

金融アプリ開発における 大規模アジャイル変革

5年間の実践から見た「人・プロセス・契約」による持続可能な品質担保

アジャイル / SAgile® / 金融アプリ開発

発表者：岡本 あかり 木全 将介 | 日本総合研究所

本日の流れ



変革の舞台：SMBCアプリを中心とした巨大な顧客接点

当部のミッション

SMBCスマホアプリ・SMCCスマホアプリ
・SMBCIDの開発
エンドユーザ接点となるスマホアプリを担当
Olive関連の大型案件を継続推進

SMBCアプリ開発の現場

SMBCアプリを所管
SMBC/JRI/開発ベンダー一体運営でア
ジャイル開発
スマホアプリサーバはAWS上で中継・
制御

SMBCアプリ開発の特性

価値提供

残高・振込・カード・証券・外部連携まで“生活接点”
化

品質要求

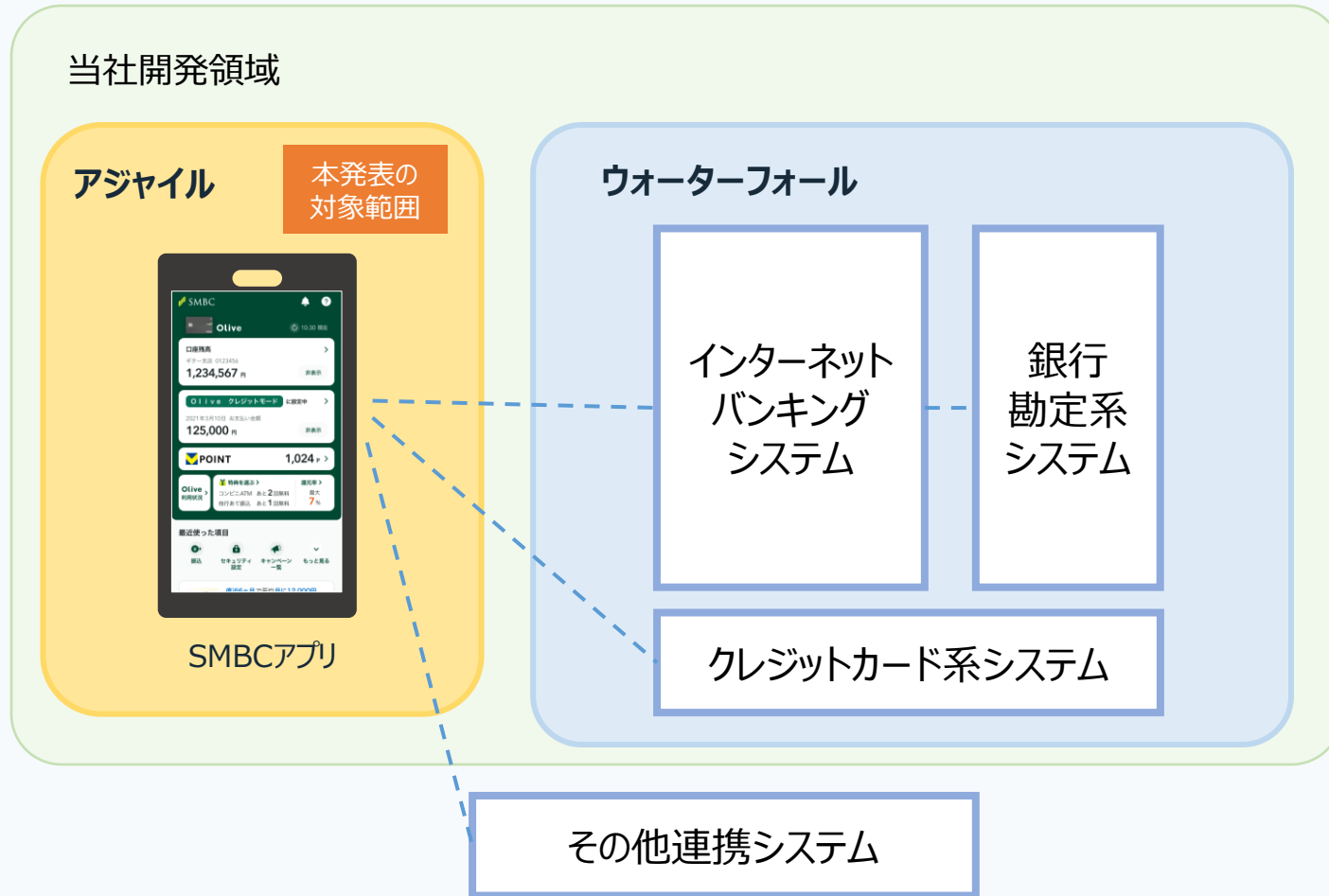
金融機関として安定稼働・セキュリティ・監査性が前
提

変化速度

Olive、PayPay、MoneyForward等の施策が継
続拡大

環境：周辺システムをはじめとした SMBCアプリを取り巻く環境

開発プロセスの違い



金融システム特有の制約

品質

社会的影響の大きさ

規模

複数システム連携

契約

スコープ固定の商習慣

人

ユーザ部門・ベンダー・自社要員の流動性

5年間の変革ストーリー：スクラムからSAFe®、そして制度設計へ

変革の焦点が「手法」から「組織構造・商習慣」へ移動



※ SAFe® ... Scaled Agile Framework
大規模な組織や企業全体でアジャイルを進めるためのフレームワーク

SAFe®導入後に見えた、3つの「現実的課題」

1

マインドセットの浸透と相互理解の欠如

ベンダーやユーザー部門に加え、自社要員も含めたメンバーの入れ替わりが多く、全員が同じ解像度でプロダクトに向き合っていない

2

複数チーム同期の重要性

大型案件の立ち上がりに伴い複数チーム横断の開発が増え、チーム間のより密な同期が必要になってきた

3

契約形態による機動力阻害

請負型のスコープ固定が、優先順位の変更やプロダクトの価値最大化と衝突

解決方針：人・プロセス・契約を同時に設計する

人 ナレッジ循環

オンボーディング、
アジャイル教育の整備、
ワンチーム化

プロセス 同期と透明性

PIプランニング、
依存関係の可視化、
システムデモ

契約 価値へのコミット

要員確保の方針転換、
柔軟な優先順位

持続可能な品質担保モデル = 組織的レジリエンス

打ち手① 人：入れ替わりを前提に「ナレッジ循環」を仕組み化



オンボーディング・パッケージ

ドメイン知識とアジャイル作法を短期間で教育

混成チーム

ベンダー・ユーザー・プロパーを混在させ、品質を共同所有

ペア/モブプログラミング、コードレビュー会

リアルタイムでレビュー指摘を行い、指摘内容や開発のTips等をチームで共有

狙い：未経験者を含む体制でも品質観をそろえ、早期に自走化

具体事例① ステークホルダーを集めてLego4Scrumの開催

取り組み

プロジェクト参画者（当社、ユーザー部門、ベンダー）
総勢約60名でアジャイル体験ワークショップ（Lego4Scrum）を実施
（2024・2025年度実施、継続予定）

