

# ソフトウェア工学特論

## (5) アジャイル型プロセスモデル（Ⅱ）

# アジャイル開発

## (4)アジャイル型プロセスモデル（Ⅰ）

1. アジャイルソフトウェア開発宣言
2. アジャイル開発とは
3. XP (eXtream Programming)

## (5)アジャイル型プロセスモデル（Ⅱ）

4. スクラム
5. 大規模スクラム
6. ディシプリンド・アジャイル・デリバリー

時間があれば

# スクラム

前方にあるカオスをスクラムを組んで突破する

# スクラム

- J. Sutherlandにより考案され、Ken Schwaber と Mike Beedle によって拡張されたアジャイル開発手法
- スクラムでは、一定期間のサイクルで、動くソフトウェアを作りながら開発を進める

- 自己組織・自己管理的チーム

- 協働、連携、助け合い、改善することでチームの成果を最大化

そのための

- コミュニケーションの最大化
  - オーバヘッドの最小化
  - 暗黙知の共有の最大化

原則11: 最良のアーキテクチャ・要求・設計は、自己組織的なチームから生み出されます。

原則3: 動くソフトウェアを、2~3週間から2~3ヶ月というできるだけ短い時間間隔でリリースします。

# 参考資料

- スクラムガイド <https://scrumguides.org>
  - Ken Schwaber, Jeff Sutherland 著
  - 無料、日本語版PDFもあり
- スクラム 仕事が4倍速くなる“世界標準”のチーム戦術  
<https://www.hayakawa-online.co.jp/shopdetail/000000012733/>
  - Jeff Sutherland 著
  - スクラムが生まれた背景、思想（マインドセット）の詳細
- エッセンシャル スクラム  
<https://www.shoeisha.co.jp/book/detail/9784798130507>
  - Kenneth S. Rubin 著
  - 実践ガイド

# スクラムチーム

- 開発者
  - インクリメントのあらゆる面の開発を行う
  - スプリントの計画作成、品質、計画の適応について責任を持つ
  - 全員がプロとしての責任を負う
- プロダクトオーナー
  - プロダクトの価値を最大化することの責任を持ち、プロダクトバックログを管理する
- スクラムマスタ
  - スクラムの確立に責任を持つ
  - インクリメントの作成への集中、イベントの実施、障害物の排除を支援する

- プロダクトオーナー 1名、スクラムマスタ 1名、開発者数名から構成されるプロダクトゴールに焦点を合わせたプロの集団
- 各スプリントで十分なインクリメントを、アジャイルに開発できる人数（10名以下）
- メンバはスプリントで価値を生み出すために必要な全てのスキルを持っている
- チーム内に階層はない

