変更のAgileXDDP

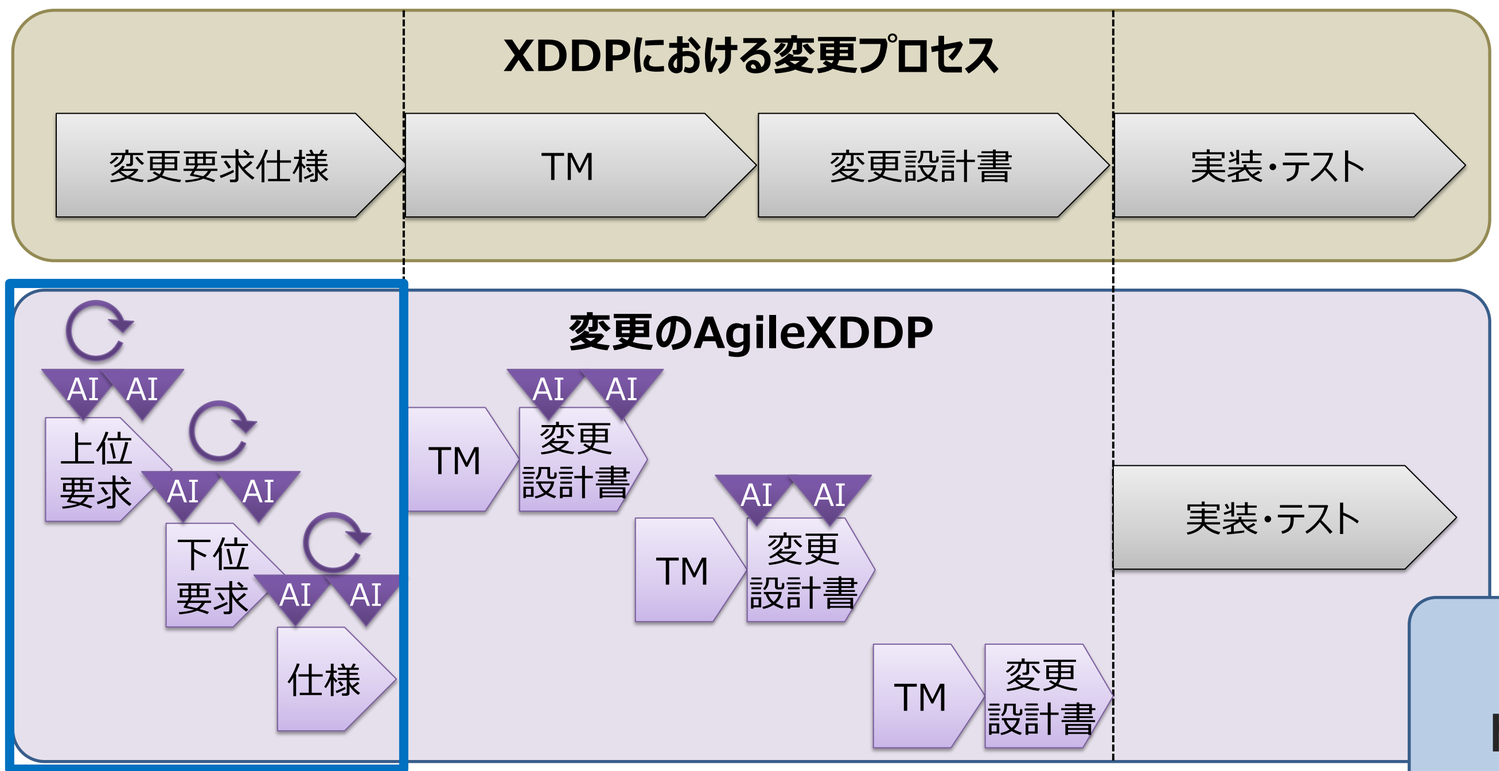
問題の深堀：

- 詳細な仕様に目が向き、要求の定義、確認がおろそかになる
- 思い込んだまま進んでしまい、仕様モレやミスに気付かない
- 最後まで書ききると、手戻りによる書き直しを防ぎたくなる
- USDMを正確かつ網羅的に記述するスキルが必要
- USDMとして記載する要求の粒度が難しい

