Scrum.org WHITEPAPERS

スクラム:リスクを軽減し早期に価値を提供するフレームワーク

スクラムの初学者や学び直したい人に向けた スクラムフレームワークの概要解説

CHRISTIAAN VERWIJS, BARRY OVEREEM, JOHANNES SCHARTAU (THE LIBERATORS) 著、長沢 智治 訳 | NOVEMBER 2020

序文

「スクラムのことは気にしていない」と、私たちはときどき言うことがある。それは確かに眉をひそめることであるが、私たちが言いたいのは、スクラムフレームワークそのものではなく、スクラムフレームワークが何を可能にするのかが重要だということである。このような見方ができると、多くの理論的な疑問が自明になったり、無意味になったりする。例えば、「スプリントバックログにあるすべてのアイテムは、スプリントゴールに関連しているべきか？」、「バグはプロダクトバックログに入れるべきなのか？」、「デイリースクラムは 15 分ちょうどで止めるべきか？」などである。大局を忘れたか、または見ていないときに、このような細かいところに行き詰まりがちである。

私たちは、スクラムフレームワークが可能にしてくれることを大事にする。公式の『スクラムガイド』は、既にそのことを簡潔に説明するという大きな成果をあげている。私たちは、スクラムチームと仕事をし、Scrum.org のプロフェッショナルスクラムトレーナーとして仕事をしている中での説明を、私たち自身の言葉で加えていきたい。私たちが目指したのは、スクラムフレームワークが「何を可能にしているのか」という観点から、実践的で地に足のついた方法で執筆することだった。このホワイトペーパーが、読みやすく、混乱の可能性を取り除き、スクラムの理解を深めるものであることを願っている。

このホワイトペーパーは、Johannes Schartau 氏と共に私たちの書籍『The Zombie Scrum Survival Guide』のために書いた。

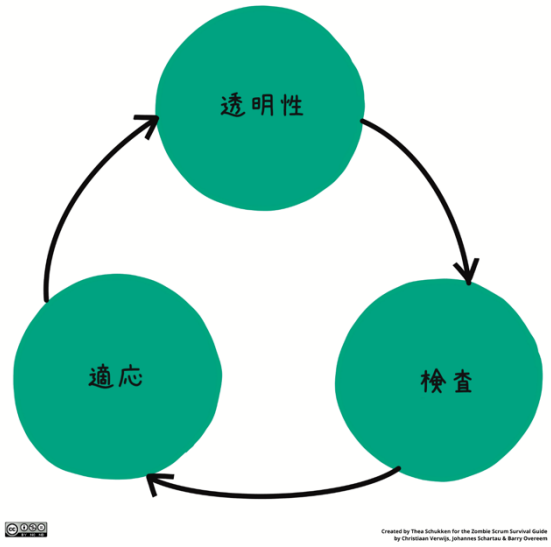
歴史の一端

スクラムフレームワークは、1990 年代に Ken Schwaber と Jeff Sutherland が開発し、プロダクトおよびソフトウェア開発に特有の複雑さに対処するものとして、1995 年にはじめて正式発表された。最近では、マーケティングから組織的な変革へ、科学研究からソフトウェア

開発へと、さまざまな領域での複雑な問題にうまく適用されている。スクラムフレームワークは、どこで用いたとしても、経験的プロセス制御を可能とする 3 本柱で支えられている。

経験主義の 3 本柱

スクラムフレームワーク[1]は、経験的プロセス制御を可能とする 3 本柱で支えられている。



- **透明性：** 何が起きているかを確認するために、メトリクスやフィードバックまたは、その他の経験などのデータを収集する。
- **検査：** 関係者全員で進捗を検査し、それが目標設定にとって何を意味するのかを決定する。
- **適応：** 目標設定に近づけるように望ましく変化する。

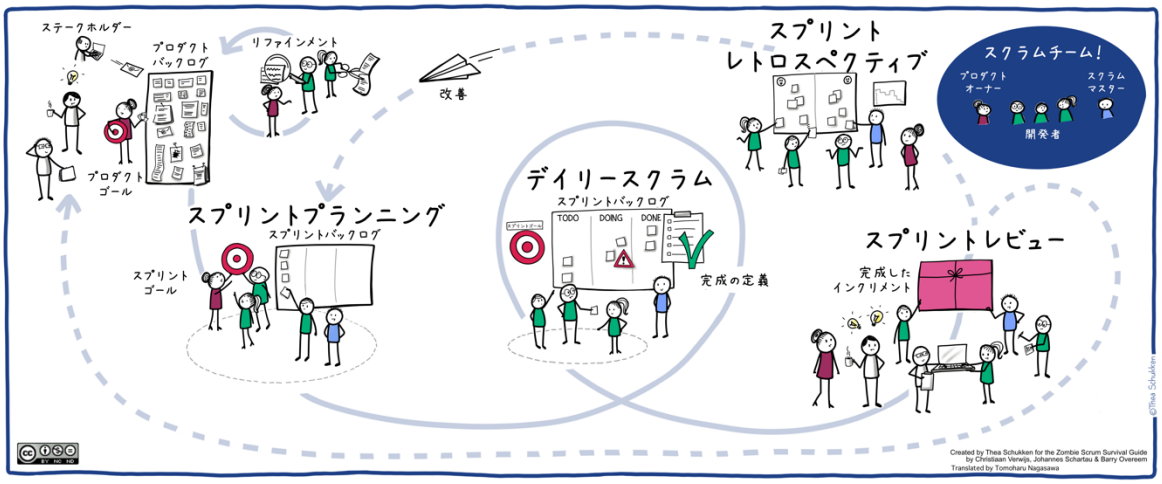
このサイクルは、仕事をしていく中で出てくる逸脱や予想外の発見、潜在的な機会を捉えるために必要に応じて何度も繰り返す。その頻度は1年に 1 度やプロジェクトの完了時ではなく、毎日、毎週、毎月のように継続的に行う。将来的な仮説に基づいた意思決定を行うのではなく、これまでに収集したデータに基づいた意思決定を行う。これが経験主義である。このホワイトペーパーによって、スクラムフレームワークのすべてが、これらの 3 本柱を中心に設計したものであることを理解するだろう。

スクラム：経験的プロセス制御のフレームワーク

一つ言えるのは、プロダクトに関する作業には透明性が必要であり、その作業を頻繁に検査する必要があり、そこから得られるインサイトが適応を促進するということである。それを具体的で現実的な方法で実践することはまた別のことである。これがスクラムをフレームワークたらしめているものである。

スクラムフレームワークは、3 つの作成物に取り組むために、5 つの繰り返すイベントと、これを手助けするための 3 つの責任と共に、これらを一つにまとめるための原則とルールを提供している。