理想的にはこうあるべき  
現実的には難しいかも

# スクラムのサイクル

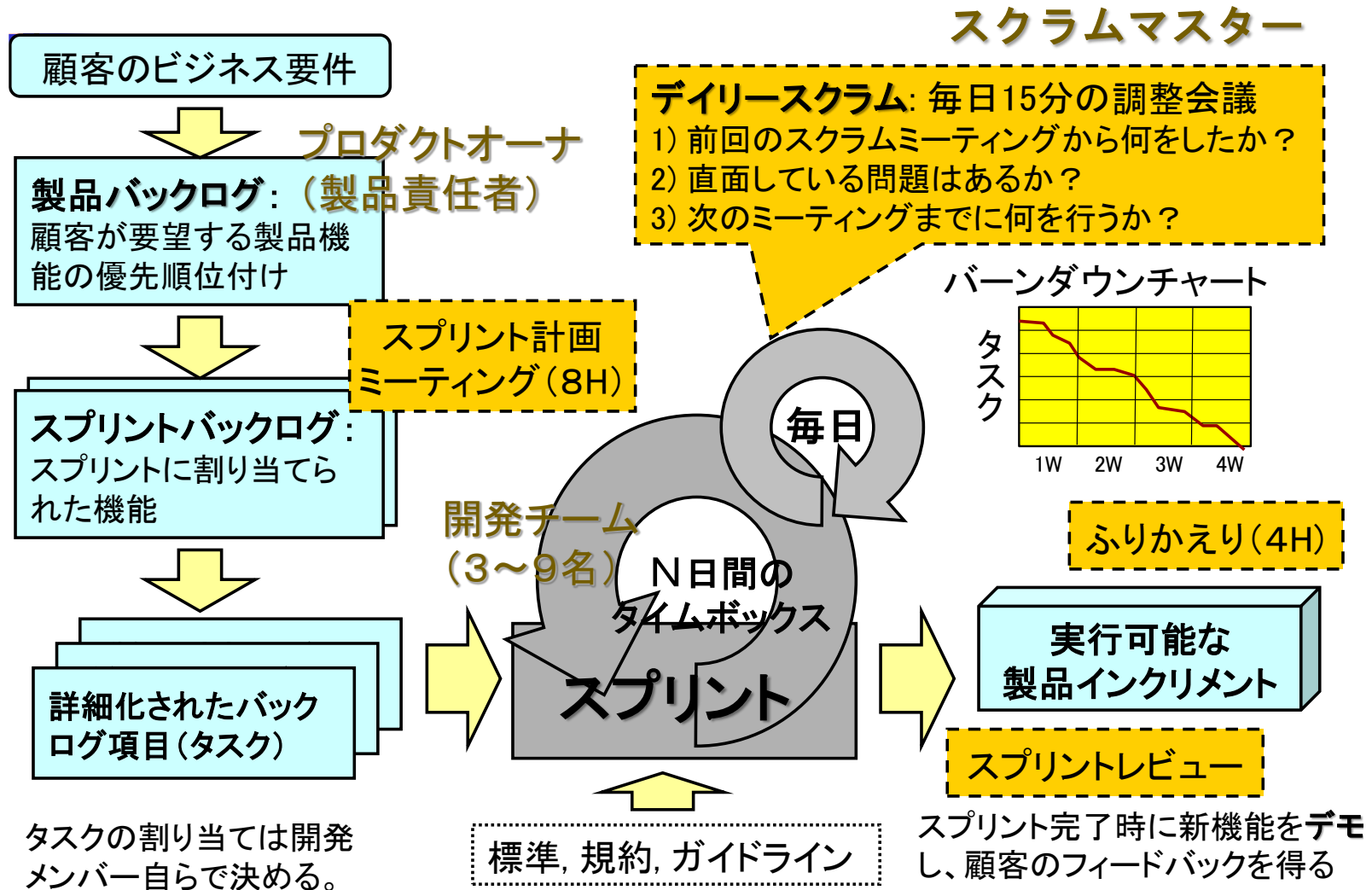


1. プロダクトオーナーが、必要な機能をプロダクトバックログに並べる
2. 開発チームは、スプリントで機能を開発し、プロダクトの価値としてのインクリメントに変える
3. ステークホルダと開発チームは、インクリメントを検査し、次のスプリントに向けて調整・適応する

## プロセスのパターン化

- バックログ：優先順位付きの機能一覧、課題管理表
- スプリント：開発を行う反復の単位
- インクリメント：利用・評価可能なスプリントの成果物

# スクラムのプロセスの流れ





# スクラムの成果物

- プロダクト（製品）バックログ
  - プロダクトの価値を向上させる機能（アイテム）の一覧
    - 優先度順に並べられており、スプリントプランニング時には選択の準備済み
  - アイテムはプロダクトゴール（プロダクトの将来の状態）を含む
    - 残りのアイテムで、プロダクトゴールを達成するための何か（What）を定義
- スプリントバックログ
  - スプリントゴール（Why）、プロダクトバックログアイテム（What）、インクリメントのデリバー計画（How）から構成
  - スプリントゴールは、スプリングプランニングで作成
  - アイテムは、進捗を管理できるような作業アイテム（タスク）に分解
- インクリメント
  - プロダクトゴールに向けたステップ

# スクラムイベント

- スプリント
  - 1ヶ月以内の決まった期間の開発を行う反復の単位
    - スプリントプランニング・レビュー・レトロスペクティブを含む
    - スプリントごとにプロダクトゴールに対する進捗の検査・適応が可能
  - スプリントゴールの達成を危険にさらさない（変更、品質）
- スプリントプランニング（計画ミーティング）
  - スプリントで実行する作業の計画
  - プロダクトの価値の向上をスプリントゴールとして定義
  - プロダクトバックログからスプリントで実行するアイテムを選択
    - アイテムを小さな作業アイテム（タスク）に分解
    - タスクの作業量見積もり
    - プロダクトバックログを必要に応じてリファインメント

# プロダクトバックログリファインメント (PBR)

- プロダクトバックログの整理
  - 作成、修正、見積もり、優先順位付け
- プロダクトバックログは要求、状況に合わせて変化していくため、継続的に整理
  - 着手が近づいてきたアイテムは、詳細化した複数のアイテムに分割
  - 不要になったアイテムの削除
- リファインメント
  - グルーミングと呼ばれる場合もある（エッセンシャルスクラム）

# スクラムイベント (2)

- デイリースクラム（毎日同じ時間に15分）
  - スプリントゴールに対する進捗に焦点を当て、必要ならば作業を調整
  - コミュニケーションをはかり、障害物を特定、迅速な意思決定を行う
- スプリントレビュー
  - スプリントの成果とプロダクトゴールに対する進捗を確認
  - 今後の適応、次にやるべきことを決定
    - 必要に応じて、プロダクトバックログを調整
- スプリントレトロスペクティブ（ふりかえり）
  - 開発の品質と効率を高める方法を計画することが目的
  - スプリントにおける個々の作業、協働、プロセス、ツール、ゴールをふりかえり、課題を見つけ、その原因を探究、改善のための変更方法を特定

# スクラムの三本柱

- 透明性

- イベントや成果物は全て、スクラムチームおよび顧客に見えている（透明性）必要がある
- 見えていることで検査ができ、検査することで、適応が可能になる
- 透明性が無ければ、検査は誤解を招き、結果ムダになる

- 検査

- プロダクトゴールに対する進捗の検査、また成果物の検査は、スクラムイベントでしっかりと行い、課題を見つける必要がある
- 課題を見つけることで、適応が可能になる

