

# XDDP と SPLE の 連携・移行・使い分け ガイドの紹介

2014-06-06

派生開発推進協議会

T14研究会

## — 目 次 —

1. ガイド作成に至った経緯
2. ガイドの予告編
3. 移行
4. 使い分け
5. 融合
6. まとめ

# 1. ガイド作成に至った経緯

# 1.1 ガイドの背景

- XDDP (eXtreme Derivative Development Process) は単発の派生開発に有効な手法であり改造時の品質向上が期待できる。
- 一方、ソフトウェアプロダクトライン開発 (SPLE) は、製品系列で長期間に渡って派生を繰り返す場合の効率向上を図る目的で導入する組織が増えてきている。
- T-14研究会では、XDDP と SPLE のそれぞれが適している条件の検討や XDDP から SPLE へ移行する際の留意点などを議論してきた。
- 議論を進める中で、課題や対応策などを広く参考にしてもらおうべく「XDDP から SPLE への移行ガイド」を作成することとした。
- 移行ガイドを作成するに当たって、移行の出発点を想定するため、派生開発の状況を調べるアンケートを実施した。
- アンケート結果から、XDDP から SPLE への移行だけでなく XDDP と SPLE の連携や住み分けといったことの必要性も、わかったが……

## 1.2 アンケートの概要

### ■ 派生開発(XDDP 開発以外も)を行っている方が対象

- 派生開発カンファレンス2013参加者
- 派生開発推進協議会会員

### ■ 20問の質問

アンケートは無記名

- 選択17問、記述3問

### ■ アンケート期間

- 2013/5/24～11/12

- ✳ 派生開発カンファレンス2013のポスターで案内
- ✳ 6/15 派生開発カンファレンス2013の開催報告に案内
- ✳ 9/10 派生開発推進協議会内メーリングリストでお願い

### ■ 回答数は20件

回答数が少ないため、少々強引に分析

- 全ての回答は派生開発推進協議会会員からと推定

※アンケート結果は2013/11/19の第4回 アフォード・フォーラムで報告

[http://www.xddp.jp/forum2013\\_04.shtml](http://www.xddp.jp/forum2013_04.shtml)

結果報告→[http://www.xddp.jp/forum/2013\\_04/forum2013\\_04\\_survey\\_stats.pdf](http://www.xddp.jp/forum/2013_04/forum2013_04_survey_stats.pdf)

問	質問内容	結果概要
1	どのように派生開発をしていますか？	決まったやり方は無いが過半数 (XDDP は4件)
2	開発しているソフトウェアの全体規模 (Lines of Code) はどの位ですか？	100K 以下が過半数
3	プログラミング言語は何ですか? 1, 1	C と C++ が過半数
4	ソフトウェアの開発期間はどの位ですか？	6ヶ月以下が過半数
5	これまで、どれ位の製品の派生開発をしてきましたか？	5製品以上が過半数
6	完成されたソフトウェアに対して、変更された部分はどの位ですか？	30%以下が過半数
7	完成されたソフトウェアは、元のソフトウェアに対して追加はどの位ですか？	30%以下が過半数
8	開発毎に設計ドキュメントを更新していますか？	更新しているは半数 (XDDP 4件は100%更新)
9	要求をどのように記述していますか？	自然言語記述が半数 (形式化された記述は1/4位)
10	ソフトウェア構造はどのような設計図で説明されていますか？	構造図が過半数
11	ソフトウェアの振る舞いはどのような設計図で説明されていますか？	シーケンス図、状態遷移図が過半数
12	要求～設計～実装のトレーサビリティをどの様に確保していますか？	3/4はトレーサビリティ未確保 (XDDP 4件は要求TMで確保)
13	ソースコードの保守性をどのように確保していますか？	コーディング規約と静的解析は半数以上に普及
14	コンパイルスイッチは、どのように使われていますか？	使われ方は問題が多い
15	構成管理ツールをどのように使っていますか？	適切に使われている
16	SPLEの導入を検討したことはありますか？	SPLE の導入検討はほとんど行われていない
17	問16 で SPLE の導入を検討したが導入にいたらなかったと回答された場合は、その理由を記入して下さい。	1件のみ回答→(体制が難しいため)
18	SPLE を導入するとした場合、一時的に派生開発の開発工数より大きな投資は可能ですか？	いいえが過半数、条件によるが半数弱
19	問18 の回答の理由を記入して下さい。	次ページ
20	派生開発で困っていることがありましたら困っていることを記入して下さい。	次々ページ

## 1.4 問19の回答

問18(SPLEを導入で派生開発より大きな投資は可能?)の回答の理由を記入して下さい  
 [Nos. 3, 11, 15, 20 は XDDP 開発、No. 9 はエンタープライズ開発]

No.	回 答	回答から推定した課題
2	私は、スタッフ部門です。開発プロセスの改善活動をしています。現在の設計スタイルは派生開発(広義)だと思っているが、XDDP(派生開発)を意識したことがないと思う。	XDDP の理解不足
3	工期は難しい。人工でカバーという面では可能かも。	マネジメント欠如
4	導入効果が不明	PL スコープ無し
8	当社では組織として USDM の導入できていない。つまり要求を決めて開発できるレベルに至っていない。この状態で SPLE を導入することは不可能でと考える。開発手法からは XDDP と SPLE は近い位置にあると思うが、要求からはやはり USDM ではないか? どれも理解が足りないのでの的外れかも知れない。	XDDP の理解不足
9	エンタープライズの受託開発であるため、SPLE を導入する欲求が乏しいため。	PL スコープ無し
10	導入する目的で変わってくるから	PL スコープ無し
11	派遣ソフト技術者であるため。	開発形態の制約
12	客先の H/W 開発のサイクルが早く、人員が不足している慢性的な状態であるため、一時的な大きな投資は難しい。	開発形態の制約
15	開発予算の枠内であれば可能	PL スコープ無し
17	受注元との共同開発の側面が強いため 自社のみでの活動は開発のコスト・期間の範囲でしか行えないまた、受注元は開発スタイルについての問題意識が薄い	開発形態の制約
18	開発の商流が2次請けとなることが多いため、1次との受注条件がそのまま制約条件になることが多いため。	開発形態の制約
19	現在慢性的な人手不足の状態のため。	マネジメント欠如
20	現状では難しいが、トップが必要と感じれば、投資は可能となると考える。	PL スコープ無し

派生開発で困っていることがありましたら、困っていることを記入して下さい。

[Nos. 3, 20 は XDDP 開発、No. 9 はエンタープライズ開発]

