

認定スクラムマスター セミナーで学んだこと

2019/01/25

今回の内容

アジャイル/スクラムの

- 思想
- 概要

- 理想論を含みます
- 極論よりの内容です
- 完璧な改善策ではないです

アジャイルソフトウェア宣言

私たちは、ソフトウェア開発の実践
あるいは実践を手助けをする活動を通じて、
よりよい開発方法を見つけだそうとしている。
この活動を通して、私たちは以下の価値に至った。

プロセスやツールよりも **個人と対話**を、
包括的なドキュメントよりも **動くソフトウェア**を、
契約交渉よりも **顧客との協調**を、
計画に従うことよりも **変化への対応**を、

価値とする。すなわち、左記のことがらに価値があることを
認めながらも、私たちは右記のことがらにより価値をおく。

スクラムとは

- メソッドではない
 - やるべきことややり方を教えるものではない
 - ツールは与えられるが、自分たちで使い方を考える
-
- ソフトウェアに限った話ではない
 - プロダクト開発であればどこでも機能する
-
- 検査と適応を可能にする枠組み
 - チームが最も重要な作業から手を付ける

スクラムとは

- 目的ではない
- 生産性の向上ではなく、生み出す価値を高める
ムダ/ムラ/ムリを省き、ROI（投資利益率）を高める
- 自己組織化により、メンバーのエンゲージメントを高める
- マネージャは必要ない

ムダ/ムラ/ムリの改善

- ムダ

読まれないドキュメント、実行されないコードはなし

- ムラ

短期間のフィードバックサイクルでチームの考えを合わせる

- ムリ

タイムボックス化により邪魔されない時間を

なぜスクラムか

- 予測に注意。特に未来予測には
- 伝統的なプロセスは長期間の予測に基づく
 - 必要とされるものの推測を間違える
 - 完了できる時期の予想が外れる

アジャイル・スクラムの仕組み

- 頻繁な検査と適応
- 要求、技術、チームの能力を創発
- フィードバックを用いて新たな事象を捕まえ、自己組織化と適応で対応する
- インクリメンタルな双発
- 現実に対処する。内部ドキュメントではなく。抽象化は悪。神は細部に宿る
- 互いに調整する

どうしたらそうになれる？

- コミュニケーションの最大化
 - 効率的な定期ミーティング（毎日）
 - 全員同席。小さなチームの力学
 - 俊敏に障害を明らかにし取り除く
- 自律性
- タイムボックス化
 - チームの空間を作り、邪魔されない時間を
- 期待するのはチームがベストをつくすこと
 - 許可を求めるな、謝罪せよ

スクラムの構成要素

- 3つのロール
 - プロダクトオーナー
 - スクラムマスター
 - 開発チーム
- 4つのイベント
 - スプリントプランニング
 - レトロスペクティブ
 - スプリントレビュー
 - デイリースクラム
- 3つのリスト
 - プロダクトバックログ
 - スプリントバックログ
 - 障害リスト

プロダクトオーナー

- プロダクトのビジョンを創り、育て、語り合う
- 投資ビジョンに対しての価値(ROIなど)を管理する
- プロダクトバックログを管理する
- リリースが”出荷可能”かどうかの判断を行う
- チーフプロダクトオーナーに率いられたプロダクトオーナーチーム になることもある。
- “開発者、プロダクトオーナー、スクラムマスター”をあわせて
“スクラムチーム”と呼ばれる

開発チーム

- クロスファンクショナル(職能横断的)
- チーム参加者は、長期的に安定している
- プロダクトバックログの一番上の項目から開発する
- 作業を見積もりやすい小さなタスクに分割する
- 開発イテレーションが予想通りになるよう管理する
- スプリントと呼ばれるタイムボックスの期間内でプロダクトを開発する

スクラムマスター

- ROIを最適化する文化と環境を維持する
- スプリント計画を運営する
- スプリントレビューミーティングを招集する
- 外部からの攪乱からチームを守る盾となる
- 進捗に対する障害を取り除く
- チームに指示しない、チームが何をすべきか言わない、という場合もある
- 振り返りを促進すべき

