「プラットフォーム開発、XDDP開発、そしてSPL開発へ」

～ 移行のきっかけ(取り巻く環境)と成功/失敗のステップ ～

オムロン ソフトウェア株式会社
プロセス革新本部 ソフトウェアプロセス開発部
筒井　賢

Agenda

- 自社紹介

- ソフトウェアプロダクトライン(SPL)開発とは
- 派生開発プロセス「XDDP」とは

- 事例対象組織の概要
- 事例対象組織における再利用開発の移行パターン
- XDDP開発の導入
  ～ プラットフォーム開発からXDDP開発への移行 ～
- SPL開発の導入
  ～ XDDP開発からSPL開発への移行 ～

- 今後の取組み
  ～ SPL移行後を考える ～

自社紹介

オムロン ソフトウェア株式会社

- 設立　　昭和５１年４月８日
- 資本金　　３億６０００万円
- 代表　　宮地　功
- 売上　　１３１億７３５６万円(２０１３年3月)
- 社員数　　４４４名(２０１３年3月)
- 拠点　　本社(京都)、事業所(東京、草津)、オフィス(幕張、大阪、尾張旭)
- 関係会社　　オムロン ソフトウェア(上海)有限公司

主な事業領域

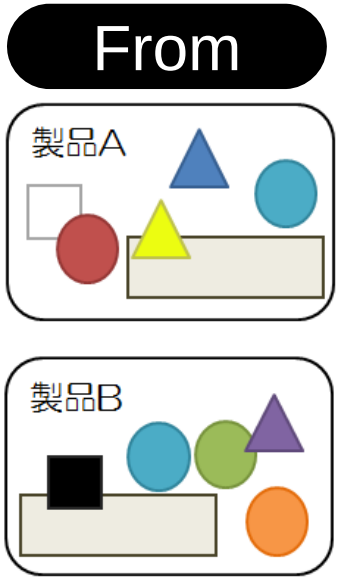
- オムロングループの組込ソフト開発
  - インダストリアル オートメーション,
  - ソーシャル ソリューション(鉄道, 交通),
  - ヘルスケア(健康, 医療機器), etc.
- 文字入力技術、画像処理技術
- 決済ソリューション
- 監視サービスソリューション
- スマートフォン関連ソリューション
- ネットワークセキュリティ
- ソフトウェア開発コンサルティング

ソフトウェアプロダクトライン(SPL)開発とは

製品ドメインにおいて、戦略的な再利用計画に基づく
製品構成部品を開発し、それを活用して製品を
組み上げる開発手法

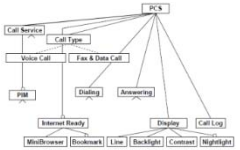
(米カーネギーメロン大学（CMU）ソフトウェア工学研究所（SEI）)

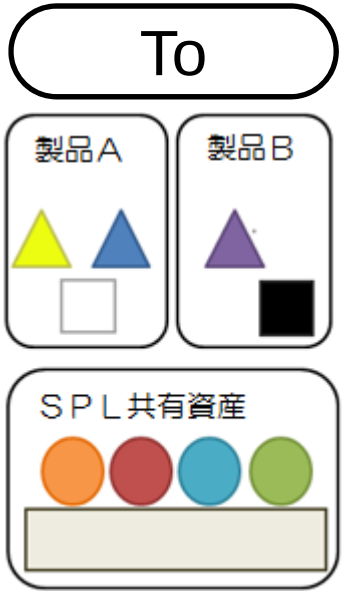
3つの
プロセスの協調



製品戦略・
ビジネス戦略

製品群の固定部・
変動部を分離



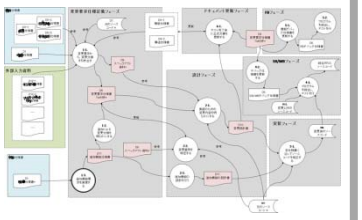


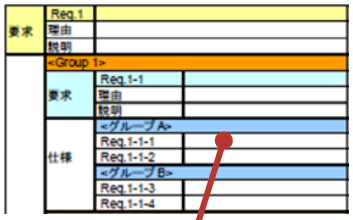
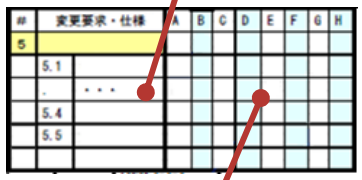
新しい最先端の技術ではなく、過去の
技術の上に立脚した、包括的な手法

