

アジャイルの受託開発実現に向けた研究 ～受託開発は日本型アジャイルとして浸透できるか～

2023.02.01

2022年度

アジャイル開発におけるプロジェクトマネジメント 分科会

本日のお話

- アジャイルやったことある？
- なぜ、このテーマに？
- ロープレやりました！
- こんなもの、出来ました！
- 研究の評価

研究メンバーの紹介



赤岩 祐真
エクサ



河合 哲二
NSKネットアンドシステム



有田 雅人
ダスキン



高山 隆一
アシスト



西原 中也
アイアクト



岡崎 徹
日本テクノス



今井 悠介
日本オフィス・システム



田中 勝之
アシスト

1

アジャイルやったことある？



本研究の初動（2022年2月）

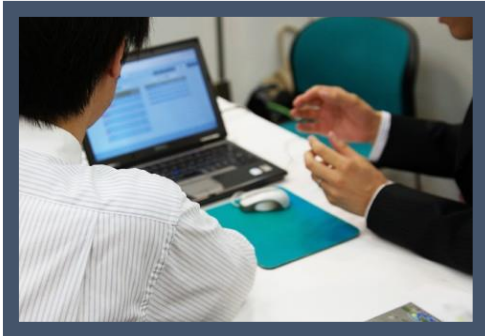
研究テーマ：アジャイル開発におけるプロジェクトマネジメント



第1回研究会での様子

まずは 情報収集 から

アジャイルのマネジメントについて情報収集



WEB

- Webサイトには、アジャイル情報は豊富だが、マネジメントに関する情報は、ほとんど無い
あっても、スプリント部分が中心



書籍

- アジャイル本と言われるものには、欲しい情報が無い
- 書店やAmazon等でも、見つからず



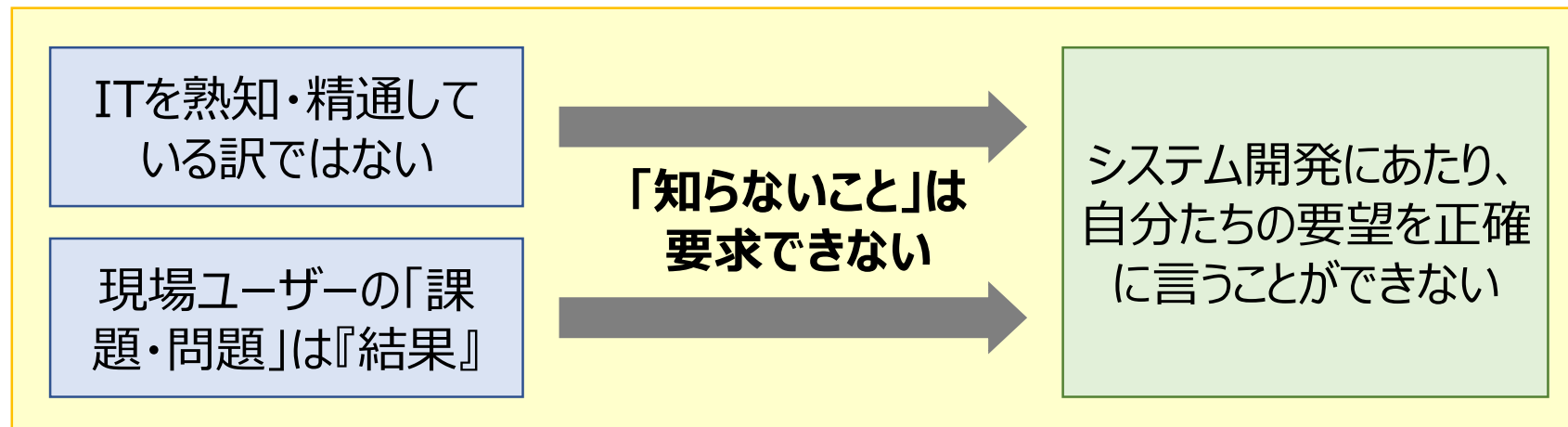
セミナー

- よさそうなセミナーは有償が多い（金額も高い）
- 無償セミナーはTool系のものが多い（もう一つ）

情報収集した結果

なぜアジャイルなのか？(DX案件の例)

