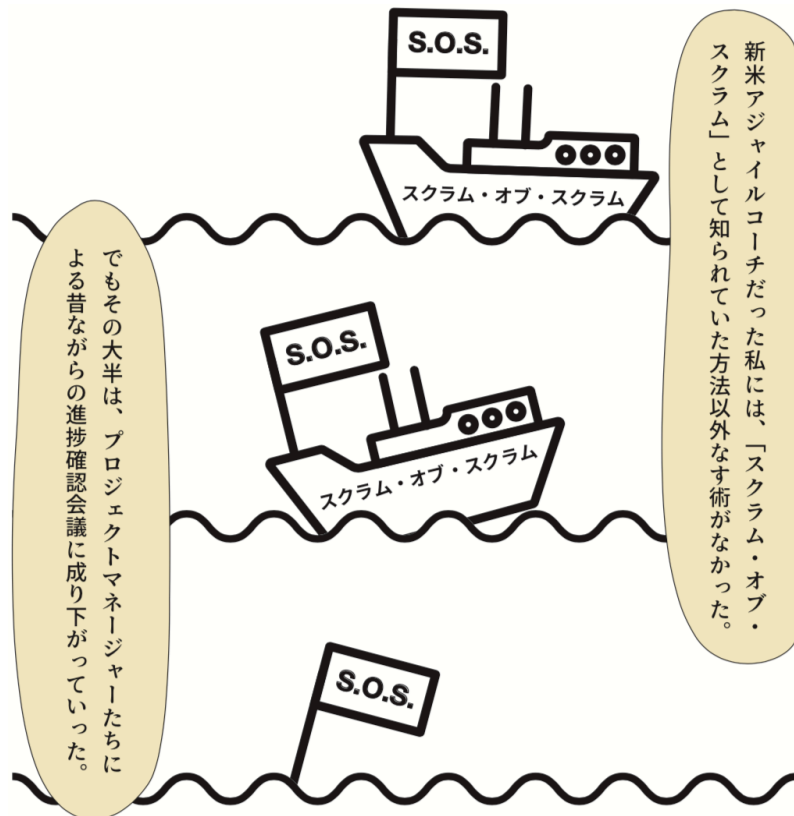


面倒なスクラム・オブ・スクラムに取って替わる7つの方法

スクラムマスター、ピーター・ウーリィはこう言っています。

スクラム・オブ・スクラム (SOS) のページ (「スクラム導入後にアジリティが減少してしまう理由」P.6) はとても愉快なものでした。これが実際にうまく行ったのを一度だけ見たことがあります。それは、スクラムマスターとプロダクトオーナーが週に3回の見ミーティングを実施していた時のことです。あらかじめ決まった事項や作業に関する同意などを共有するためのこのミーティングは、ラウンドロビンレビューや、つまらないタスクに毛が生えたレベルの近況報告になってしまい、プロジェクトマネージャーたちが互いを混乱させ収集がつかなくなることが度々ありました。

では最初に、実用的なリンクをこの記事で紹介しておきましょう。記事を読む時間のない人は、<https://less.works/jp/less/framework/coordination-and-integration.html> (<https://less.works/jp/less/framework/coordination-and-integration.html>) で、7つの代替方法の概要を読むことができます。



スクラム・オブ・スクラムの背景

ブラックブック

2001年に、ケン・シュエイバーとマイク・ビードルが初めてスクラムを扱った本を出版しました。これは、その不快な表紙にちなんで「ブラックブック」と呼ばれています。本書は、抽象的な理論の大集合であり、スクラムをいかに実践するかという側面には、小規模でさえも、ましてや複数チーム規模ではほとんど触れられていません。このブラックブックには、複数チーム

からのスクラムマスターが定期的に打ち合わせし調整するという発想が含まれていたのです。この発想には、「スクラム・オブ・スクラム」という、この発想そのもののフラクタル構造を反映した、誘惑するようなタイトルがついています。

グレーブック

ケン・シュエイバーは、2004年に「グレーブック」として知られるスクラムによるアジャイルプロジェクト管理（Agile Project Management With Scrum）という彼のキャリア史上最高の本を出版しました。ブラックブックは特に読む必要はありません。しかし、グレーブックを読んでこそ、現在のスクラム・ガイドを本当に理解しているといえます。残念ながら、本書はまだ「スクラム・オブ・スクラム」を扱っていますが、「スクラムマスターの代わりにチーム代表（2007年のシュエイバーの著書では「エンジニア」と特定されている）を送る」という改善案が含まれているのが幸いです。尚、本書の最終ページには、一読に値するケンの自己批判が掲載されています。

