

スクラム 入門

Pete Deemer

Scrum Training Institute (ScrumTI.com)

Gabrielle Benefield

Scrum Training Institute (ScrumTI.com)

Craig Larman

craigarman.com

Bas Vodde

Odd-e.com

version 1.2

読者の皆様へ：多くのスクラムに関する簡潔な説明資料はオンラインで入手可能ですが、本書はより詳細な実践的手法を提供しますが、スクラムに関する学習の最終段階ではありません。スクラムの導入を検討しているチームには、Ken Schwaber の Agile Project Management with Scrum ([和書]スクラム入門-アジャイルプロジェクトマネジメント) または Agile Software Development with Scrum ([和書]アジャイルソフトウェア開発スクラム) をお勧めします。また、多くの素晴らしいスクラムのトレーニングやコーチング等のオプションもありますので、うまく活用して下さい。全詳細は、scrumalliance.org にあります。惜しみない情報提供者である、Ken Schwaber、Jeff Sutherland 博士、Mike Cohn に感謝いたします。

© 2009 Pete Deemer, Gabrielle Benefield, Craig Larman, Bas Vodde

訳者より読者の皆様へ：本翻訳は、日本のスクラムコミュニティ
“すくすくスクラム” <http://www.sukusuku-scrum.jp/> の協力を得て作りました。
すくすくスクラムは、スクラムのプラクティスを中心に現場改善を探求する勉強会です。
惜しみない協力を頂いた“すくすくスクラム”に感謝いたします。

© 2010 Kazumasa EBATA (ebacky)

従来のソフトウェア開発

大企業から小企業まで利用されている従来のソフトウェアを構築する方法は、「ウォーターフォール」として一般的に知られている逐次的な製造工程でした。（V-モデルなどの）多くの改良型がありますが、それらは通常、詳細設計の段階から着手します。そこでは、最終製品を慎重に考え抜いた設計が極めて詳細に文書化されます。設計を実現するために必要なタスクが決定され、進捗管理はガントチャートや Microsoft Project のようなツールが使われます。チームは、個別タスクの詳細な見積りの合計をもとに開発期間を見積ります。一度、利害関係者が十分に計画のレビューを行い、承認されてから、チームは仕事にとりかかります。チームメンバーは、生産ライン形式で専門化された仕事を仕上げ、次の担当者に渡します。いったん仕事が完了すれば、製品を検査機関（品質保証：QA）に引き渡します。検査機関は、出荷前に検査を完了します。このプロセスの至るところで、厳密な管理が行われます。この管理とは、設計された通り製品が生産されているかどうかを計画が守られているかどうかで判断します。

この方法には、長所と短所があります。長所は、最も論理的なことです。構築する前に検討し、全てを書いて伝え、計画に従い、全てのことができるだけ整理されています。大きな短所は、たった1つ。人が関わっていることです。

例えば、この方法では、全ての素晴らしいアイデアは最初に求められ、計画に組み込まなければなりません。言うまでもなく、素晴らしいアイデアは最初、途中、出荷前日等、プロセスを通じて浮かびます。また、変化を容認できないプロセスは、イノベーションを抑制します。ウォーターフォールでは、後半に提案される素晴らしいアイデアは、良いモノではなく、危険なモノと思われれます。

ウォーターフォールでは、重要な情報伝達の方法で非常に重要視しているのが文書化です。妥当な想定として、もし考えていることを出来る限りドキュメントに書き留めることができれば、チームメンバーに確実に伝えることができます。さらに、ドキュメントに書けば、仕事をした明確な証拠になります。実際、50ページにわたる非常に詳細な要求仕様書は、多くの場合、読まれることはありません。それらが読まれる時、誤解はよく妥協されます。ドキュメントは不完全です。そのため、ドキュメントを読んだ時に（当時）考えたことが理解できず、別の抽象的な考え方を創造します。深刻な誤解が生じたとしても当然です。

別の現象として、初めて動作する製品を使った時に、なるほど！と気づき、すぐに改善できる20の方法を考えます。多くの場合、変化が最も難しく混乱を起こす開発終盤に有益な知見に気づきます。つまり、従来の方法では、当然のことをすると高額になります。

未来は予想できません。例えば、競争相手がどのような発表をするか予想できません。予期せぬ技術的な問題によって、突然、方向転換します。その上、将来の不確かなモノを計画するのは、特に下手です。- 8ヶ月後の1週間をどのように過ごすかを推測することは、ちょっとした幻想です。多くの手間暇をかけて構築されたガントチャートが破綻されるように。

逐次的な工程は、仕事を受け渡す人達の対立関係を助長する傾向があります。「彼は、仕様のないモノを構築するように頼んでいます。」「彼女は、気が変わっています。」「管理しないモノに責任は持てません。」

そして、逐次的な開発は面白くありません。ウォーターフォールモデルは、開発者の大きな苦痛の要因です。成果物は、開発者の創造性、技術、情熱には及びません。私達はロボットではありません。なので、ロボットのような行動を求めるプロセスは、不幸をもたらします。

柔軟性がなく変化を好まないプロセスは、あまり良くない製品を生産します。顧客は、（少なくとも、2つのステップをやめたとしたら）最初に要求した製品を得られるかもしれませんが、それは本当に求めている製品でしょうか。新しい学びや気づきを得たとしても、事前に全ての要求を集め、確定された製品は、初期のアイデアと同じくらい良くないと非難されます。

多くの逐次的な工程の実践者は、これらの苦痛を何度も経験しますが、最も論理的に思えます。よく見られる反応は、「改善すれば、より良くなります。」「より計画し、より文書化し、より変化に耐えていたら、全てのことが順調に進みます。」と内省に向かいます。残念なことに、多くのチームは反対のこと“より一生懸命にしようと努力すると、より悪くなる！”に気づきます。経営陣は、ウォーターフォールモデルを高く評価し多くのリソースを投資しました。根本的に異なるモデルへの変化は、明らかにミスを犯したことを認めることになります。そして、スクラムは根本的に異なります…

アジャイル開発とスクラム

アジャイル開発手法は、“商品開発における学び、イノベーション、変化等、より実践に基づいた方法にすれば、より良い結果をもたらす” という信念によって生まれました。アジャイルの原則は、事前に仕様を書く多くの時間を費やすことよりも、すぐに出荷できる動く製品（ソフトウェア等）を重視します。アジャイル開発は、大きな組織階層や機能による細分化よりも、決定権が与えられた職能上の枠組みを越えたチームにし、継続的な顧客への情報提供を短期間で繰り返すことを重視します。よくアジャイル開発やスクラムを学ぶと、初心に戻ったように「そうすれば良かったのか。」と気付くことがあります。

アジャイルの中で、スクラムが圧倒的に人気があります。それは、1986年のハーバード・ビジネス・レビューの「成功を収めた製品の開発グループの事例」に強く影響を受けました。この論文で「ラグビー」という用語が紹介されました。それが現在の「スクラム」に変わりました。成功する製品開発の素晴らしい問題解決手法（1991, DeGrace and Stahl）は、ラグビーの自己組織化したチームのように、製品開発の領域を掘り下げます。1993年、Ken Schwaber と Jeff Sutherland 博士によって、形式化されました。今日では、スクラムは企業の大小に関わらず利用されています。

