

派生開発カンファレンス2014

モデル指向SPL開発におけるXDDPの効果的適用

2014.06.06

株式会社 日立産業制御ソリューションズ

業務改革本部 生産技術部

渡辺 滋

XDDP: eXtreme Derivative Development Process

SPL: Software Product Line

モデル指向SPL開発におけるXDDPの効果的適用

Contents

1. 会社概要と業務紹介
2. SPL開発の概念
3. モデル指向SPL開発の概要
4. モデル指向SPL開発へのXDDP適用
5. 効果と課題

モデル指向SPL開発におけるXDDPの効果的適用

1. 会社概要と業務紹介

ミッション

人々の生活を安全・安心で豊かにするため、
優れた技術で**新しい価値**を提供し、
社会の発展に貢献する。

経営ビジョン

情報技術と制御技術の融合により、

- ・ お客さまが直面する課題に、新たな価値で応える
- ・ 最先端の技術と経験で、産業界の成長をけん引する
- ・ 社員一人ひとりが高い志を持ち、挑戦し続ける


産業ソリューション、組込みエンジニアリング、セキュリティ・画像ソリューション、システムエンジニアリングなどの情報制御ソリューション、プリント基板、セキュリティ・コンポーネントの製造事業を展開



設 立	2014年(平成26年) 4月
資本金	30億円
社員数	約4,380名(2014年4月現在)
本 社	茨城本社 茨城県日立市大みか町五丁目1番26号
	東京本社 東京都台東区秋葉原6番1号(秋葉原大栄ビル)
沿 革	(株)日立製作所の産業分野向け情報制御システム事業部門、セキュリティシステム開発・設計事業部門、プリント基板製造事業部門と(株)日立情報制御ソリューションズなど、関連グループ会社5社を統合し、新会社設立

全社の業績アップを目標にソフト生産技術の立場から業務改革に貢献する

主な業務内容

- 開発技術支援
 - 開発プロジェクト支援
 - 開発手法適用支援
 - 開発ツール適用支援
 - 生産実績収集・分析
 - 開発技術教育
 - ソフト生産技術講座
 - 各部要請技術講座、他
 - 社外連携活動
 - 派生開発推進協議会
 - 日立グループ、他
- 
- ```
graph LR; A[開発プロジェクト支援] --- B[モデル指向開発]; A --- C[SPL開発]; A --- D[派生開発XDDP];
```

XDDP: eXtreme Derivative Development Process

SPL: Software Product Line

**モデル指向開発**とは、モデルベース開発とモデル駆動開発を包含した開発手法

**モデル指向開発**  
(MOD: Model Oriented Development)

**モデルベース開発**  
(MBD: Model-Based  
Development)

制御工学

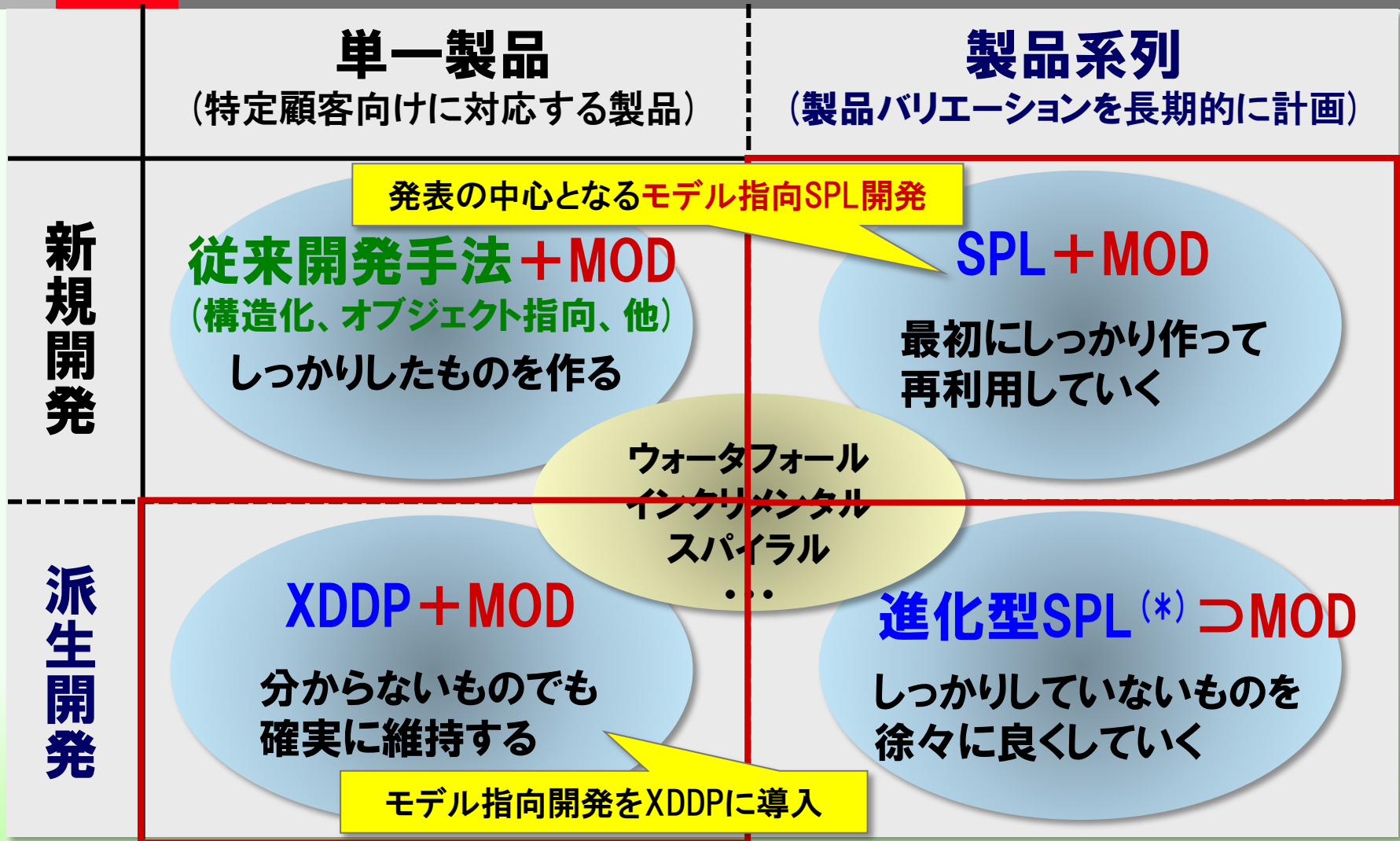
**モデル駆動開発**  
(MDD: Model-Driven  
Development)

ソフトウェア工学

連続系システム

離散系システム

# 1-4 モデル指向開発の適用対象



XDDP: eXtreme Derivative Development Process、(株)システムクリエイツ清水吉男氏が考案した派生開発プロセス

SPL: Software Product Line、MOD: Model Oriented Development (モデル指向開発手法)

(\*)派生開発時にSPLの構造に徐々に近づけていく、当社が独自に研究・考案した手法

# モデル指向SPL開発におけるXDDPの効果的適用

---

## 2. SPL開発の概念

XDDP: eXtreme Derivative Development Process

SPL: Software Product Line



# 2-1 基本概念

**SPL開発とは、同種のソフトウェア製品群におけるソフトウェアの再利用を目的とした戦略的、かつ計画的に継続していく開発**

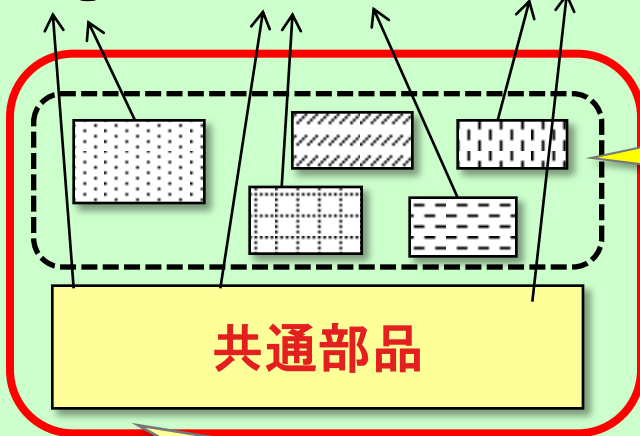
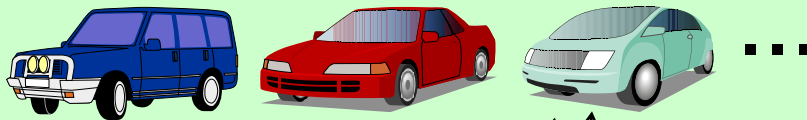


