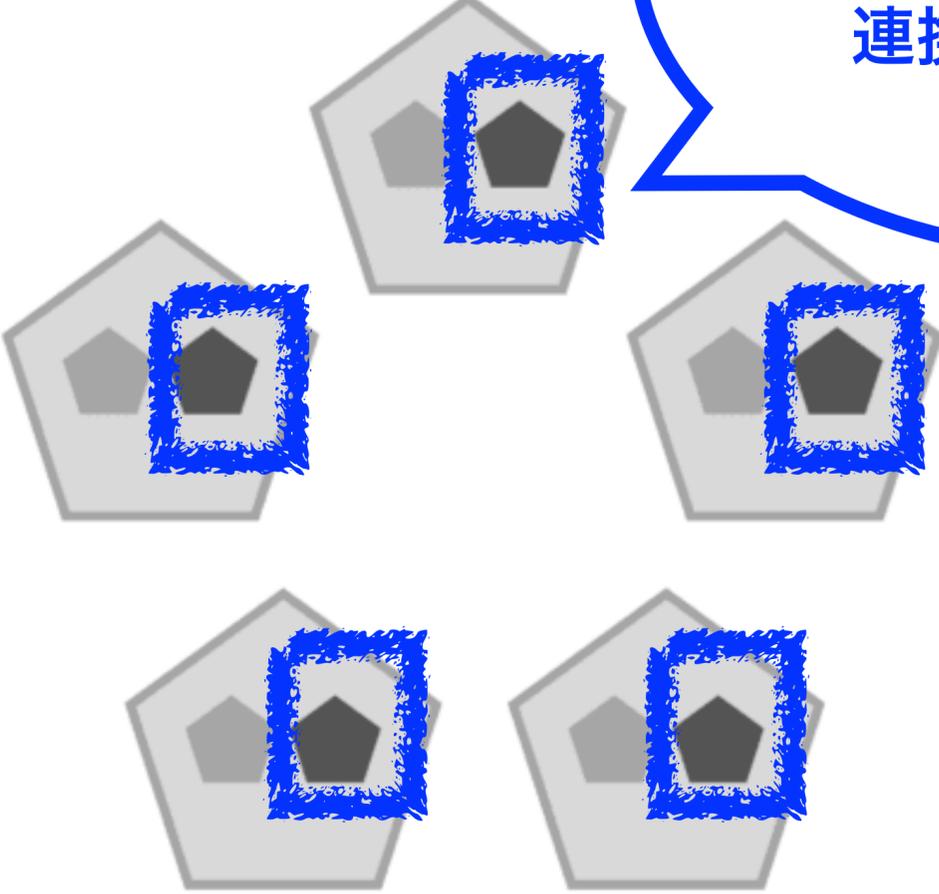


**A Scrum of Scrums Team**

# プロダクトオーナーのスケール



**A Scrum Team**



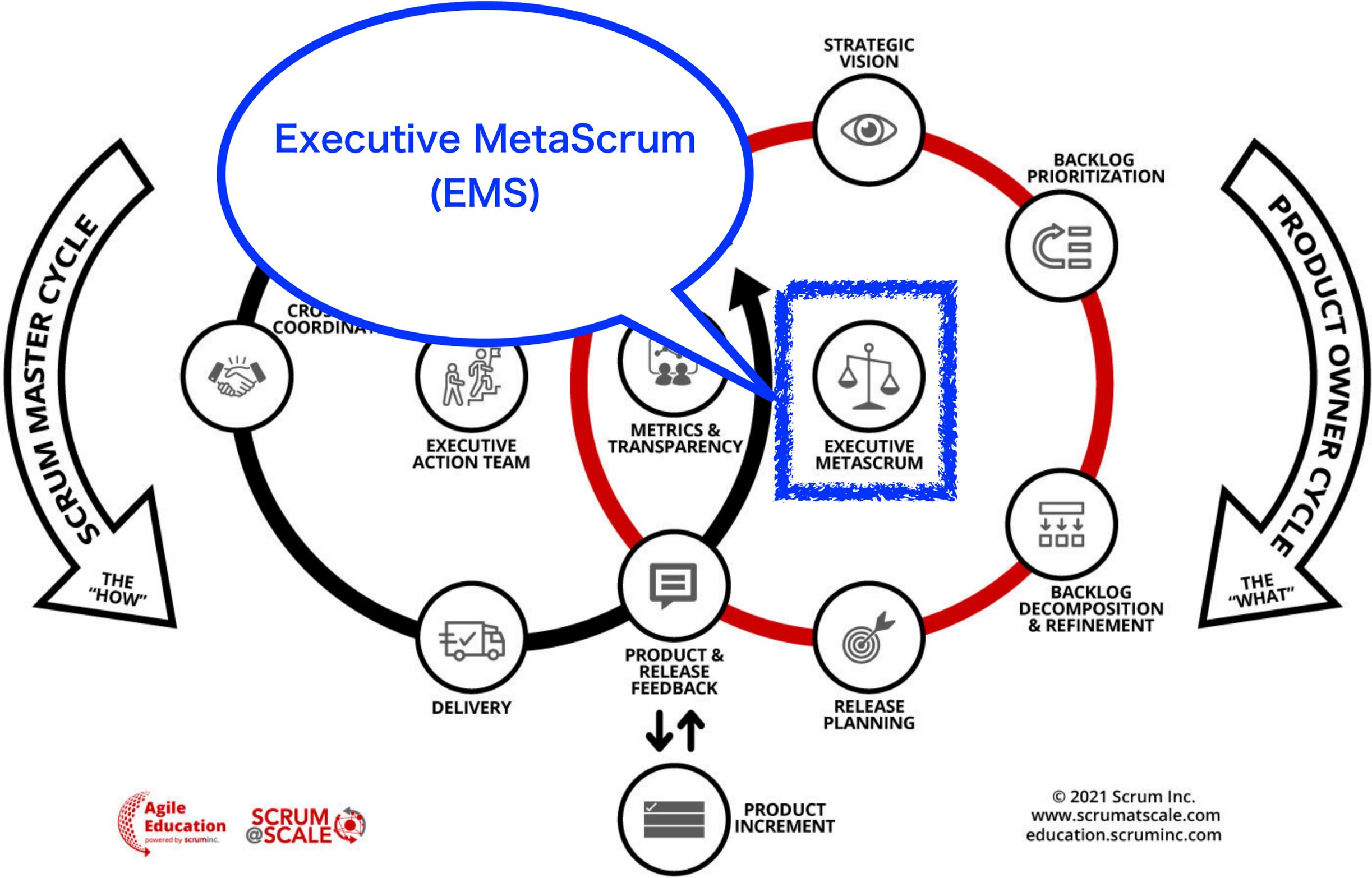
**A Scrum of Scrums Team**

各チームにPOがいる  
つまりPO同士の  
連携が必要になる

# チーフプロダクトオーナー (CPO)

- プロダクトオーナーチームと一緒に優先順位を調整する
- CPOは、この役割を兼任する個別チームのプロダクトオーナーである場合もあれば、この役割に専任した個人の場合もある
- プロダクト全体の戦略的ビジョンを設定する
- 全チームに配布される、優先順位付けされた単一のバックログを作成する
- プロダクトオーナーチームが測定するメトリクスを決定する
- 顧客からのプロダクトのフィードバックを評価し、それに応じて共通バックログを調整する