派生開発プロセス「XDDP」とは？

* XDDP：eXtreme Derivative Development Process

- XDDPは、派生開発に特化した「変更」と「追加」のプロセス
- 部分理解による「思い込み」「勘違い」があることを前提
- 以下の「3つの成果物」による、モレ・ミス・ムダの防止

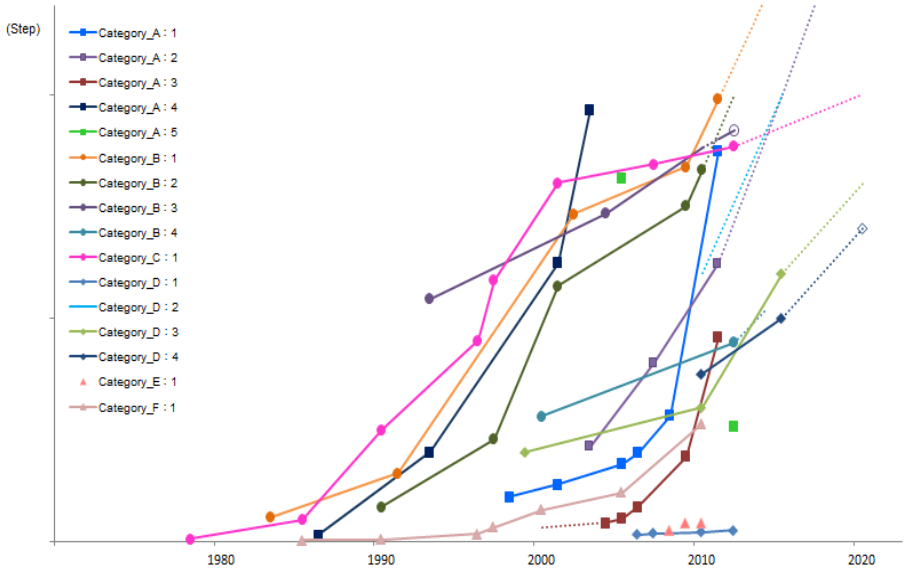


変更要求仕様書 (USDM) Universal Specific Description Manner	Why …変更要求 What …仕様	何を変更するのか？ なぜ変更したいのか？ 何を何に変更するのか？ 他に関連して変更しなければならない仕様を見つけたか？	
トレーサビリティ・マトリクス (TM)	Where …変更箇所の特定	その変更は”どこ”にあるのか？	
変更設計書	How …変更設計	具体的にどのように変更するのか？	