- ① 事前に**共通な部品**と**選択可能な部品**を開発
- ② **部品を再利用**して同種の複数製品を開発

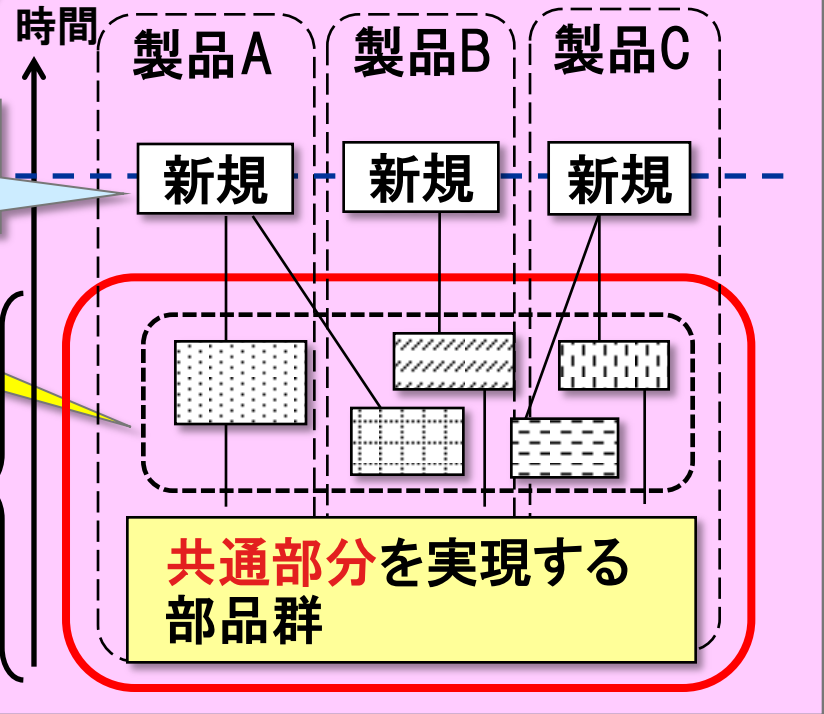
乗用車の開発

類似点比較

SPL開発



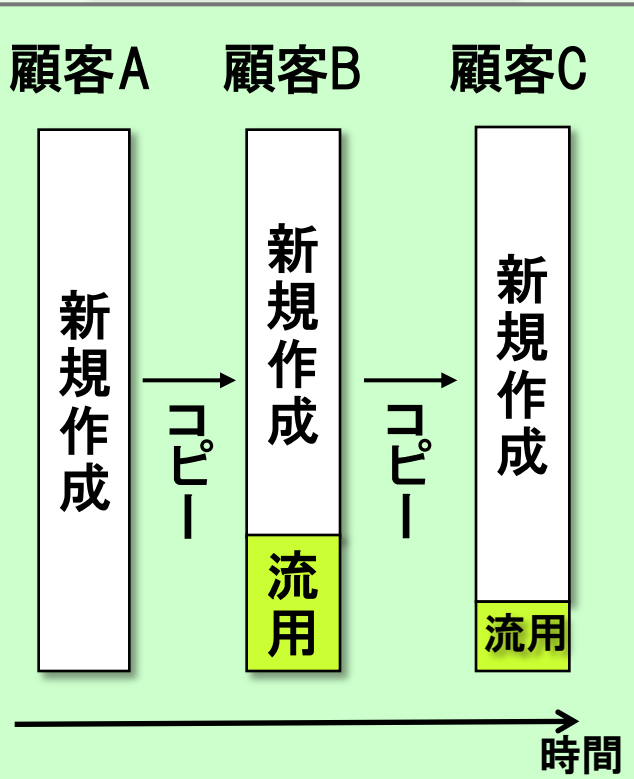
同時開発が可能



フレキシビリティのある資産

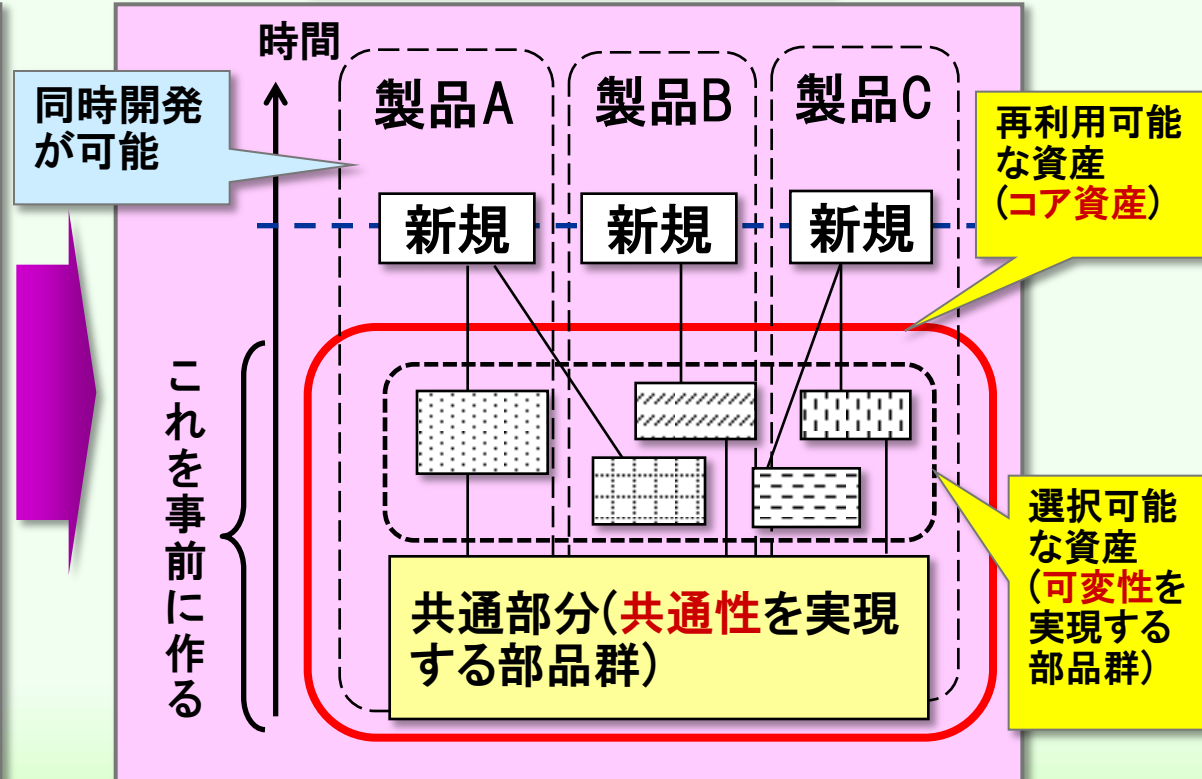
## 2-2 従来の開発方式との比較

### 従来の開発方式



全体コピーで部分的に利用する  
ケースが多く**時間的にはシリアル**  
**な開発**

### SPL開発方式



- ① **製品の可変性**に着目し、製品群で共通に  
または選択して再利用可能なソフトウェア  
資産(「**コア資産**」と呼ぶ)を**事前に開発**
- ② コア資産を利用して製品の**同時開発が可能**