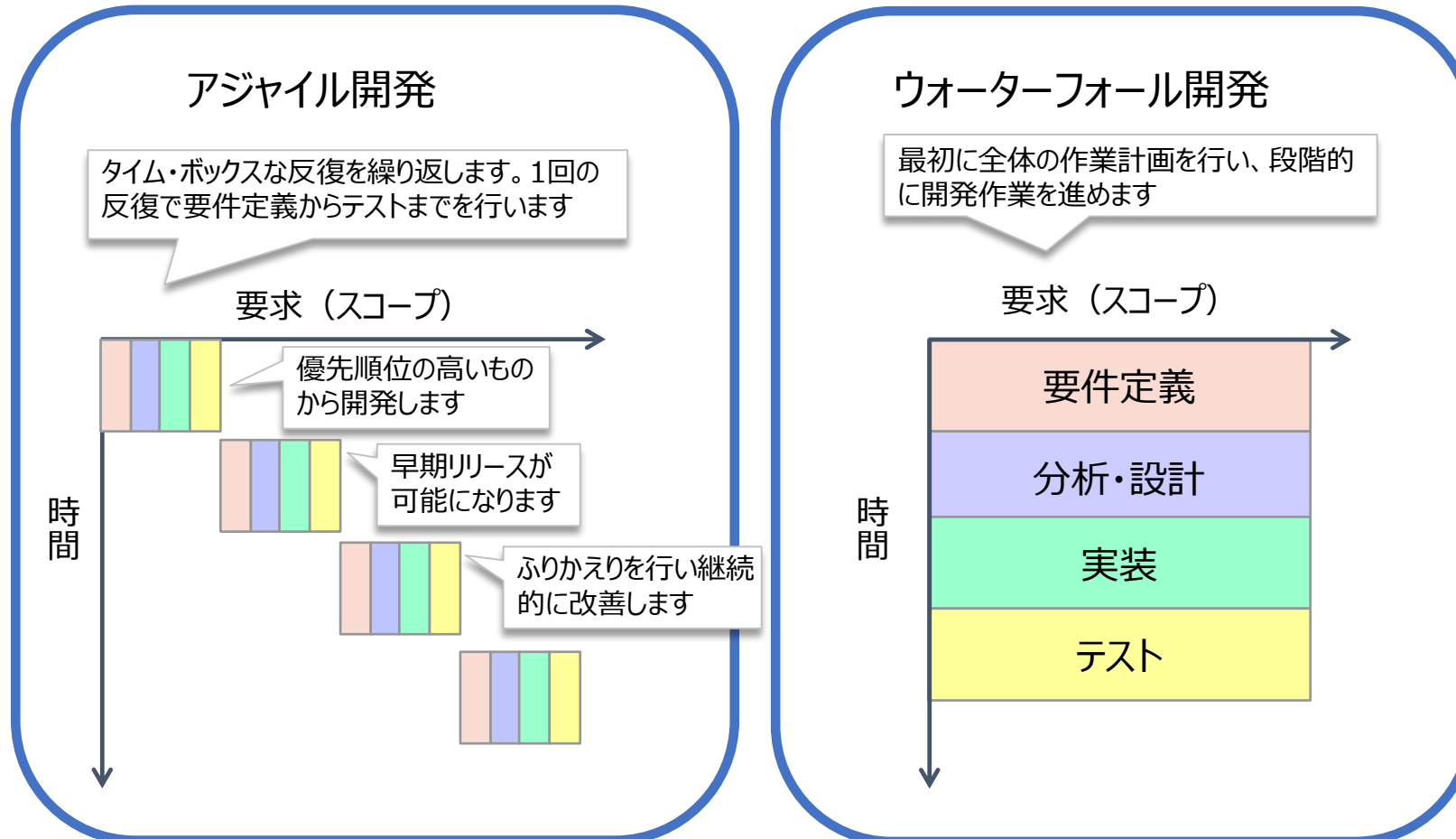
- ユーザーが要望を正確に言えないことが普通である



正確な要求を伝えきれないため、
段階的に実装して確認しながら進める必要がある。

情報収集した結果

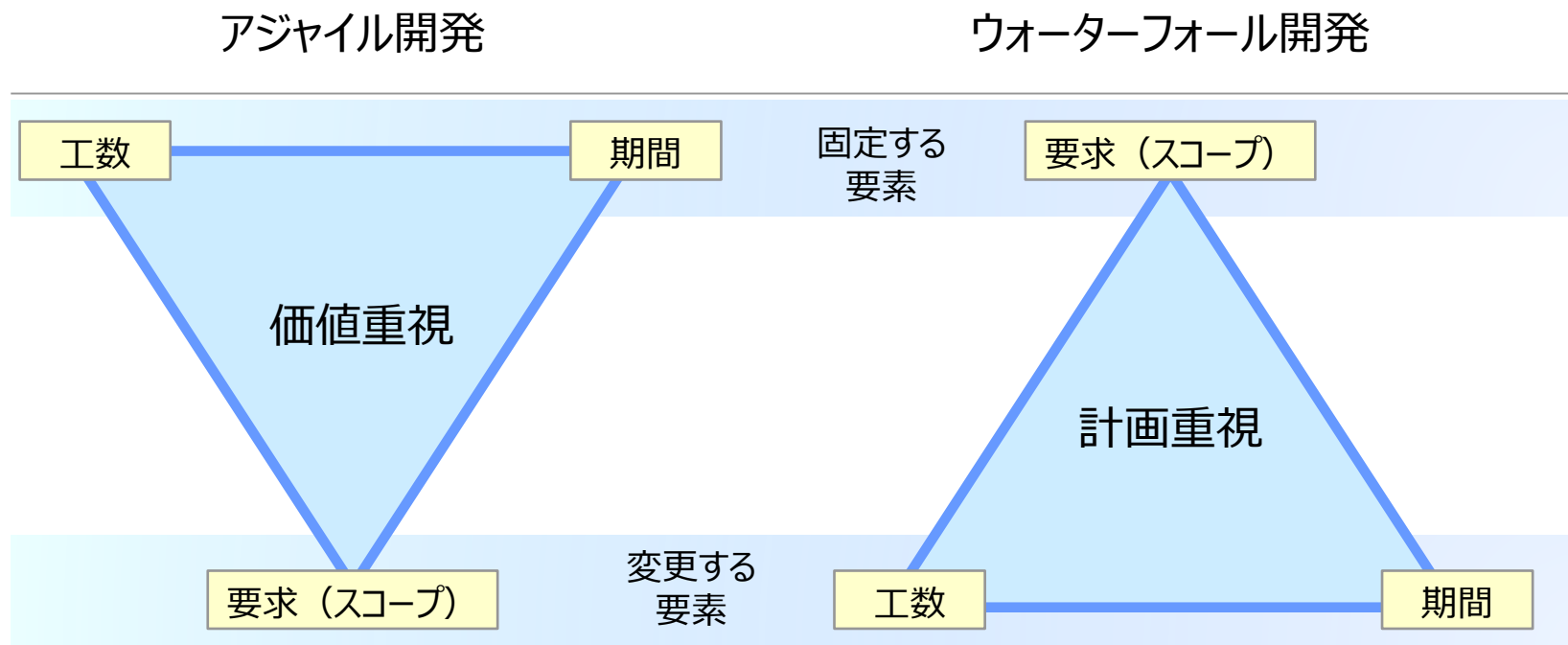
アジャイルの特徴



注) タイム・ボックスとは、固定された期間を表し、反復における作業期間を表す時間の箱のことです

情報収集した結果

スコープを固定せずに管理する



出典:日本IBM「IBM GBSにおける Agile 開発」より加工して作成

Agileでは要求 (スコープ) を固定するのではなく、反復作業における期間と工数を固定して変化に対応し、要求 (スコープ) は柔軟に変更

情報収集した結果

アジャイルは管理が難しい

- 随時仕様変更が可能のため、いつまでもスケジュールの全容が確定しない
- コミュニケーション重視の開発手法であるため、メンバー間の意見交換がうまくいかないとスケジュールが伸びがち
- 複数の開発サイクルを反復するため工程ごとの期限が明確で無い場合が多く、締切りを守らない・処理漏れが発生するなどの事態になりがち

ウォーターフォールのような管理のための標準プロセス的なものは存在しない

「外部から取ってきた知識をそのまま適用しようとしてもうまくいかない」
「実践に支えられた知識を頼りにする」という考え方が存在

アジャイルっていいじゃん！

なぜ、多くの企業ではアジャイルが浸透していない？

- 管理の難しさはあるが、それを超える良い面が多い
- ウォーターフォール型の開発には限界がある
- DX等の推進は企業にとっては必須のはず

などなど



なぜだろ？

2

なぜ、このテーマに？



アジャイル開発は内製向き

最もポピュラーなスクラムでは、透明性、検査、適応の3本柱が重要

スクラムのガイドの多くは「Face to Face、同じ部屋で」ということが前提

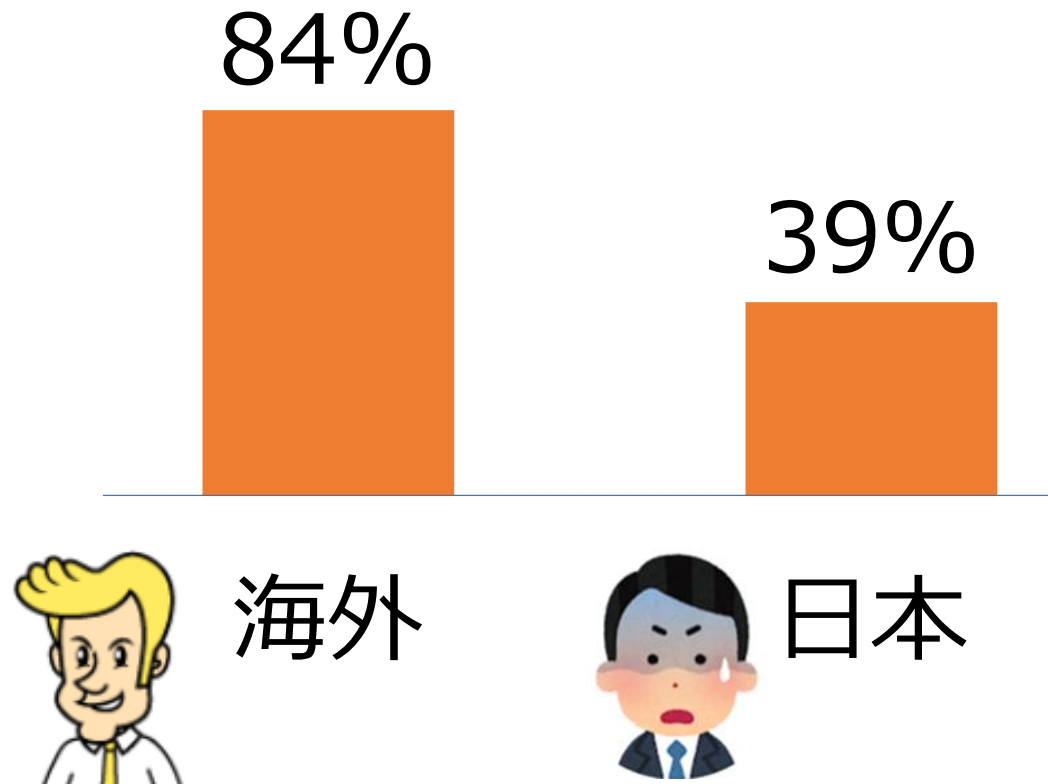
受託開発やSIでは合意形成が難しい

- アジャイル開発のように仕様変更が発生する案件は得意ではない
- アジャイル開発が適しているような案件では、完成品を納品できなくなるリスクがある
- クライアントの要望で方針が変わればスケジュールの見直しを迫られるなど、マネジメントが非常に困難

アジャイル開発の現状

日本におけるアジャイル開発導入の状況

- 2021年のアジャイル開発導入率



**海外の水準と
大きな差**

出典:15th State of Agile Reportより加工して作成

ベンダー依存体質

アジャイル開発の阻害要因

- ユーザー企業（顧客）と開発チーム間に契約をはさむ受託開発が多い（外注が主流）
- 契約の壁や、物理的に離れていたりで、コミュニケーションが取りにくい
- IT技術者はSIerやソフトハウスに所属していることが多い（IT人材の70%がベンダー企業に所属）



