

事例発表

XDDP・SPLの融合と適用事例の紹介

2011.11.7

株式会社 日立情報制御ソリューションズ

業務サポート本部

生産技術部 ソフトウェアエンジニアリンググループ

渡辺 滋

XDDP: eXtreme Derivative Development Process

SPL: Software Product Line

XDDP・SPLの融合と適用事例の紹介

Contents

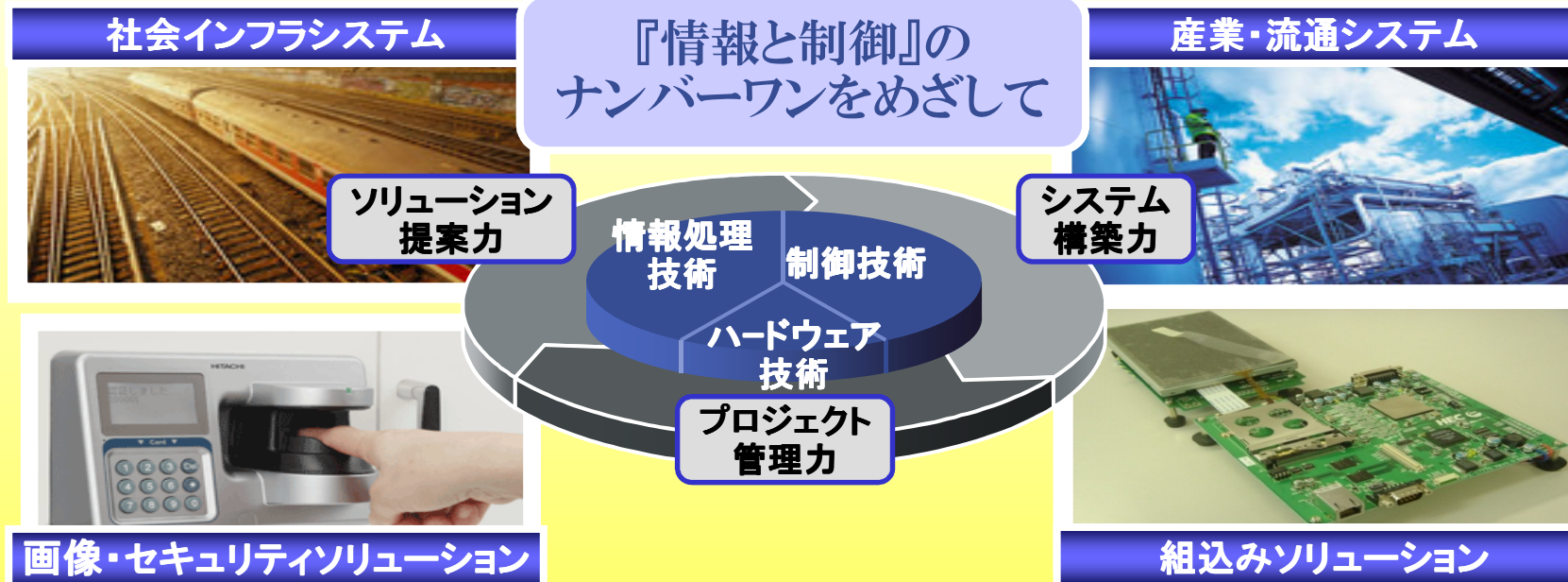
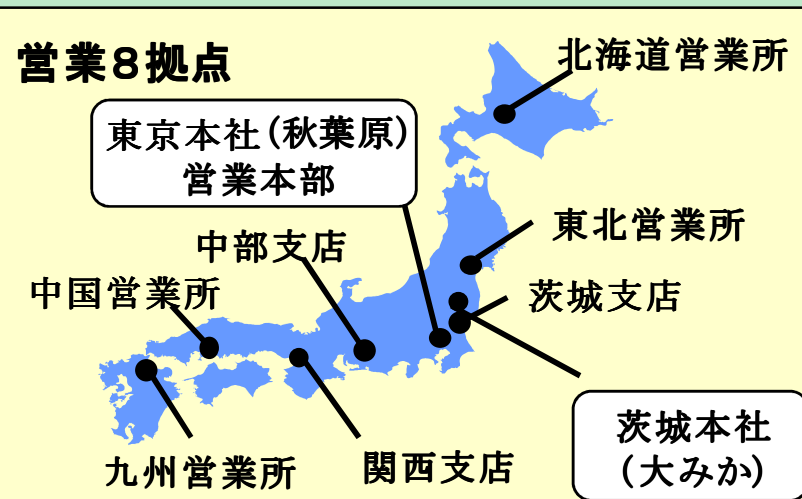
- ★ 1. 当社の紹介とソフトウェア開発概況
- 2. XDDPとSPLの融合とは
- 3. XDDPとSPLを融合した開発事例
- 4. 期待できる効果
- 5. 今後の計画と課題

XDDP: eXtreme Derivative Development Process

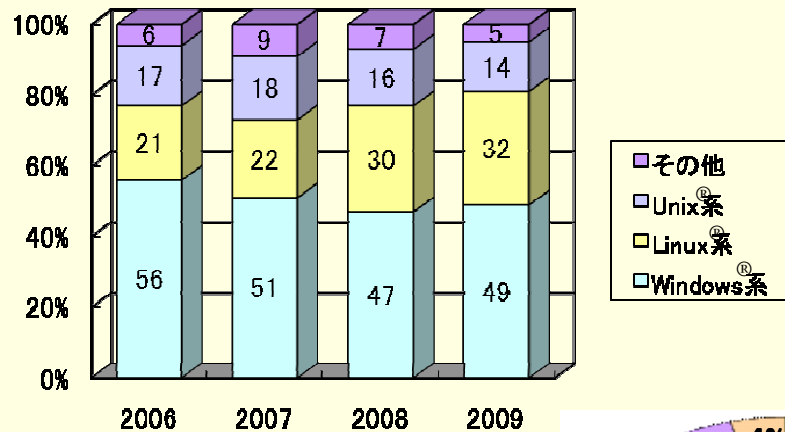
SPL: Software Product Line

1-1. 日立情報制御ソリューションズを紹介

設立：2006年4月1日
本社所在地：茨城県日立市大みか町五丁目1番26号
資本金：22億7千万円
売上高：479億円(2010年度)
代表者：茅根 修
社員数：2,759名(2011年4月現在)
URL：<http://www.hitachi-ics.co.jp>

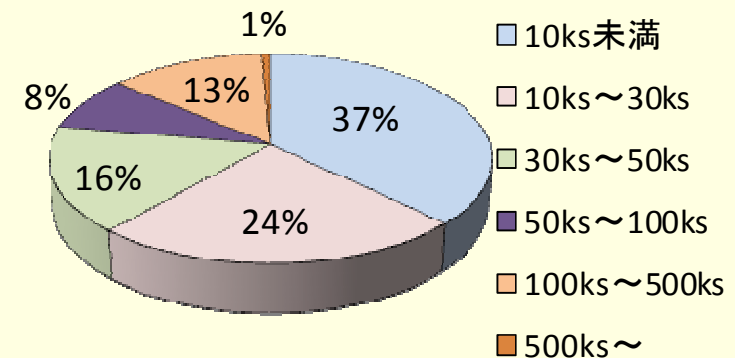


1-2. ソフトウェア開発概況



【OS適用比率の推移】

出展:2010年社内集計データより

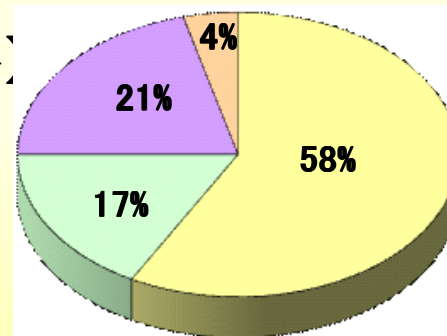


【ソフトウェア開発規模(単位:kSLOC)】

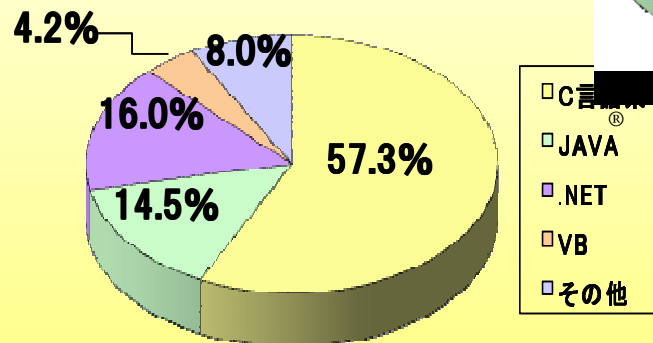
(kSLOC:Kilo Source Lines Of Codes)

(注) 規模は、改造母体、流用を含めて換算したもの

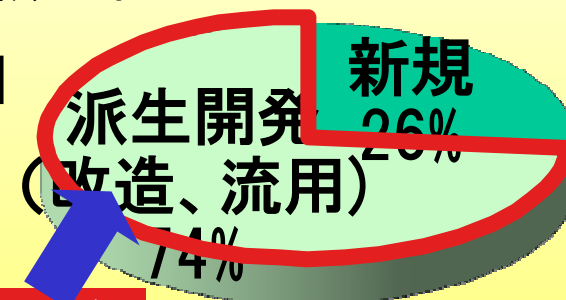
- 社会インフラシステム
- 産業・流通システム
- 組み込みソリューション
- 画像・セキュリティソリューション



【セグメント別売上高比率】



【開発言語比率】



派生開発が多い

【生産量比率】

UNIXは、X/Open Company Ltd.が独占的にライセンスしている米国およびその他の国における登録商標です。
Linuxは、Linus Torvalds氏の米国およびその他の国における登録商標または商標です。
Windowsは、米国Microsoft Corp.の米国およびその他の国における登録商標です。
Javaは、米国Sun Microsystems, Inc.の米国およびその他の国における登録商標または商標です。

XDDP・SPLの融合と適用事例の紹介

Contents

1. 当社の紹介とソフトウェア開発の状況
- ★2. XDDPとSPLの融合とは
3. XDDPとSPLを融合した開発事例
4. 期待できる効果
5. 今後の計画と課題