### 製品の可変性とは、

顧客の多様な要求から導出される製品機能に関する変異の可能性

#### ■ 可変性には、可変点と可変部がある

- ・ **可変点**： 製品、または製品群の選択可能な特性
- ・ **可変部**： 選択可能な特性を実現する個々の選択肢

【例】

**可変点**： エンジン

**可変部**：

構造の視点

- ・ レシプロエンジン
- ・ ロータリエンジン
- ・ ロケットエンジン
- ・ タービンエンジン

燃料の視点

- ・ ガソリンエンジン
- ・ ディーゼルエンジン
- ・ 水素燃料エンジン
- ・ 石炭エンジン

#### ■ 外部と内部の可変性がある

- ・ **外部可変性**： **顧客から見える**製品の可変性
- ・ **内部可変性**： **顧客から見えない**製品の可変性

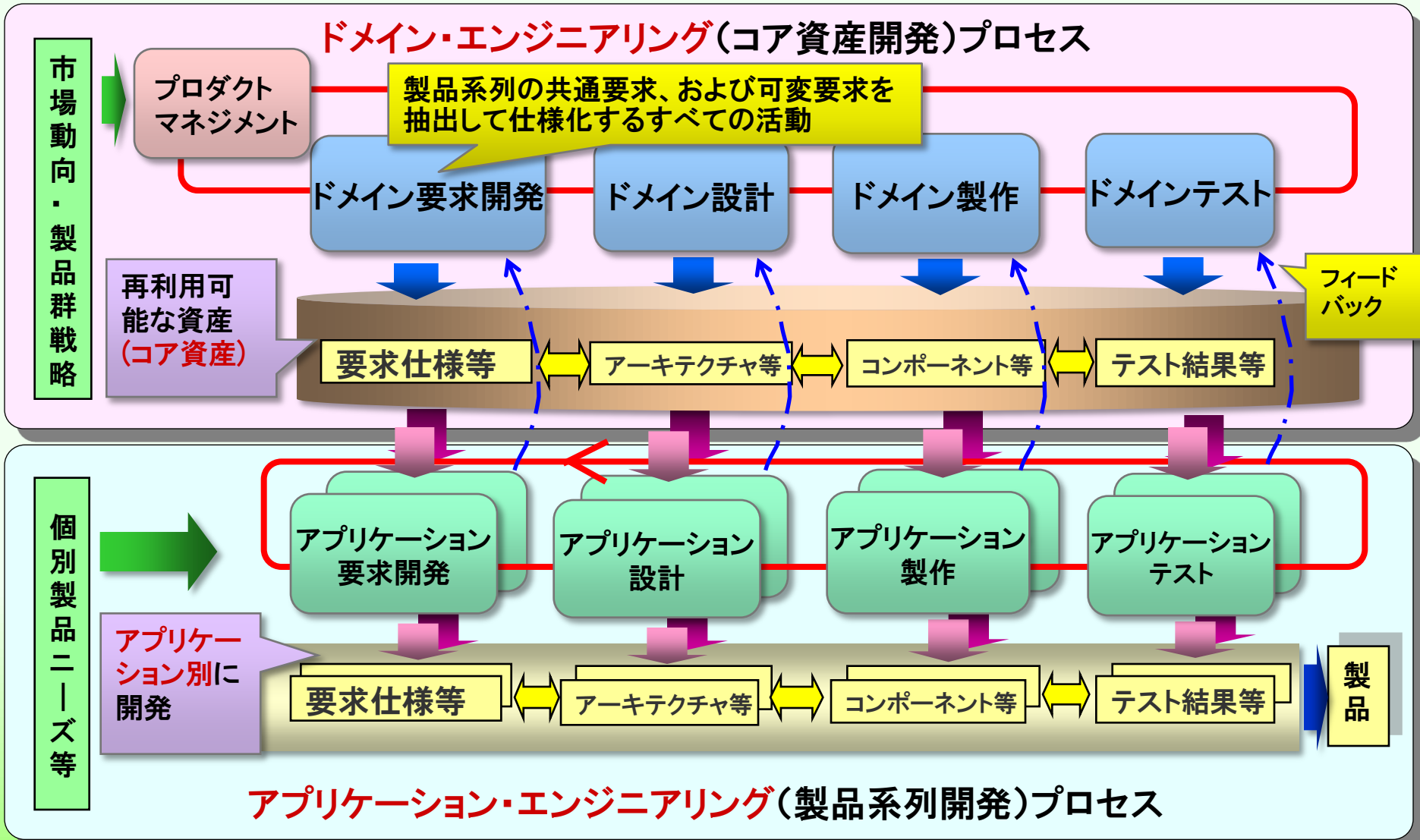
#### ■ 時間と空間の可変性がある

- ・ **時間的**可変性： **時間の経過で異なるニーズ**の製品が存在すること
- ・ **空間的**可変性： **同時期に異なる用途や形態**の製品が存在すること

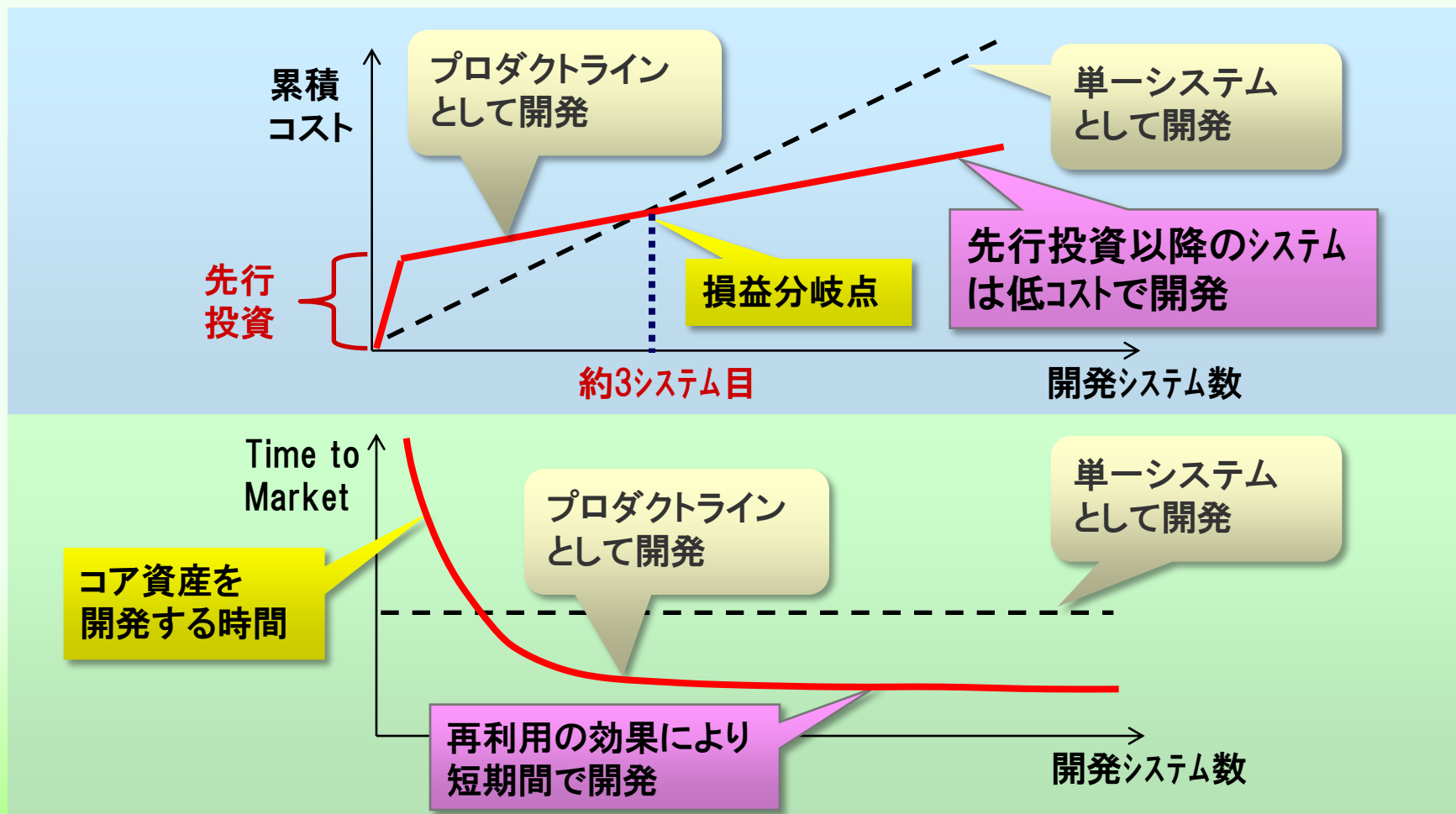
# 2-4 開発プロセス

## SPL開発における2つの開発プロセス

■ ■ ■ :サブプロセス



### SPL開発における2つの効果(開発コスト、およびTime to Marketの比較)



## モデル指向SPL開発におけるXDDPの効果的適用

---

### 3. モデル指向SPL開発の概要

### ● SPL適用上の課題

- (1) SPLは概念的には理解しやすい反面、**進め方の具体的な手順や開発ノウハウ**が示されていない
- (2) **製品機能の可変性をソフトウェアの構造につなげる方法**について確立されていない

### ● 課題解決のアプローチ [SPL開発にモデリング手法を導入]

**SPL + モデル指向開発 = モデル指向SPL開発**

- ① **顧客の視点**で**製品機能の可変性**に着目して**再利用**の単位を捉え、**フィーチャモデリング**で可変性をソフトウェア要求にマッピングする

フィーチャとは、顧客から見える製品機能(外部機能)

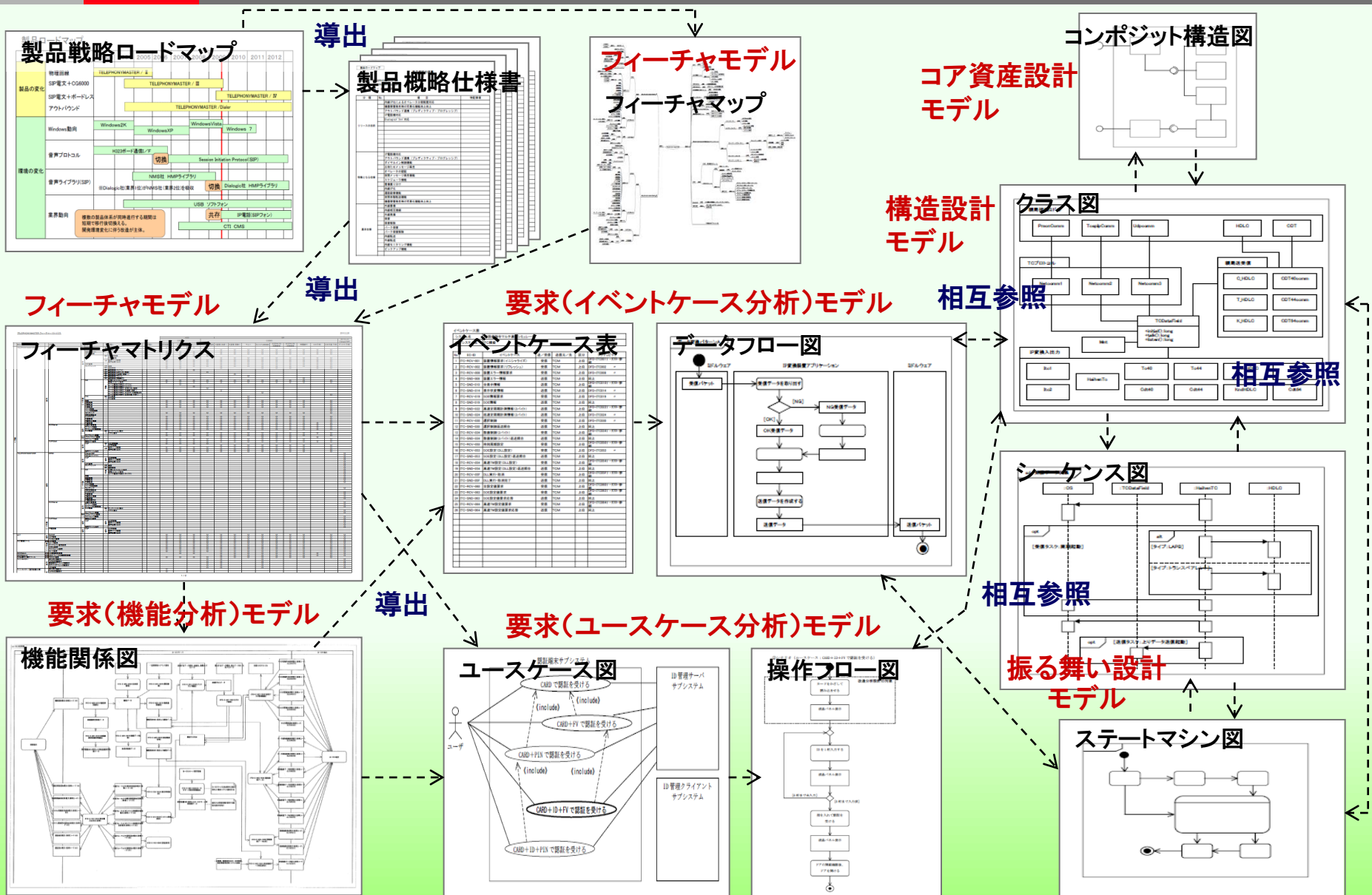
- ② **製品機能の視点**で要件分析を行い、**モデリング手法**でソフトウェア要求をソフトウェア構造にマッピングする
- ③ **再利用の視点**でコンポーネント化を行い、**モデリング手法**でソフトウェア構造を**コア資産コンポーネント構造**にマッピングする

