派生開発カンファレンス2022

MOTION & CONTROL™
NSK

産業機械向けメカトロ製品の ソフトウェア開発における XDDPの導入と実践事例紹介

日本精工株式会社
技術開発本部
新領域商品開発センター
渡辺　勇人

日本精工株式会社

目次

目次

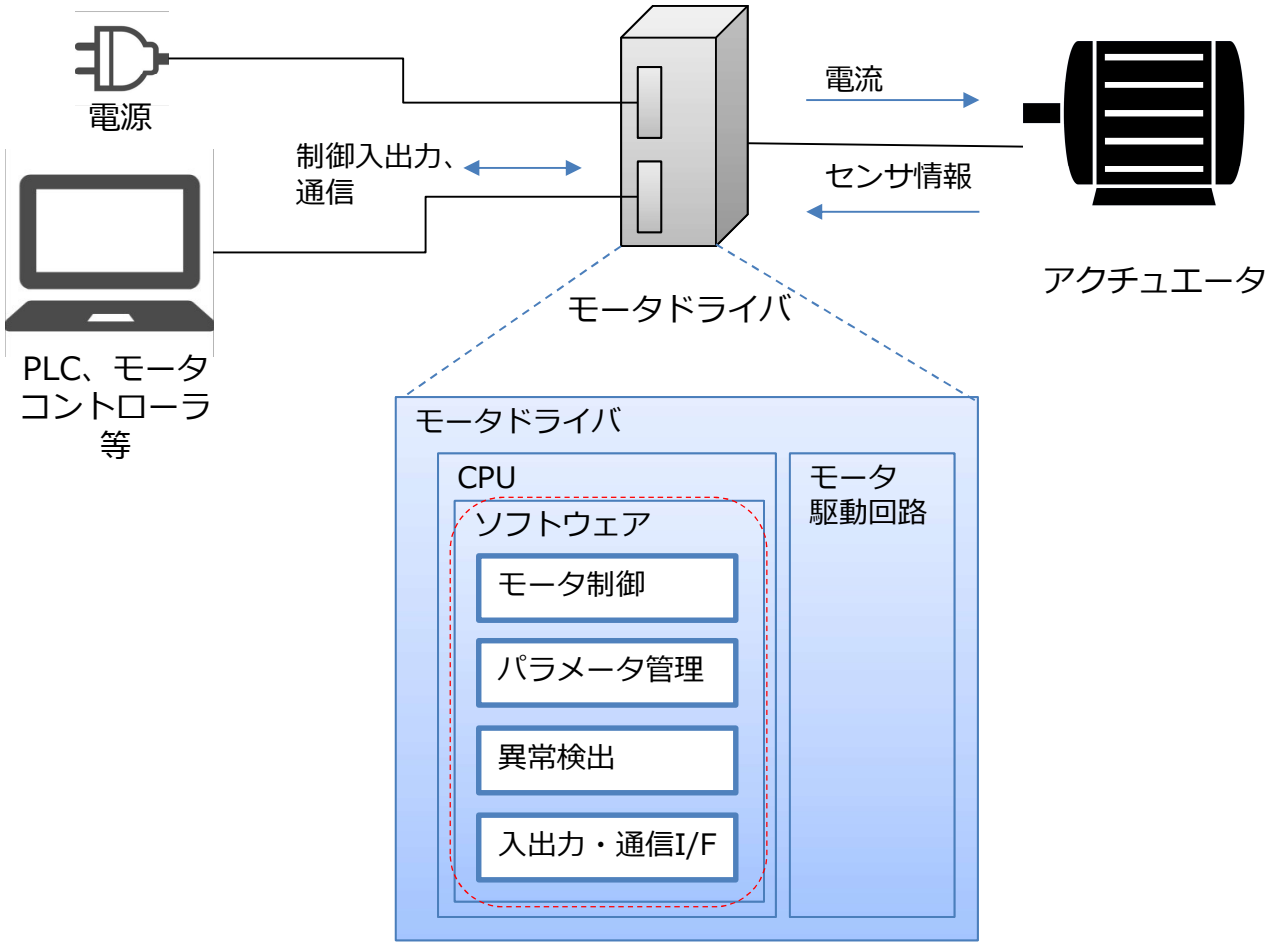
投影
のみ

新領域商品開発センターの製品グループ

製品グループ		スコープ	製品タイプ	開発種別	要求元
PG1	現行技術を流用したアクチュエータ 	現在	汎用品	派生開発	お客様から機能追加の要望を受けて開発する 本日のご説明範囲
PG2	要素技術を利用した新領域アクチュエータ 	3~5年先	特定用途	新規開発（量産） ↓ 派生開発	お客様から要望を受けて開発するものと、自社で要求を考えて開発するものがある
PG3	ニーズ発掘のためのプロトタイプ 	5~10年先	特定用途	新規開発（試作）	自社で要求を考えて開発する

