

日本における *SPI* の最新動向

— 派生開発 (*XDDP*) への取り組み —

(株) デンソー技研センター
プロジェクト・マネージャー

古畑 慶次

kkobata@ndtec.denso.co.jp

DENSO

2011.11.02 上海フォーラム @上海・科学会堂新館(思南楼)

Confidential

© DENSO CORPORATION All rights reserved.

This information is the exclusive property of DENSO CORPORATION. Without their consent, it may not be reproduced or given to third parties.

古畑 慶次 (こばた けいじ)

＜所属＞：(株)デンソー技研センター
プロジェクト・マネージャー / 産業カウンセラー

＜業務＞： **現場支援・指導**、研修企画、研修講師
－ 派生開発、プロセス設計、仕様化技術の技術支援・指導
－ 高度技術者(トップガン)研修の企画、運営

＜略歴＞

1988年：(株)デンソー 入社 (日本電装株式会社)
研究開発部 … デジタル信号処理

1990年：基礎研究所 … 音声認識

1994年：通信技術部 … ソフトウェア設計 / ハードウェア
(携帯電話 / PHS 親機)

2002年：ITS 技術部 … プロセス改善 (CMM / 現場改善)

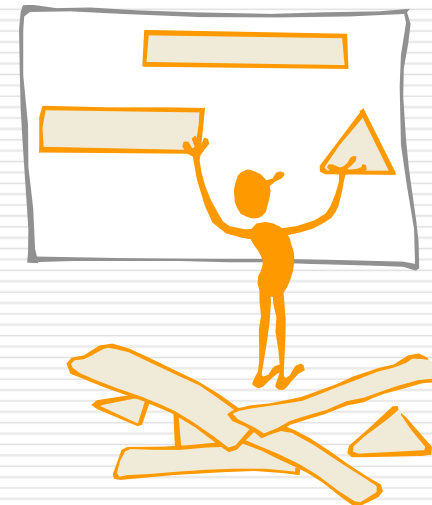
2004年：デンソー技研センター <現在に至る>

- **新規開発**をここ数年したことがない
- 最近**既存のソースコード**に基づいた開発が多い
- 既存のソースコードの**変更箇所の影響**に苦労している
- *SPLE* などの**新しい技術**に取り組む時間がない
- 現場の技術者は**疲弊**している

本日は、日本で始まった**派生開発**への
取り組み(**XDDP**)を紹介します

SPLE : Software Product Line Engineering

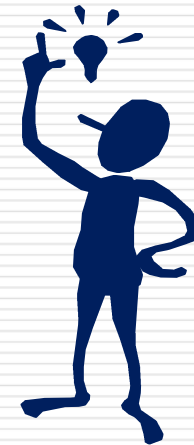
1. 日本のソフトウェア開発
2. 概説 *XDDP* (*eXtreme Derivative Development Process*)
3. 開発現場での取り組み
4. 最新の技術動向
5. まとめ
6. 参考文献