# 3-2 モデル指向SPL開発のプロセスと成果物





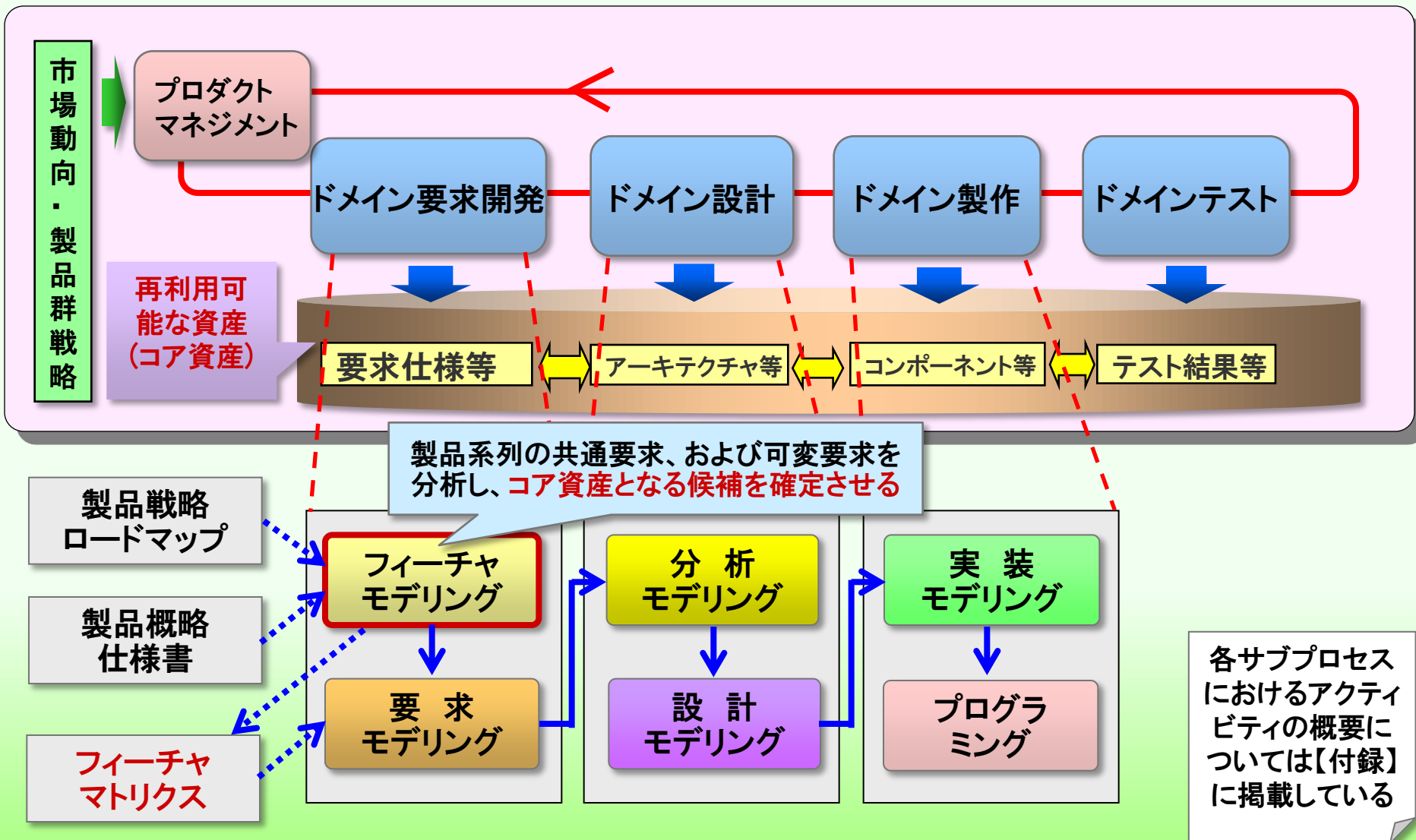
# 3-3 モデル指向SPL開発におけるモデルの流れ



# 3-4 モデル指向SPL開発のコア資産開発プロセス

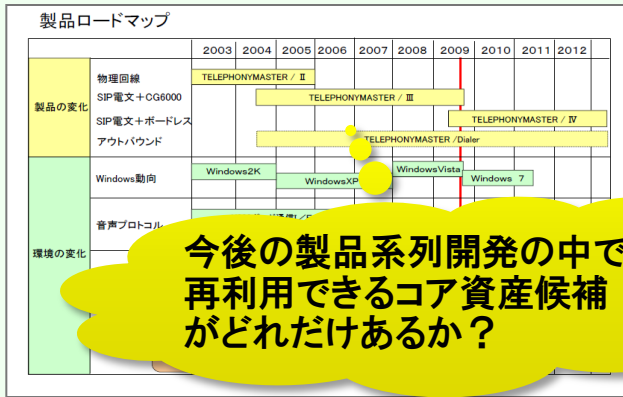
## ドメイン・エンジニアリングプロセス

サブプロセス



# 3-5 フィーチャモデリング(製品フィーチャ分析)

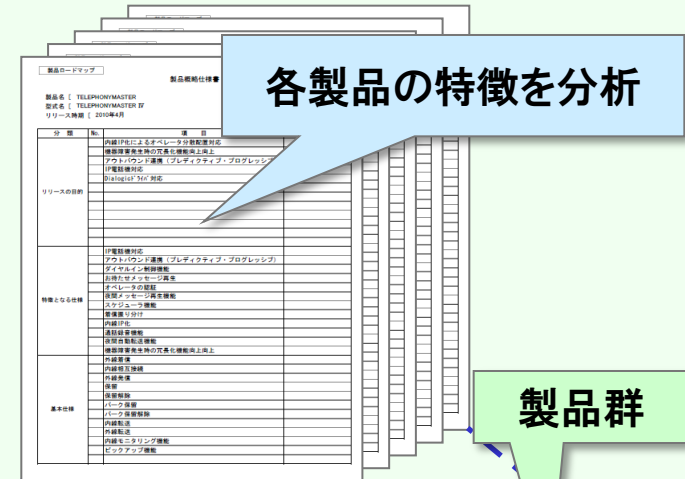
## 製品戦略ロードマップ



今後の製品系列開発の中で  
再利用できるコア資産候補  
がどれだけあるか？

## 製品分析

## 製品概略仕様書



製品群

## フィーチャ分析

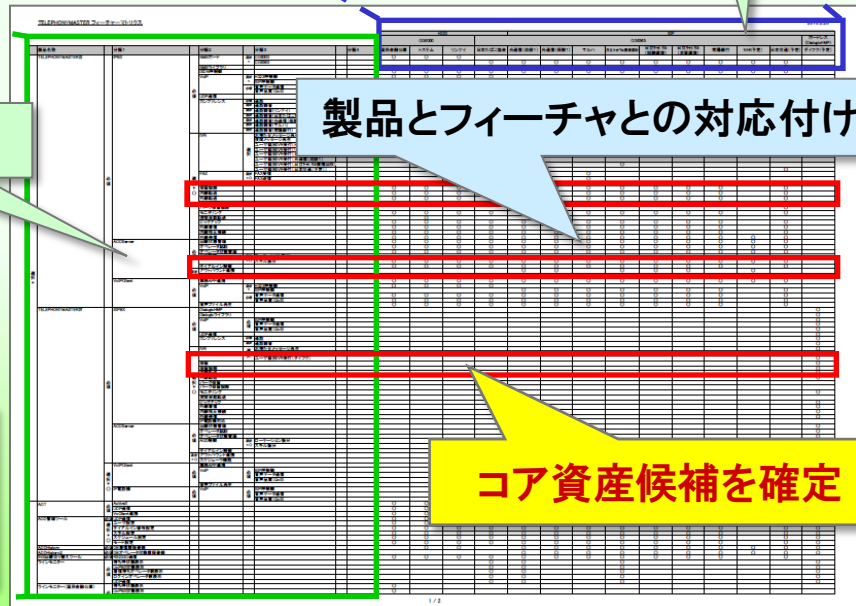
## フィーチャモデル

## フィーチャマトリクス



製品群全体の可変性を分析  
(外部可変性分析)

