

アジャイルソフトウェア開発 事例紹介

株式会社 テクノロジックアート
アジャイルソフトウェア開発グループ
東條 大介



アジェンダ

■ 事例紹介

- プロセスの選定
- プロセスの開始
- ふりかえり

■ 最後に

事例紹介 ～プロセスの選定

**機密保持のため、当該スライドは
カットさせていただきます。**

採用プロセス・技術の検討

■ 前提

- ドメイン知識を持つコンサルタントが参画
- オブジェクト指向のスキルを持つエンジニアを4～5人準備可能

■ 課題

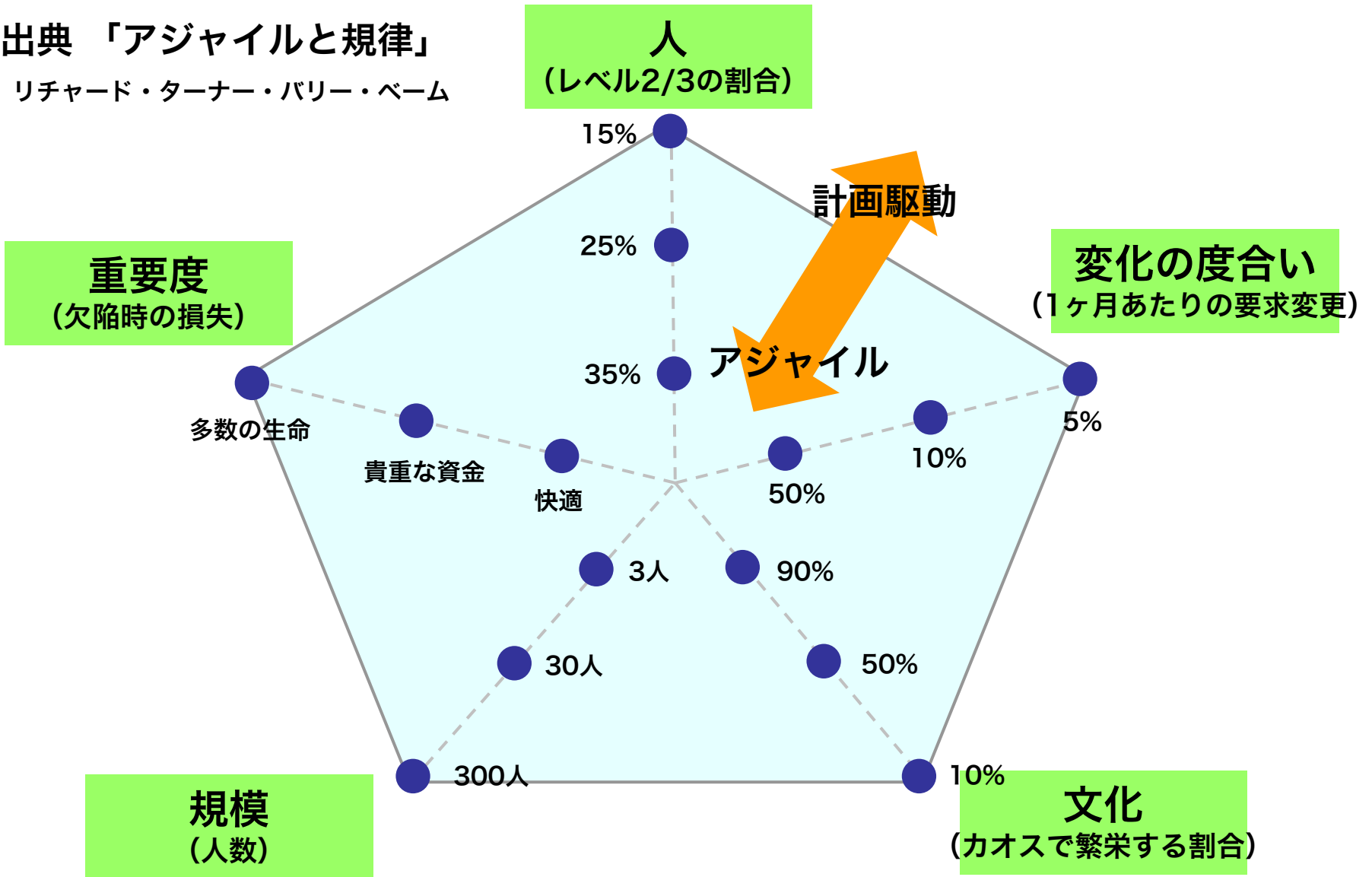
- サービス時間帯（日中）の稼働停止は許されない
- アプリケーションの仕様はまだ定まっていない
- ユーザは数百人超
- データの重要度は極めて高い、ミッションクリティカル
- 規模に対して納期が短い