[1] https://scrumguides.org



スクラムは経験的プロセス制御のフレームワーク

スクラムの作成物

経験的プロセス制御の第一の柱は、「透明性」である。プロダクトに関する作業の進捗を頻繁に検査し、何が（他に）必要かを決定するために、検査できるものが不可欠である。プロダクトの作業を「透明」にするということは、プロダクトに関係する人たちが作成物を見て、仮説を検証し、そこから新しいアイデアを生み出せる形でそれを利用できるようにすることを意味する。

スクラムフレームワークでは、チームはプロダクトに関する少なくとも 3 つの要素を透明にすることが求められる。これらを「作成物」と呼んでいる。作成物は、将来の意思決定に必要なデータや経験を収集するための主要な手段となっている。

プロダクトバックログ

1 つ目の作成物はプロダクトバックログである。これによりプロダクトゴールを達成するためのプロダクトにおける必要なすべての作業を透明にする。プロダクトゴールは、スクラムチームがプロダクトの長期的な目標を一つにまとめて確約できるようにすることで、プロダクトバックログの作業に集中することができる。スクラムチームがプロダクトゴールに向かって作業を進めていく中で、プロダクトバックログは常に変化していき、その作業中に新たなインサイトや機会が生まれてくる。新しいプロダクトゴールは、以前のプロダクトゴールを達成したとき、または放棄したときにのみ設定すべきである。

スクラムチームがプロダクトゴールを達成するために必要だと現在把握しているすべてのアイデア、仮説、ユーザー機能、バグ、タスクは、プロダクトバックログに個別のアイテムとして表現される。アイテムは、非常に大まかではっきりしないものもあれば、小さく具体的なものもある。すべてのアイテムは、プロダクトとそのステークホルダーの目標設定との関係性に応じて順番に並べられる。

プロダクトには、それに取り組んでいるチームの数に関係なく 1 つのプロダクトバックログと 1 つのプロダクトゴールがある。そうでないと、必要な作業とそのための順序について単一の「信頼できる情報源」ではなくなり、透明性と集中力を損なうことになる。

スプリントバックログ

2 つ目の作成物は、スプリントバックログである。これはスプリントの開始時に、開発者がそのスプリントで確約した単一のスプリントゴールを達成するために必要と考えたアイテムをプロダクトバックログから選択したものである。スプリントバックログは、現在のスプリントで開発者が取り組んでいるか、取り組む予定のあるすべての作業を透明にする。

それぞれのスクラムチームには、独自のスプリントバックログがある。1 つのプロダクトで複数のチームが作業していたとしても各チームに 1 つである。スプリントバックログは、静的なものではなく、スプリント中にチームが学習することで変化する。プロダクトオーナーと緊密に連携して開発者は、スプリントの残り時間とスプリントゴールに対するアイテムの関連性や必要性に基づいてアイテムを追加したり、削除したりすることができる。

インクリメント

3 つ目の作成物は、インクリメントである。それぞれのインクリメントは、包括的なプロダクトゴールに向けた一つのステップを表している。スプリントにおいては、スプリントバックログからプロダクトバックログのアイテムが完了するたびにインクリメントが作成される。混乱と期待の相違を避けるため、アイテムは徹底的に検証され、スクラムチームが確約した完成の定義を満たした場合にのみ、「完成」とみなされる。スクラムチームは、スプリント中に複数のインクリメントを作成し、スプリントレビューの前にステークホルダーに提供することもある。

インクリメントの目的は、これまでに行った作業の仮説を検証することである。プロダクトの関係者は、インクリメントを検査し、それが自分たちのニーズを満たしているかどうか、それがどのように機能するかを理解しているかどうか、そしてそれが自分たちの期待値を満たしているかどうかを判断できる。インクリメントは、新しいアイデアの原動力となる。なぜなら、インクリメントという具体的で目に見えるものを手にすることで、関係者が新しい可能性を見出すことができるからである。

インクリメントは、複雑な問題に対して経験的プロセス制御をスクラムチームが実践するための主要な手段である。すべてのインクリメントは、プロダクトをどのようにその目標設定に近づけるかという直感、仮説、あるいは価値を判断可能なアイデアから始まる。これはスプリントゴールで捉えているものである。スプリントゴールを達成するために必要な作業は、スプリントバックログに集約される。

複数のスクラムチームが 1 つのプロダクトで作業する場合は、効果的な検査が行えるようにスプリントごとに作業を 1 つのインクリメントに統合するようにする。

他の作成物は？

明らかにチームの作業には、他にも透明化できる要素が多くある。例えば、ステークホルダーマップを作成して、ステークホルダーが誰で、どこにいるのかを把握することができる。また、ダッシュボードにメトリクスを収集し、頻繁に検査することができる。これらは、立派な作成物であり検査する価値がある。しかし、経験的にプロダクトを開発する目的において、スクラムフレームワークでは、チームが 3 つの中核となる作成物であるプロダクトバックログ、スプリントバックログ、インクリメントで透明性を作り始めることを推奨している。

スクラムイベント

次に何をするべきかという意思決定を知らせるために、関係者全員が、プロダクトバックログや、スプリントバックログ、インクリメントを頻繁に理解しておくべきである。これが「検査」である。検査には、さまざまな形がある。プロダクトバックログを見て結論を出すことで、プロダクトバックログについて「それは長すぎる」や、「そのアイテムの前にこのアイテムを作業するべきではない」、「ステークホルダーにいつこれを提供できるのか？」などを理解する。ユーザーを集めてきて、仮説を検証することでインクリメントについて「ユーザーはこの新しいユーザー機能を理解しているのか？」や、「このユーザー機能はニーズを満たしているのか？」などを理解する。そして、スプリントバックログが、実現可能であり、現在のスプリントの計画を反映することを確認することで、スプリントバックログについて理解する。

どのような形であれ、目に見えるものや具体的なものに基づいて意思決定する場合には、検査が最良の意思決定の基準となる。プロダクト開発の場合、完成し、デプロイ済みのユーザー機能を検査することが、そのユーザー機能とその使われ方についての仮説を真に検証する最良の方法である。ユーザー機能のプレゼンテーションやデザイン提案を見ていると便利そうに見えるかもしれない。しかし、参加者たちはそれでも、ユーザー機能が実装され、どう機能するかについては、おそらく人によって異なった期待をしているだろう。

スクラムフレームワークでは、経験的に作業をするためには、チームが少なくとも５つの検査を繰り返すべきであると提唱している。それら５つには、それぞれに具体的なタイムボックスがあり、特定の作業の視点を提供する。これらを組み合わせることで、全体像を把握できる。これが５つのスクラムイベントである。「ミーティング」と呼ばれるが、テーブルの周りに座って、退屈して涙を流すような従来の意味でのミーティングではない。実際に、スクラムイベントが目的を明確に意識して行われれば、他のミーティングの必要性が減り、チームが仕事の中で自然と協働でき、その場で調整できるようになる。

