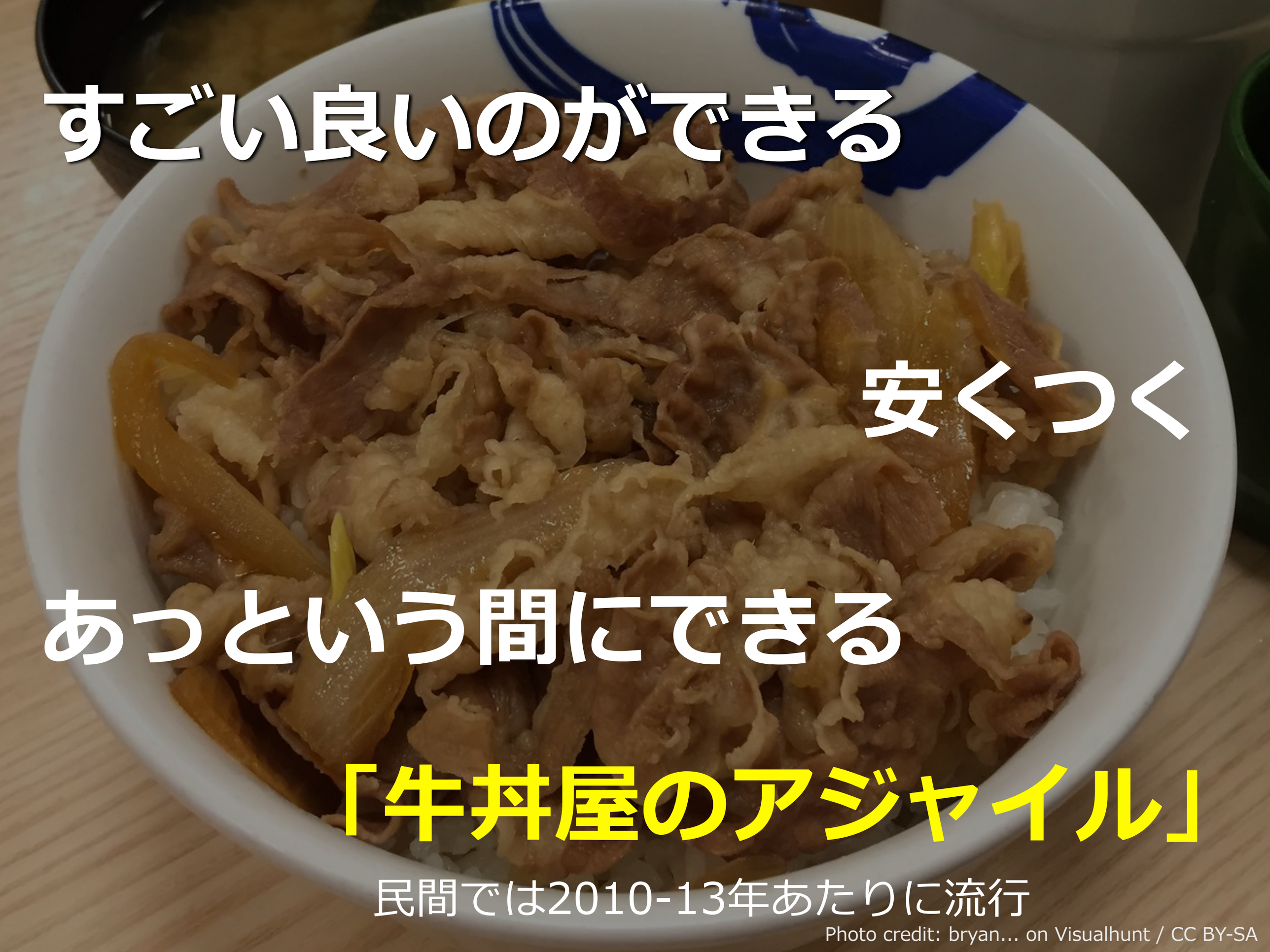
アジャイルとは何が良いのか？

アジャイル開発を
実施する意義(利点)とは？



すごい良いのができる

安くつく

あっという間にできる

「牛丼屋のアジャイル」

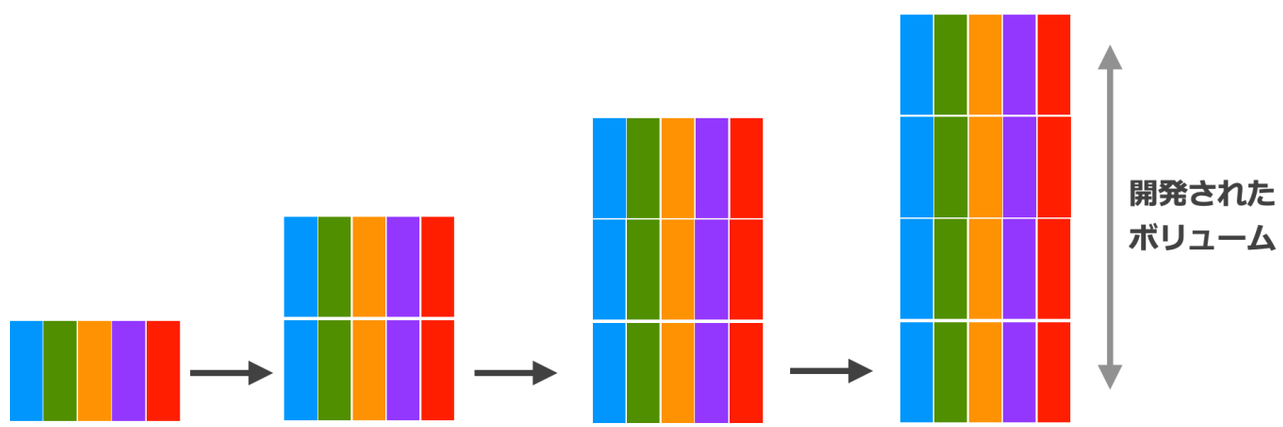
民間では2010-13年あたりに流行

アジャイルな開発のプロセス的な特徴

少しずつ反復的に開発を進めることで
必要とする人から必要なフィードバックを得て
調整し続けられる開発

フェーズゲート開発

アジャイルな開発



これからの2週間分の
やることを決める

毎日開発とコミュニケーション
（設計～開発～テスト）

できたものを確認し
フィードバックする

この2週間をふりかえり
カイゼン点を決める

2週間を繰り返せば
目的のものができるのか？
計画の検証やその為の
各種活動は必要

これからの2週間分の
やることを決める

やること決めるのは
良いけど本当に開発が
できるような状態に
なっているか？

この2週間をふりかえり
カイゼン点を決める

毎日開発とコミュニケーション
（設計～開発～テスト）

開発がどこに着地
できそうか確認する
やるべきことに優先度
をつける

できたものを確認し
フィードバックする

開発者に丸投げして
おくのではなくて、日々
対話して状況と課題の
把握に務める

これからの2週間分の
やることを決める

2週間単位なので

この2週間をふりかえり
カイゼン点を決める

毎日開発とコミュニケーション
（設計～開発～テスト）

早く形になるといっても少しずつ

できたものを確認し
フィードバックする

早く(少しだけ)形にできることに

どんな良さがあるでしょうか？

早く(少しだけ)形にできることの意義

① フィードバックに基づく開発で、目的に適したシステムに近づけていく

② 形にすることで、関係者の認識を早期に揃えられる

③ システム、プロセス、チームに関する問題に早く気付ける

④ チームの学習効果が高い

⑤ 早く開発を始められる

⑥ システムの機能同士の結合リスクを早期に解消できる

⑦ 利用開始までの期間を短くできる

⑧ 開発のリズムが整えられる

⑨ 協働を育み、チームの機能性を高める

形にすることで早めに関係者の認識を揃えられる

- そもそも、関係者(ビジネス側、ビジネスとチーム間、チーム内で)つくるものの解釈が異なっている。
- 最初から、完成型が構想できない。
  誰かが正解を持っているわけではない。
- ドキュメントですべてを書き尽くせない。
  それはコードになる。

「形にするためのコミュニケーションの過程」や
「形にしたものの動き」から共通理解を育める

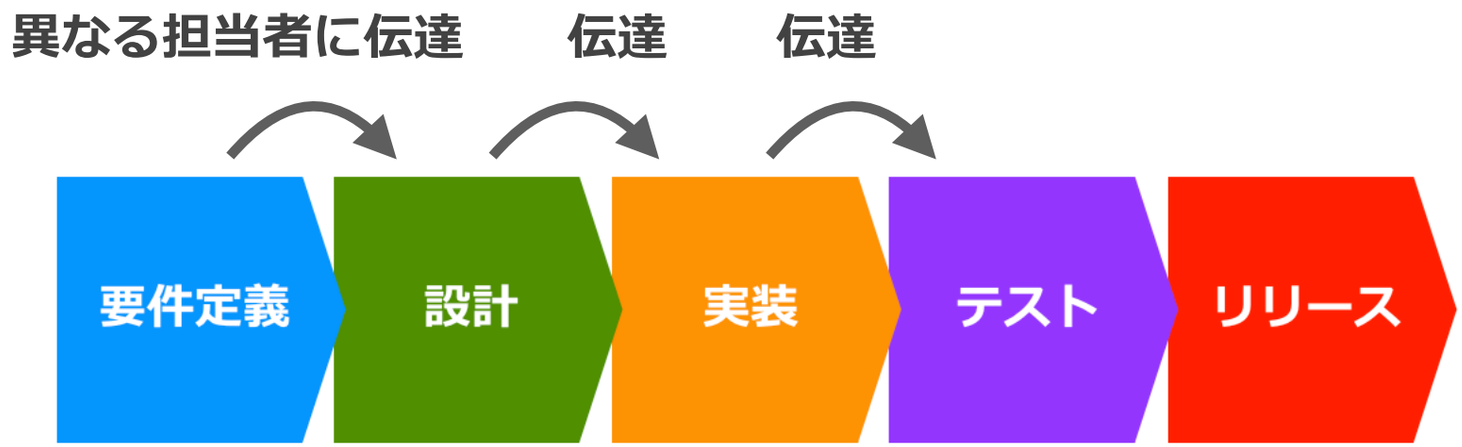
つくるものやチームについての問題早く気付ける

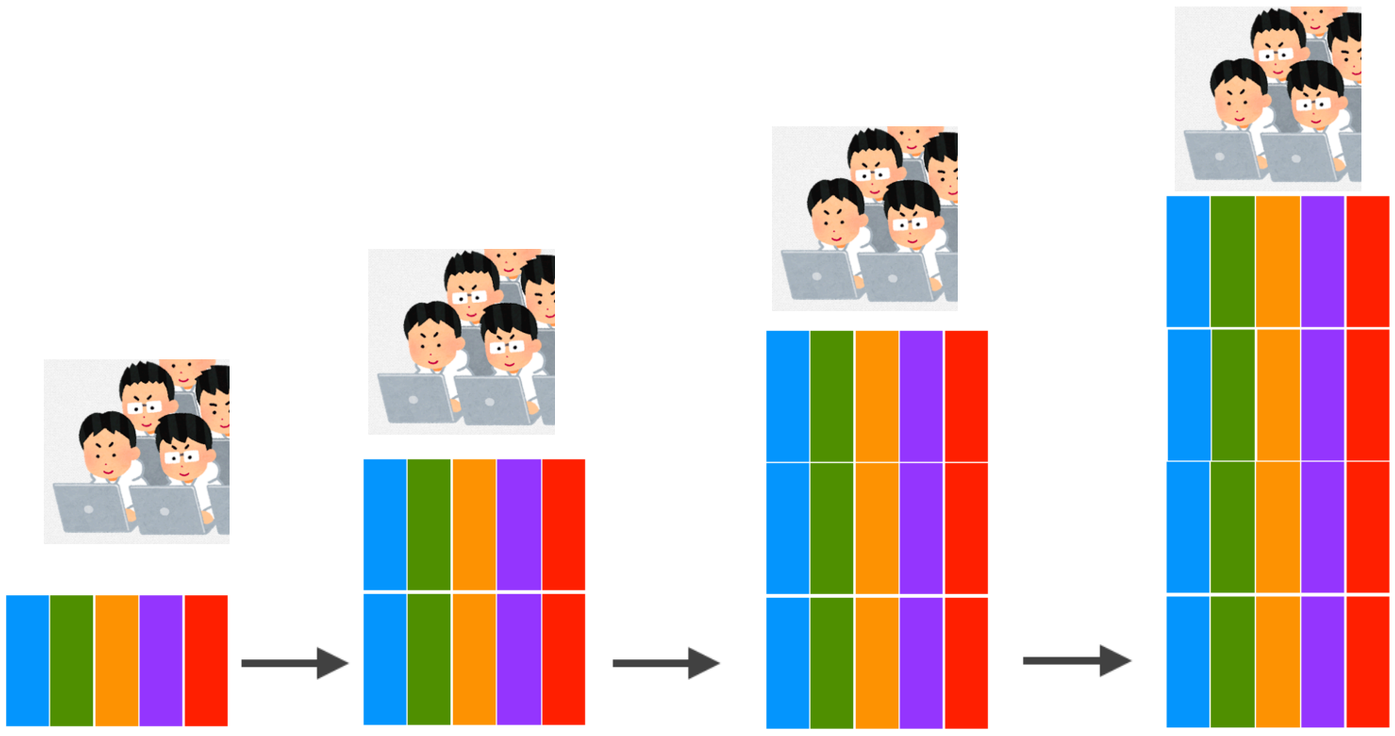
開発を一巡させるのが早いため、プロダクトやチームの問題に早く気付ける。

- ✔ コードが書けない = バックログの完成の条件を詳細にする。
- ✔ バックログが消化できない = チームに必要なタレントを巻き込む。
- ✔ 認識の齟齬が多い = コミュニケーションの内容、頻度の調整。
- ✔ 要件の実現方法が分からない = 技術的な実現可能性の検証。
- ✔ 改修時のデグレが多い = 自動回帰テストの充実。