■ 採用技術の検討

- アーキテクチャに拘束はない

採用プロセスの検討

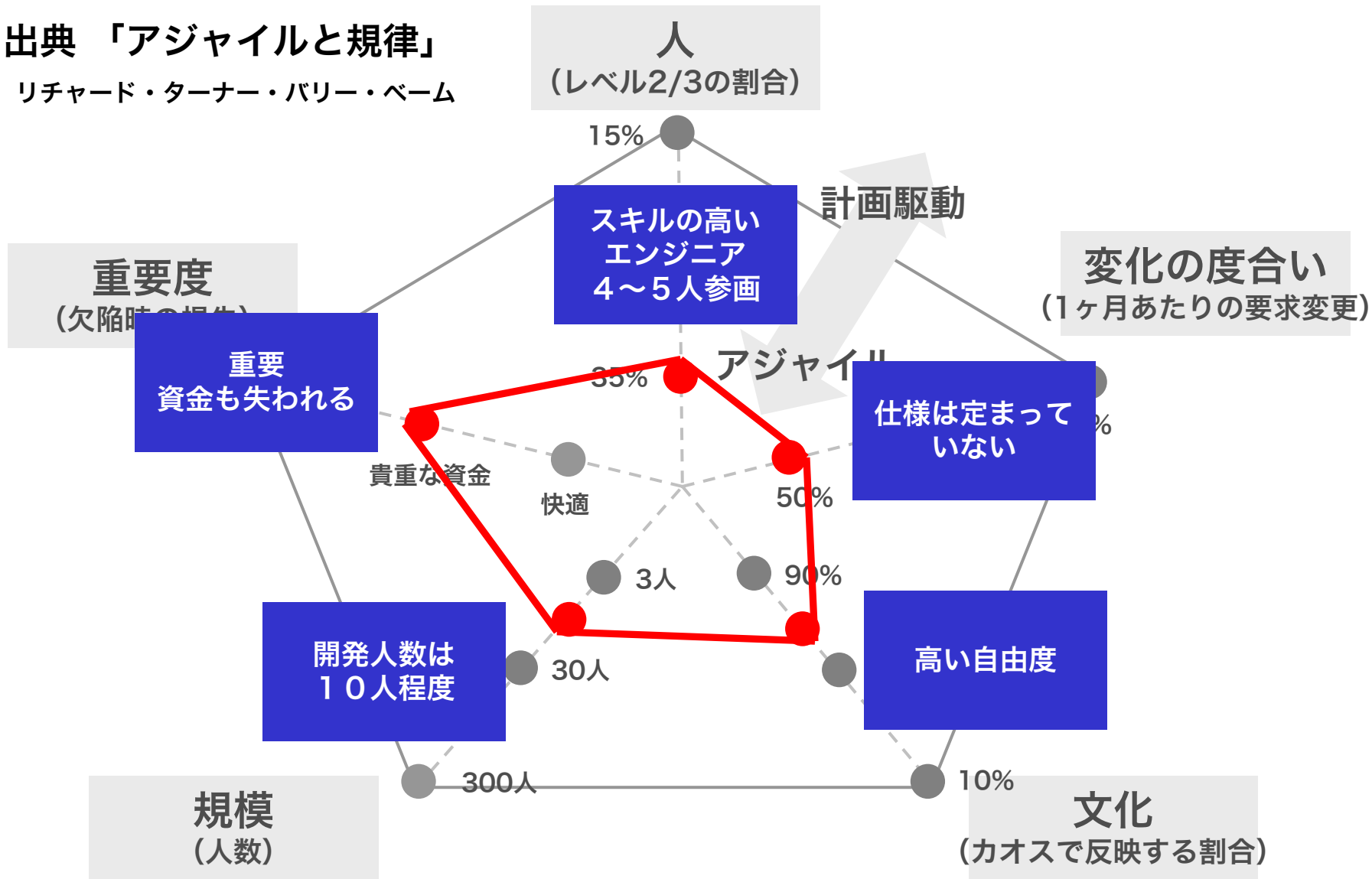
出典 「アジャイルと規律」
リチャード・ターナー・バリー・ベーム



採用プロセスの検討

出典 「アジャイルと規律」

リチャード・ターナー・バリー・ベーム



採用プロセスの検討



重要
資金が失われる



開発人数は
10人程度



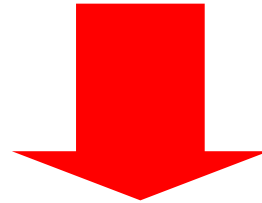
スキルの高い
エンジニア
4～5人参画



仕様は定まって
いない



高い自由度



プロジェクトは計画駆動開発より
アジャイル開発向きと判断

採用技術の検討

Perl/CGI

PHP

Ruby

.NET

Java

新規システムの立ち上げにあたって、今さら感あり