Lego4Scrumとは？

レゴブロックを組み立てながら、アジャイル開発の基本的な手法である「スクラム」の仕組みやマインドセットを体験する

目的

- ・ アジャイルな価値観を身に付ける
- ・ ユーザーやベンダーとの相互理解を深める

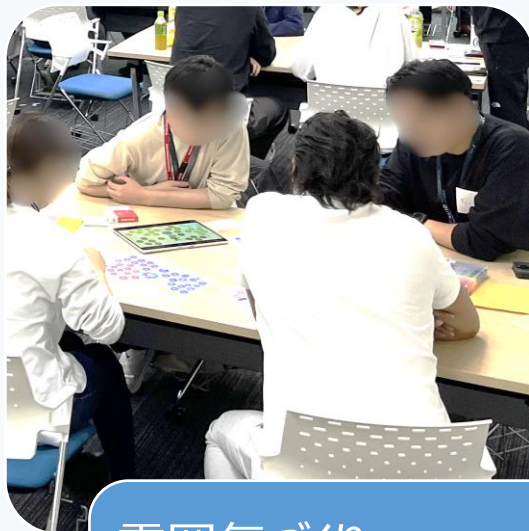


顧客体験の刷新 → 開発効率化

具体事例① ステークホルダーを集めてLego4Scrumの開催

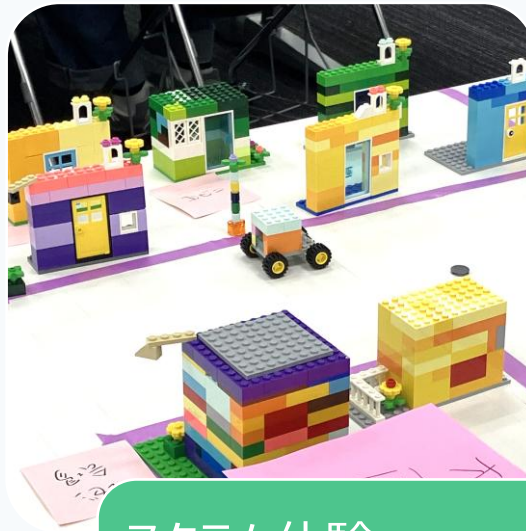
Day1

Day2



雰囲気づくり

- パーソナルマップの共有



顧客体験の刷新

スクラム体験

- プロダクトディスカバリー
- プロダクトデリバリー



クロージング

- FUN, DONE, LEARN
- プロジェクトでの今後のアクション検討

具体事例① ステークホルダーを集めてLego4Scrumの開催

成果

プロジェクト参画者（当社、ユーザー部門、ベンダー）の相互理解が進んだ

アジャイルな考え方、マインドセットの理解が深まった

参加者アンケート（抜粋）

- 実践形式を通じてアジャイル開発の考え方を体感できた
- POや各ステークホルダーの役割理解が深まった
- 3者（銀行・JRI・開発ベンダー）の連携やコミュニケーションの重要性を実感した
- 異なる立場の視点を理解する機会となった
- 会社の枠を超えた交流による学び・気づきが得られた