XDDP: eXtreme Derivative Development Process

SPL: Software Product Line

2-1. XDDPとSPLの融合とは

派生開発手法XDDP

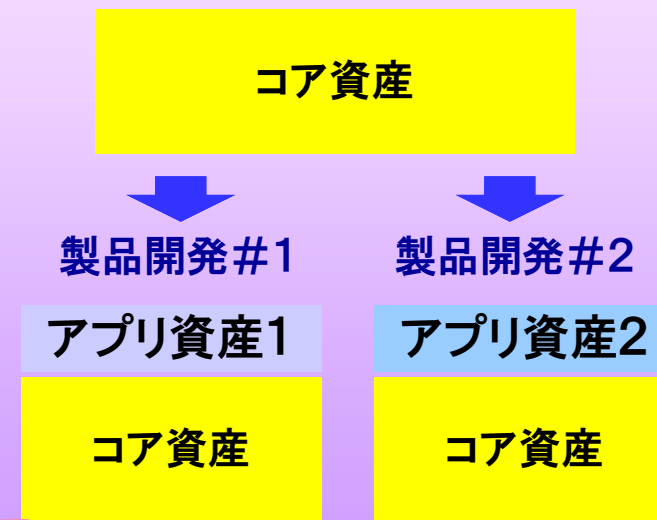
既存資産からの派生開発の際に品質向上を図ることができる手法



Software Product Line

製品系列開発に効果的な新規開発向けの手法

予め製品系列で共通となるコア資産を開発



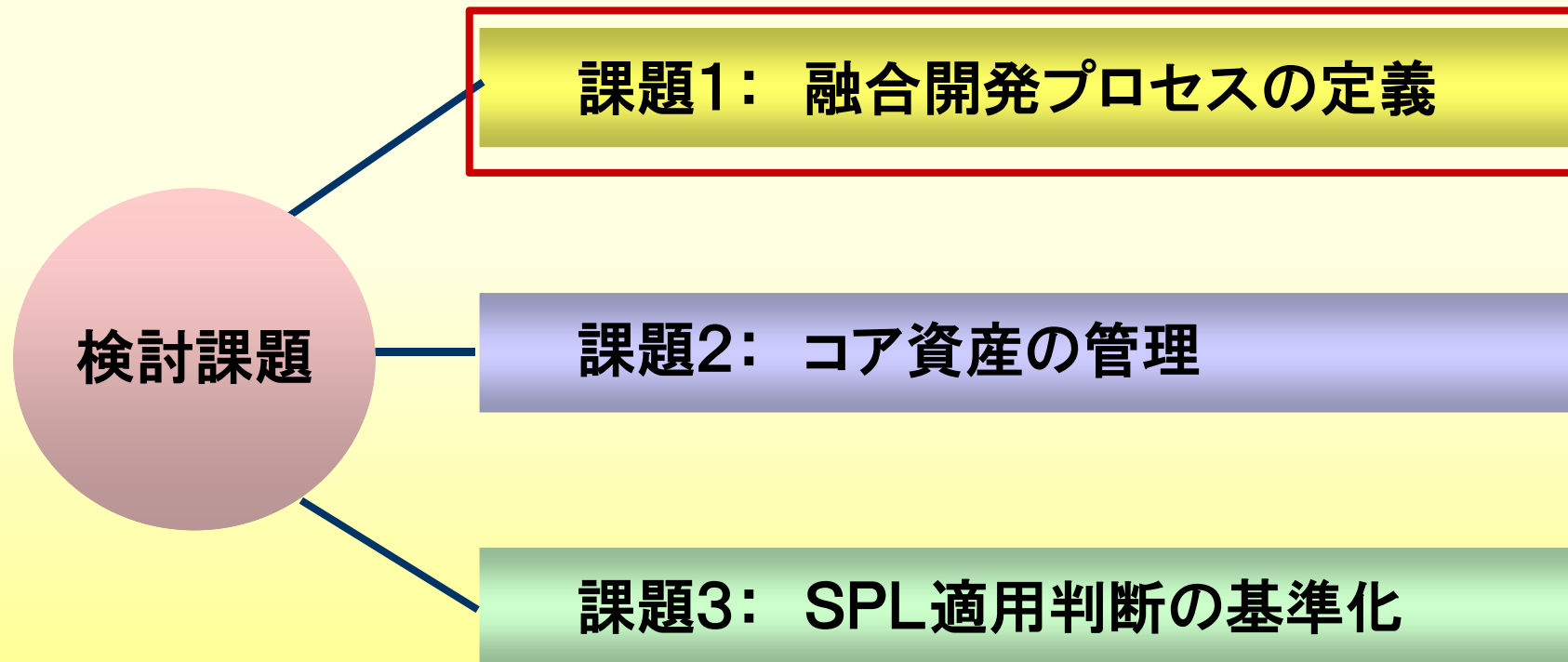
両手法を融合させて双方のメリットを享受できないか！

新規開発が少ないので適用する機会も少ない

XDDP: eXtreme Derivative Development Process

SPL: Software Product Line

XDDPとSPLを融合させた開発手法を確立するための検討課題

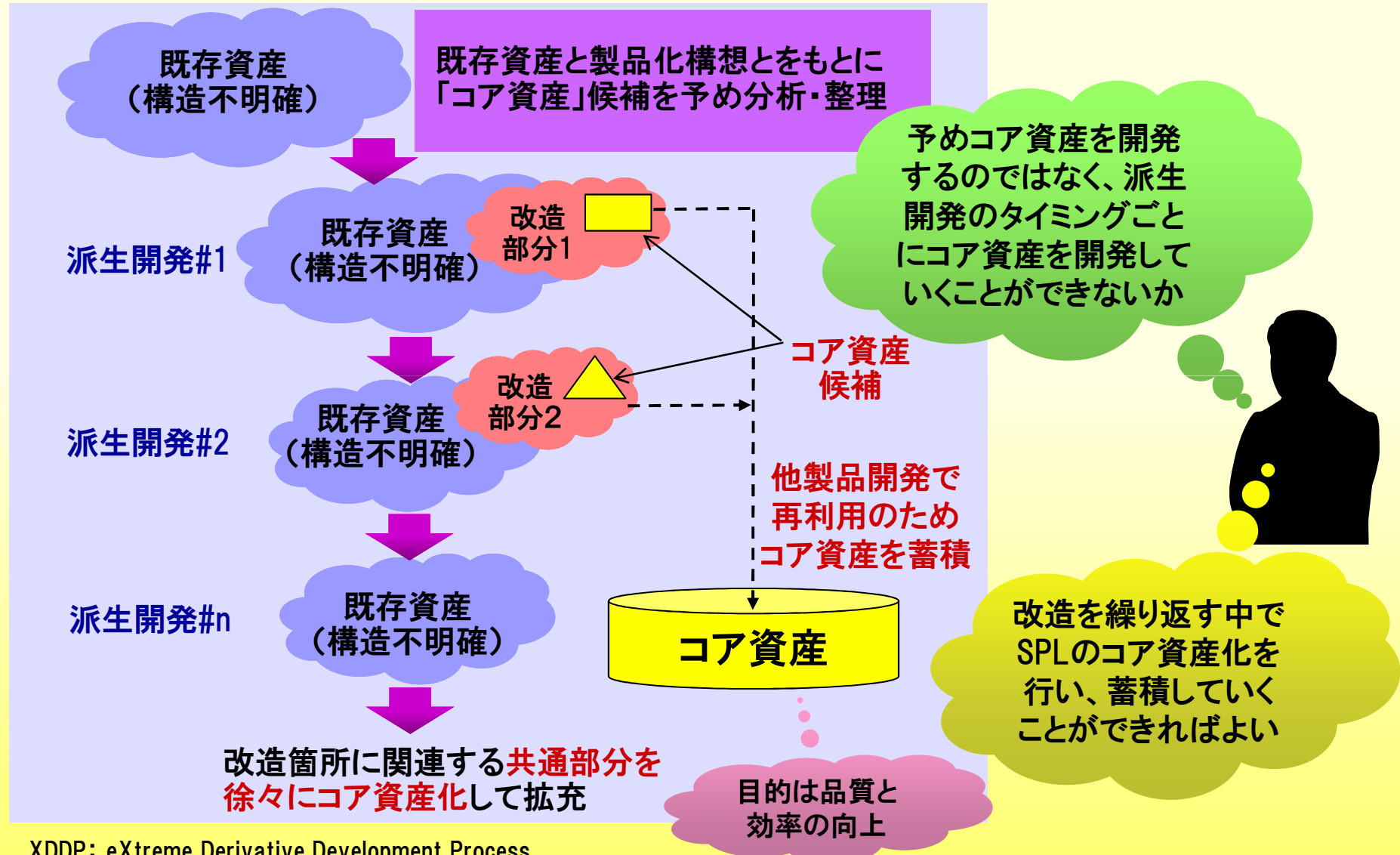


XDDP: eXtreme Derivative Development Process

SPL: Software Product Line

2-3. 課題1: 融合型開発プロセスの定義(1)

XDDPとSPLを融合させた開発のイメージ



XDDP: eXtreme Derivative Development Process

SPL: Software Product Line

2-4. 課題1： 融合型開発プロセスの定義(2)

開発プロセスを定義するための具体的なアプローチ

派生開発XDDP

変更要求に関する改造範囲に限定して開発するため、プログラム中心の開発で充分（**コード指向開発**でよい）

SPL開発手法

アーキテクチャを基盤としてフィーチャの可変性に着目して開発するため、アーキテクチャ中心の開発が必要（**モデル指向開発**が適している）



XDDP・SPLを融合した開発

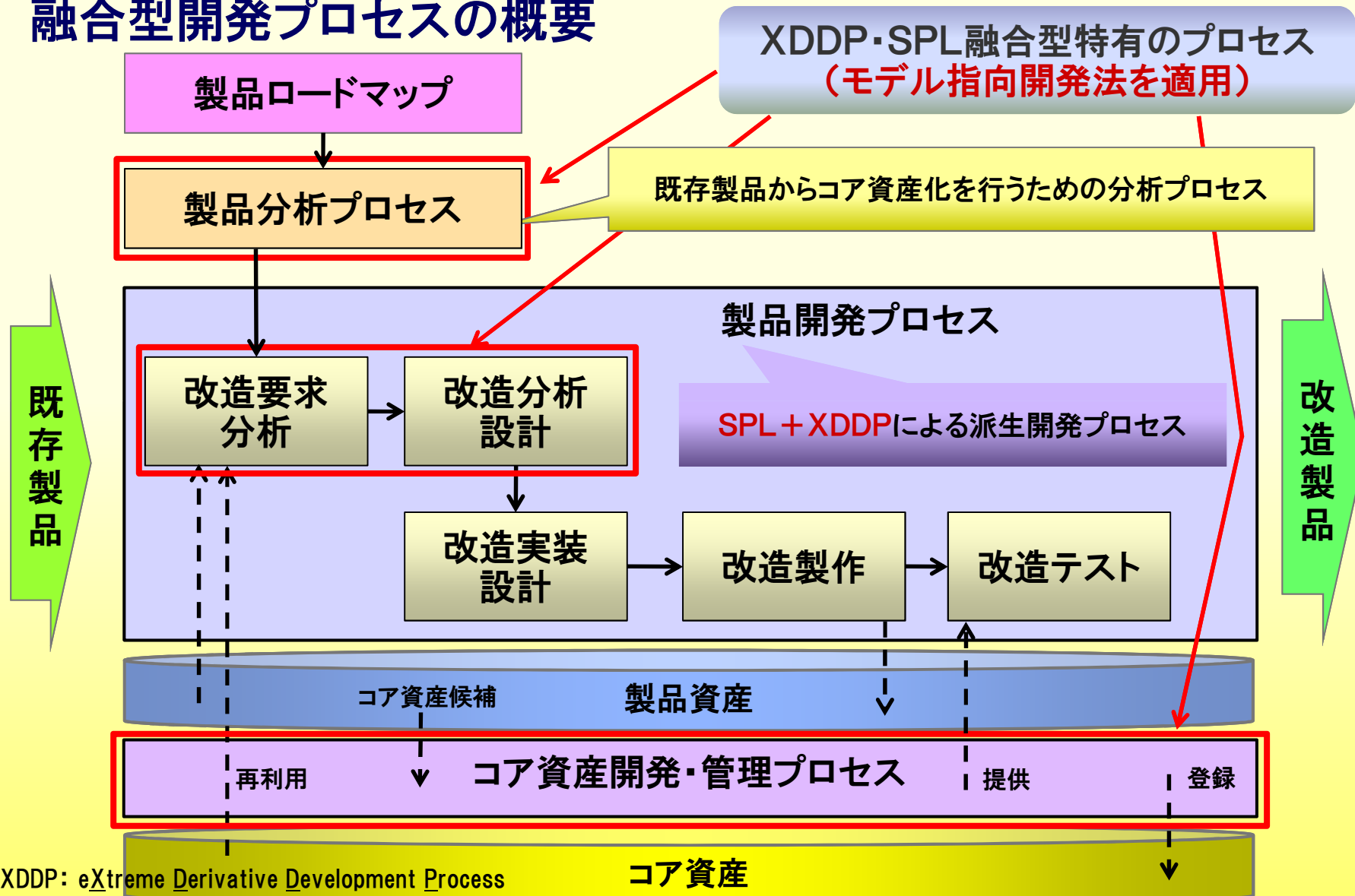
アーキテクチャ中心の開発が必要のため、**モデル指向開発**が適している

XDDP: eXtreme Derivative Development Process

SPL: Software Product Line

2-5. 課題1: 融合型開発プロセスの定義(3)