モータドライバ等のアクチュエータ

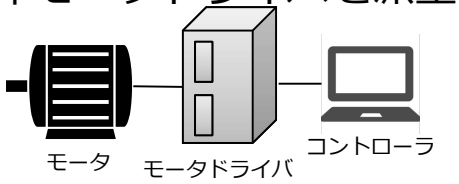
システム構成



目次

背景と課題

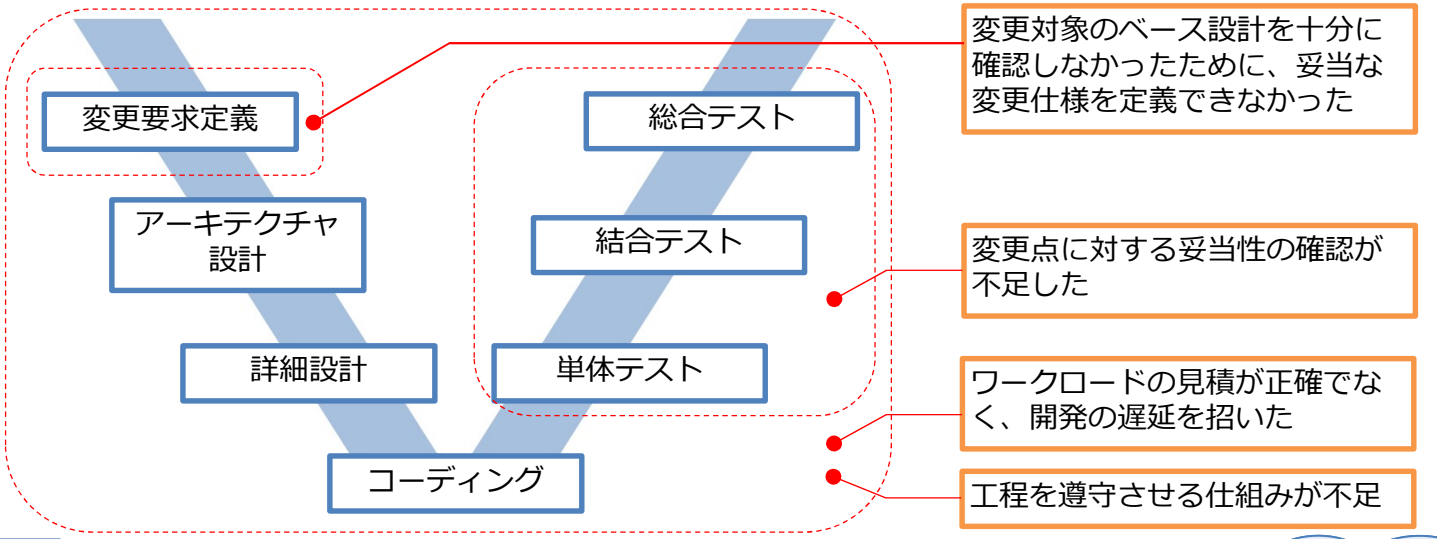
■試作モータドライバを派生開発した際に発見したソフトウェア不具合



特定コマンド実行後パラメータ書き込み不良
通信で特定コマンドを実行した際、ROMにパラメータが保存されない。

振り返り

■試作開発時に実施したプロセスを不具合の”発生”と”流出”の観点で振り返り、問題点を抽出



従来施策	問題をなぜなぜ分析し 再発防止策をプロセスに組み込む

再発防止工数増加
→開発工数余裕減少
→開発のモレが増加？
→不具合増加？
の負のスパイラルが懸念

やり方を変える

改善項目	・現行プロセス見直し ・試作モータドライバ ソフト設計再検証

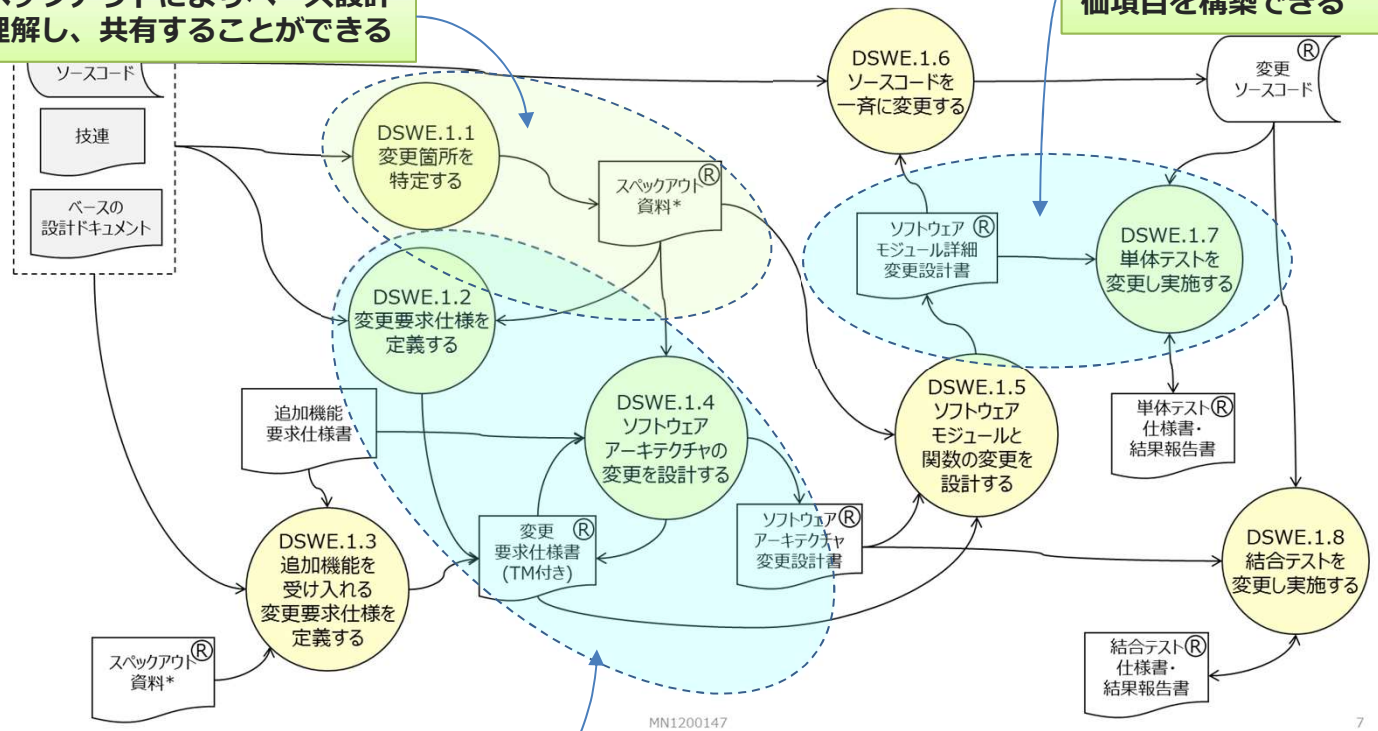
派生開発プロセス（XDDP）導入とその狙い

変更対象のベースとなる設計の
確認・理解不足

スペックアウトによりベース設計を理解し、共有することができる

変更点に対する妥当性の確認の不足

落とし込まれた設計に対応する評価項目を構築できる



ワークロードの見積もり不足
(技術課題の把握が不十分)

変更点がどのくらいの規模か、アーキテクチャへの影響はどの程度か、把握することができる

工程を遵守させる仕組みが不足

各プロセスの入出力の明確化されているため、工程の必要性が理解できる

問題点

XDDPによるメリット

振り返りから抽出された問題点を解決できるXDDPを導入

課題とアプローチ

課題

変更対象のベースとなる設計の確認・理解不足	変更点に対する妥当性の確認の不足
ワークロードの見積もり不足 (技術課題の把握が不十分)	工程を遵守させる仕組みが不足