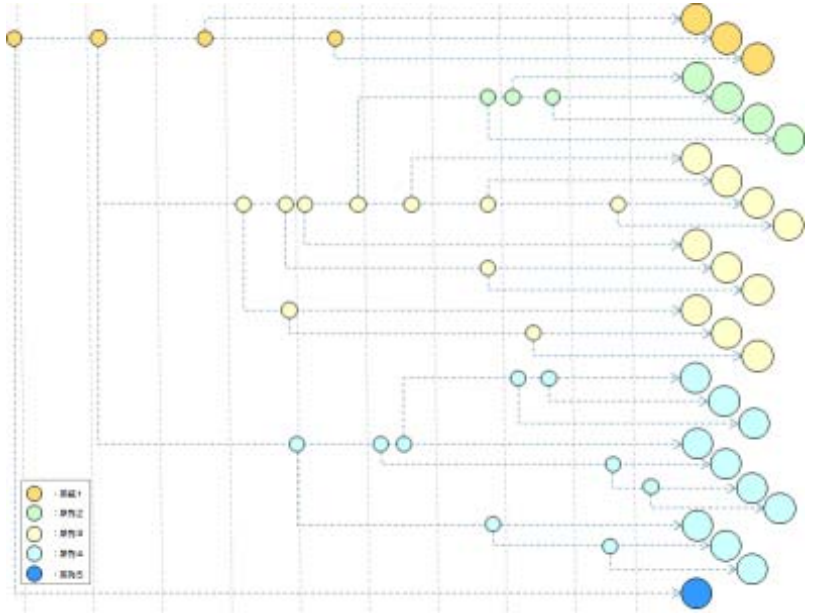
事例対象組織の概要

コンテキスト

- 比較的大規模の組み込みソフト開発
- 開発スタイルは、ベースとなるソフトウェアから改修して製品を作る派生型ソフトウェア開発である



組み込みソフトウェアの大規模化



派生型開発の広がり

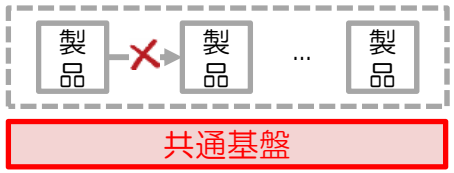
事例対象組織における再利用開発のパターン

本事例

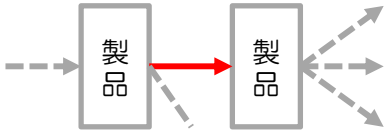
世代＼パターン	パターンA	パターン B	パターン C	パターン D
1	Platform	Platform	Platform	Platform
2	Platform	XDDP	XDDP	SPL
3	Platform	XDDP	SPL	SPL

本資料での定義(SPL視点)

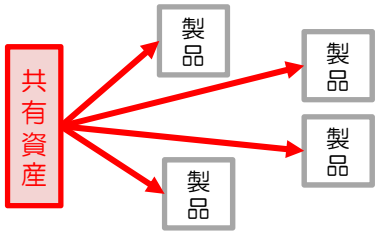
Platform：(開発全体最適)
- 共通基盤上に製品群を構築
- 製品間の共通性/可変性(点)はある程度把握されているが、管理されていない



XDDP：(個別最適)
- 過去製品をベースラインとして製品が構築(多くが再利用)
- 共通性や可変性などの性質については考慮しない



SPL：(事業全体最適)
- 共有資産から製品群を構築
- 共通性/可変性を管理
- 固定部と変動部の組合せ



XDDP開発の導入

～ プラットフォーム開発からXDDP開発へ移行 ～

なぜ、XDDP開発を選択したのか？

開発現場にマッチした手法と判断

- 開発の多くは派生開発
- 負荷変動が大きく、安定した設計者の確保が難しい
- ソースコードのすべては理解できていない
- なかなか仕様がFIXしない

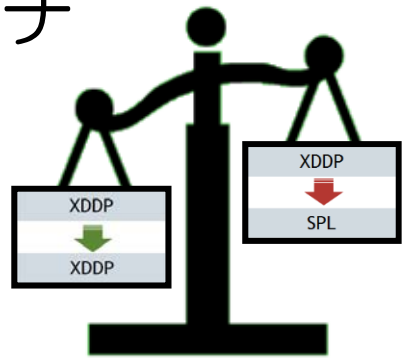
出所：技術評論社
「派生開発」を成功させるプロセス改善の技術と極意, 2007

導入への負担が少ない

- ソフトウェア開発部門だけで着手可能
- 初期投資(資金・時間)が比較的少ない
- 書籍がある(やるべきことが丁寧に記載)

ミドルマネジメント層の関心事とマッチ

- 実施成果がすぐに反映される(短中期視点)