チームの学習効果が高い

- 「常に最初から同じメンバーで継続的に活動」が前提





早く始められる

全ての要件を決めきる必要がない = 部分を後回しに出来る

結合のリスクを早めに倒すこと

常に結合。

Time to marketが短い

早いスプリントからリリースできる = 早く収益化できる。

サンクコストが小さくできる

早いスプリントから評価が出来るため、途中でやめられる。

開発チームのリズムを整えられる

チームの実状に計画をあわせられる。

2週間を繰り返せば
目的のものができるのか？
計画の検証やその為の
各種活動は必要

これからの2週間分の
やることを決める

やること決めるのは
良いけど本当に開発が
できるような状態に
なっているか？

こういう開発をやっていく ためにはどんな前提が必要か？

開発がどこに着地
できそうか確認する
やるべきことに優先度
をつける

できたものを確認し
フィードバックする

開発者に丸投げして
おくのではなくて、日々
対話して状況と課題の
把握に務める

アジャイル開発の前提

① 常にカイゼンを指向すること

② 対話コミュニケーションの重視（協働という認識）

③ 経験者の参画

④ プロダクトオーナーが時間を使えること

⑤ どのくらいアジャイルな開発でいくのか決めておく

⑥ システムの変更容易性を確保し続ける

アジャイル開発の前提

① 常にカイゼンを指向すること

最初から全ての行為が最適化されているわけではない

② 対話コミュニケーションの重視（協働という認識）

対話しなければ、思っていたものとは違うモノができる

③ 経験者の参画

先に「失敗」を知ることで回避ができる

④ プロダクトオーナーが時間を使えること

でなければ対話もできないし、出来るものも確認できない

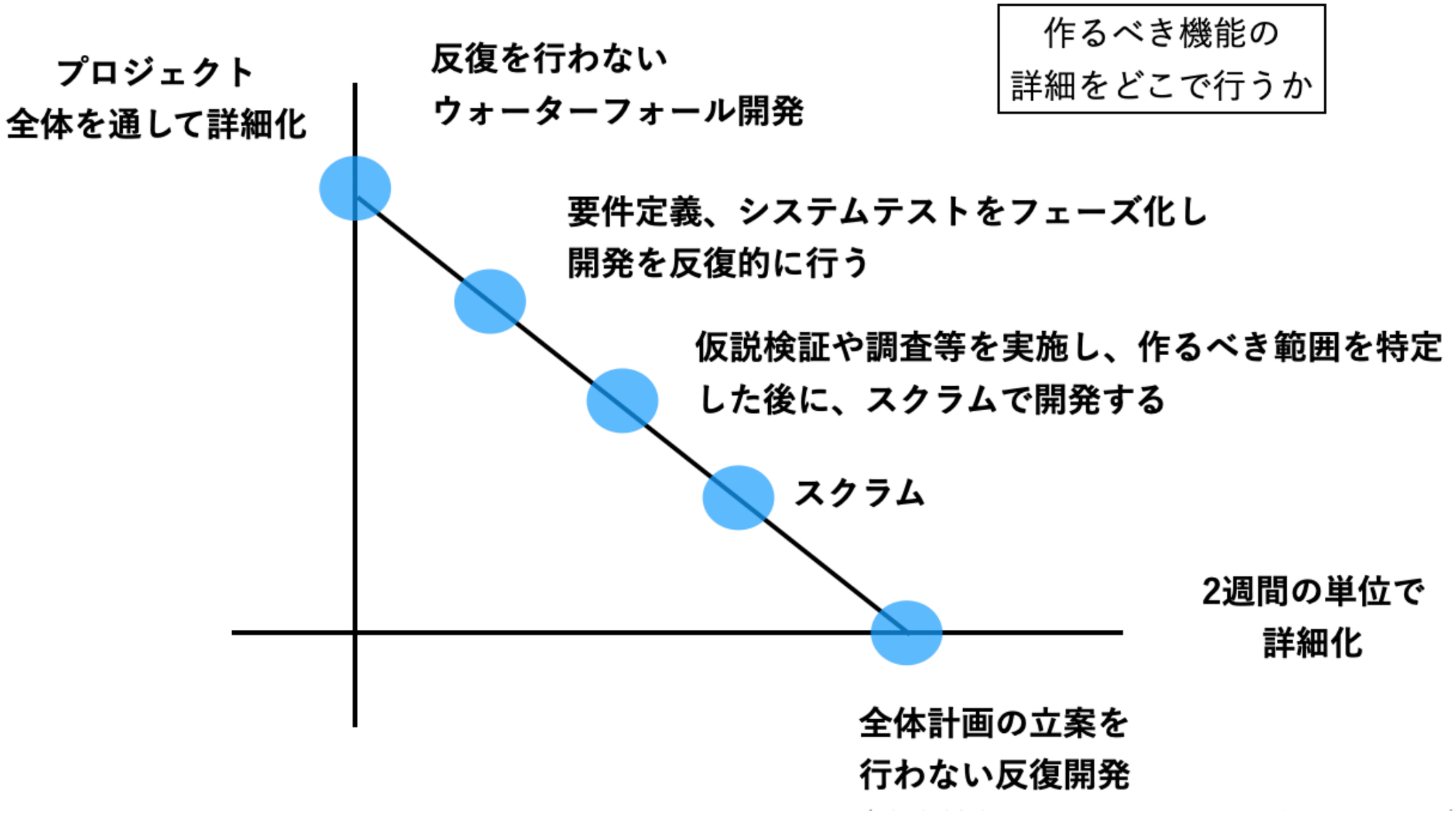
⑤ どのくらいアジャイルな開発でいくのか決めておく

とはいえ、どのくらい探索的にやるのか？認識あわせる

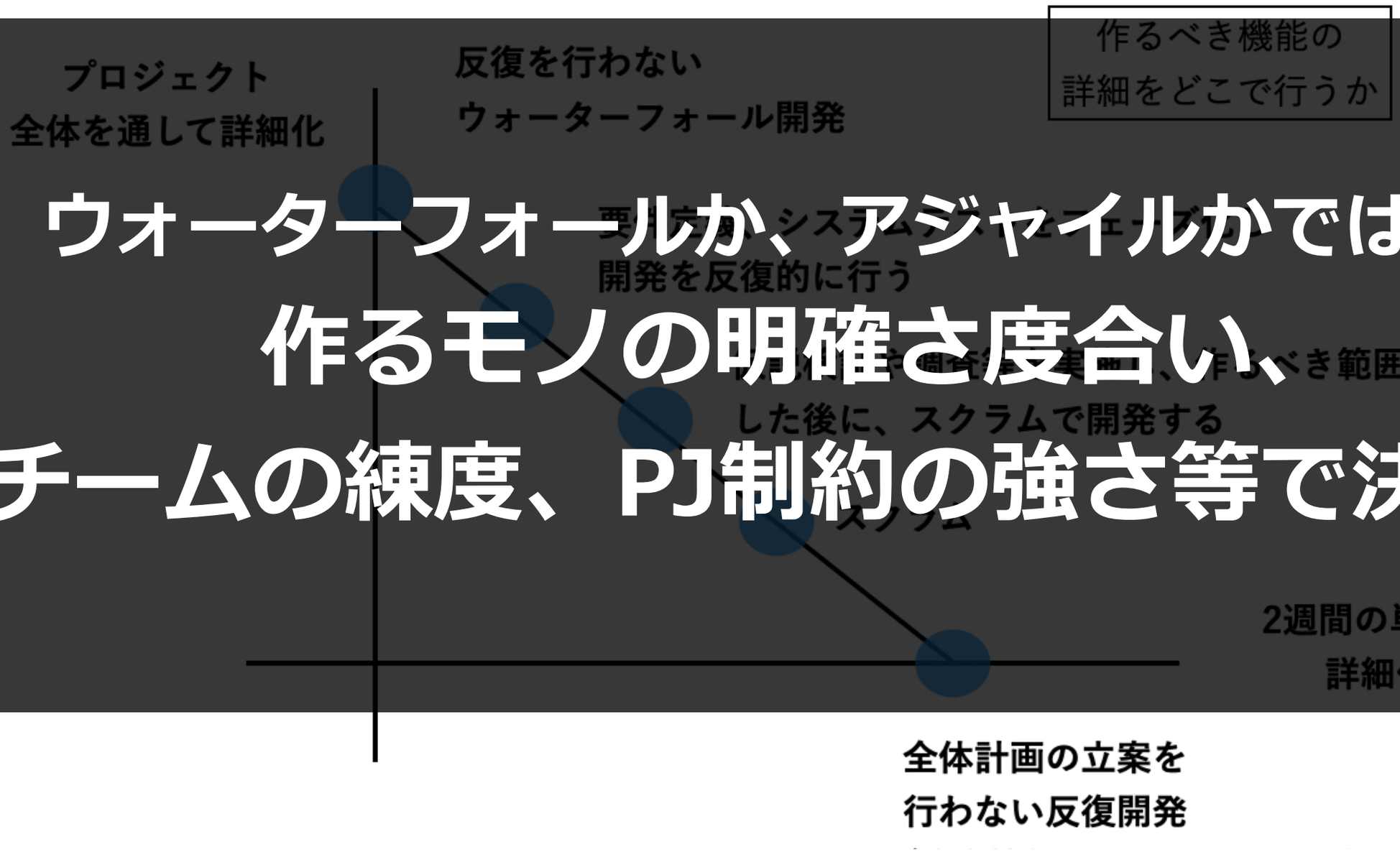
⑥ システムの変更容易性を確保し続ける

開発チームの腕の見せ所（難易度は高い）

どのくらいアジャイルな開発でいくか？



どのくらいアジャイルな開発でいくか？



ウォーターフォールか、アジャイルかではなく
作るモノの明確さ度合い、
チームの練度、PJ制約の強さ等で決める

協働のために最初に必要なこと

= これからやることの認識を揃える

インセプションデッキ

われわれは
なぜここに
いるのか

エレベーター
ピッチ

パッケージ
デザイン

やらないこと
やること
リスト

技術的な
解決策

トレードオフ
スライダー

期間を
見極める

なにが
どれだけ
必要か

プロジェクト
コミュニティ

夜も眠れない
問題

Whyを明らかにする

Howを明らかにする

なぜインセプションデッキ？

「然るべき人が集まっているか」

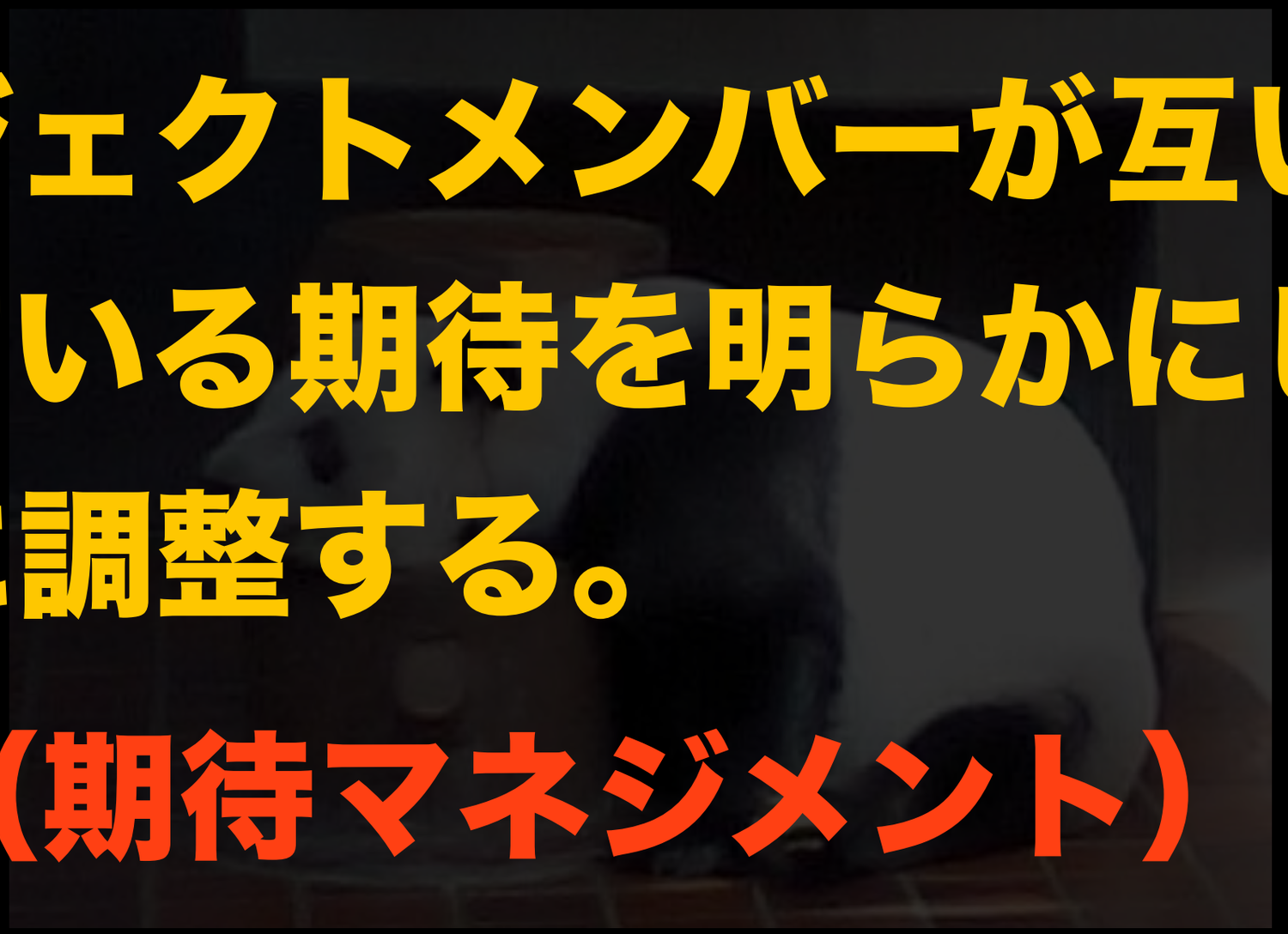
「プロジェクトが然るべき方向を向いているか」

を明らかにする

“チームメンバーが誰もいないところで合意したことを前提にしているから、プロジェクトがダメになる”

アジャイルサムライP44

“プロジェクトを始める前に
関係者の認識を合わせる”とは？



プロジェクトメンバーが互いに
持っている期待を明らかにし
適切に調整する。

（期待マネジメント）

我われはなぜここにいるのか

・大事な理由その1

・大事な理由その2

・大事な理由その3

<このプロジェクトの根幹に
関わる理由を1つ、ここに書く

絶対に実現しなければいけないこととは？

エレベーターピッチ

- [潜在的なニーズを満たしたり、
  潜在的な課題を解決したり] したい
- [対象顧客] 向けの、
- [プロダクト名] というプロダクトは、
- [プロダクトのカテゴリー] です。
- これは [重要な利点、対価に見合う説得力のある理由(プロダクトにお金を払いたい理由)] ができ、
- [代替手段の最右翼] とは違って、
- [差別化の決定的な特徴] が備わっている。

誰の何をどうやって解決するのか?

やらないことリスト

やる	やらない

あとで決める

スコープに何が入らないのか?

インセプションデッキ

われわれは
なぜここに
いるのか

エレベーター
パッケージ
デザイン

やらないこと
やること
リスト

技術的な
トレードオフ
スライダー

期間を
見極める
なにが
必要か

プロジェクト
コミュニティ

夜も眠れない
問題

Whyを明らかにする

Howを明らかにする

意外と内容がうすそう…

その通りです。

ゼロから協働するチームを立ち上げるためには
このレベルから合わせる必要があります。

(とはいえデッキの中身は必要なことばかり

インセプションデッキ

エレベーター

技術的な

期間を
見極める

われわれは
なぜここに
いるのか

やらないこと
やること

パッケージ

トレードオフ

なにが
どれだけ

前提は一緒にやること

パワポ書いたので見て置いて下さい…

ではなく

デッキのテーマを**その場で一緒に考える**ことで

合わせる認識の解像度を上げる

(言語化してみること、その過程で認識を交換することで理解が深まる)

プロジェクト
コミュニティ

夜も眠れない
問題

Whyを明らかにする

Howを明らかにする

どのようにして学んでいくのか

チームの中で、そしてチームの外との間で

共通の語彙を手に入れよう

いちばんやさしい “アジャイル・ジャーニー”

1段階目 **勉強会・研修** 1ヶ月-2ヶ月間

・まずは基本的な語彙習得のための機会を作る
・研修のていで組織内外からメンターを招くのがてっとり早い
・諸事情で研修が組めない場合は勉強会を開こう

2段階目 **サンドボックスPJ** 2ヶ月-3ヶ月間

・チームメンバー以外の関係者が少ない or いないPJで始める
・途中で終わったとしても全く問題がない状況を選ぶ
・諸事情で業務にならない場合は野良プロジェクトを始める

3段階目 **小規模PJ** 3ヶ月-6ヶ月間

・長期間に及ぶ規模の大きなPJは勿論避ける。いざとなれば
途中からいつものやり方で巻き返せるような小さなPJを選ぶ。
・メンターを招聘し「実践から学ぶ」ことを念頭に置く

Start with Why

価値観

原則

行動

何のために(価値観)、どうやって(原則)、
やるか(行動)
だから、価値観をあわせるのが大事！
…大事だけど、容易ではない。

行為から学ぶ

価値観

原則

行動

型から入って、実際にやってみる
やってみる(行動)過程と結果から、
何が大事なのかを学び直す(原則)
その学びをより良くするためには？(価値観)

Social Change (自分から変わる)

思えばXPとは、分断を認めそれを乗り越えるための価値観であった

コミュニケーション　　**シンプリティ**　　**フィードバック**

勇気　　**リスペクト**

	リモート (状況や感情が見えにくい)	DX (組織的な分断、前提や方針の不整合)
コミュニケーション	限られた接点ゆえ より重要	分断を直接的に つなぎ直す
シンプリティ	シンプルでないと 伝わりきらない	複雑なしがらみを 単純な方向へ
フィードバック	意識的に伝えないと 良くならない	実験や試行による 学習を適用する
勇気	自分から踏み込み 切欠を作る	変えるという行為には 勇気を必要とする
リスペクト	相手が見えない中で リスペクトがなければ 疑心暗鬼になる	対立軸を作って争う のではなく、相手の イマココの状況を尊重する