スクリプト言語なのでスケーラビリティに難あり

オブジェクト指向
スクリプト言語なのでスケーラビリティに難あり

.NETに明るいエンジニアが当時不在
知識習得のオーバーヘッドに時間がかかる
Windowsプラットフォームのスケーラビリティ

ミッションクリティカルなシステムで実績多数
最有力候補、無難な選択

工程の整理

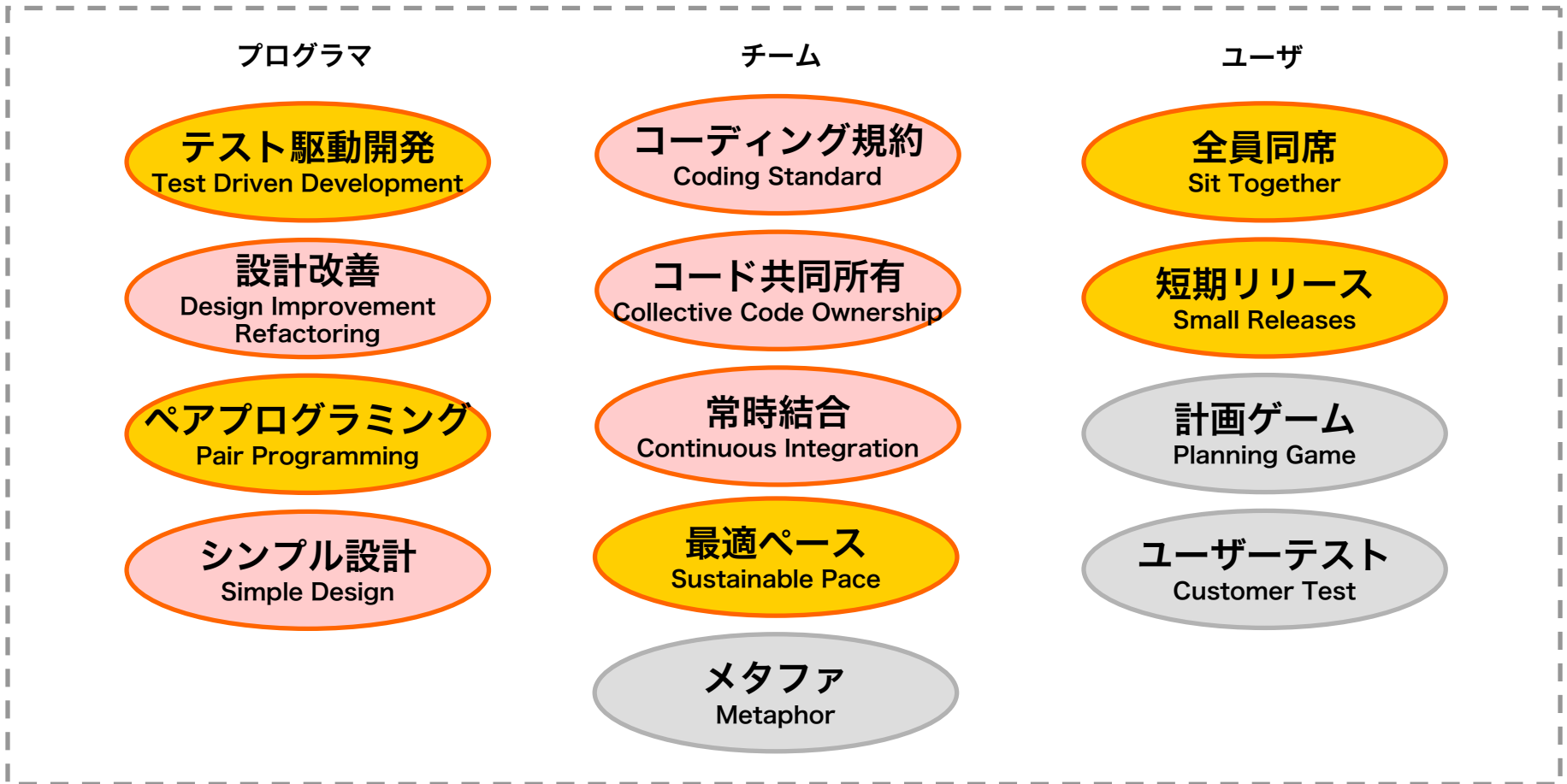


- 開発の前段として以下の成果物を定義
 - ワークフロー (UML/アクティビティ図)
 - データモデル (UML/クラス図)
 - システム化機能一覧
 - 画面紙芝居 (HTML)
 - システムアーキテクチャ構成、アプリケーションアーキテクチャ構成
- 要件定義後の開発～テストフェーズにプラクティスを適用
 - 適用可能な範囲で、適用可能なプラクティスを使う