フィーチャ群

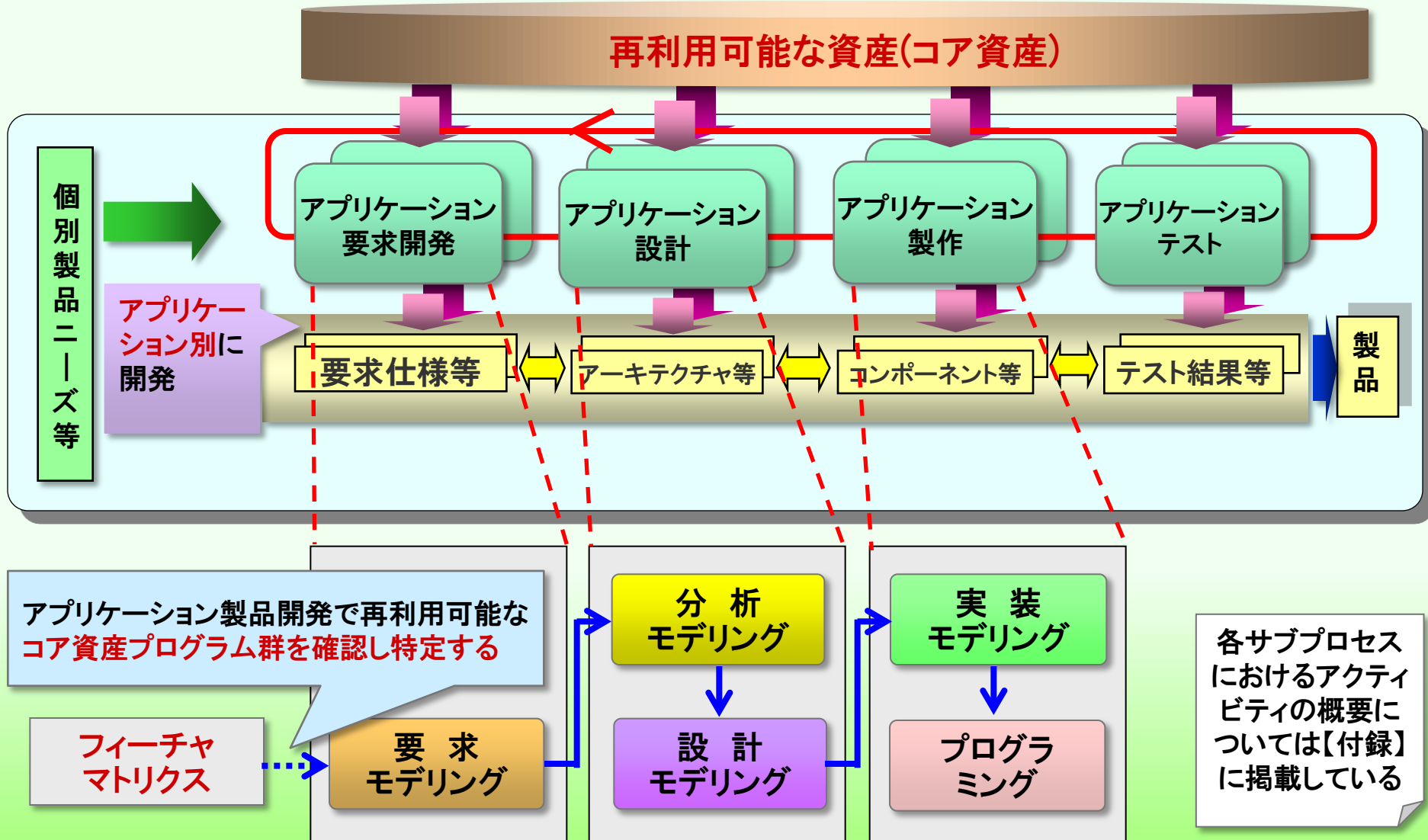


コア資産候補を確定

# 3-6 モデル指向SPL開発の製品系列開発プロセス

## アプリケーション・エンジニアリングプロセス

■ :サブプロセス



## モデル指向SPL開発におけるXDDPの効果的適用

---

### 4. モデル指向SPL開発へのXDDP適用

## ■ 製品系列の新規・リエンジニアリング開発に適用

### ● 制御系システム

- ・ 社会システム
- ・ 交通電力システム
- ・ 電力システム
- ・ 公共システム
- ・ 交通制御システム

### ● 組込み系システム

- ・ 伝送システム
- ・ 医療システム

### ● 情報系システム

- ・ 産業システム
- ・ パッケージソフト

## ■ 製品系列の改造開発に適用

### ● 派生開発プロセス手法XDDPとの融合

**進化型SPL開発手法**として社内改造開発へ適用中

## 4-2 モデル指向SPL開発の適用実績分類

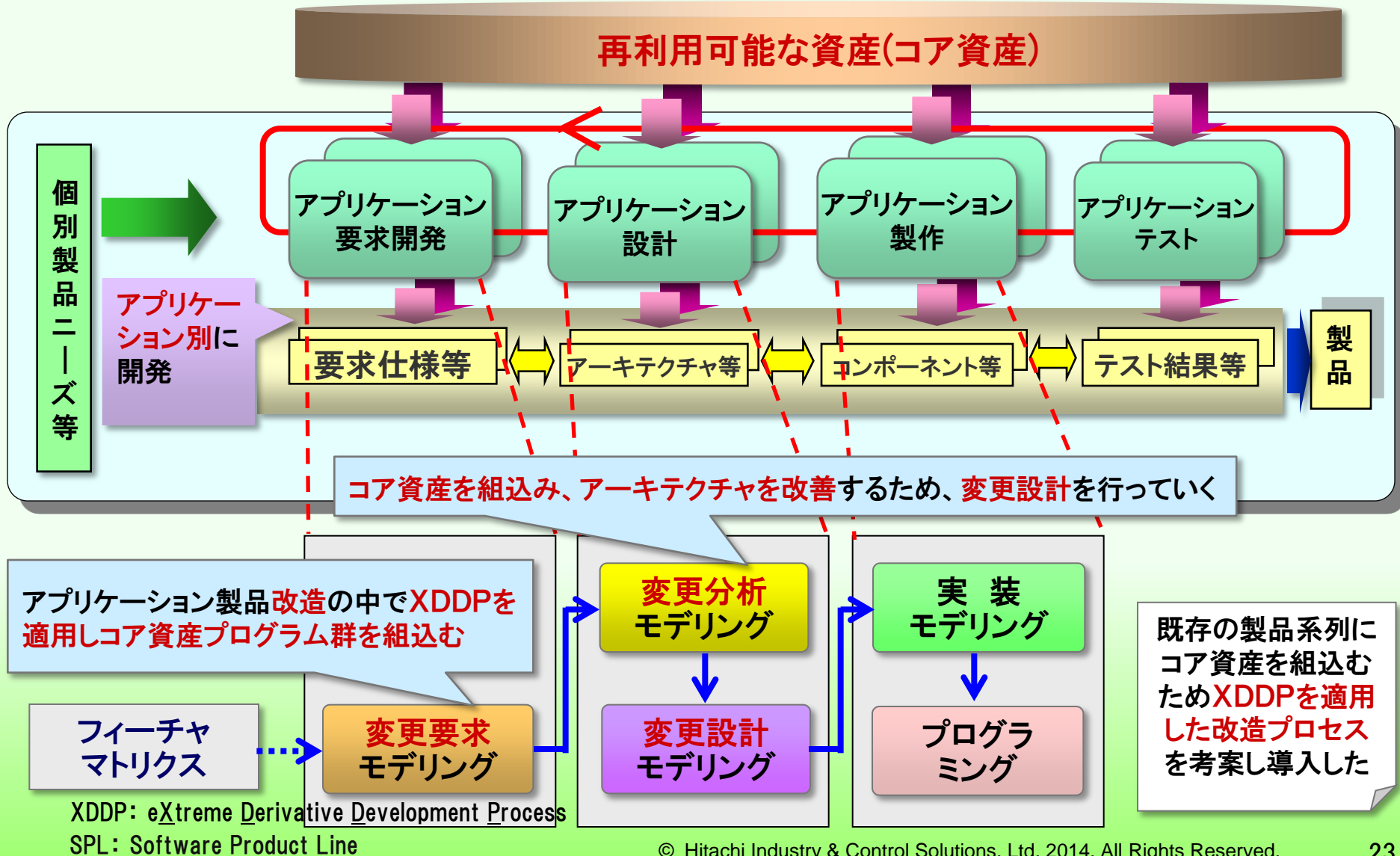
### 適用プロジェクトについてモデル指向SPL開発の適用形態の違いにより分類

| 事例 No. | 適用事例パターン                    | 適用概要                                                               |
|--------|-----------------------------|--------------------------------------------------------------------|
| 1      | リエンジニアリングによる適用              | これまで開発、保守してきた既存システムのリエンジニアリングを行う際、 <b>コア資産開発により再構築し、後続開発で再利用する</b> |
| 2      | 既存ソフトウェア流用による適用             | これまで開発、保守してきた既存システムについて、SPL開発を適用して再構築する際、可能なレベルで既存システムのソフトウェアを流用する |
| 3      | 機能追加による適用                   | 既存システムについて、 <b>機能追加のタイミングでコア資産化可能な機能を開発して組込む</b>                   |
| 4      | 機能入換えによる適用                  | 既存システムについて、 <b>外部提供機能を内部機能として入換える際、事前に内部機能をコア資産化して組込む</b>          |
| 5      | 範囲限定による適用                   | システム全体に対してSPL開発を適用するのではなく、適用範囲を限定してコア資産化を目的とした開発を行う                |
| 6      | システム改造による適用<br>【進化型SPL開発手法】 | 既に稼動しているシステム系列について、 <b>改造のタイミングで改造範囲における戦略的なコア資産化を進めていく</b>        |