打ち手② プロセス：SAFe®で同期し、依存関係を早期に露出させる



PIプランニング

全チームが同期して計画を策定

プログラムボード

依存関係とリスクを可視化

2週間スプリント

各チームが短い単位で開発・検証

システムデモ

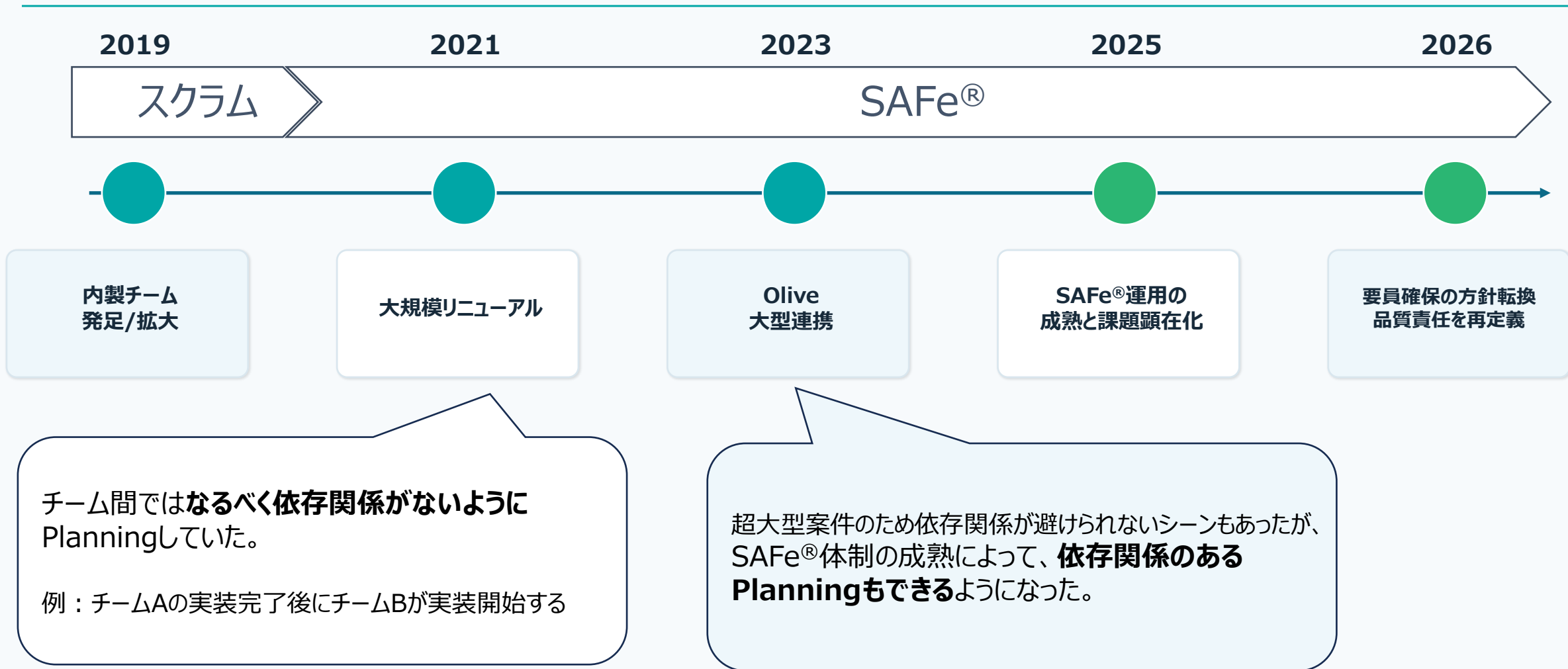
成果を統合し、ステークホルダーへ提示

Inspect & Adapt

実績からプロセスを改善

“計画で縛る”のではなく、依存関係による不具合や手戻りを
早期に発見する仕組み

具体事例② Olive構築：複数社・複数システムを横串で扱う

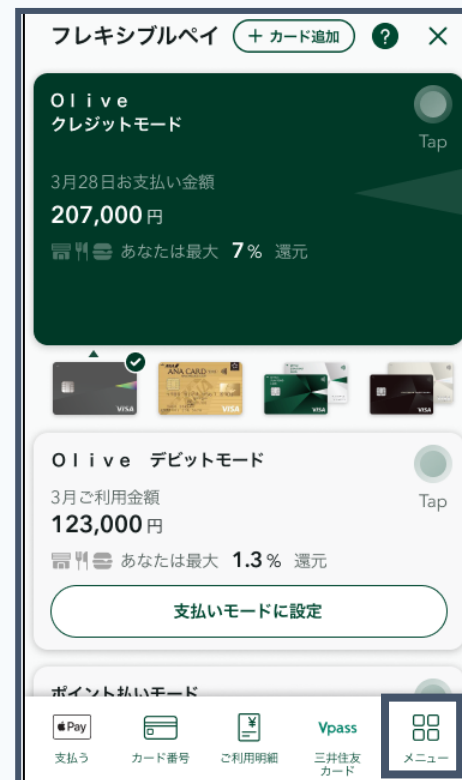


具体事例② Olive構築：複数社・複数システムを横串で扱う

ベンダー
ウィジェット