融合型開発プロセスの概要



XDDP: eXtreme Derivative Development Process

SPL: Software Product Line

コア資産

2-6. 製品分析プロセスの概要(1)

製品分析プロセス

製品戦略に基づいた製品系列全体のフィーチャと既存ソフトウェアの変更実績、全体構造を分析

SPL開発手法

製品ロードマップ

製品フィーチャ分析

コア資産化の対象を決めるため、製品のフィーチャを定義し、共通性、可変性を分析する

フィーチャマトリクス

プログラム変更履歴

変更パターン分析

将来の製品変更を予測しコア資産化に反映させるため、過去のソフトウェア変更履歴を分析する

変更パターントレーサビリティマトリクス

製品ソースプログラム

アーキテクチャ分析

将来のアーキテクチャの方向性を決めるため、現状のソフトウェア構造を分析する

目標アーキテクチャ概要図

製品分析プロセスの成果物を参照

製品開発プロセス

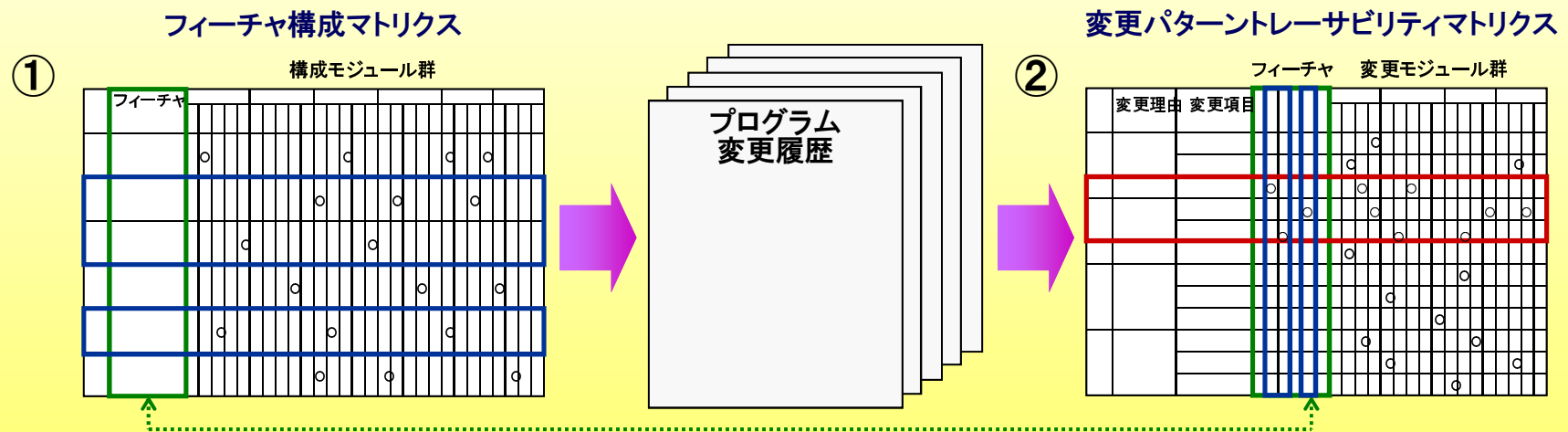
変更パターン分析

目的

製品機能について過去の変更履歴を分析し、将来のコア資産化を決定する

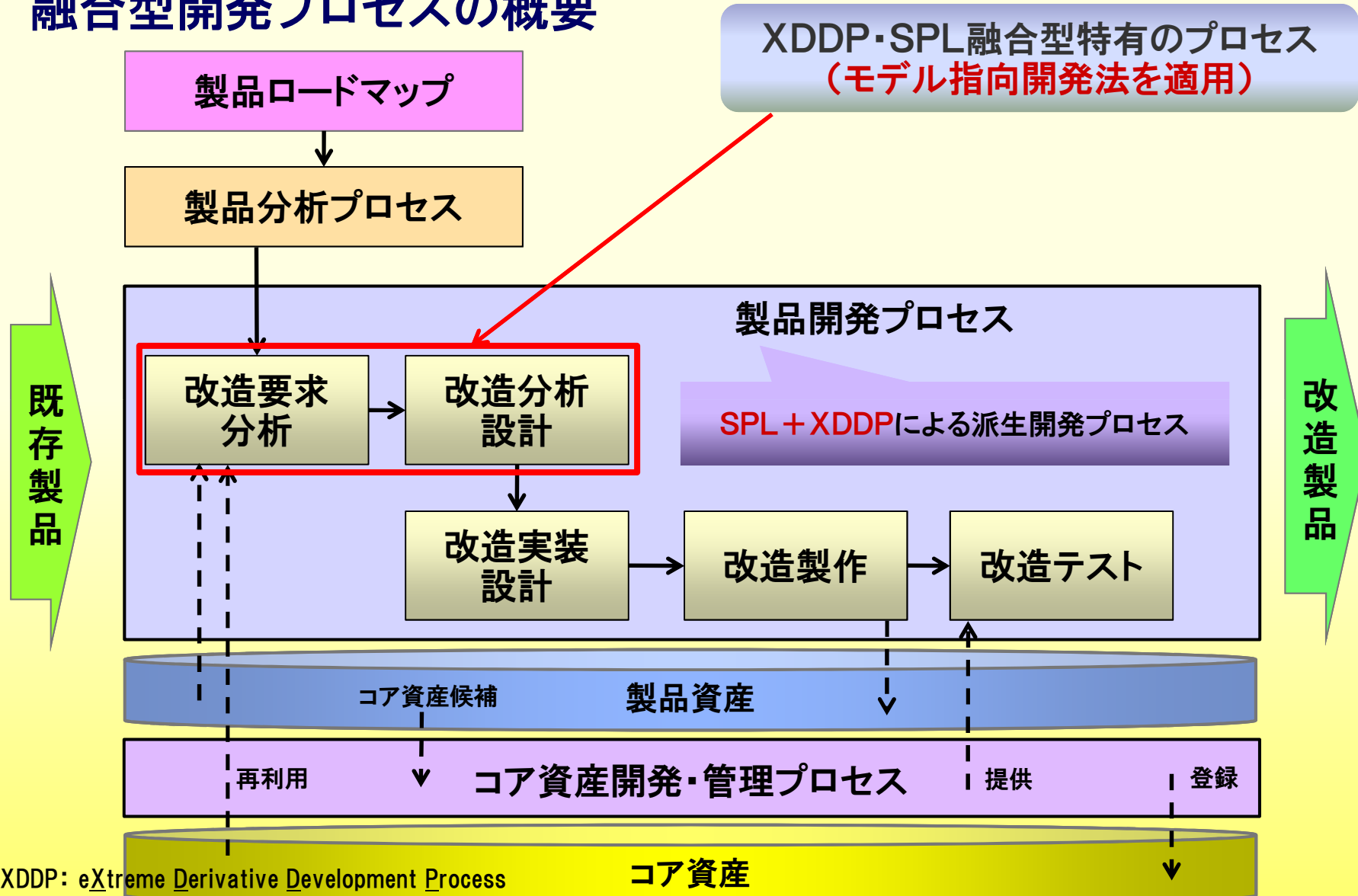
手順

- ① 各フィーチャに関係しているモジュール群を調べて、フィーチャとモジュール群の対応付けをまとめたフィーチャ構成マトリクスを作成する
- ② 変更履歴から、変更パターンレーサビリティマトリクスに変更内容(変更理由、変更項目、変更モジュール群)を記載する
- ③ フィーチャ構成マトリクスを参照して、変更項目がどのフィーチャに関係しているかを調べ、変更パターンレーサビリティマトリクスに表記する。これにより、変更内容に対応したフィーチャを抽出して整理し、将来のコア資産化対象を絞り込む



2-9. 製品開発プロセスの概要

融合型開発プロセスの概要



XDDP: eXtreme Derivative Development Process

SPL: Software Product Line

コア資産

2-11. 改造要求分析(2: コア資産化対象の決定)

④ 改造要件から見たコア資産化対象を最終決定する

各フィーチャの構成モジュール群は、**フィーチャ構成マトリクス**を参照する

コア資産化の対象を判断する際、必ずそのフィーチャが**変更パターントレーサビリティマトリクス**に無いことを確認する

フィーチャ構成マトリクス

構成モジュール群

フィーチャ	モジュール1	モジュール2	モジュール3	モジュール4	モジュール5	モジュール6	モジュール7	モジュール8	モジュール9	モジュール10
フィーチャ1	○	○	c						○	
フィーチャ2				○						
フィーチャ3					○					
フィーチャ4						○				
フィーチャ5							○			
フィーチャ6								○		
フィーチャ7									○	
フィーチャ8										○

変更パターン分析で作成

変更パターントレーサビリティマトリクス

変更要件	フィーチャ	モジュール1	モジュール2	モジュール3	モジュール4	モジュール5	モジュール6	モジュール7	モジュール8	モジュール9	モジュール10
変更要件1	c	○	c	○	○	○	○	○	○	○	○
変更要件2	○			○							
変更要件3	○				○						
変更要件4	○					○					
変更要件5	○						○				
変更要件6	○							○			
変更要件7	○								○		
変更要件8	○									○	

変更パターン分析で作成

変更トレーサビリティマトリクス

変更要件	フィーチャ	モジュール1	モジュール2	モジュール3	モジュール4	モジュール5	モジュール6	モジュール7	モジュール8	モジュール9	モジュール10
変更要件1	○										
変更要件2		○									
変更要件3			○								
変更要件4				○							
変更要件5					○						
変更要件6						○					
変更要件7							○				
変更要件8								○			

変更対象のフィーチャ 変更対象のモジュール

2-12. 改造要求分析(3: 要求モデルの作成)