アプローチ

XDDPの導入、定義

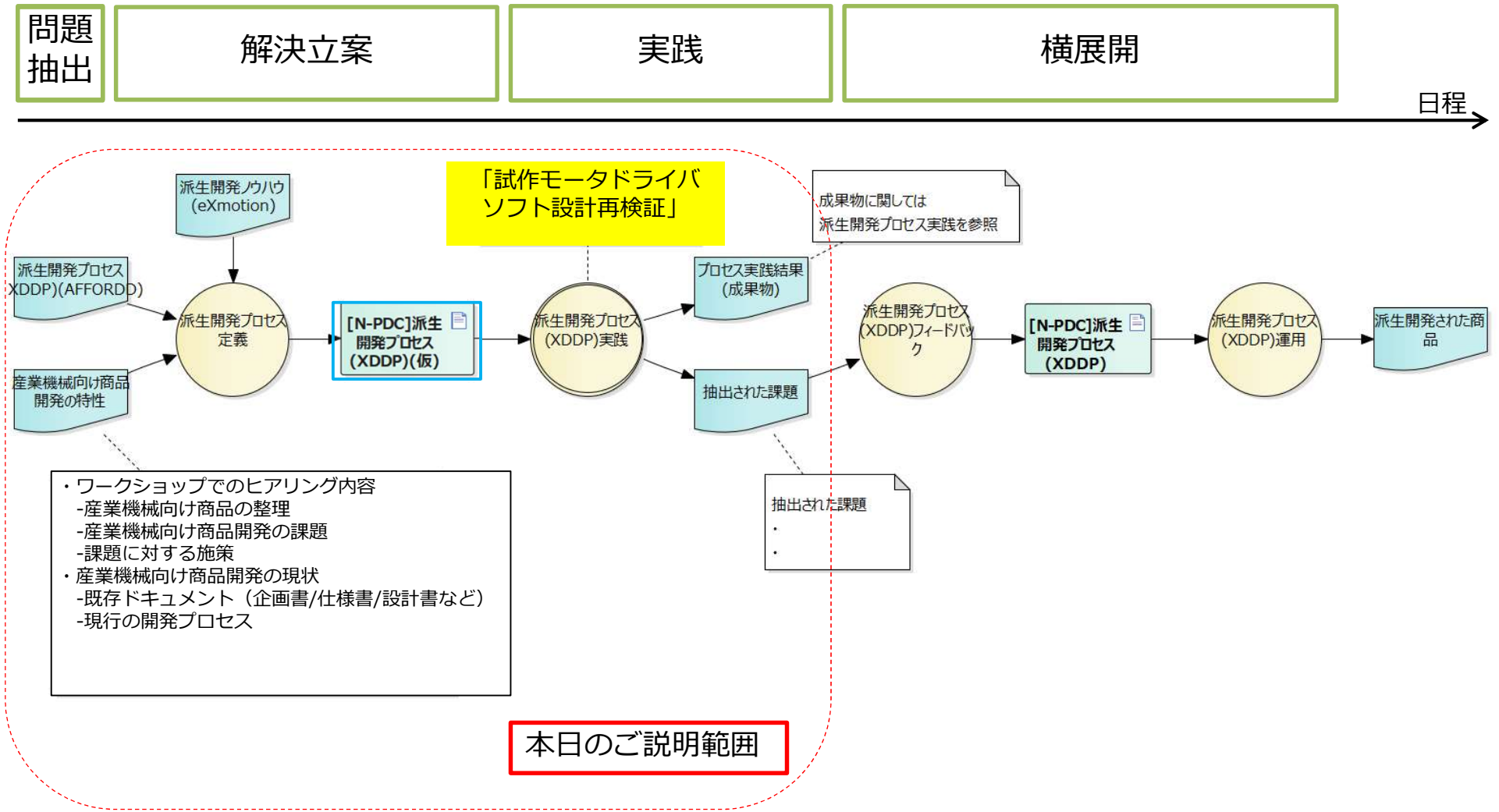
振り返りから抽出された現行プロセスの問題点を解決できるXDDPを導入、定義する

ソフト設計再検証

本来はトップダウンで開発時に品質を高める手法であるXDDPを、**既に開発されたソフトウェアに対してボトムアップ的に実施**し、ソフト設計を再検証する

ソフト設計再検証に元設計者以外のメンバも参画する

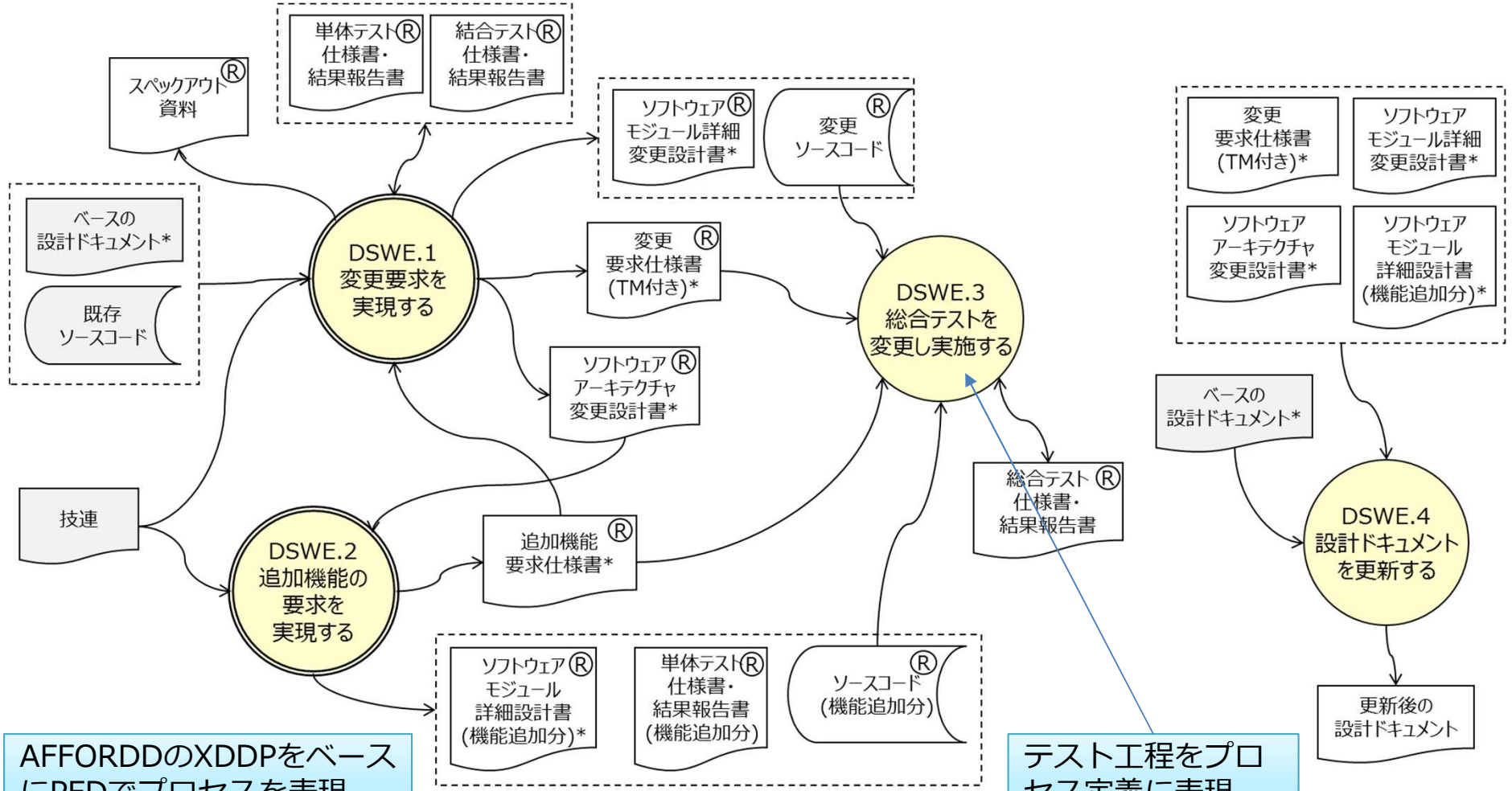
派生開発プロセス(XDDP)定義と実践の流れ



目次

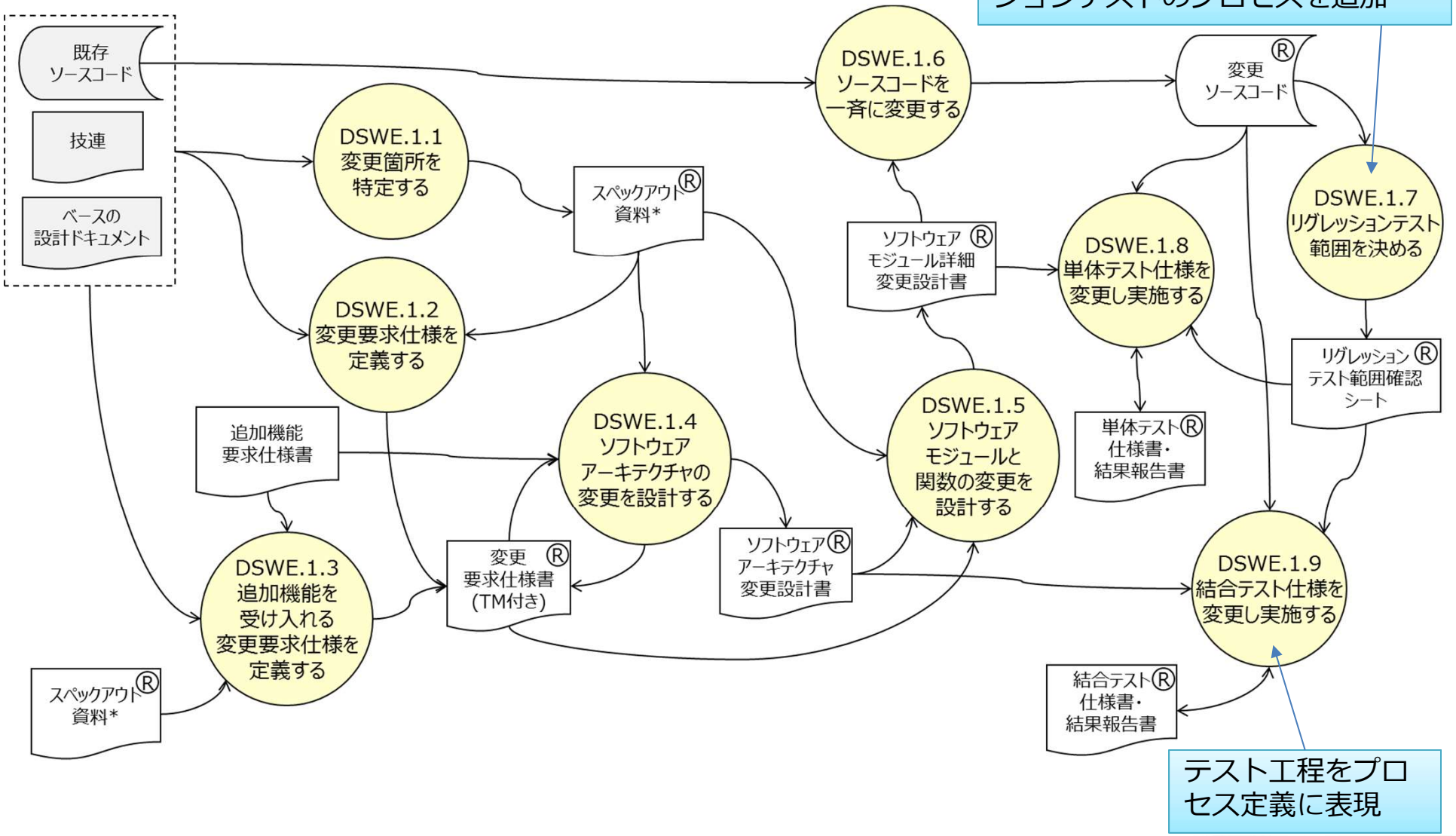
派生開発プロセス(XDDP)　全体

eXmotion様ご協力のもと、XDDPプロセス定義書を作成



派生開発プロセス(XDDP)　DSWE.1詳細

既存工程で定義していたリグレッションテストのプロセスを追加



テスト工程をプロセス定義に表現

派生開発プロセス(XDDP)　タートル図一例

ID	DSWE.1.1	Name	変更箇所を特定する
Objective	変更要求の影響範囲を理解し、ソフトウェアの変更すべき部分を特定する。		

入力成果物

- 技連
- ベースの設計ドキュメント
- 既存ソースコード

作業内容

1. 技連等の情報から変更要求を把握する。
2. ベースとなる設計ドキュメントやソースコードから変更の影響を受ける部分を調査する。
3. 影響を受ける部分の構造や振る舞いを以下のようなモデルを使ってスペックアウト

構造		振る舞い	
システム構造	パッケージ図、ブロック定義図	状態間の関係	状態マシン図、状態遷移表
データ構造	クラス図、ER図	モジュール間の相互作用	シーケンス図
モジュール、関数の関係	クラス図、構造図	制御の手順	アクティビティ図、タイミング図

4. スペックアウト資料を有識者に見せ、間違いがないか、他に調査すべき箇所がないかを確認する。
5. 必要に応じて、有識者の指摘に従いスペックアウト資料を修正する。

