

派生開発カンファレンス2016

# Agile型開発へのPFD適用と XDDPによる連携開発

派生開発推進協議会 T6研究会

星野充史

2016/5/27

## T6研究会の活動

### AgileとXDDPの連携を研究

目的：

AgileとXDDPの双方の特徴を組み合わせて、それぞれの課題を解決する。

活動成果：

派生開発カンファレンス2015 ポスター展示

「機能追加のAgile XDDP」 「変更のAgile XDDP」

活動メンバ：

永田 敦

ソニー株式会社

斎藤 賢一

株式会社エクスマーション

八木 将計

株式会社日立製作所

山田 謙輔

株式会社日立情報通信エンジニアリング

佐藤 隆雄

アルパイン株式会社

加藤 康之

アンリツエンジニアリング株式会社

# 発表の流れ

## 1. はじめに

- ✓ 事例の概要

## 2. 背景と課題

- ✓ システム開発の概要
- ✓ 開発の特長
- ✓ 課題

## 3. 施策

- ✓ AgileとXDDPの連携

## 4. 結果

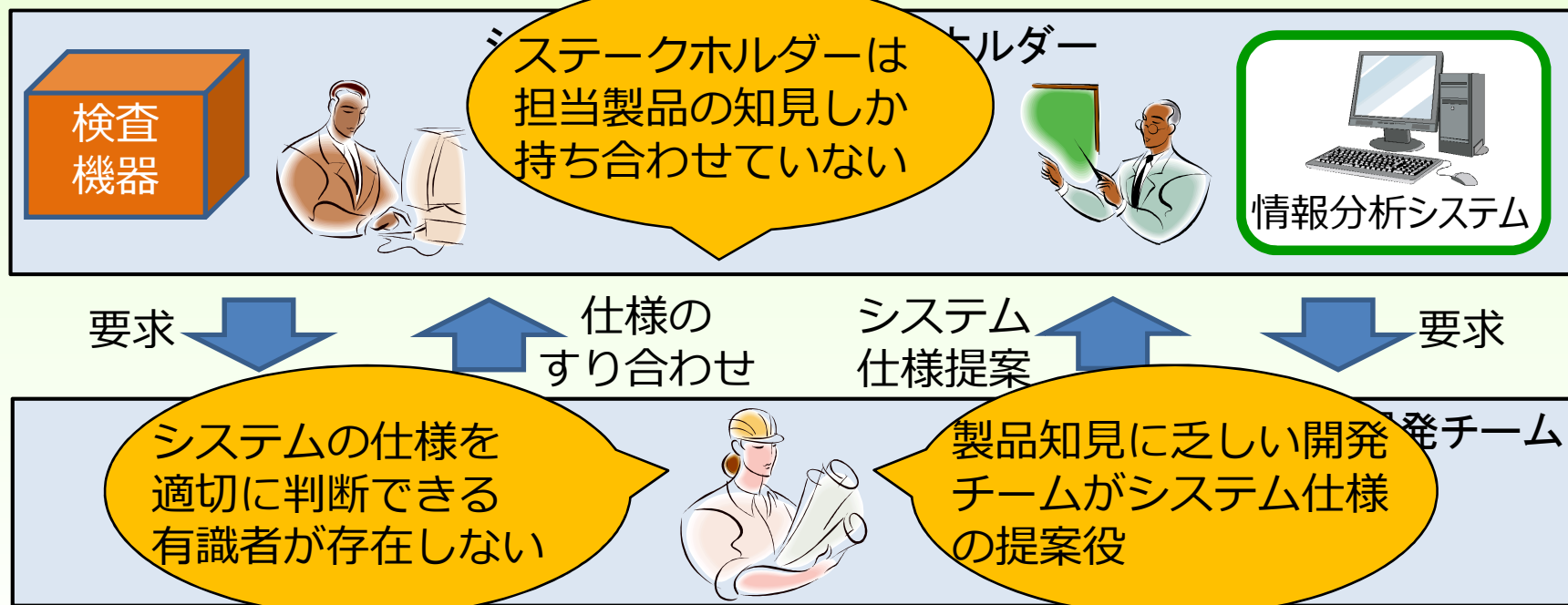
## 5. まとめ

# はじめに

## 事例の概要

複数の機器から構成される**システム**の派生開発

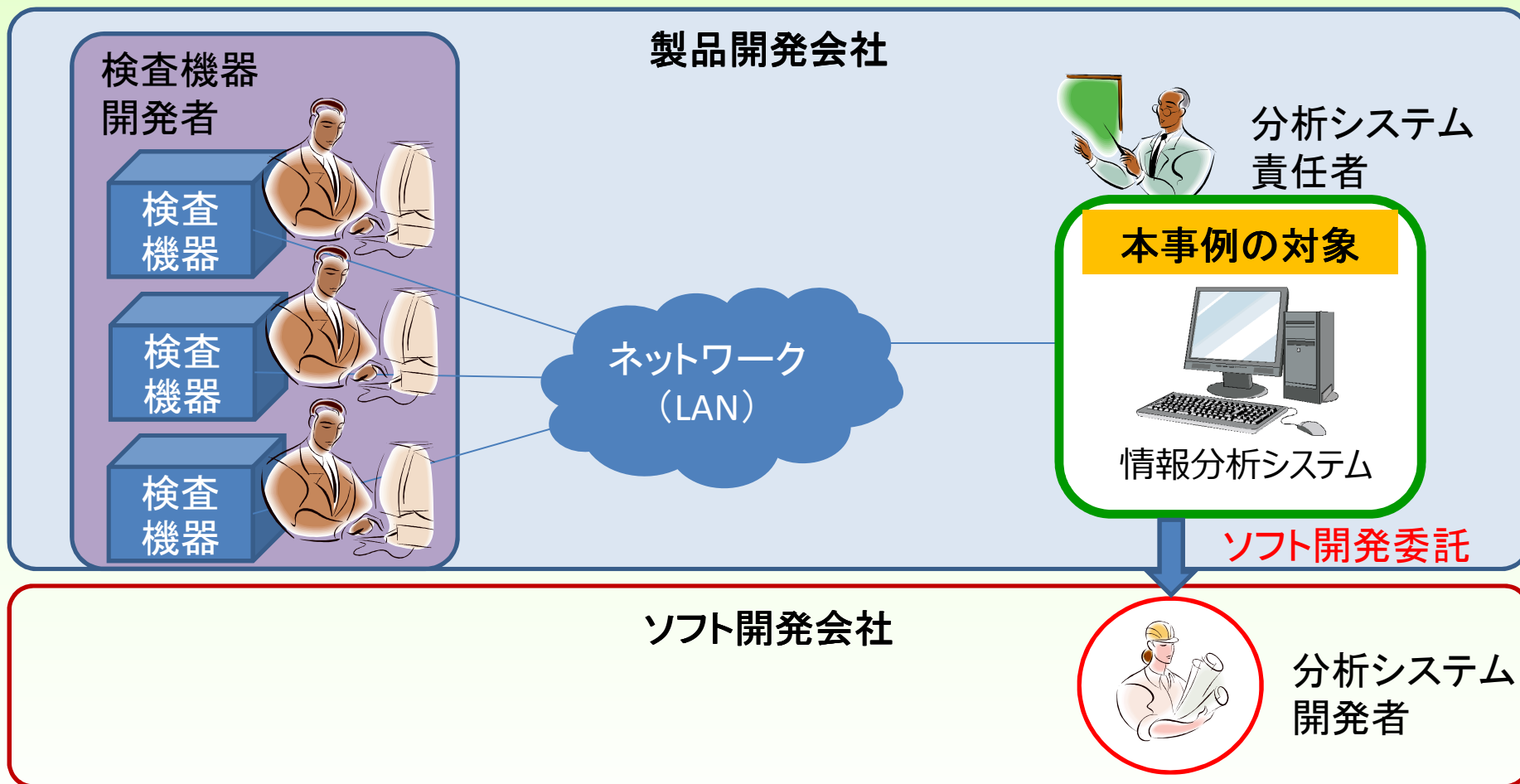
- ✓ 新規に立ち上げた開発チーム
- ✓ 製品ドメインの異なる**複数のステークホルダー**の要求
- ✓ AgileとXDDPを連携して開発