アジャイルの基本的な考えと真逆

ユーザーのジレンマ

多くのユーザーでは内製化の準備が進んでいない

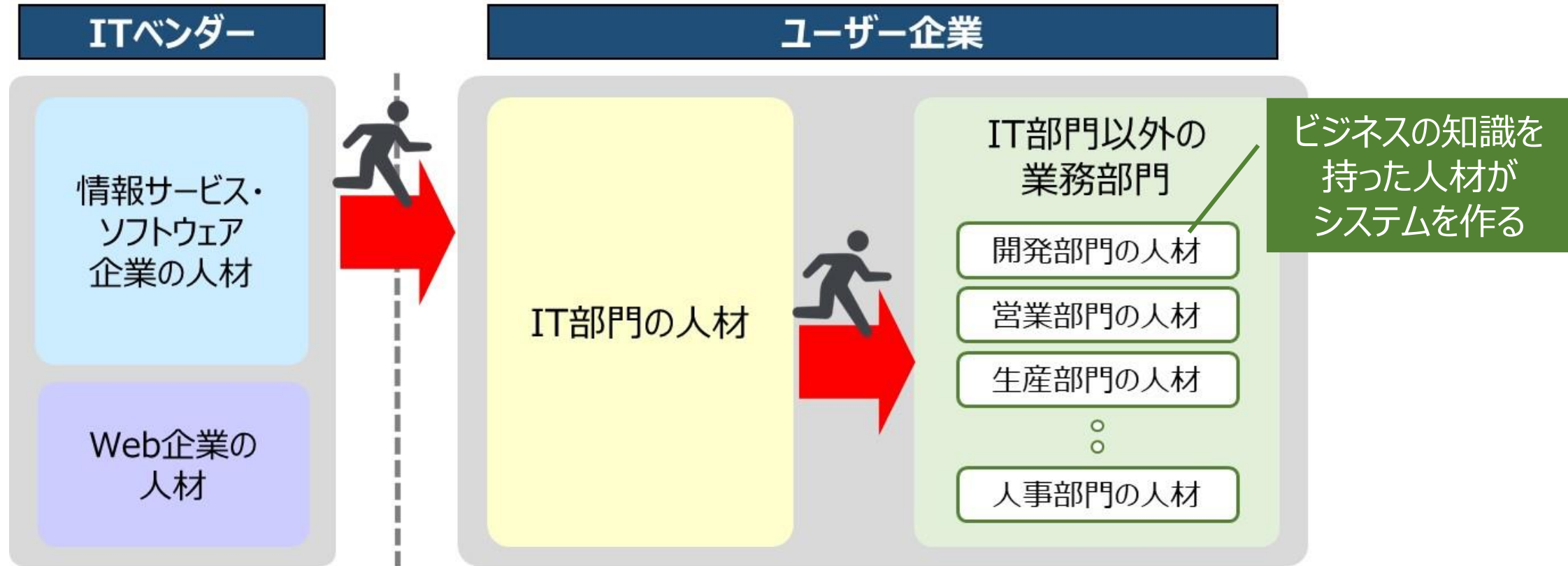
- 企業内部に経験者がいないので人財育成も進まない
- 派遣形態で要員を受け入れて、内製体制で開発するのは負担が大きすぎる
- 準委任契約でベンダーから支援を受けるとしても、解決しなければならない課題が多くある

ユーザー側の「契約≒成果保証」の意識

- アジャイル開発の場合、決められた期限内で何をどこまで実現するかを明示できない
- 掛ける費用に対して、結果を明確に示せないことと同義であり、発注者側の心理としては受け入れがたい

ユーザーのジレンマ

一足飛びに欧米並みの環境変化は訪れない



出典:みずほ情報総研株式会社,「IT人材需給に関する調査」調査報告書より加工して作成

じゃあどうする？

既に内製化に舵を切っているユーザ企業

YES

明るい未来

NO

現行システムの御守りで手一杯でもDX推進は待ってられない
ベンダーに頼らざるを得ない状況

だから

ベンダーと協力してアジャイルを進めていく必要がある

我々はそのための研究を開始した



アジャイルの受託開発に向けて

日本の状況からアジャイルの受託開発は必要(日本型アジャイル)