スプリント

1 つ目のイベントは、スプリント自体のタイムボックスである。複雑なジグソーパズルを小さな区画から始めることで解いていくように、各スプリントの目的は、複雑な問題の一部分を解決していくことである。この部分的な解決策は、プロダクトの増分的なバージョン（すなわち、インクリメント）で表し、次に何を行うべきかを意思決定するために共に検査できる。このインクリメントは、少なくともプロダクトオーナーが意思決定した場合は、スプリント後に直接ボタンを押すことで、ステークホルダーにリリースできる状態にすべきである。さらによいのは、チームがスプリントを通してリリースできる状況にあり、チームの学習能力がさらに高まることである。スプリントの成功はさまざまではある（たとえ単一のスプリントであっても、多くの複雑で予測不可能な作業を伴う）が、いずれにせよ、パズルについて多くのことを学ぶことができる。

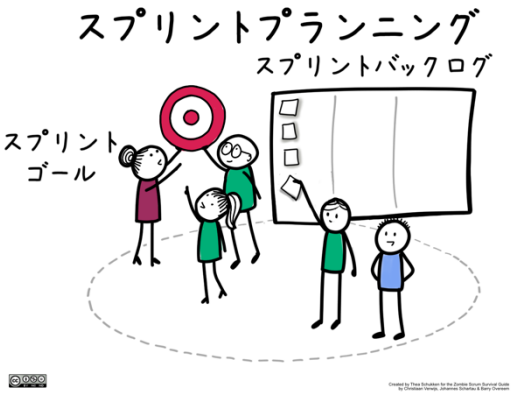
チームとステークホルダーに対してリズムと予測可能性を作り出すために、スプリントの期間は常に一定にすべきである。スプリント期間が長すぎると、仮説を検証したり、作業が正

しい方向に進んでいるかどうかを確認したりするための貴重な機会が失われてしまう。スプリントの期間が長くなればなるほど、間違ったものを開発して時間をムダにするリスクが大きくなる。スクラムフレームワークでは、スプリントの長さを指定していないが、1 ヶ月以内である必要がある。どのくらいの速さで学習できるか、学習したいかは、チーム次第である。一般的に言えば、スプリント期間は可能な限り短くし、開発者がスプリントゴールを達成するときには、プロダクトの新しい意味のあるバージョンを提供できるようにすべきである。

スプリントは、プロダクト開発という複雑な問題の特定部分を探究するタイムボックス化した機会である。他の 4 つのスクラムイベントは、スプリント期間の中で検査と適応を促進する機会を意味する。

スプリントプランニング

スクラムでの各スプリントは、スクラムチームがこれからのスプリントの基本計画を立てるときから始まる。これが、スプリントプランニングである。最初かつ最も重要な質問は、「このスプリントのゴールは何か」ということである。ゴールがなければ、スプリント期間中にメンバーが団結して、心を一つにする明確な目的がなくなる。スプリントのゴールは、特定の問題を解決する一貫したユーザー機能セットを提供することである。それは、ステークホルダーたちのためのニーズに対処することかもしれない。または、プロダクトオーナーがスプリントから出てきたインクリメントによって検証したい仮説や非常に重要な仮説かもしれない。



スプリントプランニングでは、スプリントゴールの達成に必要な作業をプロダクトバックログから選択する

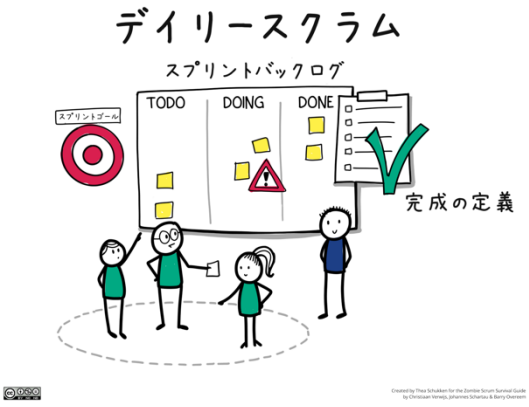
プロダクトオーナーは、目標を念頭においてスプリントプランニングに参加するのがよい。スクラムチームは、スプリントのタイムボックス内で、価値があり実現可能なゴールを洗練するために協力する。開発者は、プロダクトオーナーと協力して、スプリントゴールを達成するためにスプリント内で実施すべき作業をプロダクトバックログから選択する。必要に応じて、プロダクトバックログのアイテムの並び替えを行う。ここで選択したアイテムがスプリントバックログになる。開発者が実際に作業を行うため、開発者がスプリントバックログと最終的な決定権に責任を持つ。

1 ヶ月スプリントの場合、スプリントプランニングは 8 時間以内にすべきである。しかし、スプリント期間が短ければ短いほど時間がかからなくなる。スプリントプランニングは、スプリントの大まかな概要と最初の数日分のより詳細な計画があれば完了する。通常、この計画ではスプリントの最初の数日間の作業を分解しておく。スプリントが進むにつれて、開発者は、（スプリントゴールとスプリントバックログで設定した）大まかな大筋をどう協力し

て作業するかといったより具体的な計画に変換していく。少なくとも、デイリースクラムではこれを行う 1 つの機会である。

デイリースクラム

デイリースクラムは、24 時間ごとに実施する。開発者は、単一のスプリント内でも複雑な作業を行うことができる。開発者は、スプリントバックログを通じて透明化されたスプリントゴールの達成に向けた作業の進捗を検査し、それに応じて調整を行う。例えば、開発者が密接に連携しなければならない予期せぬ問題に遭遇したり、スプリントゴールを達成するために必要な作業をスプリントバックログに追加すべきだと気付いたりすることがある。開発者は、スプリントゴールの達成を妨げる問題や自分たちの解決能力を超えた問題に直面することがある。



デイリースクラムでは、開発者がこれからの 24 時間でスプリントゴールの達成に協力して計画する

デイリースクラムの所要時間は 15 分以内である。これからの 24 時間の協働作業を調整するための短く最小限の機会である。より多くの調整が有効であれば、開発者は 1 日を通して調整を行うことができる。スクラムフレームワークでは、デイリースクラムを効果的に行う方法は規定していない。開発者には、自分たちにとって最善なデイリースクラムの方法を見つけることが推奨されている。

スプリントレビュー

スプリントレビューは、スプリントの終盤においてスプリントレトロスペクティブの前に開催する。その目的は、これまでの作業を検査し、そこから学んだことに基づいて次のステップが意味をなすかどうかを意思決定する。スプリントレビューは、スプリント期間中にプロダクトを開発している人たちとその関係者たちが、スプリントの成果を検査するために集まる数少ない機会である。現在の市場状況、組織の変化、予算やスケジュールと共に、次のステップを一緒に決めていく。