# 背景と課題

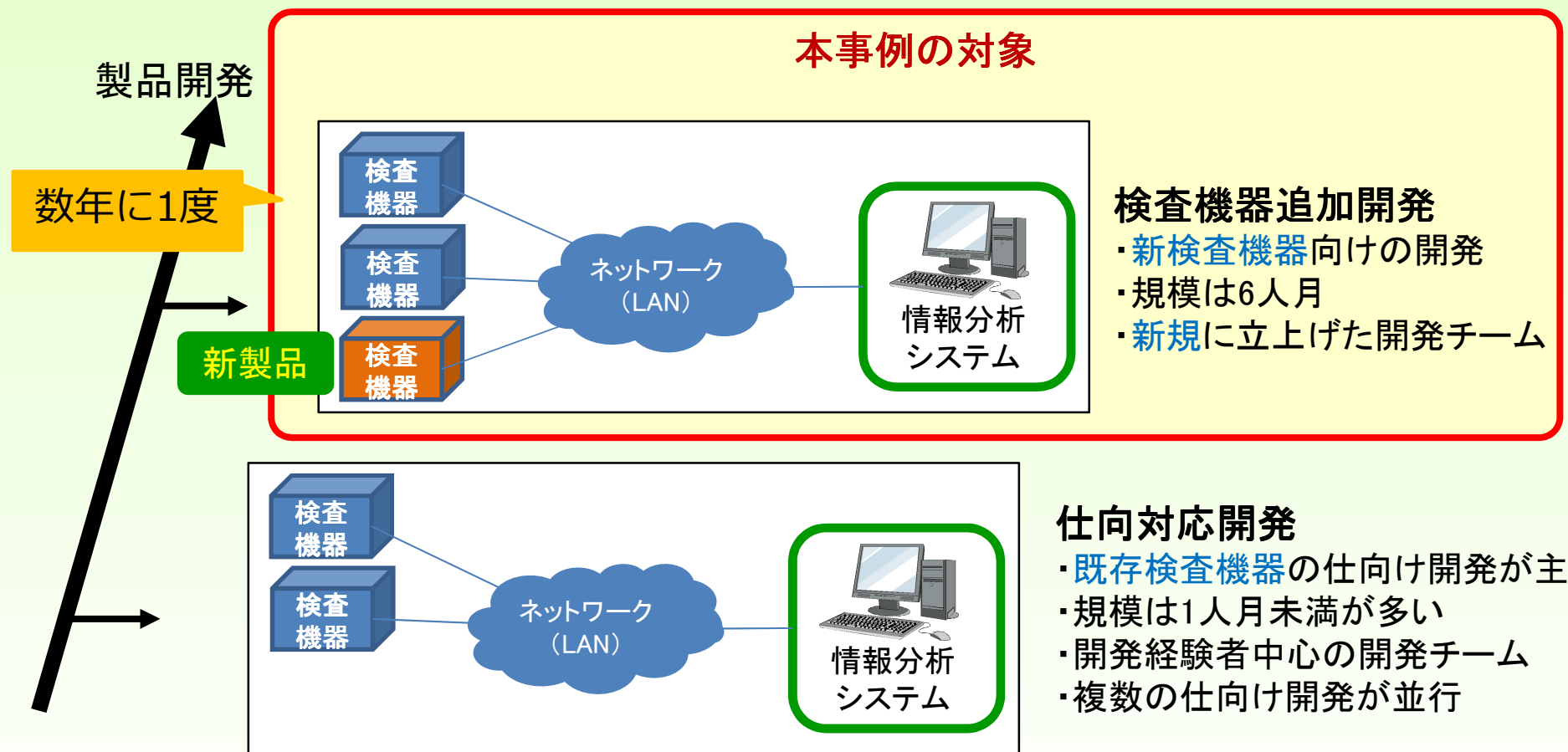
## 背景：システム開発の概要

### 開発会社と組織・システム機器の構成



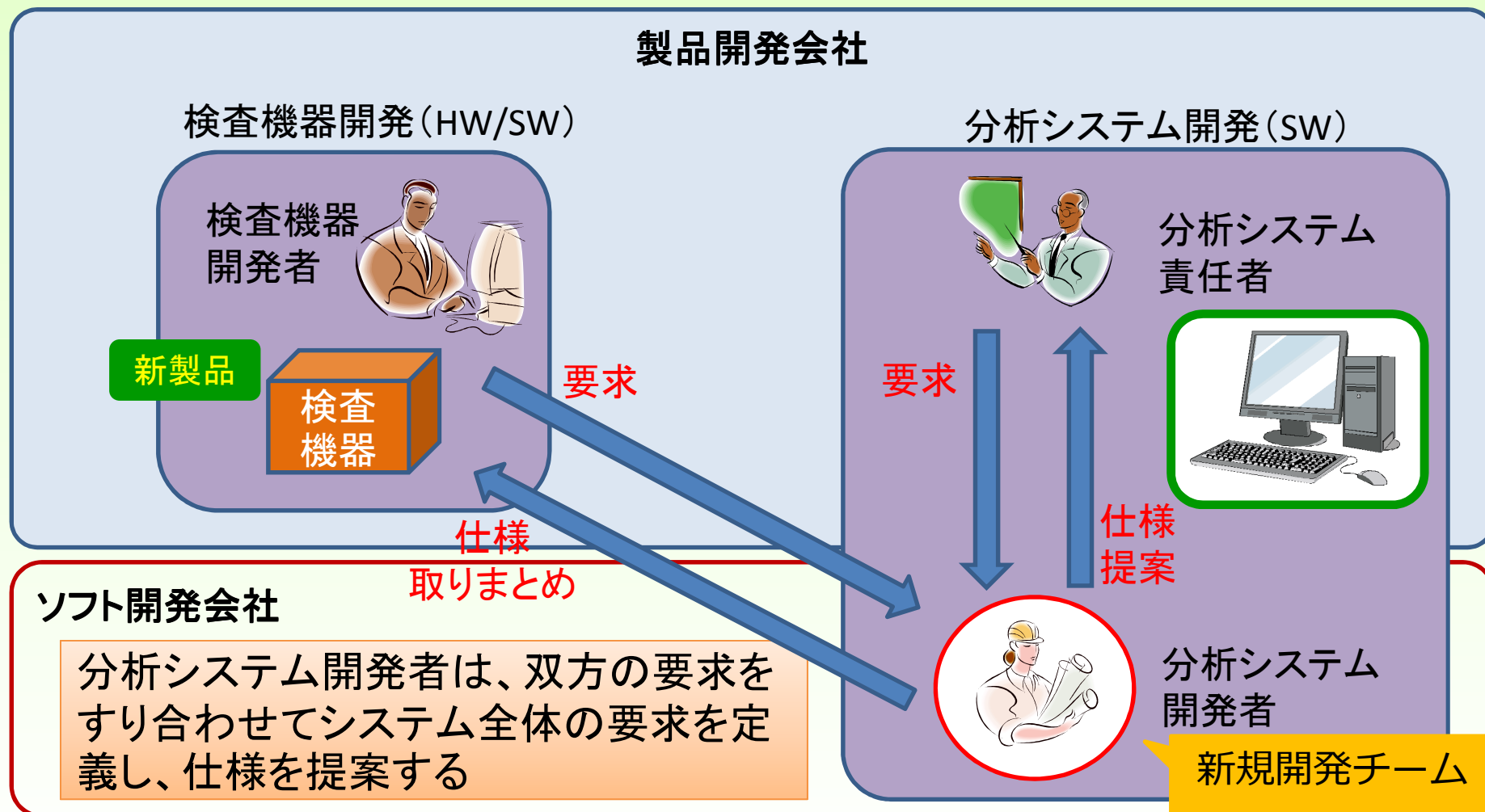
## 背景：システム開発の概要

### 製品開発のライフサイクルと本事例の対象



## 背景：システム開発の概要

### 検査機器追加開発の要件定義





## 背景：開発の特徴

### 関係者の製品知見

関係者		分析システム	新検査機器
	分析システム 責任者	○	△
	分析システム 開発者	△	○
	新検査機器 開発者	△	○