内製チーム
Fpayトップ



ベンダー
Fpayメニュー

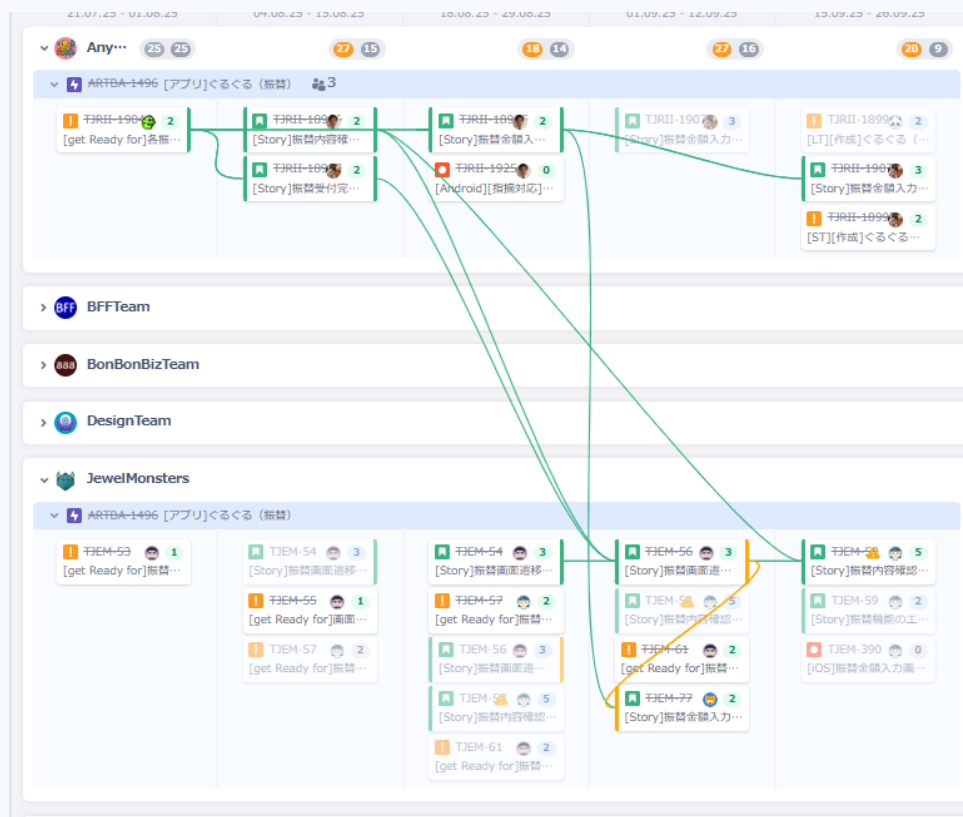


並行したスケジュールで開発を行った

具体事例② Olive構築：複数社・複数システムを横串で扱う

取り組んだこと

- ✓ ツールを使った依存関係の表現(チーム間を含む)
- ✓ PI Planning時に他チームへの依存関係を受け入れることができるか調整する
- ✓ 週2回の全チーム進捗共有で課題・遅れがあればすぐに共有する

[illegible]

「受け入れるか？」の記述ガイド

依存関係によってTeamの計画に影響ない	●
依存関係によってTeamの計画を少し変えないといけないが、Team内で解決できる	▲
依存関係の対応はTeam内でできないため、他のTeamと会話したい	✕