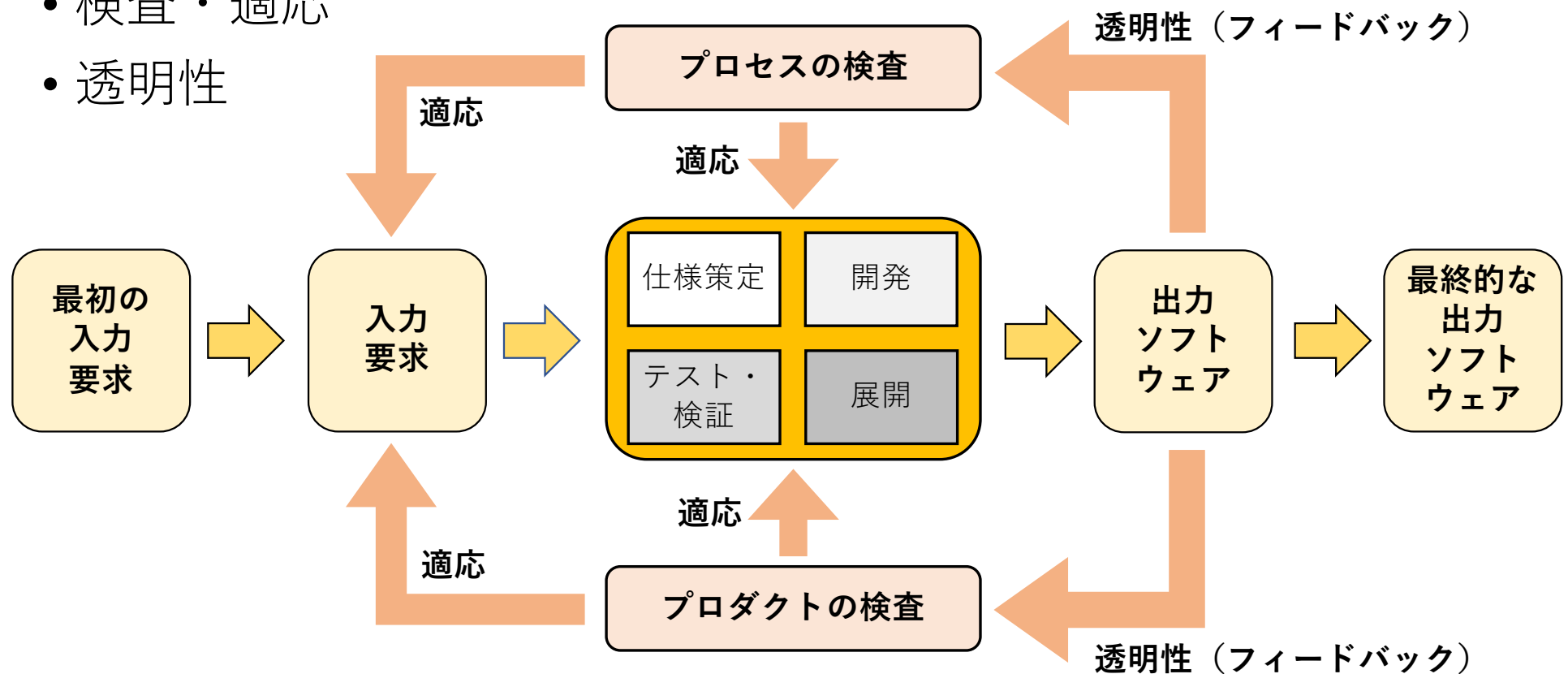
- 適応（調整、軌道修正）

- 課題や許容範囲からの逸脱は、最小限に抑えるため、できるだけ早く調整、修正する
- スクラムチームは、調整、修正の権限を持っていること（自己管理）が重要

# プロセスとプロダクトの改善

## スクラムの3本柱

- 検査・適応
- 透明性



# LeSS: 大規模スクラム

Large-Scale Scrum

# スクラムの大規模化

- スクラム
  - 基本的には1チームだけの状況を対象
  - 複数チームが協力して働く状況には重点を置いていない
- LeSS: Large-Scale Scrum
  - 1つのプロダクトを複数チームで協働して開発するために考えられたスクラムの1つ
  - スクラムを大規模開発に適応
    - 新しいスクラムやスクラムの改善版ではない
  - 2つのLeSS
    - LeSS: 2~8チーム（8はぎりぎりの上限）
    - LeSS Huge: 8チーム以上