製品知見が一致しないため  
仕様認識にズレが生じやすい

#### ◆製品知見の不一致

- ・ 関係者の知見は個々の製品に依存している
- ・ 分析システムの開発者は新検査機器の知見を有する
- ・ システム全体の知見を持つ有識者は不在

## 背景：開発の特徴

### 分析システム開発チームの体制と役割

体制	開発リーダー	開発担当者
新規開発チーム 4名	未経験者	開発経験者と 未経験者で構成

#### ◆開発体制の制約

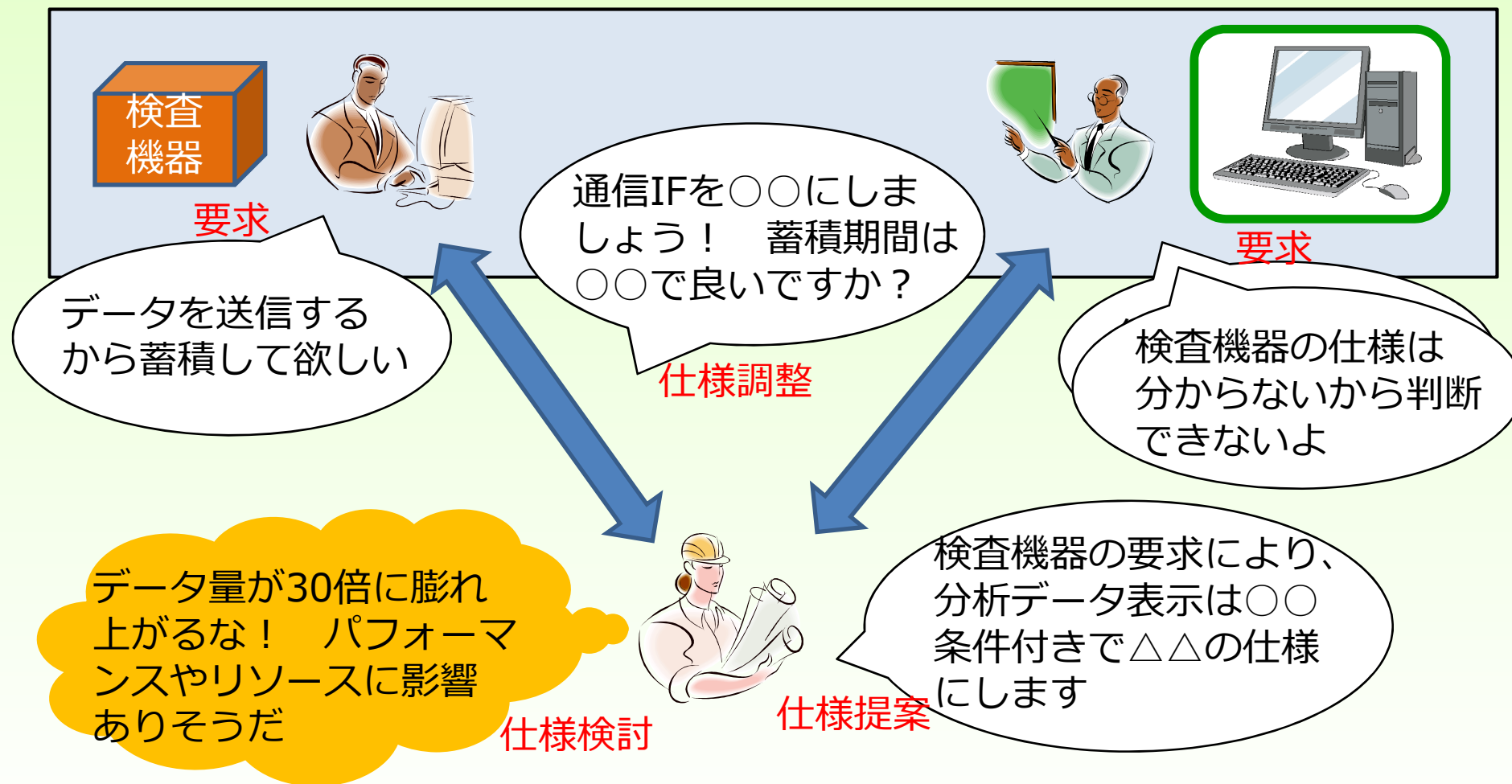
- ・ 開発経験者は複数の案件を掛け持ち（稼働が低い）
- ・ 開発者不足のため未経験者を段階的に投入