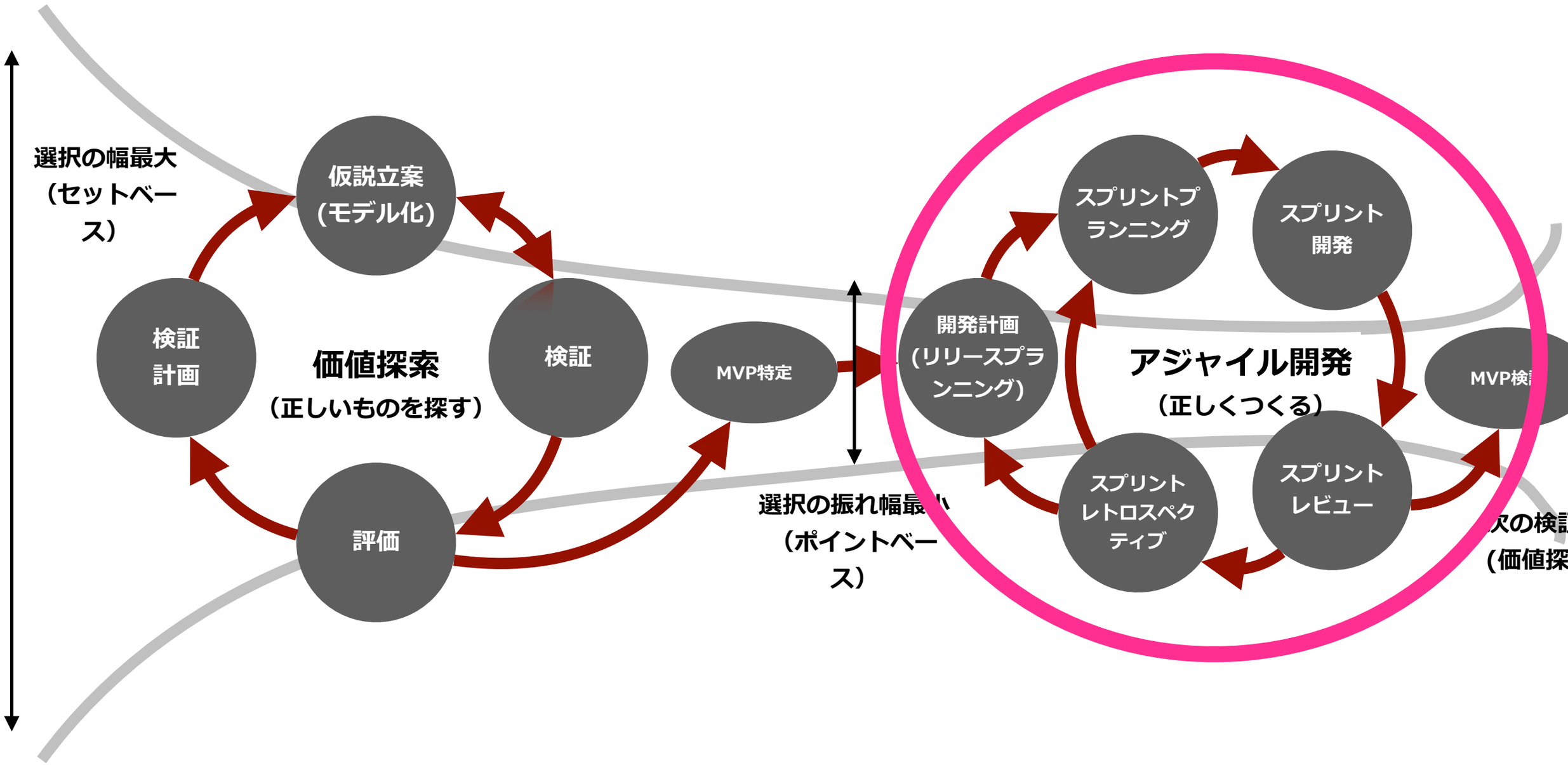
アジャイルによって

アジャイルになる

Be agile by Agile

アジャイル開発をはじめよう

仮説検証型アジャイル開発

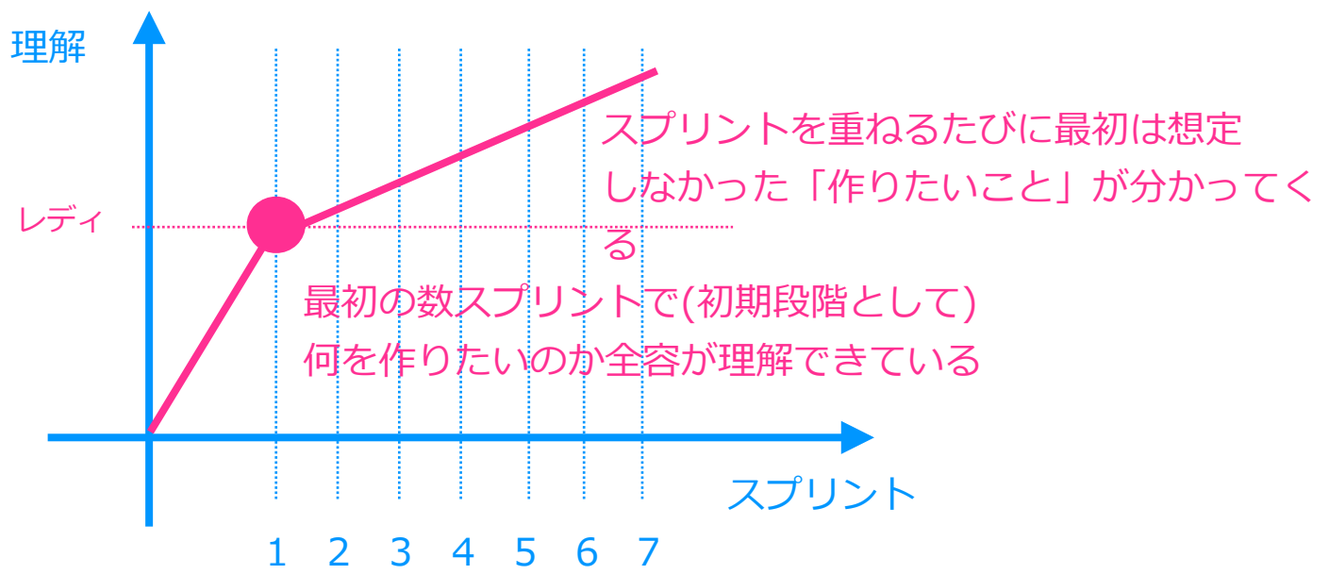


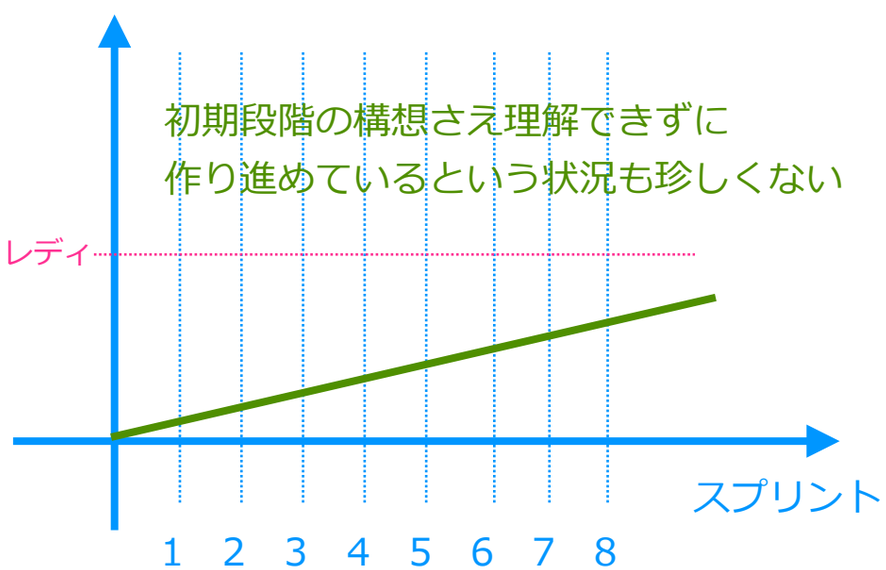
スプリント開発をはじめられるようにする

レディ(準備OK)にする観点は2つ

- **プロダクト**
  (初期段階として) 作るものが何か分かっている
- **チーム**
  お互いが何者かわかっていて、どうやって作るかが分かっている

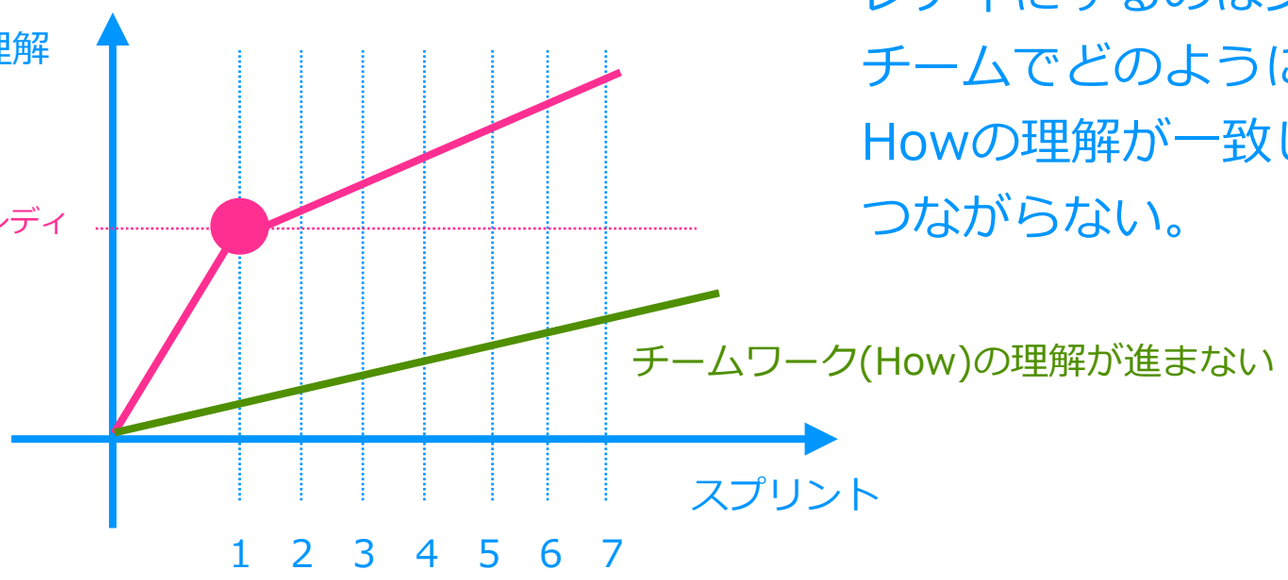
プロダクトのレディイメージ(理想)





2つのレディ

プロダクトのレディイメージ(理想)

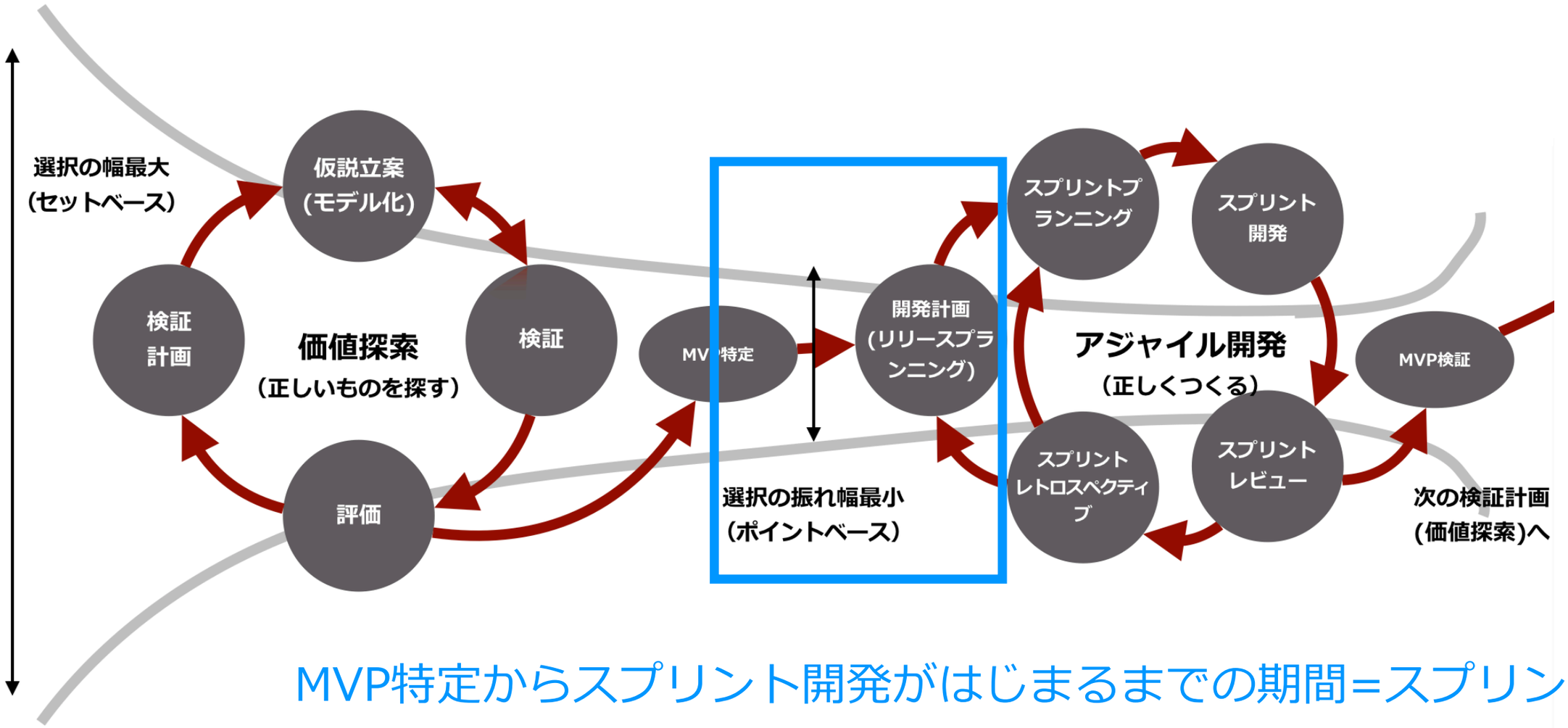


レディにするのはプロダクトの理解(What)だけではない。
チームでどのように仕事を進めるのか(How)という理解も必要。
Howの理解が一致しないと、個々の活動が噛み合わず、成果につながらない。

ゆえに、
①プロダクトについては「ソフトウェアとして何を実現すればよいのか」の言語化
②チームについては一緒に仕事を実際に行うことでやり方をあわせる
ことが必要。