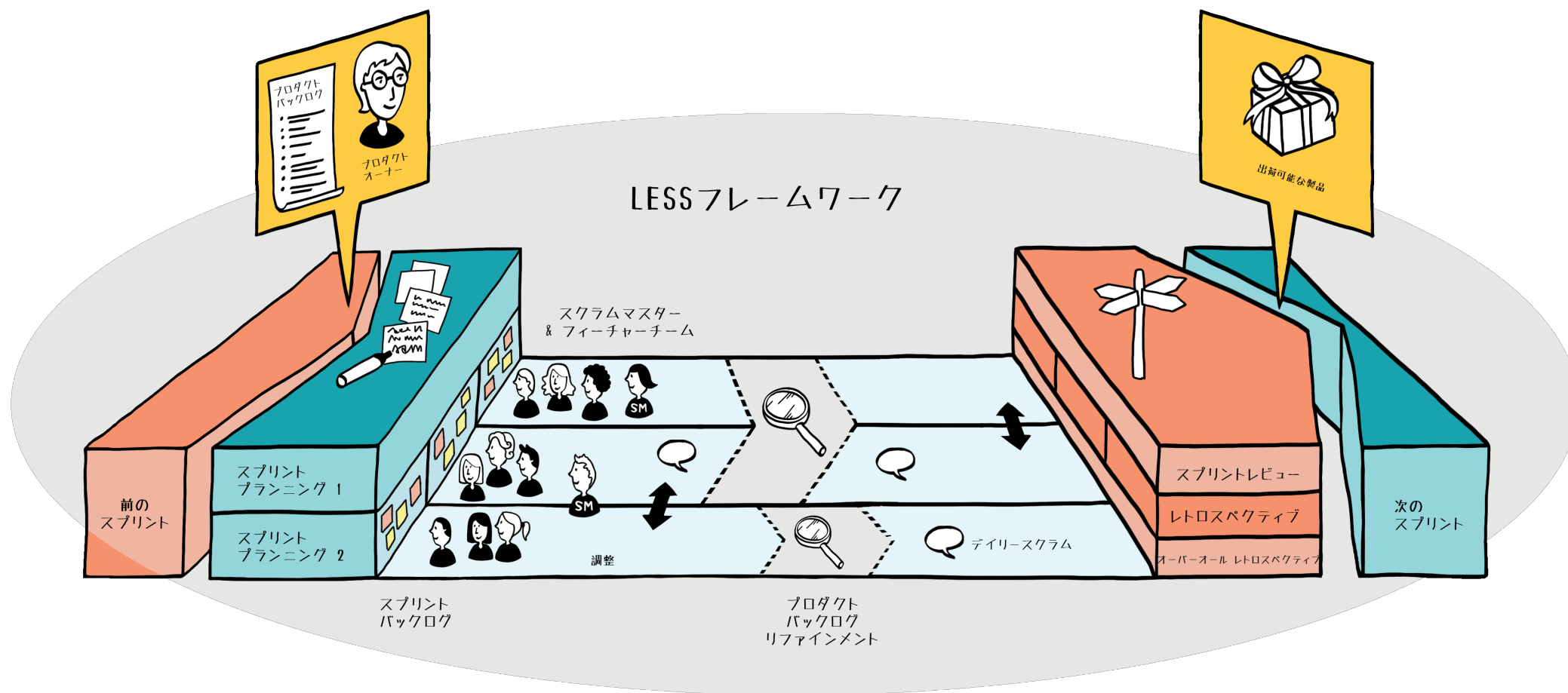
# LeSS, LeSS Huge の共通部分

- プロダクトオーナー 1人
- プロダクトバックログ 1つ
- 全チーム共通のスプリント
- リリース可能なプロダクトのインクリメント 1つ

# LeSS

• プロダクトオーナー	1人	共通部分
• プロダクトバックログ	1つ	
• 全チーム共通のスプリント		
• リリース可能なプロダクトのインクリメント	1つ	
• スクラムマスター	1~3チームに1人	
• チーム	2~8	
• スプリントバックログ	チームに1つ	

# LeSSによる開発の進め方：概要



# LeSSによる開発の進め方

1. スプリントプランニング 1

2. スプリントプランニング 2

3. 開発

4. スプリントレビュー（全体で実施）

5. チームレトロスペクティブ

6. オーバーオールレトロスペクティブ

1チームの  
スクラムと  
同じ部分

# LESS スプリントプランニング

スプリント  
プランニング 1



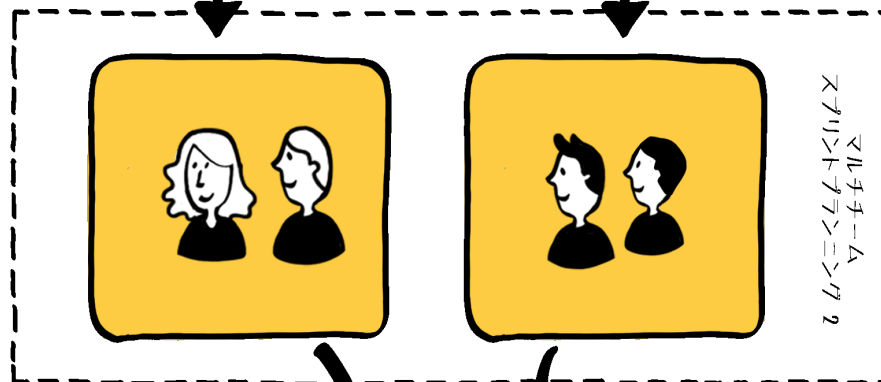
選択されたアイテム

選択されたアイテム

スプリント  
プランニング 2



スプリントバックログ



初期の設計  
と計画

# スプリントプランニング 1, 2

## 1. スプリントプランニング 1 【チーム合同】

プロダクトオーナーと全チームの代表者で、プロダクトバックログから各チームがスプリントで作業するアイテムを選択

- プロジェクト初期、全チームで共有すべき情報が多い段階では、代表者ではなくメンバーが参加することが有効
- 優先度の高いアイテムはチームに分散して割り当てると、何か問題が起こっても優先度の高いアイテムが終了しない事態は避けられる可能性が高い
- 関連するアイテムについてチーム間で議論し、疑問を解消

## 2. スプリントプランニング 2 【チームごと／複数チーム】

- チームごとのスプリントプランニングを並行に実施
- 関連するアイテムを担当する複数のチームは合同で実施した後に、個別に実施

# 開発

- 繰り返し
  - コーディング
  - 統合
  - テスト
- スプリントの終盤で、スプリントレビューのために、あわてて統合したり、まとめてテストの実施をするようなことはしない
  - そうならないようにするための仕組みづくりが重要
    - ← プランニング、調整活動（後述）

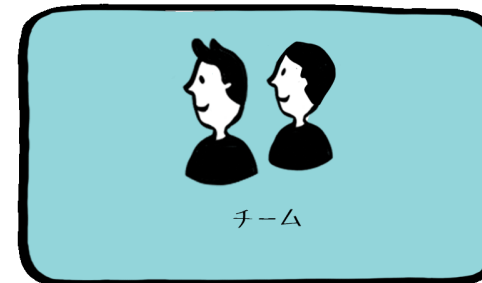
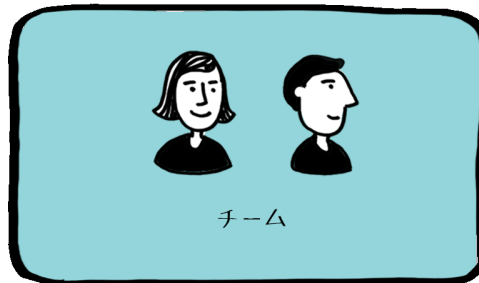
## LESS スプリントレビュー & レトロスペクティブ

スプリント  
レビュー



全体で実施

チーム  
レトロスペクティブ



チームごとに実施

オーバーオール  
レトロスペクティブ



全体で実施  
・ チームからは  
代表者



# オーバーオールレトロスペクティブ

- チームレトロスペクティブの後に、チーム間・全体の課題を話し合い、改善について議論
  - 参加者：プロダクトオーナー、スクラムマスター、各チーム代表者
  - 各チームの局所的な課題を持ち寄るのではなく、チーム間・全体の課題が議題
- 原則：継続的改善
  - 現状の課題についての議論
  - これまでの改善案の考察
  - 改善案の策定