Yahoo!、Microsoft、Google、Lockheed Martin、Motorola、SAP、Cisco、GE、CapitalOne、アメリカ連邦準備銀行等が含まれます。スクラムを利用している多くのチームは、多くの状況において全面的な変革、生産力とやる気において顕著な改善を公表しています。“場当たりの解を探し出して来て問題を解決するような手法”によって、ひどい目に遭ってきた製品開発者にとって重要です。スクラムは、単純で良い効果を発揮します。

スクラムの概要

スクラムは、企画や製品開発もしくはアプリケーション開発を繰り返しながら漸増するフレームワークです。スプリントと呼ばれる周期で、開発を構成します。これらの繰り返し期間（スプリント）は1カ月未満で、途切れることなく連続します。スプリントはタイムボックス化されます。作業完了の有無に関わらず、具体的な期日で終了します。決して延長されません。

各スプリントの始めに、職能上の枠を超えたチームは優先されるリストから（顧客の要求）項目を選びます。チームは、スプリント終了までに選択した全ての項目を完了させることを公約します。スプリントの間は、選ばれた項目は変更しません。毎日チームは集まって、手短に進捗を確認し、残りの仕事を終わらせるのに必要な次の作業を調整します。スプリント終了時に、チームは利害関係者と共に、構築した製品を実際に用いてスプリントを検査します。参加者は、次のスプリントで具現化できるフィードバックを得ます。スプリント終了時に、本当に価値のある製品が「実現」されることを強調します。ソフトウェアの場合、統合されたコードが十分にテストされていて、出荷可能な状態であることを意味します。図1に、重要な役割、技能によって作り出したもの（アーチファクト）、イベント等を示します。

スクラムの大きなテーマは、「検査と適応」です。開発には、必然的な学び、イノベーション、予期しない驚きがあります。スクラムは開発工程を短くし、成果物と実務の有効性を検査します。そして、プロダクトのゴールとプロセスを適応します。これを継続的に繰り返します。

スクラム



図1：スクラム

スクラムの役割

スクラムには、3つの役割があります。プロダクトオーナー、チーム、そして、スクラムマスターです。同時に、これらはスクラムチームとして知られています。プロダクトオーナーは、製品特性を特定することによって、投資収益率（ROI）を最大化することに責任があります。最優先リストの優先順位付けを継続的に見直し、次のスプリントのために、どの（顧客の要求）項目が上位にあるべきかを決めて、リストを洗練します。

プロダクトオーナーは商用製品に対して、（製造途中の）製品の損益に責任があります。内製のアプリケーションに対しては、（収入が得られる）商用製品という意味では、投資収益率（ROI）に責任はありません。各スプリントの「ビジネス価値の最大化、低コスト化」を選択するという意味では、投資収益率（ROI）を最大化することに責任があります。実際問題として、「価値」とは曖昧な表現です。また、優先順位付けは、顧客要求、戦略的な目的、リスク対応、改良、他の要因などに影響を受けます。プロダクトオーナーと顧客が同じということもあります。これは、内製のアプリケーションではよくあります。一方、顧客が多様なニーズを持つ何百万人という場合では、プロダクトオーナーは、多くの製品開発の組織における製品管理者か製品販売管理者と同じような役割になります。しかしながら、プロダクトオーナーは従来の製品管理者とは少し異なります。なぜならば、開発権限をプロジェクト管理者に委ねのではなく、自らスプリント毎に優先順位をつけ、結果を検査するなど、チームと積極的かつ頻繁に交流します。スクラムでは、最終的な権限を持つプロダクトオーナーは、必ず一人であり、仕事の価値に対して責任を持つことに注意しなければなりません。

チームは、アプリケーションやウェブサイト等、プロダクトオーナーが示唆する製品を造ります。スクラムにおけるチームとは、職能上の枠を超えて、各スプリントで出荷可能な製品を納品するために必要な全ての専門技能を持っています。また、“自己組織（自己管理）”し、とても高い自律性と製品に対して責任があります。チームは、何をすべきであり、そのべきことを達成する最適な方法を決めます。スクラムのよくある話では、チームは“ブタ”、他者は“ニワトリ”として知られています。（これは、「ハムエッグ」というレストランを開店しようとするブタとニワトリの物語（冗談話）からきています。その物語の中でブタは「私は身を削るのに、ニワトリは少し関わるだけじゃないか。」と迷います。）

スクラムにおけるチームは、7人±2人です。ソフトウェア開発の場合、チームは、分析、開発、テスト、インターフェース設計、データベース設計、アーキテクチャ、書類作成等の能力を持った人で構成されます。チームはプロダクトオーナーに、製品開発と製品の価値向上に関するアイデアを提供します。スクラムにおけるチームは、全メンバーが、スプリント期間に1つの製品開発に100%集中することができれば、最も生産性が高く効果的です。なので、複数の企画や製品の仕事をすることを避けて下さい。安定したチームには高い生産力があるので、メンバー変更は避けて下さい。多人数の集団に適用する場合は、複数のスクラムチームで構成されます。綿密な調整を行い、各チームが製品の異なる特性に集中します。1つのチームが、徹底的な顧客中心思考で、全ての作業（計画、分析、プログラミング、テスト）を行います。チームは、フィーチャーチームと言われることもあります。

スクラムマスターは、製品開発チームがビジネス価値を実現するために、スクラムの学習と適応を支援します。スクラムマスターは、チームとプロダクトオーナーがうまくいくための支援者です。スクラムマスターは、チームの管理者でもプロジェクト管理者でもありません。むしろ、スクラムマスターは、外部からの干渉を守り、プロダクトオーナーとチームが、スクラムを最大限に活かせるように導きます。スクラムマスターは、（プロダクトオーナー、管理者を含む）全員に対し、スクラムの正しい理解と実践を支援します。また、しばしば、アジャイル開発で成功することを義務付けられる組織の困難な変化を牽引します。スクラムは、多くの障害、チームの危険、プロダクトオーナーの有効性を明らかにします。それは、精力的に働くスクラムマスターが問題解決を手助けるか、チームやプロダクトオーナーが成功することは難しいと気付けるようにすることが重要です。専任の献身的なスクラムマスターは必要ですが、小さいチームでは、チームメンバーが役割を果たせるかもしれません（この場合、通常業務の負担を軽減します）。どのような経歴もしくは、工学、デザイン、テスト、製品管理、プロジェクト管理、品質管理等の能力をを持っていても、優秀なスクラムマスターになれます。

スクラムマスターとプロダクトオーナーは、同じ人になることはできません。時には、スクラムマスターがプロダクトオーナーの依頼を断ることもしなければならないからです（例えば、スプリントの途中で、新たな成果物の発表）。また、スクラムマスターはプロジェクト管理者とは異なります。スクラムマスターは、何をすべきか指示したり、タスクを割り当てたりしません。彼らは、工程を促進したり、チームの自己組織化と自己管理化を支援します。もし、スクラムマスターが以前はチーム管理者だった場合、スクラムで成功させるためには、考え方や交流スタイルを大きく変えなければなりません。