プロダクトの理解をレディにする①

ソフトウェア的に何を作るべきかの情報を整理し、理解する



MVP特定からスプリント開発がはじまるまでの期間=スプリントゼロ

プロダクトの理解をレディにする②

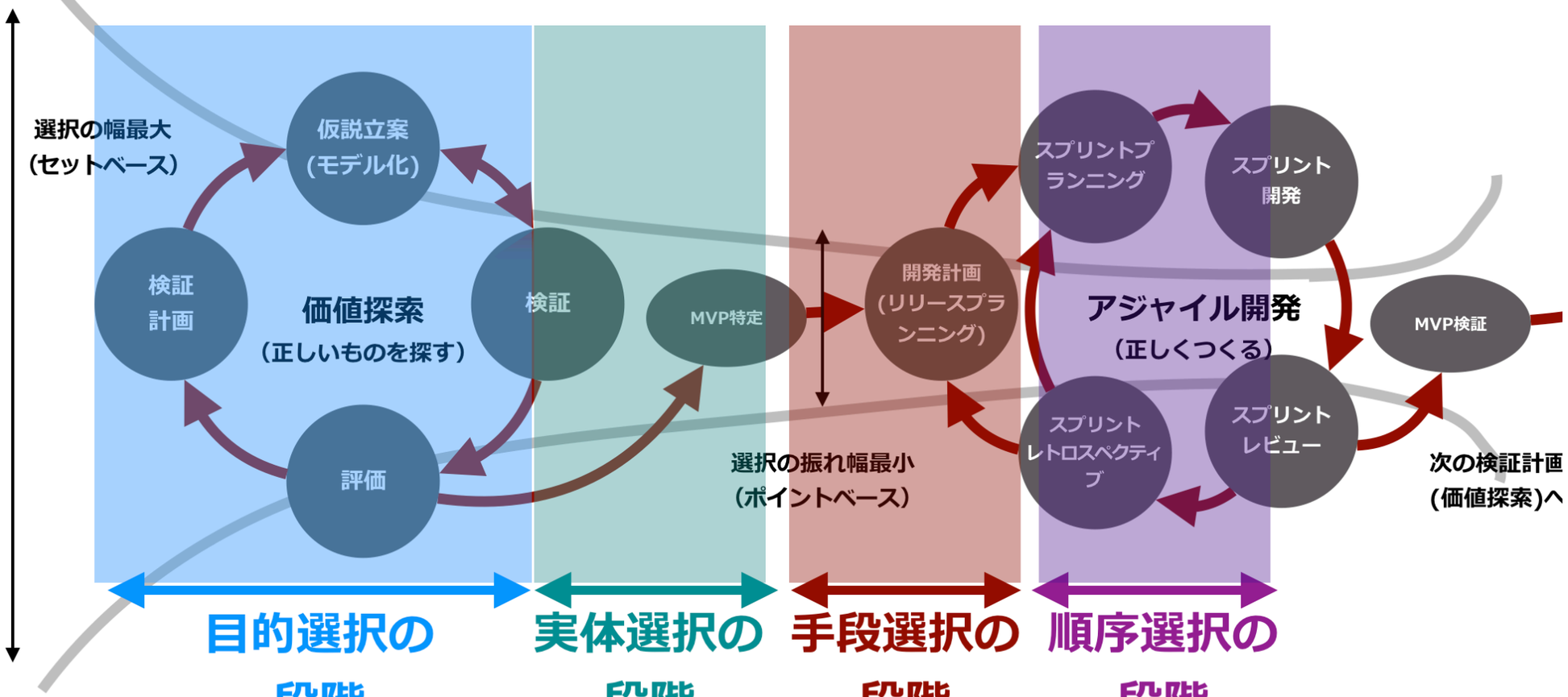
整理する対象の情報は「全体」「部分」2側面ある

プロダクト全体に影響を及ぼす情報 (プロジェクトやテーマ単位で検討)	部分を開発するために必要な情報 (スプリント単位で検討)
・初期段階のPBL及びリリースプランニング ・アーキテクチャ設計 ・ドメインモデル(どういう知識を扱うか)の特定 ・上記を踏まえて、UIとしての情報設計・構造設計、概念レベルのデータ設計	・スプリントバックログ(SBL) ・SBLの実装に必要な機能やデータの設計 ・各ページのレイアウト設計

全体に影響を及ぼす情報の整理を誤ると、後々の影響が大きくなる
(部分を作り重ねていくことで分かってくる「全体」もありうる)

プロダクトの理解をレディにする③

選択を”段階”にすることで不確実性を対処する



コンセプトの検証

ユーザーにとって有用かつ必要最小限の範囲の機能特定

ドメイン特定
情報設計
構造設計

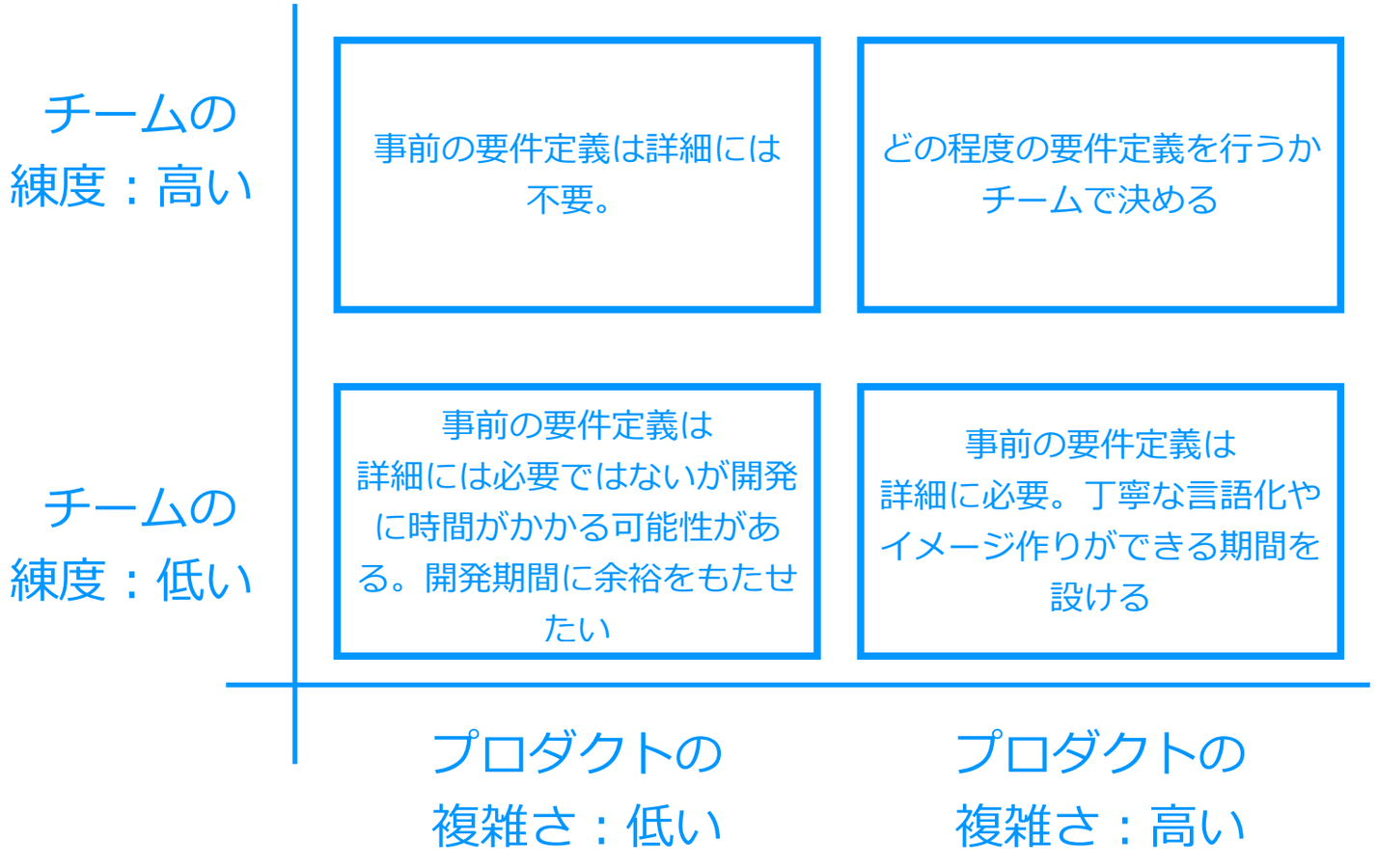
プロダクトバックログのリファインメント

例えば、手段選択の段階でコンセプトを変える決定の影響の大きさは明らか

プロダクトの理解をレディにする④

どこまでスプリントゼロで詳細化するのか？

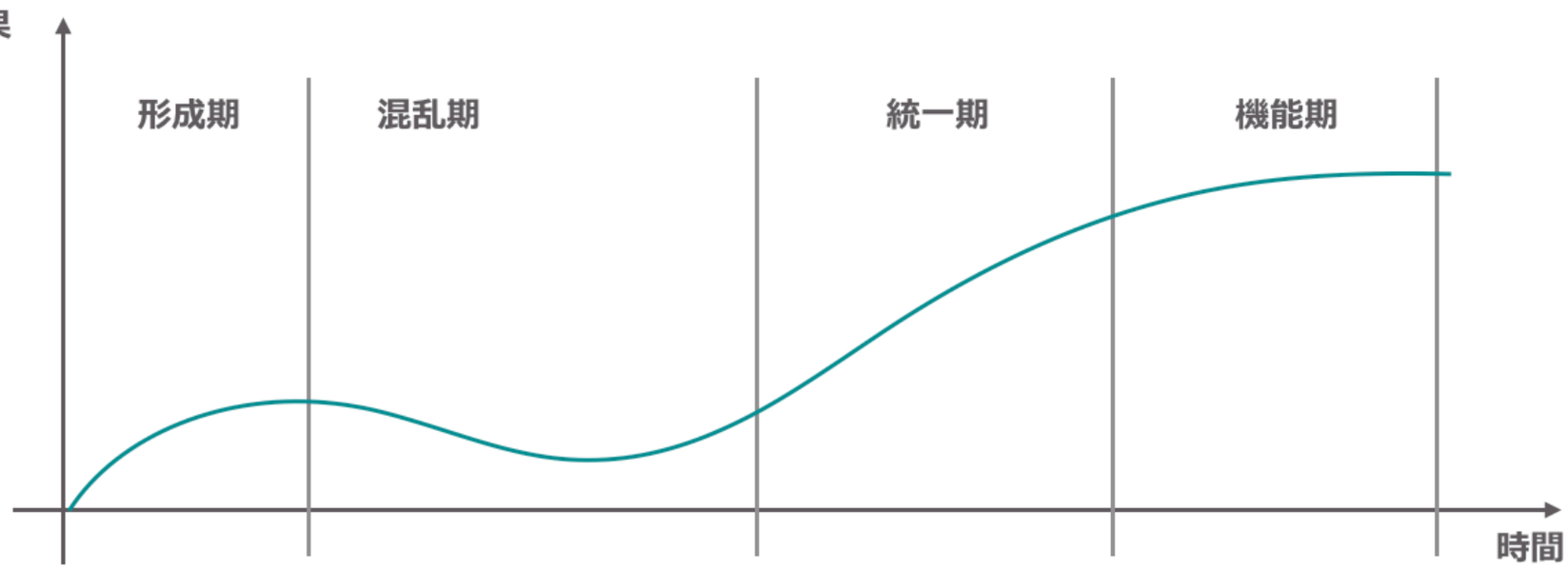
チームの練度とプロダクトの複雑さによって決まる



チームの理解をレディにする①

チームビルドのための基本概念「タックマンモデル」

タックマンモデル

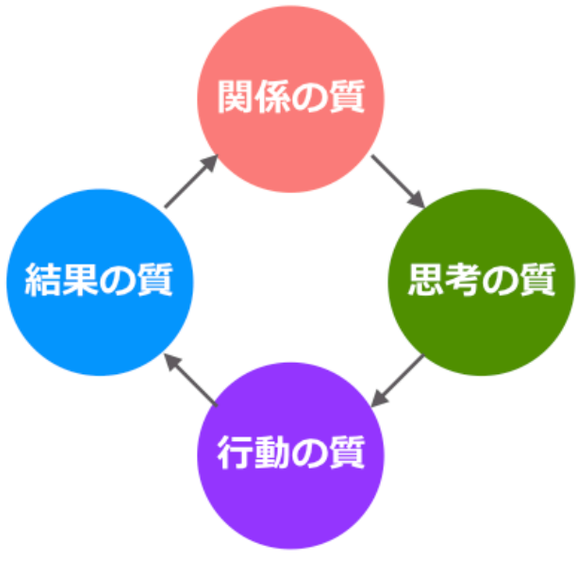


チームを形成していくプロセスには「段階」がある。
①形成期(Forming)　お互いのことを知らない段階。目標も不明瞭。
②混乱期(Storming)　お互いの役割・考え方で意見が生まれて対立する。
③統一期(Norming)　行動規範や役割が確立し他人の考え方が受容できる。
④機能期(Performing)一体感が生まれ、目標達成に向かう状態。

チームの理解をレディにする②

チームビルドのための基本概念「成功循環モデル」

成功循環モデル



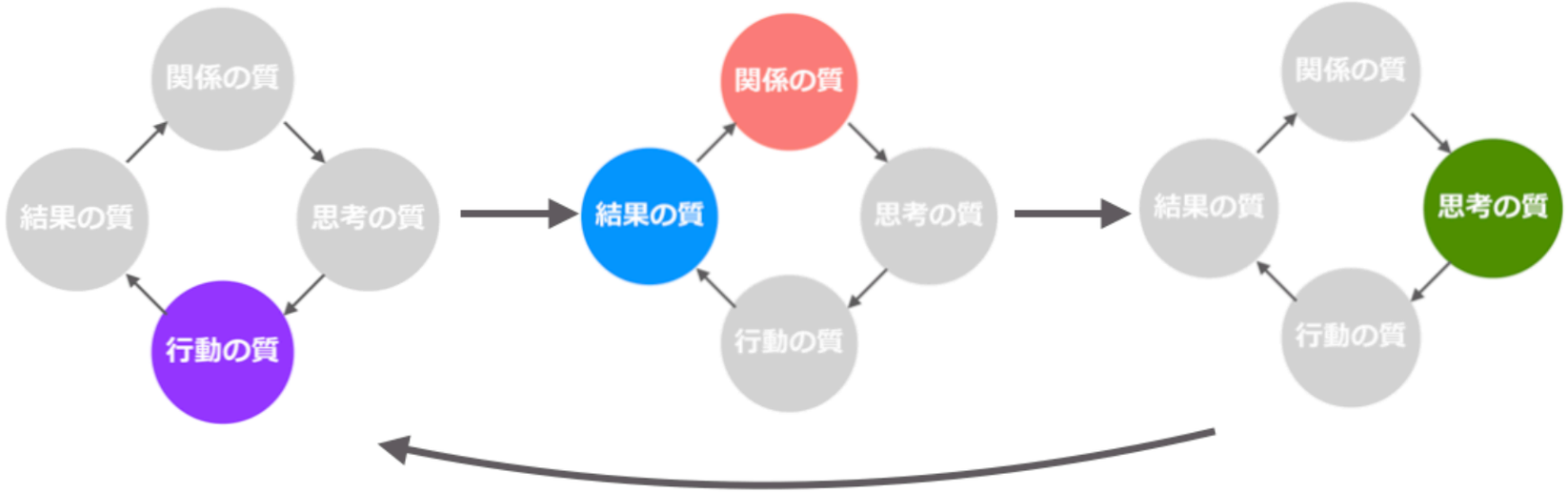
成功循環モデルでは、成功や成果といった“結果の質”を高めるためには、まず、メンバー間の”関係の質”を高めるべき、という考え方に立っている。

関係性が良くなれば、個人の”思考の質”も向上する。思考の質が高まれば”行動の質”も良くなり、結果の質が向上していく。

逆に、結果が伴わないために対立・押し付けが増えて、関係の質が悪化することもある。関係が悪くなれば、協調に欠け、思考の質が低下する。積極的に動かなくなり、行動の質の低下に繋がり、結果が出ない…というサイクルが考えられる。