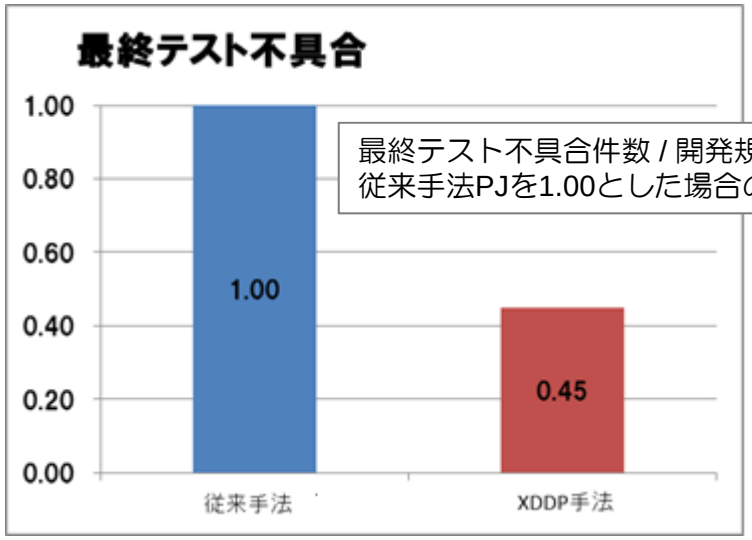


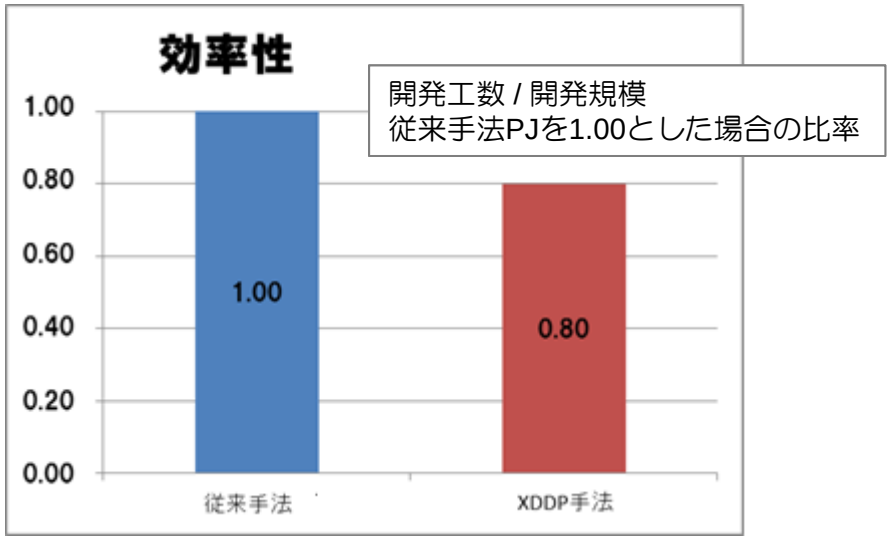
事業は成熟期にある

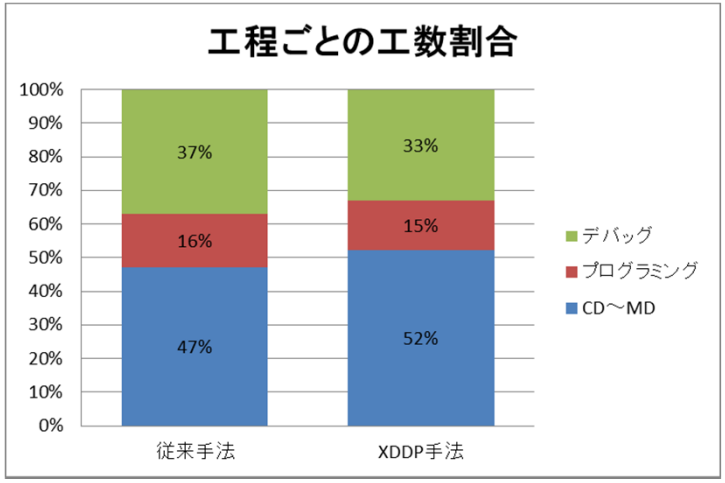
- 膨大な資産もあり、作り変える時期ではない

XDDP開発を実践して　～品質・効率性の両立～

XDDPの展開により、品質(55%)・効率性(20%)の両立を確認







開発上流に工数シフト
一気呵成のプログラミング

XDDPらしい効果も確認

SPL開発の導入

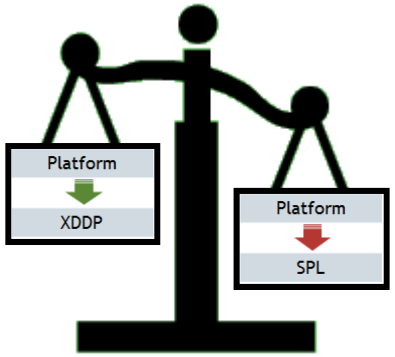
～ XDDP開発からSPL開発へ移行 ～

SPL開発の事前準備としてXDDP開発で実施しておきたいこと

- マネジメント層の関与度をあげる
  - 組織的活動への迅速な対応 (組織計画、資金調達、運営)

- 開発を可視化する(製品・プロセス)
  - ツール導入による負荷軽減への対応 (構造理解、構成管理、データ収集)
- 導入効果(定量的/定性的)を検証する
  - より効率化するための新たな手段への対応 (計画策定、データ収集)
- 効率化された工数の使い道を考える
  - リファクタリング、SPL、改善活動継続への対応 (資金調達、組織計画)
- USDMの記載(要求・理由)に利害関係者を巻き込む
  - 要求や仕様をコントロールしていく活動への対応 (関係づくり)
- USDM, TMを蓄積し、変化する/しない部分を理解する
  - 固定部、変動部といった可変性抽出への対応 (エンジニアリング全般)