#### ◆システム開発における開発チームの役割

- ・ 新検査機器を含むシステムの仕様を取りまとめ
- ・ 分析システムを開発してシステムテストを実施

## 背景：開発の特徴

### 要求から仕様を取りまとめる現状の構図



## 問題

### 検査機器追加開発の問題

#### 関係者の仕様に関する認識が異なる

- システムの仕様理解は双方の担当製品の知見に影響
  - ✓ **仕様漏れ**や**仕様解釈ミス**に気付きにくい
- 仕様提案を関係者で適切にレビューできない部分あり
  - ✓ 書面では**具体的な仕様を共有しにくい**

重要な要求  
に存在

#### 製品知見と開発体制に制約がある

- 開発経験者のノウハウに頼る部分が多い
  - ✓ 未経験者の**製品知見の不足が品質に影響**しやすい
- 開発案件増加に対して体制構築が追いついていない
  - ✓ 未経験者の段階的な投入は立上がりに時間が掛かる

# 課題

## 検査機器追加開発の課題

### 問題

#### 関係者の仕様認識が異なる

- ✓ 仕様漏れや仕様解釈ミスに気付きにくい
- ✓ 書面では具体的な仕様を共有しにくい

重要な要求に存在

#### 製品知見と開発体制に制約がある

- ✓ 未経験者の製品知見の不足が品質に影響しやすい
- ✓ 未経験者の段階的な投入は立上げりに時間が掛かる

### 課題

製品として重要な要求に対する関係者の仕様認識を動作確認により早期に一致させ、手戻りのリスクを低減する

未経験者の製品知見不足による不具合を防止する

段階的に投入する開発者の作業が効率よく進むように準備する

# 施策

# 施策

## 課題

製品として重要な要求に対する関係者の仕様認識を、動作確認により早期に一致させ、手戻りのリスクを低減する

未経験者の製品知見不足による不具合を防止する

段階的に投入する開発者の作業が効率よく進むように準備する



## 施策

### Agile型開発(Scrum)を適用

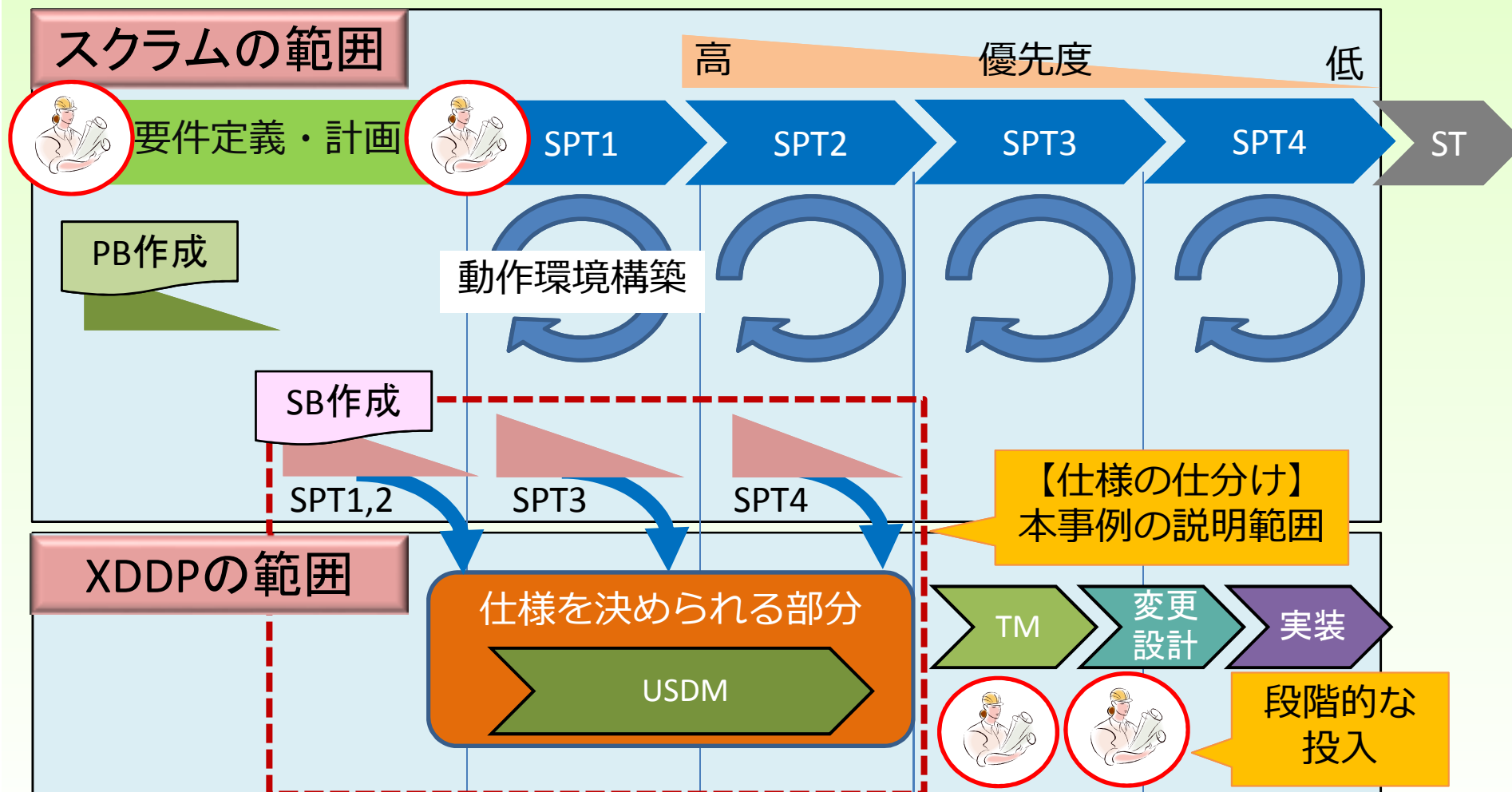
- 関係者の中で**仕様の認識が異なる部分**に適用
- PFDで**適切な作業タスク**を定義
- **実際の動作**で仕様の認識を早期に確認して手戻り低減