他チーム起因の待ちや手戻りを抑えながら推進することができた

打ち手③ 要員の確保：スコープ遵守から価値最大化へ



Before : 開発案件ごとに要員調達

- 要員が変動するためプロジェクトが終わると有識者が去っていく
- プロダクト価値に基づいて柔軟に対応を変えることが難しい

変更は追加調整・対立の火種になりやすい

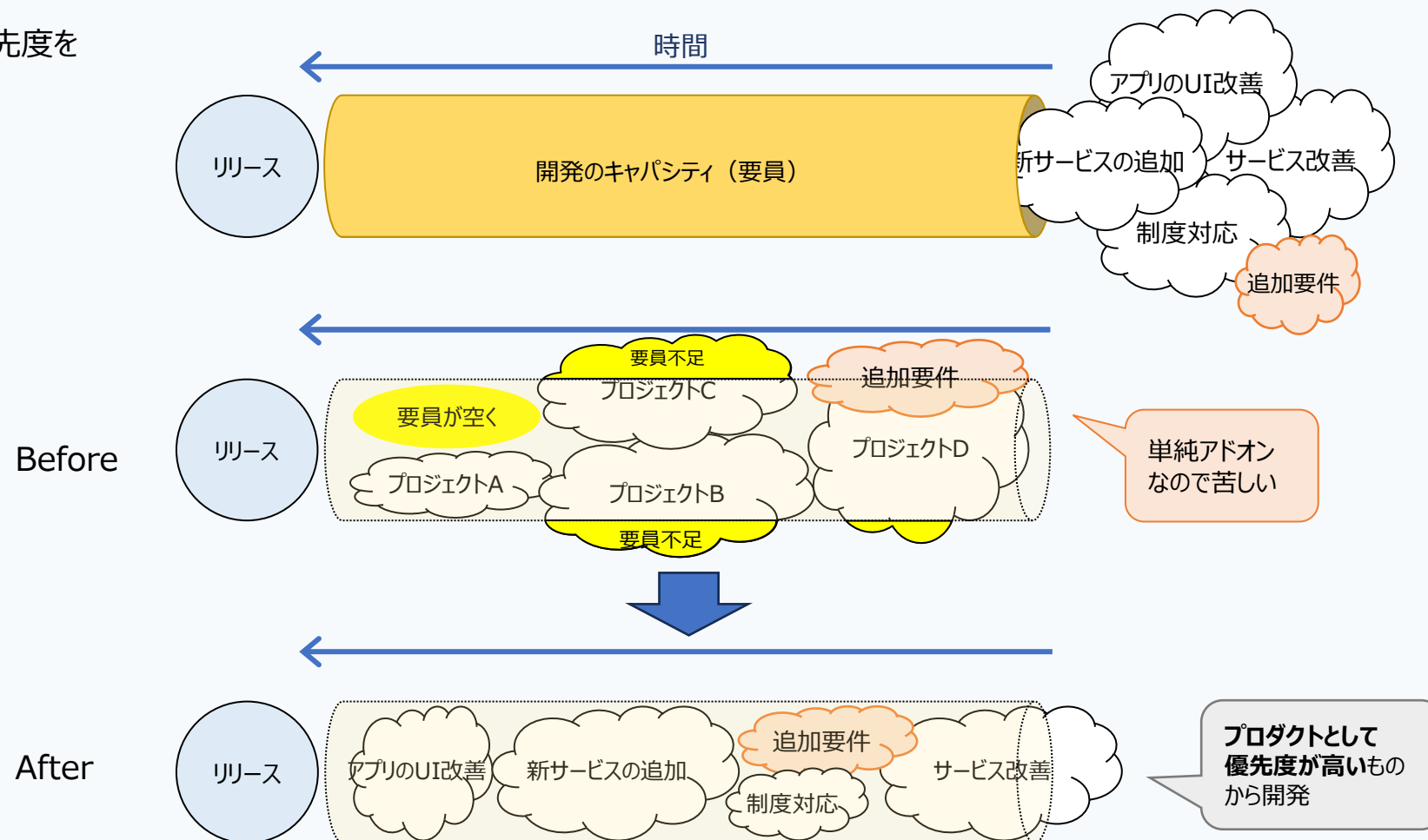
After : 要員固定の体制

- プロジェクト経験者の継続的な参画
- プロダクト価値の最大化にフォーカス

優先順位変更を受け止め、スピード感を持って対応

打ち手③ 要員の確保：スコープ遵守から価値最大化へ

要員確保方針の変更により、優先度を考慮した計画が立てやすくなった



実施結果：得られた3つの成果

1 組織的レジリエンス

人の入れ替わりがあっても品質を維持できる体制へ

昨年度実績：重大障害発生なし、軽微障害1件

Outcome

2 生産性と保守体制