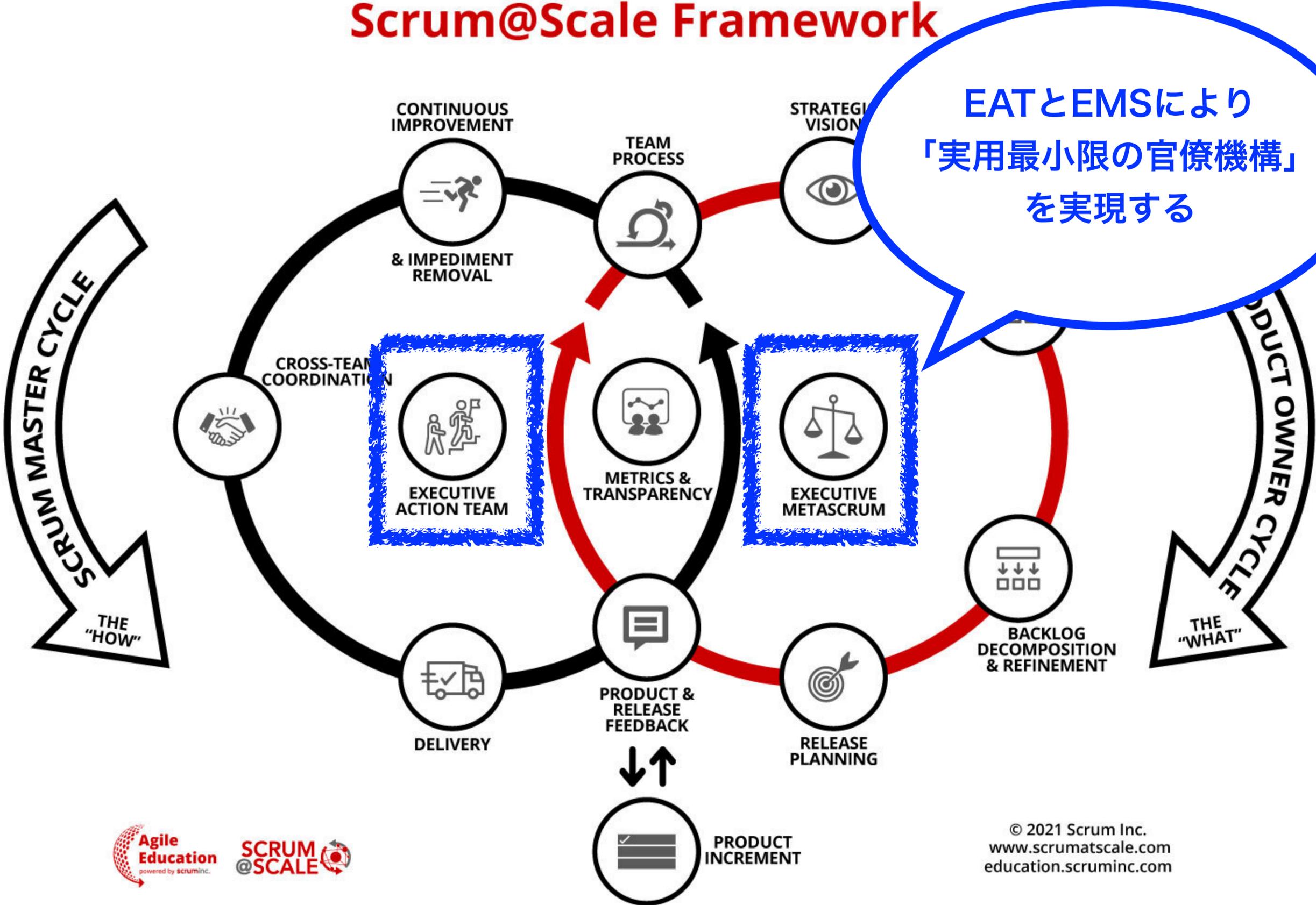
# Scrum@Scale Framework



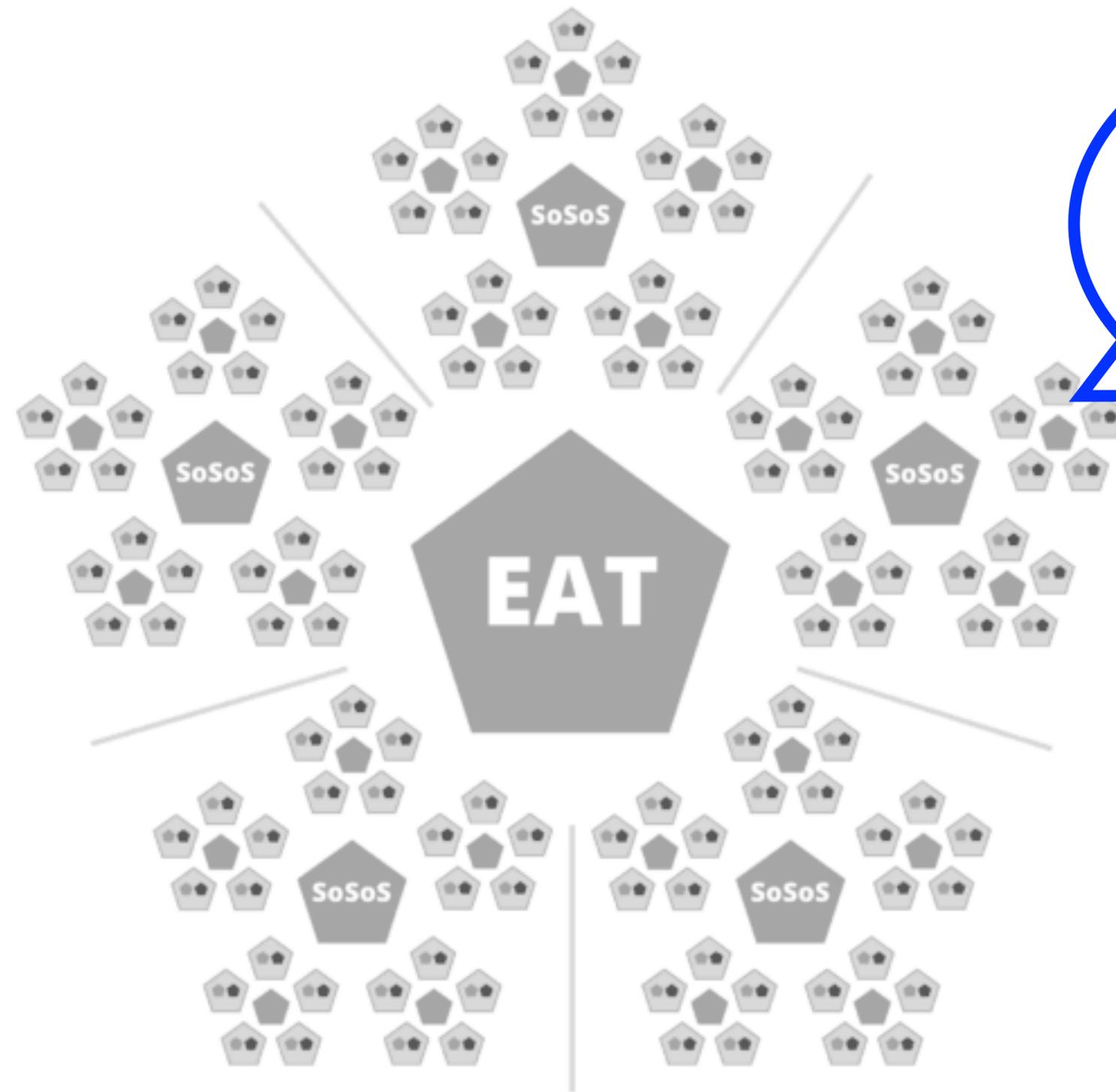
# Executive MetaScrum (EMS)

- アジャイル組織全体に対してプロダクトオーナーの役割を果たすために、CPOは、EMSイベントでエグゼクティブや主要なステークホルダーと会合する。
- プロダクトオーナーサイクルの以下のコンポーネントの要求を満たすために活動する
  - 戦略的ビジョン
  - バックログの優先順位付け
  - バックログの分割とリファインメント
  - リリースプランニング

# Scrum@Scale Framework



# 組織のリファクタリング



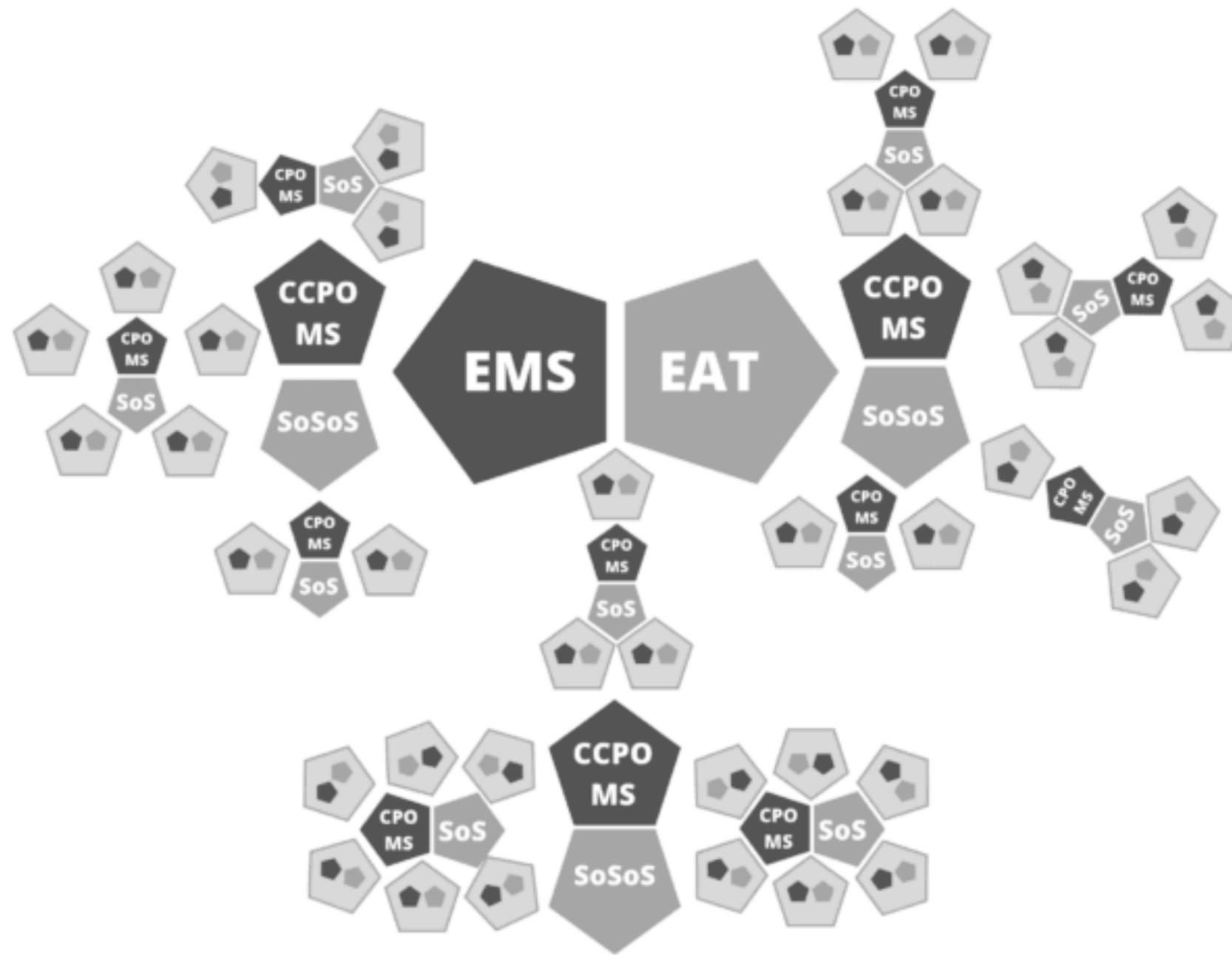
普通は組織はこんなにきれいなフラクタルにはならない

# 組織デザインの例



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

# 組織デザインの例

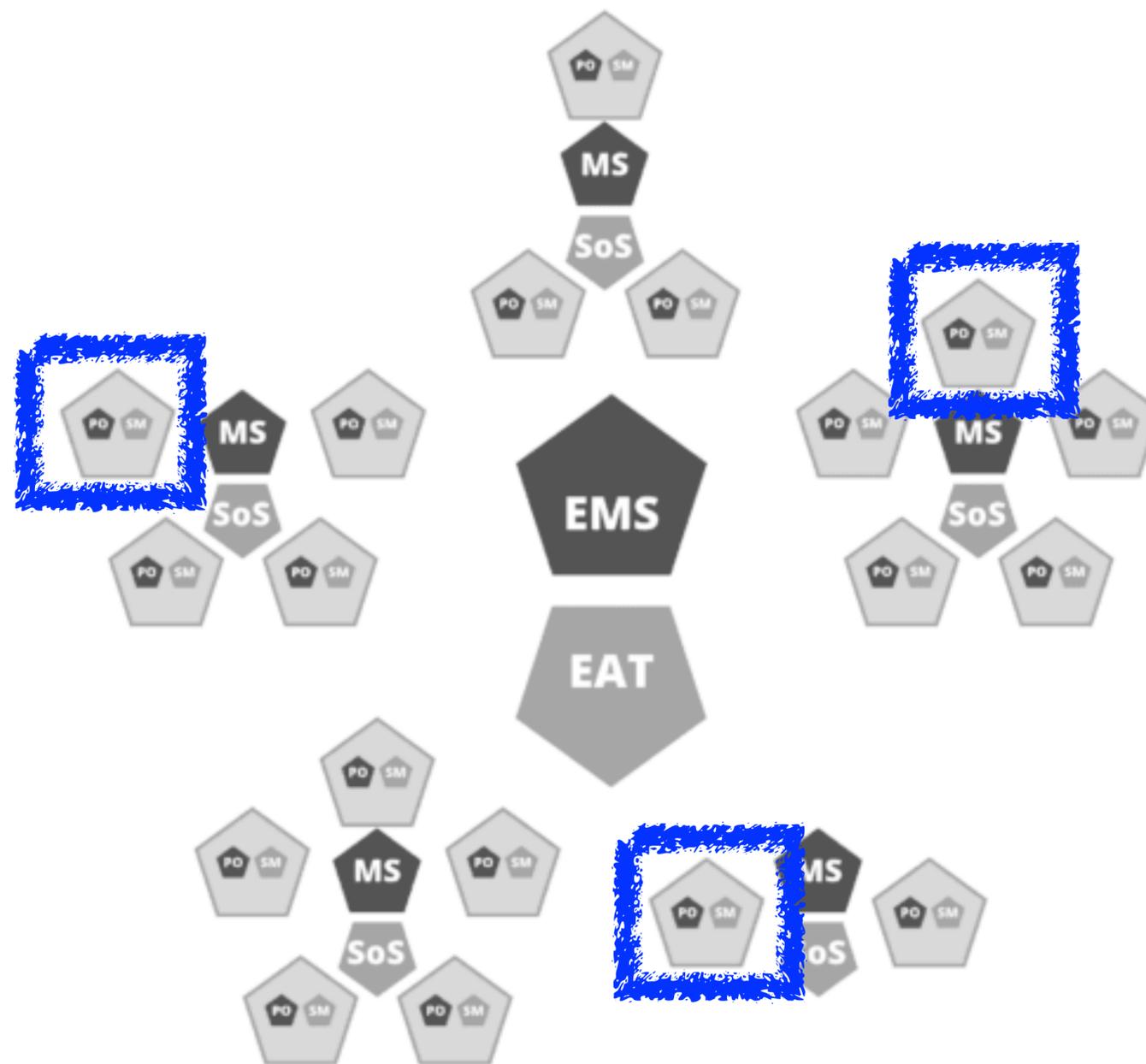


**3 SoS's with 10, 13, & 15 Teams**

# 組織のリファクタリング

- 事業や、プロダクトの状況に応じて組織は適宜リファクタリングされたい
- 人を動かすのはコストがかかる
  - 個人の異動
  - チームの構成メンバーの大幅な変更
  - ↑ これらはやりづらい