アジャイルの受託開発実現に向けて、ユーザーとベンダーが、どのような体制や役割でプロジェクトを実施すべきかの研究を行う



アジャイル開発の各プロセスにおいて、どのような管理ポイントがあるかを洗い出し、そこにどんな対策を行うべきかを検討する

アジャイルの受託開発に向けて

研究方針

- ユーザー、ベンダー共に参考になるアウトプットを作りたい

アジャイルの受託開発の成功において、ユーザー企業とベンダー企業との
合意形成やコミュニケーションが最も重要なポイント



プロジェクトの立ち上げ工程（提案・見積もり-計画-キックオフ）に注力し、開発範囲や見積もり、コミュニケーションの方法、体制に着目する



スプリント（イテレーション）の実施については開発手法の違いはあるものの、バックログの管理が出来れば、開発者には管理上大きな問題にならない

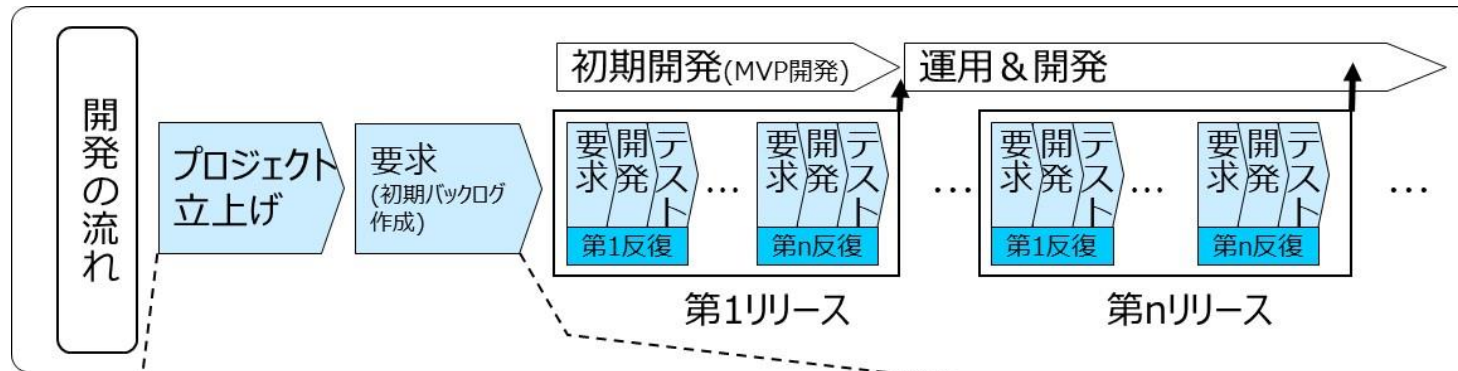
3

ロープレやりました！



立ち上げ工程に着目（研究方針）

『立ち上げ』工程はプロジェクト進行に大きく影響する



研究方針

- ユーザー、ベンダー共に参考になるアウトプットを作りたい

アジャイルの受託開発の成功において、ユーザー企業とベンダー企業との合意形成やコミュニケーションが最も重要なポイント

- プロジェクトの立ち上げ工程（提案・見積もり-計画-キックオフ）に注力し、開発範囲や見積もり、コミュニケーションの方法、体制に着目する

ビジョンの策定
• ビジネスのビジョン、戦略の共有
• プロジェクトとしての目標
• 基本的価値観の共有
• インセプションデッキ など

バックログの作成
• システムの目的やゴールの共有
• ユーザストーリーの洗い出し
• システムのアウトライン作成
• バックログの作成 など

プロジェクト計画
• プロジェクトチームの編成
• 役割分担の設定
• コミュニケーション計画
• マイルストーン設定 など

要員育成
• アジャイル基礎研修
• スクラム研修
• プロダクトオーナー研修
• 開発者研修 など

開発環境構築
• クラウド環境構築
• アジャイル関連ツール準備
• マイクロサービス基盤構築
• テスト環境整備 など

標準・ガイド整備
• アジャイルプロセス整備
• 標準化・ルール化
• アジャイル適用判定基準
• プロジェクト事例 など

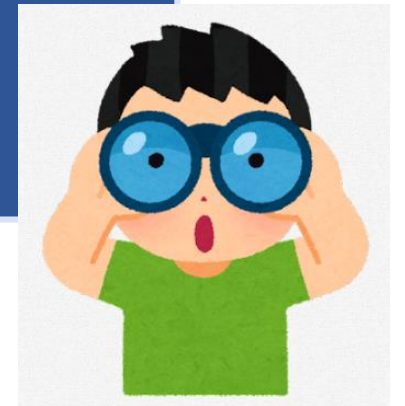
本論とは別の問題として、
研究の対象としない

研究の進め方

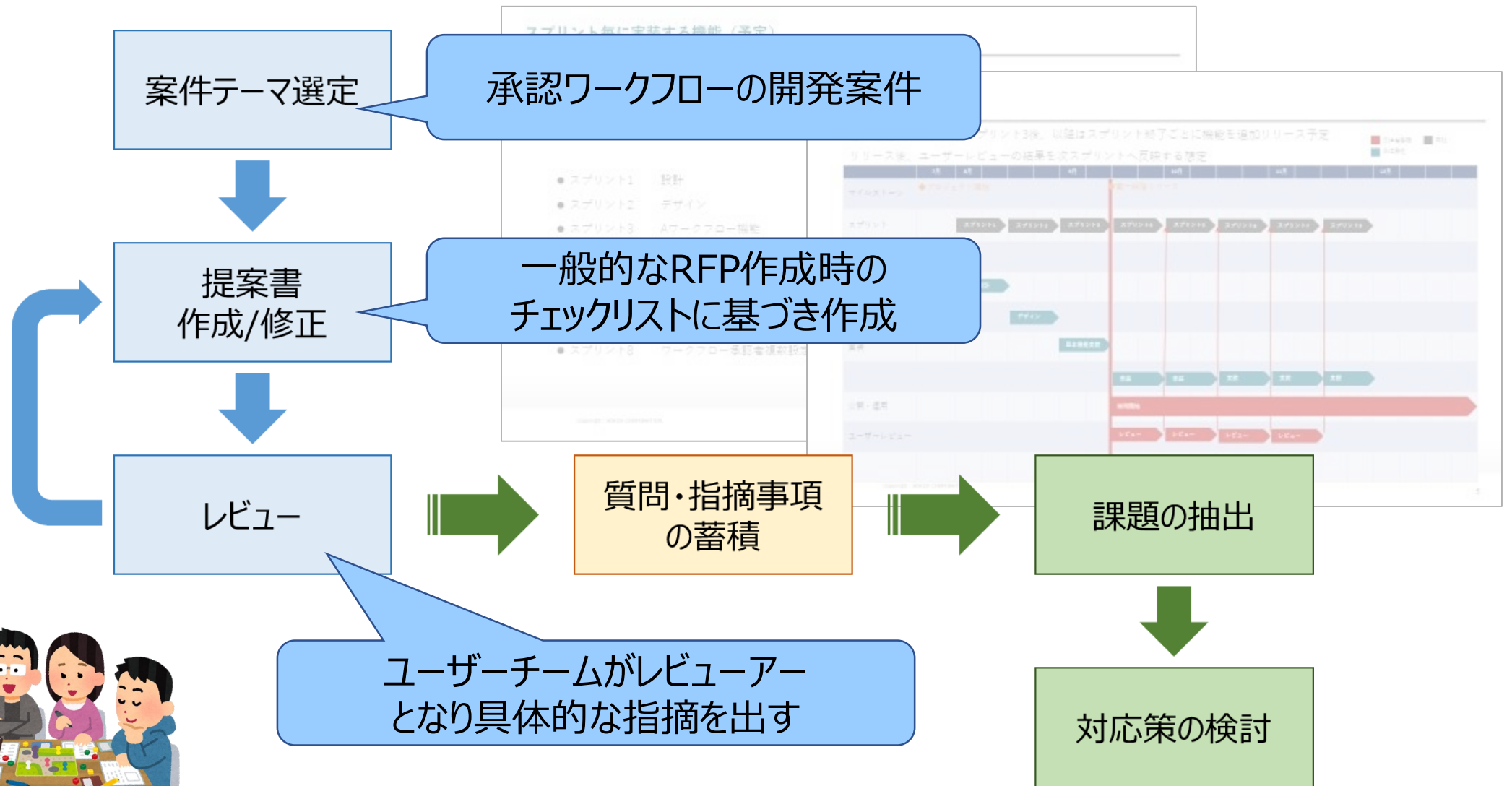
課題のあぶり出し

模擬プロジェクトにおける、提案からプロジェクト開始に至る立ち上げ部分に関するプロセスをロールプレイしてみる！

ロールプレイにて挙げられた課題事項から、アジャイル開発立ち上げ時の重要検討事項を一覧化し、優先度をつけて考察対象を絞り込む！



ロールプレイの流れ



ロールプレイでの指摘

ウォーターフォールでの提案をベースに試行錯誤

契約形態は？
請負or準委任？

開発体制は？

用語がわからない

役割分担については？

工程が分からない
期限内でどこまでできる？

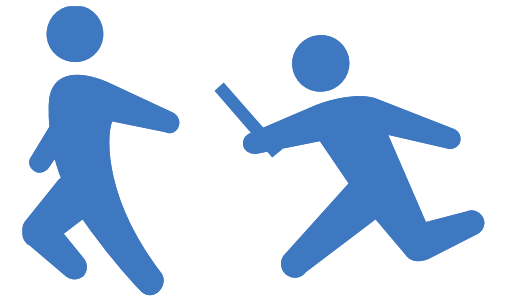


解決すべき4つの課題

契約形態



開発体制



合意形成
(信頼関係)



コミットメント



契約形態



ユーザーが主体性を持って開発することが望ましい

ベンダー依存度の高いユーザー企業にとって、派遣形態で要員を受け入れて、内製体制で開発する事は負担が大きすぎる



- 受託開発で推奨される契約形態はなにか
 - ✓ 請負？準委任？
 - ✓ 偽装請負とならないための体制

開発体制

従来の受託開発に比べユーザー側の負担が大きい



「利用現場の真のニーズに則した想定される以上の機能を作る」ことができるというアジャイルの価値をユーザー側が理解し、負担増を受容する必要がある



- 特に負担の大きいユーザー側プロダクトオーナーの負荷軽減
- ユーザー側とベンダー側という分断された体制の見直し

合意形成（信頼関係）



積極的なプロジェクト関与が不可欠

ボールを投げる・受けるという旧来の形ではなく、アイデアを共に出し機能に帰結させる協力の形が求められる



- ベンダー側からの巻き込み、ユーザー側の意識改革や協力体制がプロジェクト成功の重要なカギとなる
- 立ち上げ段階で共通理解を深めるなど合意形成を図る方法とは

コミットメント



受託開発は暗黙の成果保証がセットの文化

開発する機能の追加・変更やその優先順位に変更の可能性があるため、機能保証・成果としてコミットメントすることは難しい



- ユーザー側を納得させるコミットメントとは
 - ✓ 初期段階で見えている要求に対して最初の実現範囲の調整
 - ✓ スプリント毎の真摯・誠意あるプロセス管理などを提示・表現

4

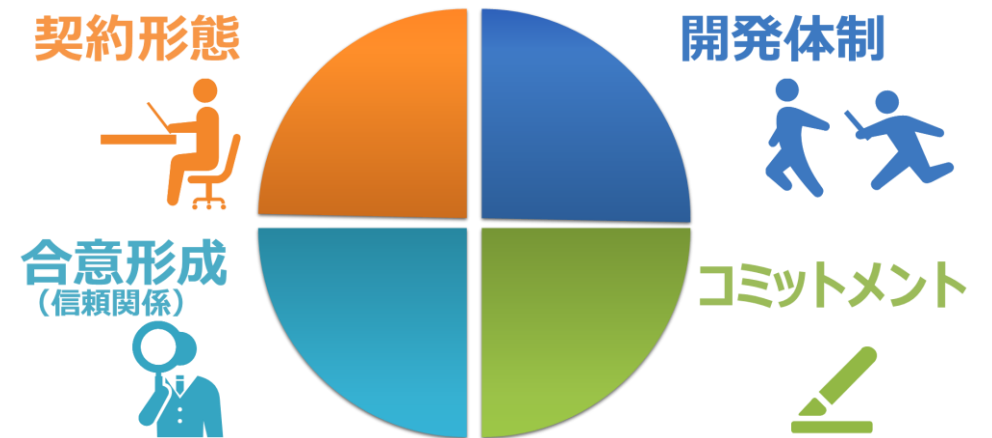
こんなの、出来ました！



受託開発に向けて実施すること

4つの課題に対してどう対応するべきか？

- 受託開発の特性を考慮しながら、アジャイルの特性を活かすような方向を探る
- アジャイル開発手法については、最も普及していると思われるスクラムを前提
- エクストリームプログラミング（XP）の手法なども取り込む