# 4-3 モデル指向SPL開発の製品系列改造プロセス

## アプリケーション・エンジニアリングプロセス

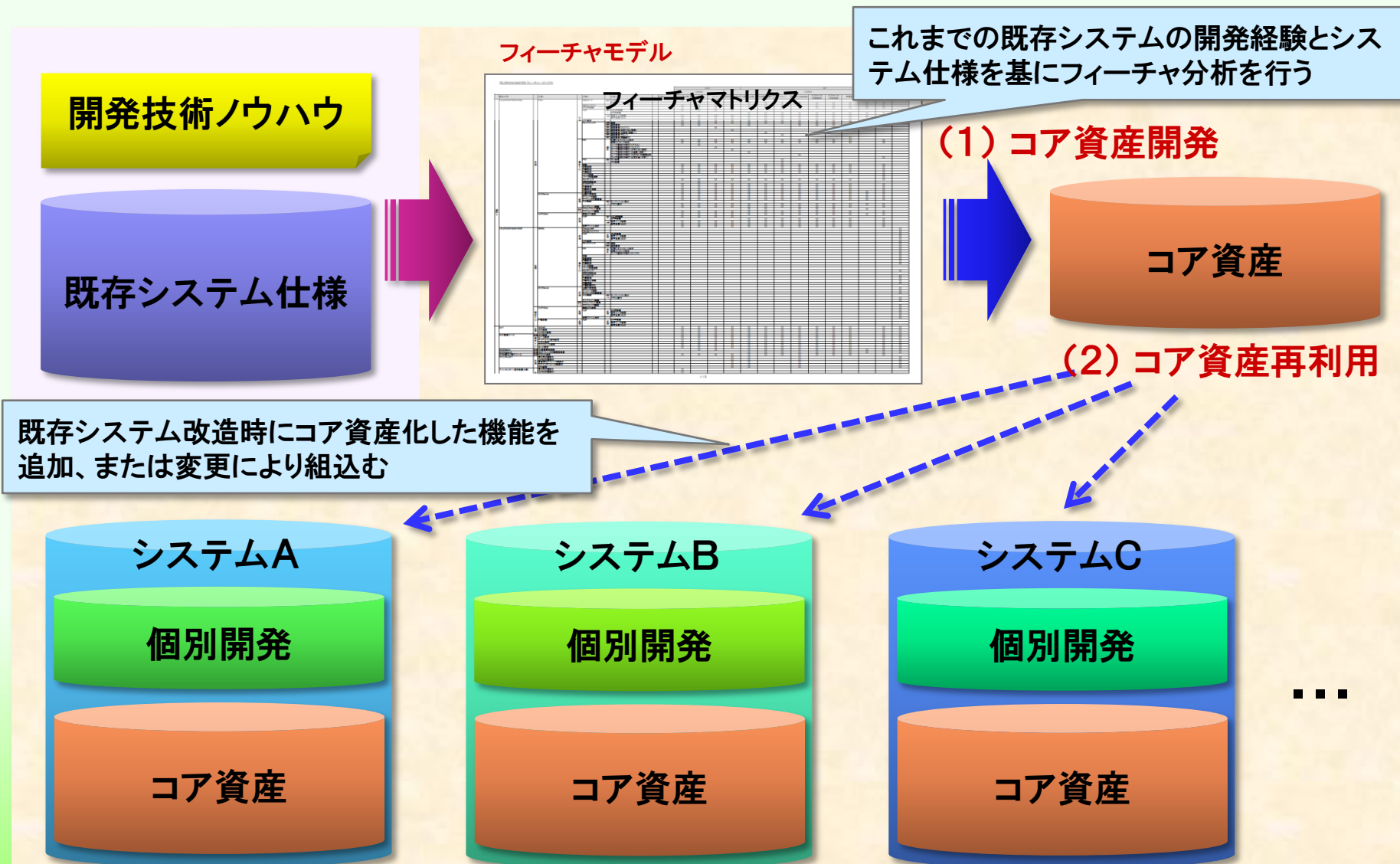
■ :サブプロセス





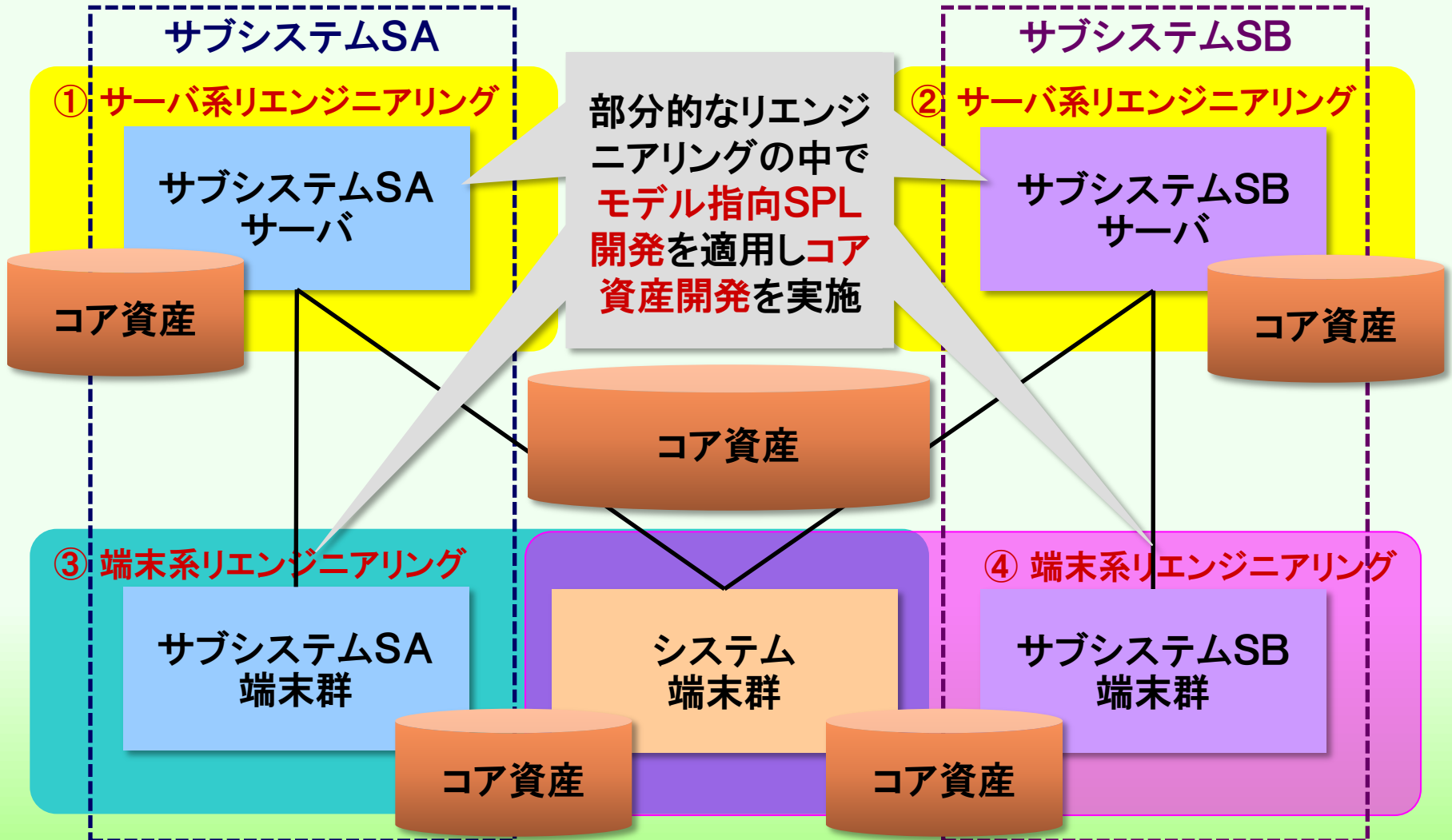
# 4-4 適用事例No.1：リエンジニアリングによる適用

既存システムのリエンジニアリングを行う際、モデル指向SPLの**コア資産開発**により再構築し、**後続開発**で再利用する



# 4-5 適用事例No.1: (1) コア資産開発の概要

システム全体ではなく部分的なリエンジニアリングを繰り返しながら、戦略的なコア資産開発を行っていく



# 4-6 適用事例No.1: (2) コア資産再利用の概要

既存システムについて、**リエンジニアリングにより事前にコア資産化した機能**を改造のタイミングで組込む

フィーチャモデル

フィーチャマトリクス

依存関係モデル(DSM\*)