### XDDPを適用

- Agile開発の過程で**仕様を特定できる部分**を抽出し、その部分に**XDDPを適用**
- 段階的に投入する未経験者へ**変更要求仕様**を提供し、**投入時の立ち上げを促進**

# 施策：AgileとXDDPの連携

## 開発プロセスの俯瞰図





## 施策：AgileとXDDPの連携

### 仕様の仕分け

スプリントバックログの作成過程で仕様を仕分ける

#### 適切な作業タスクの抽出

- 作業タスクが漏れない
- 作業タスクに無駄がない

Agile

#### 仕様を決められる部分

XDDP

- 既存仕様に対する変更内容が明確である
- 関係者が仕様をレビューできる

変更要求  
として

USDMを作成

#### 仕様を決めにくい部分

Agile

- 関係者の仕様に関する認識が異なる

作業タスク  
として

スプリントバックログへ登録

# 施策：AgileとXDDPの連携

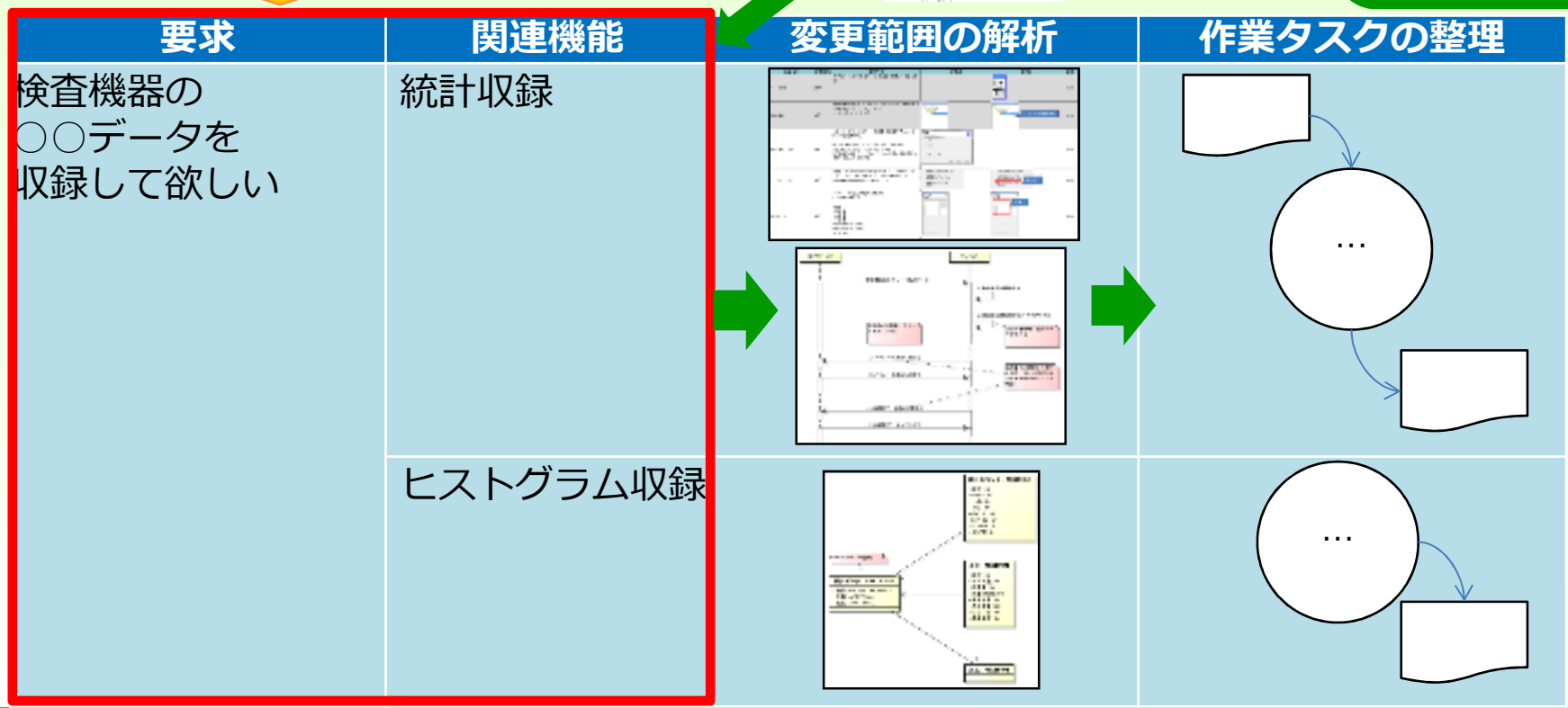