契約形態

- 受託開発で推奨される契約形態はなにか
 - 請負？準委任？
 - 偽装請負とならないための体制

契約形態



契約形態



アジャイルの受託開発には準委任契約

請負契約



アジャイルの特性からすれば、あらかじめ内容が特定された成果物を予定したとおりに完成させることに対して対価を払う請負契約では、完成できないリスクが高い

準委任契約



業務を受託したベンダー企業が専門家としての注意義務を果たしながら業務を遂行することそれ自体に対価を支払う準委任契約の方が適している



ユーザー側の「契約≒成果保証」の意識を変えて、プロジェクト成功に向けて、ユーザーとベンダーが協調して推進していく覚悟が必要である

契約形態



偽装請負リスクの回避

発注者側と受注者側の開発関係者が相互に密に連携し、随時、情報の共有や、システム開発に関する技術的な助言・提案を行っていたとしても、実態として、発注者と受注者の関係者が**対等な関係の下で協働し、受注者側の開発担当者が自律的に判断して**開発業務を行っていると認められる場合であれば、偽装請負と判断されるものではありません。

『労働者派遣事業と請負により行われる事業との区分に関する基準』（37号告示）に関する疑義応答集より抜粋

自らの特性やプロジェクトの内容に合わせて最適な進め方を模索し、柔軟かつ自律的に改善を行うというアジャイル開発の特徴を活かしながら、開発を進めていくためには、開発体制における仕組みづくりが重要（開発体制と併せて進める）

開発体制

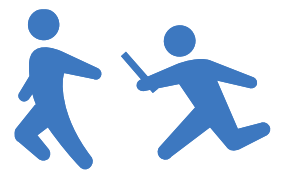
- 特に負担の大きいユーザー側プロダクトオーナーの負荷軽減
- ユーザー側とベンダー側という分断された体制の見直し