目的

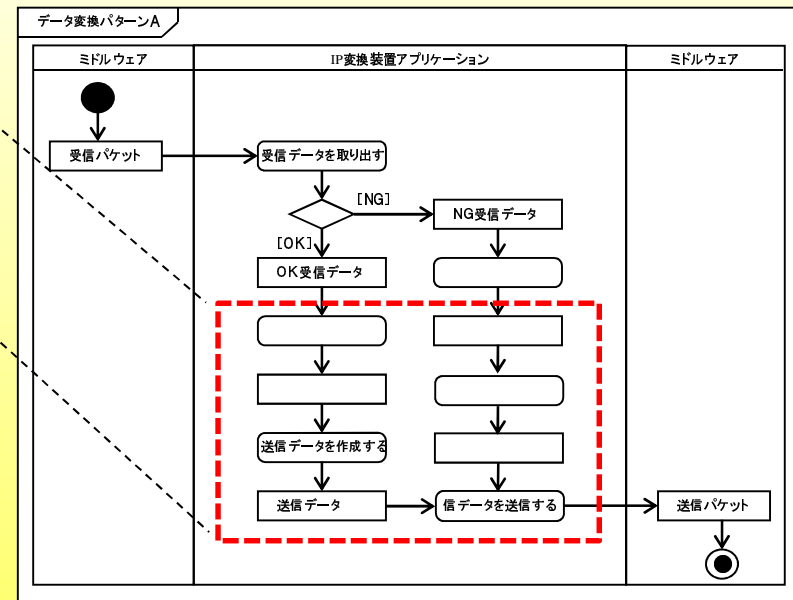
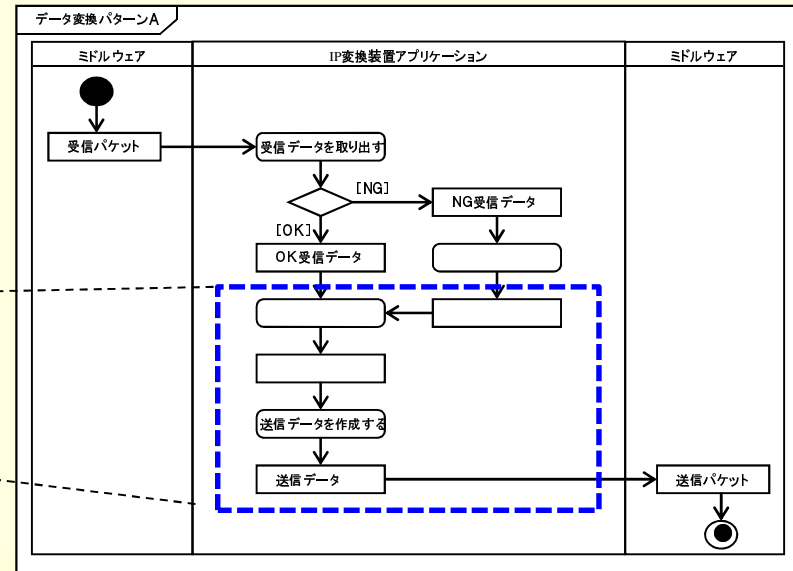
変更要求仕様から改造分析設計につなげるため、変更の要求モデルを作成する

変更要件	
変更要求	
変更理由	
変更仕様	Before After
変更仕様	
変更要求	
変更理由	
変更仕様	

Before
要求モデル

After
要求モデル

変更要求仕様の内容に関して対象となるフィーチャのBefore要求モデルとAfter要求モデルを作成して改造範囲を明確にする



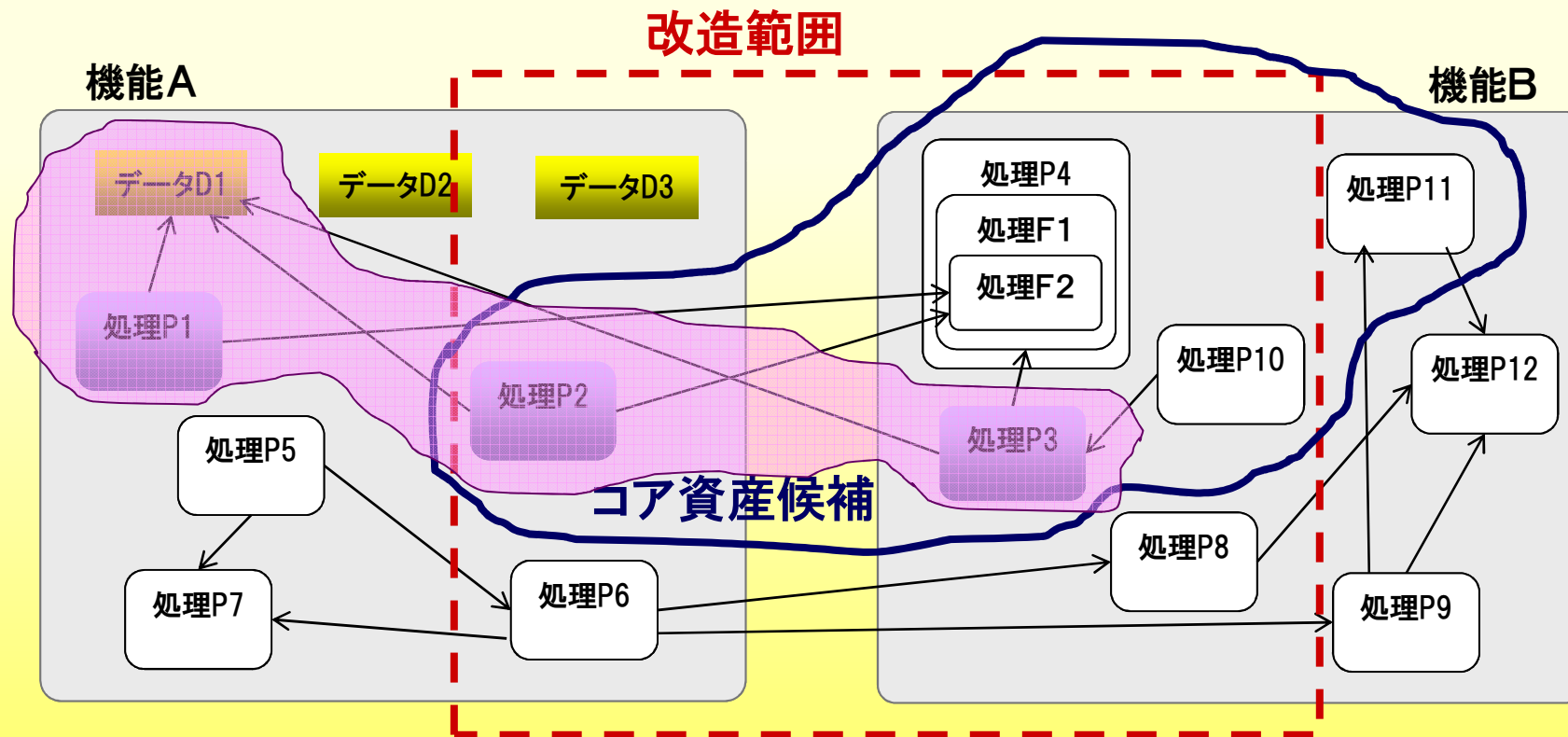
2-13. 改造分析設計(1: 依存関係の局所化)

目的

フィーチャをコア資産化するため、コア資産化の対象となる機能間、処理間の依存関係を局所化する

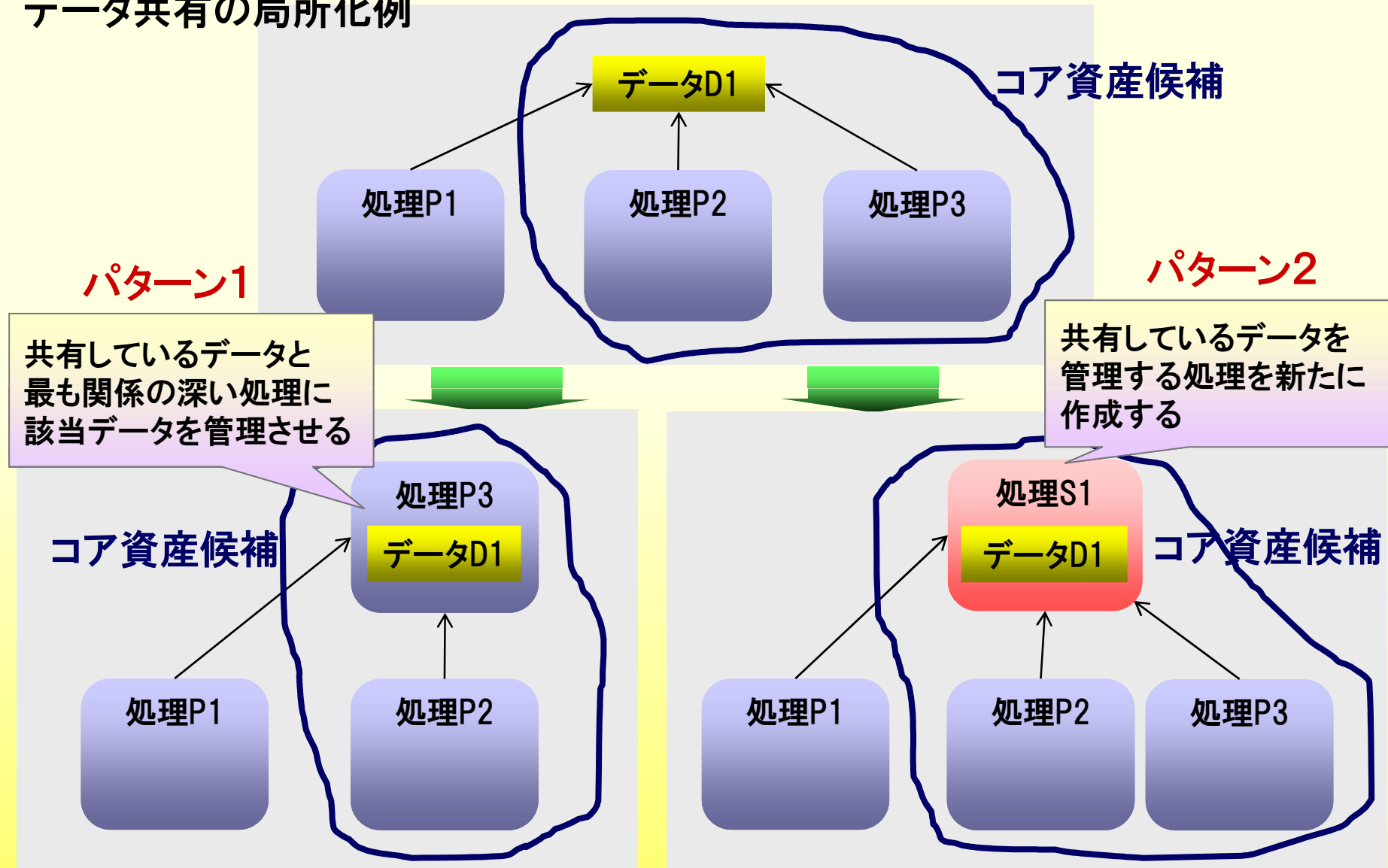
依存関係に関しては、以下の3つが重要となり、おののくに分類して局所化する:

- ① データ共有
- ② 呼出し関係
- ③ クローン関係

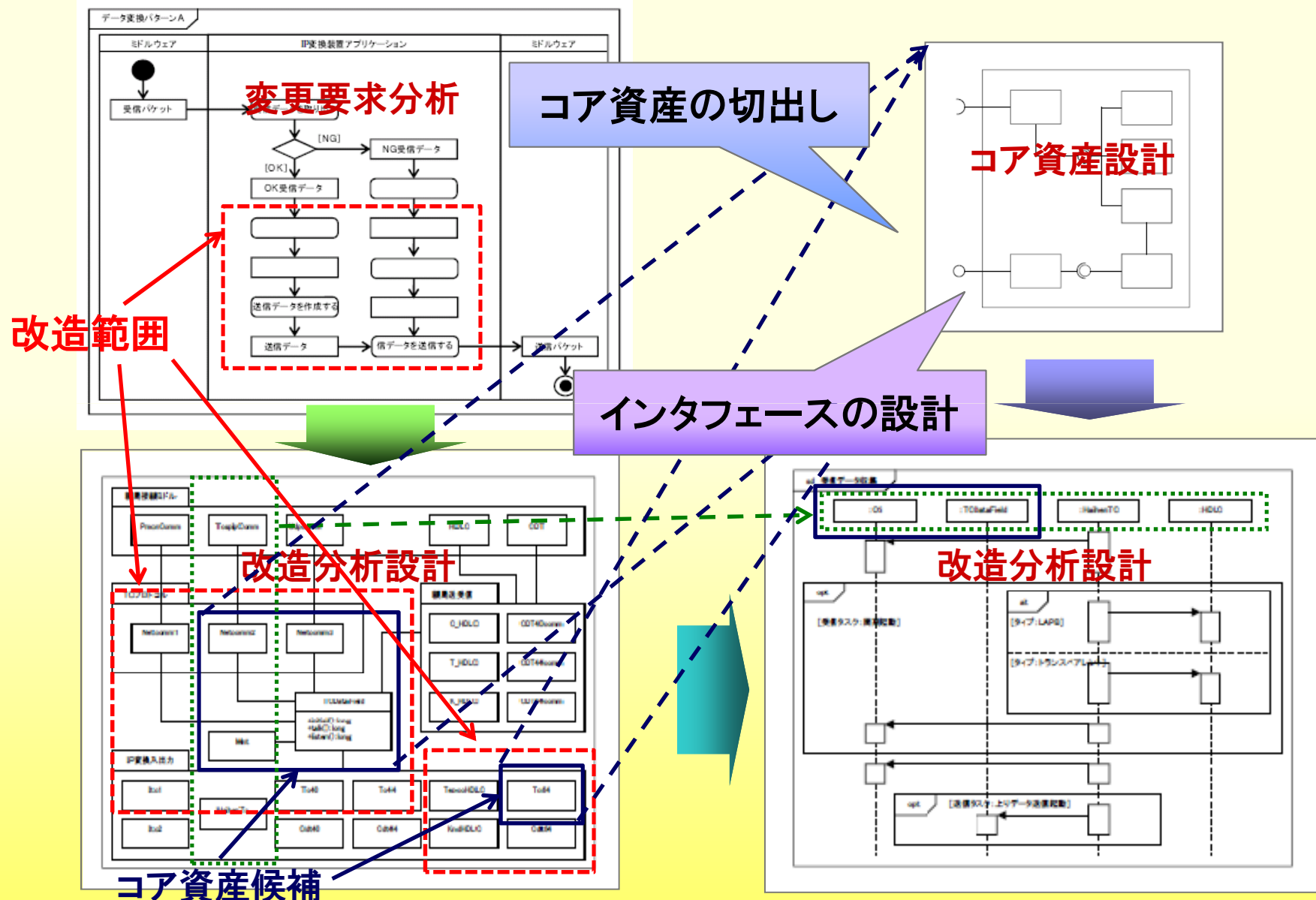


2-14. 改造分析設計(2: データ共有の局所化)

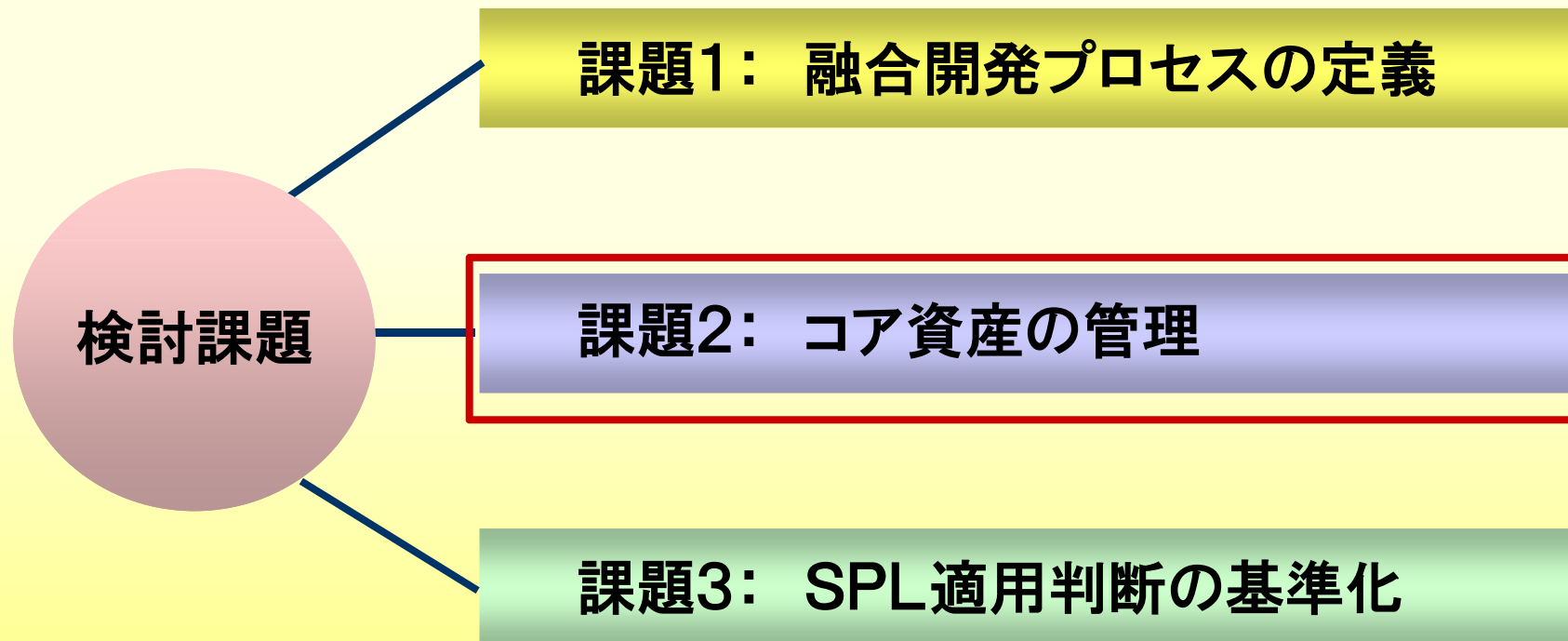
データ共有の局所化例



2-15. 改造分析設計(3: コア資産設計までの流れ)