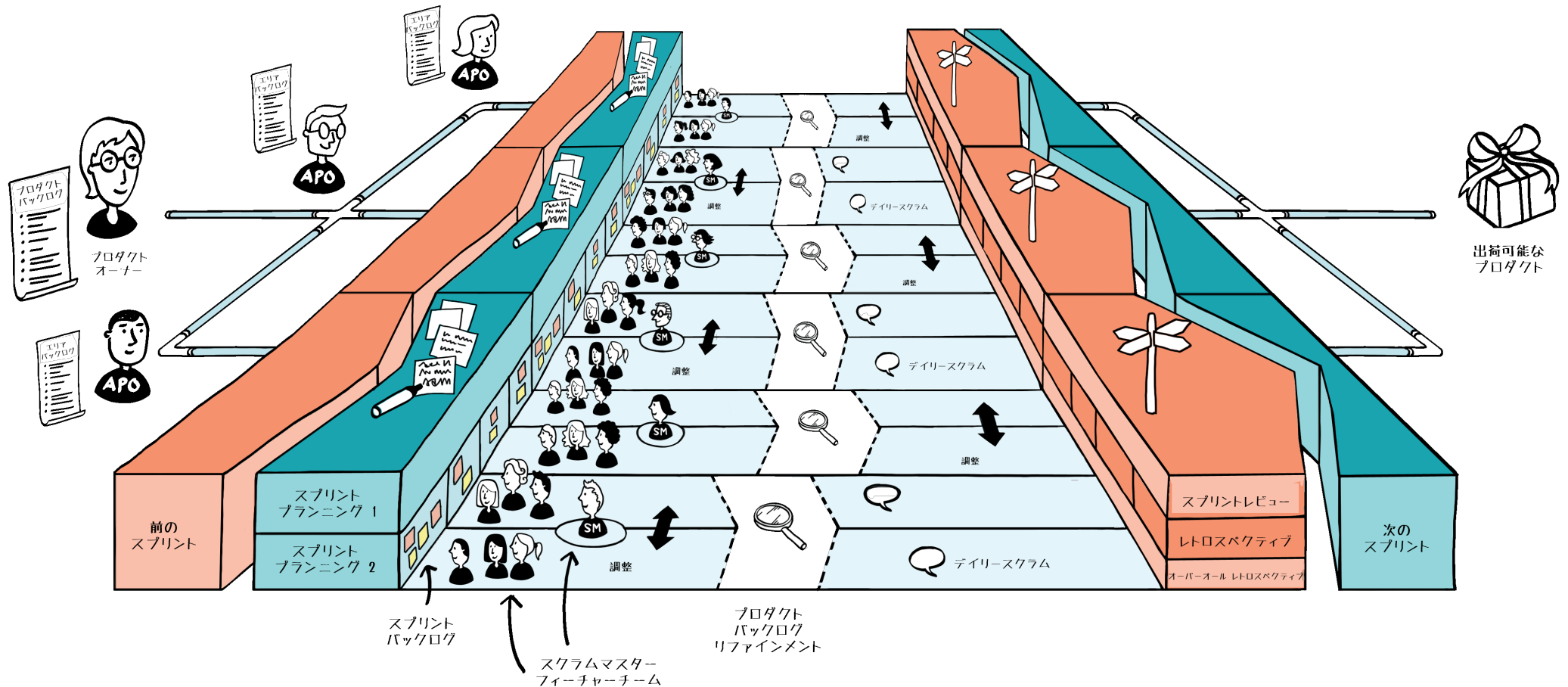
# LeSSによる開発の調整活動

- 複数チームの設計ワークショップ
  - 関連するアイテムを担当する複数チームでのスプリントプランニング2では時間が不足する場合、複数のチームメンバ全員で実施
  - アイテムへの理解の共有、一貫性のある設計
- 開発中の調整・継続的デリバリー
  - 全チーム・メンバは継続的にコードを統合
    - 依存性はできるだけ素早く解消
  - 関連するコードの変更（追加・修正・削除）について調整
    - 気づいたら素早く（全体ミーティングを待たないで）当事者間で話し合い、共同作業によりムダを排除

# LeSS Huge

- |                        |    |      |
|------------------------|----|------|
| • プロダクトオーナー            | 1人 | 共通部分 |
| • プロダクトバックログ           | 1つ |      |
| • 全チーム共通のスプリント         |    |      |
| • リリース可能なプロダクトのインクリメント | 1つ |      |
- 要求エリア 2以上
  - 要求エリアごとに
    - エリアプロダクトオーナー 1人
    - エリアプロダクトバックログ 1つ
    - チーム 4~8
    - スプリントバックログ チームに1つ
  - スクラムマスター 1~3チームに1人

# LeSS Huge による開発の進め方：概要



# LeSS Huge: 要求エリア

- 顧客の関心事でアイテムを分割
  - 4~8チームでの開発となるの大きさ…1つの要求エリアは十分に大きい
  - アイテムは1つのエリアプロダクトバックログにのみ割当
- 各要求エリアは動的に変化
  - 重要性の変化に伴う縮小・拡大
  - 変化に沿ってチームはエリアを移動
- エリアプロダクトオーナー
  - 各要求エリアに1人