## 適切な作業タスクの抽出①

要求と作業タスクを関連付ける

検査機器の要求と  
分析システムの機能を紐付け

機能一覧

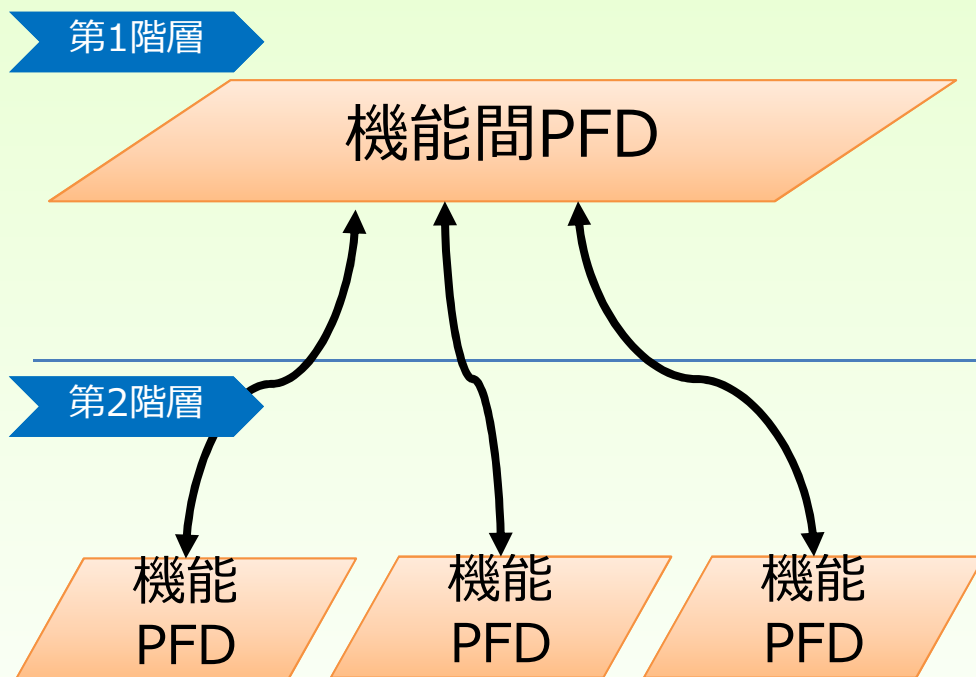
操作  
マニュアル



## 施策：AgileとXDDPの連携

### 適切な作業タスクの抽出②

作業タスクの整理  
PFDを2階層に分けて表現



#### 第1階層

- 機能間における成果物との関係を表現する

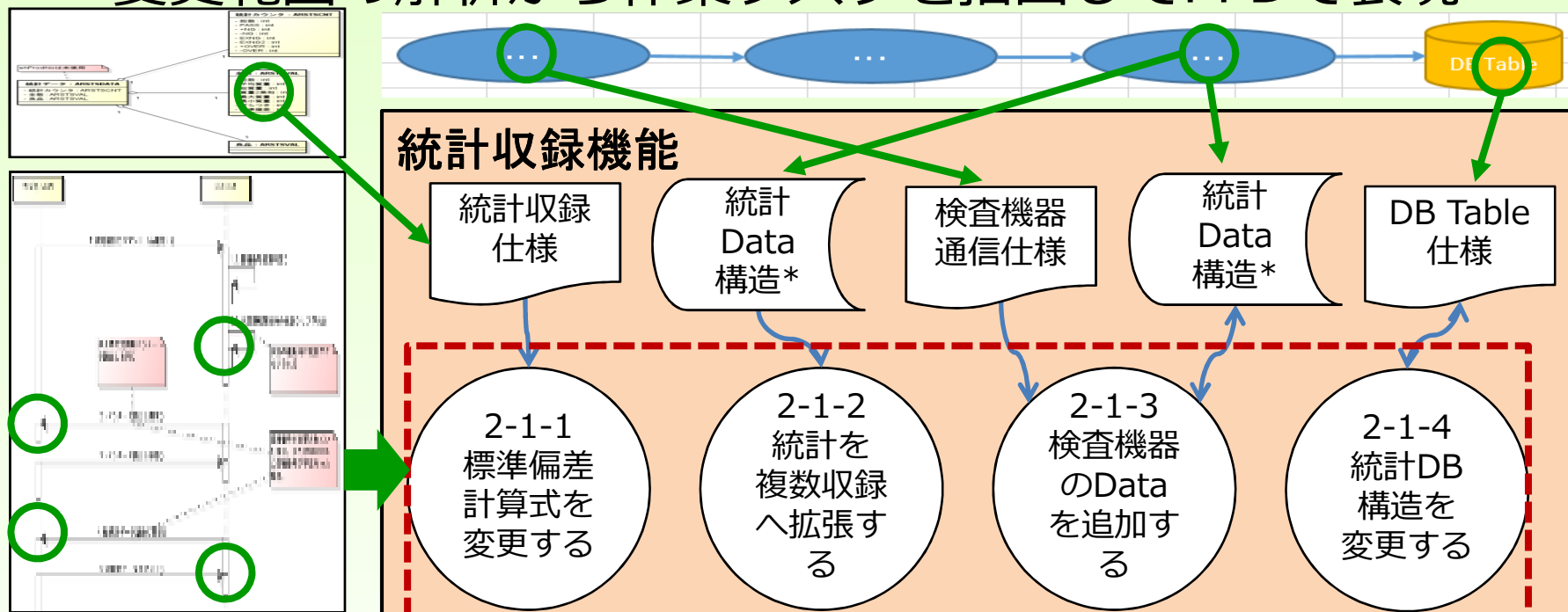
#### 第2階層

- 機能を実現するための作業タスクと成果物との関係を表現する
- 機能毎にPFDを作成する

# 施策：AgileとXDDPの連携

## 適切な作業タスクの抽出③

変更範囲の解析から作業タスクを抽出してPFDで表現



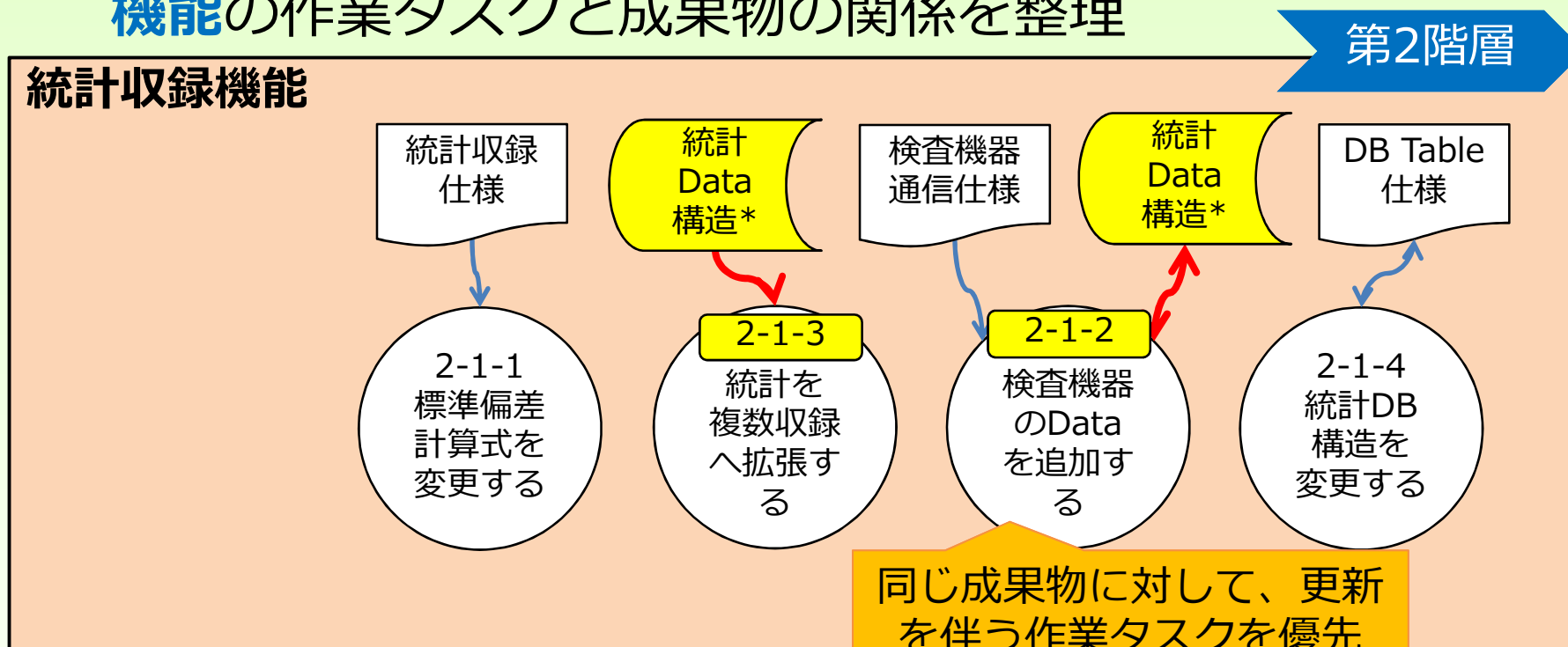