No.	回 答	回答から推定した解決策
2	開発の形態は、かなりの領域で派生開発だと認識していますが、具体的な開発手法をとってはいない。部門まちまちな開発手法。今後派生開発の開発プロセスを社内に紹介、展開している段階です。	正しい XDDP の導入
3	大きなシステムでの TM の活用	TMの階層化はしてるとして、 要求の上手な分解⇒※FODA の採用～(トレーサビリティ管理ツールの採用)
8	社内勉強会で利用可能な、若手にもその効果がわかるテキストがほしい。	AFFORDD 勉強会テキストの活用
9	ドキュメントとソースコードの乖離が時間とともに広がっていく点	正しい XDDP の導入
12	問19 と重複しますが、開発物件が山積みで仮に XDDP を使用したとしても、後で正規のドキュメントをメンテナンスする時間を取るまでに、次の開発が始まるため、正規のドキュメントのメンテナンスがされないのではないかと心配されています。 問2 についてですが、100K～500K が選択になかったので、10K～100K、問8 についてですが、「開発毎にドキュメントは更新している」と言う選択支がなかったため、異なった選択をしています。	正しい XDDP の導入
18	他社の SPL 参画時に派生開発を導入するにはどうしたら良いのか?	・SPLE(コア資産部)と XDDP (製品固有部)の住み分け ・コア資産のVer.Upに XDDP 活用
19	現在マンパワーとプロセスで品質を確保している状態にある。そのため定数変更などのルーチ的な作業なら良いが新規にソフトを追加したりすると不具合が出やすくなる。	正しい XDDP の導入
20	影響範囲(チェンジインパクト)を抑えること。テスト(主にシステム～受け入れテスト)の十分性と妥当性の確保。	※FODA の採用～※FOT の採用

※FODA: Feature Oriented Domain Analysis ⇒ <http://www.sei.cmu.edu/reports/90tr021.pdf>

※SPLE の技法

※FOT : Feature Oriented Testin g⇒ <http://www.risec.aist.go.jp/files/events/2012/0910-ja/SSSymposium2012Sep10-kitamura.pdf>

## ■ XDDP だけでは解決できないこと

- 大きなシステムでの TM の活用 [問20 回答 No.3]
- 影響範囲(チェンジインパクト)を抑えること。テスト(主にシステム～受け入れテスト)の十分性と妥当性の確保。 [問20 回答 No.20]

XDDP に SPLE や他のエンジニアリングの技法を付加して解決

XDDP と SPLE の連携

## ■ SPLE と XDDP の併用

- 他社の SP L 参画時に派生開発を導入するにはどうしたら良いのか [問20 回答 No.18]

[他社の SPLE は出来ているとして]  
XDDP と SPLE を  
適材適所に適用して解決

XDDP と SPLE の住み分け

移行だけでなく連携や住み分けも考える

という報告をフォーラムで行ったが、  
連携って何？住み分けって何？  
住み分けは使い分けと同じ？？？トイ  
フォーラム以降の研究会で議論が続く、……

## 1.7 アンケートの主たる目的である移行の出発点

以下の表のNo.1～13の状態を出発点と想定

No.	良い状態 [問1,8,9,10,11,12,15の結果から]
1	XDDP による派生開発が適正に行われている/トレーサビリティが確保されている
2	満足な設計ドキュメントが作成され最新になっている
3	ソースコードの保守性がチェックされている
4	構成管理ツールが適正に使われている
<b>正しい XDDP 開発が出来ている</b>	
No.	改善が必要な状態(課題) [問13, 14 の結果から]
5	ソースコードの保守性が適正に確保されているかどうかわからない
6	コンパイルスイッチが適正に使われているかどうかわからない
No.	プロファイル [問2, 3, 4, 5, 6, 7, 16, 18, 19 の結果から]
7	開発規模は比較的小さい(100KLOC 以下)
8	プログラミング言語はC, C++
9	開発期間は短い(6ヶ月以下)
10	5製品以上の派生開発を行ってきた
11	派生開発のソフトウェアの変更量は30%以下
12	派生開発のソフトウェアの追加量は30%以下
13	SPLE の導入は検討したことが無い/SPLE への投資は条件次第
<b>SPLE の効果を得やすい条件</b>	

No.	ガイド化項目	移行	連携	住み分け
1	過去製品のフィーチャー分析(FODA:フィーチャーモデリング)	✓	✓	
2	フィーチャー分析からのプロダクトラインスコーピング	✓		
3	プロダクトラインスコーピングからの投資対効果判断	✓		
4	フィーチャーモデルとUSDM(要求TM)の対応付け	✓	✓	✓
5	USDM(要求TM)の共通性と可変性の対応	✓	✓	✓
6	可変性の実現方法(コンパイルスイッチの使い方、他)	✓	✓	
7	フィーチャー分析からの品質シナリオ抽出	✓		
8	品質シナリオによるアーキテクチャの検証(ATAM)	✓		
9	アーキテクチャ再構築(ADD)[必要なら]	✓		
10	ソースコードのリファクタリング[必要なら]	✓		
11	製品固有部の設計ガイド	✓	✓	✓
12	フィーチャーモデルとテストの対応付け(FOT)	✓	✓	
13	テストケースの自動生成ツールの採用(例えばPair-wise法)	✓	✓	

アンケート結果からは技法視点で  
ガイド化項目を抽出してみたが、  
プラクティスとして表現したほうが良い！！  
移行のパターンも示したほうが良い！！  
これまた、フォーラム以降の研究会で議論が続く、、、

## 2. ガイドの予告編

- 当ガイドの背景
  - このガイドを作成した動機
  - 発端となったアンケート集計結果の概説
- 当ガイドの使い方
- 想定読者
  - 技術者、プロセス支援者(SEPG, SQA, etc)、技術のわかるマネジャ
  - XDDP はできていること
  - SPLE に関する基本的な知識があること
- XDDP と SPLE
  - それぞれの概要
  - それらの違い
  - 導入/移行のシナリオ
- プラクティス(仮題)
  - フィーチャ分析とその成果の扱い
  - 過去製品の分析
  - 共通性・可変性の扱い
  - リファクタリング  
(他のプラクティスも検討中)

- プラクティスの適用
- 移行、使い分け、融合
  - それぞれの違い
    - 移行、融合の際に必須のこととそうでないこと
  - プラクティスの適用
    - 移行
    - 使い分け
    - 融合
  - プラクティスの組合せパターンか、それが難しくければ組合せ例(できれば)
- 結言
- 付録
  - (アンケート集計結果、これまで作成した資料、等)