2003年にミラノで行われたスクラムマスターのミーティングで私がこれらの事例を発表した時、Mel Pullenは、「スクラム内部のスクラム」のプラクティスがスクラムの自己組織化と自己管理のプラクティスに反すると指摘しました。Melは、階層的な構造は経営陣の押し付けであり、実際に作業を行なっている人によって最適に導き出されたものではないと主張しました。他のどのチームと協力して調整する必要があるのか、チームはどこで連結するかを、各チームに判断させたらどうかと彼は言うのです。そうすればスクラムマスターがチームの依存性を指摘することもでき、あるいはチームは開発の途中でその依存性に遭遇するかもしれません。チームが依存関係で行き詰まっているときは、そのチームは依存性のある他のチームの日次（デイリー）スクラムへ、「ニワトリ」となる人を送ることができます。このような他のチームが存在しなければ、依存性がないチームは高い優先度のプロダクトバックログ項目を作成することを要求できます。そして、スクラムマスターは最初のチームに依存関係を持たせるか、あるいは別のチームを作成するかのどちらかを行うことができます。

これは興味深い観察でした。スクラムの拠り所は、自己組織化と、シンプルで指針となる規則の療法です。プロジェクトを調整して拡張するためにはどちらを適用できるでしょうか。私は両方を試し、適切なソリューションは内包される複雑性に依存することがわかりました。複雑性がとても大きくて自己組織化が迅速に発生しなければ、シンプルな規則を使用すると組織はタイムリーな解決策に到達できません。しかし自己組織化がタイムリーに発生したら、これに頼る方がいいでしょう。その理由は、経営陣はチームほど頻繁かつ見事に適応を考案することはできそうにないからです。スクラムマスターは、2、3のシンプルな規則を考案することによって自己組織化を助けることができますが、スクラムマスターにとっては、自己組織化を助けないことより、助けすぎることの方が簡単です。

当時のシュエイバーのCSMトレーニング材料やオンラインで閲覧できる談話等から、彼が「スクラムマスターはチームに対する権限を持つべきではない」と徐々に気づき始めていることが分かります。

否定

あるスクラム関連の会合（おそらく2007年後半のストックホルム）にて、シュエイバーは公然と「スクラム・オブ・スクラム（SoS）」を否定しました。SoSを扱ったスライドは、彼のトレーニング資料からほとんど削除されていました。数年後のディスカッショングループでは、マイク・ビードルが「ブラックブックの中で、スクラムマスターをチーム代表として使うことを薦めたのを悔やんでいる」と話していました。

2008年、クレイグ・ラーマンとバス・ボッデはリーン開発&アジャイル開発のスケールリング (Scaling Lean & Agile Development) という本の中で、

SoSについて最も多い誤解は、スクラムについての調整ミーティング実施にSoSが最高または唯一の方法だと思われていることです。SoSは（ほとんど実験をせずに）初めて提案された時には合理的な考えだと思われたのですが、現在では、もっと効果のある良い方法がどんどんと出現しています…

SoSミーティング推奨が構造推奨へと変貌

バス・ボッデから、私に下記のようなメッセージが届きました。

スクラム・オブ・スクラムの二つの概念をはっきりと区別させるほうがいいのではないかな。

1) SoSミーティング：これはシュエイバーの出版物の多くに現れており、スケールド・アジャイル・フレームワーク (SAFe) と関連している。ほとんどの記事がこれに言及している。

2) 「スクラム・オブ・スクラム」の構造的概念：これはセサリオが言及していることで、Nexus（大規模スクラム）内で「ネクサス」と呼ばれている。それはLeSSにおける要求領域として考えることもできる。（私はそう考えていないが、その根拠は理解できる。）