### 作業タスクと成果物

- 機能の振る舞いから作業タスクを抽出
- 作業タスクに関連する成果物を定義して紐付け

# 施策：AgileとXDDPの連携

## 適切な作業タスクの抽出④

機能の作業タスクと成果物の関係を整理



### 作業タスクの整理

- 同じ成果物に関連する作業タスクは統合・分離・優先順位を検討

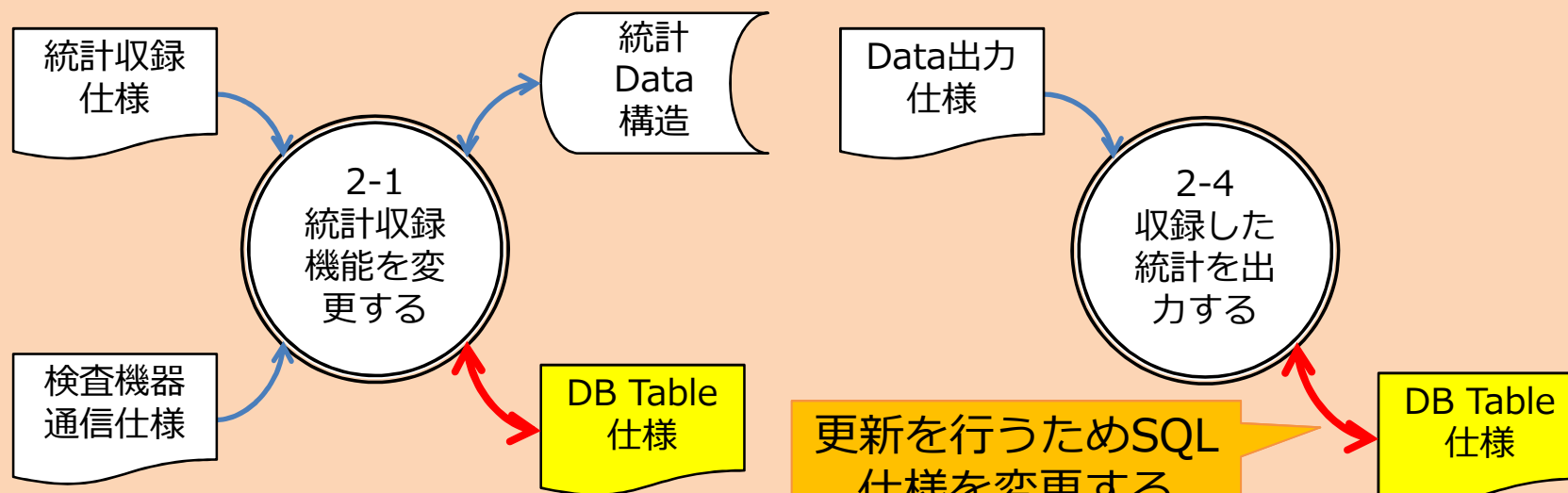
# 施策：AgileとXDDPの連携

## 適切な作業タスクの抽出⑤

機能間で成果物の関係を整理

第1階層

### 機能タスクの階層



### 作業タスクの整理

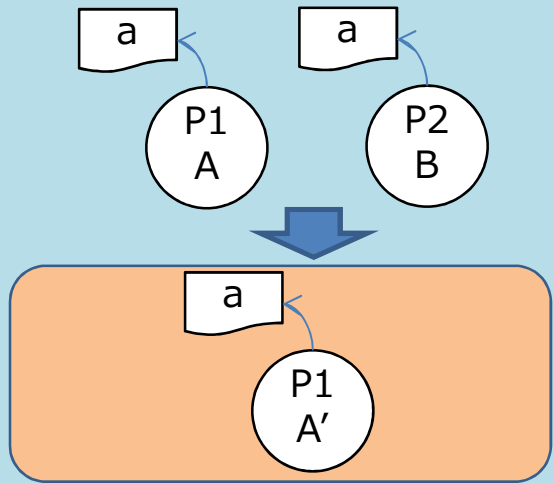
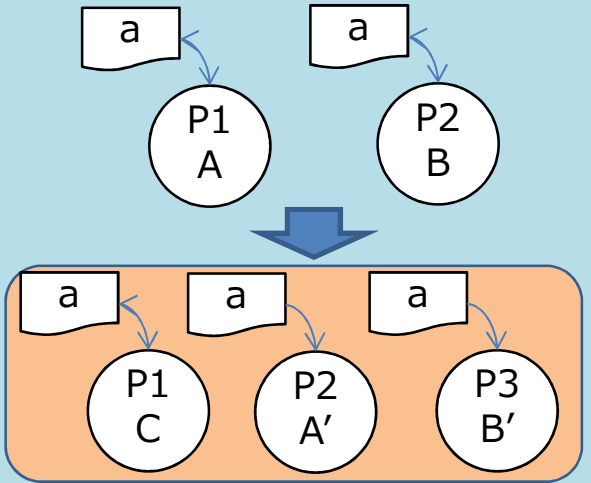
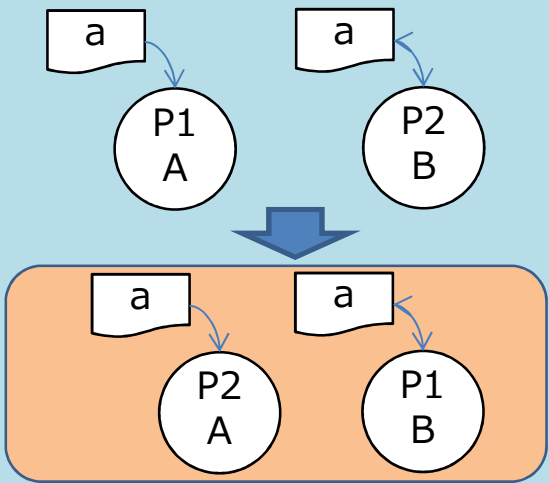
- 各機能と成果物の関連を表現して同様に検討
- 作業タスクの優先順位、および成果物の関連をシミュレーション