# 組織デザインの例

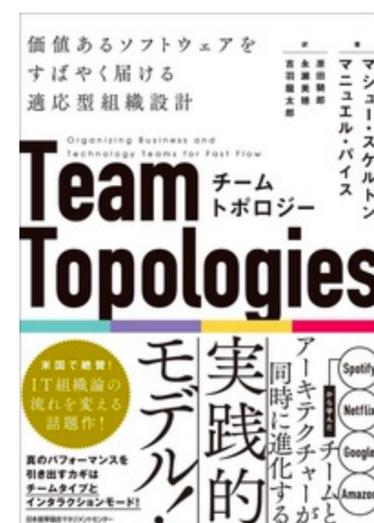


単一の  
「普通のスクラムチーム」  
がSoSやMSによってつながっ  
ている

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

# 組織のリファクタリング

- Scrum@ScaleではSoSやMSといったバーチャルなチームによって各チームが接続する
- 単一のスクラムチームの構成メンバーは動かさなくて良い
- SoSなどの組み合わせを調整することで、各チームの接点を調整していく
- 公式ガイドでは、どういう考え方でチームの組み方を検討するといったかなどは書かれていないので『チームトポロジー』などを参考にすると良いと思います（Scrum@Scaleと相性が良いという所感



# QRC チーム トポロジー

Based on Team Topologies, QRC by Henny Portman, May 2020

**コンウェイの法則:** “システムを設計する組織は... その組織のコミュニケーション構造をコピーした構造の設計を生み出すことになる”

**チームファーストの取り組み:** ソフトウェアデリバリーの効果を高めるためのチームから始める。勘案すべき観点は複数ある: チームサイズ、存続期間、関係性、認知負荷

**組織的センシング(Organizational Sensing):** 組織構造を適応・発展させることを目論む

**チームのスケール:** 組織構造のグルーピングはダンバー数に従う。まずは5-8名から始める。次に15名程度、それから50名、150名、500名、とする

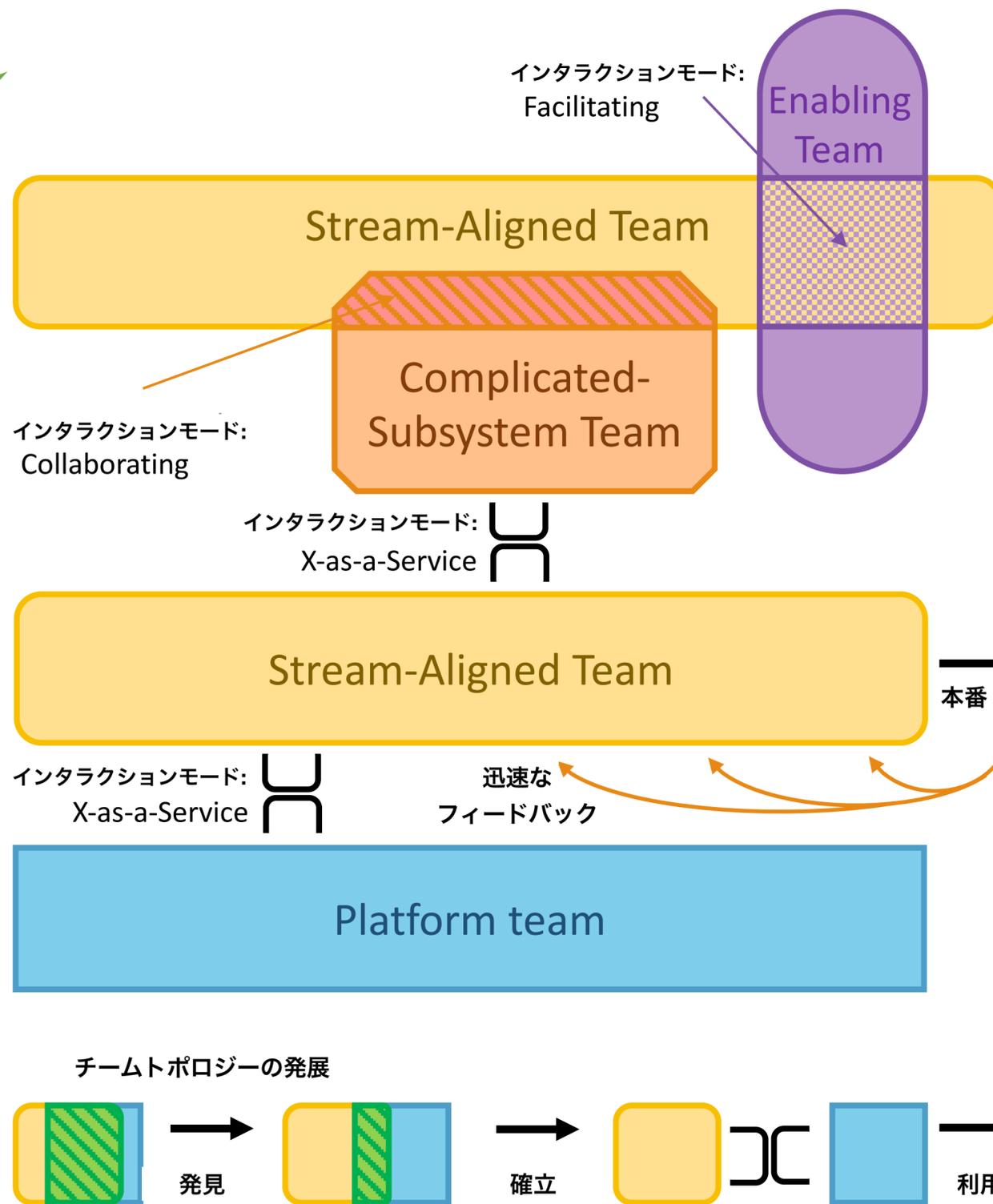
**ブルックスの法則:** “人員を追加しても、チームのキャパシティはすぐに増加するわけではない”

**認知負荷:** “ワーキングメモリに割り当てられるメンタルエフォートの総量”。チームの責務はチームの認知負荷の上限が制約となる。

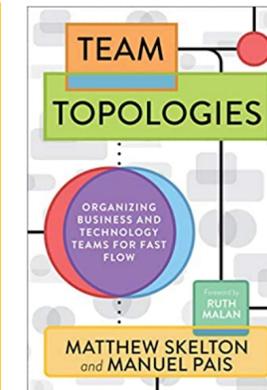
・ **内在性(intrinsic)認知負荷** — 問題領域の本質に関連するタスクにまつわる負荷

・ **外在性(extraneous)認知負荷** — 取り組むタスクの環境にまつわる負荷

・ **適切な(germane)認知負荷** — 学習やハイパフォーマンスを発揮するために取りわけ注意力を要するタスクにまつわる負荷



**ストリームアラインドチーム:** ビジネス的変更の主要な「流れ(Flow)」に向けて編成されるチーム。職能横断スキルの融合体であるため、他チームの作業を待つことなく意義のあるインクリメントをデリバリーできる能力が備わっている



**プラットフォームチーム:** ストリームアラインドチームのデリバリーをサポートする基盤となるプラットフォームを仕事の領域とするチーム。プラットフォームは複雑な技術をシンプルにすることで、それを利用するチームの認知負荷を軽減する

**イネイブリングチーム:** 移行や学習期間の一環として、他のチームがソフトウェアを採用することや変更することをアシストするチーム

**コンプリケイティッド・サブシステムチーム:** 複雑すぎて通常のストリームアラインドチームやプラットフォームチームでは対処できないサブシステムに特化して取り組むチーム。本当に必要な場合、それが最善である場合にのみ編成される

基礎となるチームトポロジー(4種類)の間のインタラクションモード:

- コラボレーション:** 他のチームと密接に連携する
- X-as-a Service:** 最小限のコラボレーションで「何か」を提供/利用する
- ファシリテティング:** 妨害物を取り除くために他のチームを援助する(援助される)

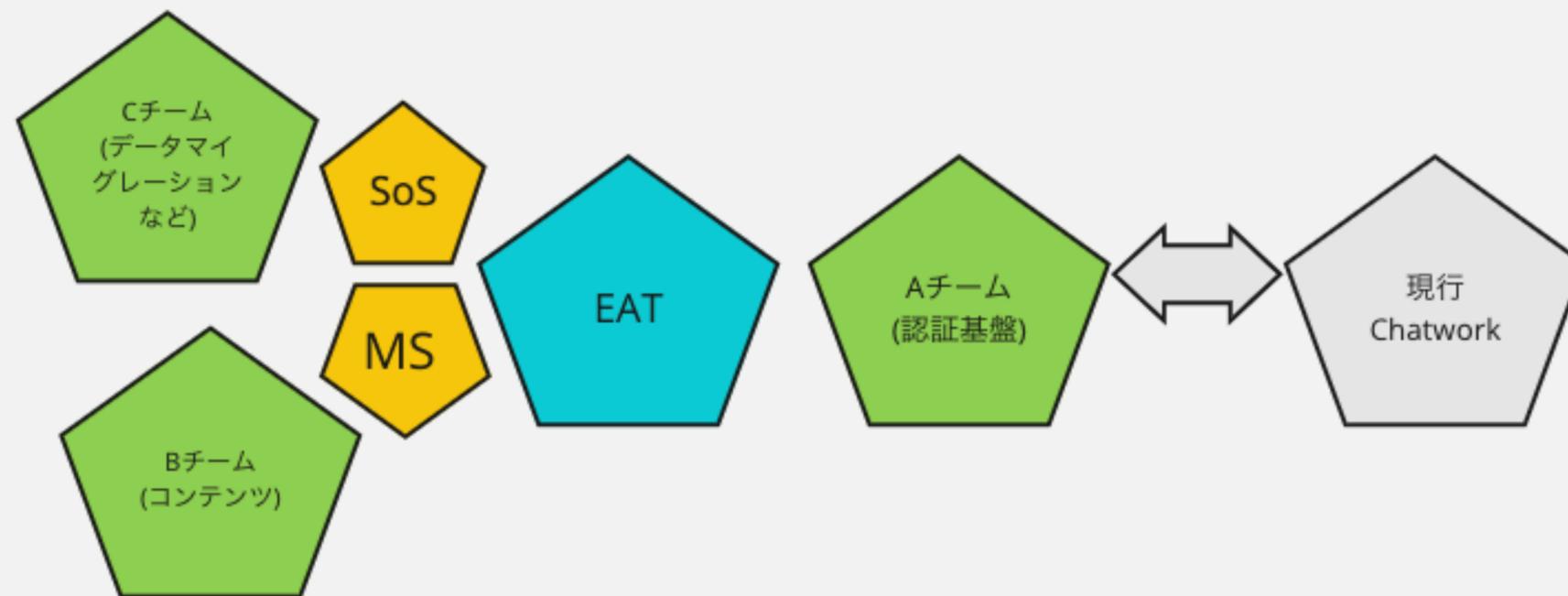
# Chatworkにおける実例

# 前提

- **アーキテクチャ刷新プロジェクトでScrum@Scaleを採用**
  - **刷新後はこの新しい体制がChatworkのコアになるので既存組織も徐々にScrum@Scaleに移行していく算段**
- **去年の4月に立ち上げ**
- **アーキテクチャ刷新プロジェクトの序盤なので、プロダクトバックログは技術的課題が多い**