事例紹介 ～プロセスの開始

**機密保持のため、当該スライドは
カットさせていただきます。**

■ 以下のプラクティスの適用を検討



常時利用

時々利用

ツールの準備

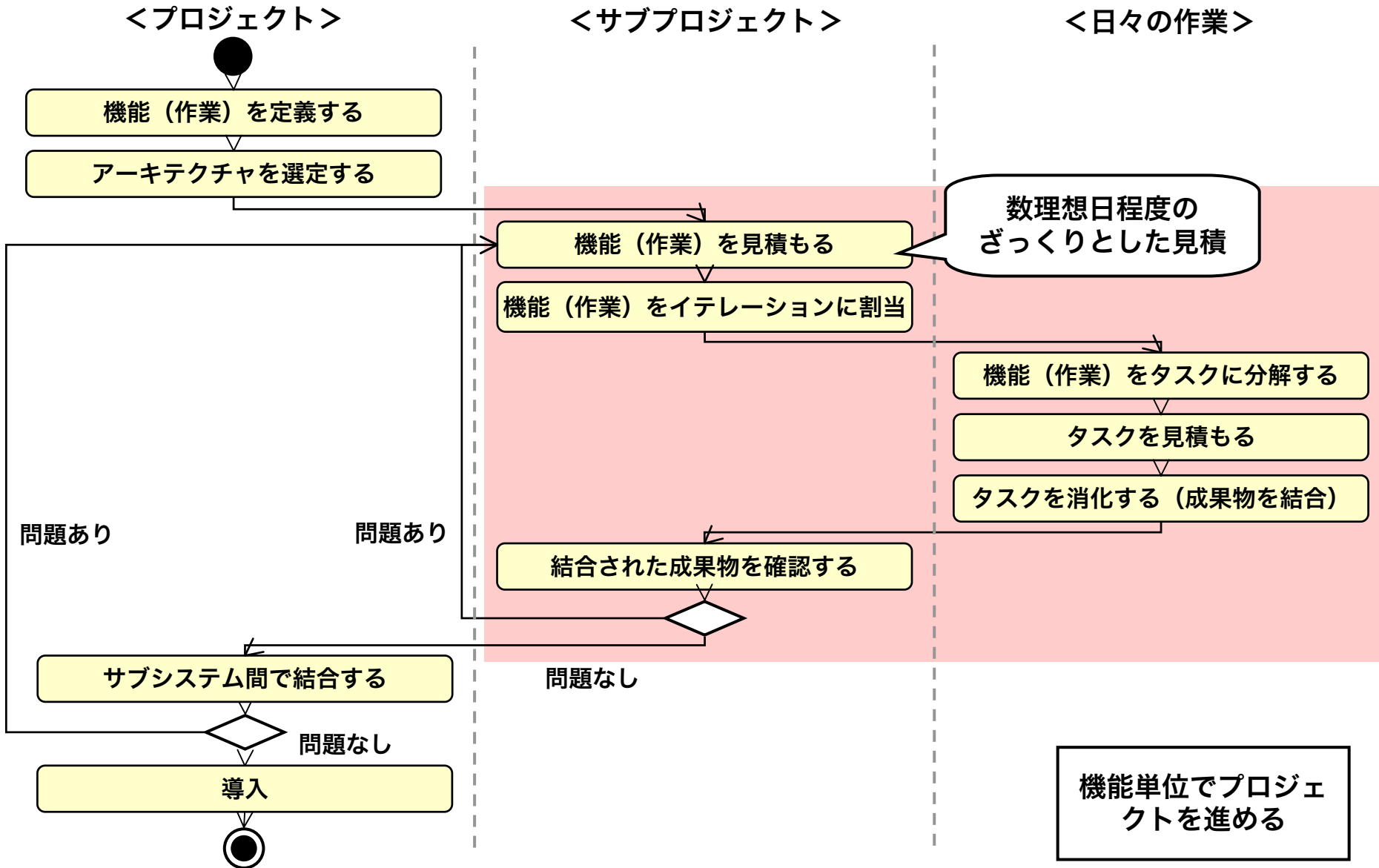
■ 開発環境の整備

- 構成管理 / Subversion
 - SQLスクリプトも管理
- 結合マシンと自動ビルド / LinuxサーバーとAnt
 - ウェブとバッチでそれぞれ準備
- タスク管理 / 影舞
 - イテレーティブな開発では随時不具合が発生する
- 統合開発環境 / Eclipse
 - リファクタリング支援機能を活用する
 - Junitを使用してTDDを実践する

■ その他

- ホワイトボード
- 開発に関わる人はロールに関係なく1つのオープンな部屋に

プロセスの進行



プロセス進行上の問題点

■ 仕様の伝達

- 人数が多いため、仕様の伝達が上手く行かない（口頭では厳しい）
- 特にPGへの仕様の伝達にドキュメントが必要
- 上手く行かない場合はプラクティスに拘らず、柔軟な対応を！

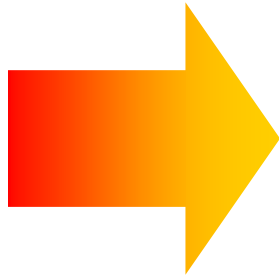