チームの理解をレディにする③

基本は仕事を「小さく、短く、一巡させる」



さっそくチーム仕事を小さく、短く、一巡させる

一巡目の”行動の質”は低いが
早く結果を出すことを優先する

小さな成功体験から関係性を高める

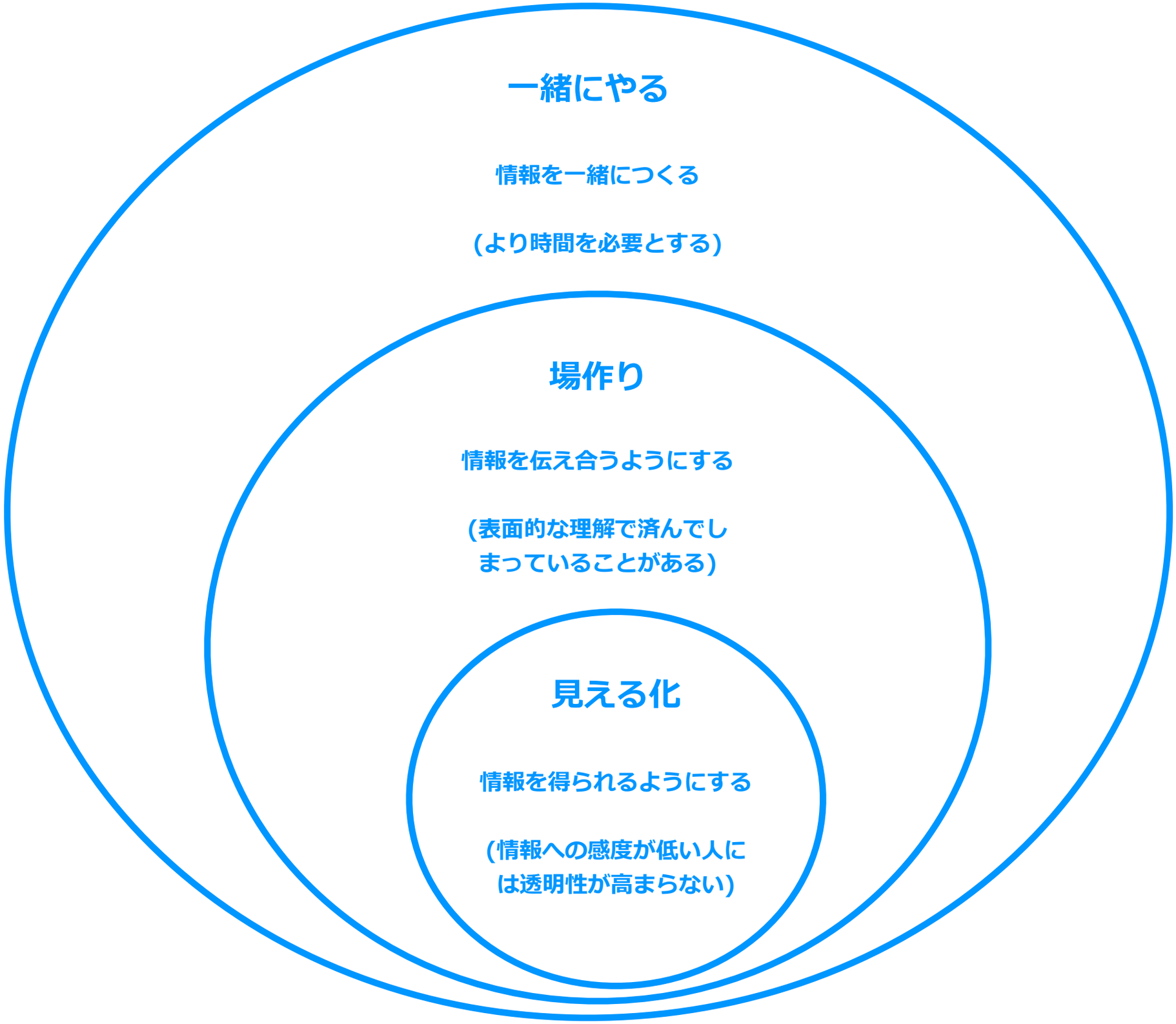
一巡目の結果は大したもの
ではない。だが、ゼロから
イチが生み出されたのは事実

ふりかえりによってさっそく改善を始める

一巡の結果分かったことから
次にやることを定める。
二巡目の行動の質を高める。

このサイクルを繰り返し、巡回毎に質を高める。

共通理解が
より深まる


一緒にやる
情報を一緒につくる
(より時間を必要とする)
場作り
情報を伝え合うようにする
(表面的な理解で済んでし
まっていることがある)
見える化
情報を得られるようにする
(情報への感度が低い人に
は透明性が高まらない)

共通理解が
より深まる

一緒にやる

情報を一緒につくる

(より時間を必要とする)

場作り

情報を伝え合うようにする

(表面的な理解で済んでし
まっていることがある)

見える化

情報を得られるようにする

(情報への感度が低い人に
は透明性が高まらない)

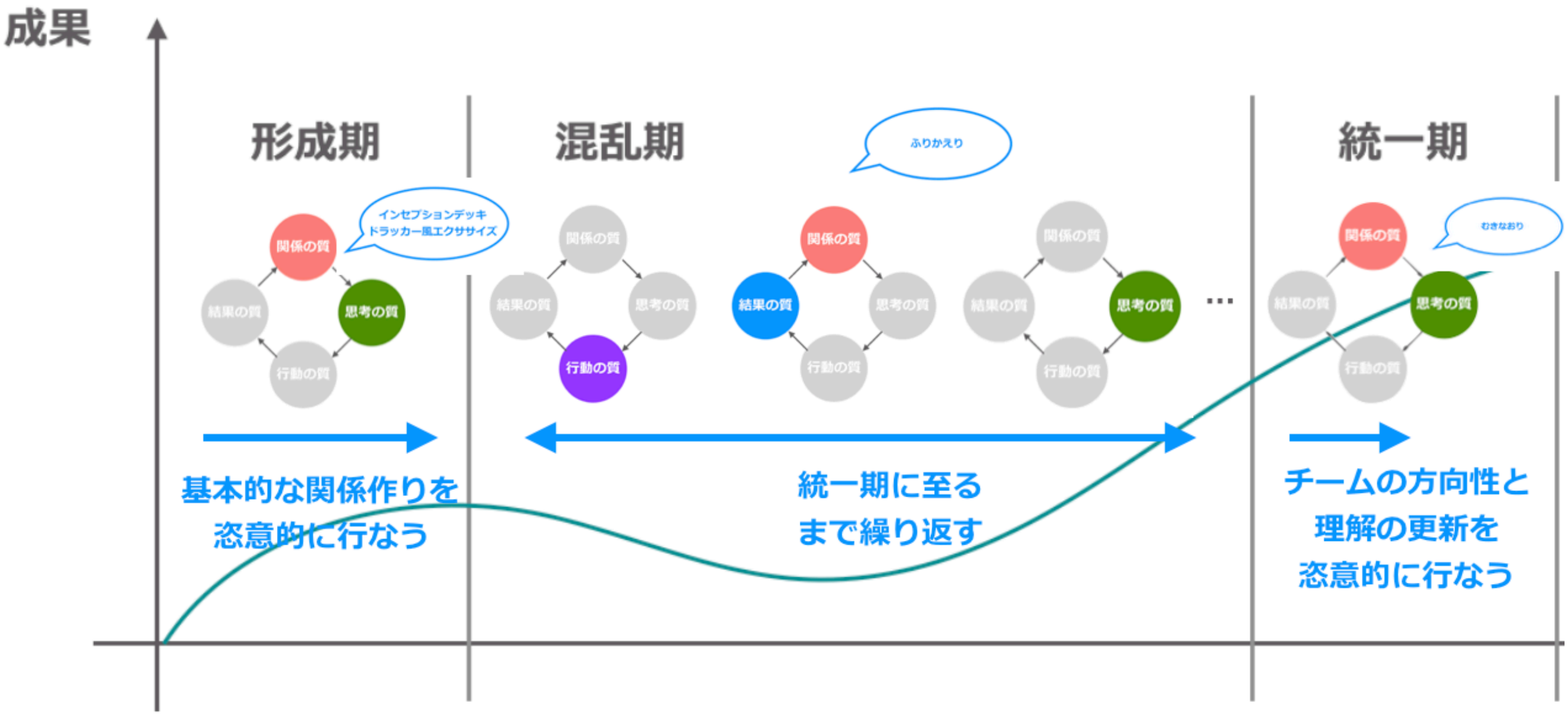
一つのタスクを一緒に
やることで理解を深める
(ペア、モブ)

同期イベントを開催し
状況及び意思決定を
協働で行う

ツールを使って
情報を見えるようにする
(共同所有する)

チームの理解をレディィにする④

タックマンモデルX成功循環モデル



チームの理解をレディにする⑤

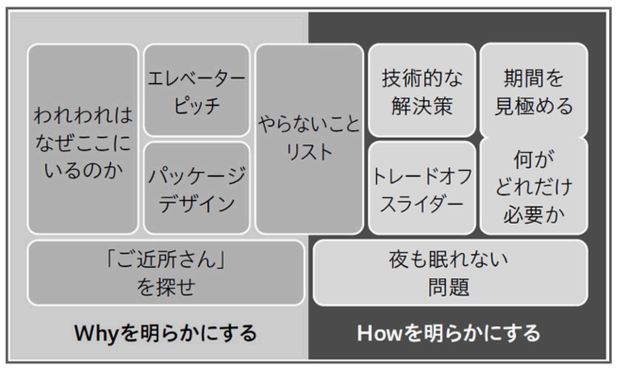
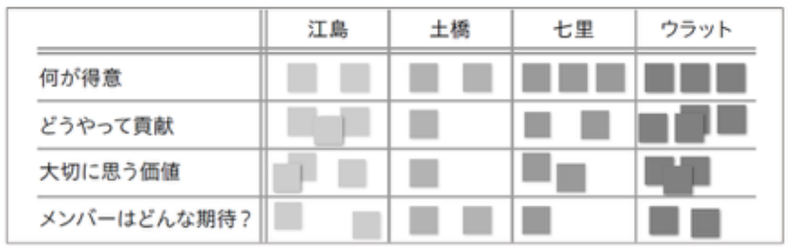
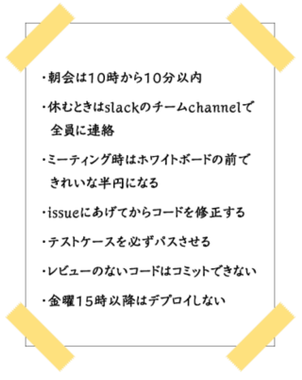
非日常と日常を使わけるチームビルディング

「小さく、短く、一巡させる」は**日常**におけるチームビルド。
形成期においては意図的なチームビルドも理解を促すためには必要。
非日常のチームビルドとして、Workshopを意識的に開催する。

共通理解を育むための3つの方法

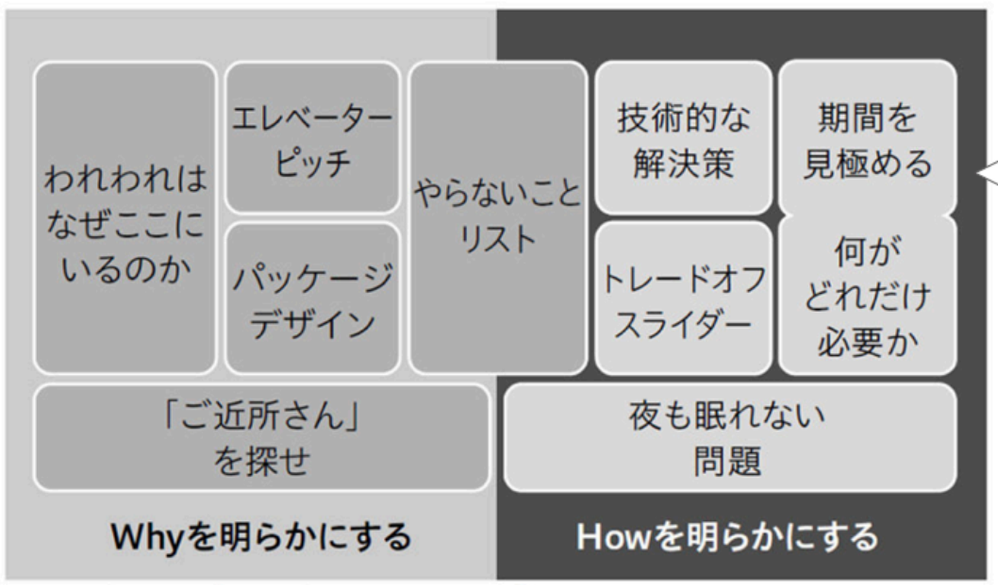
・インセプションデッキ
・ドラッカー風エクササイズ
・ワーキングアグリーメント

チームの理解をレディにする⑥

	理解するべきこと	方法
プロジェクト・プロダクトレベル	目的や目標、 前提や制約、 優先基準	 インセプションデッキ
チームメンバーレベル	考え方や 得意技	 トラッカー風エクササイズ
チーム活動レベル	共同のための ルール、約束事	 ワーキングアグリーメント

チームの理解をレディにする⑦

インセプションデッキ



・10個の問い(アジェンダ)で構成されている
・根幹を問うWhy寄りの問いと、
実現手段を問うHow寄りの問いの2部構成
・チーム関係者全員が一同が会して、問いに
向き合い、答えていく
・回答結果ははプロジェクトルームに張り出す
などしていつでも目につくようにする。

①われわれはなぜここにるのか？
プロジェクトやプロダクトで達成したい目的、
目標を挙げる
②エレペータピッチ
端的にプロダクトの特徴を言語化する。
解決する問題／対象顧客／重要な利点
代替手段／差別化要因など
③パッケージデザイン
顧客へ伝えたいメッセージとビジュアルイメージ
④ご近所さんを探せ
プロジェクト関係者の図示化
⑤やらないことリスト
やらないと明確に決めている一覧
⑥夜も眠れない問題
プロジェクトのリスクを挙げる
⑦トレードオフスライダー
品質／予算／リリース日／品質などどの基準を
優先するか
⑧技術的な解決策
利用する技術やアーキテクチャの図示化
⑨基幹を見極める
プロジェクトに必要な期間の見立て
⑩何がどれだけ必要か
費用、期間、チームなど必要なリソースの定量化

チームの理解をレディにする⑧

ドラッカー風エクササイズ

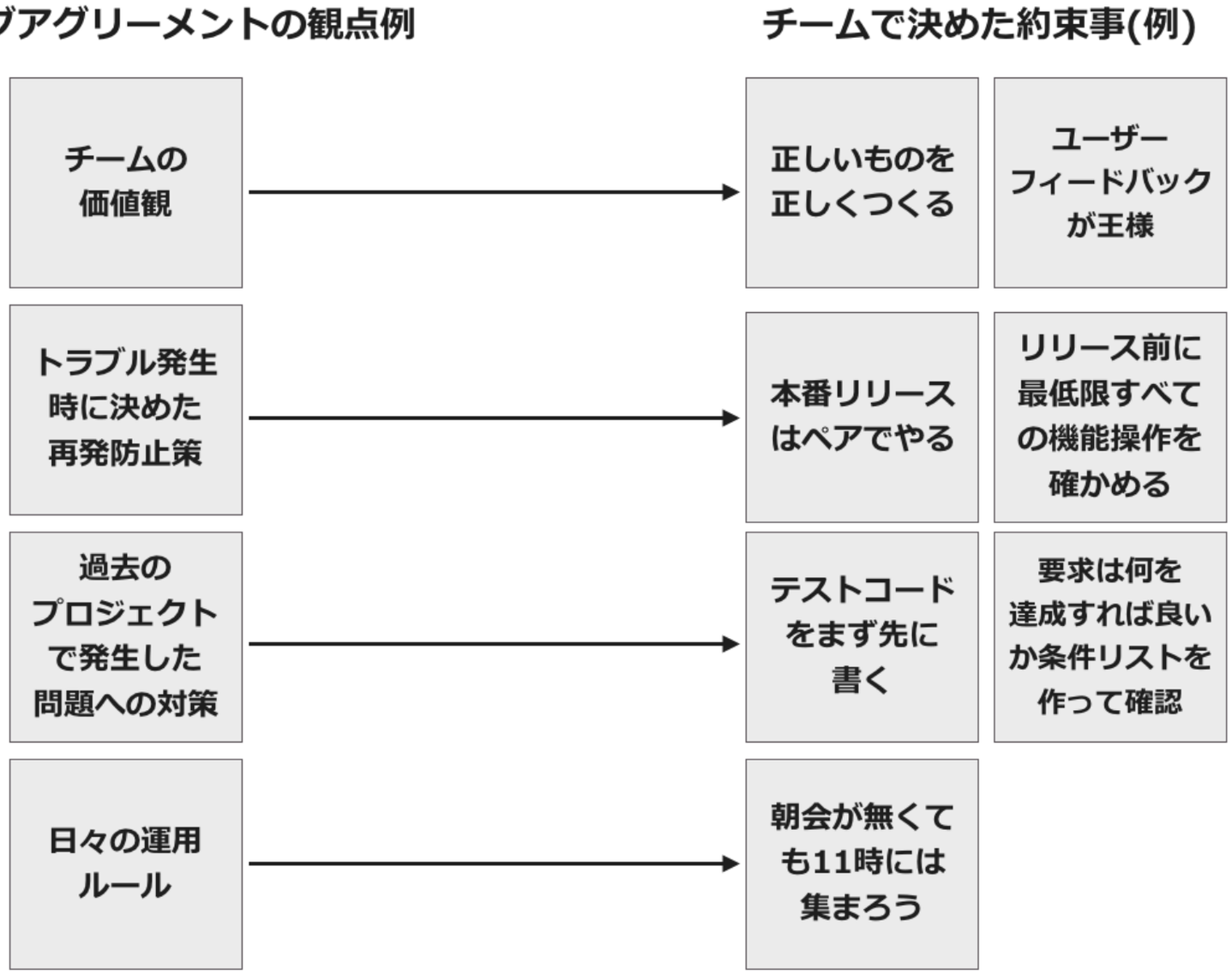
チームビルディングの一環として実施すると良い。
4つの質問にそれぞれが答えた後に、全員で眺めて認識違いや感想をフィードバックする。
特に4つ目の「成果期待」に対する自己認識を正し、チームで合わせるのが狙い。