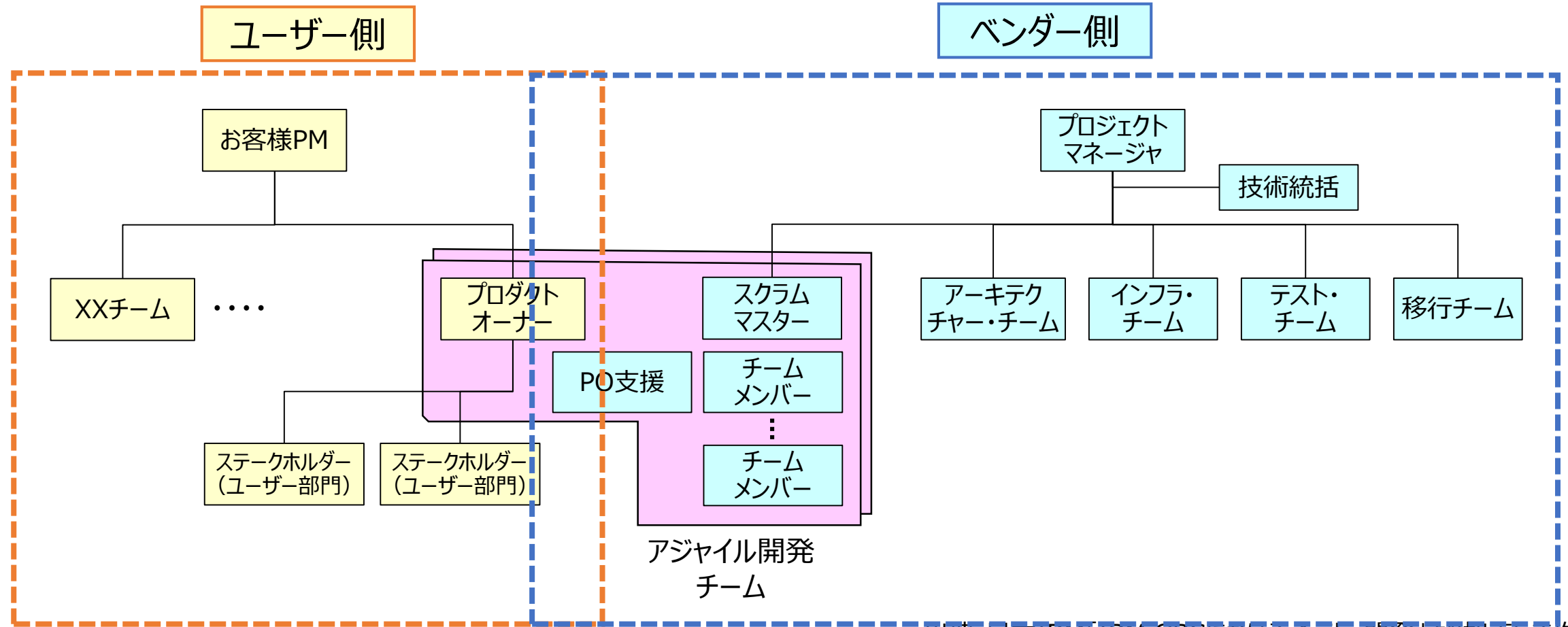
開発体制



開発体制

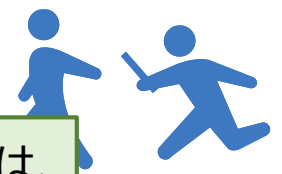


プロジェクト体制のイメージ

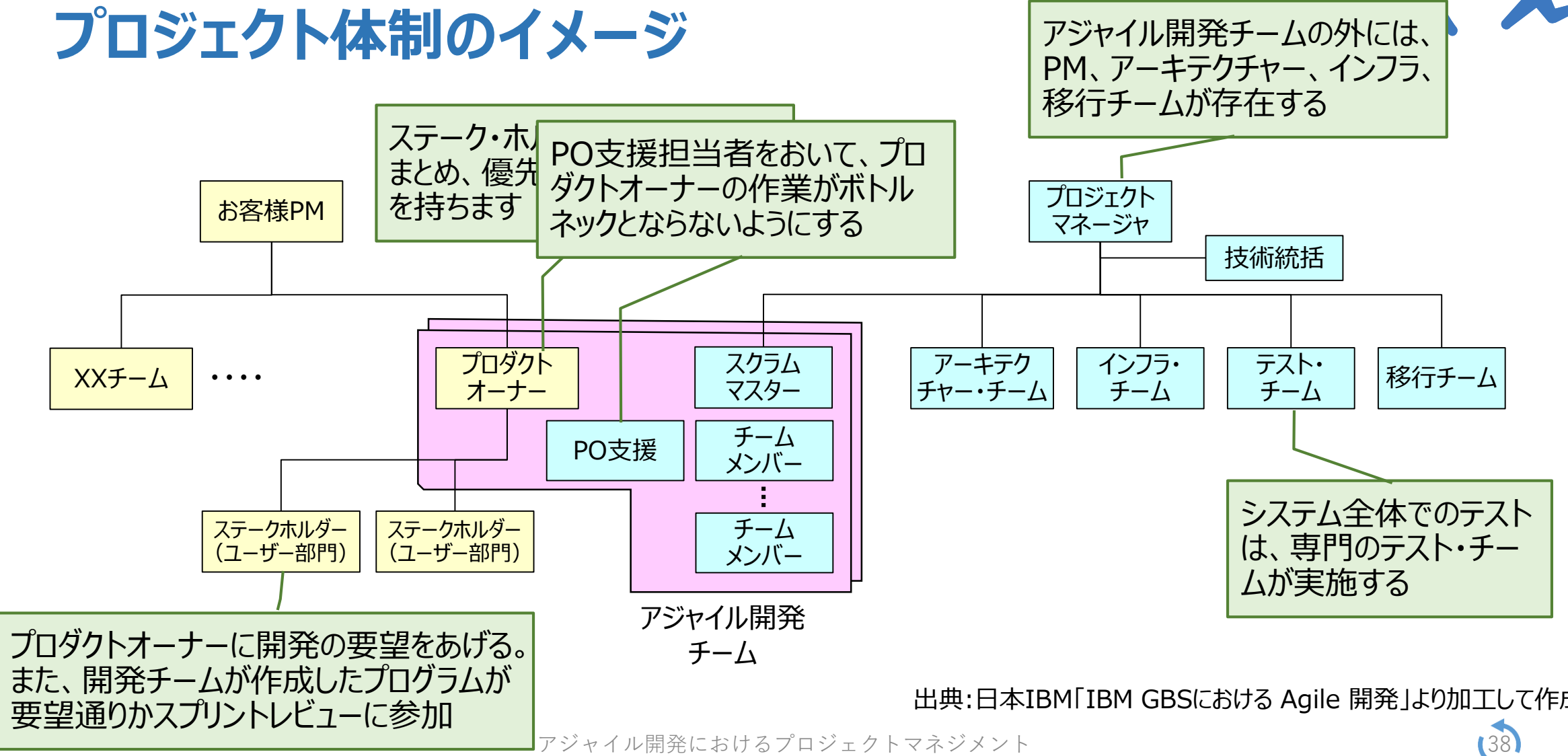


出典: 日本IBM「IBM GBSにおける Agile 開発」より加工して作成

開発体制



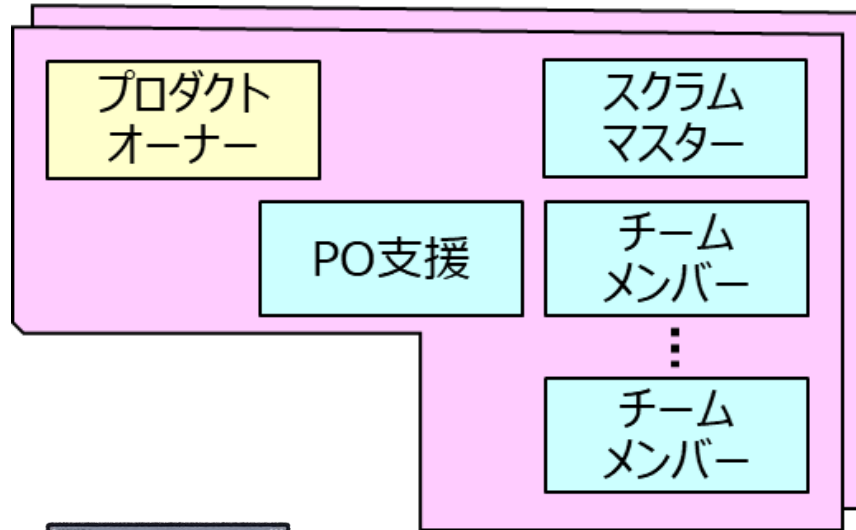
プロジェクト体制のイメージ



開発体制



アジャイル開発チーム



アジャイル開発
チーム



- プロダクトオーナー以外は、ベンダー側で構成
- ユーザーの業務を理解し、協調して推進するためにビジネスアナリストの知識を有するメンバーを加える
- プロダクトオーナーの作業がボトルネックとならないようにするため、ベンダー側からPO支援担当者を出す
- 上記以外に開発メンバーとの要求確認などのコミュニケーションをスムーズにするとともに偽装請負リスク回避も目的
- 開発対象の変更において大枠の変更は、契約内容の変更として開発チーム以外のPMを含めて変更協議
- 小さな変更はバックログの追加・変更というアジャイルプロセスの中での対応

合意形成（信頼関係）

- ベンダー側からの巻き込み、ユーザー側の意識改革や協力体制がプロジェクト成功の重要なカギとなる
- 立ち上げ段階で共通理解を深めるなど合意形成を図る方法とは

合意形成
（信頼関係）



合意形成（信頼関係）



従来開発に比べてユーザーの役割も負担もはるかに大きい

- ユーザーとベンダー間での密なコミュニケーションを必要とする

ロール	スプリント プランニング	デイリースクラム	スプリント レビュー	スプリント レトロスペクティブ	リファインメント
プロダクトオーナー	◎	△	◎	◎	◎
開発チーム	◎	◎	◎	◎	◎
スクラムマスター	◎	○	◎	◎	○
ステークホルダー	△	△	○	△	△

◎絶対に参加 / ○参加しなくて良いこともある / △チームの求めに応じて参加 / ×参加できない

合意形成（信頼関係）



ユーザーの積極的関与による協力体制を確立

立ち上げ時にプロジェクトの目的、ゴール、スコープや価値などについて理解を深め、合意形成を図り、互いに協調してプロジェクトを推進していくことで、信頼関係を築くことが不可欠

立ち上げ時に関係者を集めて、インセプションデッキの作成を必須とすべき

- プロジェクトメンバーが目的・ゴールに対して共通認識を持てる
- プロダクトの利用者・価値を明確化できる
- プロジェクトのスコープを明確にできる
- 起こり得るトラブルや課題を洗い出し、事前に対応方針を検討できる
- プロジェクトでの優先順位・重要度を明確にできる

合意形成（信頼関係）



インセプションデッキとは？

- プロジェクトのWhyとHowを明確にするために作成
- 10項目の内容をチームで作成する

1. 我われは なぜここに いるのか？	2. エレベーター ピッチ	3. パッケージ デザイン	4. やらないこと リスト	5. ご近所さん を探せ
6. 技術的な 解決策を 描く	7. 夜も 眠れない 問題	8. 期間を 見極める	9. トレード オフ スライダー	10. 何がどれだけ 必要か

合意形成（信頼関係）



インセプションデッキを実施した結果

- 実施した結果、下記のような意見が得られた

問題が発生したときだけでなく、
方向性を見失いそうになった際
の指標にできそう

全10項目じゃなくても、項目を
選択した上で、絞って実施しても
効果はありそう

**関係者全員で実施することで合意形成を得ることは可能！
時間はかかるが作成すべき！**

コミットメント

- ユーザー側を納得させるコミットメントとは
 - 初期段階で見えている要求に対して最初の実現範囲の調整
 - スプリント毎の真摯・誠意あるプロセス管理などを提示・表現



コミットメント

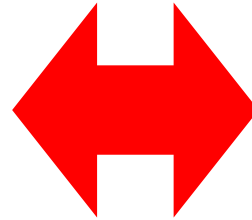


コミットメント



ユーザーの心情？

ベンダー
アジャイルを準委任契約で！



ユーザー
成果物の保証がされない
のは情動的に辛い…

準委任により成果物の納品義務がなくなったと考えるのではなく
成果に価値を与える努力が必要ではないか

コミットメント



MVP (Minimum Viable Product) の決定

※MVP:顧客に価値を提供できる最小限のプロダクト

- 契約内容の調整段階、遅くともプロジェクトの立ち上げ段階（スプリント0以前）までにMVPを決定
- ユーザーが抱える課題を解決できる最低限の状態プロダクトを提供する
- 提供後は、ユーザーからのフィードバックなどを参考にし、新機能の追加や改善点の見直しを図ることでプロジェクト方針の合意をする

受託アジャイルにとってMVPの決定がより重要になる

コミットメント



MVP合意にあたっての注意点

- ユーザー企業のステークホルダーを含めたプロジェクト関係者にて合意を行うこと
- MVPが提案工数内にて実現可能であり、かつ追加要件への対応工数の余白を残していること