スプリントレビューの目的は、これまでの作業を検査し、そこから学んだことに基づいて次のステップが意味をなすかどうかを意思決定すること

スプリントレビューは、1 ヶ月スプリントでは、4 時間以内にすべきである。しかし、スプリント期間が短ければ短いほど時間がかからなくなる。スプリントレビューの結果は、学んだ内容に基づくプロダクトバックログの調整という形で現れる。これには、スプリントレビュー中に現れた新しいアイデア、発見したバグ、プロダクトバックログに既にあるアイテムへの変更、プロダクトバックログ自体の並び替えを含む。ある意味で、スプリントレビューとは、「このスプリントで学んだことに基づいて、次のステップは何か？」という質問に回

答することである。この答えは、スプリントプランニングと今後のスプリントのためのスプリントゴールの判断に貴重な情報を提供する。

スプリントレビューでは、開発者が行ったことを一方的にプレゼンテーションしても「検査」にはならない。実際に、インクリメントを検査するということは、共に試してみて、フィードバックをすることを意味している。私たちは、スプリントレビューを「デモ」ではなく、「フィードバックパーティー」と表現する。スプリントレビューは、開発者が行ったことをプロダクトオーナーが初めて見る機会にすべきではない。その代わりに、スプリントレビューは、プロダクトオーナーがステークホルダーを招待し、プロダクトを共に検査する繰り返し行う重要な機会である。

スプリントレトロスペクティブ

スプリントレトロスペクティブは、スプリントの最後に開催する。通常はスプリントレビューの直後に開催する。スクラムチームは全員参加する。その目的はスクラムチームがスプリントゴールの達成のためにどのように協力したのかを検査することと、次のスプリントで効果性と品質を高めるための具体的なステップを特定することである。スプリントレビューがプロダクトや作業内容に焦点を当てているのに対して、スプリントレトロスペクティブは、その作業がどのように行われたかのプロセスを検査することが目的である。

スプリントレトロスペクティブでは、スクラムチームがスプリントゴール達成のためにどう協力したか、次のスプリントで何を改善できるかを検査する

1 ヶ月スプリントの場合、スプリントレトロスペクティブは、3 時間以内にすべきである。しかし、スプリント期間が短ければ短いほど、そのタイムボックスは短くなる傾向がある。スクラムチームが、協働作業の検査から学んだことに基づき、より効果的に作業ができるように調整することを決断する場合がある。これには、完成の定義の更新や、新しいツールや技術の調査、ワーキングアグリーメントの変更、別チーム編成などを含む。少なくとも 1 つの実行可能な改善が次のスプリントのスプリントバックログに直結する。

プロダクトバックログリファインメント

プロダクトバックログのリファインメントについてはどうだろうか？『スクラムガイド』では、どこかで行うべきこととして言及されている。プロダクトバックログのリファインメントはイベントにならないのか、という問いについては、確かにスクラムフレームワークには欠かせないものであるが、他のイベントとは異なる。プロダクトバックログのリファインメントは、むしろ継続的な活動である。言葉遊びのように見えるかもしれないが、重要な違いがある。すべてはプロダクトバックログのリファインメントのさまざまな形式を理解するところから始めよう：

- 作業を始めるにはあまりにも不明確なプロダクトバックログのアイテムを明確にする。（ステークホルダーのために）アイテムを開発する人たちと直接行うべきである。

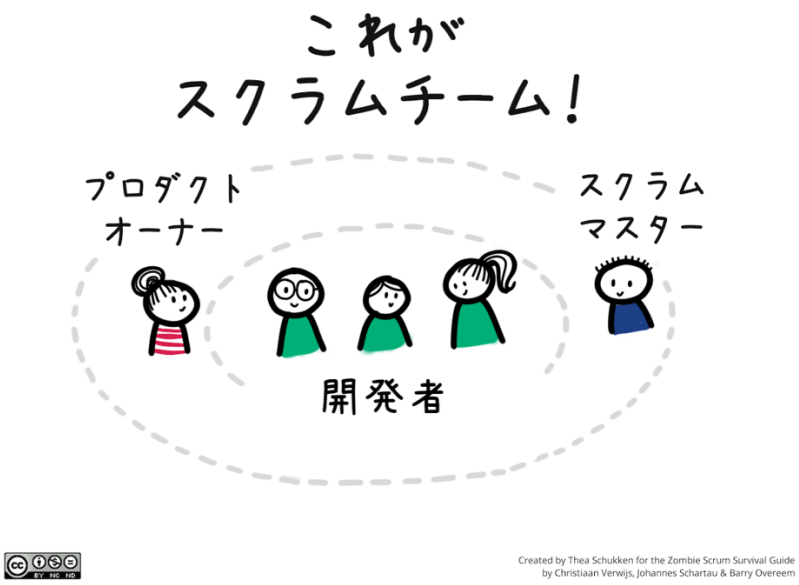
- 単一のスプリントで完成するには大きすぎるプロダクトバックログのアイテムを分解する（これはそれらがあまりにも不明確であることを意味していることがよくある）。
- 今後のスプリントを可能な限りスムーズで価値のあるものにするために、必要に応じてプロダクトバックログのアイテムを並び替える。
- 新しいインサイトによってプロダクトバックログにアイテムを追加したり、削除したりする。
- 特定のアイテムを実装する際の作業量を見積もる。これは、ストーリーポイントの割り当てのように「形式的」である必要はない。（スクラムでは任意のプラクティスであるが）T シャツのサイズ分けなどのサイジングのテクニックなど何でも構わない。直感（「やるべきことは十分わかっているし、スプリント内でやれそうだ」）でもよいだろう。

ある意味では、プロダクトバックログのアイテムは「今後に必要となる会話」を思い出させるものである。そして、リファインメントは、そのような会話をするための継続的なプロセスである。これは、ステークホルダーと次のスプリントで実施する可能性のあるアイテムについて話し合うことを意味することもあれば、チームが既に取り組んでいるアイテムを明確にすることを意味することもある。これらの会話の中には、スクラムチーム全体との会話もあれば、より小さなグループとの会話もある。リファインメントの中には、個別に行えるものもある。

そのため、プロダクトバックログのリファインメントは、スプリント期間中にチーム全員が参加する一度だけ行う形式化されたミーティングとして理解するのではなく、今後のスプリントのためにプロダクトバックログのアイテムを改訂するための一連の継続的な活動とするべきである。リファインメントは、さまざまな人たちによって構成される参加者が、さまざまな形式で行う継続的な活動である。そして、リファインメントの最善の方法を決定するのは、スクラムチームの責任である。

スクラムチーム（責任）

複雑な問題を解決するためには、スキルの高い専門家同士の協働が不可欠である。創造的で既成概念に囚われない解決策は、スキルの高い専門家の視点を取り入れることでより簡単に発見できる。この協働作業を促進することで、コミュニケーションと意思決定の複雑さを軽減するために、スクラムフレームワークは意図的に責任（あるいは「役割」）を 3 つに制限している。誰かが他の人に対して権限を持つという意味での階層的な役割ではない。これは経験的に仕事をするために最低限必要な異なる視点を、3 つの責任が持つという意味である。この 3 つの責任を組み合わせることで、どのような環境であっても経験的に十分に機能する。他の責任あるいは役割は必要ない（もしくは、邪魔になるだろう）。