出力成果物

- スペックアウト資料

実施者

- 担当するソフトウェア開発者

支援文書

プロセス実施上の留意点など

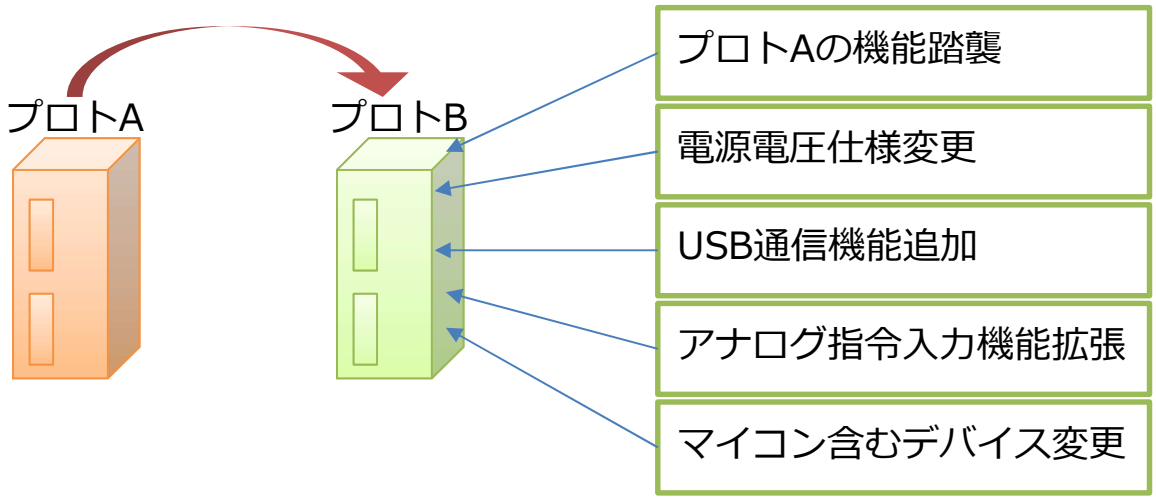
- XDDPトレーニング資料 P.41～45を参照

タートル図で各プロセスごとに作業内容と、入出力成果物を明確化することで、工程の必要性が理解できる

目次

試作モータドライバ 派生開発の経緯

プロトAの後継機種として機能踏襲しつつ、いくつかの機能変更するプロトBを開発

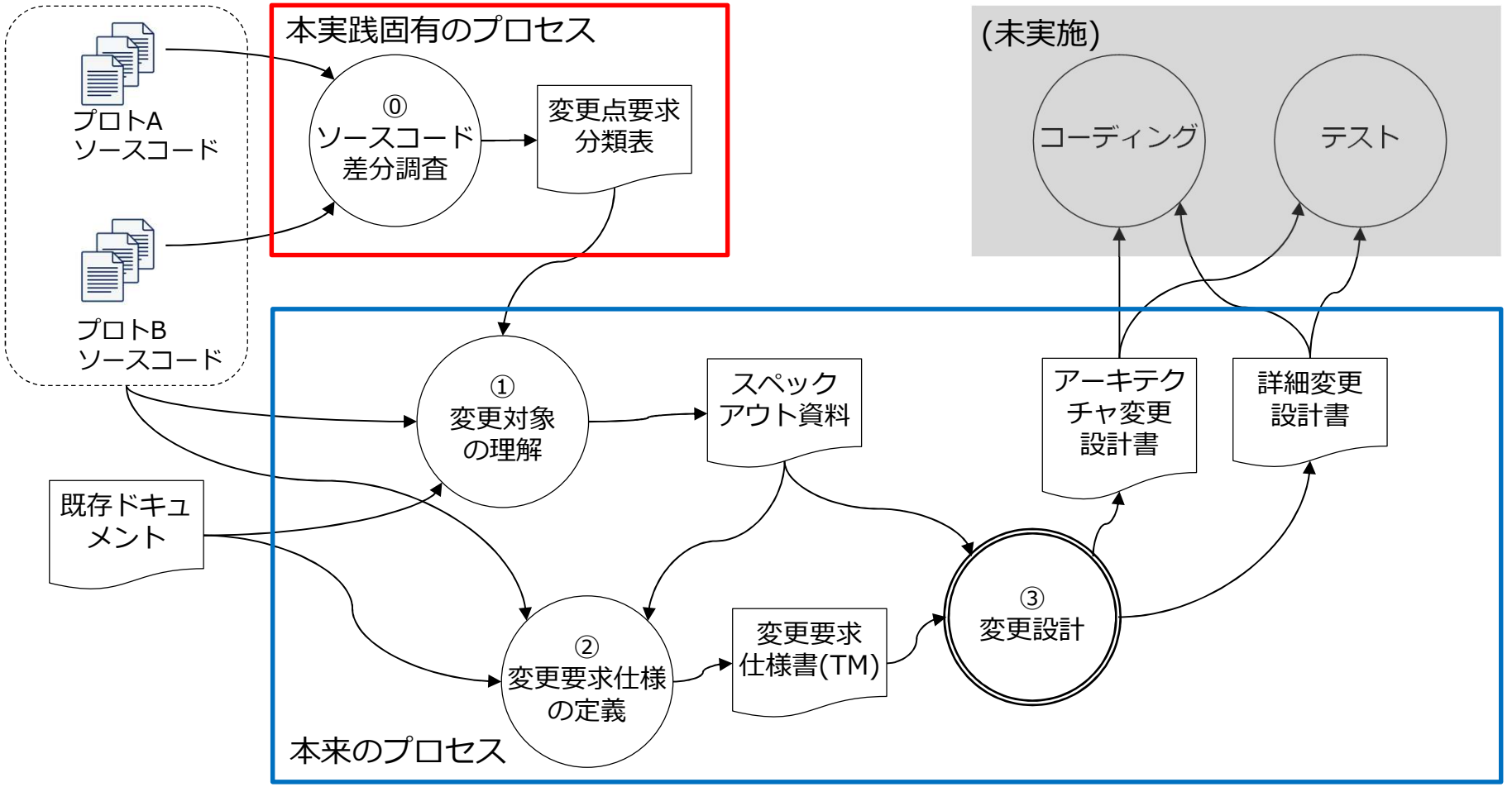


プロトAからプロトBのモータドライバを派生開発した案件をモチーフに
すでにあるソフトウェアの全差分について、XDDPのプロセスを適用し、
改めて成果物を作り直してレビューすることで、
プロトBのソフトの設計再検証・品質検証を行う