# 現在の体制

# 現在の体制



## 各チームの構成

### 開発チーム×3

- プロダクトオーナー
- スクラムマスター
- 開発メンバー

### SoS (Scrum of Scrum)

- スクラムオブスクラムマスター
- 各チームからの代表者

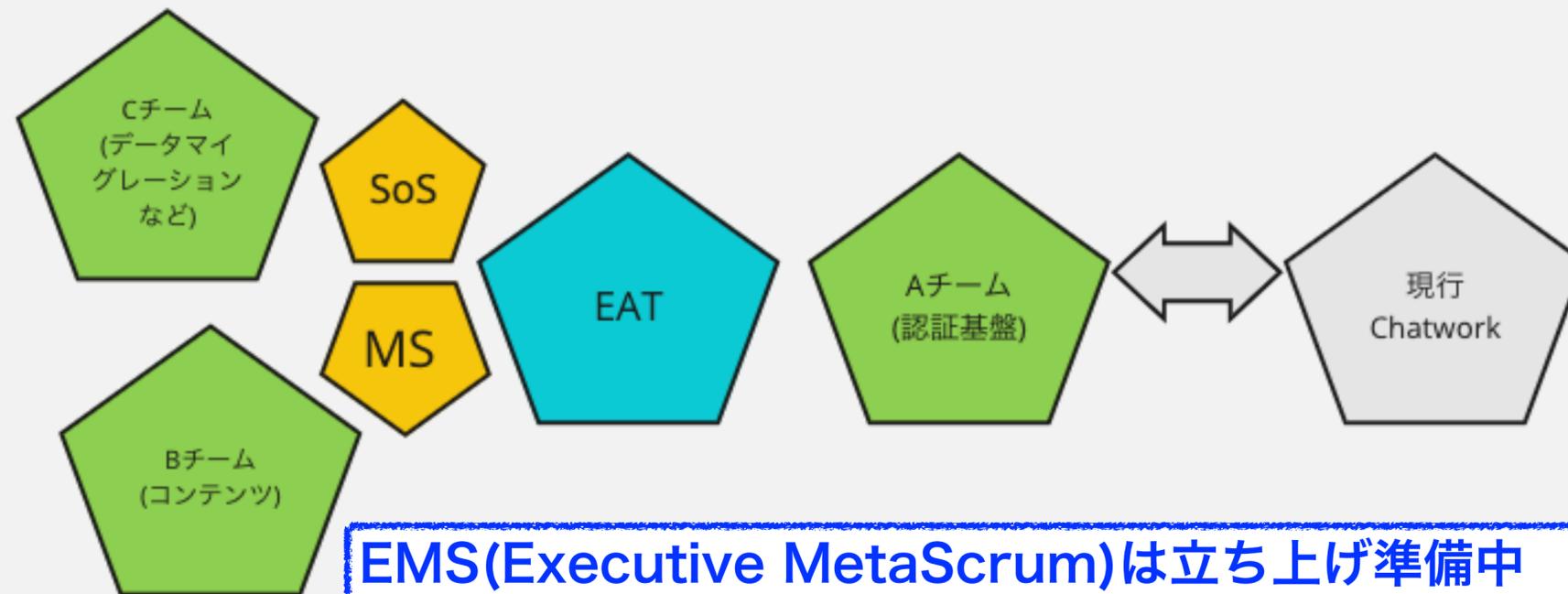
### MS (MetaScrum)

- 各チームのPO

### EAT (Executive Action Team)

- CTO
- DevHR (開発組織の人事責任者)
- 各チームのPO
- 各チームからの代表者

# 現在の体制



EMS(Executive MetaScrum)は立ち上げ準備中

## 各チームの構成

### 開発チーム×3

- プロダクトオーナー
- スクラムマスター
- 開発メンバー

### SoS (Scrum of Scrum)

- スクラムオブスクラムマスター
- 各チームからの代表者

### MS (MetaScrum)

- 各チームのPO

### EAT (Executive Action Team)

- CTO
- DevHR (開発組織の人事責任者)
- 各チームのPO
- 各チームからの代表者

# 現在の体制

	月	火	水	木	金
9時			EATミーティング		
10時	各チームデイリースクラム SoSデイリースクラム MetaScrumデイリースクラム	各チームデイリースクラム SoSデイリースクラム MetaScrumデイリースクラム	各チームデイリースクラム SoSデイリースクラム MetaScrumデイリースクラム	各チームデイリースクラム SoSデイリースクラム MetaScrumデイリースクラム	各チームデイリースクラム SoSデイリースクラム MetaScrumデイリースクラム
11時		Aチーム スプリントレビュー	Bチーム スプリントレビュー		
12時	Aチーム バックログリファインメント	Cチーム スプリントレビュー	Aチーム プラン ニング Cチーム プラン ニング Bチー ム レトロ スペク ティブ		Bチーム バックログリファインメント
13時	昼休み	昼休み	昼休み	昼休み	昼休み
14時	Cチーム バックログリファインメント	Aチーム レトロスペクティブ Cチーム レトロスペクティブ	Bチーム プランニング		
15時					
16時		SoSレトロスペクティブ			

- 各チームのスクラムイベントの実施は、各チームに裁量がある
- ただし、情報の流れをスムーズにするため、以下の制約を設けている
  - デイリースクラムは一気通貫に流れること
  - 1週間のタイムボックス(水曜-火曜)はあわせること

# 現在の体制

	月	火	水	木	金
9時			EATミーティング		
10時	各チームデイリースクラム SoSデイリースクラム MetaScrumデイリースクラム	各チームデイリースクラム SoSデイリースクラム MetaScrumデイリースクラム	各チームデイリースクラム SoSデイリースクラム MetaScrumデイリースクラム	各チ So MetaSc	
11時		Aチーム スプリントレビュー	Bチーム スプリントレビュー		
12時	Aチーム バックログリファインメント	Cチーム スプリントレビュー	Aチーム プラン ニング Cチーム プラン ニング Bチー ム レトロ スペク ティブ		
13時	昼休み	昼休み	昼休み	昼休み	昼休み
14時	Cチーム バックログリファインメント	Aチーム レトロスペクティブ Cチーム レトロスペクティブ	Bチーム プランニング		
15時					
16時		SoSレトロスペクティブ			

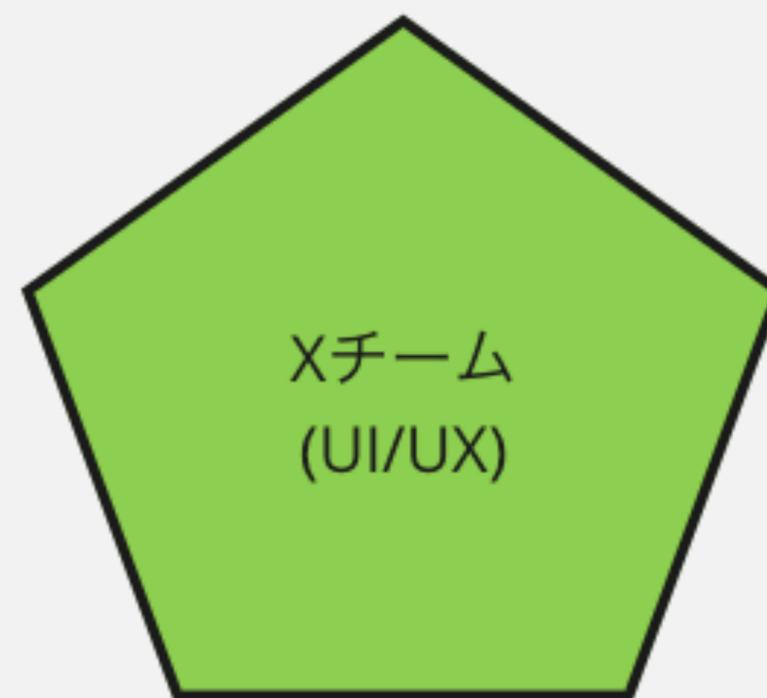
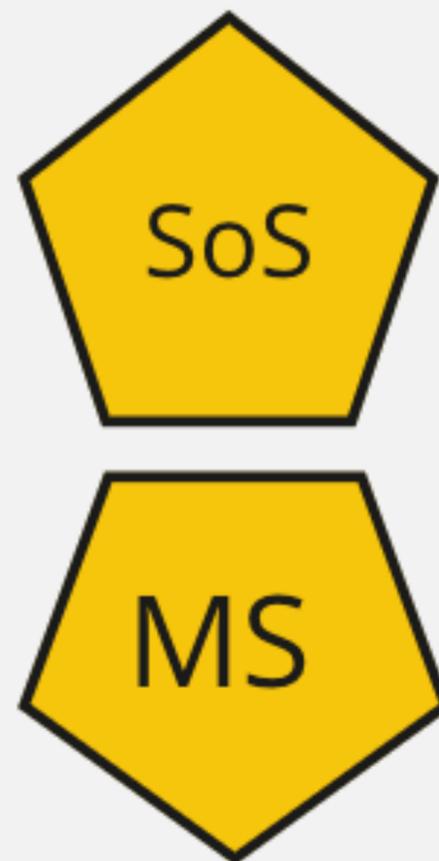
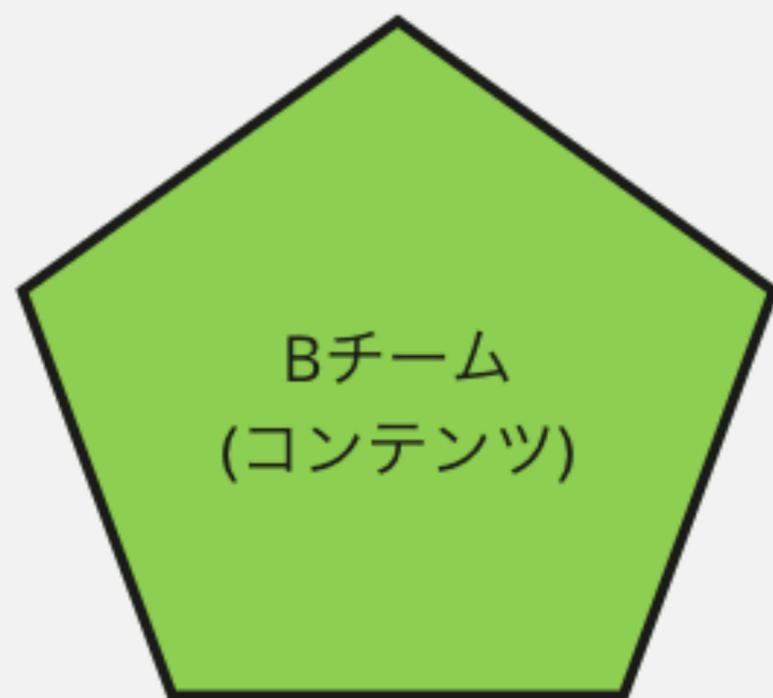
A, Cチームは火曜にレビュー, ふり  
 かえり, 水曜プランニング  
 Bチームは水曜にレビューからプランニング  
 まで詰め込んでいる  
 このようにチームによって制約の範囲内で  
 アレンジがある