異なる 3 つの視点（価値、品質、プロセス）のバランスをとる 3 つの責任

プロダクトオーナー

プロダクトオーナーには、プロダクトに対する目標設定に関して何に価値があるのか（何に価値がないのか）という視点が必要である。スクラムチームが、時間と費用をかけてプロダクトに取り組む間、プロダクトオーナーは、この投資がステークホルダーに価値を還元するものなのかを確認する。何に価値があり、何に価値がないのかを意思決定するには、開発者だけでなく、プロダクトに関係する人たちとの緊密な協力が重要である。

1 つのプロダクトには、1 人のプロダクトオーナーと、1 つのプロダクトゴールを持つ 1 つのプロダクトバックログがある。意思決定を迅速に行い、迅速に適応するためには、プロダクトオーナーがプロダクトに対して完全な権限を必要とする。プロダクトオーナーは、「プロダクトに対する目標設定が何であるか」、「プロダクトバックログに何を載せるか、何を載せないのか」、「予算をどのように使うか（もしくは、それを設定するかどうか）」について、最終的な決定権を持っている。

プロダクトオーナーは、プロダクトゴールと順序づけられたプロダクトバックログを確保し、スクラムチームとステークホルダーの両方にとって利用できるようにする。スクラムチームのメンバーの中でプロダクトオーナーだけがこの作業を行っているという意味ではない。各スプリントで開発者による作業の価値を最大化するためには、プロダクトオーナーが開発者と積極的に協力して、アイテムを書いたり、アイテムを改訂したり、アイテムの並び替えをしたりすることが理にかなっている。協働作業が大事なのである。

開発者

開発者は、スプリントゴールを達成するためのインクリメントの提供に積極的に貢献するスクラムチームのメンバー全員のことである。彼らがどのような種類の作業をしているのか、正式な肩書きが何かは関係なく、スクラムフレームワークでは全員を「開発者」とする。開

発者は、プロダクトの目標設定を実現し、高い品質を維持するために必要な作業をどのように行うのかという視点を共に持つ。

プロダクト開発は複雑な作業である。そのため、単一のスプリント内という近い将来であっても予測することは困難である。インクリメントを提供するチームの能力を妨害するような課題や、挑戦、問題が出てくる可能性は十分にある。また、インクリメントに何を含めるべきか、何を含めるべきでないかについて、スプリント期間中に新しいアイデアが出てくる可能性もある。単一のスプリントであっても予測不可能なこの性質には、開発者がどのように作業をし、どのように組織化するかについて、3 つの重要な要件がある。

第一の要件は、チームの一員でないが完成したインクリメントを提供するために頻繁に必要とされるような他の人や部門、スキルへの依存度を最小限に抑える努力をすることである。それぞれの依存関係はチームが制御するには限度がある。チームは、毎回のスプリントで、完成したインクリメントを提供する能力を持つべきであるが、依存関係はこの能力をスローダウンさせたり、完全にブロックさせたりしてしまう。このことは、チームの経験的に作業する能力に悪影響を与える。依存関係は、（テストやデプロイの）自動化、新しいスキルのトレーニング、または、必要なスキルを持った人のスクラムチームへの加入など、さまざまな方法で最小限にできる。

第二の要件は、開発者がチーム内でのスキルを分散させることによってボトルネックを軽減することである。例えば、スクラムチームに 1 人のテスト担当者が入ることは、1 人もいないよりもよいことであるが、テスト担当者が作業に追われていると結果としてチーム全体のスピードが低下してしまう。同じことがバックエンドの開発や設計と分析などでも当てはまる。チームは、チーム内で 1 人以上のメンバーが特定の種類の作業をこなせるように取り組む必要がある。ここでの目標は、誰もが同じようにその作業をできるようにすることではない（これは明らかに非現実的で、何年もかけて身に付けたスキルにとって失礼である）。必要に応じて他の人が手助けをしたり、引き継いだりできるようにすることである。スクラムフレームワークでは、「スクラムチームは機能横断的である」という記述でまとめている。

第三の要件は、開発者がスプリント期間中に起こるであろう多くの予想外の事態を乗り越えるために、知性と創造性に頼らなければならないことが多いということである。そのため、スクラムチームは、どのように作業を行い、作業を完了するためにどのように改善するかを意思決定できるように自己管理すべきである。自己管理型のチームでは、チームのためにこれらの意思決定を行うような「ボス（上司）」は存在しない。作業をしているすべての人たち（それは開発者を意味する）は、スクラムフレームワークの境界の中で意思決定を行うために協力する。

スクラムマスター

スクラムマスターには、経験的プロセス制御の視点とスクラムチームの内外において、透明性、検査、適応を形成する資質が必要である。スクラムマスターは、スクラムフレームワークの要素をチームやより広範囲な組織で実現するために存在する。そのため、スクラムマスターは自身が置かれている状況に応じて、いくつかのスタンスを取り入れている：

- **ティーチャー：** 経験的な仕事をするための手段としてスクラムフレームワークの目的を説明する。作成物、イベント、責任、そして原則がどのように経験主義とアジリティを促進するかを誰もが理解できように努める。
- **ファシリテーター：** スクラムフレームワークの透明性、検査、適応の機会が最大化されるように、スクラムフレームワークの 3 本柱を促進する。その一例が、スクラムイベントのファシリテーションである。ファシリテーションは、スクラムチームからの要望や必要に応じて実施する。もう一つの例が、プロダクトオーナーがプロダクトバックログとステークホルダーをうまくやり繰りするためのテクニックを見つけたり、使ったりするのを手助けすることである。
- **妨害要因リムーバー：** スクラムチームがスプリントゴールを達成するのを妨げる問題を取り除く（もしくは手助けする）。スクラムマスターは、スクラムチームが自ら問題を解決する能力を高めることを支援する。これはチームが学ばなければならないことであり、スクラムマスターはそれを支援するということである。最初のスプリントでは妨害要因と考えられるものであっても、もっと後のスプリントではチームが自ら簡単に解決できる問題になっている可能性がある。
- **チェンジエージェント：** 経験主義とアジリティを阻害するようなスクラムチームの環境の妨害要因を取り除く手助けをする。例えば、「テスト」と「デプロイ」を別のチームや部門として分けている組織がある。あるいは、チームではなく、個人ごとに報酬を与えるべきという人事のプラクティスがある。これらの妨害要因を取り除くことは、スクラムマスターが自分 1 人でできることではない。そのため、他のスクラムマスターやプロダクトオーナーなどと協力することになる。
- **コーチ＆メンター：** スクラムチームに対して強力な自由回答を促す質問をすることにより、スクラムチームを指導する。スクラムマスターは、他のスクラムマスターのメンターでもある。また、スクラムチームのメンバーが自分たちを手助けできるだけの経験とスキルを持ち合わせたメンターを見つける手助けをする。