■ 進捗の可視化

- サブプロジェクト間の進捗調整が難しい
- 長期的な予測が立てづらい
- サブプロジェクト内でのスタンドアップミーティングの他に、サブプロジェクト間での打合せが必要

事例紹介 ～ふりかえり

結果

- 全ての機能要件を満たす
- アーキテクチャが整っている
- 運用開始後もさしたる不具合がない
- 納期に対して余裕を持ってリリース
- 予算超過がない
- 厳しい残業がない（退職者、体調不良者ゼロ）
- 必要最低限のドキュメントが整っている



当たり前のこと、でも難しいことを
アジャイルのエッセンスで実現出来た

プロセス進行上のポイント

■ 要件定義

- 運用設計やデータモデル、システム化機能の整理を予めしておく

■ 言語やアーキテクチャ

- 極力使い慣れた言語、フレームワーク、ライブラリを使う
- レイヤリングする

■ 立ち上がり

- プロジェクトの進め方に試行錯誤や戸惑いがある

■ 品質の確保

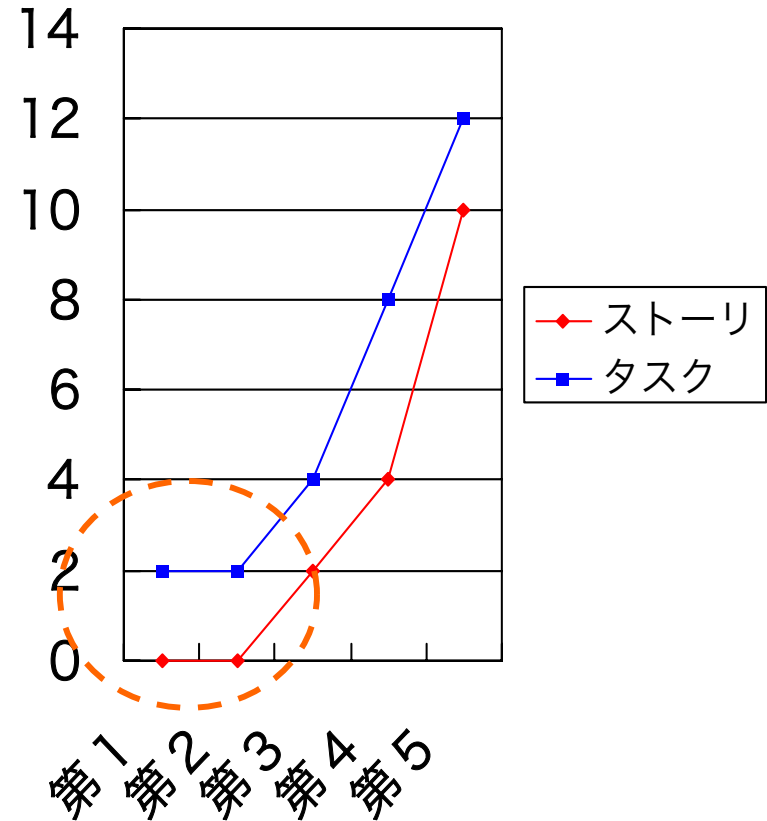
- TDDは部分的に適用 → ユーティリティクラスなど
- 適用が難しい部分は従来型のテストを実施し、品質を確保