私の観点では、歴史的に（1）が（2）へと変貌していったように思える。

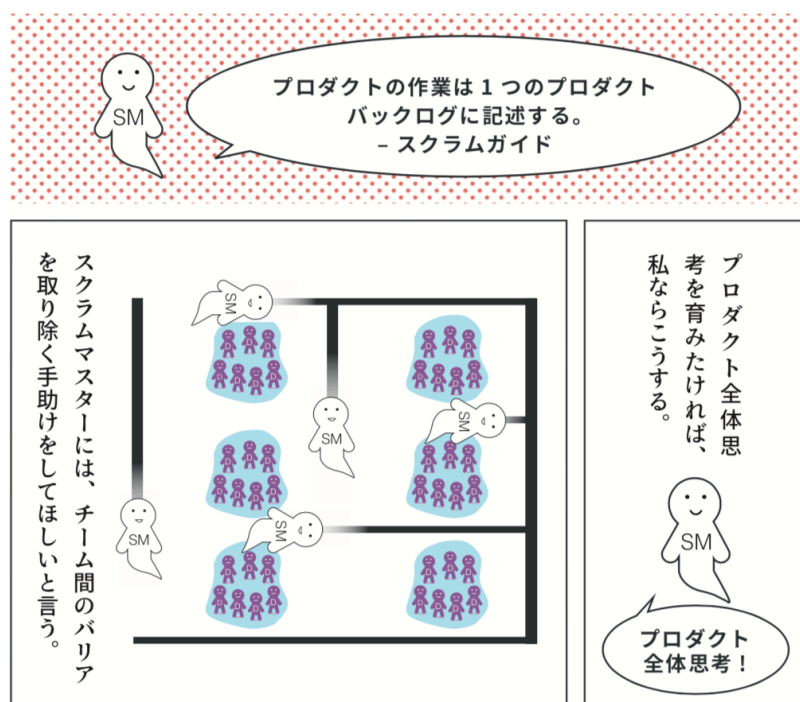
バス

問題発生から16年、シュエイバーの否定から10年、なぜ未だにSoSは実践されているのか

私は、SoSとはいつまでも脳内で繰り返すどうでもいい歌のようなものではないかと思っています。11名以上の組織に対する実践についての一貫した考えがないため、影響力を持つ人たち（2008年にスクラムトレーナーとなったマイク・コーンやジェフ・サザーランドなど）がSoSを未だに推奨しています。SoSはスクラム・ガイドには登場しませんが、大々的に宣伝されている スケールリングへの大規模アプローチ ("big box" approaches to scaling)

(<http://www.lafable.com>)に組み込まれています。SoSは、LeSS（あまり宣伝されていないスケールリングアプローチ）ではほぼ推奨されていません。それは、組織の学習・適応能力を上げるのに、より優れた代替方法があることが分かってきているからなのです。

調整と統合：代わりの方法



以下の方法についての説明は、全て<https://less.works/jp/less/framework/coordination-and-integration.html> (<https://less.works/jp/less/framework/coordination-and-integration.html>)からの引用となっています。

従来のプロジェクトマネジメントでは管理された調整が行われてきました。スクラムでは集中管理ではなく、分散と自己組織化された調整と統合に価値があると考えます。LeSSでは必要十分な境界線と構造を提供し、カオスな状態になるのを回避し、分散型で調整が可能な状態になることを重視しています。

1. ただ話す

チーム間の調整にもっとも良い方法は単純に、ただチーム間の調整をすることです。課題があれば自己管理されたチームの誰もが他チームの所に行って議論することが期待されています。もし、「ただ話す」ことが出来ない状況なのであれば、電話したり、最悪の場合にはメールを送れば良いわけです。調整の為に形式張った場は不要で、調整が遅れるだけです。立ち上がって話しに行ってください。

誰かが話したいと思っているかどうかはどのようにわかるのでしょうか。LeSSでは、各チームに別々のチームアウトプットオーナーがいるといったような障害物を取り除きます。一つの出荷可能なプロダクトインクリメント (<https://less.works/jp/less/framework/potentially-shippable-product-increment.html>) を扱う一つのスプリントレビュー (<https://less.works/jp/less/framework/sprint-review.html>) に向けて、複数チームが一つのスプリントに取り組みます。スクラムマスターは、各チームの個別アウトプットではなく、チーム間コラボレーションを奨励します。

2. コードでのコミュニケーション

LeSSを導入しているグループでは継続的インテグレーション (<https://less.works/jp/less/technical-excellence/continuous-integration>)が取り入れられており、全てのコードは1つのレポジトリにチェックインされています。ブランチは不必要な複雑性を生むので使わないことをオススメします。全員が日に数回はレポジトリと同期し、他の人が行った全ての変更を取り込むようにします。

updateをした時に2分程度の時間を使って他の人の作業と自分の作業が被っていないか確認します。もし、作業が被っていそうであれば、ぜひ席を立て、「ただ話す」をしに行って同じ箇所を触っている人と作業の調整をしてください！

ブランチに分けるとインテグレーションが遅れます！ 時間と共に大きく増えていく課題をまとめてください。

- 他にもコードでのコミュニケーションを難しくする要素があります。それは多くの企業に存在する 私のコード、あなたのコード (<https://seattlecrum.com/my-code-your-code/>) の方針や習慣です。