スクラムマスターは、真のリーダーである。自分自身に注目を集めるのではなく、他の人たちが可能な限り効果的になるよう手助けをする。何をすべきか、どのようにすべきかを指示してチームを管理したり、リードしたりすることはない。スクラムマスターが強い立場を取るべきなのは、経験的なプロセスやチームメンバーが対人関係におけるリスクを負っていても安全だと感じられる度合いに悪影響を及ぼす意思決定がなされた場合だけである。

スクラムチームとして協力

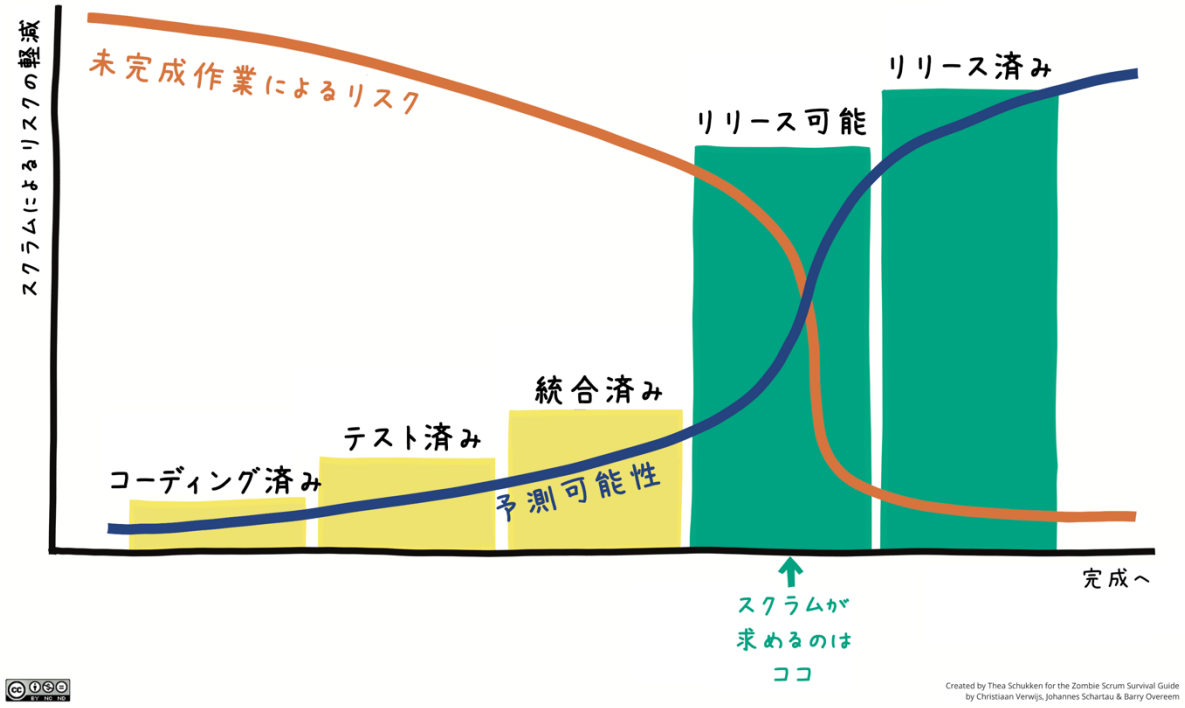
3 つの責任を組み合わせてスクラムチームが成り立つ。スクラムチームは、自分たちのプロダクトに関するすべての意思決定権を持っている。プロダクトオーナーは、ステークホルダーにとっての価値を最大化するために、利用可能な予算をどのように使うかを意思決定する。開発者は、そのプロダクトをどのように作るかを意思決定し、そのための必要なすべてのスキルを持ち合わせている。そして、スクラムマスターは、これらが経験主義を最大化する方法で行われることを保証する。他の役割は必要ない。

スクラムフレームワークは、「複雑さに直面してもシンプルさが私たちの選ぶ道である」という Gunther Verheyen 氏[2]の名言を見事に捉えた形で設計されている。自分たちの組織は違うと思っていると、責任（役割）や、ルール、プラクティス、構造をスクラムフレームワークに追加したくなるかもしれない。状況によってはそうする必要があり得るかもしれないが、複雑な環境では、物事をできるだけ簡素に維持することに焦点を当てることが最善の方法である。これはスケーリングにも当てはまる。チームを増やすと複雑さも増加する。その代わりに、より少ない人数でより多くのことを行う方法を模索したい。

[2] https://www.gunterverheyen.com/

2 つの推進原則

スクラムフレームワークの特徴を述べるとき、よくスクラムフレームワークのイベントと責任を説明することがある。これらの要素は確かに重要であるが、その効果性はチームが 2 つの推進原則をどれだけ理解しているかによって決まる。



完成したインクリメントを提供するスクラムチームの能力が図の右側に向かうほど、
未完成の作業のリスクを軽減し、予測可能性を向上できる（Gunther Verheyen の業績からヒントを得た）

スプリントごとに完成したインクリメントを提供する

スクラムフレームワークの目的を 1 つの文章にまとめるとすれば、『スプリントごとに完成したインクリメントを少なくとも 1 つ提供することで、経験的に作業をすること』になるだろう。これは、スクラムチームが、完成したインクリメントを開発する上で何が関係しているのかを時間をかけて明確にすることが重要な理由でもある。例えば、「これを行うためにはどんな作業が必要か？」、「社内の品質ガイドラインに準拠するためにはどのようなチェックとテストが必要か？」、「誰がこれに関与するべきか？」などである。このような共通認識を完成の定義と呼ぶ。完成の定義はよくチェックリストの形式をとる。

プロダクトバックログのアイテムの作業は、完了してユーザーの手に渡るまでは、仮説が詰まっている傾向がある。例えば、「このアイテムを実装することでユーザー体験は向上するのか？」、「ユーザーはその仕組みを理解できるのか？」、「性能は十分なのか？」などである。他にも、「このアイテムをテストしてデプロイするのはどれくらい簡単なのか？」、「このアイテムの作業中にどんな問題にぶつかるだろうか？」、「何を開発すべきかをどれくらい理解しているか？」といったように、アイテムを実装するために必要な作業についての仮説も出てくる。

すべての仮説はリスクを表す。例えば、ユーザーが要求した内容を誤解しそれを修正するために手戻りするリスク、コードを書き始める際に技術的な限界に陥るリスク、ユーザー機能が期待よりも悪くなるリスク、自動テストを書くことがはるかに困難であることが分かるリスク、あるいは、結局使われないユーザー機能にお金と時間を費やしているリスクなどである。これらのリスクに共通しているのは、どこかで予期せぬ作業が発生するということである。この種の「持ち越し」作業は思いがけないときまで現れない傾向にあり、そのスプリントで取り組んでいる作業を台無しにしてしまう傾向がある。これを「未完成（Un-done)の作業」と呼ぶ。