「連携」よりもこの用語の方が適切と判断し、変更

## 【移行】

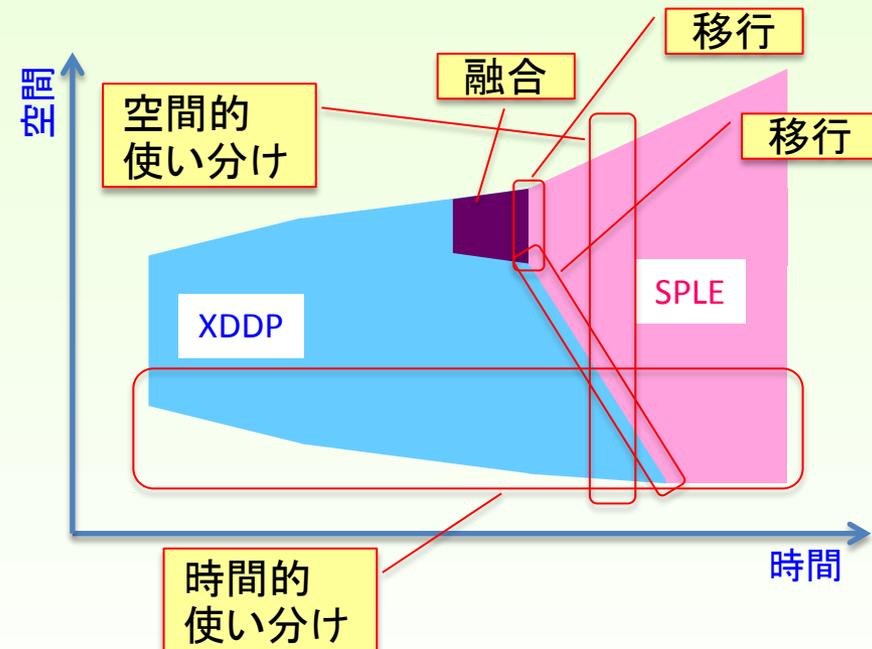
- 同一システム/製品のシリーズに対して、XDDP (SPLE) のプロセスでの開発していたのを、SPLE (XDDP) のプロセスでの開発に切替える
  - XDDP → SPLE または SPLE → XDDP
- 形態には
  - 全面的な切替
  - 部分的な切替
 があり、後者では次の「使い分け」が生じる

## 【使い分け】

- 同じシステム/製品の中で、XDDP と SPLE を別々の部分に適用する
  - 「移行」においてこのパターンが現われることがある
- 極端な例として、どちらか一方しか適用しない場合もある
  - 途中で条件が変わり、「移行」を実施することはありうる

## 【融合】

- 基本的には XDDP を実施するが、SPLE に使う技術を応用してより良い開発を行う
- SPLE に使う技術 = 可変性を扱う技術
  - 可変性モデリング(フィーチャモデリング)
  - 可変性モデルと資産の関連付け
  - 可変性の実現(サブクラス化、設定ファイル、モジュールの入れ換え、等)



# 3. 移行

## ■ソフトウェアプロダクトライン(SPLE):成熟期に適した開発方法

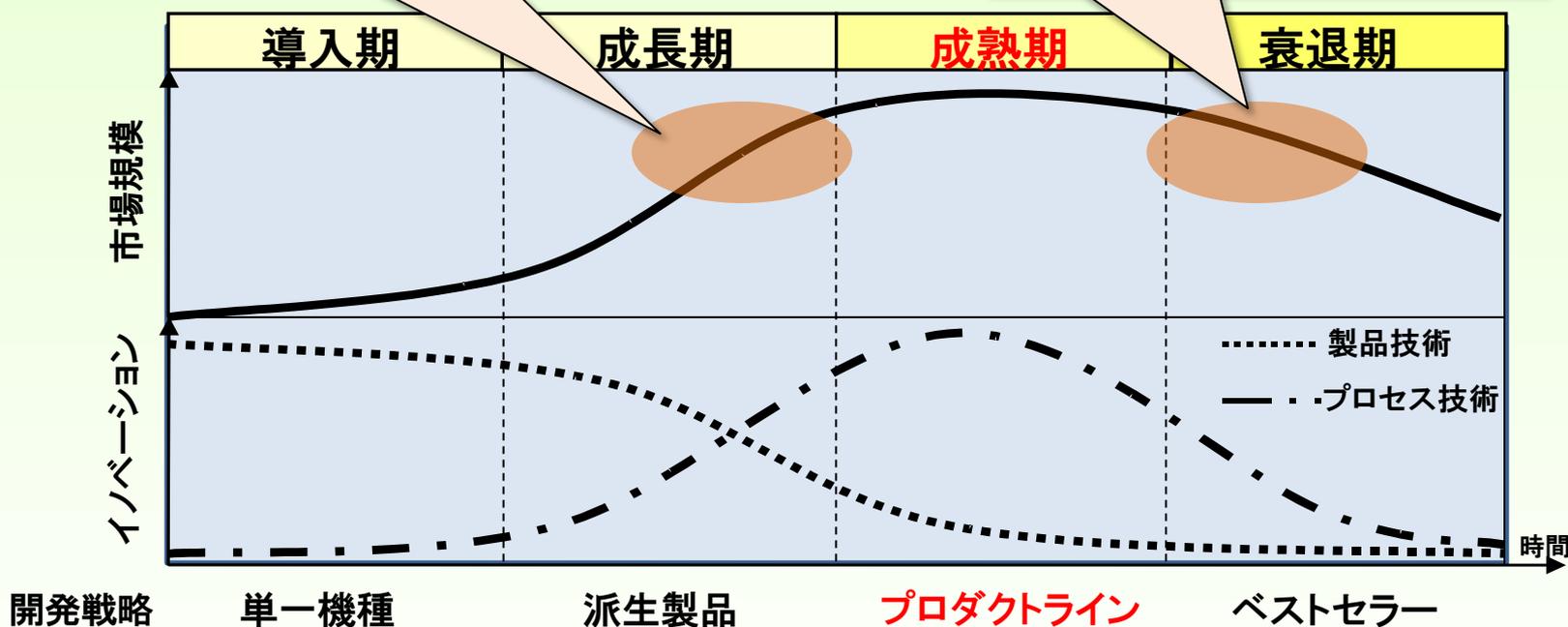
⇒導入期→成長期→成熟期→衰退期と移行する「製品ライフサイクル」に対応した移行シナリオを想定する