XDDPの組織定着	ナレッジ共有

実践事例紹介（試作モータドライバ ソフト設計再検証）

実践の全体の流れ

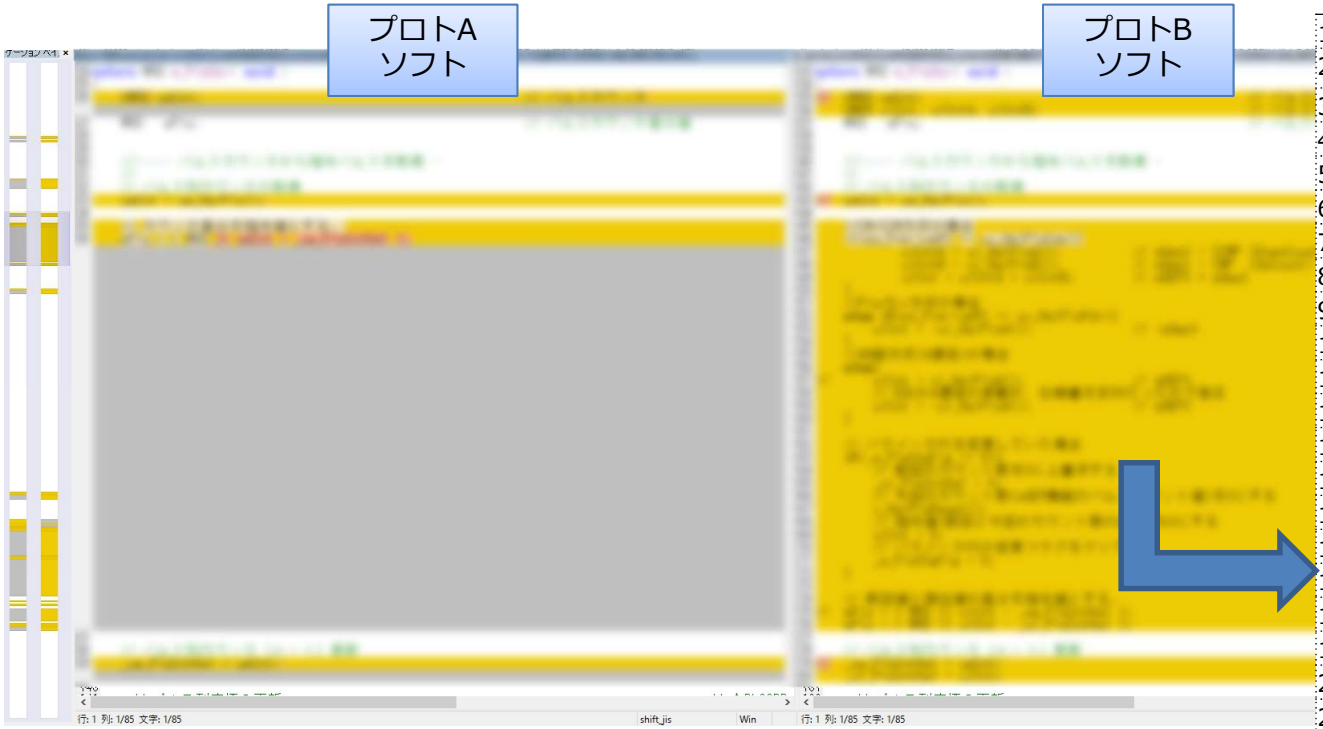


本来のプロセスとは以下の点で異なるプロセスで取り組んだ
- ソースコード差分調査を行い、全差分を要求項目に分類
- 設計まで実施し、以降のプロセスは未実施

実践事例紹介（試作モータドライバ ソフト設計再検証）

⓪ソースコード差分調査

ポイント



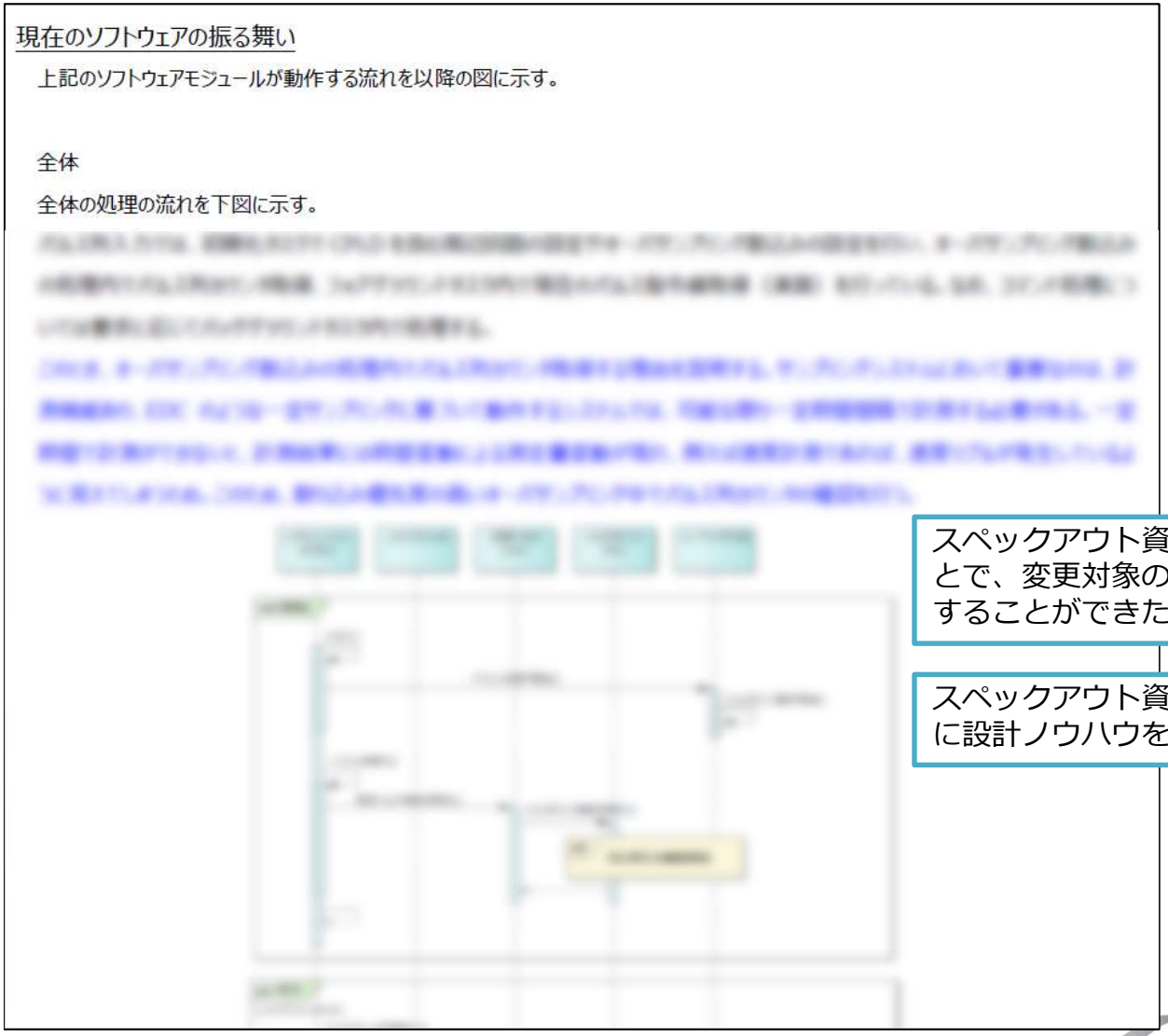
1.パルス列カウント方式の変更
2.位置検出の変更
3.位置フィードバック方式の変更
4.
5.
6.
7.
8.
9.
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24.リファクタリング
25.その他ハードの変更
26.デバッグ用処理
27.プロトAの改善