スクラムには、製品管理者の役割がないことに注意して下さい。なぜなら、必要ないからです。従来の製品管理者の責任は、スクラムの3つの役割で分担します。時々、（元）製品管理者がスクラムマスターの役割をすることがありますが、成功するのは難しいです。なぜなら、2つの役割には、（日々の責任と成功しなければならないという考え方等）、根本的相違があります。スクラムマスターの役割をしっかりと理解したり、成功するために必要な核となる開発技術を習得する良い方法は、スクラムアライアンスの認定スクラムマスター研修です。（日本での研修は、Odd-e Japan が研修を行っています。お問い合わせは、[nintei\(@\)odd-e.com](mailto:nintei(@)odd-e.com) まで）

3つの役割に加え、製品を成功させる貢献者には機能的な管理者（技術管理者等）もいます。スクラムの役割が変化する間、とても貴重な役割です。例えば

- スクラムの規律と精神を尊重することによって、チームを支援します。
- チームとプロダクトオーナーの障害を取り除くことを支援します。
- 専門技術と経験を活かせるようにします。

スクラムでは、以前の“子守”としての役割（分担、現状報告、その他のマイクロマネジメント形式等）をやめ、チームの“奉仕者”と“リーダー”の役割（指導、コーチング、障害除去支援、問題解決支援、有意義な情報提供、チームメンバーの開発技術指導等）を担います。この転換は、管理者は、管理方法を変える必要があるかもしれません。例えば、単に問題解決策を決め、それをチームに指示するのではなく、問答形式でチームが問題解決を見いだすのを支援します。

スクラムの開始

スクラムの第一歩は、プロダクトオーナーが製品の構想を明確に伝えることです。最終的には、プロダクトバックログと呼ばれる、洗練された特性（項目）の優先順位リストに発展します。このバックログは、製品の存続期間を終えても残り（進化）します。これは、製品のロードマップです。（図2）いつでも、プロダクトバックログは、1つです。“優先順位に従って、チームが完成させた全ての項目”も含め最終的な状態を表したり、プロダクトオーナーが、利害関係者の利益やチームに影響されること等、すべての領域を考慮して優先順位付けに必要です。

Item	Details (wiki URL)	Priority	Estimate of Value	Initial Estimate of Effort	New Estimates of Effort Remaining as of Sprint...					
					1	2	3	4	5	6
As a buyer, I want to place a book in a shopping cart (see UI sketches on wiki page)	...	1	7	5						
As a buyer, I want to remove a book in a shopping cart	...	2	6	2						
Improve transaction processing performance (see target performance metrics on wiki)	...	3	6	13						
Investigate solutions for speeding up credit card validation (see target performance metrics on wiki)	...	4	6	20						
Upgrade all servers to Apache 2.2.3	...	5	5	13						
Diagnose and fix the order processing script errors (bugzilla ID 14823)	...	6	2	3						
As a shopper, I want to create and save a wish list	...	7	7	40						
As a shopper, I want to add or delete items on my wish list	...	8	4	20						

図2：スクラム

プロダクトバックログは、様々な項目、主に新しい顧客要求（全てのユーザーが、買い物カゴに本を入れることができる）を含んでいますが、仕事の改善目標（トランザクション処理をするモジュールを拡張するように造り直します）、探求や調査の仕事（クレジットカード合法化を早く解決するために調査します。）は含まれません。また、ほんの少し問題がある場合は、もしかすると、既知の不良（注文を処理するスクリプトの不良に対し、原因を特定して修正します。）が含まれます。（不良の多いシステムには、だいたい、別に追跡システムがあります。）プロダクトバックログは、明確で持続可能な何らかの方法によって議論できます。とはいえ、製品の顧客価値に関して、“ユースケース”か“ユーザーストーリー”のどちらかでプロダクトバックログ項目を表現されることが多いです。

最新の出荷を対象としたプロダクトバックログ項目の部分集合は、リリースバックログと言われます。一般的にプロダクトオーナーは、主にこれに注力します。

プロダクトバックログは、プロダクトオーナーによって継続的に、顧客要求の変化、新しいアイデアや洞察、競合他社の動向、技術的障害等が更新されます。チームは、プロダクトバックログの各アイテムに必要な規模見積りをプロダクトオーナーに提供します。さらに、プロダクトオーナーには、各項目のビジネス価値を見積る責任があります。大抵、プロダクトオーナーはこの作業に慣れていません。この場合、スクラムマスターは、プロダクトオーナーが出来るように支援します。プロダクトオーナーは、投資収益率（ROI）を最大化するため（少ない労力で、高い価値のある項目を選びます）もしくは、副次的にいくつかの主なリスクを軽減するために、バックログの優先順位付けます。（実際は、リリースバックログの部分集合）とすると、これらの規模と価値の見積りは、各スプリントで学ぶことによって、新しくなるかもしれません。その結果、プロダクトバックログの継続的な再優先順位付けは絶えず更新されます。

スクラムは、実現するための技術または、プロダクトバックログ項目の優先順位付けの方法を定義しません。そして、見積りの技術も定義しません。一般的な技術は、（規模の要因分解、複雑性、不確実性）を相対的に見積ります。単位は、“ストーリーポイント”もしくは“ポイント”がよく使われます。

徐々に、チームは、各スプリントでどのくらい仕事ができるかを計測します。例えば、1スプリントの平均は26ポイントです。継続的に平均を達成でき、（バックログアイテム等が）何も変化しないなら、この情報を利用して、出荷日までに、全てのバックログ項目を完成することができるか、もしくは、いくつかのバックログアイテムを完成することができるかを予測できます。この平均は、チームの「ベロシティ」と呼ばれます。ベロシティは、プロダクトバックログアイテムの見積り単位と同じなので、利用できます。

プロダクトバックログ項目は、規模や作業量に応じて大きく異なります。プロダクトバックログの見直し、もしくは、スプリント計画の際に、大きいモノを小さいモノに細分化し、バックログを調整します。次回以降、直近のスプリントに関するプロダクトバックログ項目は、チームが理解できるくらい小さくて細かい粒度にすべきです。また、スプリント計画で確約できるようにすることが重要です。

スクラムの神話の1つに、詳細仕様を書く必要がないという虚構があります。でもそれは、どのような詳細仕様が必要であるかを決めるのは、プロダクトオーナーとチームです。そして、これはバックログ項目、チームの深い洞察力やその他の要素によって異なります。最小限必要で、重要なことは何か伝えて下さい。言い換えれば、全ての項目を可能な限り詳細に表現するのではなく、理解するために必要な情報だけを明確にして下さい。優先度の低い項目は、着手まで時間があり“粒度が荒い”もしくは、大きい状態でよく、従来の詳細な要件は必要ありません。優先度が高く粒度の細かい項目は、より詳細に開発される傾向があります。

スプリントの計画

各スプリント開始時に、スプリントの計画を行います。これは2部構成で、最初はスプリント計画第1部（スプリントプランニング・パート1）です。

プロダクトオーナーとチーム（スクラムマスターが会議を促進）が、プロダクトバックログの優先順位の高い項目、プロダクトオーナーが、このスプリントで着手したい項目を確認します。プロダクトバックログの優先順位の高い項目の目的や状況を議論します。また、プロダクトオーナーの考えをチームで共有します。プロダクトオーナーとチームは、全項目の（前もって定めた）“完了の定義”を満たさなければなりません。例えば、「コード標準を満たし、レビュー、テスト駆動開発（TDD）、テスト自動化100%、統合、ドキュメント作成済」等です。第1部は、プロダクトオーナーの要求を理解することに焦点を合わせます。スクラムの規律に従うと、第1部が終わると（いつも忙しい）プロダクトオーナーは席を外しますが、第2部の間は、連絡可能（例えば、電話できる状態）でなければなりません。しかしながら、第2部に出席されたら歓迎します。。。