\* DSM: Dependency Structure Matrix

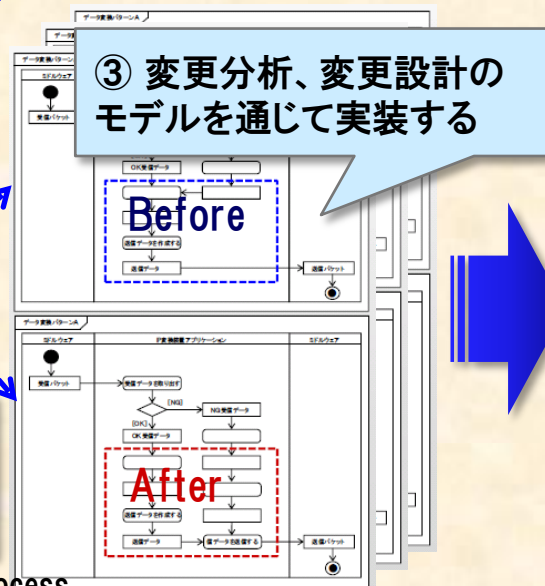
② コア資産組込みのための変更箇所と影響範囲を分析し、変更トレーサビリティマトリクスに反映させる



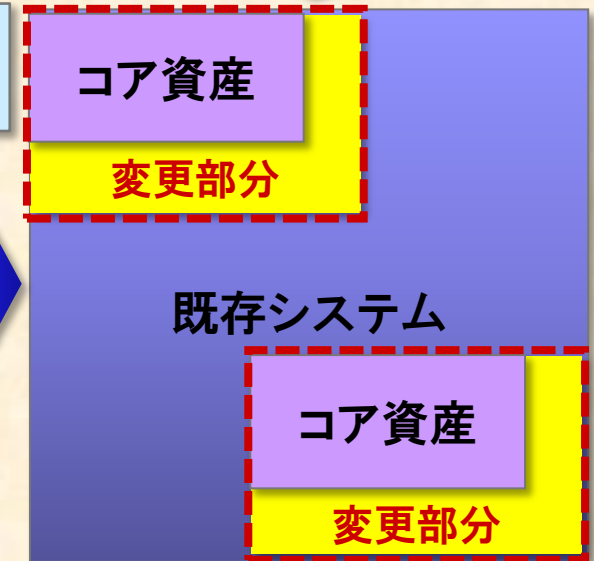
変更要求モデル

XDDP変更トレーサビリティマトリクス

変更分析モデル



① システム改造時、コア資産組込みのための変更要求、変更要求仕様、変更箇所を一緒にまとめる



XDDP: eXtreme Derivative Development Process

SPL: Software Product Line

# 4-7 適用事例No.3: 機能追加による適用

既存システムについて、機能追加のタイミングでコア資産化可能な機能を開発して組込む

フィーチャモデル

フィーチャマトリクス

① コア資産となる機能開発

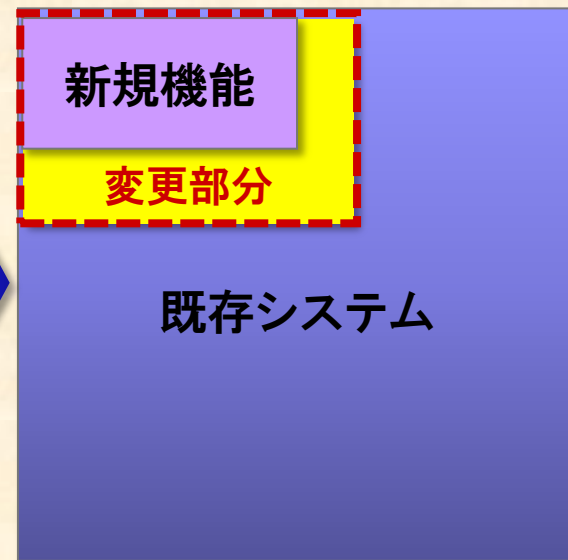
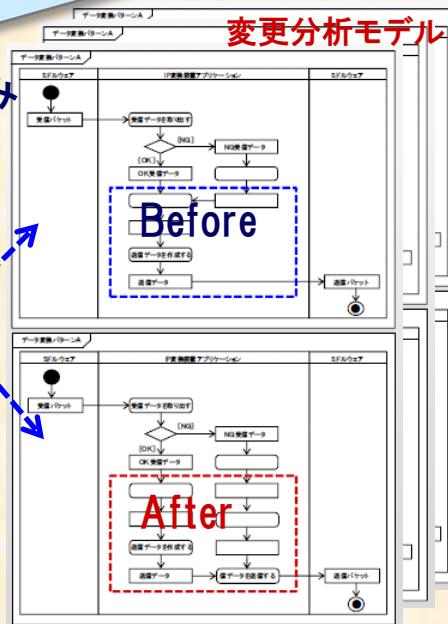