# DAD: ディシプリンド・ アジャイル・デリバリー

エンタープライズアジャイルと呼ばれるものの1つ

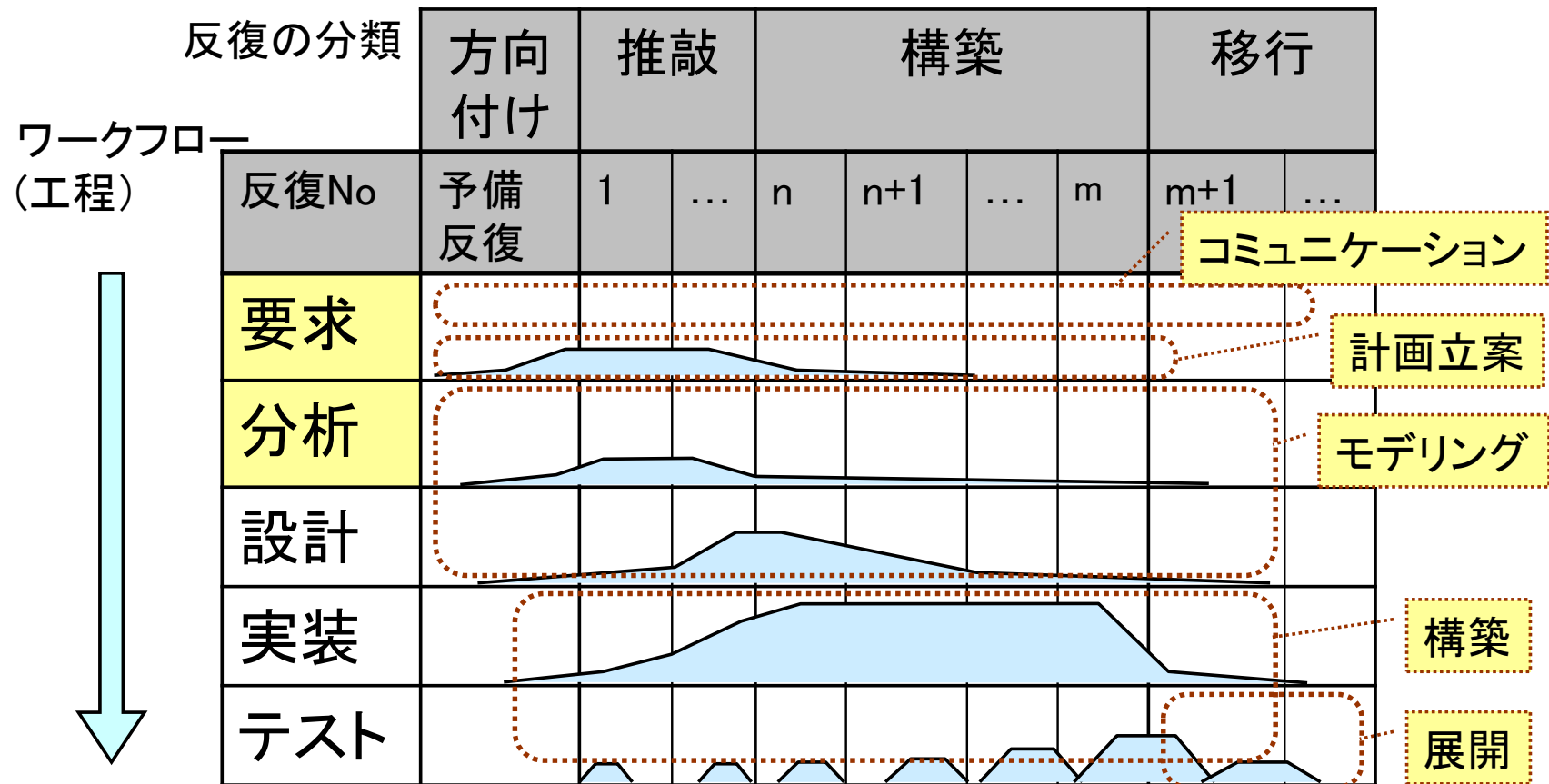
# アジャイル開発プロセスの明確化

- ユニファイドプロセス等の規範的なプロセスモデル
  - 工程を定義：要求分析、開発（設計・実装）、テスト・検証、展開
- アジャイル開発
  - 開発に集中している印象を持ちやすい
    - 要求分析・仕様策定が疎かになる場合がある
  - プロセスモデルとしては工程は明確ではない
    - 要求分析・仕様策定の組み込み方は任されている



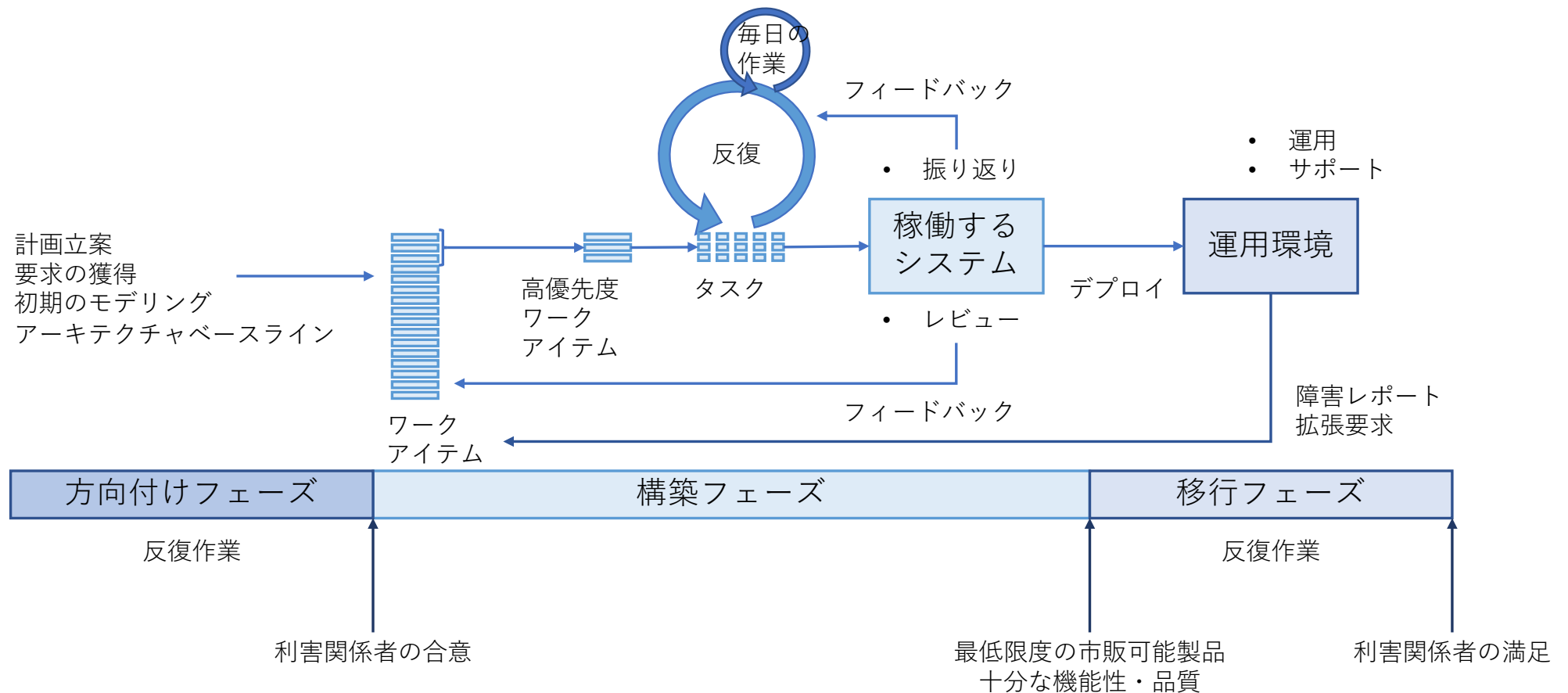
- 全体工程が定義されたアジャイル開発プロセスモデル
  - DAD: ディシプリンド・アジャイル・デリバリー