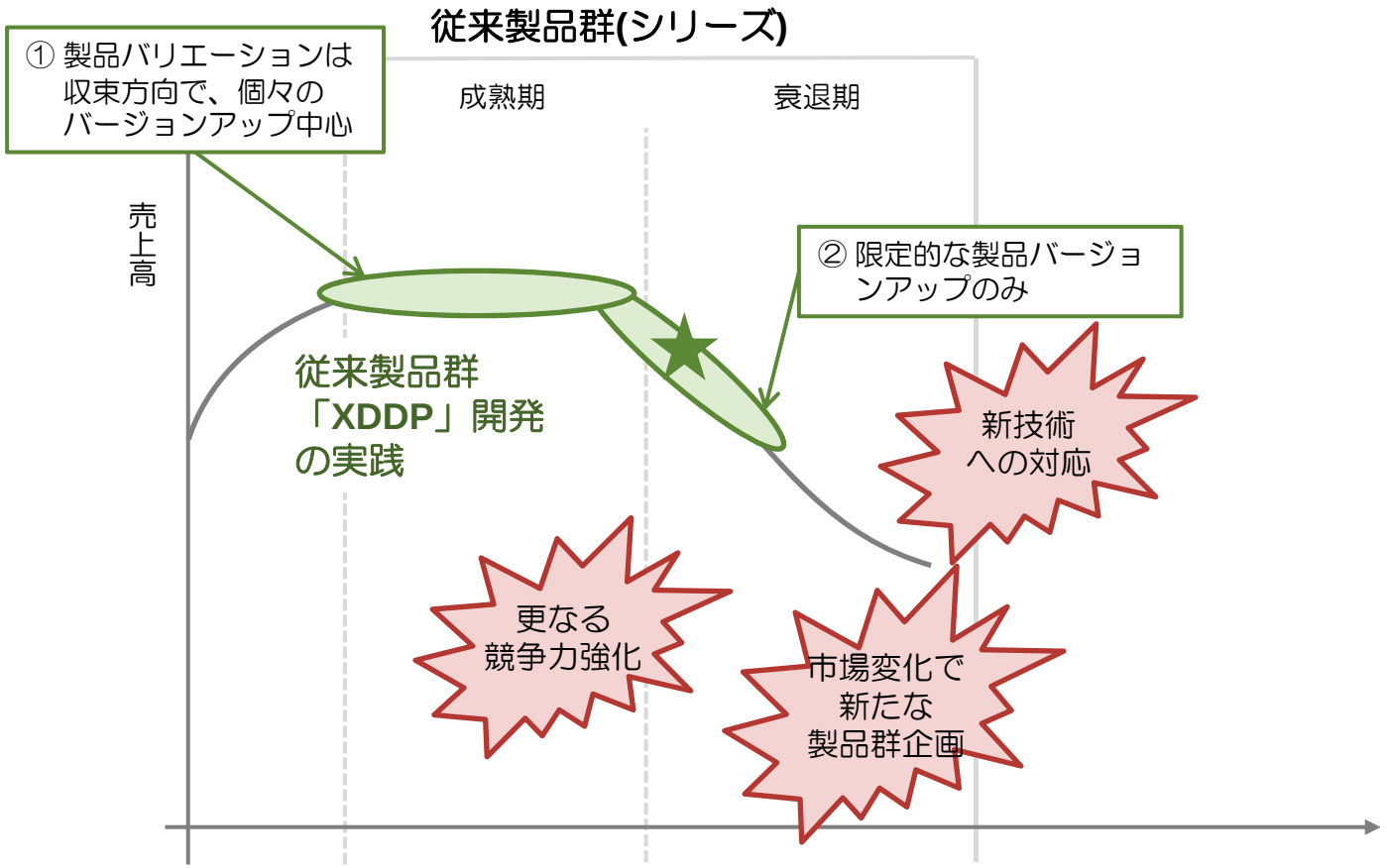
なぜ、SPL開発を選択するのか？



- 更なる競争力強化
  - 高品質・低コスト・短納期を実現
    - XDDP効果を超える高い要求
  - プラットフォームの変更容易性追求
    - 派生元プラットフォームが複雑すぎる
  - 品質確保工数中心からの脱却
    - 影響範囲調査、テストに多大な時間と費用
- 新技術への対応
  - 保守性を高めるアーキテクチャの実現
    - 現行アーキテクチャでは、顧客要求への対応が困難
      - 理解性、魅力性といった使用性への対応
      - 解析性、変更性といった保守性への対応
- 事業は成熟期にある・・・
  - 市場変化で新たな製品群企画(製品群ライフサイクル)が立ち上がる
  - 新たなバリエーション / バージョンアップ開発
  - 投資回収の目途が立った