プロセス進行上のポイント

プロジェクトの立ち上がりは、どうしてもオーバーヘッドが発生！

- 言語は慣れたもの！
- フレームワークは慣れたもの！
- 要件定義は事前に
- イテレーティブ開発経験者を投入！

過去の実績

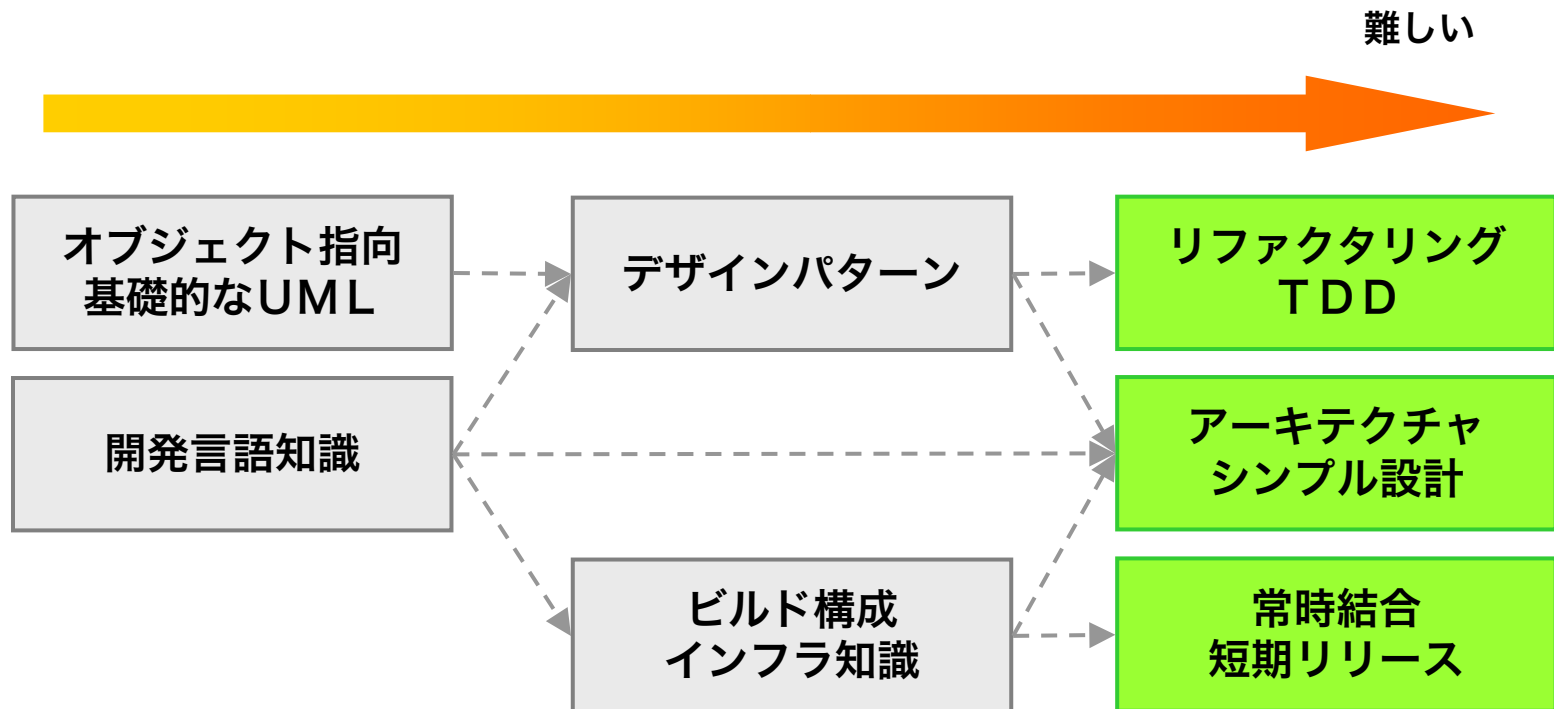


最後に

アジャイルで使うテクニック

■ プラクティスの実践のために

- チームにスキルのあるエンジニアを投入する事は欠かせません。
- トレーニングや開発の実践の中で、以下のようなスキルを身につけていく必要があります。



■ **ご静聴、ありがとうございました。**