イベントの実施は、各チー  
 ーズにするため、以  
 貫に流れること  
 (水曜-火曜)はあわ

**これまでの推移**

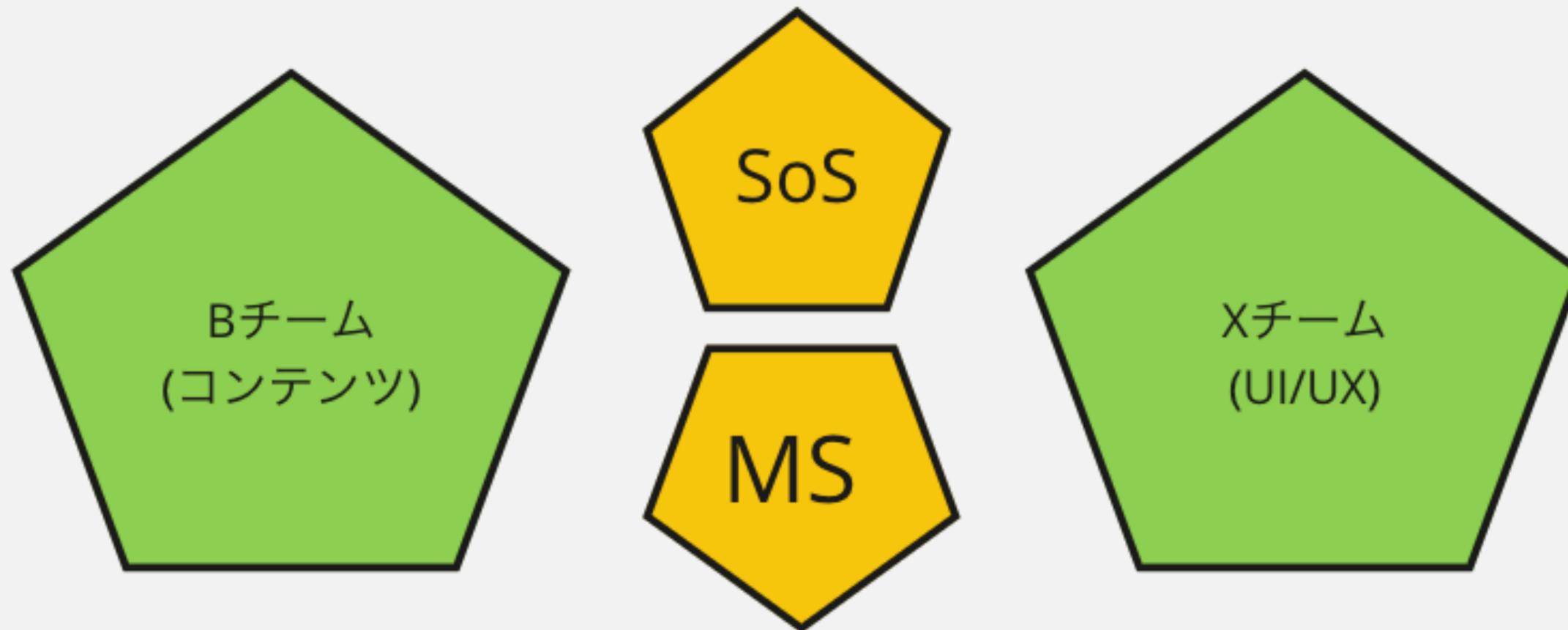
# 推移

2021年4-6月



# 推移

2021年4-6月



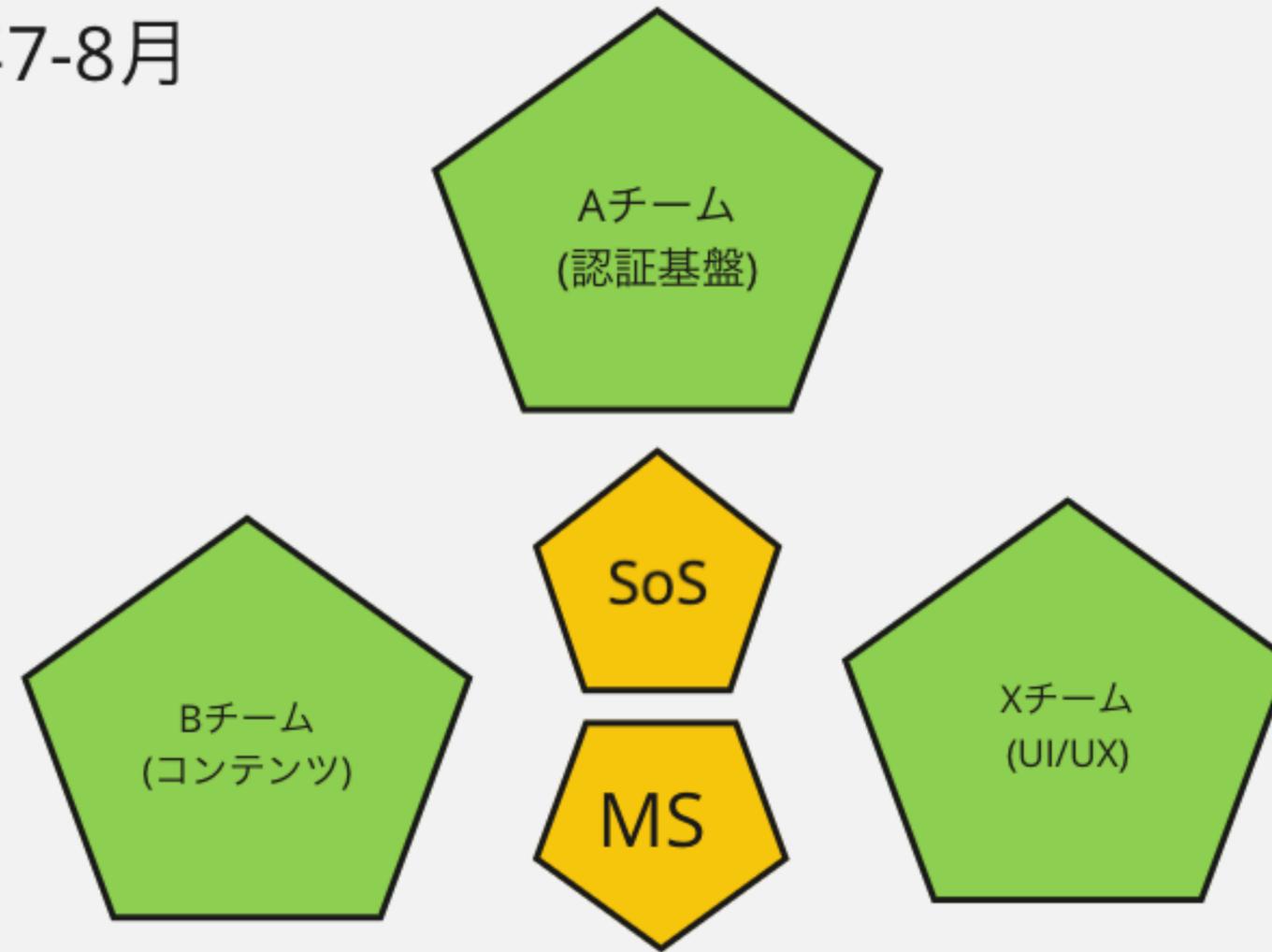
2チームからスタート

外部から2名のプロコーチに入ってもらい、それぞれのスクラムチームを立ち上げた。

5月に粕谷が入社。

# 推移

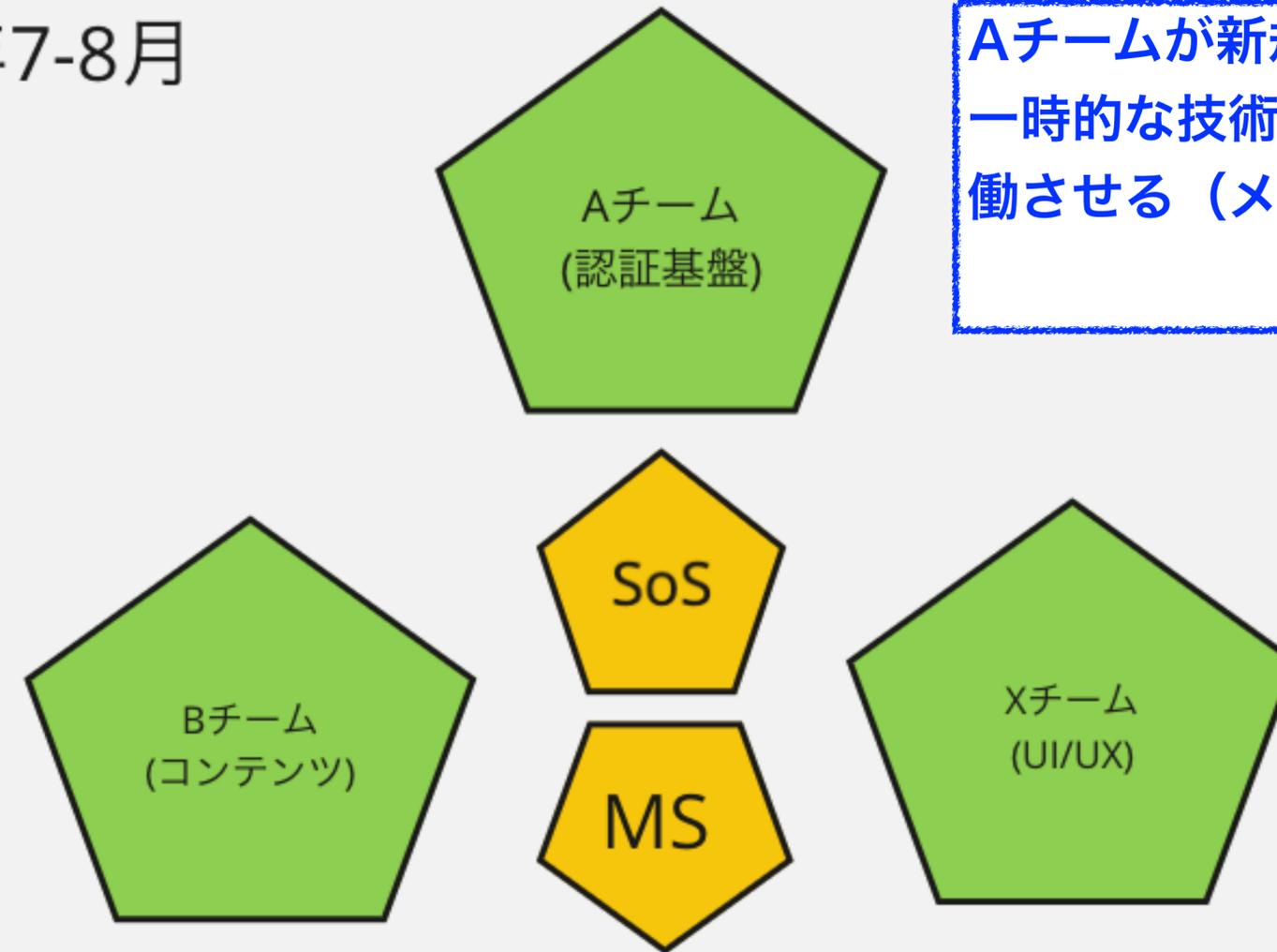
2021年7-8月



とある要素技術についての  
タスクフォース

# 推移

2021年7-8月



Aチームが新規で立ち上がる  
一時的な技術調査のためのタスクフォースを稼働させる (メンバーは各チームの一部が兼務)