# UPのフェーズとワークフロー

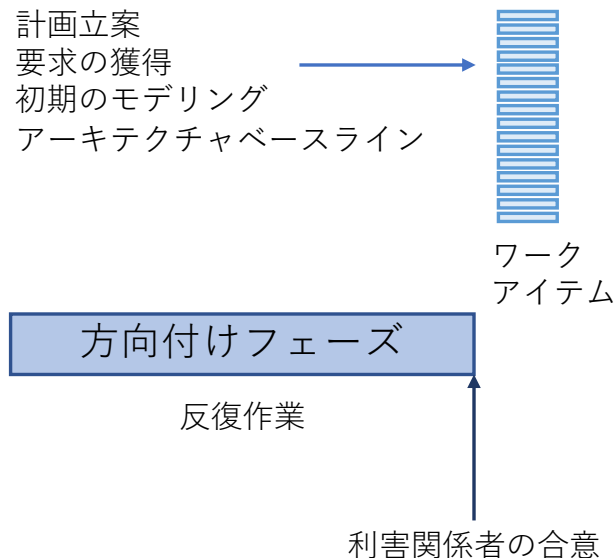




# DADのライフサイクル



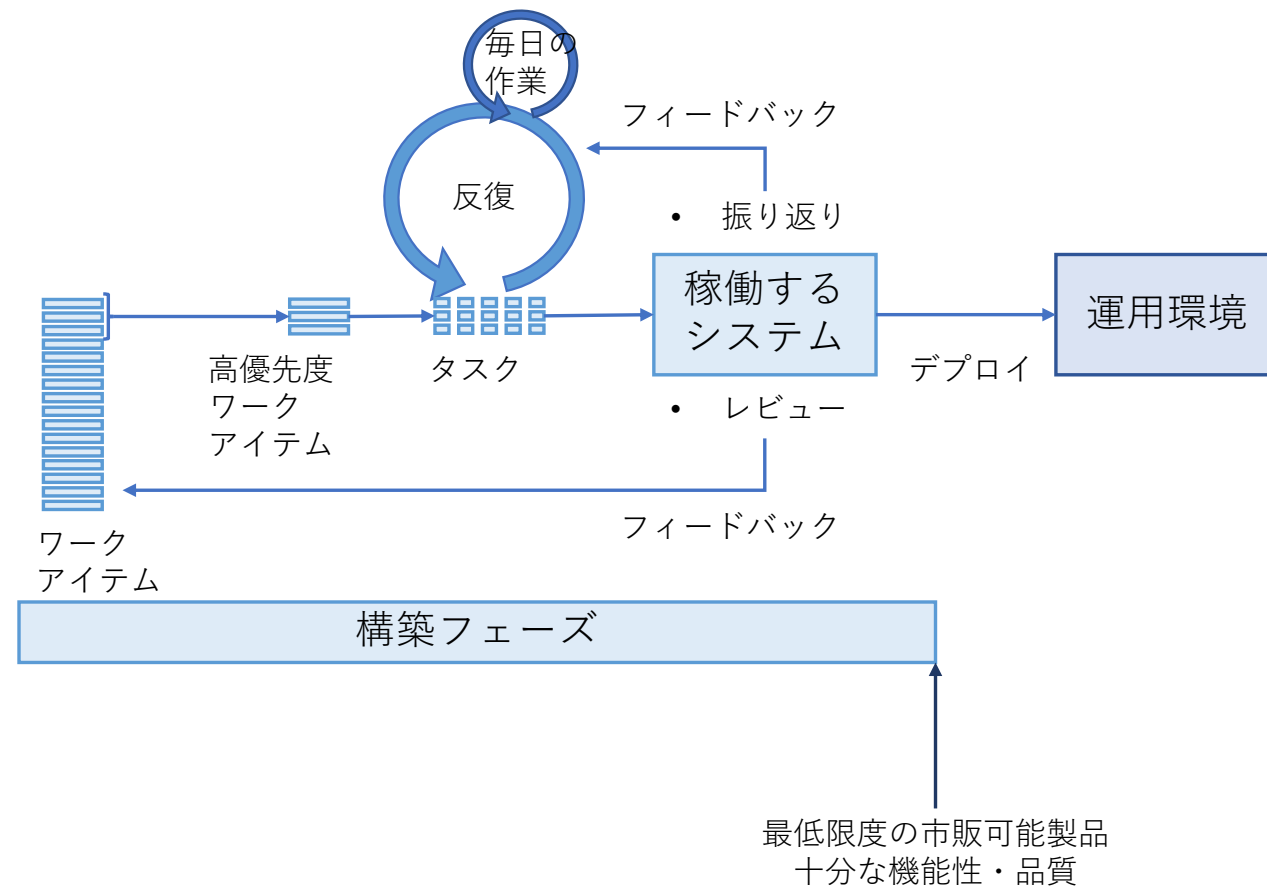
# 方向付けフェーズ



- 計画立案
  - プロジェクトのゴールの確認・計画
  - 実現可能性の検討
  - 進む方向を正しく設定
- 要求の獲得（構想）
  - 対象のスコープを理解
  - 初期の要求モデリング
  - ワークアイテムの作成
- アーキテクチャベースライン（構想）
  - 機能を肉付けする骨格
- フェーズのマイルストーン
  - 利害関係者の合意を得た時点で終了