### ■移行シナリオ1:

XDDP で派生開発をしている組織が、SPLE に切替える

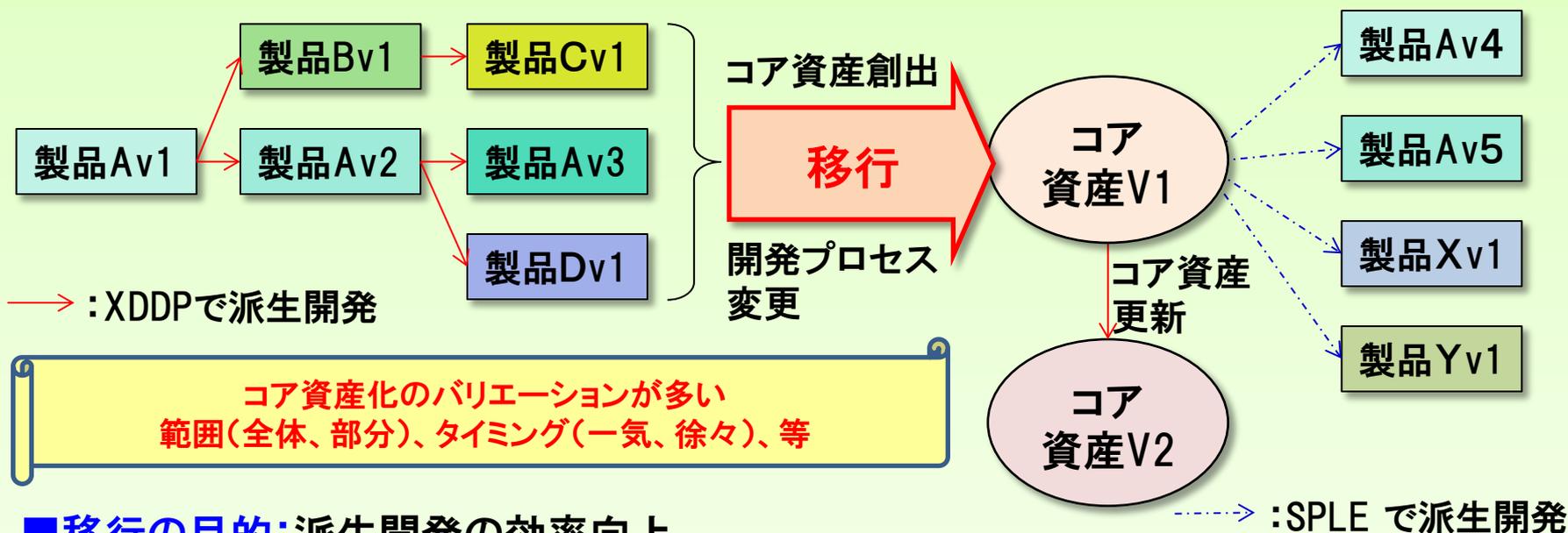
### ■移行シナリオ2:

SPLE で派生開発をしている組織が、XDDP に切替える



参考文献: 吉村, 菊野, 組込みシステムにおけるソフトウェアプロダクトラインの導入, 情報処理, Vol. 50, No. 4, pp.295-302, 2009.