とある要素技術についての  
タスクフォース

# 推移

2021年7-8月

コンプリケイテッド・サブシステムチーム

Aチーム  
(認証基盤)

Aチームが新規で立ち上がる  
一時的な技術調査のためのタスクフォースを稼働させる（メンバーは各チームの一部が兼務）

Bチーム  
(コンテンツ)

SoS

Xチーム  
(UI/UX)

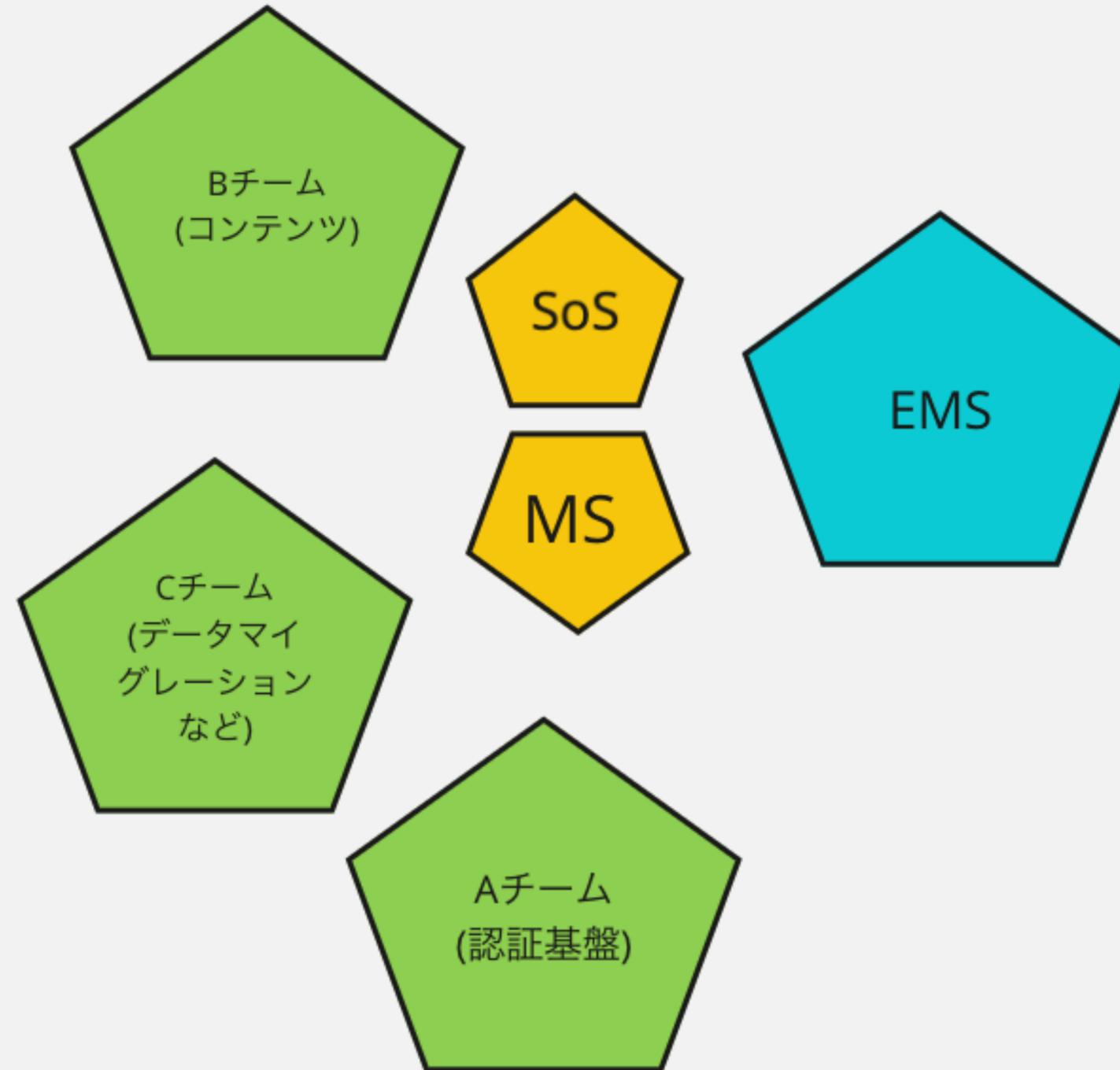
MS

イネイブリングチーム

とある要素技術についての  
タスクフォース

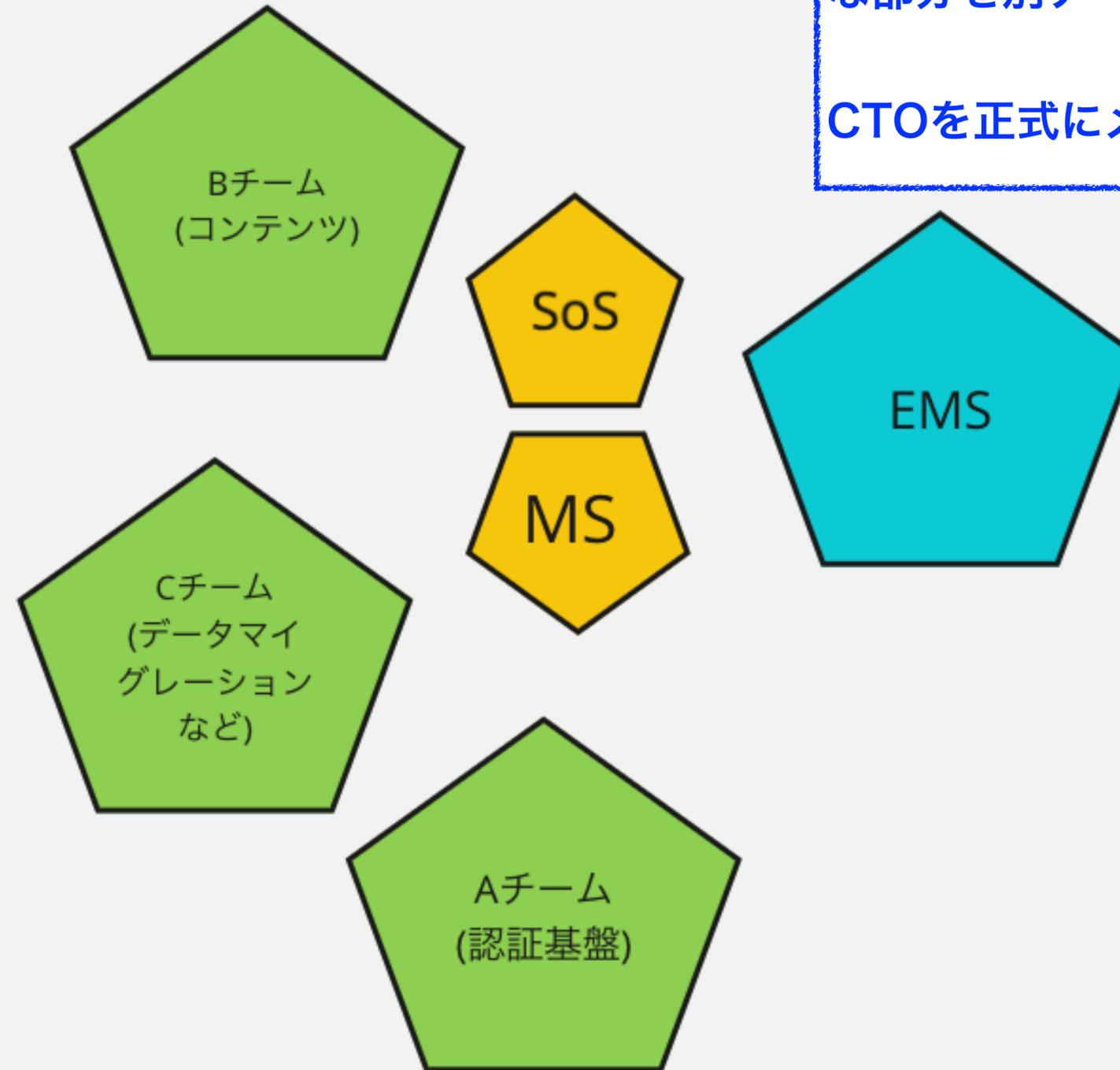
# 推移

2021年9月-10月



# 推移

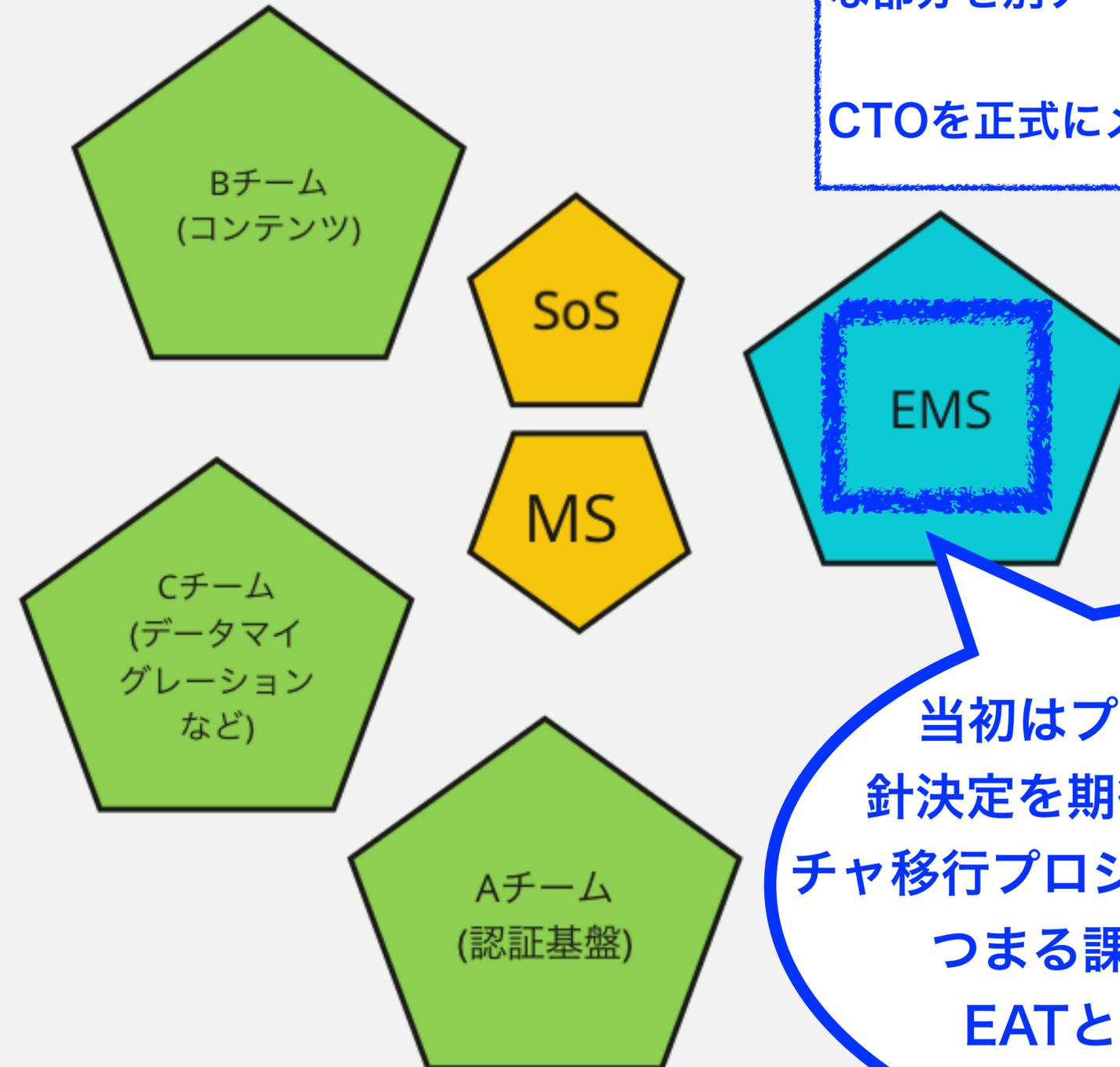
2021年9月-10月



プロジェクトのスコープ調整を実施  
Xチーム(UI/UX)がいったん休止  
Bチームの認知負荷軽減のためにサポータティブな部分を別チーム化  
  
CTOを正式にメンバーに入れたEMSを発足