顧客について

- スクラムには顧客という定義はない
プロダクトオーナーはすべてのステークホルダーと話す
開発チームもステークホルダー：良いコードで仕事する権利
- ソフトウェアの世界では顧客というのが一般的。
- エンドユーザーのほうがずっと大事
アジャイルはここに注力すべきと教えている。
- 顧客というと、モノを引き渡す関係
それより、パートナー関係を築くほうがよい。

鶏と豚

- スクラムのミーティングにはトリ(鶏)とブタ(豚)がいる
 - トリさんとブタさんが、一緒にレストランを開くこと にしました。
 - ブタさん「お店の名前はなににしようか？」
 - トリさん「ハムエッグなんてどう？」
 - ブタさん「やめてよ、あなたは協力するだけでいいけど、私は身を切らなきゃいけないじゃないの」
- スクラムでは、豚は話してよいが、鶏は話してはいけない

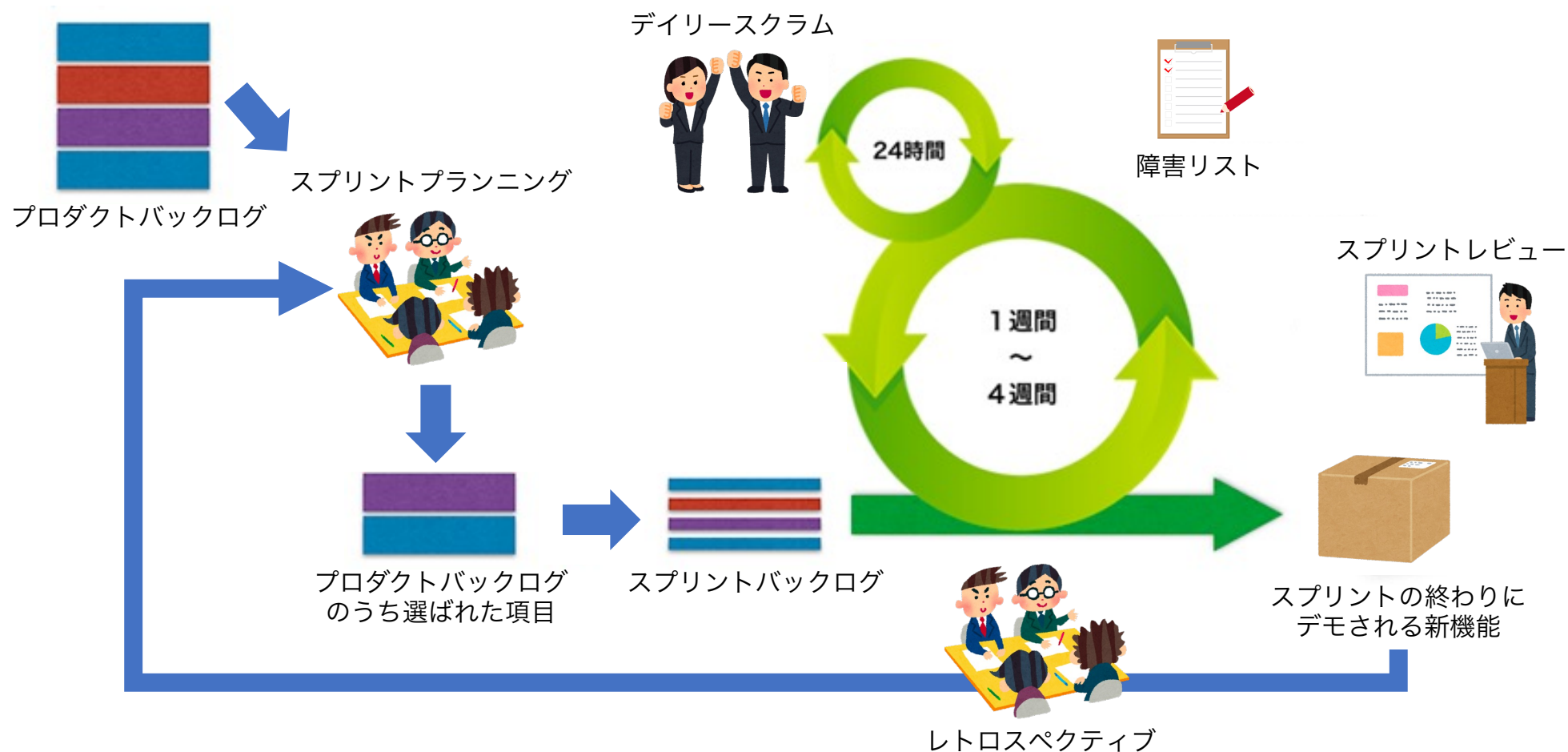
スクラムチームの鶏と豚

- スクラムチームは、スクラムマスター、プロダクト オーナー、開発チームから構成される
- 開発者は“豚”と呼ばれる
- プロダクトオーナーはプロダクトバックログの“豚”
- 開発チームはスプリント作業の“豚”
- スクラムマスターはスクラムプロセスの“豚”
- それ以外は“鶏”