未完成の作業のリスクを防ぐ最善の方法は、各スプリントでリリース判断可能な完成したインクリメントという結果を（少なくとも 1 度は）確実に出すことである。これは、「インクリメントが完全にテストされて、動作していること」、「テキストとビジュアルが最終段階になっていること」、「文書化とデプロイパッケージが最新になっていること」、「パフォーマンスとセキュリティが組織の基準に適合していること」、そして「ここから、ボタンを押すだけでインクリメントをユーザーにリリースできること」を意味する。この時点から、今後のスプリントの焦点と予測可能性を台無しにするような未完成の作業がほとんどないか、全くないことが分かる。さらに重要なことは、インクリメントをリリースすることで、プロダクトがその目標設定に近づいているかどうかを確証なく決めてかかるのではなく実際に確認することで、経験的プロセス制御を実行できる。

スクラムチームの完成したインクリメントを提供する能力を右側に移動することにより、未完成の作業によるリスクが減少する。さらに、予測可能性は高まる[3]。これは Gunther Verheyen の業績からヒントを得ている。スプリントごとに完成したインクリメントを開発することは確かに難しいことである。しかし、意図的にそうするのである。プロダクトを開発する仕組みにプレッシャーをかけることで（これを達成するために必要なすべての作業を行うことで）、どこに改善が必要であるかが明確になるからである。例えば、スキルやツール、技術が不足しているところはどこか、官僚主義が邪魔をしているところはどこか、実際に経験的に作業を行い複雑な作業のリスクを軽減するために取り除く必要がある妨害要因はどこか、である。スプリントごとに少なくとも 1 回は完成したインクリメントを提供することにレーザーの焦点を当てることで、仕組みが適切なところで痛み始める。そして検査と適応の機会を作り出す。

結束力を高めるために共通のプロダクトゴールとスプリントゴールを用いる

『スクラムガイド』では、「ゴール」について 40 回も言及している。これはスクラムフレームワークのどの要素よりも多い。さて、私たちは、スクラムを用いた仕事の進め方についての意思決定のためにこのような解釈を使いたいわけではない。しかしこのことは、共通ゴールの重要性について何かを教えてくれるはずである。

プロダクトゴールとスプリントゴールが重要な理由は、スクラムフレームワークが複雑な問題をより効果的に克服するためとリスクを軽減するために設計されているからである。プロ

[3] 訳註: 上のグラフを参照のこと

ダクトゴールとスプリントゴールは、この複雑さを 3 つの補完的な方法で克服するのに役立つ。

1 つ目は、スクラムチームが何に時間をかけるかを意思決定するときに、プロダクトゴールとスプリントゴールがスクラムチームに指針を提示してくれることである。スプリントゴールを達成するためには、スプリントバックログにある作業の方が他の作業より重要であるはずである。2 つ目は、これによってチームが重要なことに集中できるようになることである。状況が困難になったときに、チームはある作業を手放し、他の作業を優先することができる。または、時間が許せば、スプリント期間中でもゴールを達成するために作業を追加することもできる。3 つ目は、スプリントゴールによって、スクラムチームは明確で共有された目的を持つことができ、分業ではなく、協働作業を促進でき、自己組織化[4]することができる。協働により、複雑な問題を解決する時に必要な既成概念に囚われない思考とチームの精神を可能にする。

プロダクトゴールとスプリントゴールもまた、さまざまなスクラムイベントに色合いと目的を与え、それらを単に作業のためのミーティングにとどまらないものにしている。スプリントプランニングでは、これからのスプリントのゴールを決定し、そのために必要な作業を選択する。デイリースクラムでは、開発者がこのゴールに向かって、今後の 24 時間をどのように作業するのか、それに対する妨害要因は何かを中心に行う。スプリントレビューでは、スプリントゴールの結果がプロダクトゴールとどう関係しているかをステークホルダーと一緒に検証し、次に注力すべきゴールを明確にしていくことが目的となる。スプリントレトロスペクティブは、スプリントゴールを達成するために、より効果的に協力する方法を見つけるものである。

しかし、プロダクトゴールとスプリントゴールはスクラムフレームワークの 3 つの責任にも焦点を当てている。スプリントゴールは、開発者に何を中心に自己組織化すべきかの指針と方向性を与えている。プロダクトオーナーは、プロダクトに対する目標設定とそれをどのようにさまざまなスプリントの目標に変換するかに集中することができる。これにより、それをどのように実装するかといった実装方法の詳細に関与せずに済む。最後にスクラムマスターは、プロダクトゴールとスプリントゴールを炭鉱のカナリアのように使うことで、経験的プロセス制御の効果性を高めることができる。チームがプロダクトゴールとスプリントゴールの設定に苦労している場合、スクラムマスターがシャーロック・ホームズの帽子を被り、何が原因なのかを探りに行くよい理由になる。例えば、チームの規模が大きすぎたり、小さすぎたりで、特定のスキルや人が不足することがある。または、プロダクトオーナーには、権限やビジョンを持ち合わせていない場合がある、もしくはリファインメントが行われていない場合がある。

ここでプロダクトごとにプロダクトゴールが 1 つ、スプリントごとにスプリントゴールが 1 つしかない理由を明確にしなければならない。複数のゴールは、焦点を混乱させ、確約を減

[4] 訳註:「自己組織化」としている箇所は、著者の意図でこの表現を採用している

らし、透明性を制限するだけである。もちろん、以前のゴールが達成されたり、放棄されたりしたときには、新たなゴールを立てることができる。

簡単に言うと、複雑な作業において、プロダクトゴールとスプリントゴールのような共通ゴールは、濃霧の中で港にたどり着くための灯台である。それがないと航路に迷ったり、座礁してしまったりする可能性が高くなる。

経験主義を可能にする 5 つの価値基準

ここまでは、スクラムフレームワークの仕組みと原則について説明してきた。経験的プロセス制御を可能にするための仕組みと原則には驚くべき効果があるが、それを手助けする振る舞いがなければ大きな違いを作ることはできない。スクラムチームが直面している技術的な課題について公開することを恐れ、代わりに「大丈夫だ、問題ない」と言ってしまうのでは、スプリントレビューでの検査はどれだけ正直なものになるだろうか？プロダクトバックログの目標設定に合わないアイテムをプロダクトオーナーが断ったり、拒否したりすることを恐れている場合、スクラムチームはどれだけ効果的になれるのだろうか？最新かつ煌びやかな技術に気を散らし続けているスクラムチームがどれだけの価値を生み出せるのだろうか？チームの知識や承諾なしにチームの構成を変えることは、チームとして働くことへの意欲にどのような影響が出るのだろうか？