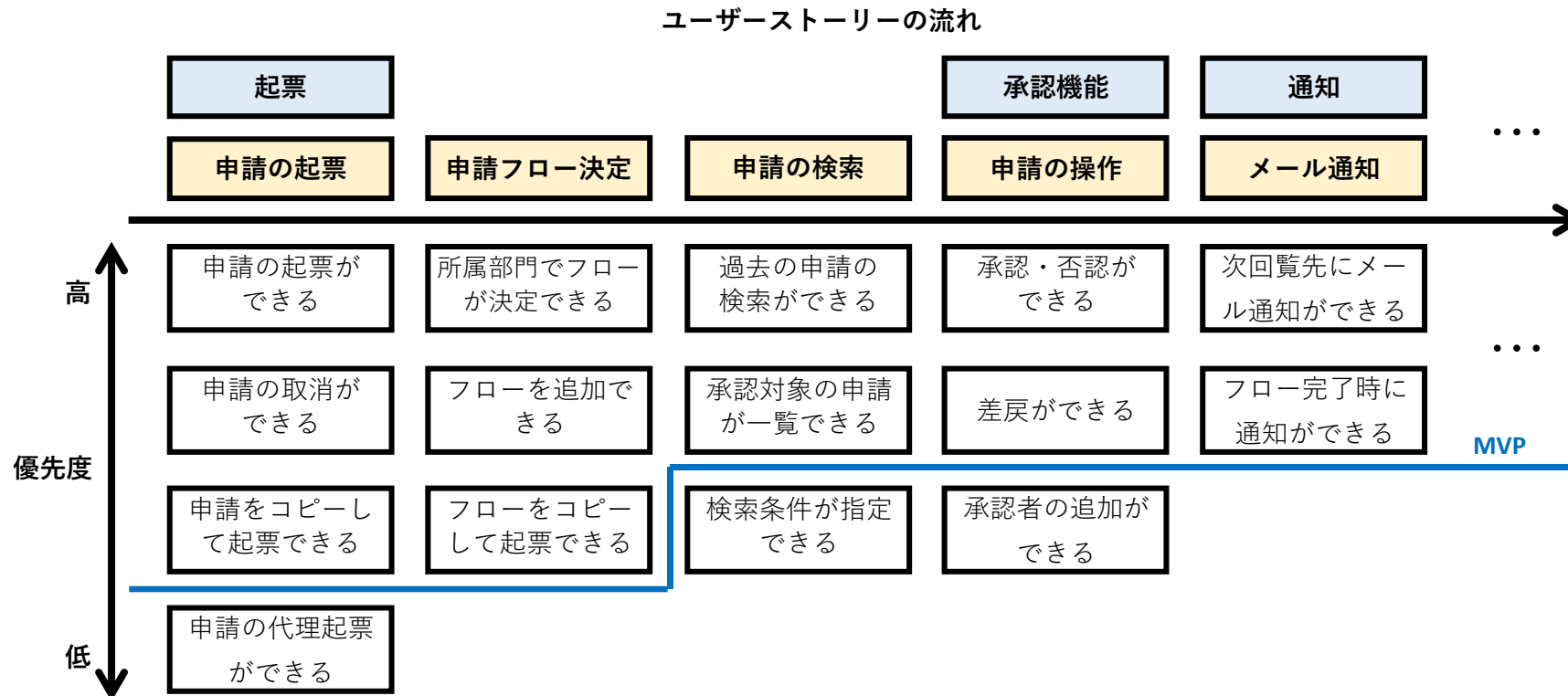


コミットメント



MVP作成はユーザーストーリーマッピングで

- プロダクトを実際に利用するエンドユーザーに何を提供するのか
- ユーザーストーリー（時系列）×優先度でマッピング



コミットメント



ユーザーストーリーマッピングを使用するメリット

- プロダクトの価値が見える化される
- 要求事項の整理に顧客が参加しやすい
- 機能過多を防ぐことができる（全体像の把握）
- 優先すべき課題が明確になる
- 洗い出した項目と優先順位が最初のプロダクトバックログに整理される（スクラムでの管理）

ユーザーストーリーマッピングは、最初のプロダクトバックログとして整理され、フィードバックを受けてリファインメントされ続けられる

5

研究の評価



**ロールプレイングをきっかけとして
アジャイルが受託で通用するのかを
肌で感じられた**

- フレームを使うのは、意味があるのか
- 今までの進め方との違い
- 今までの役割との違い

感じられたことを、どうまとめよう

think

私たちの研究は、世に何を残せるのか

ans

**ウォーターフォールで使ういつものモノを
アジャイルで使えるモノとして定義**

final ans

誰でも活用できるように

私たちの研究で、概論・指針をフレームに

IPAがまとめている「アジャイルの指針」

● 契約前チェックリスト とても参考になる重要な「指針」

項目	チェックポイント
1. プロジェクトの目的・ゴール	プロジェクトの目的（少なくとも当面のゴール）が明確であるか
	ステークホルダーの範囲が明確になっているか
	目的についてステークホルダーと認識が共有されているか
2. プロダクトのビジョン	開発対象プロダクトのビジョンが明確であるか
	開発対象プロダクトのビジョンについてステークホルダーと認識が共有されているか
3. アジャイル開発に関する理解	プロジェクトの関係者（スクラムチーム構成員及びステークホルダー）がアジャイル開発の価値観を理解しているか
	プロジェクトの関係者がスクラムを理解しているか
4. 開発対象	開発対象プロダクトがアジャイル開発に適しているか
	1チーム（最大で10名程度）の継続的対応にて、開発可能な規模であるか
5. 初期計画	プロジェクトの初期計画が立案されているか
	プロジェクトの基礎設計が行われているか
	完了基準、品質基準が明確になっているか
	十分な初期バックログがあるか（関係者間で初期のスクールの範囲が合意できているか）
6. 本契約に関する理解	本契約が準委任契約であることを理解しているか
7. 体制（共通）	ユーザ企業とベンダ企業の役割分担を理解しているか
	今回のプロジェクトにおける体制を理解しているか
8. ユーザの体制	適切なプロダクトオーナーを選任し、権限委譲ができるか
	ユーザ企業としてプロダクトオーナーへの協力ができるか
9. ベンダの体制	アジャイル開発の経験を有するスクラムマスターを選任できるか
	必要な能力を有する開発チームを構成できるか
	開発チームを固定できるか

チェック項目

- プロジェクトの目的（少なくとも当面のゴール）が明確であるか
- ステークホルダーの範囲が明確になっているか
- 目的についてステークホルダーと認識が共有されているか

どうチェックすればいいの？

IPAがまとめている「アジャイルの指針」

項目	チェックポイント
1. プロジェクトの目的・ゴール	プロジェクトの目的（少なくとも当面のゴール）が明確であるか
	ステークホルダーの範囲が明確になっているか
	目的についてステークホルダーと認識が共有されているか
2. プロダクトのビジョン	開発対象プロダクトのビジョンが明確であるか
	開発対象プロダクトのビジョンについてステークホルダーと認識が共有されているか
3. アジャイル開発に関する理解	プロジェクトの関係者（スクラムチーム構成員及びステークホルダー）がアジャイル開発の価値観を理解しているか
	プロジェクトの関係者がスクラムを理解しているか
4. 開発対象	開発対象プロダクトがアジャイル開発に適しているか
	1チーム（最大で10名程度）の継続的対応にて、開発可能な規模であるか
5. 初期計画	プロジェクトの初期計画が立案されているか
	プロジェクトの基礎設計が行われているか
	完了基準、品質基準が明確になっているか
	十分な初期バックログがあるか（関係者間で初期のスクールの範囲が合意できているか）
6. 本契約に関する理解	本契約が準委任契約であることを理解しているか
7. 体制（共通）	ユーザ企業とベンダ企業の役割分担を理解しているか
	今回のプロジェクトにおける体制を理解しているか
8. ユーザの体制	適切なプロダクトオーナーを選任し、権限委譲ができるか
	ユーザ企業としてプロダクトオーナーへの協力ができるか
9. ベンダの体制	アジャイル開発の経験を有するスクラムマスターを選任できるか
	必要な能力を有する開発チームを構成できるか
	開発チームを固定できるか