## ■移行シナリオ1: XDDP で派生開発をしている組織が、開発方法を SPLE に切替える



■移行の目的: 派生開発の効率向上

■背景: 製品系列のバリエーション、派生開発件数の増加に伴う開発コスト増大

■課題: ①既存資産を基に SPLE コア資産の開発方法検討

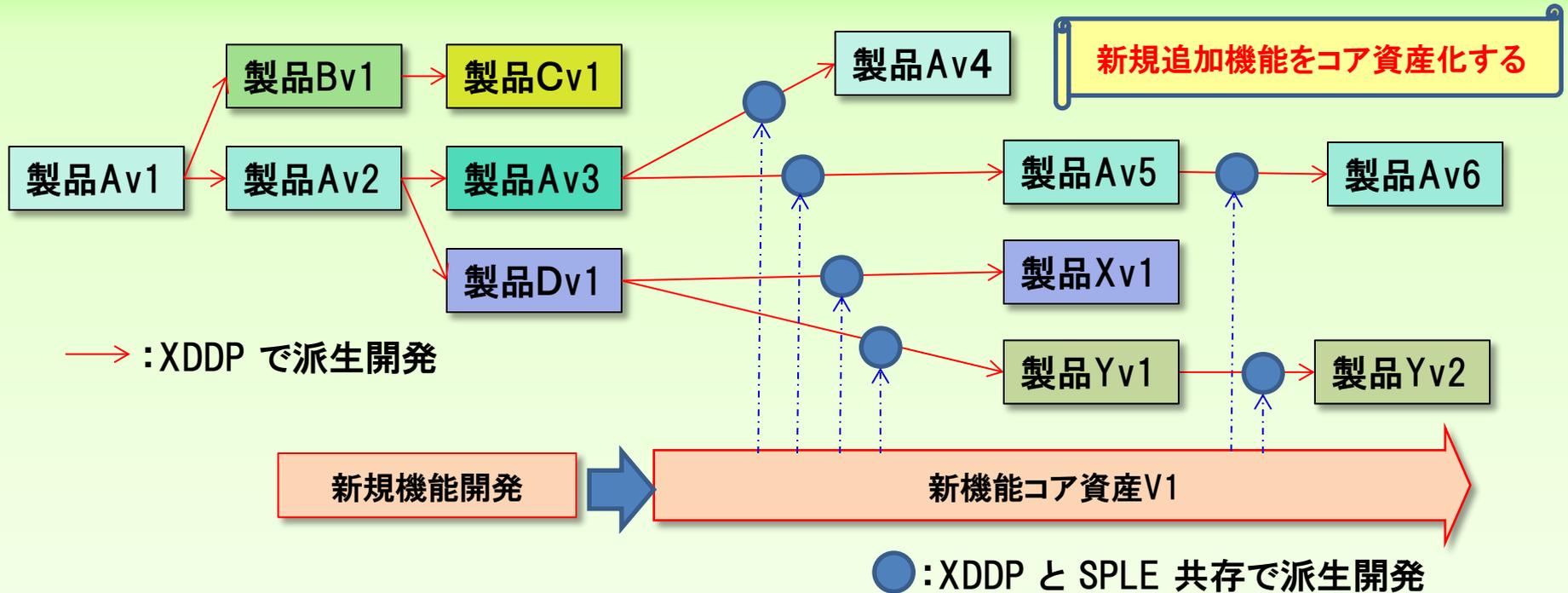
⇒現状資産の状態や組織等の状況に適した SPLE の開発方法を採用する

②コア資産の可変性を実現するためのアーキテクチャ検討

③移行の際に XDDP の成果物を活用する方法の確立

⇒過去の XDDP 成果物(変更要求トレーサビリティマトリクス)をコア資産化の際の検討材料として使用する、等

■ 移行シナリオ1-①: XDDP で派生開発をしている組織が、**一部を SPLE に切替える**



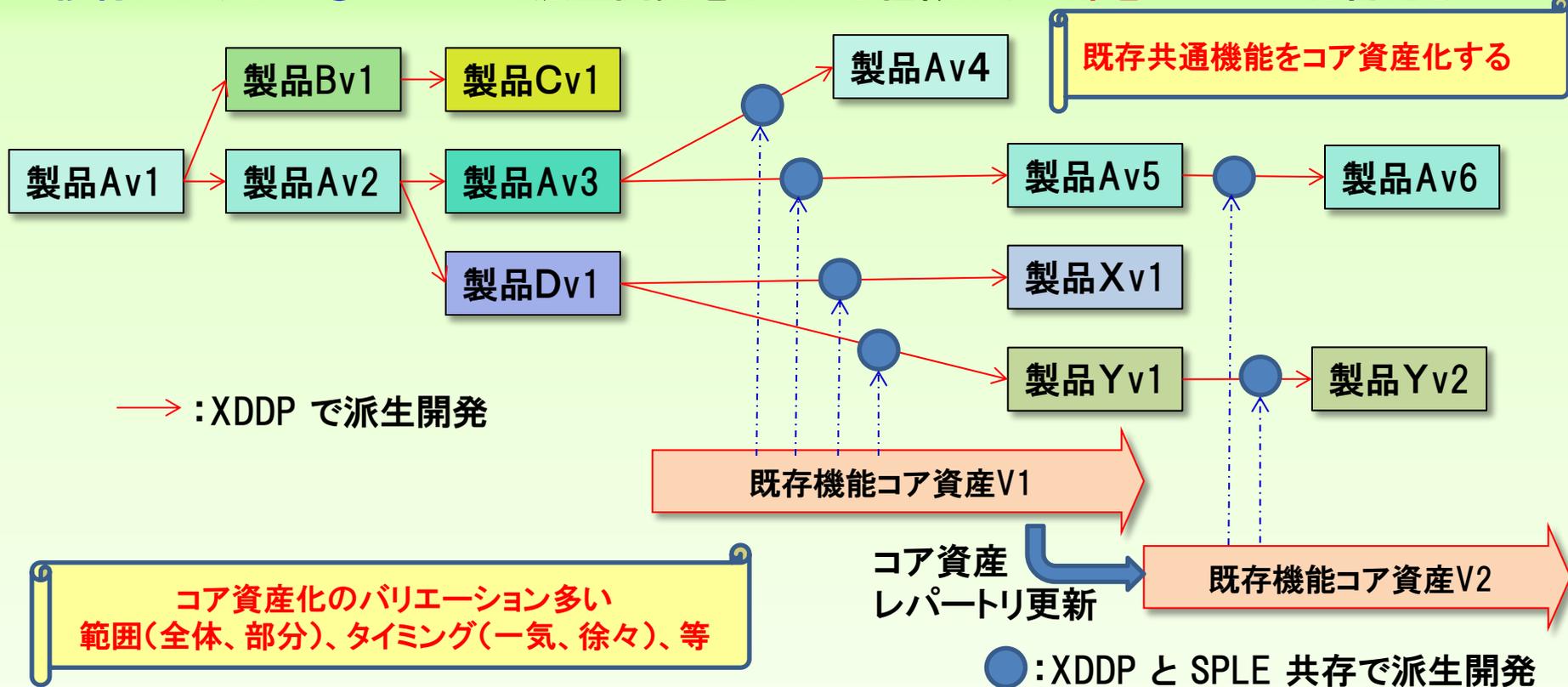
■ 移行の目的: 既存製品への新規追加フィーチャのコア資産化による効率向上

■ 背景: 既存製品群共通の新規機能追加

■ 課題: ① 新規機能をコア資産化(フィーチャ分析実施)  
⇒ 既存製品群へのコア資産適用

② コア資産の可変性を実現するためのアーキテクチャ検討

■ 移行シナリオ1-②: XDDP で派生開発をしている組織が、**一部を SPLE に切替える**



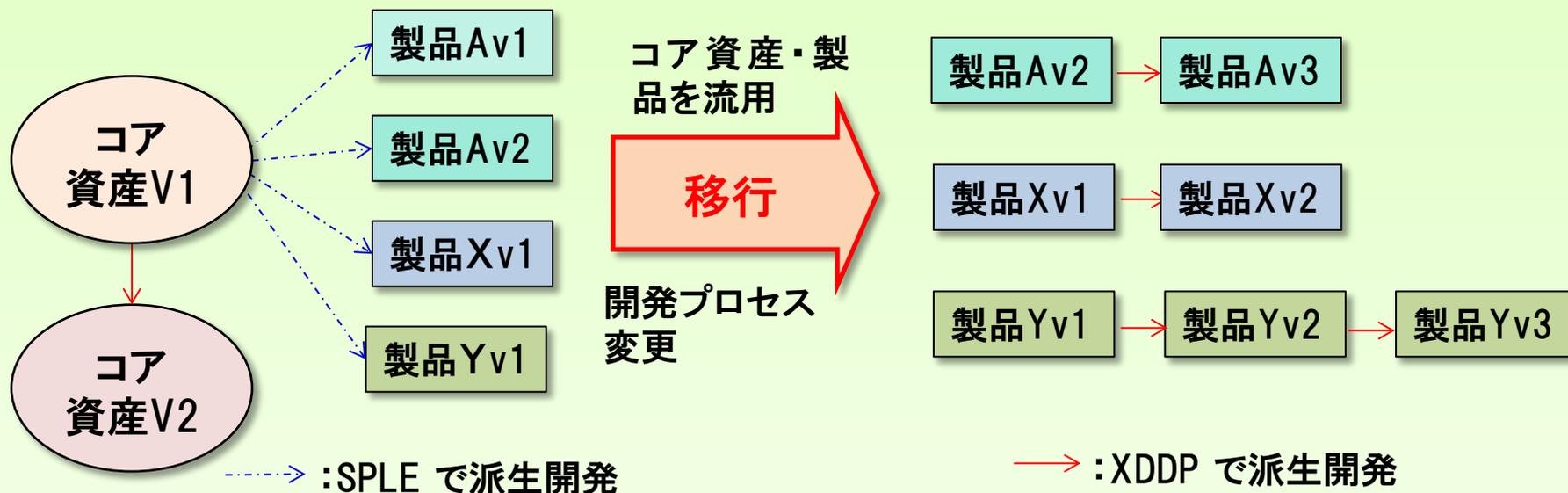
■ 移行の目的: 既存製品の共通フィーチャのコア資産化による効率向上