XDDPとSPLを融合させた開発手法を確立するための検討課題

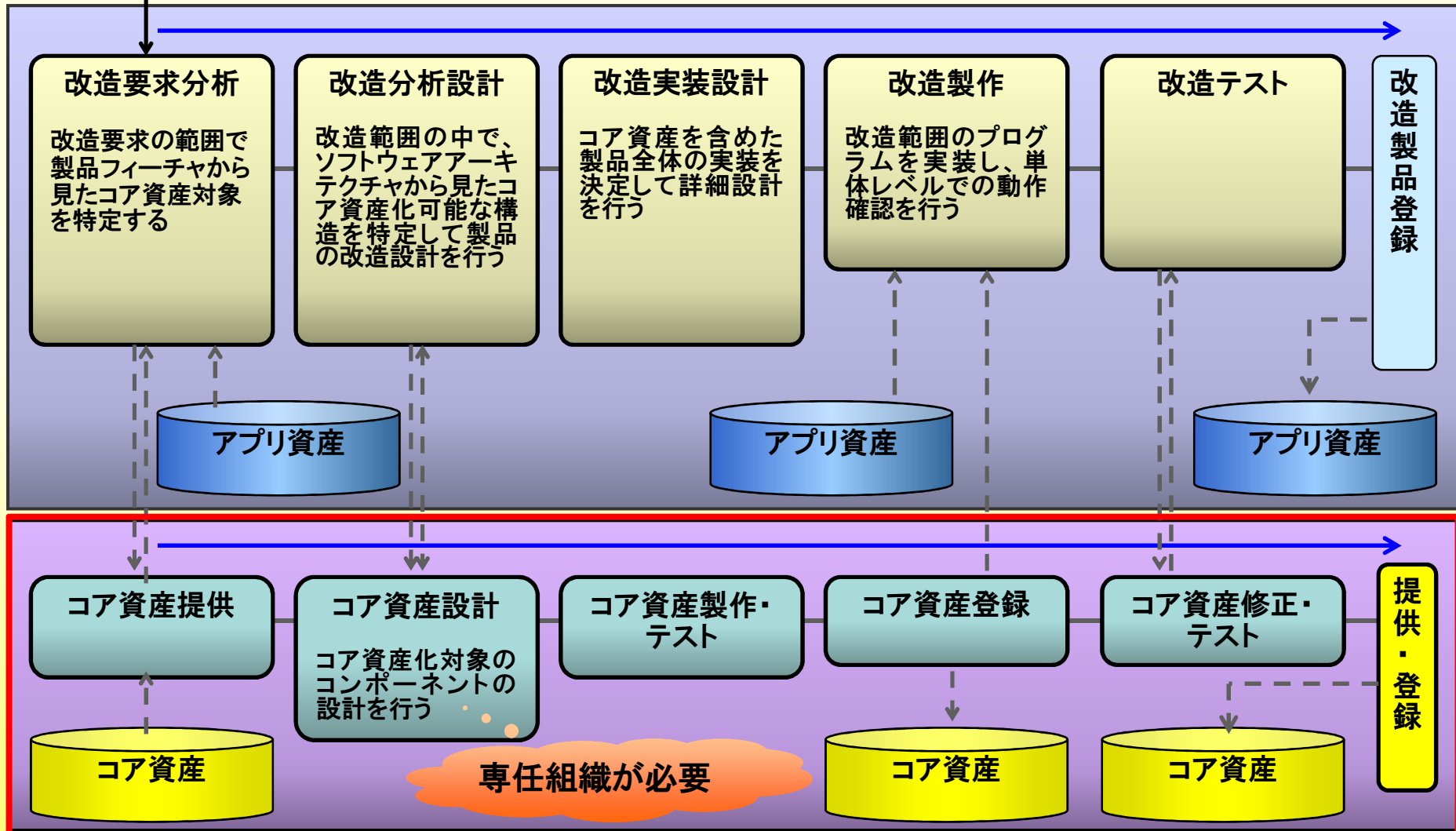


XDDP: eXtreme Derivative Development Process

SPL: Software Product Line

2-17. コア資産管理の概要

製品開発プロセス



コア資産開発・管理プロセス

コア資産管理に要求される項目

1. SPL共通の項目

製品系列全体のコア資産とアプリケーション資産を分けた管理、およびプロダクトとしての各製品の管理が必要

各製品におけるフィーチャとコア資産の対応付けの管理が必要

コア資産とアプリケーション資産から各製品をビルドする機能が必要

2. XDDP・SPL融合に特有な項目

製品の改造開発とコア資産開発を並行して進めるため、コア資産化の部分と残りの部分(アプリケーション資産)との連携した管理が必要

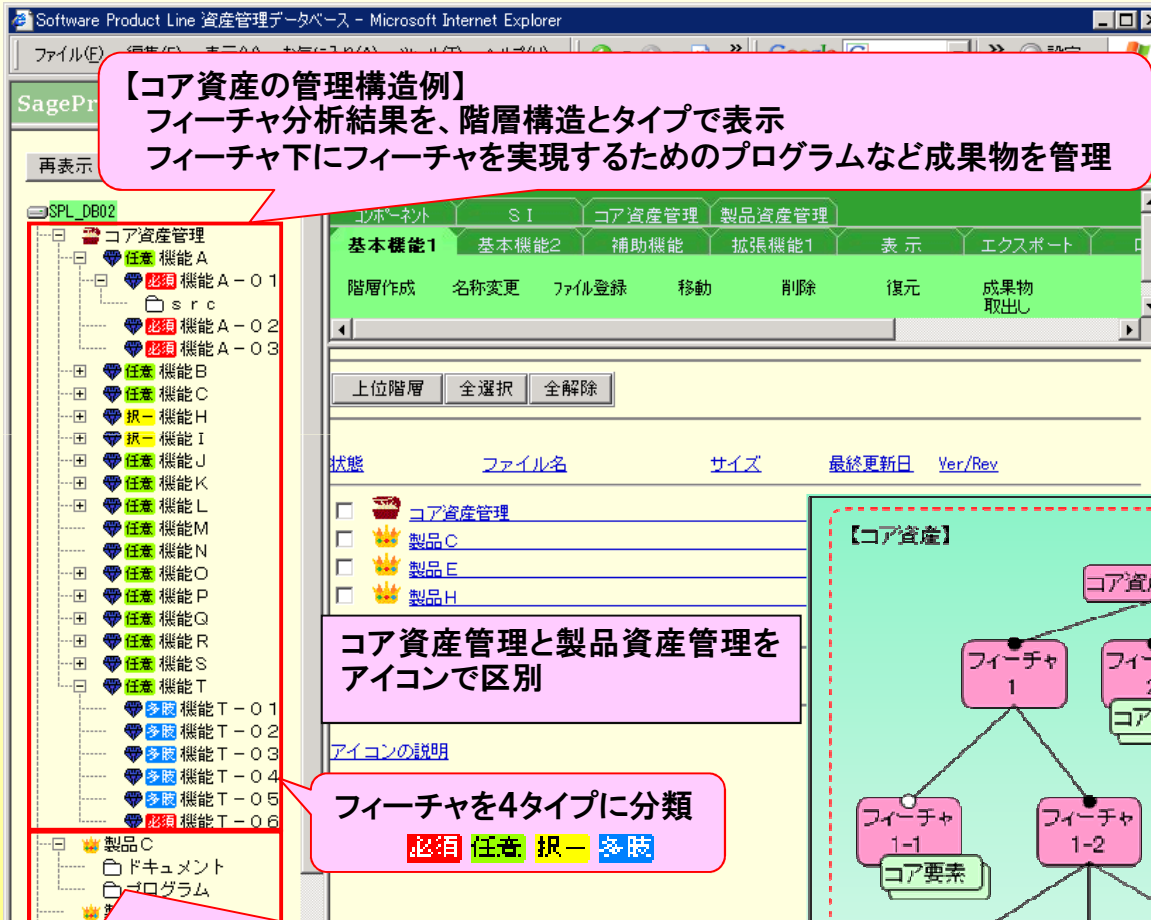
コア資産化する前の既存製品ではコア資産を使用していないため、コア資産化前の製品とコア資産化後の製品を分けた管理が必要

XDDP: eXtreme Derivative Development Process

SPL: Software Product Line

2-19. SPL支援ツールの活用: SagePro®/CM for SPL

- **コア資産管理** 製品系列群における共通の、および可変的に利用可能なソフトウェア資産を
フィーチャ分析結果の階層構造を用いて管理

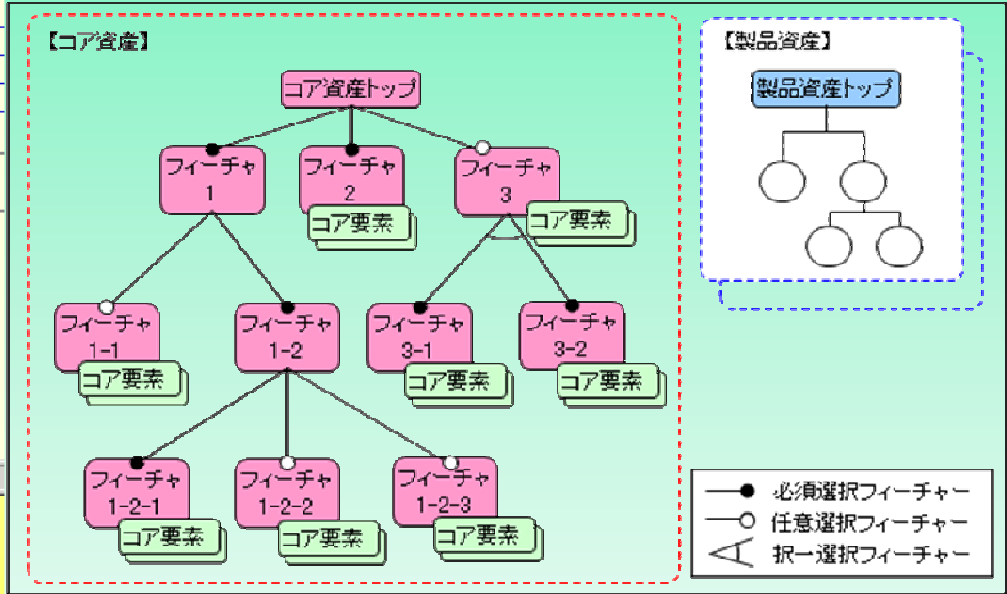
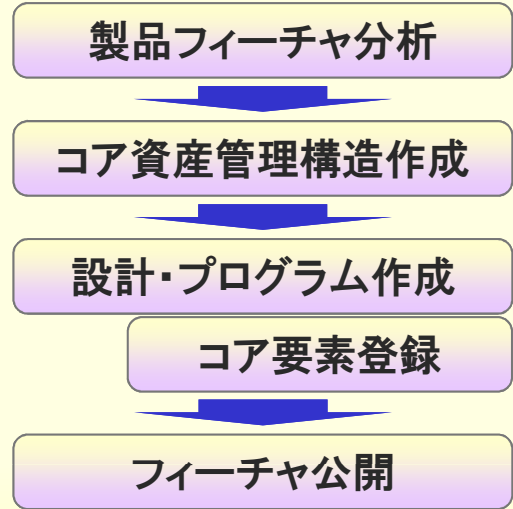


【コア資産の管理構造例】
フィーチャ分析結果を、階層構造とタイプで表示
フィーチャ下にフィーチャを実現するためのプログラムなど成果物を管理

コア資産管理と製品資産管理を
アイコンで区別

フィーチャを4タイプに分類
● 必須 任意 振- 多肢

【製品資産の管理構造例】
最上位階層のアイコンで製品資産管理であることを表示
各製品特有のプログラムなど成果物を管理



SPL: Software Product Line

2-20. SPL支援ツールの活用: SagePro®/CM for SPL

● 製品資産管理 製品間で共通に利用できるコア資産と各製品固有のソフトウェア資産を管理

コア資産管理 製品資産管理

製品資産 設定 製品資産 解除 コア資産 選択 コア要素 配布一覧 マトリックス 表示

コア資産選択

製品で利用するコア資産を設定します。
利用するフィーチャを選択して、『選択実行』ボタンをクリックしてください。

前画面 メイン画面 取消 選択実行

製品資産: 製品C

製品で使用するコア資産(フィーチャ)を選択

- ☐ コア資産管理
 - ☐ 機能A
 - ☐ 機能A-01
 - ☐ 機能A-02
 - ☐ 機能A-03
 - ☑ 機能B
 - ☑ 機能B-01
 - ☐ 機能B-02
 - ☑ 機能C
 - ☑ 機能C-01
 - ☑ 機能C-02
 - ☑ 機能C-03
 - ☑ 機能C-04
 - ☑ 機能C-05

選択フィーチャのタイプを判断し自動設定と不整合時のWarning出力

SPL資産マトリクス出力

コア資産と製品との対応関係マトリクス表を出力します。
対象の製品を選択し、『表示実行』、または『CSV出力』ボタンをクリックしてください。

前画面 メイン画面 表示実行 CSV出力

複雑な管理も
楽にできる!

表示設定

コア資産: コア資産管理

製品資産:

- ☑ 製品H
- ☑ 製品C
- ☑ 製品E

各製品と適用コア資産(フィーチャ)の関連をマトリクスで表示

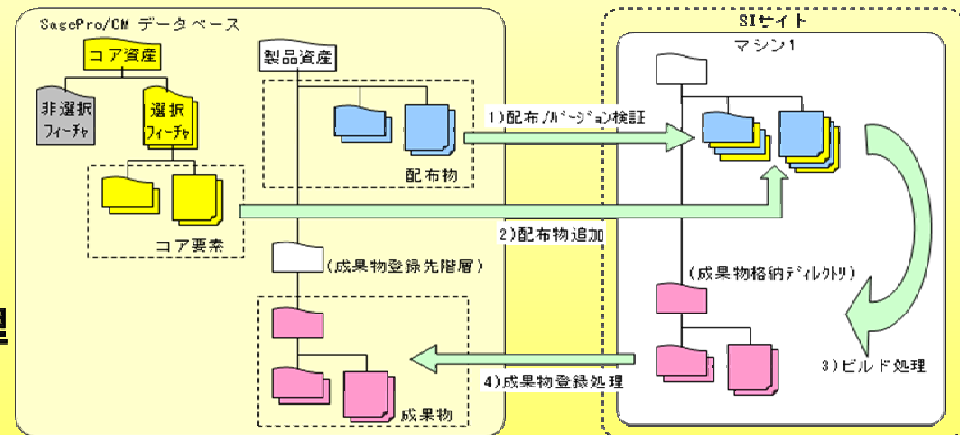
出力結果

No	コア資産管理		製品H		製品C		製品E	
	フィーチャ/コア要素	状態	V/R	選択	V/R	選択	V/R	選択
1	機能A	公開						○
2	機能A-01	公開						○
3	appframework.dll		1.1					
4	機能A-02	公開						○
5	機能A-03	公開						○