解決すべき課題

変更3点セットの効率的なレビューの機会と記述スキルの向上

変更のAgileXDDP



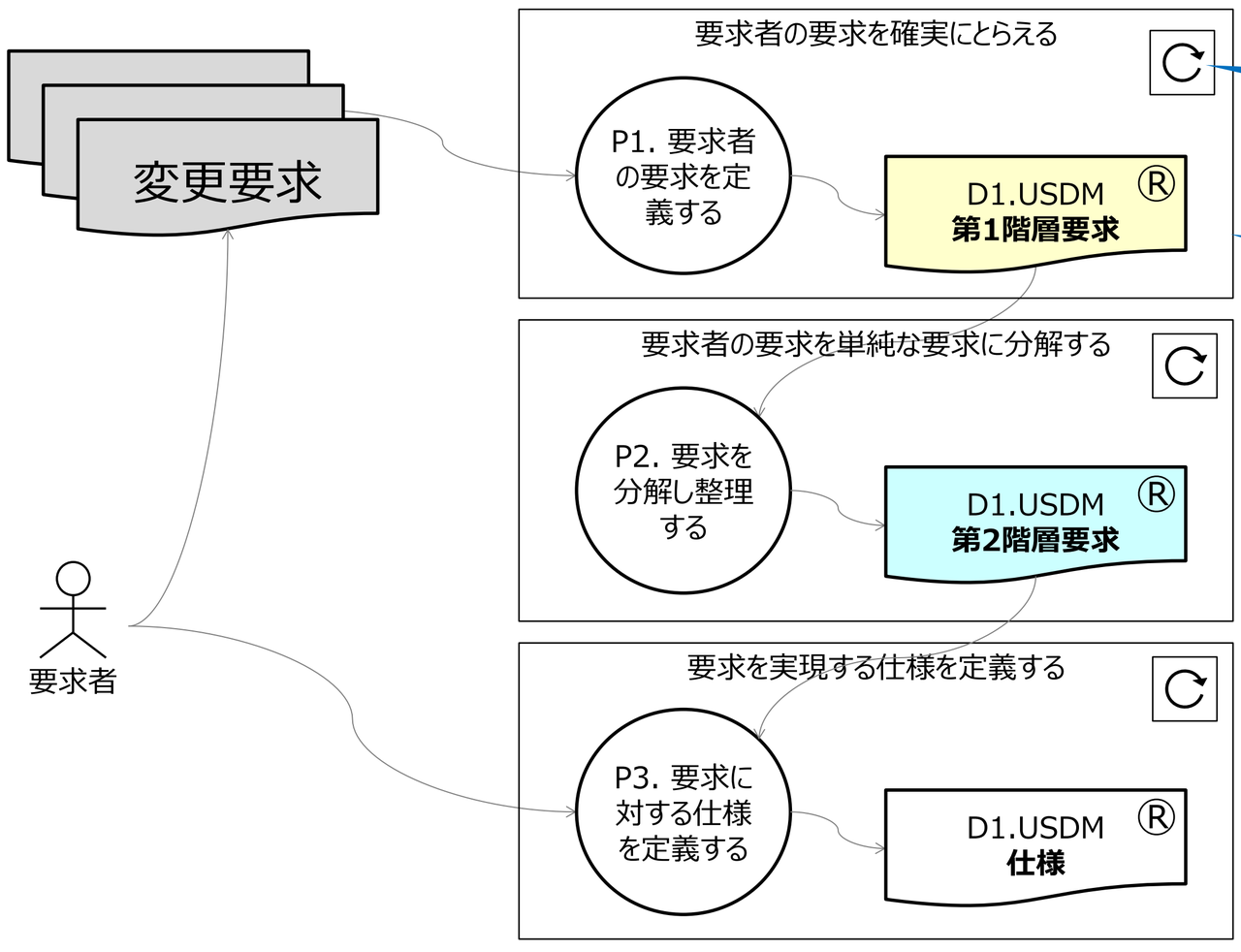
- 変更要求仕様・TM・変更設計書をイテレーティブに行う。
- 変更要求を確実に抑えてから下位要求・仕様を固める。
- アジャイルインスペクションにより欠陥の早期発見と記述スキルを向上