■ 背景: 既存製品群共通機能の増加および維持・保守効率悪化

■ 課題: ① 既存機能のコア資産化(フィーチャ分析実施)  
⇒ 既存製品群へのコア資産適用

② コア資産の可変性を実現するためのアーキテクチャ検討

## ■移行シナリオ2: SPLE で派生開発をしている組織が、開発方法を XDDP に切替える



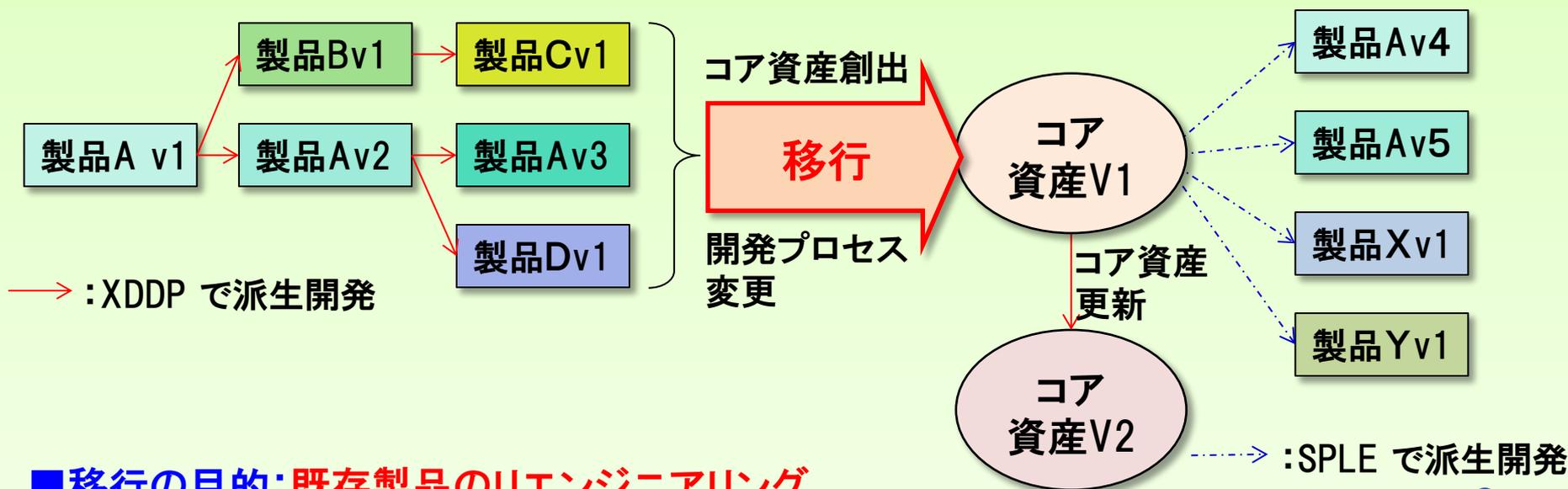
■移行の目的: **開発効率の維持** ← コア資産維持工数の削減

■背景: 製品系列寿命の衰退期への遷移に伴う派生開発件数の減少

■課題: ① SPLE コア資産の運用方法改訂(例: コア資産管理をコア資産管理チームから製品開発チームに移管してしまう)

② SPLE 開発で納入した製品の保守方法確立(担当組織、改訂プロセス、等)

## ■移行シナリオ3: XDDP で派生開発をしている組織が、開発方法を SPLE に切替える



■移行の目的: 既存製品のリエンジニアリング

■背景: 既存資産の複雑化による維持・保守コスト増大

■課題: ①既存資産を基に SPLE コア資産の開発方法検討  
⇒現状資産の状態や組織等の状況に適した SPLE の開発方法を採用する

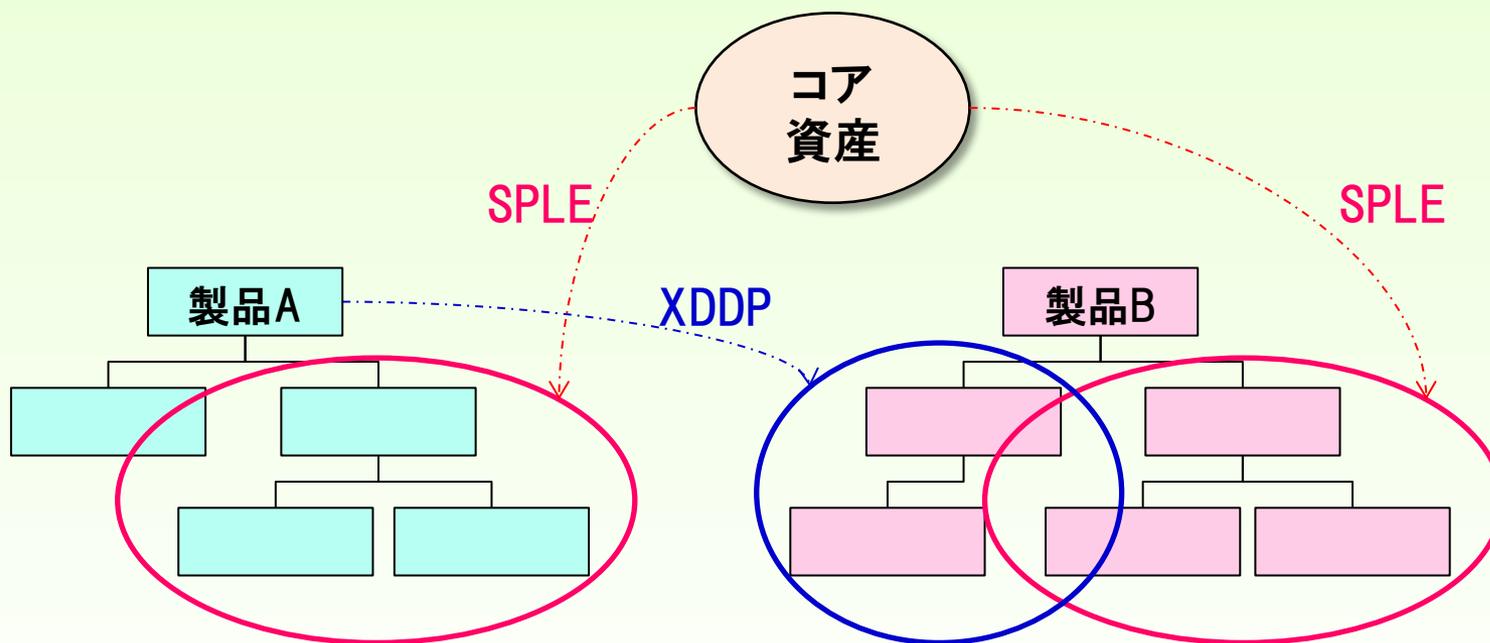
②コア資産の可変性を実現するためのアーキテクチャ検討

③移行の際に XDDP の成果物を活用する方法の確立  
⇒過去の XDDP 成果物(変更要求トレーサビリティマトリクス)をコア資産化の際の検討材料として使用する、等