スプリント計画第2部（スプリントプランニング・パート2）では、バックログ項目を実現するための詳細なタスクを決めることに焦点を合わせます。チームは、スプリント期間で実現出来ると公約できる量のプロダクトバックログ項目を選択します。プロダクトバックログの最上部（プロダクトバックログの最優先項目）から始まり、1つずつ優先度の順に確認します。これはスクラムを実践するために重要です。チームは、プロダクトオーナーに指示されるのではなく、実現可能と公約できる量を決めます。これは信頼して公約できます。なぜなら、誰からに指示されたことではなく、チームの分析と計画に基づいているからです。その間、プロダクトオーナーは、チームの公約する量を調整しません。なぜなら、チームが公約する項目はプロダクトバックログの最優先項目から選ばれます。チームは、その項目が最も重要であると理解しています。チームは、たまに優先度の低い項目を開発することがあります。これは、チームとプロダクトオーナーが優先度の高い項目の開発と合わせて、優先度の低い項目を簡単に開発できると気付いた時に生じます。

このスプリント計画は、しばしば何時間にも及ぶことがあります。しかし、4週間のスプリントの数時間に過ぎません。チームはこの時、仕事を終了する大切な公約をします。そして、この公約は実現するために注意しなければなりません。

チームは、各メンバーがスプリントに費やすことができる時間（他の会議に費やす時間、メールする時間、お昼休み等を差し引いた労働時間）を見積もるところから、スプリント計画第2部は始まります。たいていの方は、スプリントに費やすことができる1日の平均時間は、4～6時間です。これが、次のスプリントにおけるチームの許容量になります。図3を参照して下さい。

Sprint Length	2 weeks		
Workdays during Sprint	8 days		

Team Member	Available Days During Sprint*	Available Hours per Day	Total Available Hours
Tracy	8	4	32
Sanjay	7	5	35
Phillip	8	4	32
Jing	6	5	30

* Net of vacation and other days out of office

図3：実現可能時間の見積り

（チームの）許容量を決めると、チームが次のスプリントでいくつのプロダクトバックログ項目を達成することができるか分かります。そして、どのように達成するのかも分かります。しばしば、ホワイトボードを利用した設計議論を行います。全体設計を理解すれば、チームはプロダクトバックログ項目をタスクに分解します。チームは、プロダクトバックログの最初の（プロダクトオーナーの最も優先度の高い）項目から始めます。全員で個別のタスクに分解します。それは、スプリントバックログと呼ばれるドキュメントに記録されます。先述した通り、スプリント計画第2部の間は、プロダクトオーナーと連絡可能（例えば、電話）でなければならないので、明確化は可能です。チームは見積もった許容量を使い切るまで、同様にプロダクトバックログ項目を順次、明確化します。第2部の終了時には、チームは見積り済の全タスクのリストを作ります。（大体、数時間程度）

スクラムは、テストだけをする“テスター”のように、専門の仕事だけをするのではなく、多能工を奨励します。つまり、チームメンバーは、仕事を助け合えなくてはなりません。もし多くのテストタスクがあれば、全メンバーが助かるかもしれません。これは、全員が「万能でなければならぬ」という意味ではありません。間違いなく何名かは、特にテスト等の能力に長けていますが、チームメンバーは、一緒に働きながら、お互いに新しい技術を学習します。必然的に、スプリント計画のタスク抽出と見積りにおいて、不必要な奉仕活動がなくなり、全タスクに対して、“全員で最善を尽くす”ようになります。また、あえて苦手なタスクを選んだ方が良い時もあります。この場合、技術を学ぶという目的も含まれるので、1つのタスクに専念する方が良いかもしれません。（もしかすると、専門的な仕事かもしれません）

とは言うものの、ジョンが特定のタスクをしなければならない稀な場合もあります。なぜなら、それは、多くの時間を費やしたり、学ぶことが不可能だからです。おそらく、ジョンは、絵を描く芸術的な能力をもっている唯一の人だからです。もしそうなったら、他のチームメンバーは“ステイック・マン”を描くことはできません。この事例において、これが稀でなかったり、チームの学びとして稀でなければ、チームは不健全です。もし、全体計画の中で、この短いスプリント期間において、ジョンによって実現可能な絵を描かなければならないならば、チームはそれを確認する必要があるかもしれません。

多くのチームは、タスクを視覚化するツールを利用します。スプリント期間中、壁サイズのタスクボードに“未着手”、“着手中”、“完了”とラベルされたところを（ポストイットに書かれた）タスクが移動します。図5を参照下さい。

Product Backlog Item	Sprint Task	Volunteer	Initial Estimate of Effort	New Estimates of Effort Remaining as of Day...					
				1	2	3	4	5	6
As a buyer, I want to place a book in a shopping cart	modify database		5						
	create webpage (UI)		8						
	create webpage (Javascript logic)		13						
	write automated acceptance tests		13						
	update buyer help webpage		3						
Improve transaction processing performance	merge DCP code and complete layer-level tests		5						
	complete machine order for pRank		8						
	change DCP and reader to use pRank http API		13						

図4：スプリントバックログ



図5：可視化した管理 - 壁にはったスプリントバックログ項目

スクラムの柱の1つは、チームが公約すると、次のスプリントまでは追加や変更できません。それは、プロダクトオーナーがスプリントの途中で、チームに新しい項目を開発して欲しいと思っても、次のスプリントまで変更することはできないことを意味します。大きく優先順位を変更することによって、継続して働くことが時間の浪費になるなら、プロダクトオーナーもしくはチームは、スプリントを停止できます。チームが止まり、新しいスプリント計画から新しいスプリントを開始します。スプリントの停止は、大変混乱をまねきます。この劇的な決定を用いると、プロダクトオーナーもしくはチームの阻害要因になります。

チームへスプリント期間に目的を変更しなければ、力強く、好ましい影響を与えます。まず、チームは公約を変更せず、絶対確実なものとして働きます。それは、チームが公約により集中します。また、プロダクトバックログ項目の優先順位の決定とチームへスプリントで対応して欲しい項目の提案を通じてプロダクトオーナーを教育します。

スクラムの規律に従うことで、プロダクトオーナーは2つのことを学びます。

1つは、チームが現実的で明確に設定し、完了するという公約を信用することを学びます。そのうち、チームは現実的な公約を選択し、果たすように熟練します。もう1つは、プロダクトオーナーは、次のスプリント開始までにプロダクトバックログを変更しなければならないことを学びます。その時、追加、削除、変更、優先順位付けするのが好ましいです。プロダクトオーナーは、現スプリントの開発期間は、選択した項目を変更できませんが、それは、継続するスプリントの様々な変更要求のほんの1つに過ぎません。方向転換、要件の変更、もしくは、単に気が変わったといった不名誉な理由による変更はなくなり、プロダクトオーナーは他の誰よりもスクラムに熱狂的になります。