	稲村	由比	音無	長谷
自分は何が得意なのか？				
どういうふうに仕事をして貢献するつもりか？				
自分が大切に思う価値は何か？				
チームメンバーは自分にどんな成果を期待していると思うか？				

チームの理解をレディにする⑨

ワーキングアグリーメント

ポリシーのような価値観や具体的な行動指針、または日々実施するタスクレベルのルールなど
どのようなものでも良い。
チームの関心事、対立やトラブルが発生していることを中心に、意識合わせしておくことが重要。
多すぎても運用できないため、全体で7つ程度を目処とする（適宜入れ替え、内容を改める）。



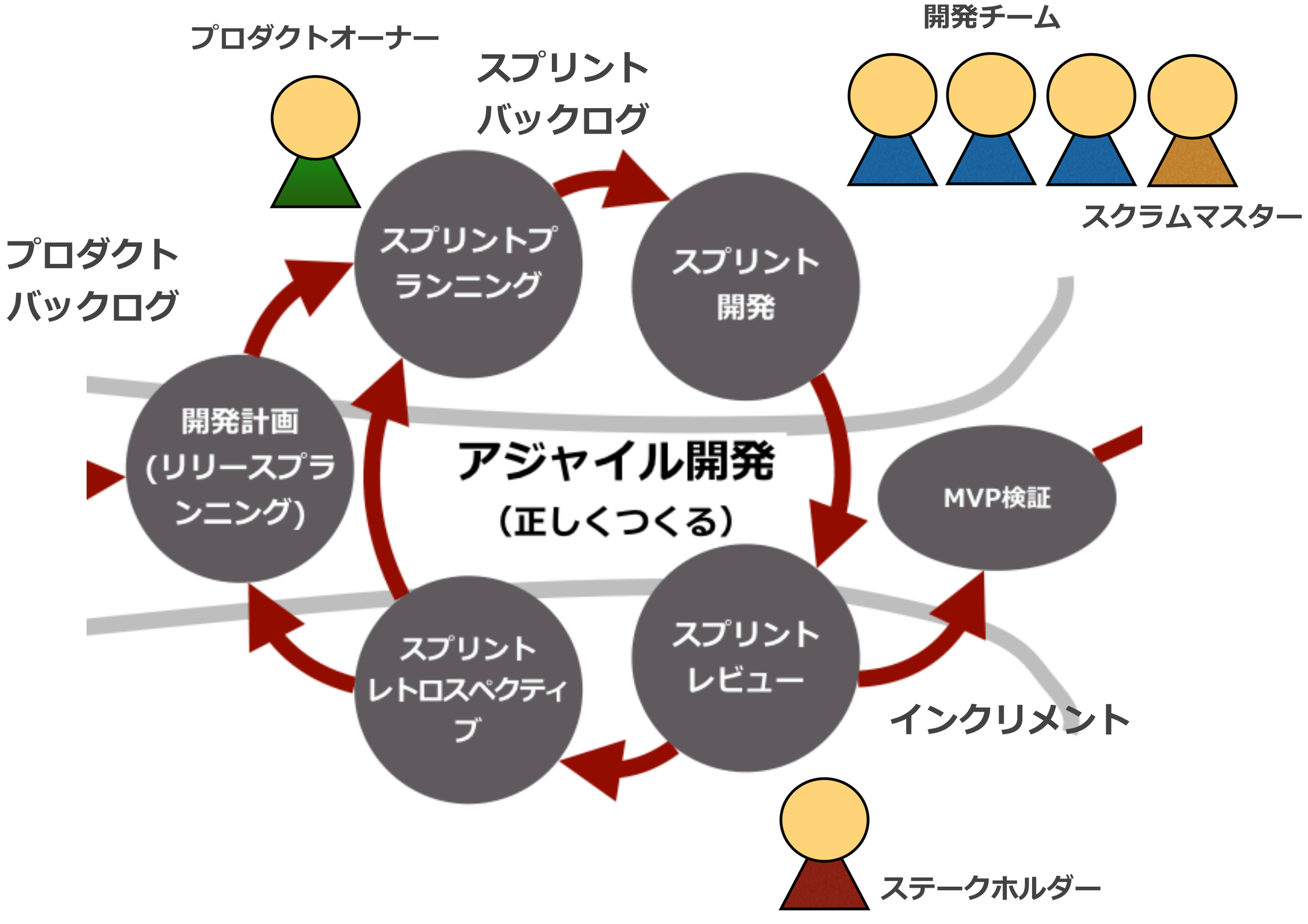
「スプリント1」が基礎になる

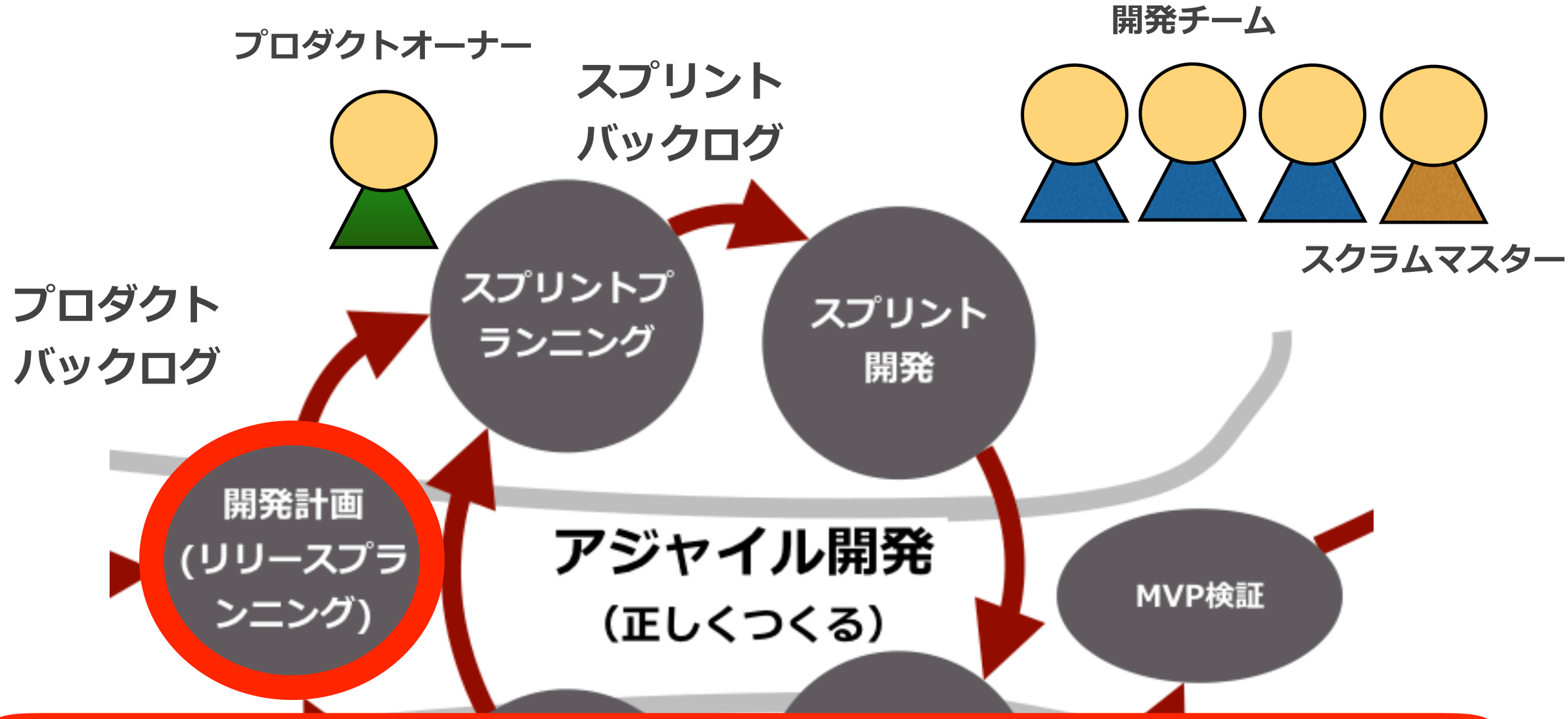
プロダクトやチームの理解が十分に深まったか計測する手段
＝「スプリント1の結果」

プロダクトやチームの準備を進めて臨んだスプリント1。
その結果が、思うようなものにならなかったときにどう捉えるか？
「最初だから、仕方ない。そのうちよくなっていくだろう」では成り行き任せ。
スプリントゼロの活動の結果としての「プロダクト」と「チーム」が存在していると捉えて、どういうインプットや時間が足りていないのかを分析すること。

例えば…
アウトプットがほとんどなかった ＝ PBLの詰め方が甘いのかもしれない
スクラムイベントがピリッとしない ＝ スクラムへの基本的な理解が足りない
デモがまともにできない ＝ 完成の定義が無いあるいはずれている

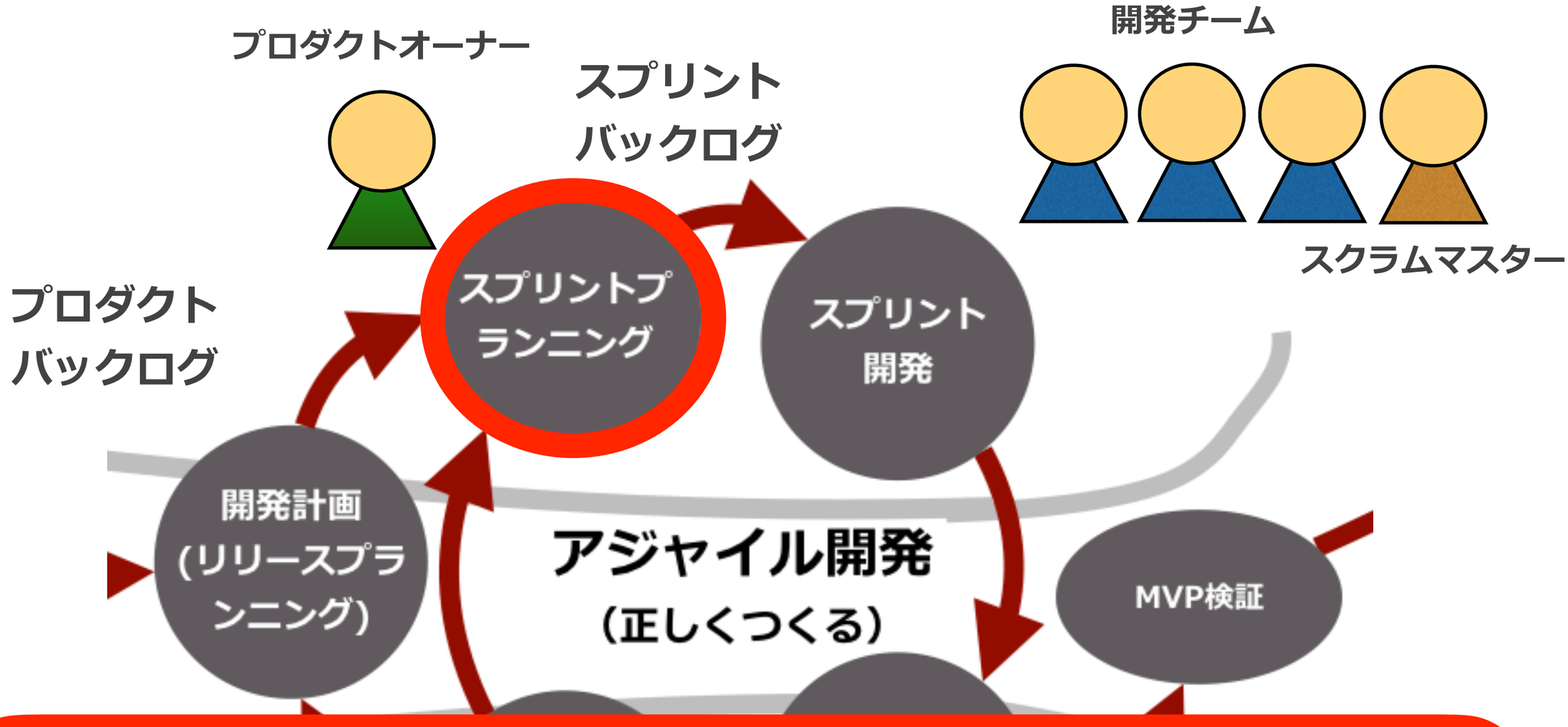
スクラムイベントを
はじめよう


プロダクトオーナー
スプリント
バックログ
開発チーム
スクラムマスター
プロダクト
バックログ
スプリントプ
ランニング
スプリント
開発
開発計画
(リリースプラ
ンニング)
アジャイル開発
(正しくつくる)
MVP検証
スプリント
レトロスペクティ
ブ
スプリント
レビュー
インクリメント
ステークホルダー