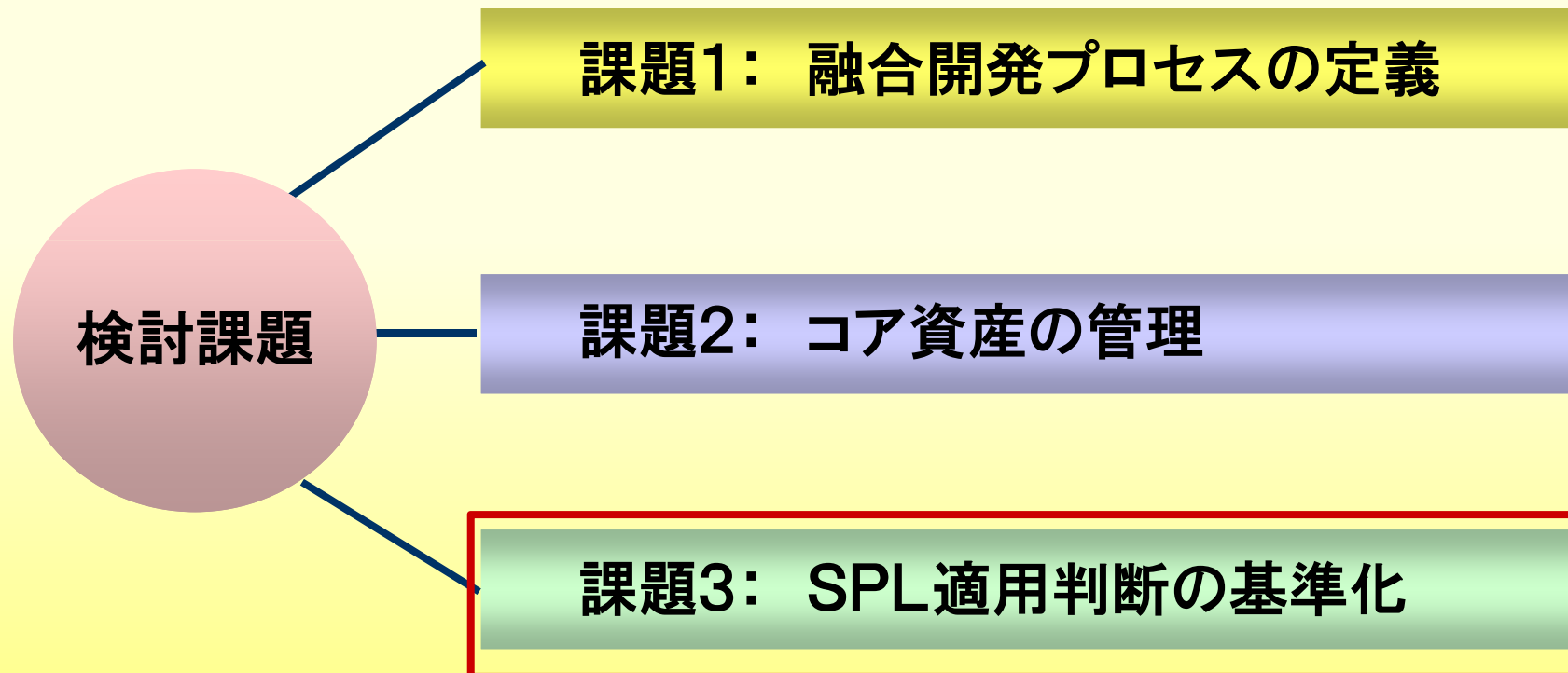
● SI管理機能

- ・製品で選択された最新のコア資産も配布
- ・ビルドで作成された成果物もVer/Rev管理
- ・テスト環境、実機環境の構築

SPL: Software Product Line



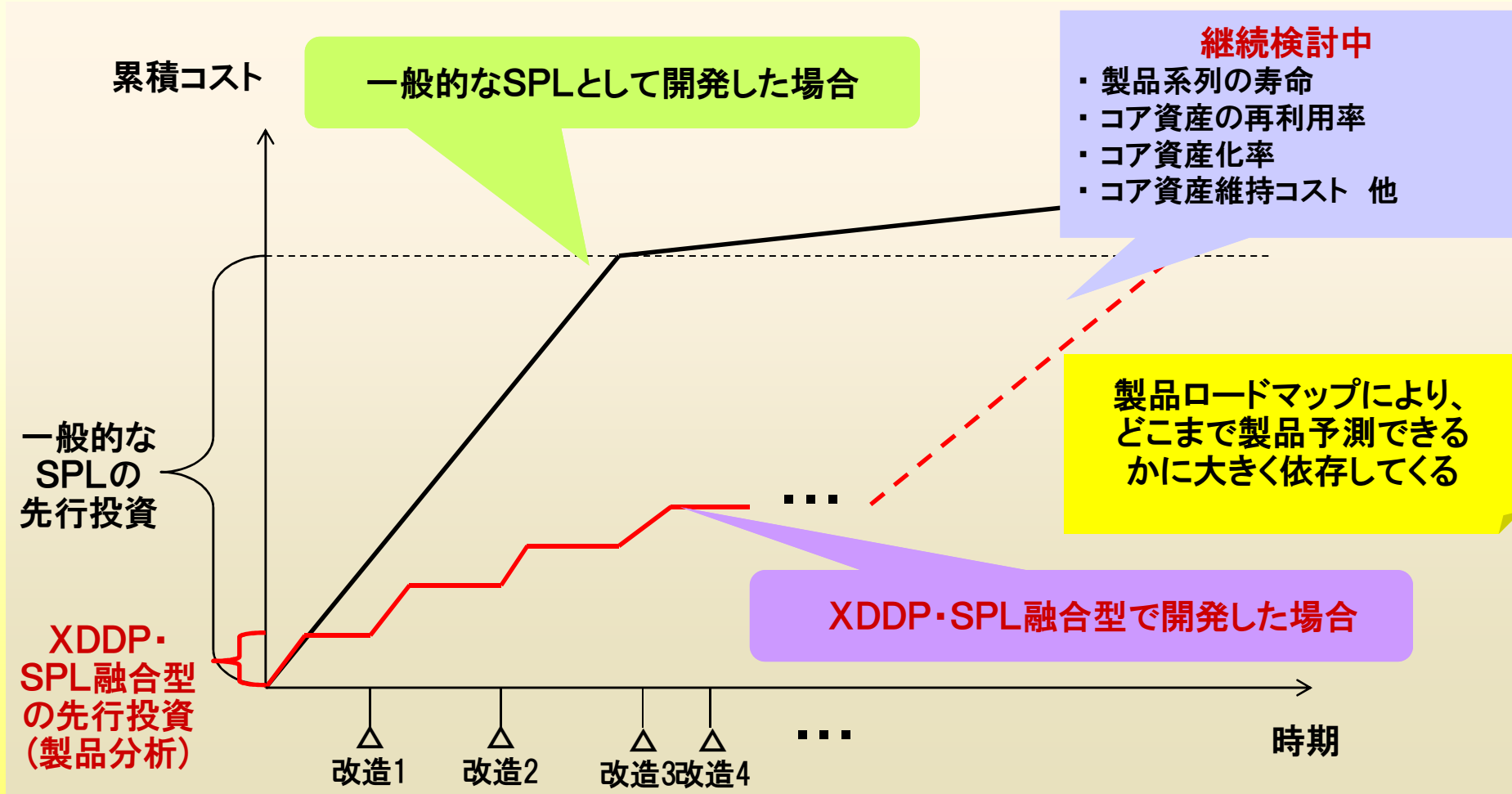
XDDPとSPLを融合させた開発手法を確立するための検討課題



2-22. 課題3: SPL適用判断の基準化

開発コストを最優先した適用判断基準

一般的なSPLとXDDP・SPL融合型の開発コスト比較



XDDP: eXtreme Derivative Development Process

SPL: Software Product Line

XDDP・SPLの融合と適用事例の紹介

Contents

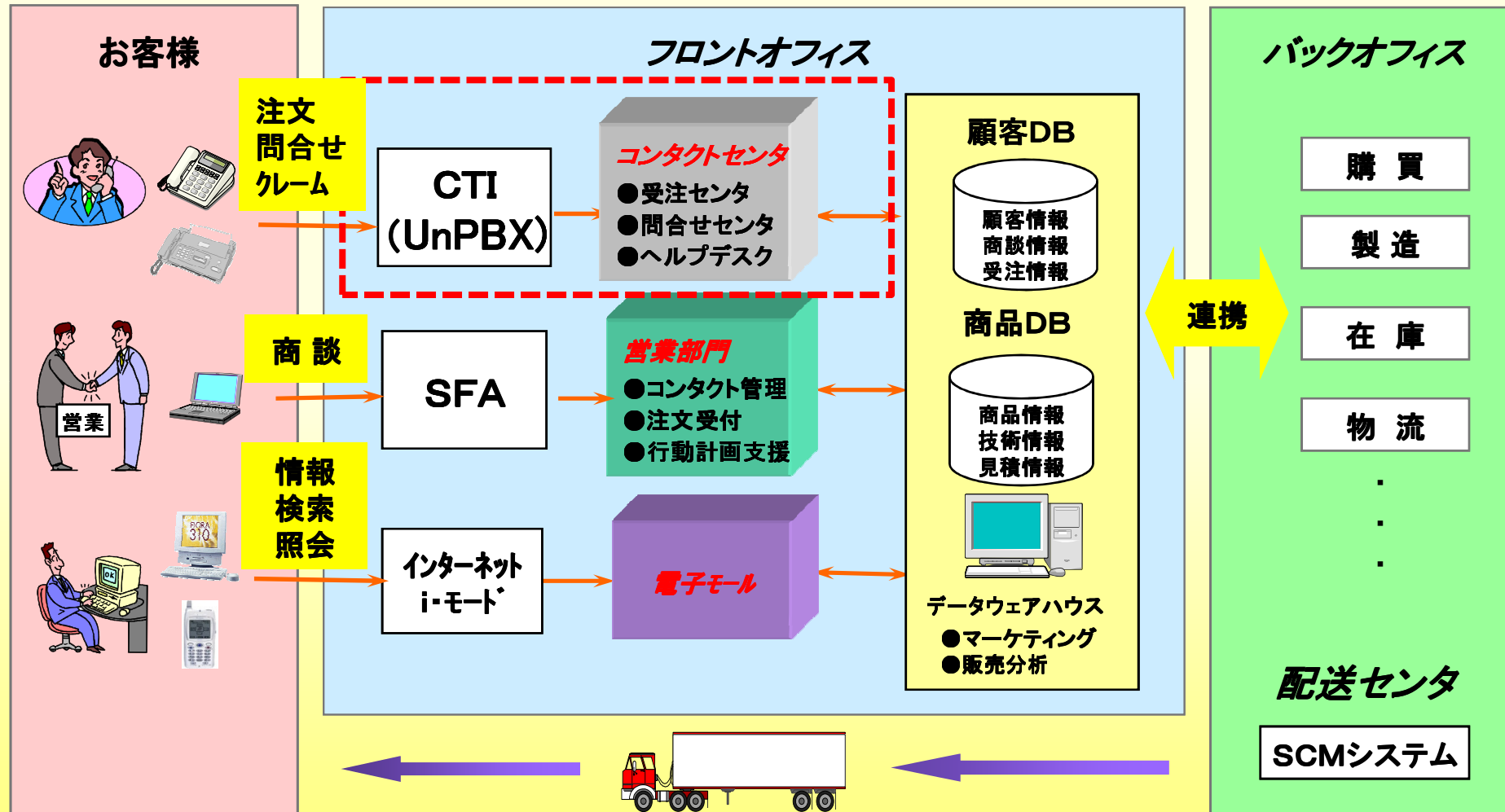
1. 当社の紹介とソフトウェア開発の状況
2. XDDPとSPLの融合とは
- ★ 3. XDDPとSPLを融合した開発事例
4. 期待できる効果
5. 今後の計画と課題

XDDP: eXtreme Derivative Development Process

SPL: Software Product Line

3-1. XDDPとSPLを融合した開発事例

CTI製品「TELEPHONYMASTER®」によるコンタクトセンタ業務の概要



CTI : Computer Telephony Integration

SFA : Sales Force Automation

SCM : Supply Chain Management

・TELEPHONYMASTERは、株式会社日立情報制御ソリューションズの登録商標です。

3-2. 開発事例：開発プロジェクトの課題

CTI製品「TELEPHONYMASTER®」開発の問題点

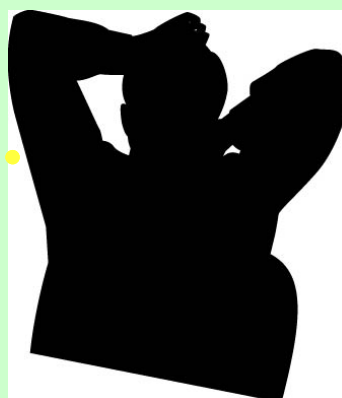
サイトごとに製品を
個別開発している

共通の製品機能が
多いが共通化できて
いない

同一機能でも顧客要
求で変化する部分
がある

電話機呼制御は基本
共通機能だが個別に
管理している

開発効率が悪く、開発
コストがかかる



類似製品が多くあり、
構成管理が複雑化
している

・TELEPHONYMASTERは、株式会社日立情報制御ソリューションズの登録商標です。

CTI製品「TELEPHONYMASTER®」開発の問題点に対する解決策

製品の共通機能と個別機能を
分けて開発すればよい

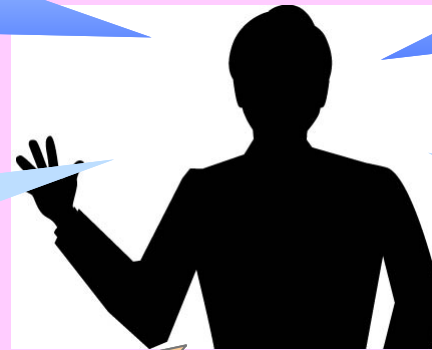
サイト間で共通の製品機能は
共通部品とすればよい

顧客要求で変化する部分は
独立させて開発すればよい

基本共通機能は唯一独立
した機能として管理すれば
よい

徐々に共通部品を増やし、
再利用性を高めて
いこう

共通部分と可変部分に
分けて開発、管理して
いけばよい



XDDPとSPLを融合させた開発手法を適用しよう！

XDDP: eXtreme Derivative Development Process
SPL: Software Product Line

・TELEPHONYMASTERは、株式会社日立情報制御ソリューションズの登録商標です。

All Rights Reserved, Copyright © 2011, Hitachi Information & Control Solutions, Ltd.