● 契約前チェックリスト

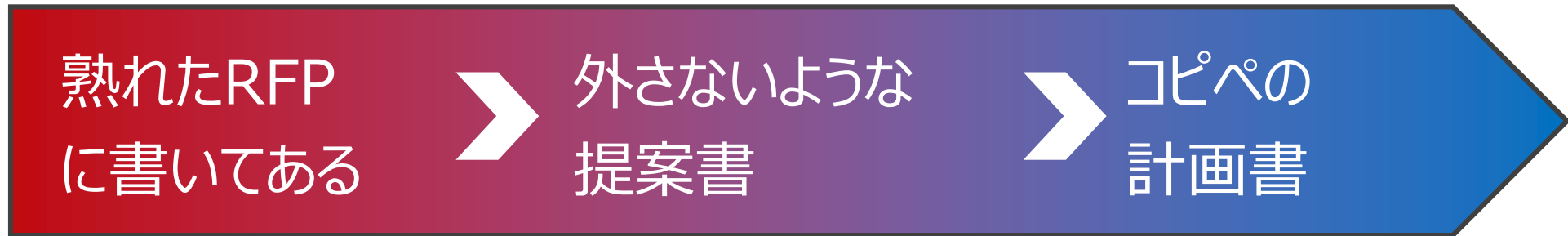
とても参考になる重要な「指針」

「従来の受発注体質が色濃く残る」

私たちでは活用しきれない

今までどう確認していたか

いままでの方法



**最初からなんとなく、目的が書かれていて
なぞりながらブラッシュアップした
「プロジェクト計画書」で事足りていた**

チェック項目

- プロジェクトの目的（少なくとも当面のゴール）が明確であるか
- ステークホルダーの範囲が明確になっているか
- 目的についてステークホルダーと認識が共有されているか



- プロジェクトの目的を明確にする方法
- ステークホルダーの範囲を明確にする方法
- 目的についてステークホルダーと認識を共有する方法

チェック項目を「方法」に落とし込む

例えば「目的・ゴール」について

IPAの項目

項目	チェックポイント
プロジェクトの目的・ゴール	プロジェクトの目的（少なくとも当面のゴール）が明確であるか
	ステークホルダーの範囲が明確になっているか
	目的についてステークホルダーと認識が共有されているか

ロープレでわかった解決方法

「目的・ゴール」を明確にする手順

- 初回の開発の内容を明確に
- その後の工程でどう機能が追加されていくかを
ユーザー側が理解し、
不安感を減少させるためベンダー側も真摯に説明
- 工程、特に、スプリントの移行（前の終わり、次の始まり）は何を持って
行い、その時に誰が何をするのかをプロジェクト内で共通認識を持つ

「目的の確認」に有効な方法

受託のアジャイルで使える方法

- ユーザーストーリーマッピング（機能一覧の洗い出し）
- インセプションデッキ（目的の共有・常態化）
- プロダクトバックログ、スプリントバックログの説明
- スプリントの移行をポイントとした工程表
- PM、PMO、スクラムマスタを明確にした体制図

研究成果：項目を方法に落とし込む

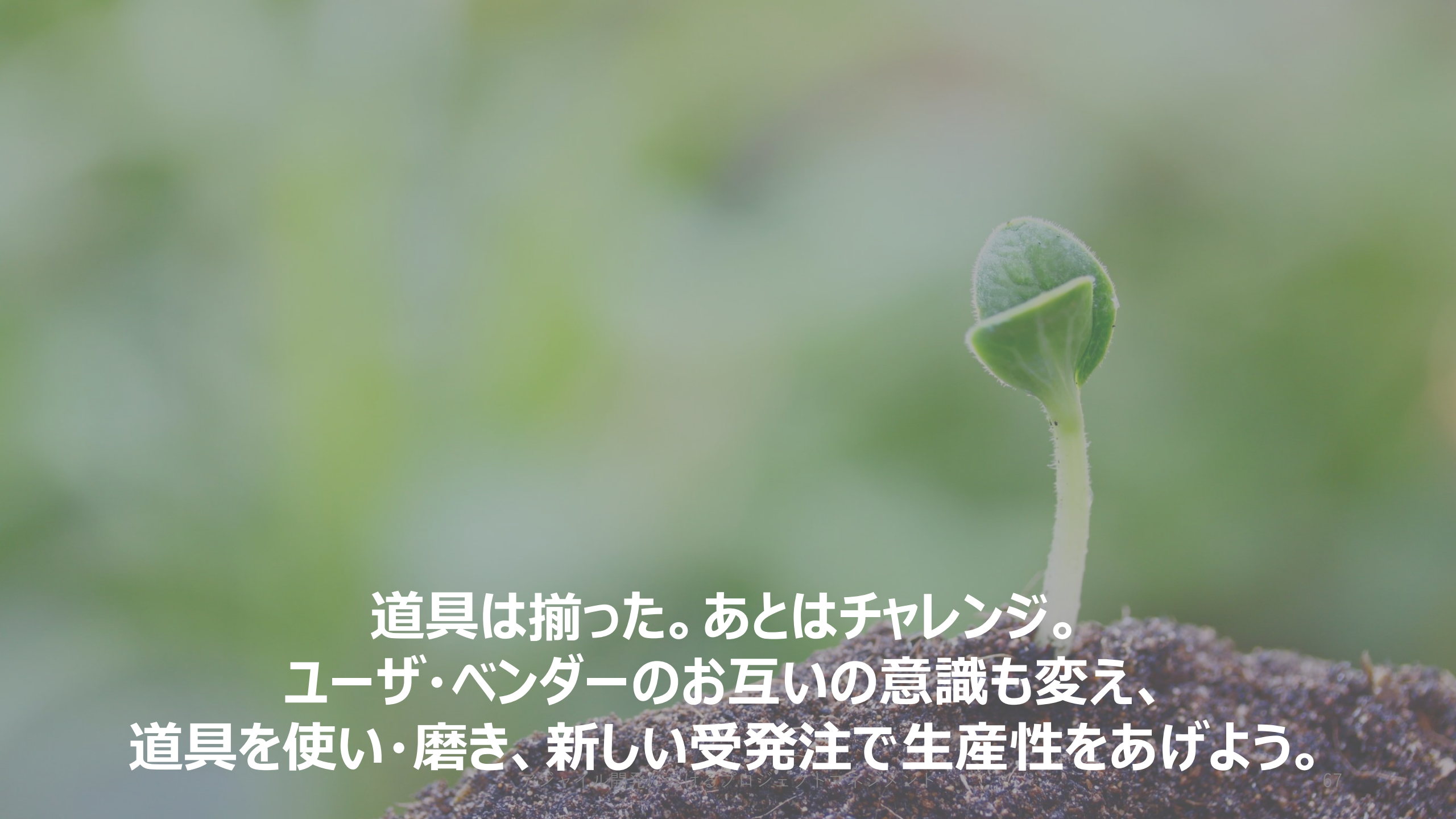
項目	チェック項目	有効な手段・フレームワーク
1. プロジェクトの目的・ゴール	<ul style="list-style-type: none"> プロジェクトの目的が明確であるか ステークホルダーの範囲が明確になっているか 目的についてステークホルダーと認識が共有されているか 	<ul style="list-style-type: none"> ユーザーストーリーマッピング（機能一覧の洗い出し） インセプションデッキ（目的の共有・常態化） プロダクトバックログ、スプリントバックログの説明 スプリントの移行をポイントとした工程表 PM、PMO、スクラムマスタを明確にした体制図
2. プロダクトのビジョン	<ul style="list-style-type: none"> 開発対象プロダクトのビジョン（あるべき姿・方向性）が明確であるか 開発対象プロダクトのビジョンについてステークホルダーと認識が共有されているか 	<ul style="list-style-type: none"> ユーザーストーリーマッピング（機能一覧の洗い出し） インセプションデッキ/エレベーターピッチ（目的の共有・常態化） 先々の見えないことの不安より、まずは上記をSprint #0として実施する

・
・
・



詳しくは報告書に別添のExcelを。
周りに配ってください、改変配布してください
common senseなopen knowledgeにしましょう

<https://www.ashisuto.co.jp/solution-studygroup/workgroups/reports/2022.html>



道具は揃った。あとはチャレンジ。
ユーザ・ベンダーのお互いの意識も変え、
道具を使い・磨き、新しい受発注で生産性をあげよう。

A woman with blonde hair tied up, wearing a white tank top and patterned leggings, is captured mid-jump with her arms raised in a gesture of joy. She is positioned on a concrete ledge overlooking a vast desert landscape. The landscape features a winding asphalt road that curves through the sand, leading towards a large, layered rock formation in the background. The sky is clear and blue. The overall scene conveys a sense of freedom and achievement.

ご清聴ありがとうございました。

私たちの研究から、次の価値が生まれますように。