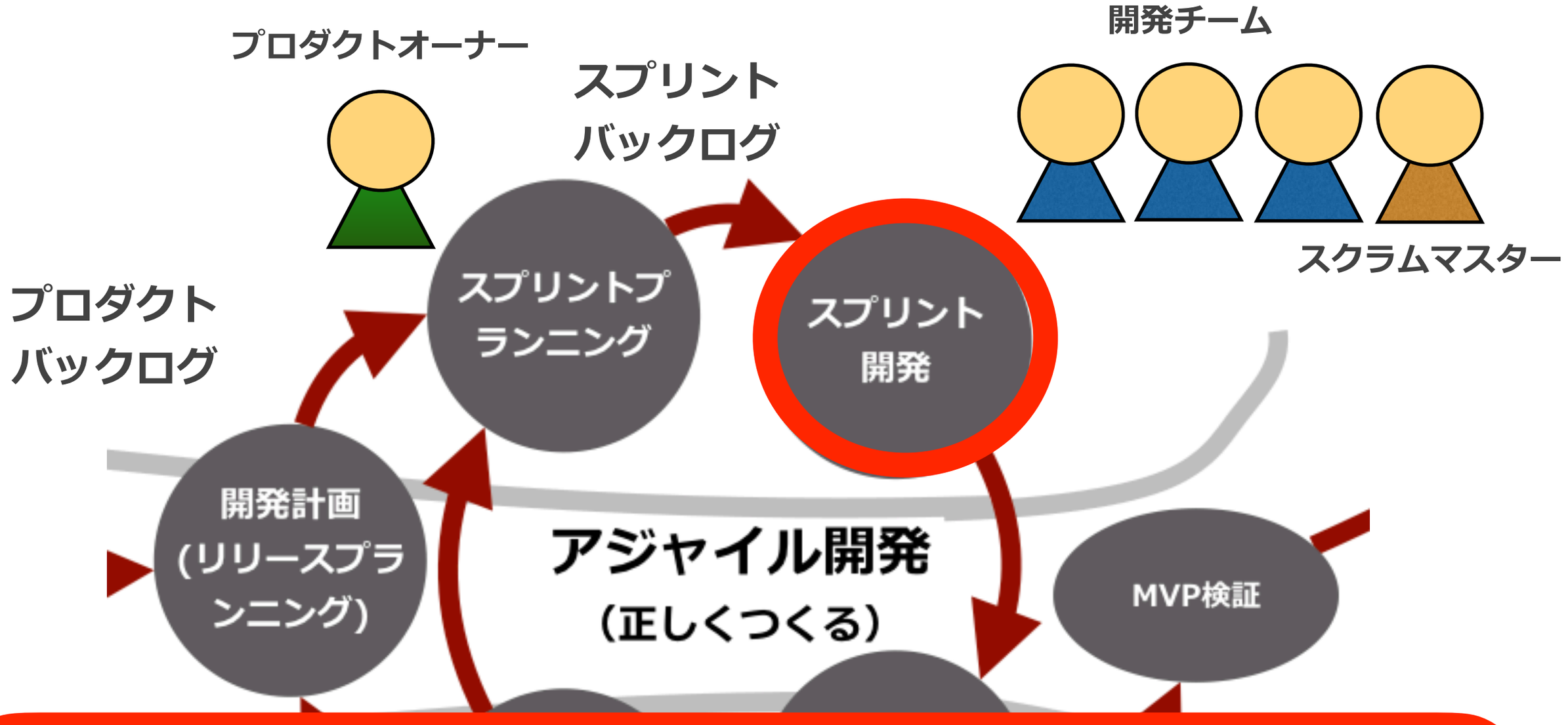
全体の計画

何をどのくらいの期間で作ろうとしているのか？を見立てる。
おおよそ作りたい範囲(ベース)を実現しようとしたら、
何スプリント必要なのか？ これを開発期間中予測し続ける。



反復毎の計画

このスプリント(1-2週間)で何を実現したいかを決める会合。

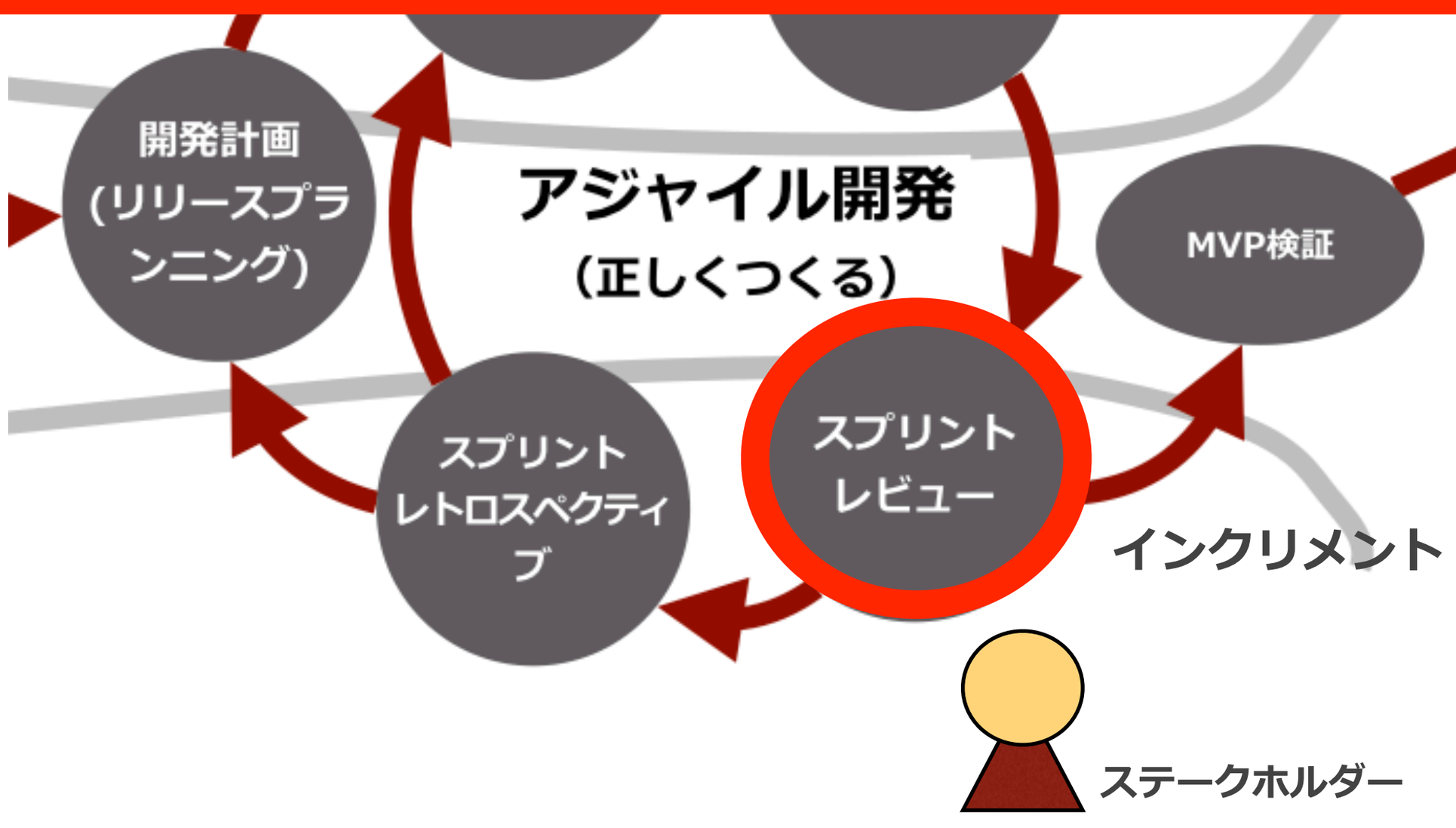


開発

機能のリストに基づき、開発チームが作り進めていく。
状況の同期や問題検出を行なうべく、日々短い会合を開く
(“デイリースクラム”と呼ぶ)

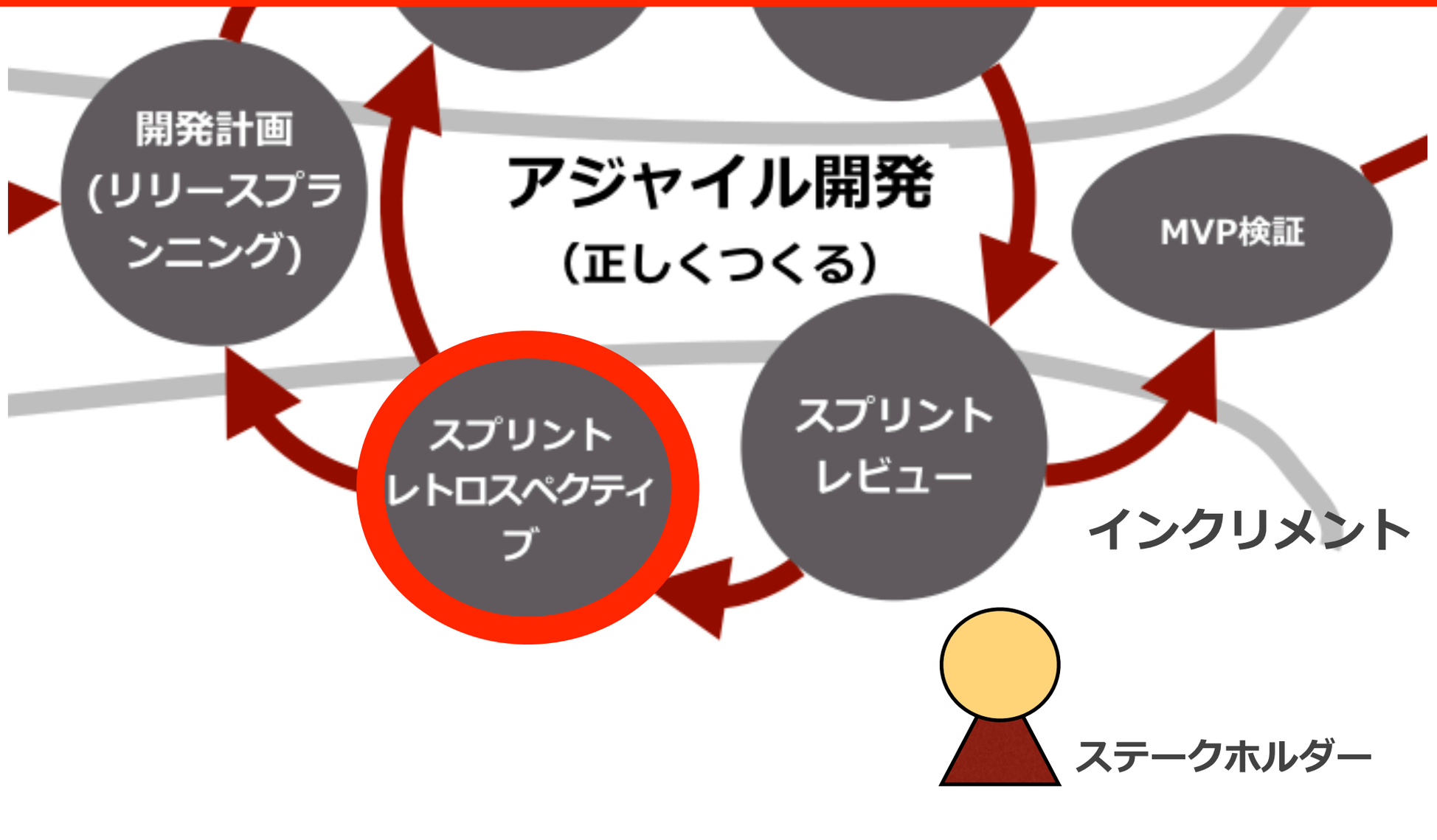
反復毎に確認する

スプリントプランニングで決めたことがどのように実現できているかを確認する会合。開発チームが作ったモノをデモし、プロダクトオーナー及び関係者がそれを確認、フィードバックする。



反復毎にふりかえる

チーム全体でふりかえりを行なう。プロセス、仕事の進め方としてどのような問題があるか、またそのためのカイゼン策を全員で検討する会合。



アジャイル開発運営の2本柱

スプリントプランニング
スプリントレビュー

スプリントプランニング

この2週間をふりかえり
カイゼン点を決める

スプリント
の活動

毎日開発とコミュニケーション
（設計～開発～テスト）

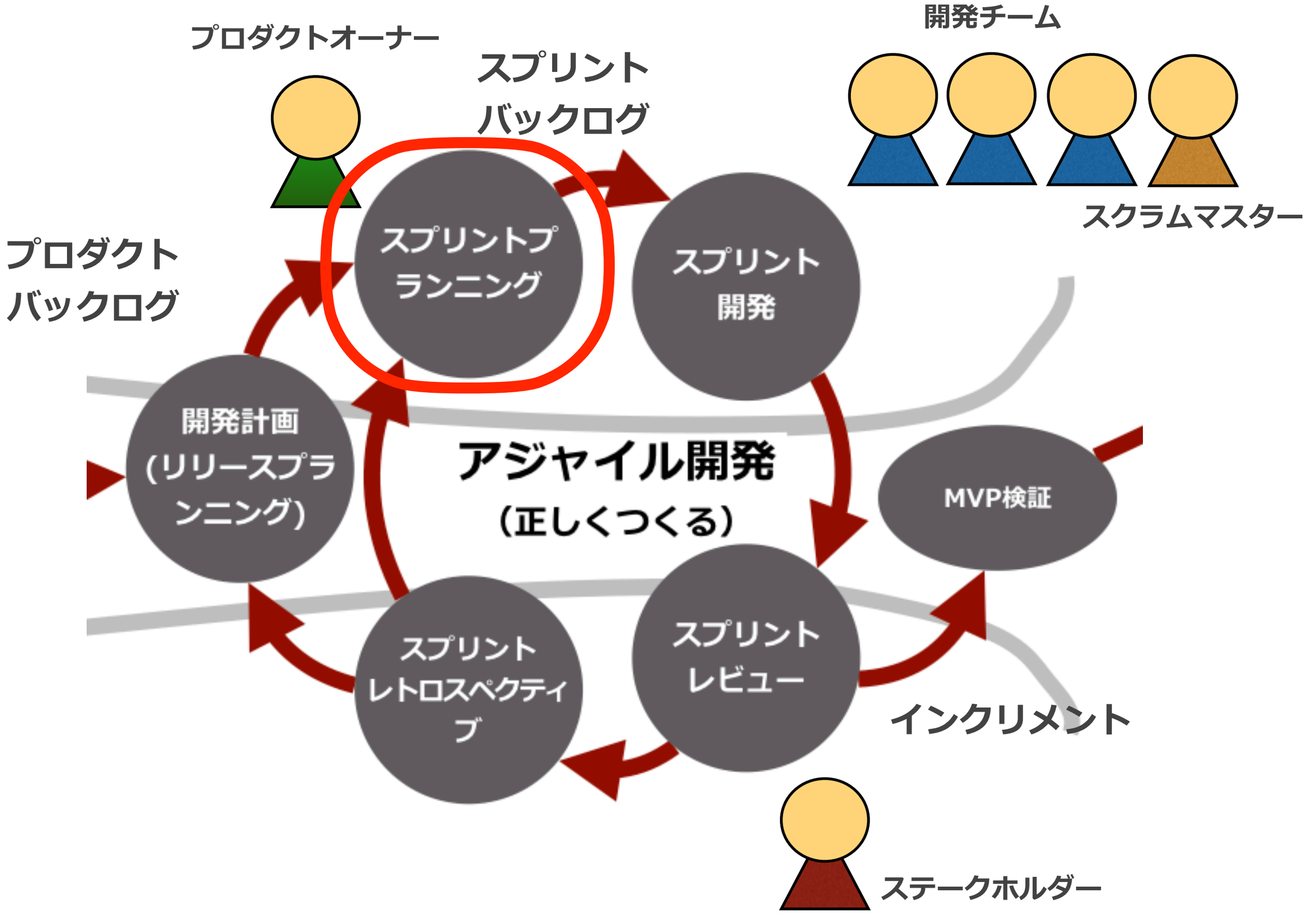
スプリントレビュー

なぜ、重要なのか？

この2つが運営の「レバー」そのもの

スプリントプランニングで何がどこまでできるか予測する
スプリントレビューで結果を見て、次に行うべきことを判断する
つまり、**繰り返せば繰り返すほどプランニングの正確性は高まる**

スプリントプランニングを実施しよう


プロダクトオーナー
スプリント
バックログ
開発チーム
スクラムマスター
プロダクト
バックログ
スプリントプ
ランニング
スプリント
開発
開発計画
(リリースプラ
ンニング)
アジャイル開発
(正しくつくる)
MVP検証
スプリント
レトロスペクティ
ブ
スプリント
レビュー
インクリメント
ステークホルダー

プロダクトバックログ
(機能一覧相当)

スプリントバックログ
(このスプリントでやる分)

スプリントプランニング

何をやる？

どうやってやる？

プロダクト

スプリント
ゴール

このスプリントで実現するべきこと？

スプリント
ゴール

このスプリントで実現するべきこと

プロダクト
オーナー

開発
チーム

スクラム
マスター

ベロシティ

スプリントプランニングとは？(目的)

目的

スプリントで実施するべきことの整理と合意形成を行う

（スプリントの計画づくり）

・スクラムとは、スプリントという時間の箱を繋ぎ続けることで、
必要な意味のあるプロダクトを生み出すゲームのこと

・スプリントプランニングとは、その時間の箱の中で何を実現するかを
決めるもの

・計画上「できたら良いね」で構成してしまうと、プロダクト作りは迷走する
「ここまでならできる」という予想を立て、コミットメントを高められるよう
開発チームおよびPOの両者で、計画作りを行う