デイリースクラム

スプリントが始まれば、チームはスクラムの重要な習慣に従います。その1つが、デイリースクラムです。これは毎就業日、定刻にチーム全員が参加して行う（15分以下の）短い打合せです。手短にするために、全員が立った状態で行うことをお勧めします。チーム内で、お互いに障害になっていることや仕事の状況を同期する良い機会になります。デイリースクラムでは、各メンバーが3つのことを他のメンバーに説明します。(1) 前のデイリースクラムから何を完了したのか。(2) 次のデイリースクラムまでに何を完了させようと考えているのか。(3) 現在、止まっていることや障害になっていることは何か。デイリースクラムは、管理者への状況報告ではないことに注意して下さい。これは、チームが自己組織化するために、お互いの情報共有や調整するための打合せです。誰かが障害に関するメモをとります。そして、スクラムマスターは、チームメンバーが自立的に問題解決を支援する責任があります。デイリースクラムでは、議論は行いません。3つの項目を説明するだけです。もし、すぐに議論が必要ならば、デイリースクラムの後にフォローアップ会議を行います。この会議への参加義務はありません。このフォローアップ会議は、デイリースクラムで聞いたことをチームに適応するための会議です。つまり、サイクルの検査と適応します。一般的にデイリースクラムには、管理者や権威者が参加しないことを勧めます。これは、チームが“監視されている”と感じるリスクがあります。毎日の進捗報告を必要とする圧力がある環境では、問題説明は抑制されます。また、チームが自己管理を怠り、マイクロマネージメントになる傾向があります。その変わり、利害関係者が打合せに参加したり、チームの進捗を遅くする障害を助けた方が効果的です。

スプリントバックログとスプリントバーンダウンチャートの更新

スクラムチームは自己管理するために、状況を把握しなければなりません。毎日、チームメンバーは、スプリントバックログ（図6）の現在着手しているタスクを完了するために必要な残時間の見積りを更新しなければなりません。この更新の後に、メンバーの誰かがチームの残時間を合計してスプリントバーンダウンチャート（図7）に描画します。このグラフは、チームがタスクを終わらせるまで、全タスクの（計測された）残時間の新しい見積りを示します。グラフの傾きが、スプリント最終日に“ゼロになる”ような軌道を描くことが理想的です。だから、バーンダウンチャートと呼ばれます。しばしば、良く見えたり、悪く見えたりしますが、これが製品開発の現実です。重要なことは、どのくらいの時間を費やしたのか（不適切な進捗の事実）ではなく、ゴールに向かってチームの進捗を示していることです。また、残りの仕事の目的とチームを分けます。

もし、バーンダウンの曲線がスプリントの終盤で完成に向かっていない場合は、チームは調整が必要です。例えば、仕事の範囲を減少させるか、持続可能なペース保っているうちに、より効率的に仕事の方法を見つける等です。

スプレッドシートを利用してバーンダウンチャートを作ることはできますが、多くのチームは作業場の壁に大きな紙を貼り、ペンで更新していくことの方が、より効果的だと気付いています。この“低い技術／高い手作り”の解決法は、迅速かつ簡単で、コンピュータのチャートより、目に触れる機会が多くなります。

Product Backlog Item	Sprint Task	Volunteer	Initial Estimate of Effort	New Estimates of Effort Remaining at end of Day...					
				1	2	3	4	5	6
As a buyer, I want to place a book in a shopping cart	modify database	Sanjay	5	4	3	0	0	0	
	create webpage (UI)	Jing	3	3	3	2	0	0	
	create webpage (Javascript logic)	Tracy & Sam	2	2	2	2	1	0	
	write automated acceptance tests	Sarah	5	5	5	5	5	0	
	update buyer help webpage	Sanjay & Jing	3	3	3	3	3	0	
...						
Improve transaction processing performance	merge DCP code and complete layer-level tests		5	5	5	5	5	5	
	complete machine order for pRank		3	3	8	8	8	8	
	change DCP and reader to use pRank http API		5	5	5	5	5	5	
...						
	Total (person hours)		50	49	48	44	43	34	

図6：毎日のスプリントバックログ残作業の更新

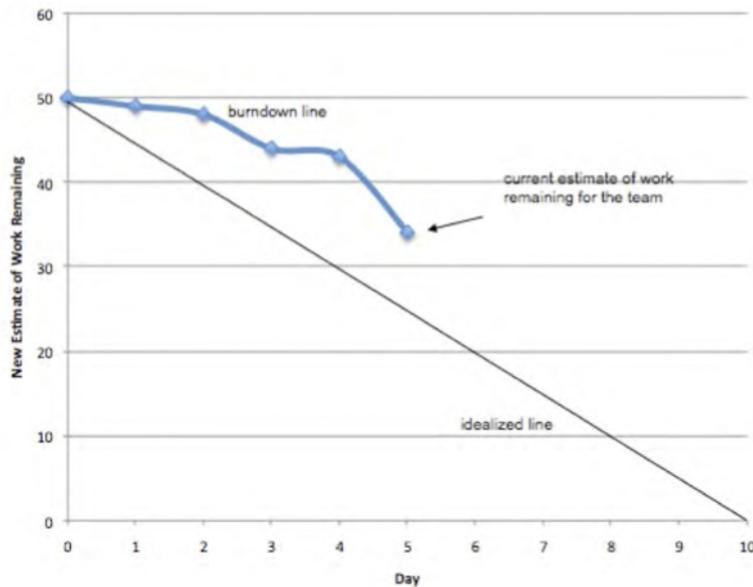


図7：スプリントバーンダウンチャート

プロダクトバックログの見直し

あまり有名ではありませんが、各スプリントの5-10%を使って、チームによるプロダクトバックログを洗練（もしくは、手入れ）することは、スクラムの貴重な原則の1つです。これは、詳細な要求分析、大きい項目の細分化、新しい項目の見積り、既存の項目の最見積りを含んでいます。

スクラムは、どのように仕事を完了させるかを定義しません。しかし、頻繁に利用される技術は、障害を取り除きチームとプロダクトオーナーが献身的に働けるスプリントの終盤に集中します。2週間のスプリントでは、スプリントの5%（半日）をプロダクトバックログの見直しに費やします。この改善活動は、現スプリントの項目は選びません。1、2スプリント後のために行います。この習慣に従うと、プロダクトオーナーとスクラムチームは明確で、詳細分析され、丁寧に見積られている項目がある状態でスプリントの計画を始めるので、比較的簡単になります。この改善活動が（十分に）終わっていない状態では、重要な質問や気づき、曖昧や不十分と感じるスプリント計画になることがあります。しばしば、計画がスプリントに入りきらないことがあります。それは望ましくありません。

スプリントの終了

スクラムの原則の1つは、スプリント期間を決して延長しません。それは、チームが公約した仕事を終えたかどうかに関わらず、終了します。一般的に、チームは最初から数スプリントは、多めに公約して達成できません。時々、多めに公約した後は、少なめに公約して早く終了します。（その場合、プロダクトバックログ項目で完了できる項目をプロダクトオーナーと相談できます。）しかし、チームは、だいたい3、4スプリントで（スプリントで最も）許容できる量を把握します。それは、今後のスプリントのゴールをより明確にします。チームは、（2週間の）スプリントでそれを持続し、変えないよう推奨されます。チームが、どのくらい達成することが出来るかを学ぶことで、見積りと長期のリリース計画を支援します。また、チームが仕事のリズムを得ることを支援します。これは、よくスクラムでは、チームの“鼓動（ハートビート）”と呼ばれます。