3. オブザーバーをデイリースクラムに送り込む

チーム間の簡単な調整の方法の1つがスクラムマスター以外の代表者を関係する仕事をしているチームのデイリースクラムに発言しないオブザーバーとして送り込むことです。そして、代表者が自分のチームに戻ってチームに内容を共有し、次のアクションにつなげます。

あいにく、局所最適化志向のスクラムマスターは、これを妨げようとすることがあります！逆に、プロダクト全体思考を持ったスクラムマスターは、これを可能にする努力を惜しまないでしょう。

4. コンポーネントのコミュニティとメンター

同じコンポーネントを扱っている人達は質問したり、コードレビューを相互にする為に、お互いを知っておく必要があります。これはコンポーネントのコミュニティ・オブ・プラクティス(CoPs) (<https://less.works/jp/less/structure/communities.html>)を作ることで対応していきます。CoPsはメーリングリスト、チャット、ミーティングや、様々なりモートでの協働方法を用います。

これらのコミュニティは殆どの場合、コンポーネントメンターにより組織されます。コンポーネントメンターはフィーチャーチーム内で次の様な役割を担うことがあります。(1) コンポーネントがどのように機能しているのかを教える先生役、(2) 中長期的にコンポーネントの健全性が維持されるようにメンターとなる、(3) コンポーネントのコミュニティを主催する、(4) 設計のワークショップを主催、(5) コードのレビュー。

コンポーネントメンターがコードレビューを行うのは、コミット前に承認をする為ではありません。彼らは先生であり、世話役なだけで、門番ではないのです。

1つのコンポーネントに複数のメンターがいる場合もあります。複数人いることにより、1人当たりの仕事量も減らせますし、キーマンに依存する状態も回避できます。

たいへん重要な点ですが、調整をサポートする以外にもこれらの活動がコンポーネントのコードや設計などの品質を維持、改善し、学習を促進することに繋がります。

以前に一緒に仕事をしたことのある組織では、コンポーネントメンターを別のチームから招き、メンターがキーボードには触れないという条件で、よく知らないコードについてのサポートを得たことがありました。このような事例は、[知識格差の真の代償 \(https://seattlescrum.com/misconception-1-dependencies-are-caused-by-immutable-laws-of-physics/\)](https://seattlescrum.com/misconception-1-dependencies-are-caused-by-immutable-laws-of-physics/)に気づかない人には「非効率」に思えるかも知れません。

5. スクラム・オブ・スクラム

スクラム・オブ・スクラムのミーティングはチーム間向けのミーティングで週に2〜3回実施されます。

多くの場合はデイリースクラムのように下記の3つの質問がされます。：

1. 他のチームに関係あることで、どのようなことを私のチームは行ったか
2. 他のチームに関係あることで、何を行うか
3. 他のチームに知っておいて欲しい、又は助けて欲しい私のチームの課題は何か

これらの質問はグループにとって有用な質問に変わっていくことが多いです。

注意点：スクラム・オブ・スクラムが必要だと感じるのは、コンポーネントチームやシングルファンクションのグループになってしまっているか、協働が出来てないか、協働しようと思わない環境になってしまっていることにより、不必要な依存や調整が発生してしまっている兆候かもしれません。

スクラム・オブ・スクラムはLeSSの一部ではありません。どちらかというと、集中管理をする方法ですし、オススメもしませんが、スクラム・オブ・スクラムで上手くいっているのであれば、やめる必要もありません。ただ、LeSS導入に必須ではないということです。

それでも、スクラム・オブ・スクラムミーティングを実施するというのであれば、チームメンバーを入れ替えるようにしてください。スクラムマスターはチームを調整する権限がないため、当然ながらこのミーティングにはスクラムマスターを送ることはできません。

6. 複数チームのミーティング

いくつかのチームが密接に連携する必要があるが、他のチームはそれほど密接に連携をする必要がないという状況がよくあります。密接に連携する必要があるチームが下記のようなミーティングを一緒におこなうことはよくあります。

- 複数チーム [プロダクトバックログリファインメント \(https://less.works/jp/less/framework/product-backlog-refinement.html\)](https://less.works/jp/less/framework/product-backlog-refinement.html)
- 複数チーム [スプリントプランニング2 \(https://less.works/jp/less/framework/sprint-planning_two.html\)](https://less.works/jp/less/framework/sprint-planning_two.html)
- 複数チーム [設計ワークショップ \(https://less.works/jp/less/technical-excellence/architecture-design.html\)](https://less.works/jp/less/technical-excellence/architecture-design.html)
- デイリースクラムで相互にメンバーを送り合う