経験的な仕事に慣れていない組織では、経験的な作業をすることは特に難しい。チームとそのステークホルダーに意思決定の原動力となる何かを与えるために、スクラムフレームワークでは 5 つの基準となる価値（価値基準）を具体的に強調している。どのような意思決定に直面したとしても、チームは自分たちの選択肢が 5 つの価値基準にどのような影響を与えるのかを自問自答することができる。

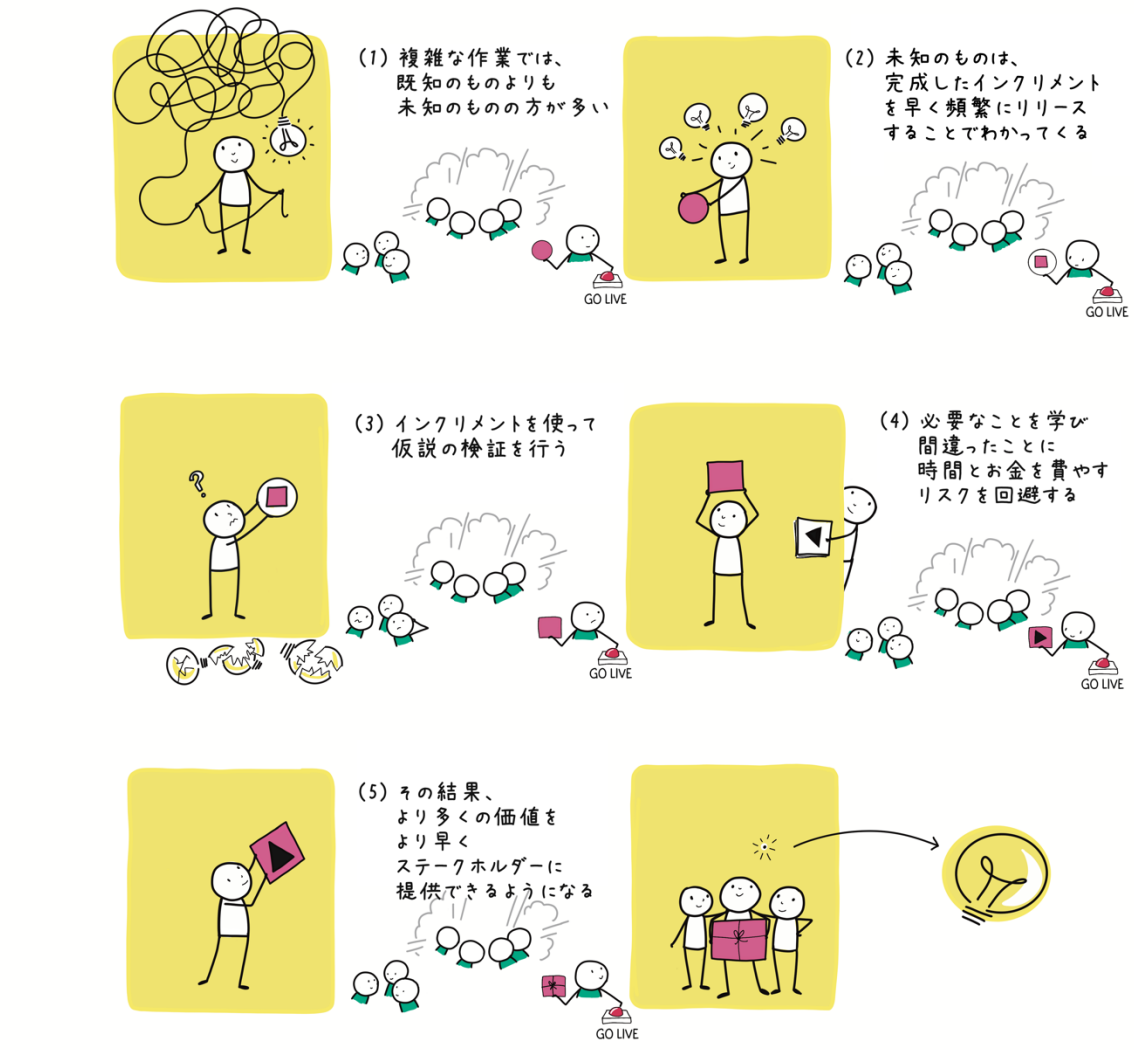
- **公開：** 物事がどのように進んでいるかを公開しなければならない。何がうまくいっているのか？何がうまくいっていないのか？課題がどこにあり、機会はどこにあるのか？
- **勇気：** 正しいことをする勇気を持たなければならない。経験的なプロセスを妨げるものには「いいえ」と言わなければならない。困難な課題に一緒に取り組むことで勇気を発揮しなければならない。わからないことについては、質問したり、フィードバックをしたりしなければならない。質問をして、わからないことや不確かなことは認めなければならない。
- **集中：** スプリントゴールとスクラムチームの目標に集中しなければならない。集中し続けることができる環境を作らなければならない。
- **尊敬：** スクラムチームのメンバーのスキル、専門知識、知性を尊敬しなければならない。複雑な問題に対して自己組織化するスクラムチームの能力を信頼しなければならない。そして複雑な作業についてくる不確実性を尊重しなければならない。
- **確約：** スプリントゴールに向かってチームとして協力することを個人的に確約できる環境を作らなければならない。

これらの価値基準を日々の振る舞いに取り入れることを学ぶことは、生涯の旅となる。チームの能力が高くなればなるほど、頻繁な透明性、検査、適応がより効果的に行われるように

なる。朗報がひとつある。スクラムフレームワークは、これらの価値基準を学び、成長するための優れた新境地を提供してくれる。

シンプルなフレームワークを共に

スクラムは、複雑で適応性のある問題を克服するためのシンプルなフレームワークである。スクラムは、チームが経験的に作業をするために何をすべきかを規定しているだけで、それをどのようにすべきかを規定していない。すべてのチーム、すべてのプロダクト、すべての組織は異なるからである。そのため、チームは自分たちに合った方法を見つけなければならない。そうすれば、複雑な作業のリスクを軽減し、ステークホルダーに価値を早期に提供し始め、対応力を高めることができる。この道のりは、スクラムチームによっては簡単であり、他のチームにとってはそうではないが、やり続けることで変化していく。



複雑な作業のリスクを軽減し、早期に価値を提供するためのフレームワーク

◈ ◈ ◈

すべてのイラストは、The Liberators のために、Thea Schukke が描いたものである。

日本語訳は、長沢智治が担当した。

初期版の日本語訳レビューは以下の方々にお願いした：
秋元利春さん、伊藤宏幸さん、川北英治さん、高橋芽里さん、中原慶さん

翻訳に関する連絡先: 長沢智治 (nagasawa@servantworks.co.jp)