単一首絞め点

- スクラムマスターは開発チームのスプリントでのデリバリーの最終責任者
- プロダクトオーナーは企業の ROI の最終責任者

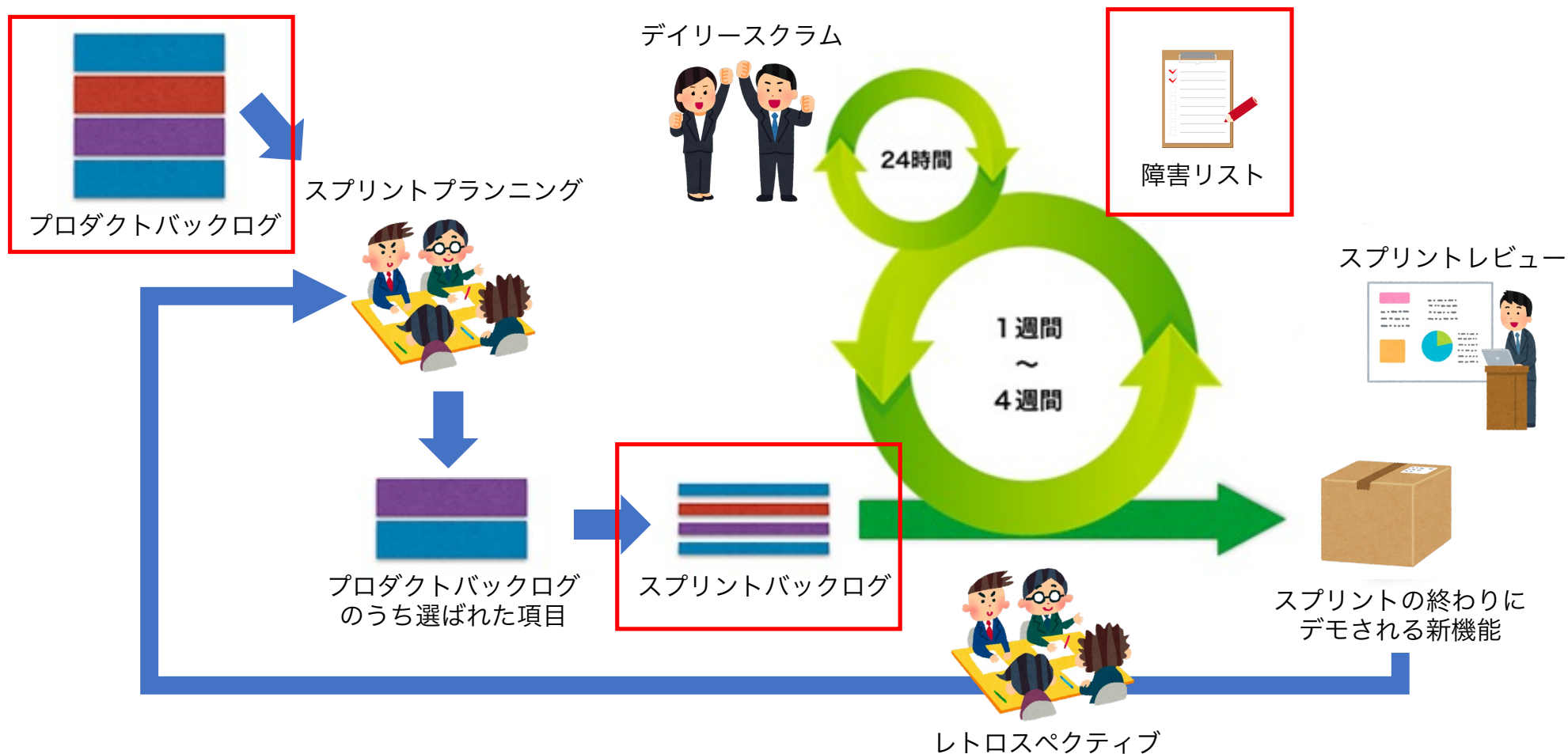
スクラムの基本フロー



スプリント –タイムボックス化–

- チームが邪魔されずに働ける時間
- プロジェクトのリズムを作る
- スプリントはタイムボックス化の主たる単位
- スプリントはチームの予想を表す
- スプリントの間は、
 - デリバリーのスコープ(PBI)は変化しない
 - スプリントゴールを脅かす変更は受け付けない
 - 外部からの新たな要求は受け入れないかも
- フィードバックループをしっかりと守る

3つのリスト



プロダクトバックログ

- 機能、技術、課題をリストにしたもの
→「プロダクトに含まれるもの」
- 上位3スプリント分は実現可能な仕様
- 課題とは後日定義される作業のプレースホルダ
- 後発的、順序付けされ、見積もられた
- 上位のバックログ項目ほど詳細
- 複数のチームに一つのリスト
- プロダクトオーナーは順位付けに責任をもつ
- あらゆるステークホルダが貢献(記入)できる
- 見えるところで管理し、追加する