コア資産として新規に機能追加するための  
変更要求仕様、変更箇所をまとめる

変更要求モデル ② コア資産の組み込み

XDDP変更トレーサビリティマトリクス

変更分析モデル





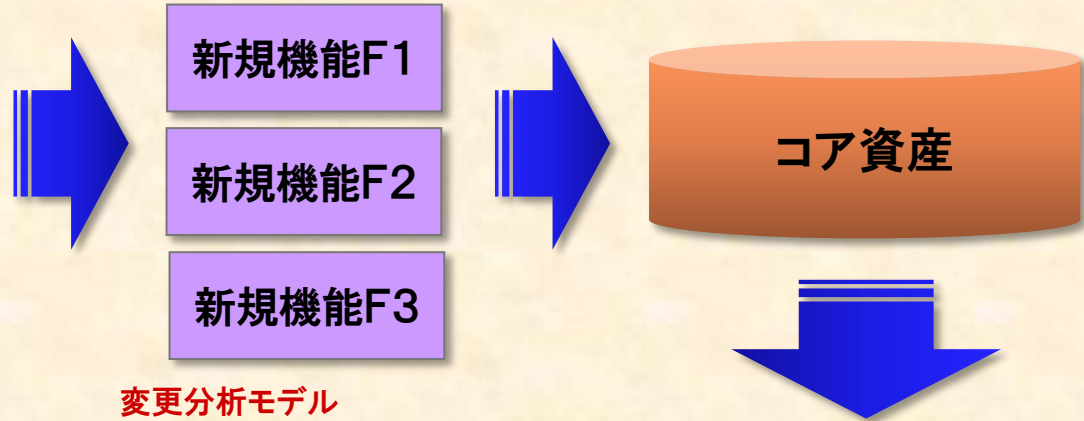
# 4-8 適用事例No.4: 機能入換えによる適用

既存システムにおける外部提供機能(市販パッケージ)を内部機能(コア資産)として開発して入換える

フィーチャモデル

フィーチャマトリクス

## ① コア資産となる機能開発

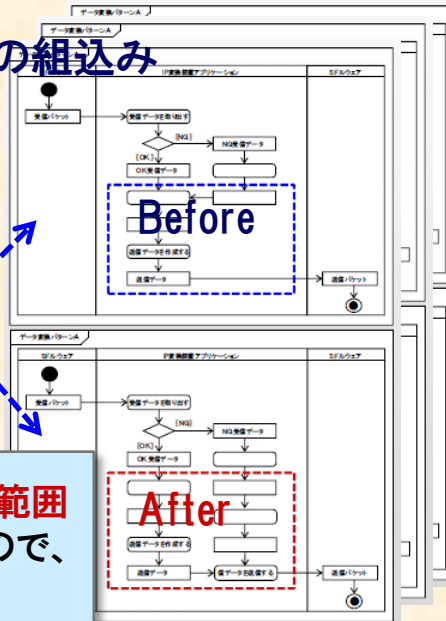


変更要求モデル

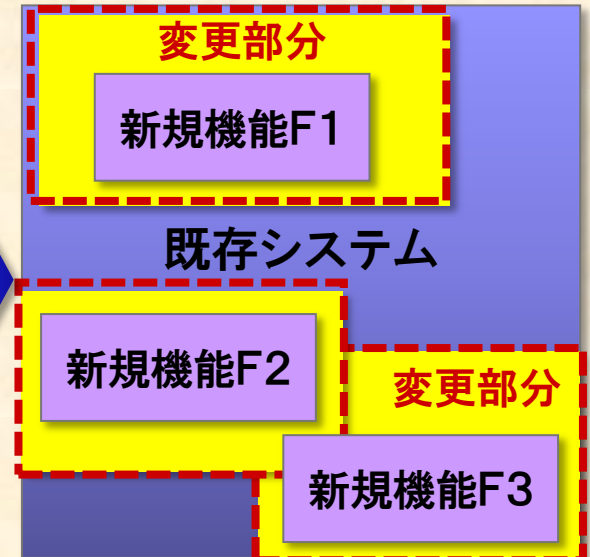
## ② コア資産の組み込み

XDDP変更トレーサビリティマトリクス

変更分析モデル



機能入換えはアーキテクチャ全体に影響範囲がおよぶ場合があり、変更が複雑になるので、特に影響範囲の洗出しが重要となる

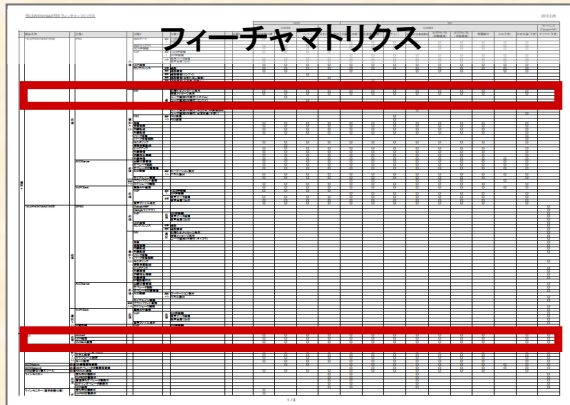


# 4-9 適用事例No.6: システム改造による適用

既に稼動しているシステム系列について、改造のタイミングで改造範囲における戦略的なコア資産化を進めていく

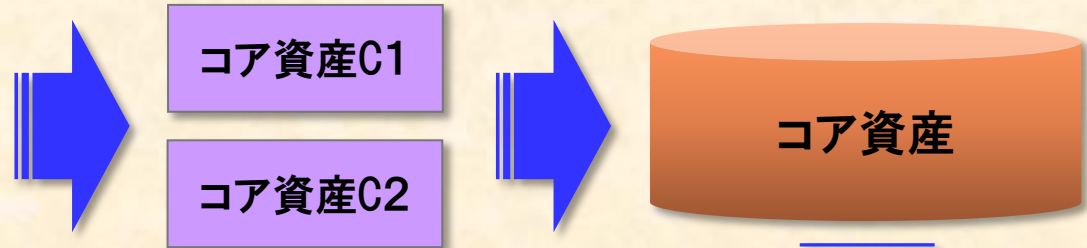
フィーチャモデル

フィーチャマトリクス



既存システムの各改造時に改造開発の中でコア資産化可能なフィーチャを特定してコア資産化を行う

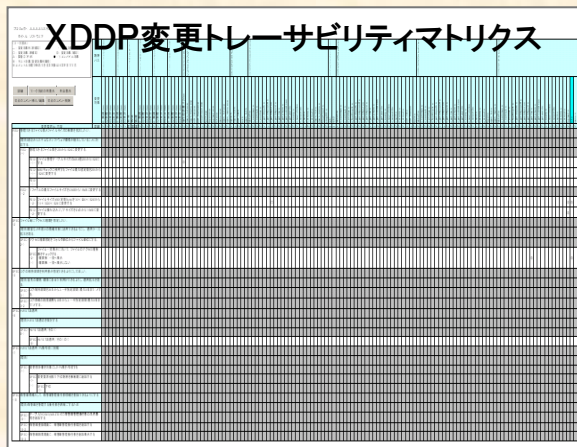
① 改造範囲内でのコア資産開発



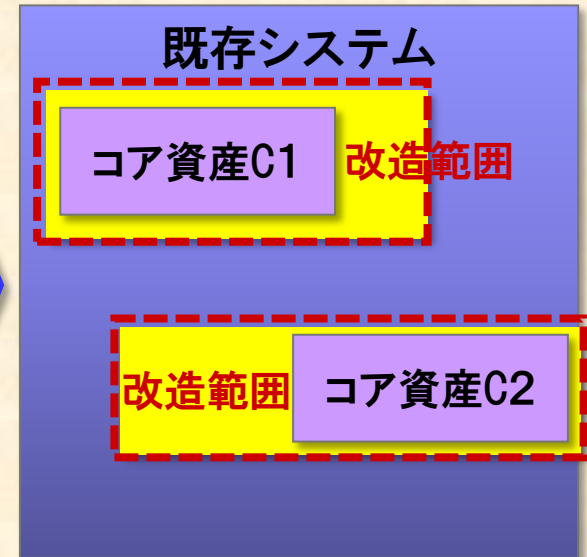
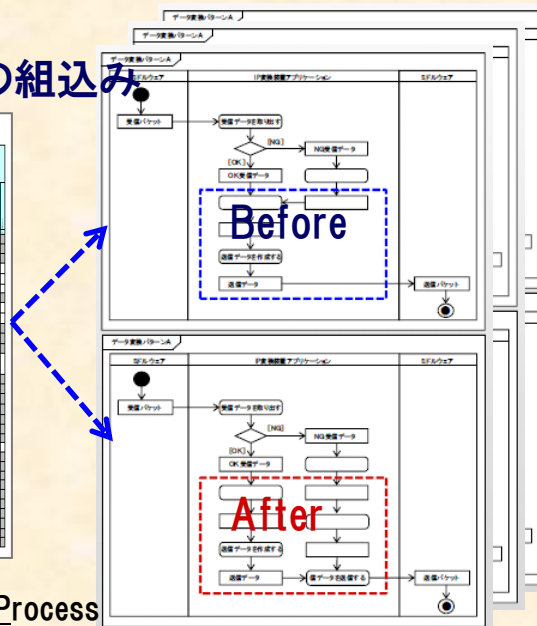
変更要求モデル

② コア資産の組込み

XDDP変更トレーサビリティマトリクス



変更分析モデル



XDDP: eXtreme Derivative Development Process

SPL: Software Product Line

## モデル指向SPL開発におけるXDDPの効果的適用

---

### 5. 効果と課題

モデル指向SPL開発におけるXDDPの適用によって得られた効果は極めて大きい

## ■ コストダウン(SPL開発の効果)

- ・ コア資産化後、**3~4システムの後続開発で損益分岐点に到達**
- ・ 損益分岐点到達後、先行投資額を回収、**低コスト開発を実現**

## ■ 品質アップ(SPL開発とXDDPの効果)

- ・ コア資産を再利用することで開発後の**社外不良ゼロ**を実現
- ・ XDDPを適用することで**コア資産組込み時のヌケ、モレを防止**

## ■ 開発スタイル改善(モデル指向開発とXDDPの効果)

- ・ 図面中心の開発ドキュメントで**設計情報の曖昧さを排除**
- ・ XDDPの3点セット活用で**システム改造の効率アップ**を実現



今後、社内外のソフトウェア開発について技術改革していくために課題と考えていること

### ■ SPL開発適用の普及

- ・ 同種のソフトウェアを**繰返し作らない開発**を目指す  
(コピー&ペースト開発から脱却する)
- ・ これまでの**適用実績をノウハウとして活用**する

### ■ コア資産管理方式の確立

- ・ **手を加えない再利用**の考え方を浸透させる
- ・ **コア資産の管理、運用**を容易にする

### ■ テストへの適用拡大

- ・ **テストフェーズにSPLを適用**し、テスト要件、テスト方式  
(テストアーキテクチャ)をコア資産化する

**END**



## モデル指向SPL開発におけるXDDPの効果的適用

株式会社 日立産業制御ソリューションズ

業務改革本部 生産技術部

渡辺 滋

## プロセス

## アクティビティ概要

フィーチャモデリング  
(製品フィーチャ分析)

**【目的】**コア資産化の方針を決める  
**【概要】**製品戦略ロードマップに基づき各製品の概略仕様を決め製品系列全体におけるフィーチャを詳細化して可変性を分析する

要求モデリング  
(フィーチャ要件定義)

**【目的】**製品フィーチャの要件を定義する  
**【概要】**製品系列全体としての振る舞いと各フィーチャに必要な振る舞いについて可変性を考慮してモデル化する

分析モデリング  
(アーキテクチャ分析)

**【目的】**基盤となるアーキテクチャを構築する  
**【概要】**全フィーチャを実現させるソフトウェア構造を定義しコンポーネントとコンポーネント間のインタフェース仕様を決める

設計モデリング  
(コア資産構造設計)

**【目的】**コア資産コンポーネント構造を決める  
**【概要】**フィーチャレのベルから見えない内部可変性を分析し、詳細化されたクラス構造のレベルで可変性を設計する

実装モデリング  
(コア資産処理設計)

**【目的】**コア資産実装のための準備をする  
**【概要】**コア資産の実装に近いプロセス構造やコンポーネント、クラスにおけるプログラム構造レベルで処理メソッドを設計する

トレーサビリティ  
管理

コア資産開発の  
中で作成される  
モデルをすべて  
トレースする

## プロセス

## アクティビティ概要

フィーチャマトリクス  
の参照・更新

【目的】再利用と開発すべき機能を見積る  
【概要】開発対象の製品について、製品間で  
共通のフィーチャ(コア資産)と製品固有の  
フィーチャについて確認する

要求モデリング  
(製品ソフトウェア要件定義)

【目的】製品ソフトウェアの要件を定義する  
【概要】製品の持つ各フィーチャに必要となる  
振る舞いについて、可変性を考慮してモデル  
化する

分析モデリング  
(製品アーキテクチャ分析)

【目的】製品アーキテクチャを構築する  
【概要】製品の各フィーチャを実現するクラス  
とコンポーネントの構造を定義して、クラス、  
およびコンポーネントの関係を定める

設計モデリング  
(製品構造設計)

【目的】製品のソフトウェア構造を確定させる  
【概要】再利用するコア資産コンポーネントの  
インターフェースを含め製品全体のソフトウェア  
構造を設計する

実装モデリング  
(製品処理設計)

【目的】製品ソフトウェア実装の準備を行う  
【概要】製品ソフトウェアを実装するプロセス  
構造、およびプログラム構造レベルでの処理  
メソッドを設計する

トレーサビリティ  
管理

製品系列開発の  
中で作成される  
モデルをすべて  
トレースする