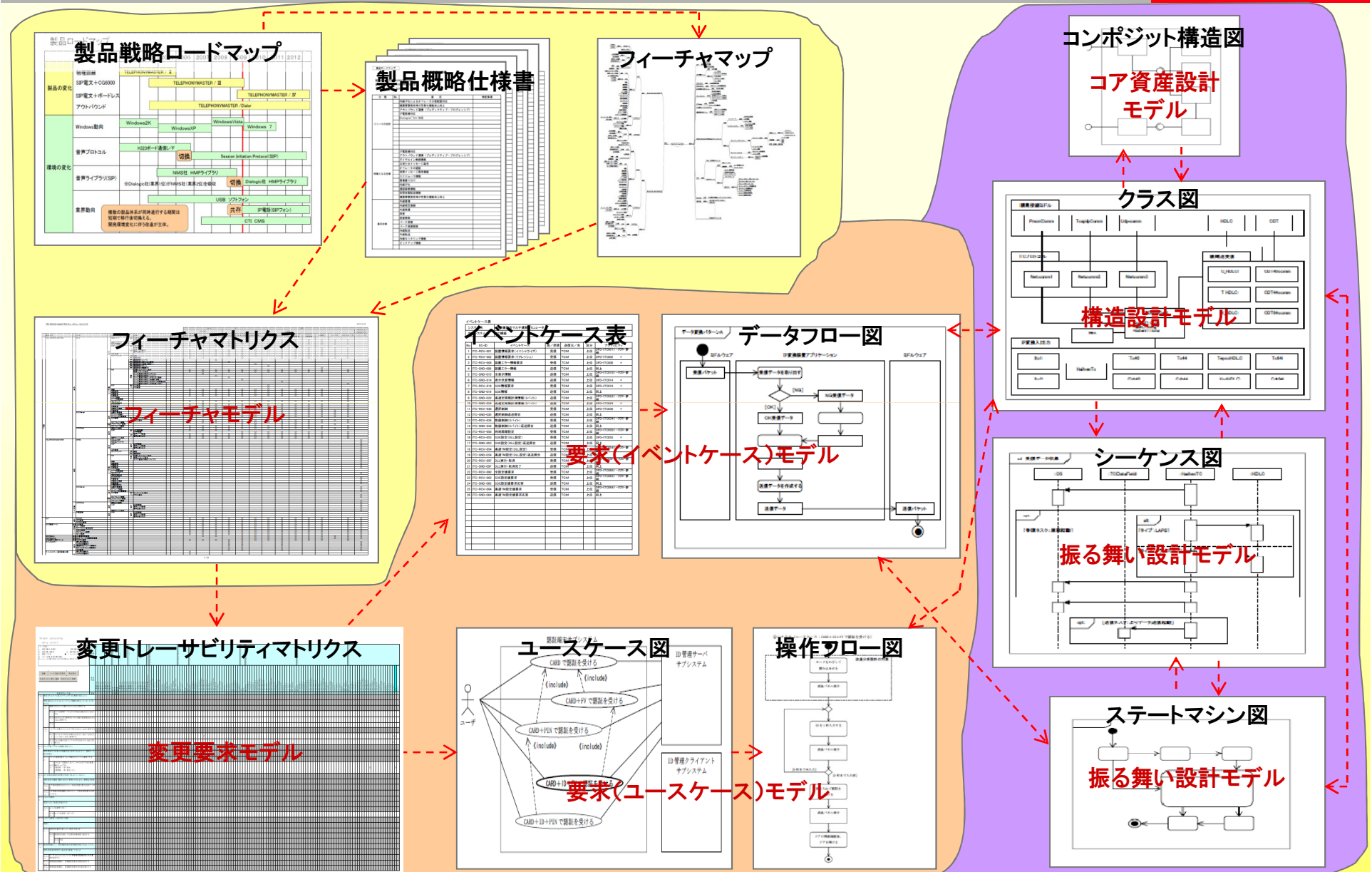
3-4. 開発事例： 開発の状況

CTI製品「TELEPHONYMASTER®」の改造開発状況

No.	改造開発名	区分	期 間	概 要	コア資産数
1	ログ出力機能強化	追加	2010年 5月～8月	製品障害解析のため、状態遷移に関するログ出力機能を強化する。	1機能
2	アウトバウンド機能 バージョンアップ	追加	2010年10月 ～ 2011年7月	他社のアウトバウンド製品に変わる自社製のアウトバウンド機能を開発する。	3機能
3	PBX機能のオープン ソース化	変更	2011年 4月～7月	購入品である呼制御ドライバから、オープンソースの呼制御モジュール群へ移行する。	1機能
4	ユニファイドコミュニ ケーション機能実装	追加	2011年9月 ～ 2012年3月	不在時対応へのボイスメール機能と電話会議サービスの動作機能を実装する。	2機能

・TELEPHONYMASTERは、株式会社日立情報制御ソリューションズの登録商標です。

3-4. 開発事例： 開発の流れ(具体的イメージ)



XDDP・SPLの融合と適用事例の紹介

Contents

1. 当社の紹介とソフトウェア開発の状況
2. XDDPとSPLの融合とは
3. XDDPとSPLを融合した開発事例
- ★4. 期待できる効果
5. 今後の計画と課題

XDDP: eXtreme Derivative Development Process

SPL: Software Product Line

4-1. 期待できる効果(定性的効果)

- 製品間の共通部品化を図り、開発量を縮減できる
- 実績のある部品を再利用し、品質を向上できる
- 製品のアーキテクチャを整備でき保守性が上がる
- 製品系列が増えると共にコスト削減を拡大できる
- アーキテクチャ中心の開発スタイルに移行できる

XDDP・SPLの融合と適用事例の紹介

Contents

1. 当社の紹介とソフトウェア開発の状況
2. XDDPとSPLの融合とは
3. XDDPとSPLを融合した開発事例
4. 期待できる効果
- ★5. 今後の計画と課題

XDDP: eXtreme Derivative Development Process

SPL: Software Product Line

CTI製品「TELEPHONYMASTER®」開発における適用計画

XDDP・SPLの融合型開発を製品開発プロジェクトに継続適用

モデル指向開発により、アーキテクチャの整備を目指す

製品の改造開発におけるコア資産化と適用評価

コア資産拡大により、製品開発のコスト低減を目指す

2010年: コア資産化開始
2011年: コア資産化継続中
2012年: コア資産化継続、10%低減
2013年: コア資産化継続、30%低減



XDDP: eXtreme Derivative Development Process
SPL: Software Product Line

・TELEPHONYMASTERは、株式会社日立情報制御ソリューションズの登録商標です。

■ XDDPとSPLを融合させた開発手法の確立と公開

■ XDDPとSPLを融合させた開発への支援ツールの適用

■ SPL導入が必要な派生開発プロジェクトへの適用拡大

■ 評価基準の策定と適用プロジェクトに対する継続的評価

END

XDDP・SPLの融合と適用事例の紹介

2011.11.7

株式会社 日立情報制御ソリューションズ

業務サポート本部

生産技術部 ソフトウェアエンジニアリンググループ

渡辺 滋

XDDP: eXtreme Derivative Development Process

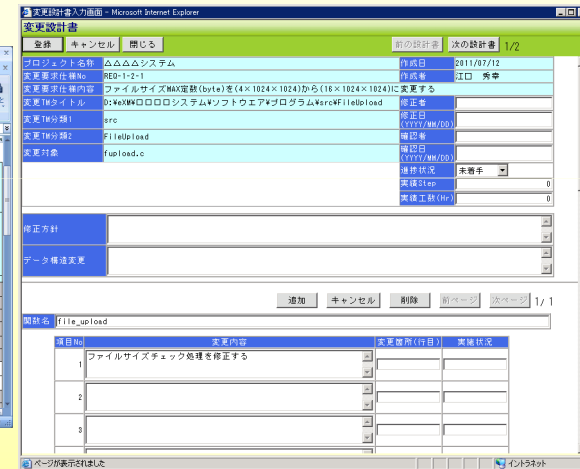
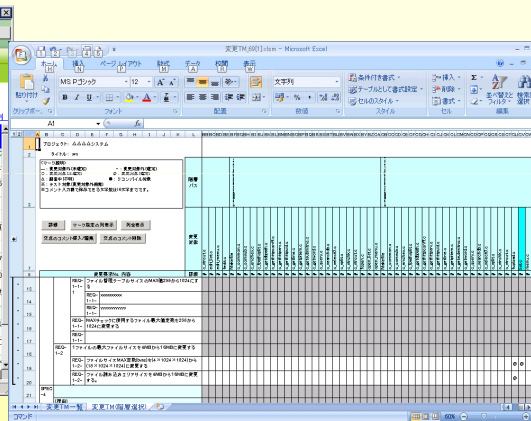
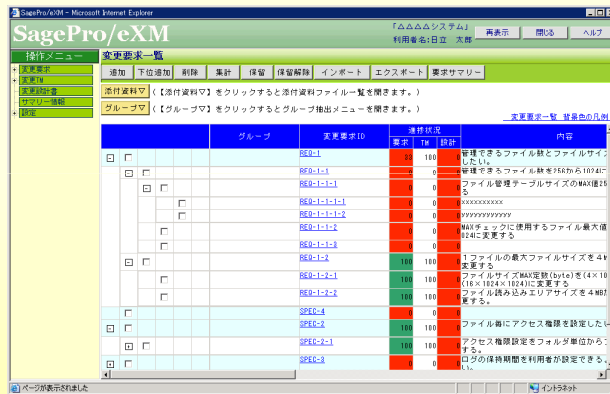
SPL: Software Product Line

**開発
コンセプト**

『派生開発プロセス「XDDP」で行う要求仕様の抽出から
変更設計書作成までの、**設計者の負担を軽減**』

◆ **変更要求仕様書、変更TM、
変更設計書作成支援**

◆ **効率的なレビュー
実施支援**



2010年4月より社内で本格適用開始

◆ **変更TMの行/列の単位で
変更設計書を参照できる**

継続的に
使い勝手を
改善

◆ **変更TM同時操作時の
ロック粒度を交点単位とする**

XDDP: eXtreme Derivative Development Process

補足. XDDP支援ツールの活用: SagePro®/eXM

● 変更要求・変更要求仕様管理

XDDP: eXtreme Derivative Development Process

変更要求/要求仕様のグループや作業進捗などを表示

変更要求と要求の背景・目的を入力

変更要求仕様をbefore/after形式で入力

● 変更TM管理

変更要求仕様

● 変更設計書管理

変更内容や確認項目などを入力

変更確定マークの設定で変更設計書作成対象となる