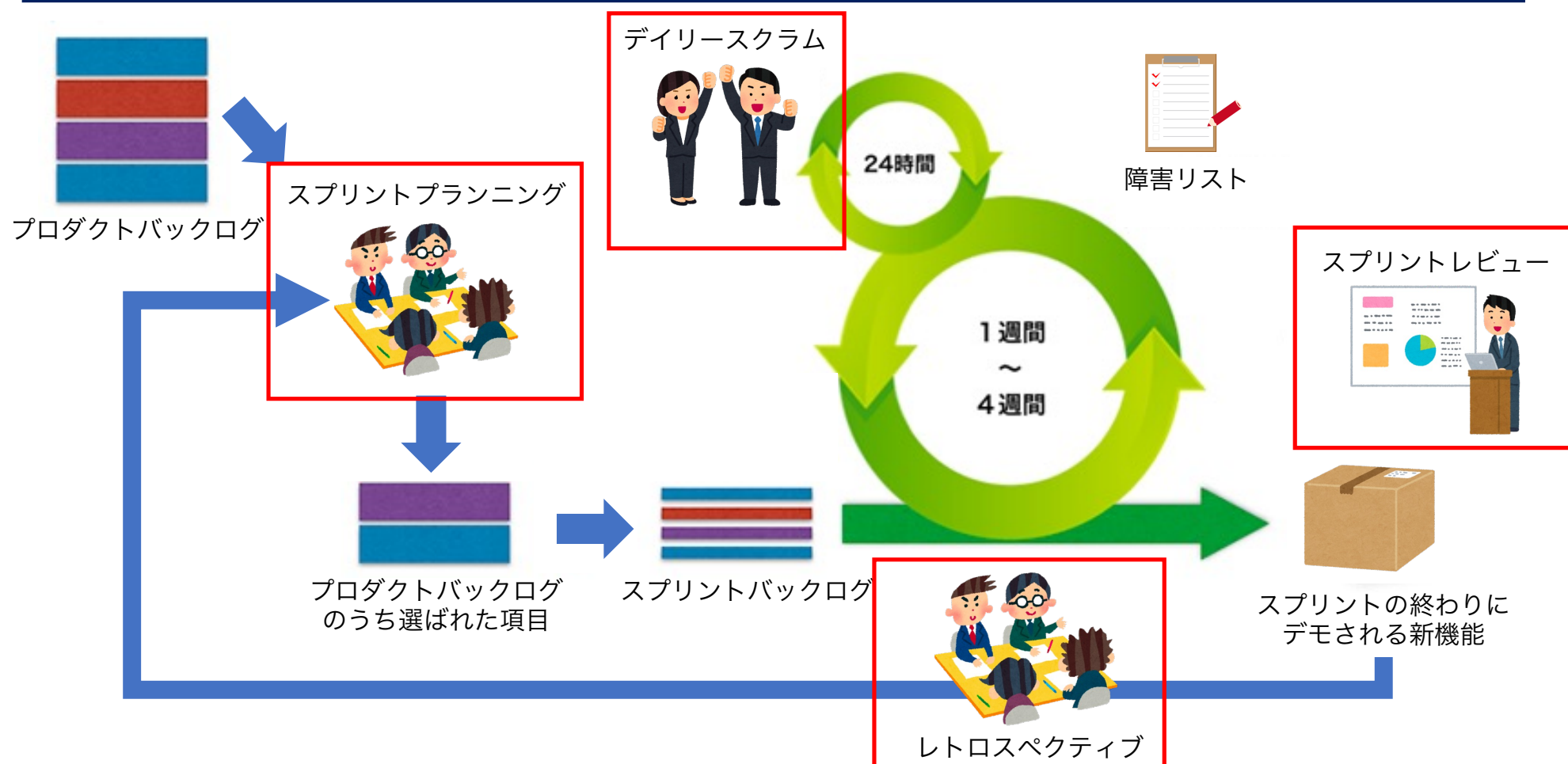
スプリントバックログ

- スプリント中に実施するタスクのリスト
- 開発メンバーがどう作るかを決める
- プロダクトバックログの順位を反映する
 - プロダクトオーナーの意思が反映される

障害リスト

- スクラムマスターのバックログ
- 毎日更新
- オープン、可視化、正直
- チームは、まず自分たちで障害を解決するよう努力しなければならない。スクラムマスターは、まずは押し返す
- スクラムマスターはチームが解決できなかった障害のみ、責任をもつ
- 解決できなければ、最後にマネジメントにエスカレーション

4つのイベント



スプリントプランニング

- 初期のプロダクトバックログを作り、継続的に更新するミーティング
- すべてのステークホルダーが出席する: プロダクトオーナー、スクラムマスター、開発チーム。さらに他のステークホルダーにも出てもらうことも
- プロダクトオーナーが主催し、スクラムマスターはミーティングが行われることを確認する
- トピック1: プロダクトバックログに関する作業
- トピック2: 作業計画に関する作業

デイリースクラム

- 同じ場所で、同じ時間に、毎日行う
- チームメンバー 一人につき平均一分くらい
 - スタンドアップ!
- 見える化にフォーカスする — 問題解決でなく
 - 問題解決は、オフラインで
- 完全な正直さ、オープンネス、見える化
- 個人的な障害だって、れっきとした障害
 - チーム全員が、チームの全ての個人を助ける

スプリントレビュー

- 2-3時間
- 最大で、プレゼン1時間
- パワポ禁止
- ソフトウェアの開発やテストをする環境で行う
- チームから、プロダクトオーナーと顧客/ユーザーに向けて披露する
- 次のスプリント計画の土台となる
- 製品の機能の「いつでも出荷できる増分」を説明しなければならない

レトロスペクティブ

- スプリントで起こった良かったこと、改善したほうが良いことを明確にする
- 改善のアイデアを全員で出す
- 何をすべきかを決定する