プロジェクト体制が成熟し継続して大規模開発ができています

2026年度も1500人月超の開発を予定

Outcome

3 真のワンチーム化

当社、ユーザー部門、ベンダーが協働する文化が醸成されてきた

2026年度からオフィスを1箇所に集結

Outcome

他組織に展開できる示唆

1 フレームワークは“同期装置”であり、目的ではない

イベントを整えるだけでなく、依存関係・リスク・意思決定を見える化する。

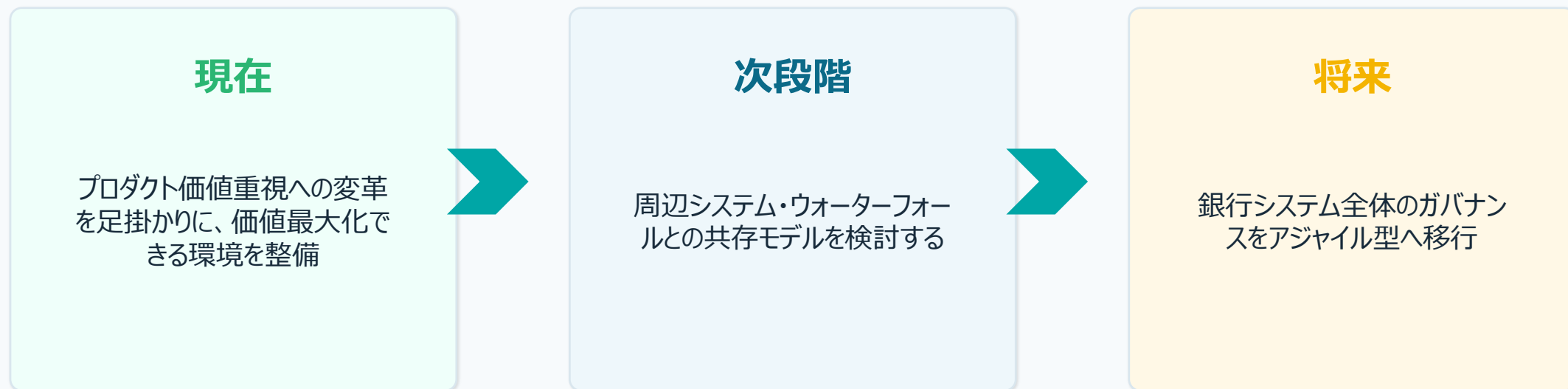
2 人の流動性はなくなる。だから教育を標準化する

オンボーディングを「初日研修」ではなく、継続的なナレッジ循環にする。

3 良いプロダクトは価値にフォーカスできる環境から生まれる

スコープ・スケジュール重視の文化から脱却し、真のワンチームとして価値にフォーカスできる環境をつくる。

今後の展開：アジャイル型ガバナンスへ



残課題：周辺システムとの連携にはウォーターフォールとの摩擦が残る

ご清聴ありがとうございました