・プロダクト作りとは持続していくものなので、ムリにムリを重ねるような
計画作りにしてはならない。

プロダクトバックログ
(機能一覧相当)

プロダクト

スプリント
ゴール

このスプリントで実現するべきこと？

ベロシティ

スプリントプランニング

何をやる？

どうやってやる？

プロダクト
オーナー

開発
チーム

スクラム
マスター

スプリントバックログ
(このスプリントでやる分)

スプリント
ゴール

このスプリントで実現するべきこと

スプリントプランニングとは？(参加者)

参加者

開発チームおよびプロダクトオーナー

スクラムマスター

・プロダクトに必要なものは何かを決められる役割(PO)
・作りたいものをどうやって実現するかその見立ができる役割(開発チーム)
外野の皆さん(関係者)はご遠慮頂く。計画作りに恣意的な力学が働かないように。
専門家の知見が必要であれば、別の機会に集めるようにしておく。
(POが収集して、参考として取り入れるなど)

計画づくりは、立てた内容にコミットメントするメンバーで行う。

スプリントプランニングとは？(タイミング)

タイミング

スプリントの開始初日

・スプリントの計画作りだから、スプリントの最初に行う

・タイムボックスは、

2週間スプリントの場合 → 4時間

1ヶ月スプリントの場合 → 8時間

プロダクトバックログ
(機能一覧相当)

プロダクト

スプリント
ゴール

このスプリントで実現するべきこと？

ベロシティ

スプリントプランニング

何をやる？

どうやってやる？

プロダクト
オーナー

開発
チーム

スクラム
マスター

スプリントバックログ
(このスプリントでやる分)

スプリント
ゴール

このスプリントで実現するべきこと

スプリントプランニングとは？(インプット)

インプット

プロダクトバックログ

チームのベロシティ

これまで作ったプロダクト

スプリントゴールと制約

・プロダクトバックログから当該スプリントで何を作るか範囲を決める
(ゆえに、プロダクトバックログの整理はプランニング前に行っておきたい)
・範囲を決めるのに必要なのがスプリントゴール。POがスプリントで達成したい
ビジネスゴールを言語化して、持ってくる。
・どこまでできるかは、チームのベロシティ(速度)が前提となる

スプリントプランニングとは？(アウトプット)

アウトプット

スプリントバックログ

洗練されたスプリントゴール

・プランニングを通じて、当該スプリントで何をやるべきなのかが明らかになる
・スプリントバックログで、範囲と内容を明確に表現できていること
(スプリントでやりきれるかどうかの見立てを立てるためやることがタスクまで落とし込まれている。もちろん誰がやるのかも)
・プランニングでのやるべきことの選択を通じてスプリントゴールはよりはっきりとした内容になっていること。チームで何を実現すればこのスプリントが白星になるのか、分かるように。

スプリントプランニングとは？(事前準備)

事前準備

バックログリファインメント (バックログの洗練)

・スプリントプランニングの前に行っておくとプランニングがスムーズに進む

・実施内容は、

- プロダクトバックログの順序を入れ替える (形にしたいもの順に並び替え)
- プロダクトバックログへの追加、詳細化

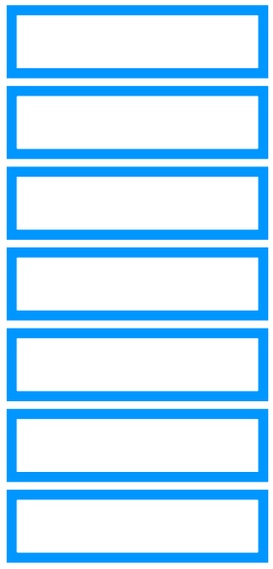
※ 受け入れ条件(どういう条件を満たせばできたと言えるのか) の記載

※ 開発チーム側が内容を見て、作れると判断できるか？（”開発レディ”）

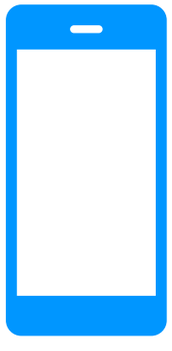
- プロダクトバックログの規模見積もり

※ プランニングポーカーなどで開発チームが実施

プロダクトバックログ
(機能一覧相当)



プロダクト



スプリント
ゴール

このスプリントで実現するべきこと？

ベロシティ

スプリントプランニング

何をやる？

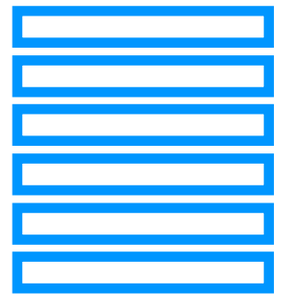
どうやってやる？

プロダクト
オーナー

開発
チーム

スクラム
マスター

スプリントバックログ
(このスプリントでやる分)



スプリント
ゴール

このスプリントで実現するべきこと

スプリントプランニングとは？（内容①）

内容

2部構成／第1部 Whatパート

・第1部では、やるべき範囲を決める。

①プロダクトオーナーがスプリントゴールの達成に必要な
プロダクトバックログの提案を行う

②抽出したリストについて開発規模の見立てを行う

※仕様の確認や、受け入れ条件のすりあわせを適宜行う
(バックログリファインメントで実施済にしておきたい)

③対象範囲がスプリントに収まりきるかを判断する。収まりきらない場合は、
対象を再選別する。

※収まる収まらないはベロシティを用いて判断する。
ベロシティ…1スプリントあたりでチームが開発できる規模

スプリントプランニングとは？（内容②）

内容

2部構成／第2部 Howパート

・第2部では、どうやって実現するかを見立てられるようにする。(How パート)

①選んだバックログの実現にあたり必要なタスクを洗い出す。

②タスクの見積もりを行い、プランニングの調整が必要か判断を行う。

※場合によって、スプリントバックログの抜き差しを行う

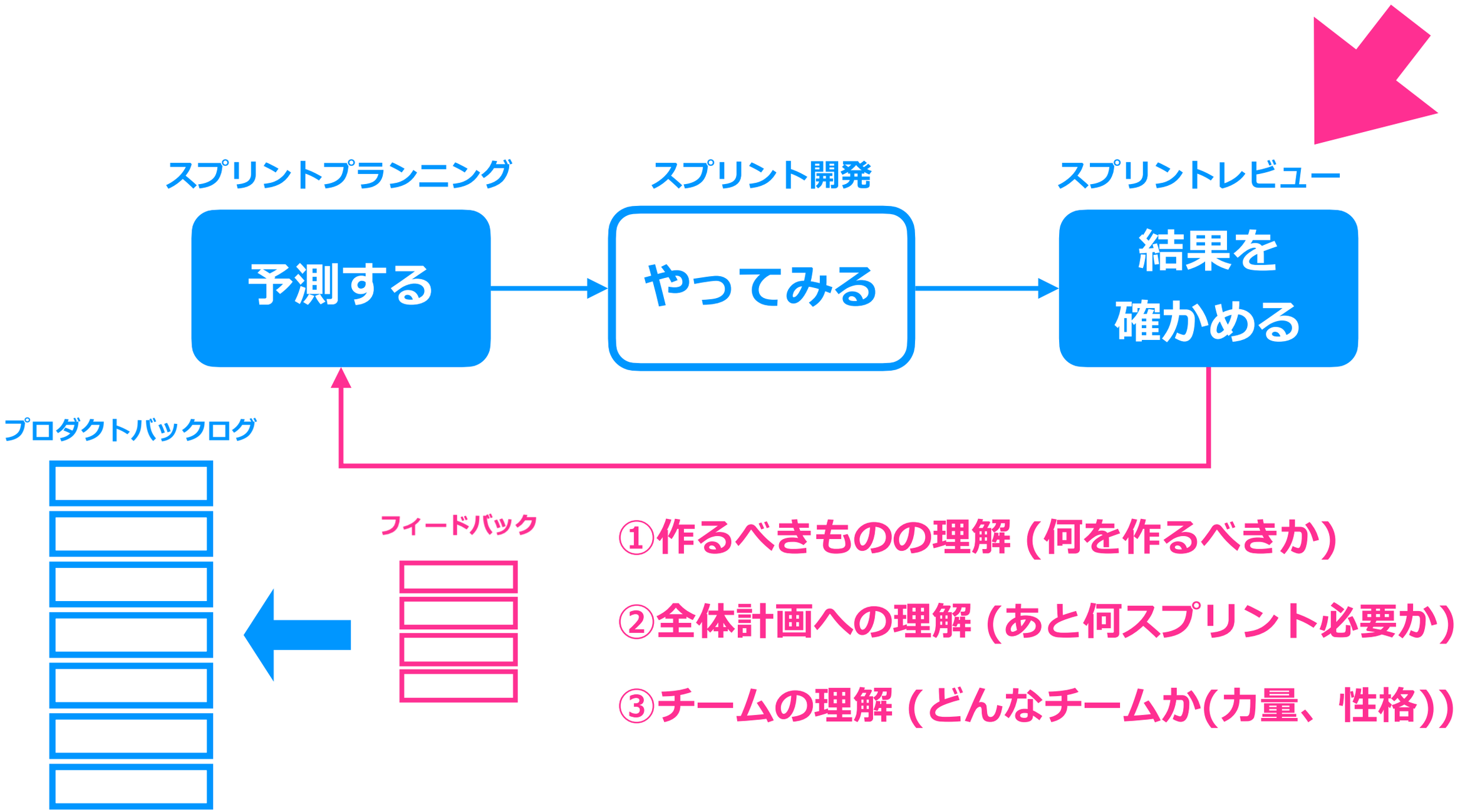
③各スプリントバックログへのサインアップを行う。

※すべてを決めきる必要はない。スプリント中にもサインアップする。

④最後にスプリントゴールの内容、表現を洗練する。

スプリントレビューで結果を見て、次に行うべきことを判断する

つまり、**繰り返せば繰り返すほどプランニングの正確性は高まる**



①作るべきものの理解 (何を作るべきか)

②全体計画への理解 (あと何スプリント必要か)

③チームの理解 (どんなチームか(力量、性格))

スプリントプランニングを
より上手く進める工夫

規模見積もりの方法

プランニングポーカー

全員で見積もりを行うことで、
「**作るものに対する理解を共通にする** (何を作るのか？どう作るのか？)」

- 所要時間ではなく「規模」の見積もりを行う
  ※ 時間の見積もりは難しい。規模は相対見積もりがやりやすい。
- 手順例
  1. 最初に「規模=2」の基準となるプロダクトバックログアイテムを決める
  2. 次のアイテムに対して、何倍の規模感になるかを全員が個々人で考える
  3. 一斉に、自分が考えた見積もりを提示する
  4. 最大の数と最小の数を確認し、その差分理由をチームで捉える
  5. 再度見積もりを行う

キャパシティプランニング

ベロシティを用いたキャパシティ判断

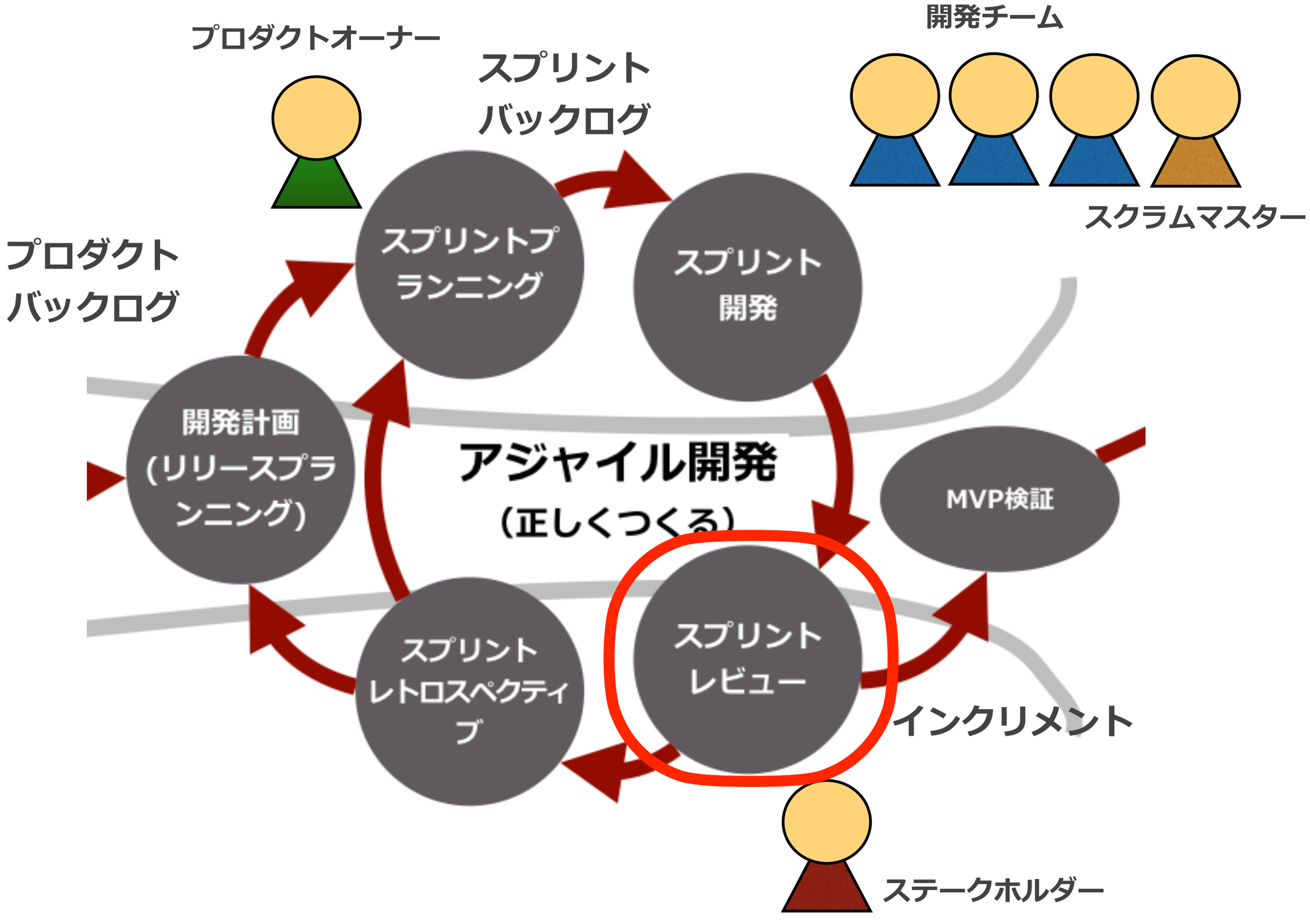
ベロシティとは、「チームが1スプリントで開発できる規模」のこと

・ゆえに、最初のスプリントではベロシティの実績値はない。
最初は仮置をおこなう。数スプリント進んだところで実績値から算出する。
・スプリントバックログ対象として選んだアイテムの規模総数 ÷ ベロシティ
＝ 1以下なら、キャパシティ内。1以上の場合は対象を見直す。
・あるいは、1以上の場合は「努力目標対象(できればここまではやる)」のアイテムを置く。
・ベロシティは3スプリント平均を目安に、定期的に再算出を行う

スプリントゴールの決め方

「われわれはなぜここにいるのか」を分割する

・われわれはなぜここにいるのかにあげた「ミッション」を実現するためにはどのようなステップがあるか検討する。
「ステップを全てクリアしたらミッションに到達できる」
・「1つのステップ = 1スプリントゴール」のイメージ
・つまり、スプリントゴールとは、実現することでミッション達成に近づけるようなもの。
・スプリントゴールが描けないということは、ミッション達成の道筋が見えていないということ。


プロダクトオーナー
スプリント
バックログ
開発チーム
スクラムマスター
プロダクト
バックログ
スプリントプ
ランニング
スプリント
開発
開発計画
(リリースプラ
ンニング)
アジャイル開発
(正しくつくる)
MVP検証
スプリント
レトロスペクティ
ブ
スプリント
レビュー
インクリメント
ステークホルダー