スプリントレビュー

スプリントが終わると、スプリントレビューを行います。ここでは、チームとプロダクトオーナーがスプリントを見直します。しばしば、“デモ”と誤った認識をされますが、“デモ”では本当の目的を得ることは出来ません。スクラムの重要な目的は、検査と適応です。確認と学習は、フィードバックの繰り返しに基づきます。スプリントレビューは、製品の検査し、適応を学びます。スプリントレビューは、プロダクトオーナーが製品とチームについて学ぶ機会になります。また、チームはプロダクトオーナーと市場を学ぶ機会になります。従って、レビューの最も重要な要素は、チームとプロダクトオーナーで徹底的に話し合い、状況把握やアドバイス等を得ることです。レビューは、チームがスプリントで構築した内容をデモすることを含んでいますが、レビューが会話よりデモに集中するのは、バランスが悪いです。

しばしば見落とされますが、役立つスクラムの原則は、製品もしくは納品に関する“完了の定義”を明確にします。これは、スクラムマスターの責任です。スクラムマスターは、チームがデモする、もしくは、“完了の定義”に従って完了していない項目に関して議論しないようにします。“完了”されていない項目はプロダクトバックログに戻されます。そして、プロダクトオーナーによって再度、優先順位付けされます。この方法は、仕事の品質に透明性があります。チームは、製品（例えば、ソフトウェア等）の品質をごまかすことはできませんが、多分、乱雑で質の悪い仕事を積み重ねることはできます。（例えば、テストされていないコード等）

この打合せの内容には、プロダクトオーナー、チームメンバー、スクラムマスター、そして、顧客、利害関係者、専門家、幹部等が関心があります。

スプリントレビューの一部のデモは、チームの“発表会”ではありません。なので、スライド等は不要です。スクラムの原則は、レビュー準備に費やす時間は30分未満であるべきです。そうでなければ、チームに何か問題があります。これは、単に構築されたモノのデモです。出席している誰もが、自由に質問したり、触れたりすることができます。

スプリントレトロスペクティブ（ふりかえり）

スプリントレビューは製品の点検と適応に焦点をあてます。レビューに続き、スプリントレトロスペクティブ（ふりかえり）はスプリントの工程（プロセス）の点検と適応に焦点をあてます。いくつかのチームは、省略することがありますが、それはとても残念です。なぜなら、改善可能な領域を理解する主な方法だからです。それは、製品に影響します。チームにとって何が有効で、何が有効ではないのかを議論し、認識や変更等をするための良い機会です。チームとスクラムマスターは参加しますが、プロダクトオーナーには、（歓迎しますが）参加の義務はありません。時々、スクラムマスターはレトロスペクティブ（ふりかえり）を効果的に進行しますが、中立的な外部者がよい場合もあります。良い方法は、スクラムマスターが、チームでレトロスペクティブ（ふりかえり）をするように促し、チームの交流を容易にすることです。

レトロスペクティブ（ふりかえり）には多くの方法があります。Agile Retrospectives (Derby, Larsen 2006)（和書：アジャイルレトロスペクティブズ 強いチームを育てる「ふりかえり」の手引き）という本に、役立つ技術が書かれています。議論を構築する簡単な方法は、ホワイトボードに、“良かったこと”、“良くなかったこと”の2つに分類した表を書き、各メンバーは1つ以上の意見を書きます。同じ意見の項目の横にはチェックマークが付くので、共通項目は明確になります。それから、チームは根本原因を探求し、次のスプリントで少し改善を試み、次のレトロスペクティブ（ふりかえり）でその結果を確認することに同意します。

レトロスペクティブ（ふりかえり）の最後に役立つ方法は、チームが各項目に対し「スクラムが要因の（スクラムを通じて気付いた）項目には“C”、スクラムによって顕在化した（スクラムに関係ないが、スクラムが顕在化した）項目には“E”、スクラムに無関係項目には“U”」と分類します。チームは、“良かったこと”の欄で多くの“C”を、“良くなかったこと”の欄で多くの“E”を見つけるでしょう。“良くなかったこと”の欄の項目が多いのは良いことです。なぜなら、根本原因の問題解決への最初の一步は、明らかにすることだからです。そして、スクラムはそのための良いキッカケになります。

リリースバックログとバーンダウンチャートの更新

この時点では、いくつかの項目が、完了や追加、新たに見積られたり、リリースから外されたりしています。プロダクトオーナーは、リリースバックログ（プロダクトバックログ）に変更を確実に反映することに責任があります。加えて、スクラムはリリース日への進捗を示しているバーンダウンチャートがあります。これは、スプリントバーンダウンチャートに似ていますが、細かなタスクではなく、高いレベルの項目（要求）を描画します。新しいプロダクトオーナーは、このチャートをなぜ、また、どのように作るのかを理解していない場合は、別の機会に、スクラムマスターがプロダクトオーナーに教えています。図8と図9は、リリースバックログとリリースバーンダウンチャートの例です。

Item	Details (wiki URL)	Priority	Estimate of Value	Initial Estimate of Effort	New Estimates of Effort Remaining at end of Sprint...					
					1	2	3	4	5	6
As a buyer, I want to place a book in a shopping cart (see UI sketches on wiki page)	...	1	7	5	0	0	0			
As a buyer, I want to remove a book in a shopping cart	...	2	6	2	0	0	0			
Improve transaction processing performance (see target performance metrics on wiki)	...	3	6	13	13	0	0			
Investigate solutions for speeding up credit card validation (see target performance metrics on wiki)	...	4	6	20	20	20	0			
Upgrade all servers to Apache 2.2.3	...	5	5	13	13	13	13			
Diagnose and fix the order processing script errors (bugzilla ID 14823)	...	6	2	3	3	3	3			
As a shopper, I want to create and save a wish list	...	7	7	40	40	40	40			
As a shopper, I want to add or delete items on my wish list	...	8	4	20	20	20	20			
...			
Total				537	580	570	500			

図8：リリースバックログ（プロダクトバックログの部分集合）

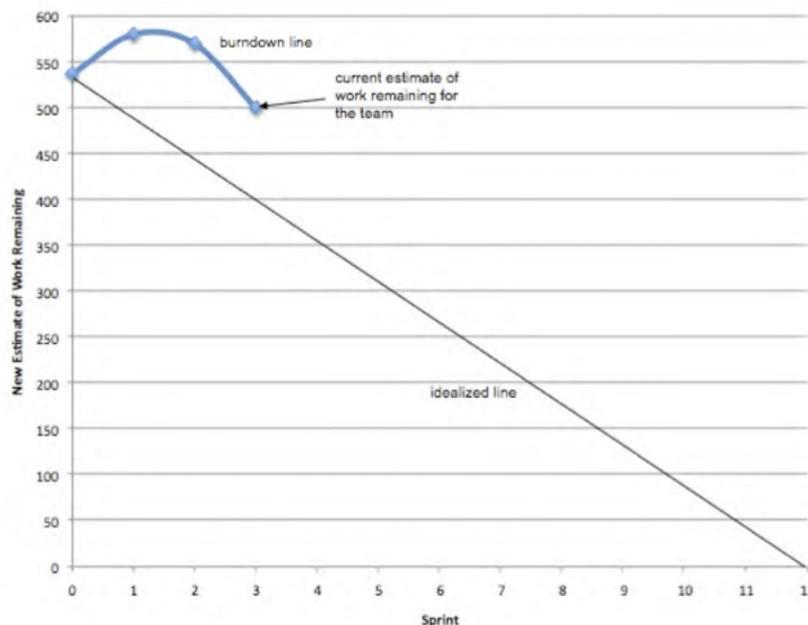


図9：リリースバーンダウンチャート

次スプリントの開始

スプリントレビューで何かを得て、プロダクトオーナーはプロダクトバックログを更新するかもしれません。この時点で、プロダクトオーナーとチームは次のスプリントを開始する準備ができています。スプリント間の中断時間はありません。チームは通常、ある午後のレトロスペクティブ（ふりかえり）から、次の朝（もしくは、月曜の朝）にスプリント計画を行います。