アジャイルインスペクションとは

ドキュメントと書き手の品質レベルを上げる

- ドキュメント作成途中にサンプリングし短いタイムボックスでレビューする
- 決めた品質になるまで修正／サンプリング／レビューを繰り返す

変更要求仕様のPFD



- 要求の勘違いを早期に防止
- 仕様ミス・勘違いゼロ
- USDM記述のスキル向上

AgileXDDP

AgileXDDPとは？

2つのAgileXDDPから成り立つ、Agileの課題とXDDPの課題の両方を解決する手法です。

変更のAgileXDDP

XDDPの課題を解決する「変更」のための手法

☜XDDPの問題

- 変更3点セットで進めても、適切なレビュー機会がないと、モレやミスが発生する。
- 変更3点セットの各成果物を書ききってからレビューを行うと手戻りが発生しやすい。
- 変更要求仕様書に用いられるUSDMを、正確かつ網羅的に記述することが難しい

機能追加のAgileXDDP

Agile開発の課題を解決する「機能追加」のための手法

Agile開発の問題☞

- イテレーションを繰り返し、ソフトの規模が増大化すると、デグレードの問題が多発する。
- 既に動いているソフトウェアへ、Agile開発を適応し、機能を追加するのは難しい。特に，仕様やテストケースが失われている場合，デグレードを防ぐために多大な労力が必要になる。

機能追加のAgileXDDP

問題の深堀：

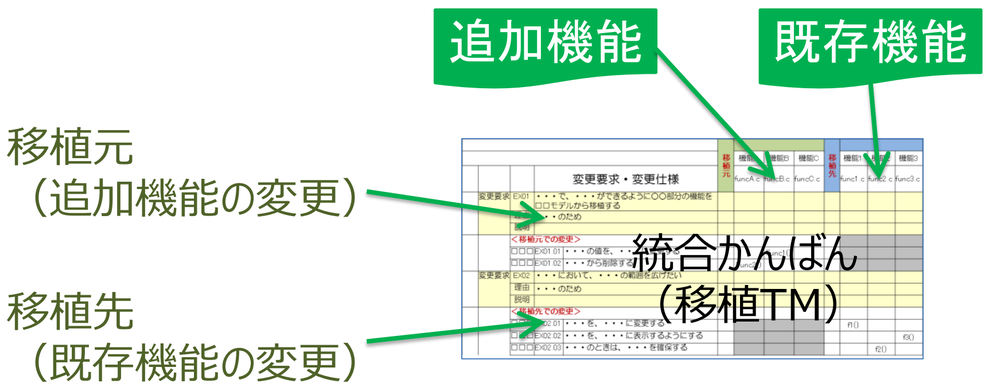
- 機能追加時に既存機能への影響の確認が十分に行われていない
- デグレードによる手戻りが大きく、なかなか品質が安定しない
- 単体テストで安心しても、統合テストでバグがバラバラ出る