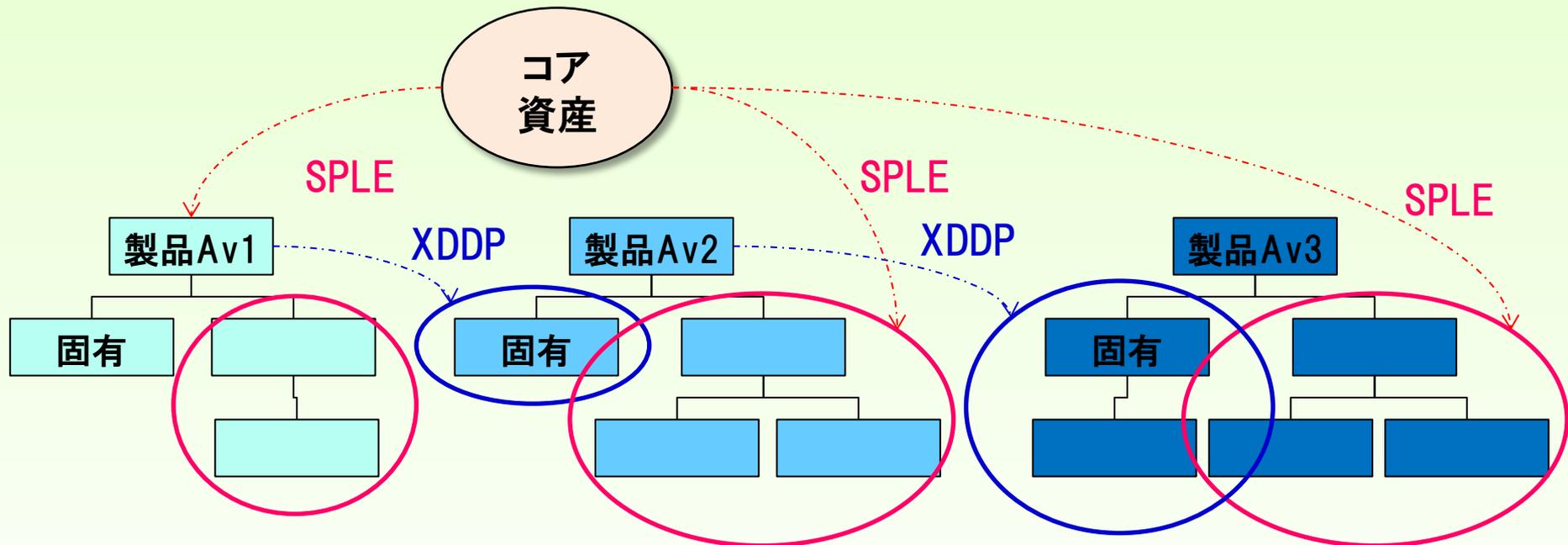
移行シナリオ1 と目的・背景  
が異なる

# 4. 使い分け

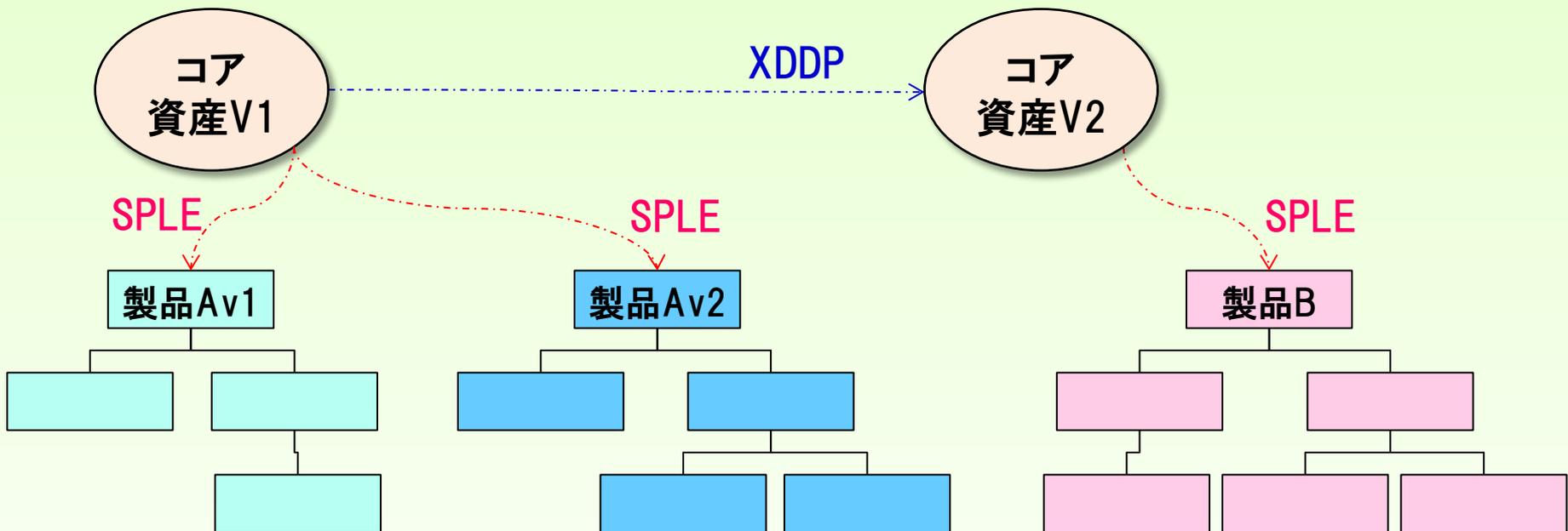
- コア資産による恩恵(効果)が得やすい部分には SPLE を、そうでない部分には XDDP を適用
- コア資産の効果が大きい例: その部分は製品間での共通性が高い、今後開発する内容が比較的明らか、等
- 例えば、GUI は組織全体で Look & Feel を統一するためにコア資産に基づいて構築する