# 施策：AgileとXDDPの連携

## 適切な作業タスクの抽出⑥

作業タスクの統合・分離・優先順位調整のパターン

⇒成果物を更新する作業タスクに着目

統合	分離	優先順位調整
<p>更新を伴う作業タスクが複数ある場合、それらをまとめるかどうかを検討する</p>	<p>参照と更新の両方を行う作業タスクが複数ある場合、更新をひとつの作業タスクにまとめるか、または更新手順を検討する</p>	<p>参照する作業タスクと更新する作業タスクがある場合、作業タスクの優先度調整を検討する</p>
		

## 施策：AgileとXDDPの連携

### 仕様を決められる部分の抽出

PFDと解析結果から、条件に一致する部分を抽出

#### 仕様を決められる条件

- 既存仕様に対する変更内容が明確である
- 関係者が仕様をレビューできる



#### 制約事項

- Agileの開発範囲と独立して開発可能であること  
⇒AgileとXDDPの**開発プロセスの違いに影響**しない



USDMを作成

作業タスクを「仕様」  
で表現し、理由付けを  
行い、関係者でレビュー



## 施策：AgileとXDDPの連携

### 仕様を決めにくい部分の抽出

PFDと解析結果から、条件に一致する部分を抽出

#### 仕様を決めにくい条件

- 関係者の仕様に関する認識が異なる



#### 「仕様を決められる部分」で条件や制約を満たさない部分

- 関係者の製品知見の範囲が一致せずレビューが困難
- Agileの開発範囲のプログラムと結合度が高い



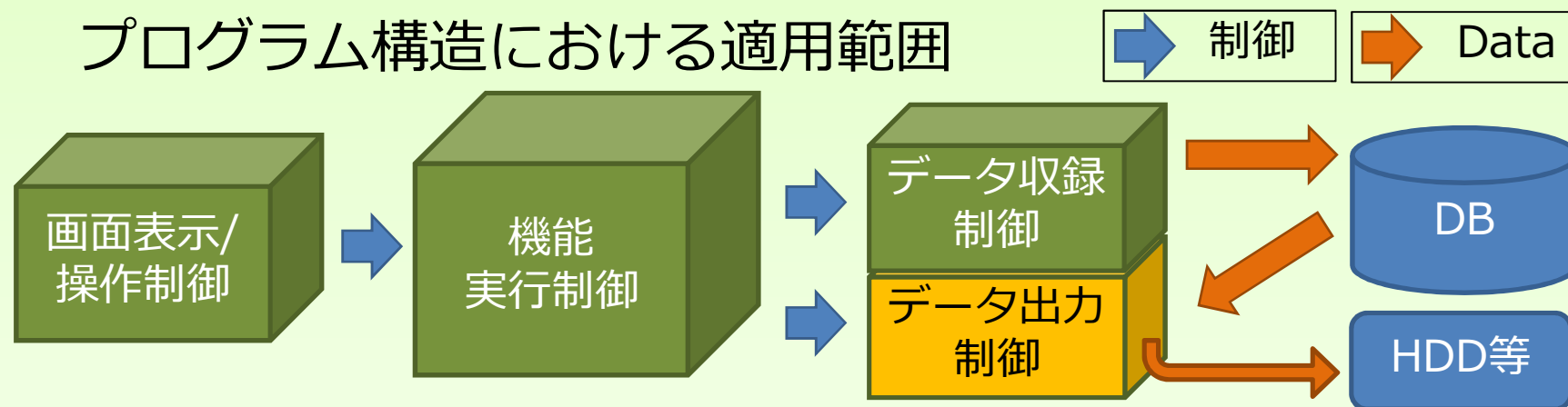
スプリントバックログへ登録

作業タスクを  
各SBへ登録

## 施策：AgileとXDDPの連携

### XDDP適用範囲の具体例

プログラム構造における適用範囲



#### 「データ出力制御」にXDDPを適用

- データ出力仕様はレビュー可能
  - ✓ 検査機器の要求に影響しない
  - ✓ 分析システムの関係者で仕様を決定できる
- Agileの適用範囲と独立して開発可能
  - ✓ データベースを境にしてDataフローを分離できる
  - ✓ データ参照先のDBは要件定義にて仕様決定済み

XDDP経験者がサポート

# 結果

# 結果

## 課題

製品として重要な要求に対する関係者の仕様認識を、動作確認により早期に一致させ、手戻りのリスクを低減する

未経験者の製品知見不足による不具合を防止する

段階的に投入する開発者の作業が効率よく進むように準備する

## Agile型開発へPFDを適用した効果

- 作業タスクの無駄を排除し開発効率が向上
  - ✓ 作業タスクの**衝突（重複・待合わせ）を低減**
  - ✓ 設計過程で成果物の変化による影響が把握でき、**手戻りを低減**
  - ✓ シミュレーションにより作業タスクが確認でき、**モレを防止**

## XDDPを適用した効果

- 担当範囲の変更要求仕様を準備
  - ✓ 仕様理解が促進して**変更設計期間を守れた**
- 変更設計はBefore/Afterで表現
  - ✓ 担当者とレビュアーの仕様に対する**変更範囲の理解が一致**して設計漏れを防止できた
- 高い生産性と品質
  - ✓ プログラミングは **70LOC/H** を達成できた
  - ✓ **仕様漏れ不具合は1件のみ（過去実績3~8件）**
  - ✓ 担当者の開発期間を**20%短縮**できた

# まとめ

## まとめ

### 本事例のポイント

- 数年に1度の既存システムへの検査機器追加開発
  - ✓ ステークホルダーの**仕様認識にズレ**
  - ✓ 新規に立上げた**未経験者中心の開発チーム**
  - ✓ システムの仕様を適切に判断できる**有識者が不在**
  - ✓ **製品知見が乏しい開発者**が仕様の取りまとめ役
- そこで、AgileとXDDPを連携させて開発
  - ✓ XDDPの適用範囲を抽出するため**仕様の仕分け**を実施
  - ✓ **PFDにより適切に作業タスクを整理**して開発効率向上
- そして、課題を解決
  - ✓ 各課題は**AgileとXDDPの特徴**を使い分けて解決
  - ✓ 開発体制の問題も施策が有効に働いて計画通り遂行

## まとめ

### 本事例の適用を想定する開発

- 未経験製品の派生開発でステークホルダーとの仕様の認識にズレがある開発
- 派生開発にAgileを適用している開発
  - ✓ 開発者の入れ替えがあり製品知見が蓄積しにくい
  - ✓ 仕様漏れや設計漏れが多く発生している
- 長年の派生開発でソフトウェア構造が複雑化している開発
  - ✓ 作業タスクの衝突（重複・待合わせ）が多く開発効率が低下している

ご清聴ありがとうございました