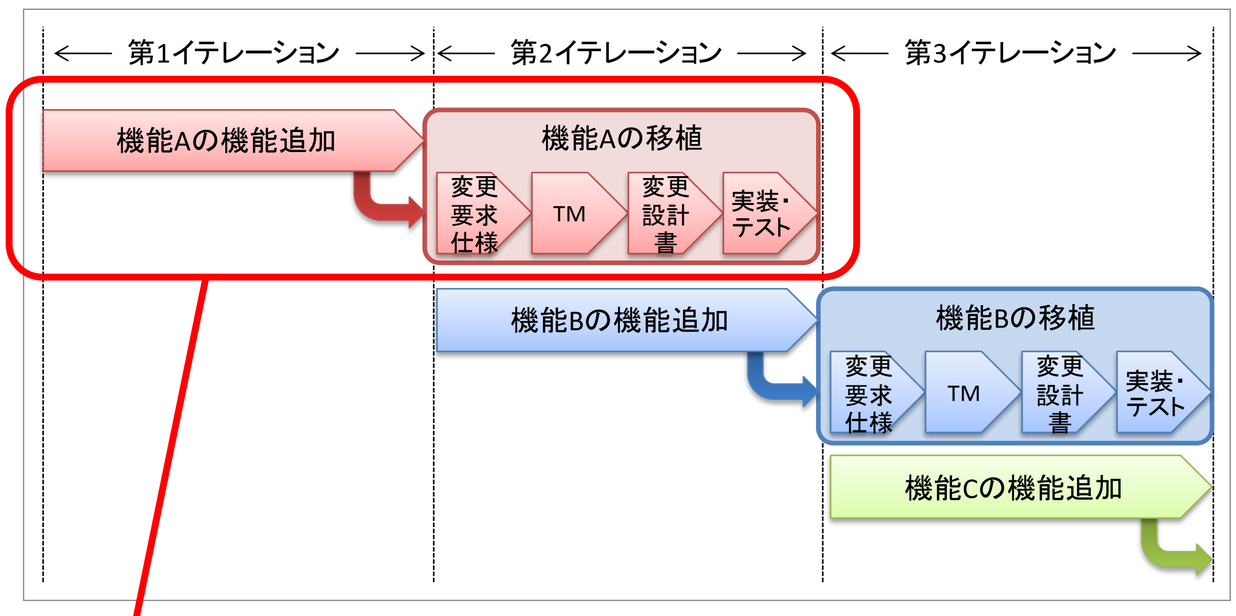
解決すべき課題：

Agileのスピード感を保ちつつ、デグレゼロで機能を追加する

機能追加のAgileXDDP

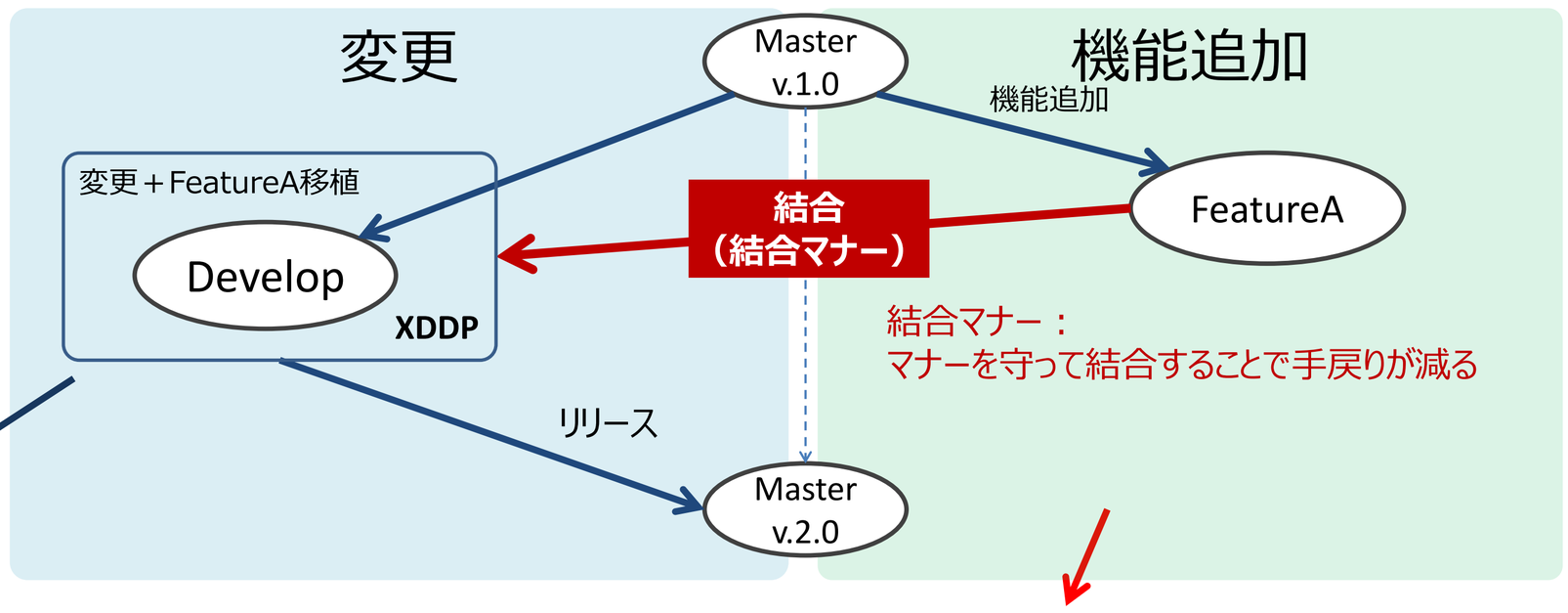
- 機能ごとに開発を行い、その後XDDPのプロセスを用いて移植（結合）を行う