アジャイル開発の原則の1つに、“持続可能なペース”があります。適切な水準の業務時間内で、チームはこのサイクルをいつまでも続けることができます。

リリーススプリント

スクラムの完成とは、スプリント終了時に出荷可能な製品を納品できることです。それは、テストやドキュメントのような必要な“仕上げ作業”がないことも含みます。スプリント毎に完全に作業が終わっているということは、実際に納品するか、スプリントレビュー直後に展開できます。これは、各増分は完成品の一部ということを意味し、プロダクトオーナーと利害関係者に透明性を提供します。彼らは、各スプリントの最後をより正確に把握できます。

しかし、多くの組織は、脆弱な開発手法、ツール、基盤、一定水準が完成できない、もしくは、その他の酌量すべき事情（コンピュータが壊れた等）があります。こうした場合は、いくつかの作業が残るので、完成品の統合テスト等、残った作業を行う“リリーススプリント”が必要になります。

リリーススプリントが必要ということは、いくつかの改善点があるサインなので、注意して下さい。理想は、それがない状態です。必要とする場合、プロダクトオーナーが製品の納品準備が整ったと判断するまで、スプリントが続きます。これは、納品準備のためのリリーススプリントになります。もし、チームが良い開発（継続した改善や一体化、各スプリント内での効果的なテスト等）を実践すれば、プレリリースの安定化、もしくは、最終確認の作業が必要なくなります。

リリース計画と初期プロダクトバックログの見直し

時々、「どのようにしたら反復モデルで長期のリリース計画ができますか。」という質問をされます。よく考えると2つのケースがあります。(1)は、新製品の初期リリース。(2)は、既存製品の次回以降のリリース

新製品、既存製品のどちらにスクラムを導入する場合でも、最初のスプリント後に初期のプロダクトバックログを見直す必要があります。そうすることによって、プロダクトオーナーとチームは適切なプロダクトバックログを作ることができます。これは、最初のリリースに必要な全項目の構想（考え方）、いくつかの詳細な要求分析、見積りを行うために、数日から1週間かかります。

意外にもスクラムでは、プロダクトバックログが固まっている既存製品の場合、次リリースのための特別もしくは詳細なリリース計画は必要ありません。なぜでしょうか。それは、プロダクトオーナーとチームは全スプリントで、（各スプリントの5-10%）プロダクトバックログの見直しを行いながら、継続して将来（リリース）の準備をするからです。この継続した製品開発の方法は、従来の逐次的な開発方法で、特に強調される「準備-実行-結果」の工程が必要なくなります。

初期のプロダクトバックログの見直しや各スプリントでの継続的なプロダクトバックログの見直しの中で、チームとプロダクトオーナーは、リリース計画、見積りの精査、優先順位、製品等について学びます。

リリース駆動になることもあります。例えば、11月10日の展示会にて、製品のバージョン2.0をリリースします。この状況では、チームは出来るだけ多くのスプリント（多くの機能構築）を行います。

他製品は、完成前に特定の主要なモノを必要とされ、製品はいくら時間がかかろうとも要件を満たさなければ出荷できないでしょう。スクラムでは、スプリント毎に出荷可能な状態の製品を生産することを強調するため、プロダクトオーナーは、顧客がより早く完成した製品の恩恵を得るために、中間リリースを選択できます。

あらかじめ、全てのことを把握することはできません。焦点は、（例えば、スケジュールの範囲等の）計画を作り洗練する時に、リリースの広い方向性を加え、明確なトレードオフを決めることが出来ることです。これを最終的な目的地へ導くロードマップを考えて下さい。旅の正確な（最終的な）道の決定は、旅の途中で決まるかもしれません。

ほとんどのプロダクトオーナーは、1つのリリース方法を選択します。例えば、出荷日を決めて、その日までに完成できるプロダクトバックログ項目を見積るために、チームと協働します。決められた“価格、期日、成果物”を公約しなければならない状況（例えば、開発の契約）では、多少の不確実性と変化を許容できるようにしなければなりません。

アプリケーションもしくは製品の焦点

市場または内製に関わらず、アプリケーションもしくは製品のために、スクラムは、古いプロジェクト中心モデルから離れ、連続的なアプリケーションもしくは製品開発モデルに変えます。そこには、長期プロジェクトの序盤、中盤、終盤はありません。それに、従来のプロジェクト管理者も存在しません。むしろ、安定したプロダクトオーナーと寿命の長い自己組織化されたチームが、製品もしくは、アプリケーションが償還されるまで、固定したスプリントを“終わりのなく”繰り返しながら協働するだけです。“プロジェクト”管理に必要な全ての仕事は、チームとプロダクトオーナー（内部のビジネス顧客や製品管理者等）によって処理されます。IT管理者やプロジェクト管理部の誰かに管理されることはありません。

スクラムは、実際の（長期のアプリケーションに発展しない）1度きりのプロジェクトにも利用できます。この場合でも、プロジェクトの管理をチームとプロダクトオーナーが行います。

長期間専用で保証しなければならない多くの既存のアプリケーションを持っているチームに、不適切な新たな仕事が発生したらどうしますか。この場合、安定した寿命の長いチームは、1つのスプリントで、1つのアプリケーションの項目に対応し、他の項目には次のスプリントで対応します。こうした状況では、スプリント期間は極めて短く、1週間くらいになります。

たまに、不適切な新たな仕事の事前策は、チームが同じスプリントで、いくつかのアプリケーションの項目に対応しないことです。しかし、この事前策では、複数アプリケーションの作業をするという非生産的な平行作業をしないよう注意しなければいけません。スクラムの基礎的な生産力は、チームが1スプリントに1つの製品もしくはアプリケーションに集中することで高まります。

共通課題

スクラムは習慣を明確にするだけでなく、より重要なことは、透明性を提供するフレームワークということです。これは、“検査と適応”ができる仕組みです。スクラムの機能障害と障害となっている

物事を明らかにする働きは、プロダクトオーナーとチームの有効性に影響を与え、それらに取り組みさせることができます。例えば、プロダクトオーナーは実市場、特性、またはビジネス価値との関係を知らないかもしれません。もしくは、チームは努力見積り、開発業務が下手かもしれません。スクラムのフレームワークは、迅速に弱点を明らかにします。スクラムは開発の問題を解決しません。小さな改善の試みと短期間での問題解決方法を探求する枠組みを苦痛なほど明確化して提供します。