お分かりになりましたか？つまり、必要なミーティングだけを実施しようということです。[オープンスペースカンファレンス \(https://www.amazon.com/dp/B005X00KOY/\)](https://www.amazon.com/dp/B005X00KOY/)を読んだことのある人は、学習も貢献もしていない時には自分の足でそれらが出来る場所を探すという「二本足の掟」を知っているでしょう。

7. トラベラーを活用してボトルネックを解消し、スキルを伸ばす

プロダクトグループは数名の経験豊富な技術的なエキスパート達を頼みにしていることがよくあります。このように全てのチームから必要とされているが、十分な人数がいない時に、全てのチームが彼らの知識を使える状態にするにはどうしたら、良いのでしょうか？彼らにはトラベラーになってもらいます。毎スプリント彼らには別のチームのメンバーになってもらい、ペアを組んでコーチしてもらったり、ワークショップを開催してもらったり、技術的な教育をしてもらいます。 .

そして、よく勘違いされるのですが、トラベラーは直接的な調整役を担うわけではありません。彼らは複数のチームに入ることによって、入っているチームが他のチームとの幅広い関係性を作ったり、既存の関係性を強化することにより、形式ばらない相互での調整を可能にしていきます。彼らの活動を通じて、チームを横断して、知識や技術の習得を促進し、チーム間の調整も可能にしていきます。

(チーム自己設計ワークショップ (<https://www.ahmadfahmy.com/blog/2013/12/5/the-rise-of-the-team>) を使って) ある組織の再編成をの手助けをしたことがあります。驚いたことに、そこでは数名の専門家がチームに参加することを断り、自分たちをトラベラーと呼んでいました。彼らも数か月後にはめでたくチームに所属していました。

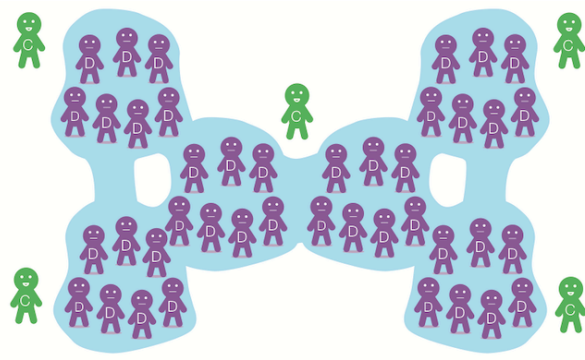
8. リーディングチーム

いくつかの領域では、フィーチャーはすごく大きい場合があります。この巨大なフィーチャーを作るには多くのチームが協働しながら小さく分割されたプロダクトバックログアイテムを通じて巨大なフィーチャーを完成させていきます。

巨大なフィーチャーを作るもう1つの方法がリーディングチームです。リーディングチームは通常のフィーチャーチームですが、巨大なフィーチャーを先行する役割を担います。彼らは自分達の開発作業の他に、他チームの仕事内容を確認し、同期を支援する責任を持ちます。簡単にいうと、彼らは巨大フィーチャーを作る上で、自分達の仕事の他にチーム横断での調整を行います。

ときには複数チームが同時に巨大な開発を同時に始めることがあります。又は、リーディングチームが先行して進め、知識を伝え、設計をシンプルにします。数スプリント後にいくつかのチームが合流しますが、リーディングチームは後から合流したチームに彼らが既に知っていることを教えていくことに責任を持ちます。

最適化された環境下でチーム一つが持つ力を過小評価しないでください！以前仕事でかかわったことのある政府機関の開発部長が言っていたことですが、スクラムを導入するにあたり有能なチームを立ち上げようと思ったら、スタッフが40名いたにもかかわらず、わずか1チームしか立ち上げられないことに気付いたそうです。他にも、FBIが100人以上駆り出して処理できなかったことが、本部地下のオフィスで働く少人数のグループによって解決された、という話もあります。最適化された環境下でチーム一つが持つ力を過小評価しないでください！以前仕事でかかわったことのある政府機関の開発部長は、スクラムを導入するにあたり、有能なチームを立ち上げようと思ったら、40名いたスタッフの中でわずか一つしか作れないことに気づきました。他にも有名どころでは、FBIが100人以上駆り出して処理できなかったことが、本部地下のオフィスで働く少人数のグループによって解決された、という話もあります。



英語版 <https://seattlescrum.com/seven-alternatives-to-scrum-of-scrums/>
(<https://seattlescrum.com/seven-alternatives-to-scrum-of-scrums/>)