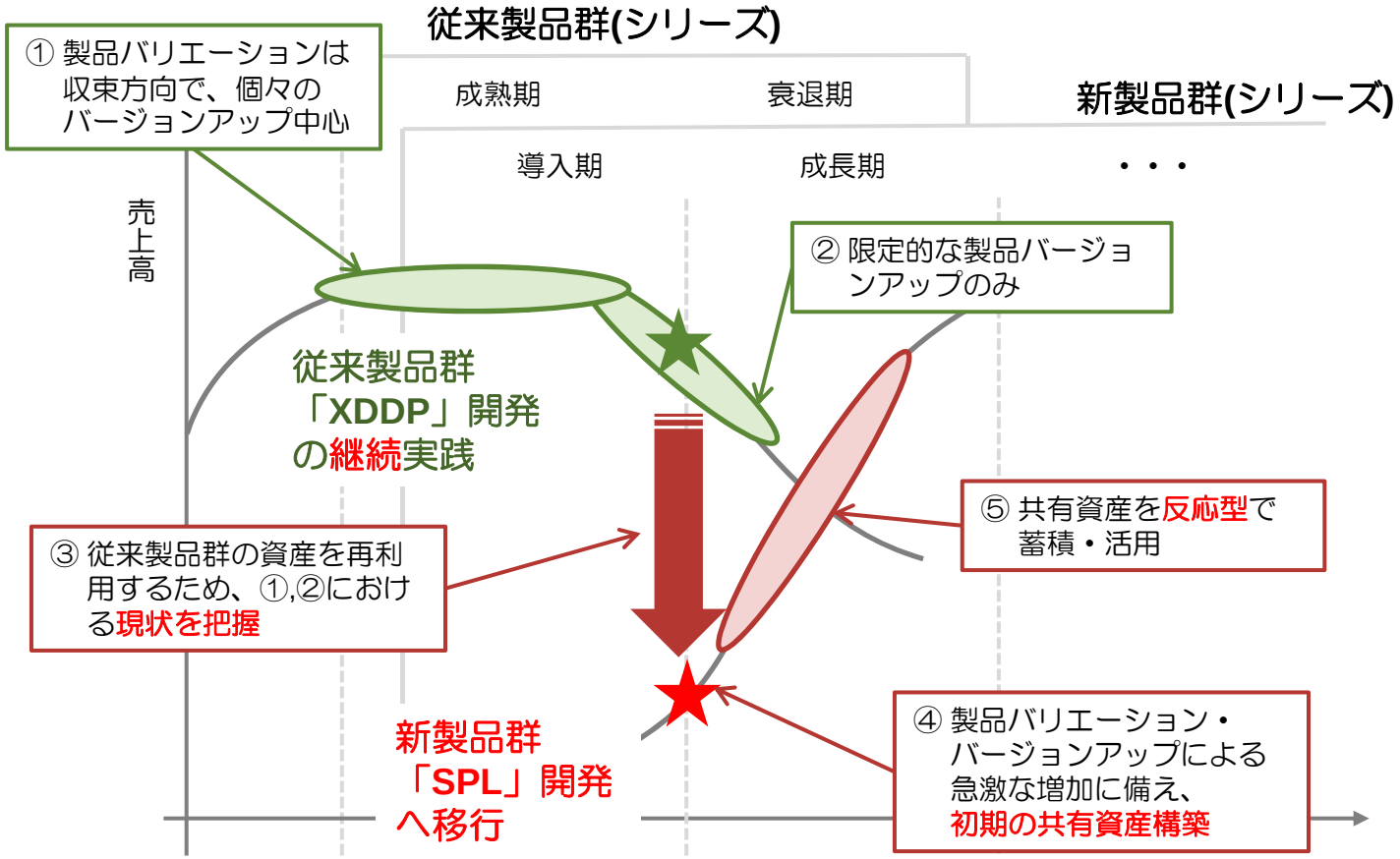
製品群ライフサイクル ～ XDDPからSPLへ移行 ～

従来製品群は「XDDP」開発



製品群ライフサイクル ～ XDDPからSPLへ移行 ～

従来製品群は「XDDP」継続、新たな製品群は「SPL」へ移行