プロトAとプロトBのソースコード全差分を要求ごとに分類
要求ごとにXDDP成果物を作成

実践事例紹介（試作モータドライバ ソフト設計再検証）

①変更対象の理解

ポイント



スペックアウト資料を作成することで、変更対象の設計を深く理解することができた

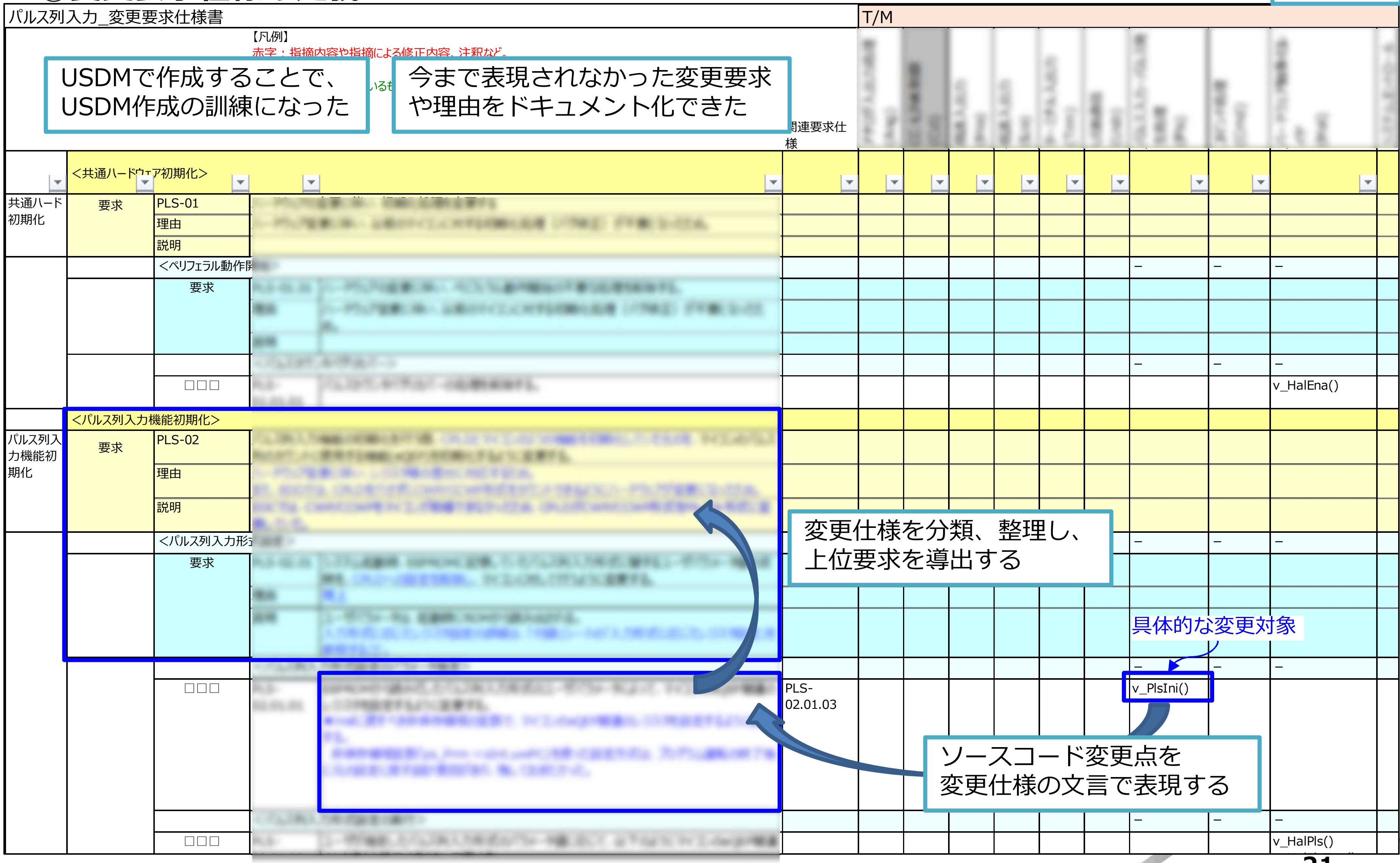
スペックアウト資料のレビュー時に設計ノウハウを共有できた

実践事例紹介（試作モータドライバ ソフト設計再検証）

MOTION & CONTROL™ NSK

②変更要求仕様の定義

ポイント



実践事例紹介（試作モータドライバ ソフト設計再検証）

③変更設計

ポイント

アーキテクチャ変更設計書

対象の追加機能要求仕様

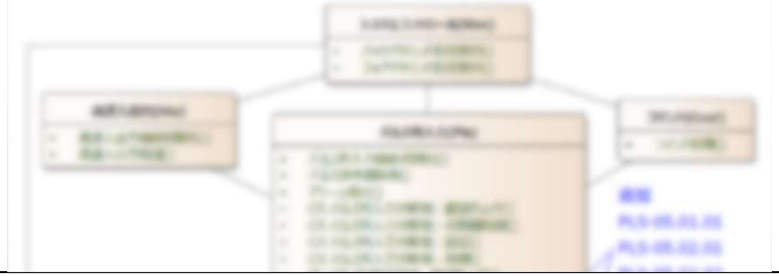
追加機能要求仕様書名：なし

修正方針

構造の変更(論理ビュー)