# 推移

2021年9月-10月

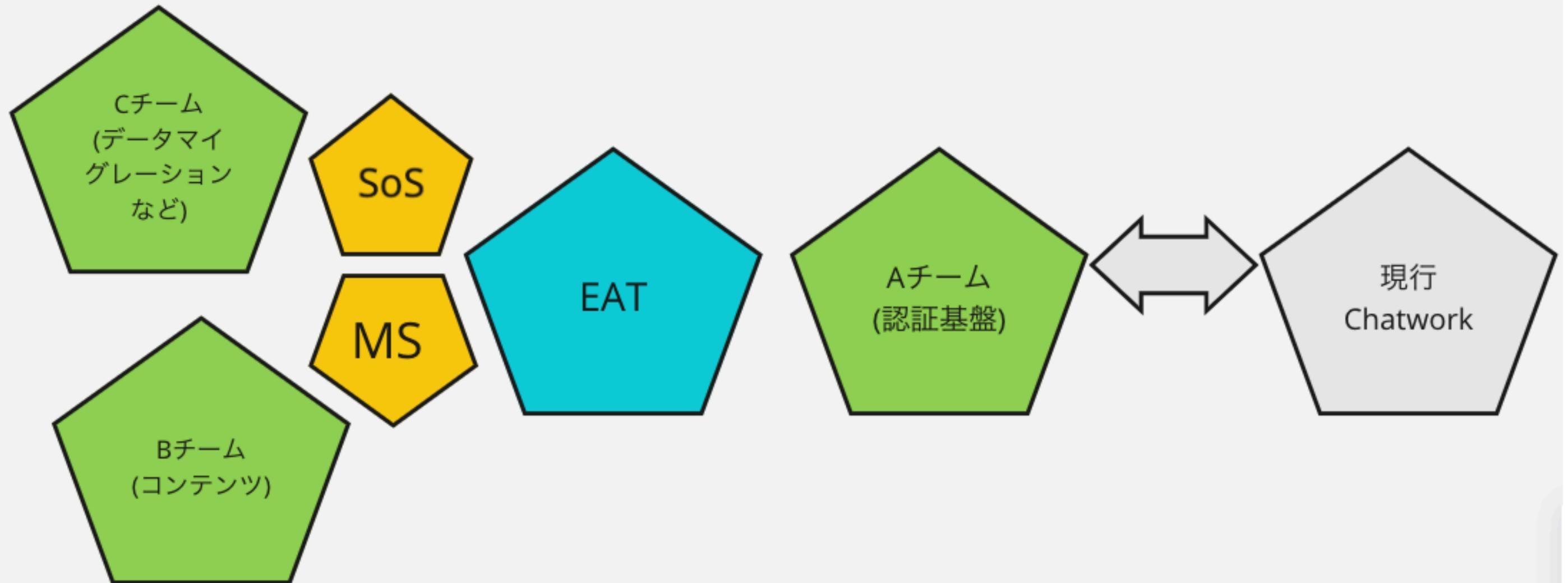


プロジェクトのスコープ調整を実施  
Xチーム(UI/UX)がいったん休止  
Bチームの認知負荷軽減のためにサポータティブな部分を別チーム化  
CTOを正式にメンバーに入れたEMSを発足

当初はプロダクトとしての方針決定を期待したが、アーキテクチャ移行プロジェクトの性質上ここにあつまる課題がHowに偏る。  
EATとして再編を決断。

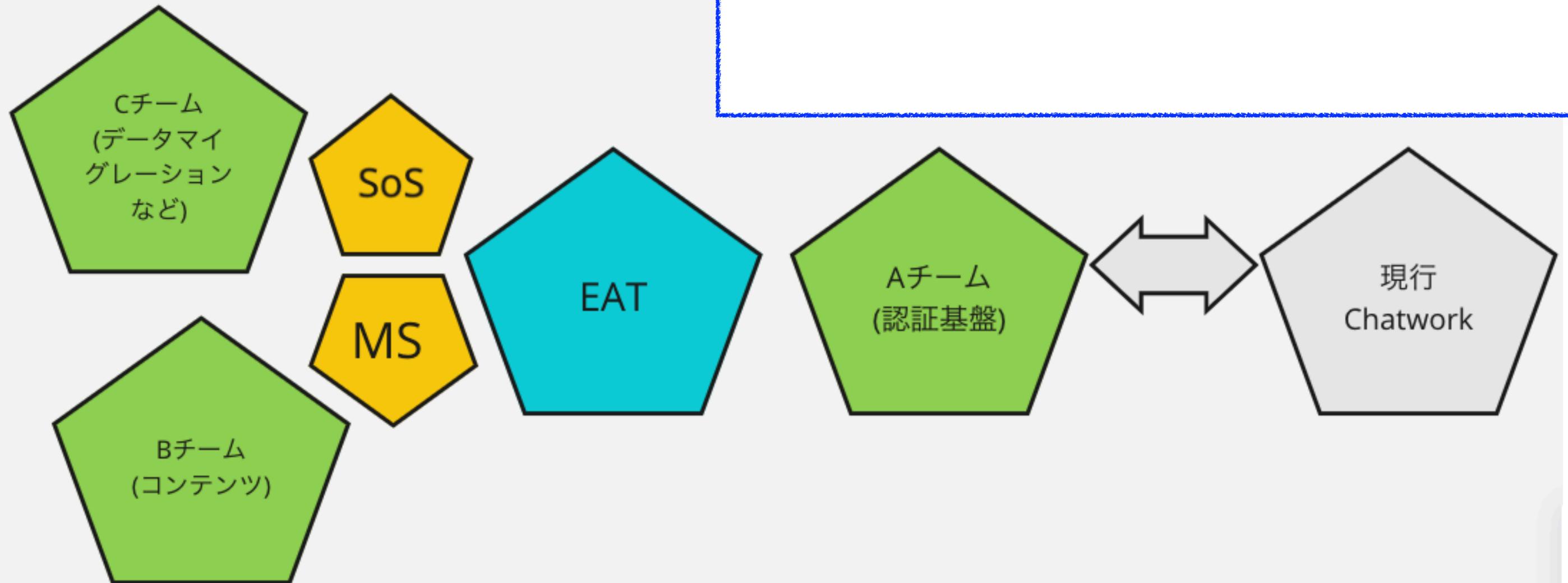
# 推移(現在に至る)

2021年11月



# 推移(現在に至る)

2021年11月



DevHR(開発チームの人事責任者)を招いて、EMSを EATに再編  
(この体制に加えてEMSを新規で立ち上げるのを準備中)

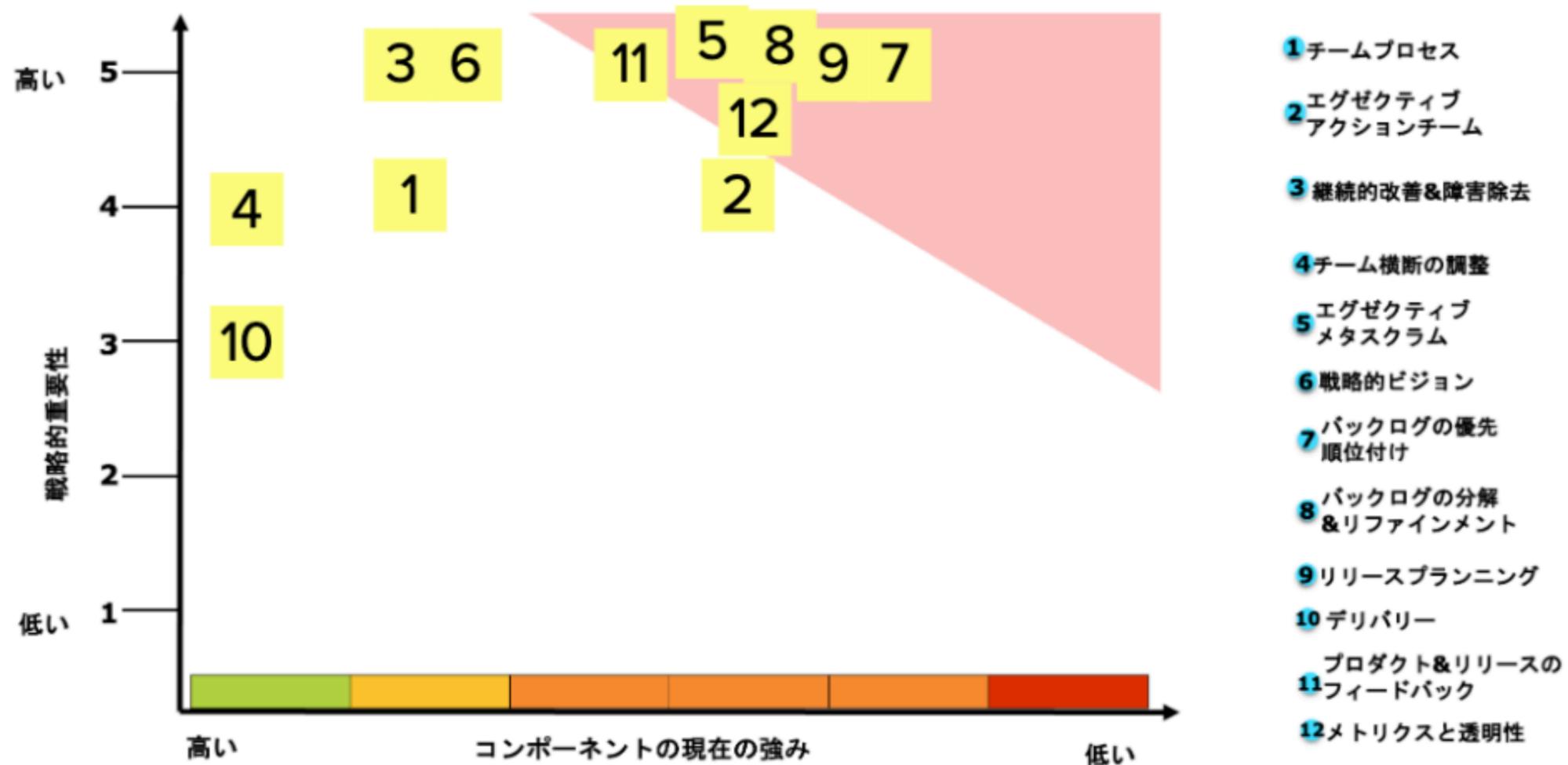
各チームの依存関係を整理して、SoSによる3チームのつなぎこみを整理。Aチームは現行システムとのつながりを強化