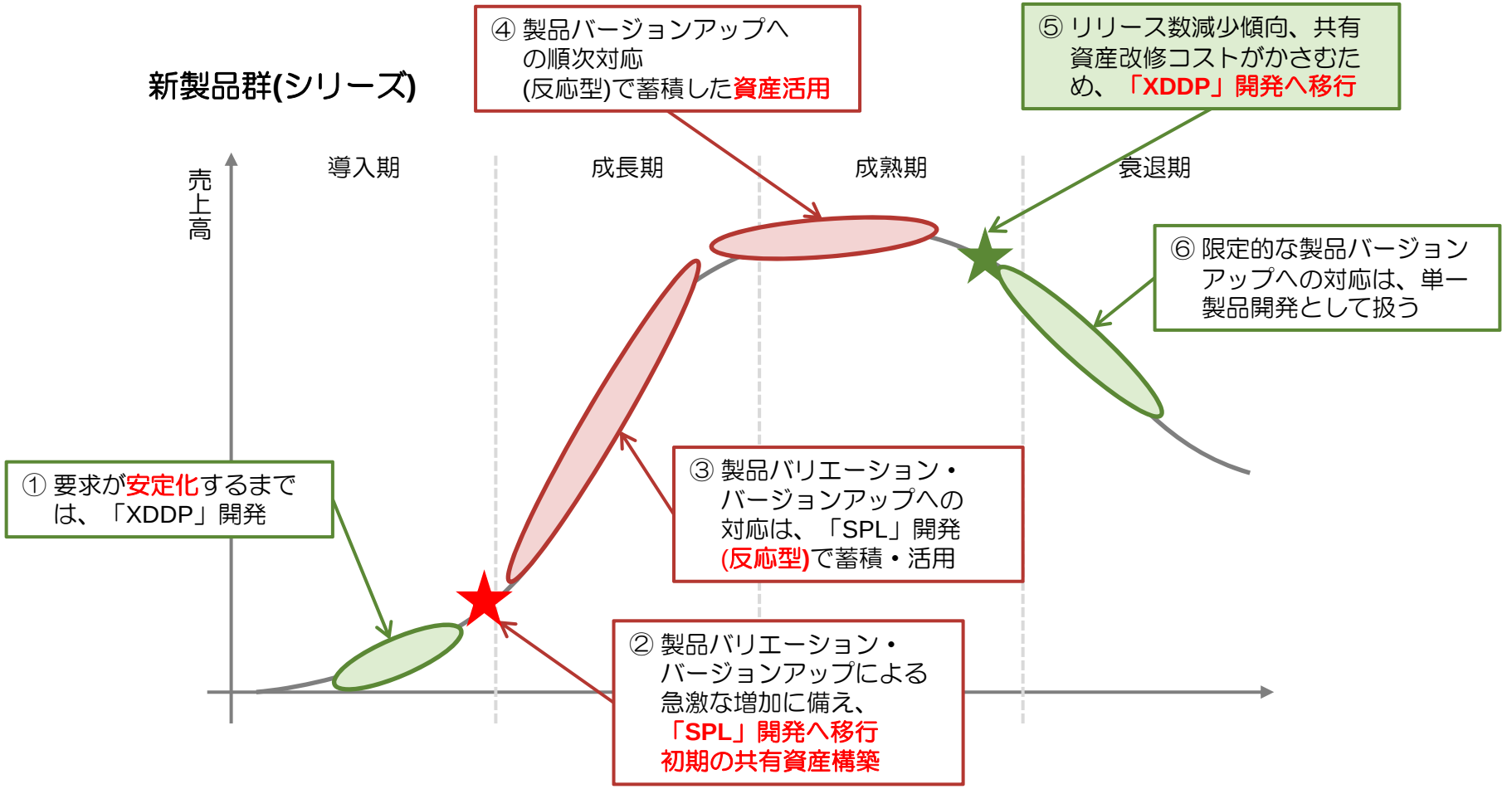


今後の取組み

～ SPL移行後を考える ～

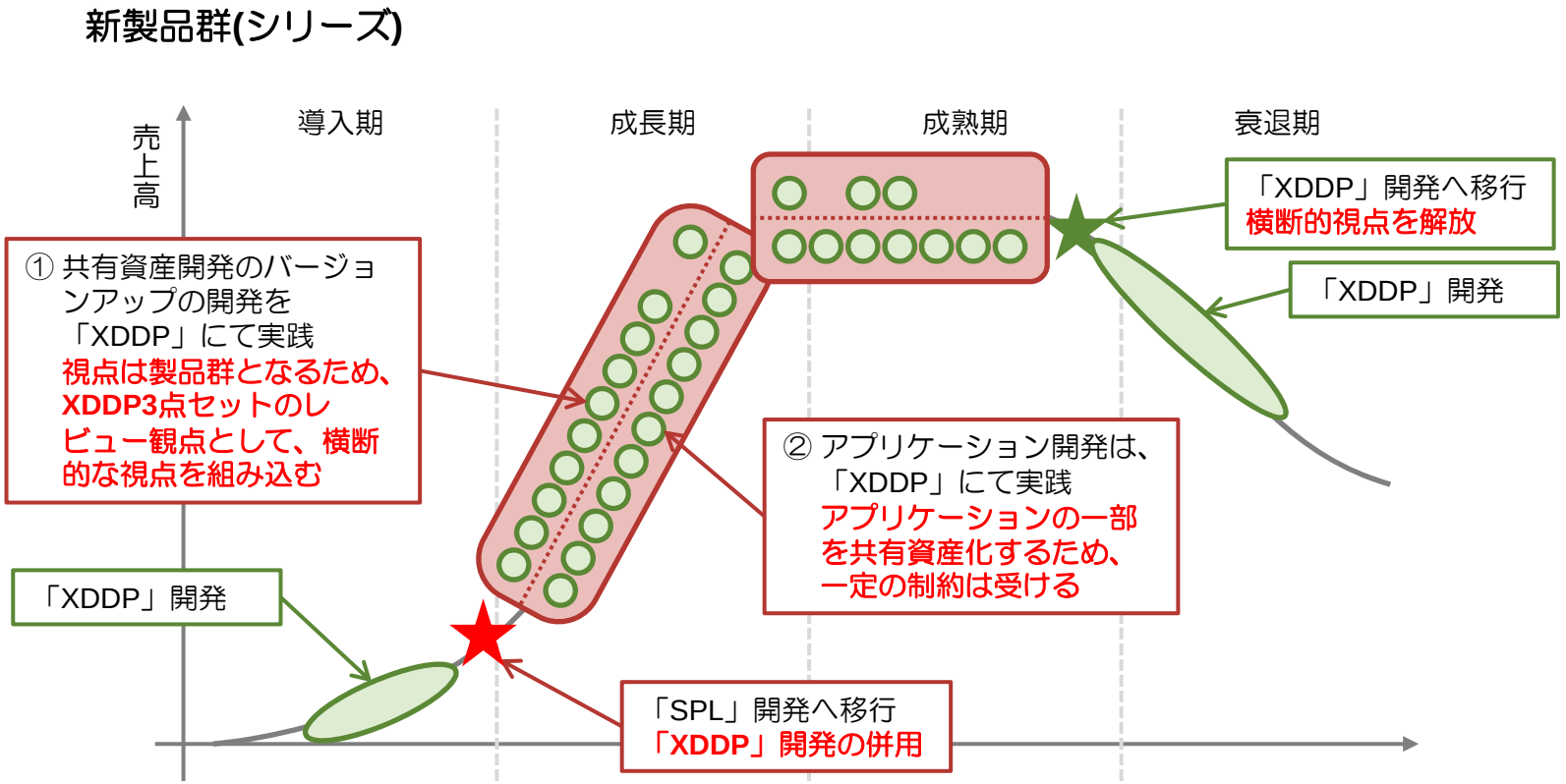
「SPLからXDDPへ逆移行」について…

製品リリースも減少傾向、共有資産改修コストがかさむなどの
兆候から「XDDP」へ逆移行するタイミングもあるはず…



「XDDP/SPLハイブリッド型」について…

SPL移行後も、
- 個々のアプリケーション開発では、「XDDP」を使えるはず…
- 共有資産開発のバージョンアップ開発でも、「XDDP」を使えるはず…



ご清聴ありがとう
ございました