変更要求仕様に対応するためには、ソフトウェアアーキテクチャの構造を以下のように変更する必要がある。

変更後の構造を下図に示す。



詳細変更設計書

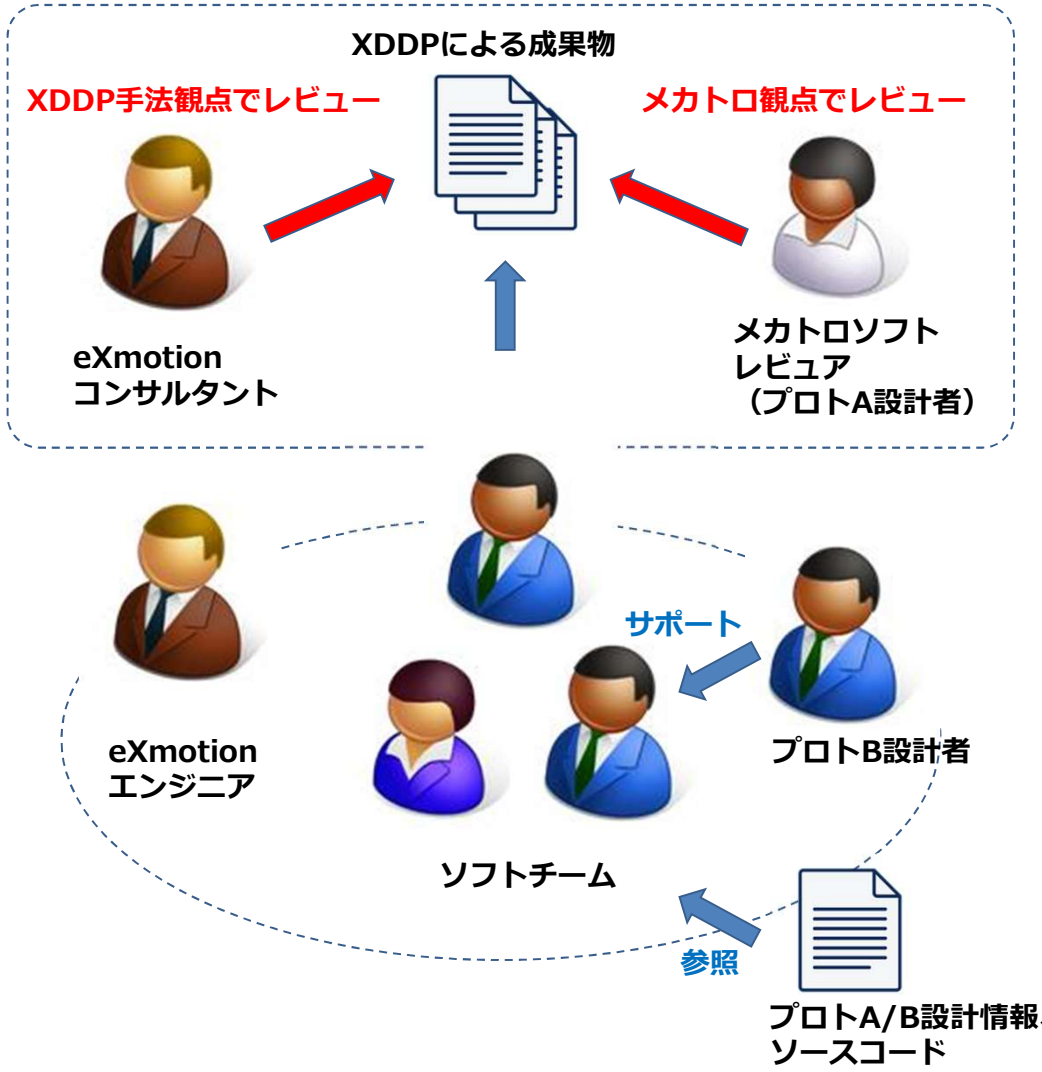
変更内容

番号	変更内容	予想行数	
1	パルスカウンタ変数 uwCnt のサイズを UWRD 型から ULNG 型に変更する。 また、サイズの変更に伴い、変数名を uwCnt から ulCnt に変更する。 ※ソースコード上では対応されていない。 Hal から 32bit で取得しているものの、Pls にて上位 16bit は切り捨てして下位 16bit が取り出されている。現状は下位 16bit だけでも十分なサイズであるため問題なし。ただし、本項目を変更する理由はない。 関連：PLS-03.02.01 ★マイコンの設定(QPOSMAX)でカウンタサイズを設定できるため、変化点を少なくする意味	1	□

- UMLモデリングの訓練ができた
- 設計レベルで設計の妥当性が議論できた
- 設計レビューを通じて、設計ノウハウやナリッジ共有することができた
- 設計レベルでリファクタリングのポイントを抽出できた

実践事例紹介（試作モータドライバ ソフト設計再検証）

実践事例の概要



ポイント

レビューが大変
→レビュー効率化を検討

レビューによる指摘事項のレベル分け

★★★：緊急を要する修正
★★　：要修正
★　　：改善項目

目次

XDDP導入による成果

既存プロセス問題点	XDDP導入による成果	今後の課題
変更対象のベース設計を十分に確認しなかったために、妥当な変更仕様を定義できなかった	• スペックアウトによりベース設計に対する担当者自身の理解が深まる。その結果、設計根拠が明確になる効果があった。 • 内容をレビューアと共有することができ、変更要求仕様書（USDM）で変更要求仕様のレビューがしやすくなった。 • 工程の後戻りが少なくなった。	メトリックス（工程後戻り回数や、仕様起因不具合件数など）の定義と定量的効果測定
ワークロードの見積が正確でなく、開発の遅延を招いた	• 変更要求仕様書、およびトレーサビリティマトリックスにより、変更設計の全体を見通すことができ、設計から実装まで抜け漏れが発生しにくくなった。 • 変更ボリュームをある程度把握できるようになった。 • WBSを作成できるようになった。	見積もり精度向上
変更点に対する妥当性の確認が不足した	要求仕様、設計に対応するテスト項目を抽出することができるようになった。	メトリックス（テスト漏れ件数など）の定義と定量的効果測定
工程を遵守させる仕組みが不足していた	• 各工程の入出力が明確になり、工程を飛ばすことが無くなった。 • レビューが活発に行われるようになった。	レビュー工数増加 レビューア負担増加

- XDDP導入により既存プロセス問題点を改善することができた
- 定義以降、現在までに7つの派生開発プロジェクトで実践
- メトリックスによる定量的効果測定と見積もり精度向上、レビュー効率化が今後の課題

試作モータドライバ ソフト設計再検証の成果

課題		成果
ソフト品質懸念	他に不具合がないか	抽出した設計課題リスト作成と、各課題のレベル分けにより、緊急性を要する不具合がないことが確認できた。
	設計指針に反した設計がされていないか	改善、リファクタリング箇所が明確になった (リファクタリング箇所の例：アーキテクチャの崩れ、不要コード、可読性の低いコード)
開発の属人化		プロトBや、そのベースとなったプロトAのソフト設計のノウハウが、レビューを通して共有でき、若年層のレベルアップとナリッジ共有することができた
XDDP実践スキル習得		計5名のメンバーがXDDP手法を習得できた

- 試作モータドライバのソフト課題を解決することができた。
- XDDP手法の習得と組織定着を進めることができた。

目次

まとめ

- 背景と課題
  - ソフトウェア派生開発の品質向上
- 解決策
  - XDDPプロセス定義と実践
- 実践事例紹介
  - 試作モータドライバのソフト設計再検証
    （コード数万行、8名、合計工数　約30人月）
    →設計課題抽出／ナリッジ共有／組織定着
- 今後
  - レビュー効率化／ソフト開発体制強化
  - メトリックスによる効果測定
  - 他プロジェクトによる実践
- 最後に

本活動に多大にご尽力くださいました
eXmotion様に感謝申し上げます。

ご清聴ありがとうございました。