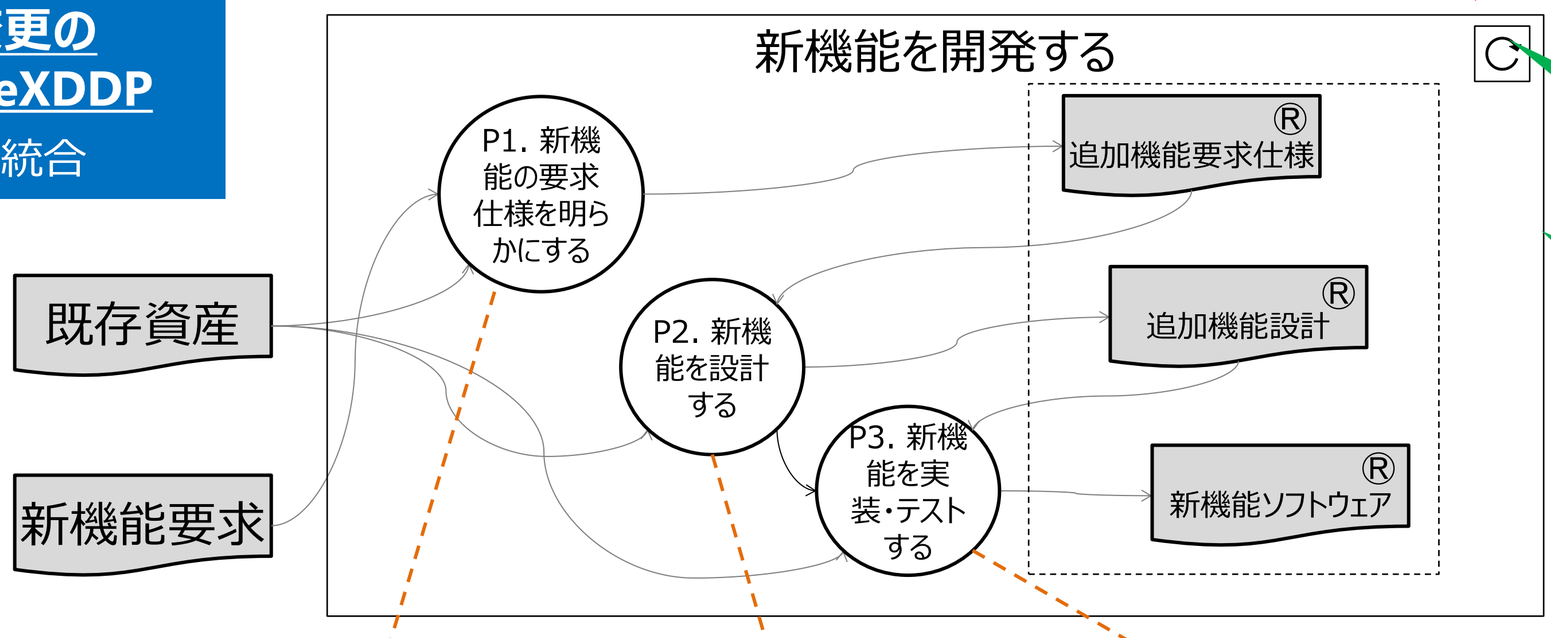


AgileXDDPのブランチ管理・開発のイメージ

- 統合かんばん（移植TM）により、母体との結合を確実に行う
- 関連した小機能なので手間がかからない
- 機能追加でデグレゼロ
- 大規模開発に有効



変更のAgileXDDPで統合



PFDとかんばんを連携

ToDo	Doing	Done
A機能の××仕様	A機能の××設計	A機能の○○機能