# Scrum@Scaleの 実装を改善する取り組み

# 12のコンポーネントそれぞれの採点

かすや

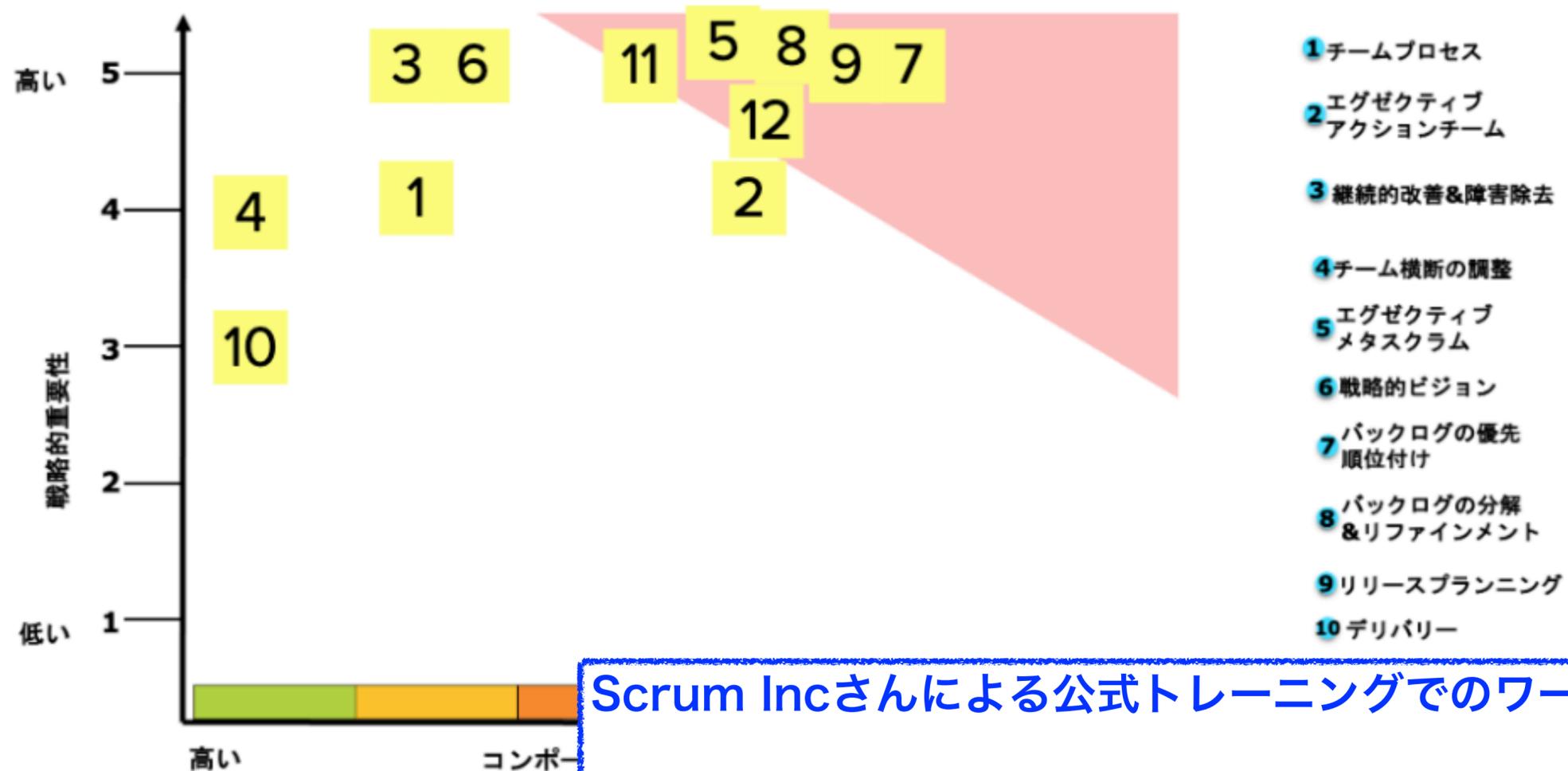
## 自組織のコンポーネントの評価サマリー



# 12のコンポーネントそれぞれの採点

かすや

## 自組織のコンポーネントの評価サマリー



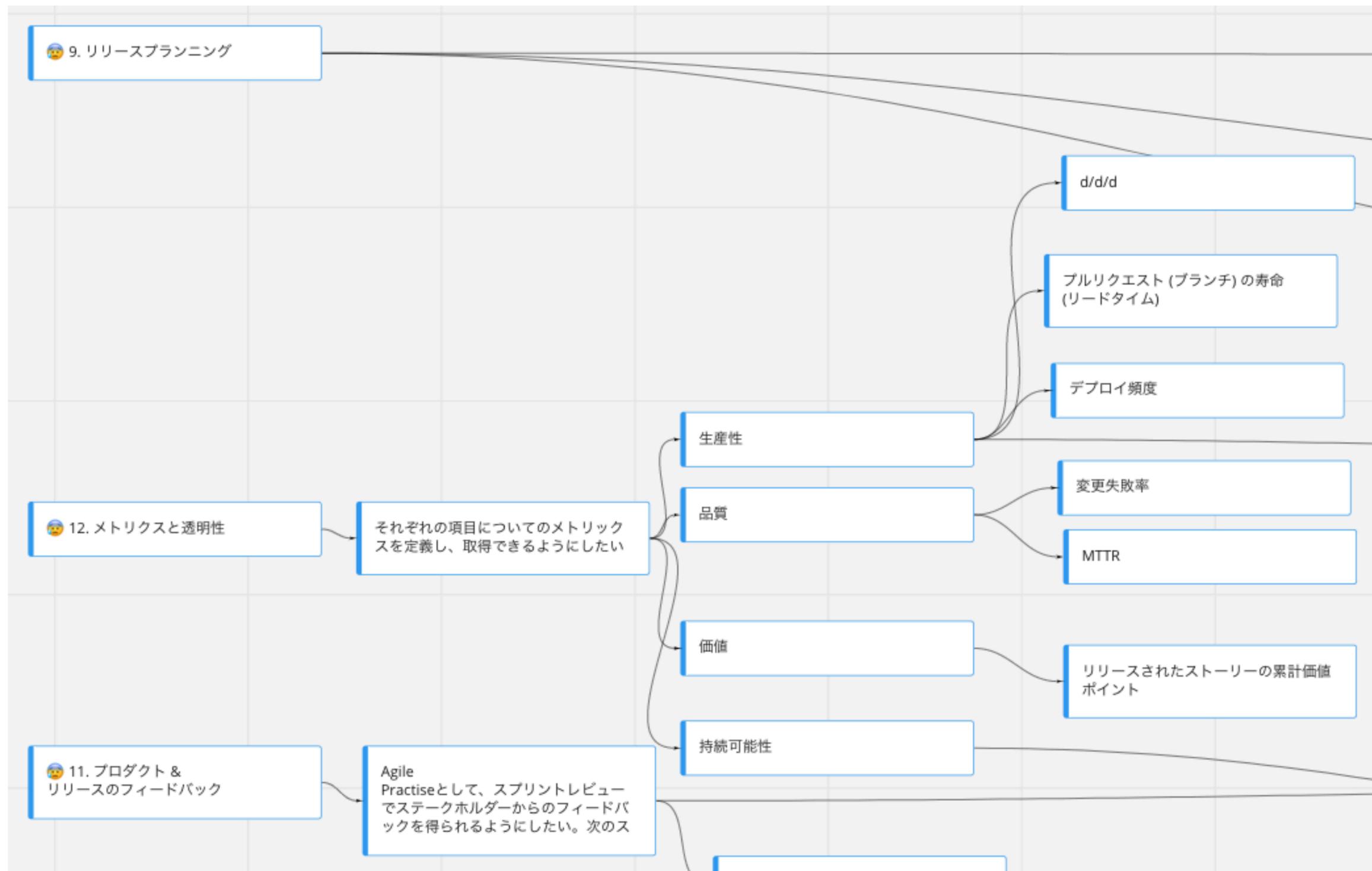
- 1 チームプロセス
- 2 エグゼクティブアクションチーム
- 3 継続的改善&障害除去
- 4 チーム横断の調整
- 5 エグゼクティブメタスクラム
- 6 戦略的ビジョン
- 7 バックログの優先順位付け
- 8 バックログの分解&リファインメント
- 9 リリースプランニング
- 10 デリバリー

Scrum Incさんによる公式トレーニングでのワークショップ

# 取り組みの優先順位付け



# 変革バックログ



# プロの支援を受ける

- 検討段階でScrum Incさんから簡単なレクチャーを受けた
- プロジェクト立ち上げ時に、2名のプロコーチに2つのチームを支援してもらった
  - 株式会社ラフダイヤモンドの山根さん
  - 株式会社豆蔵の北濱さん
- 粕谷と、2名のPOとでScrum Incさんの公式トレーニングを受講

独学でやるよりいろいろなことがスキップできたはず（知の高速道路）

# これまでScrum@Scaleを運用してきた所感

- 日々の悩みは普通のスクラムと一緒に。それぞれのチームが「普通のスクラムの悩み」と向き合いながら少しずつ「普通のスクラムチーム」として整えていく
  - バックログつくるの難しいねえ
  - ベロシティが安定しないけどどう改善する？
- 情報の流れが滞ったり違和感があればSoSなどのレトロスペクティブで扱っていく。場合によっては思い切った変更を加える（ただし単一のチームを大きく変えるようなことは滅多にやらない）
- エグゼクティブの巻き込みが難しい…！！（EMSをどう立ち上げようか頭を抱えています）

# **Regional Scrum Gathering Tokyo 2022**

# 出典など

- Scrum@Scaleガイド 日本語版
  - <https://scruminc.jp/scrum-at-scale/guide/>
- Scrum@Scale Guide
  - <https://www.scrumatscale.com/scrum-at-scale-guide/>
  - 本スライド「Scrum@Scaleの解説」パートで使用している  
Scrum@Scaleに関する図はすべてこちらのガイドのものを使用 (CC BY-SA 4.0)