- そのシステム/機種に固有の(他のシステム/機種と共有しない)部分には XDDP を適用し、他の部分に SPLE を適用する

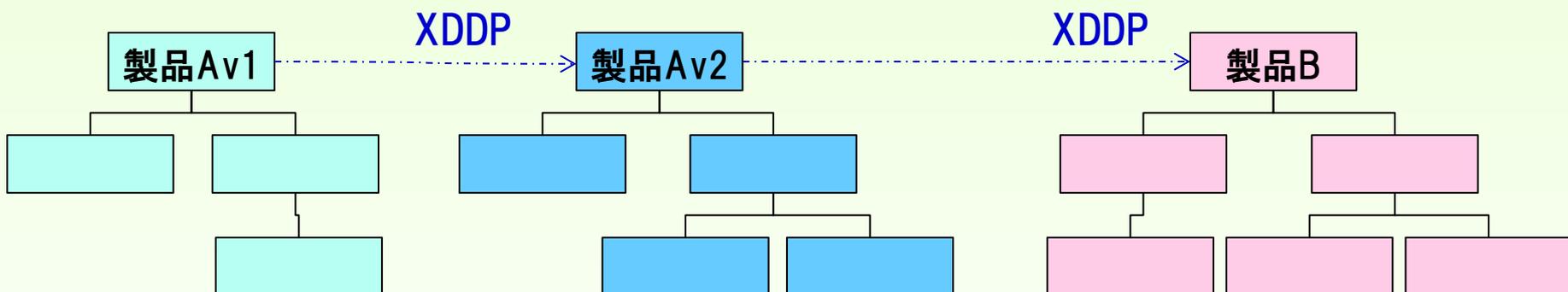


## ■ 各製品の開発には SPLE を、コア資産の進化には XDDP を適用する



■ シリーズ化の予定のない開発、またはシリーズ開発であっても開発のインターバルが長いものでは、SPLE を適用せずに XDDP のみで開発する

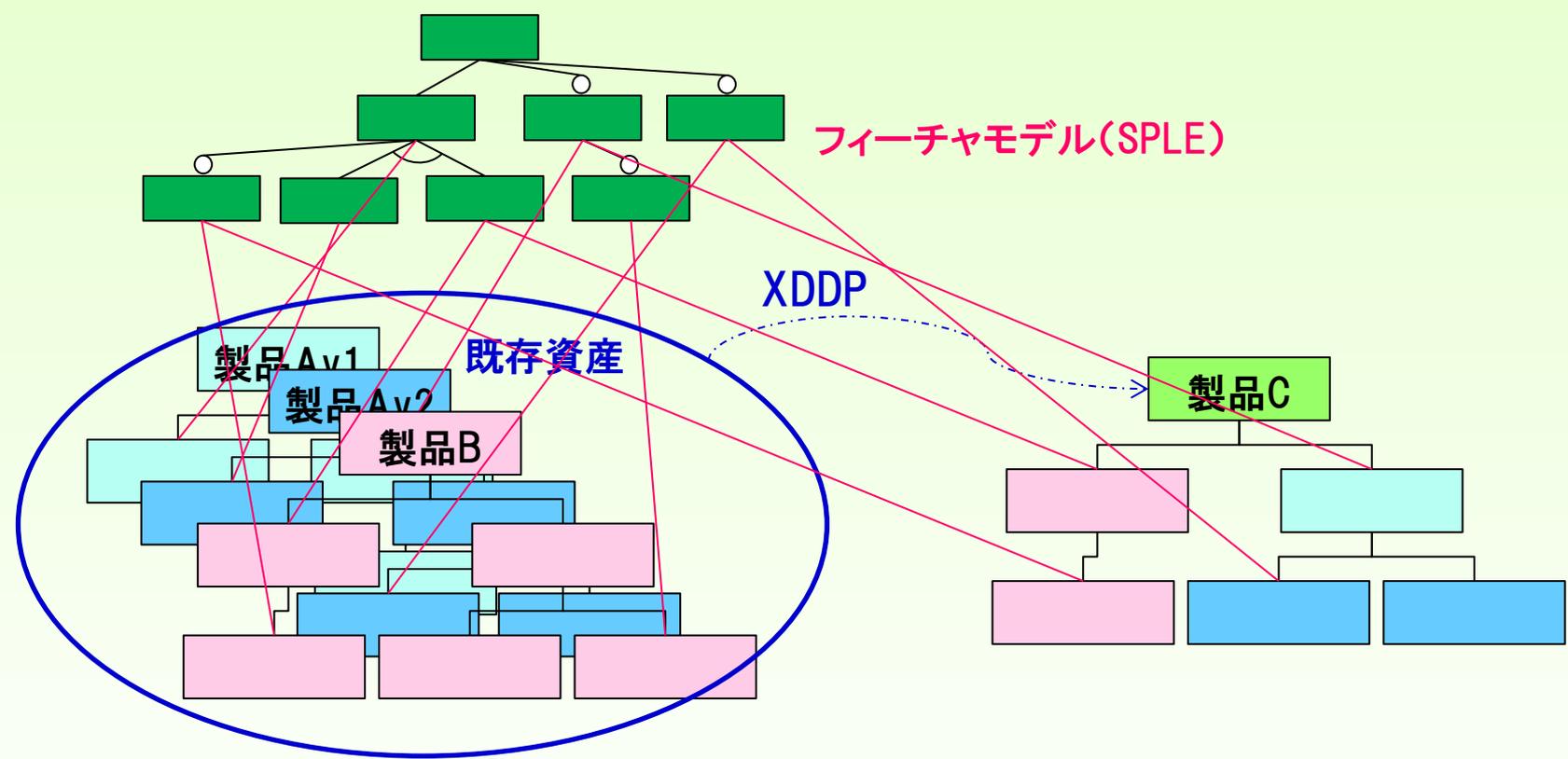
- コア資産による効果が見込めない場合の「使い分け」
- 使い分けの例1 の特別な場合



# 5. 融合

- 「融合」では、SPLE で用いられる技術を XDDP に取り入れて効率・品質のさらなる向上を図る
- SPLE に使う可変性モデルと各種成果物の可変点との間の関連付けは、可変点(どこが変わりうるのか)に関する間接的なトレーサビリティを示すことになる
  - 成果物: 要求仕様、アーキクチャ、詳細設計、実装、テスト仕様・設計、開発プロセス、等
- XDDP では可変点は陽に扱わないが、それぞれの開発で扱ったフィーチャをフィーチャモデルに組み上げ、開発間で共通であった点、異なった点を可変性モデルとして整理することによって、次回以降の開発で最も適切な過去資産(要求仕様の一部、アーキテクチャの一部、他)を見つけるのに役立てることができる

- 既存資産をフィーチャモデルを用いて整理し、次回以降の XDDP 開発のベースを識別する際に役立つ



# 6. まとめ

- XDDP と SPLE は排他的な存在ではなく、適材適所の考え方で適用するもの
  - 時間的な「使い分け」や、空間的な「使い分け」(同時並行)がある
  - 時間的な使い分けには「移行」が伴う
  - 双方のいいとこどりをした「融合」型の適用もある
- ガイドでは XDDP と SPLE に用いるプラクティスも解説する
- ガイドは11月をめどに公開予定
  - [www.xddp.jp](http://www.xddp.jp) からたどれるところに掲載