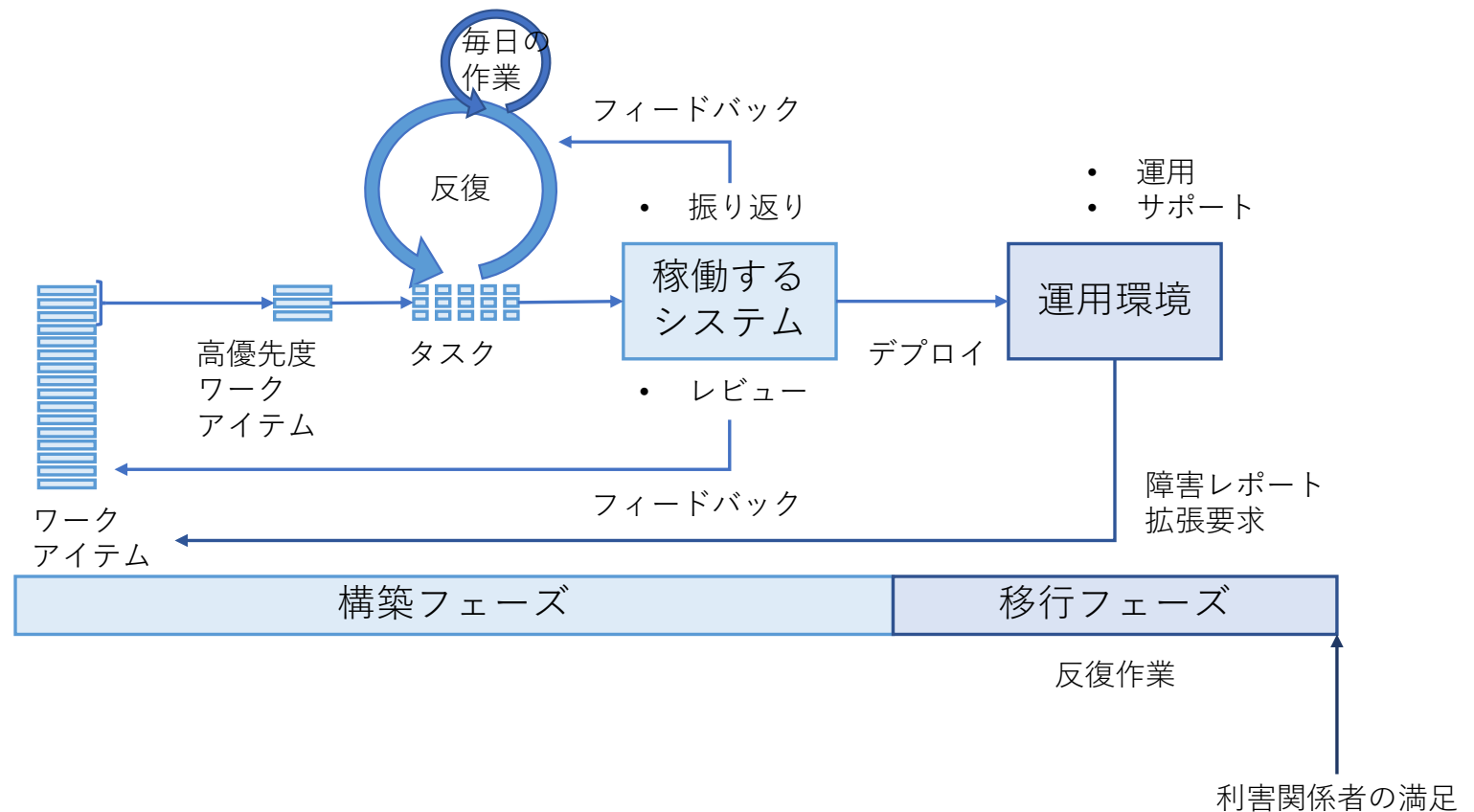
# 構築フェーズ

- アーキテクチャ
  - 方向付けフェーズでの構想を、初期の反復で実装
  - 機能要求、非機能要求の両方を満たせることを確認
- フェーズのマイルストーン
  - デプロイの準備ができ、利害関係者の合意を得た時点で終了



# 移行フェーズ

- 運用準備が整っていることの確認
  - デプロイのテスト
  - データ移行の準備
  - マニュアルの確認
- 受入準備が整っていることの確認
  - ユーザのトレーニング
  - サポート環境の準備
- 運用環境へのデプロイ



# まとめ

スクラムまたその大規模化

# まとめ

- スクラムは、プロジェクト管理のフレームワークを提供するアジャイル開発手法である。
  - 漸進的开发をスプリントという固定期間の繰り返しで実施する。計画は作業のバックログに優先度付けを行い、各スプリントで最も優先度の高いタスクを実施する。
- 大規模スクラム LeSS は、1つのプロダクトを複数チームで協働して開発するために考えられたスクラムの1つである。
  - スクラムを大規模開発に適応したものであり、新しいスクラムやスクラムの改善版ではない。
- ディシプリンド・アジャイル・デリバリー DAD は、エンタープライズアジャイルと呼ばれるものの1つである。
  - 全体工程が定義されたアジャイル開発プロセスモデルである。