仮に、タスク分析と見積り能力が乏しいため、最初のスプリントは公約を果たすことができなかつたとします。この時、チームは失敗したと感じます。しかし、実際は、この経験は、公約に関してより現実的に深く考えるようになるための必要な第一歩です。スクラムは機能障害の明確化を助け、それをチームが実行出来るようにするパターンは、チームがスクラムの経験を利用して最も大きな利益を得ることができる基本的な仕組みです。

スクラム導入時によくある過ちは、スクラムを変えることです。例えば、チームは公約したことを達成できないトラブルがあった時、スプリント期間を延長するかもしれません。そうすれば、時間がなくなることはありませんし、より良い仕事をするための見積りと時間管理をどのようにすれば良いかを学ばなくなります。このような方法では、経験豊富なスクラムマスターや組織的なコーチングや支援がなければ、不十分な鏡像や機能障害によってスクラムを変異させます。そして、スクラムの本当の利点を徐々に弱らせます。

もう1つよくある過ちは、スクラムが明確に要求しないから、実行を妨げたり禁止しようと思込むことです。例えば、スクラムはプロダクトオーナーの長期的な製品戦略を必要としません。また、技術者も複雑な技術的な問題への経験豊富な技術者から助言を求めることを必要としません。スクラムは正しい決断を関係者に任せます。そして、多くの場合、これらの（その他多くのモノを伴う）実践は、上手に情報を伝えます。

他に注意すべきことは、管理者がチームにスクラムを強制することです。スクラムは、チームに自由と管理するツールを提供し、上司から命令されても成功しません。より良い方法は、チームで同僚や管理者にスクラムを学ぶことから始め、総合的に教育を受けた専門家の訓練を経て、一定期間、忠実に実践するとチームが決断し、その期間の終わりに、チームはその経験の重要性を評価し、継続するかどうかを決めるのが良いかもしれません。

朗報は、たいていチームにとって最初のスプリントはとても挑戦的ですが、スクラムの利点が終盤までに、明らかになる傾向があり、多くの新しいスクラムチームは以下のことを主張します。「スクラムは難しいですが、これまでの方法より良くなることを確信しています！」

付録：専門用語

バーンダウン

スプリント、リリース、もしくは製品の全域で、残作業の傾向を時間で計ります。情報源は、スプリントバックログとプロダクトバックログです。縦軸は残作業、横軸は（スプリントの日数またはスプリント数）期間として、計測します。

ニワトリ

プロジェクトに関心はあるが、公的にスクラムの責任や製品の説明責任を持っていない人（チーム、プロダクトオーナー、スクラムマスター以外）。

デイリースクラム

毎日、仕事を検査、同期し、各チーム内で進捗や報告、スクラムマスターへの障害除去等を把握する短い打合せです。スプリントを最適化するために次の仕事を検討するフォローアップ会議は、デイリースクラムの後で行われるかもしれません。

完了 (done)

組織の基準、慣例、方針に従い、全メンバーが同意するように完成します。

スプリントレビューで“完了 (done)”と報告された時、同意された定義に従っていなければなりません。

(スプリントバックログ項目の) 残作業の見積り

タスクの残作業をチームメンバーが何時間か見積ります。この見積りは、スプリントバックログの仕事をしている期間は毎日、更新されます。この見積りは、仕事をする人数に関係なく見積られた残時間の合計です。

増分 (インクリメント)

各スプリントでチームによって開発された製品の機能性は、出荷可能もしくは、プロダクトオーナーの利害関係者に役立ちます。

潜在的に出荷可能な製品の機能性の増加

もし、プロダクトオーナーや利害関係者が実行することを望めば、完成した製品がシステムの全体の一部は役立ちます。

スプリント

反復、もしくは、繰り返す周期の最小単位です。この期間に、製品やシステムの増分を製造します。通常、1週間以上1ヶ月未満です。継続期間は、総合的な全仕事の初めから終わりまで役に立ちます。また、同じシステムや製品の全項目は、同じ周期で開発します。

ブタ

スクラムの3つの役割（チーム、プロダクトオーナー、スクラムマスター）の誰か。公約して、実行権限を持っていて実行します。

プロダクトバックログ

製品の機能を開発するために、必要な規模が見積られた要求の優先順位付けされたリストです。項目の優先順位が高いほど、優先してより正確に見積ります。リストによって、技術もしくは事業状態の変化が分かります。

プロダクトバックログ項目

機能要件、非機能要件、目的、ビジネス、依存関係、見積りの重要度の順に、優先順位付けされます。見積りの精度は、優先度とプロダクトバックログ項目の粒度によります。最優先項目は粒度が細かく明確な状態で、次のスプリントで選択されます。

プロダクトオーナー

プロジェクトの価値を最大化するためにプロダクトバックログを管理する責任者。プロダクトオーナーは、全員の関心を引き、プロジェクトの価値を説明し、製品の結果に責任があります。

スクラム

頭字語ではありません。ラインの外に出たボールを試合に戻す際にボールを獲得するためのラグビーのルールです。

スクラムマスター

スクラムの過程、正確な実施、利益の最大化に責任者。

スプリントバックログ

スプリントで、チームの仕事を定義したタスクリストです。リストはスプリントの間、存在します。スプリント中に完成の責任がある見積った各タスクの残量が確認できます。

スプリントバックログタスク

チームのタスクの1つ。もしくは、チームメンバーがシステムを機能拡張するために、公約の責任があるプロダクトバックログ項目を明確にしたものです。

スプリント計画

全スプリントの最初に行う会議です。4週間のスプリントの場合、1日8時間が目安です。会議は2つのパートに分かれます。4週間のスプリントの場合、4時間ずつが目安です。第1部は、プロダクトオーナーが最優先のプロダクトバックログ項目を伝えます。チームとプロダクトオーナーは、チームが次のスプリントで、機能拡張できるプロダクトバックログ項目はいくつかを決定するように協働します。チームは第1部の終わりに、達成できるプロダクトバックログ項目を公約します。第2部は、この公約を実現するか計画し、スプリントバックログに詳細な仕事を計画します。

スプリントレトロスペクティブ（ふりかえり）

スクラムマスターが進行する完成したチームは、スプリントの総括と何を改善すれば次スプリントの生産性を高め、楽しめるかを明確にします。4週間のスプリントの場合、3時間が目安です。

スプリントレビュー

各スプリントの終わりに行う会議です。チーム、プロダクトオーナーと利害関係者が協力して、スプリントを確認します。4週間のスプリントの場合、3時間が目安です。一般的には、完成したプロダクトバックログ項目の実演から始まり、議論、制限や気づき、そして、次に行く（もしかするとプロダクトバックログの変更をもたらすかもしれない）最も良いことは何かを議論する良い機会です。ただし、完成した機能しか実演できません。

利害関係者

プロジェクトの結果に関心がある出資者もしくは、影響がある人。

チーム

全スプリントで、製品開発のための自己管理に責任がある職能上の枠を超えたユニット。

タイムボックス

行事や打合せを行う延長できない時間枠です。例えば、デイリースクラムは15分のタイムボックスです。何があろうと、15分で終了します。打合せによっては、短時間に設定されるかもしれません。スプリントは、正確に期間を持続します。