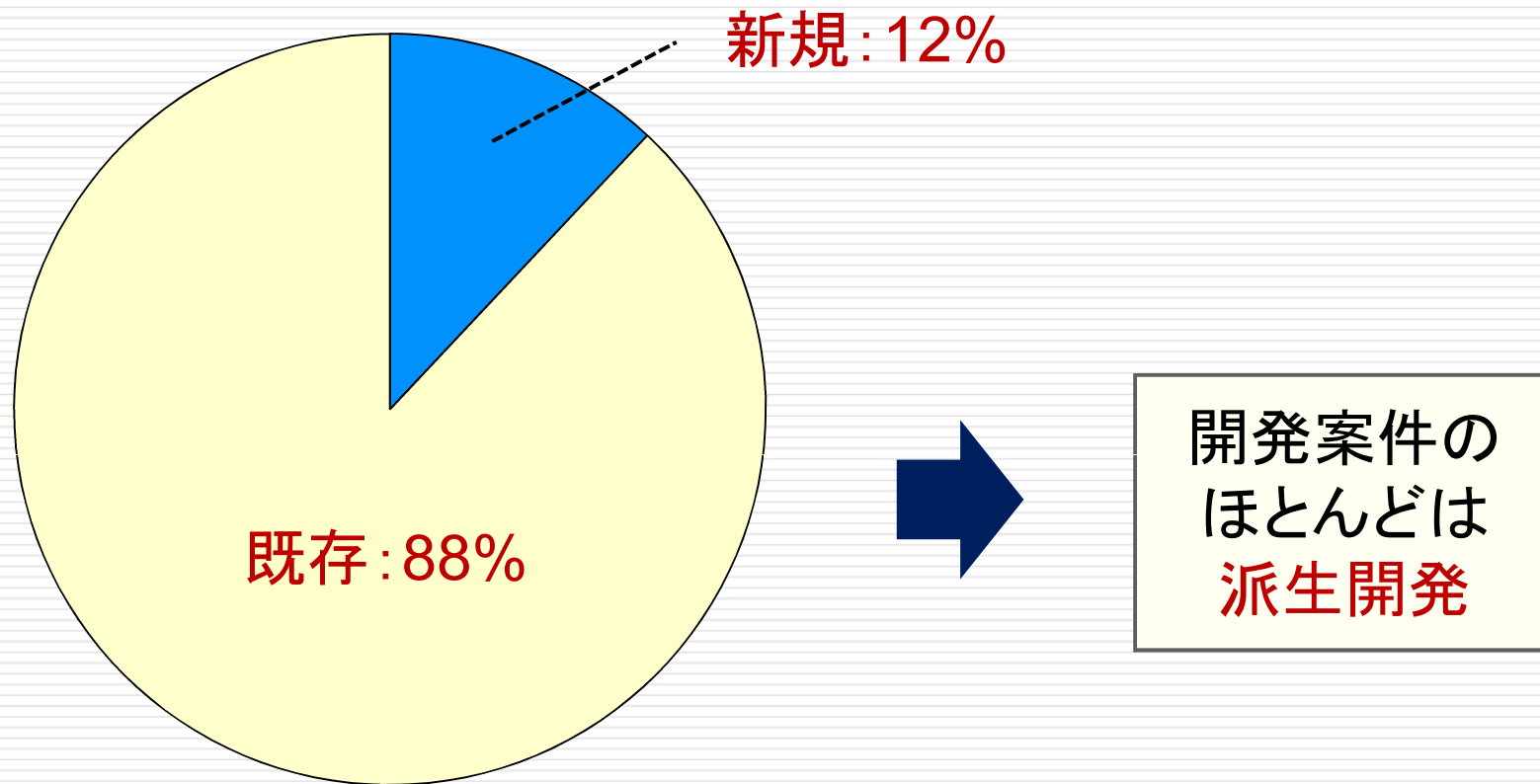
1. 日本のソフトウェア開発

4 / 36

- 1.1 日本における開発案件
- 1.2 ソフトウェア開発の現状
- 1.3 開発現場の実際
- 1.4 派生開発の現実（1）
- 1.5 派生開発の現実（2）



1.1 日本における開発案件



【 2009年度 開発行数 】

出典：独立行政法人情報処理推進機構
2009年組込みソフトウェア産業実態調査

- **派生開発**によるシステム開発
 - 組込み系、パッケージソフト、制御系ソフト、エンタプライズ系
 - 大規模化したシステムでの派生開発は**困難を極めている**
- **混乱する開発現場**
 - 結合テストやシステムテストで**大量の不具合**が発生
 - 技術者が自分の時間で**献身的にバグ**に対応
 - 次の開発のスタートが遅れ、負のスパイラル(**デスマーチ**)へ
- **企業のビジネス展開**
 - どこかで製品やシステムが**リリースできなくなる可能性**
 - *SPL* など競争上避けて通れない**取り組みにも支障**をきたしている
 - 今回の **3.11**で、競争上 **QCD** の同時達成は必至

- 現場の技術者
 - 派生開発の混乱に振り回され疲弊している
 - 劣化したコードと格闘しながら、なんとかリリースしている
 - 精神的肉体的ダメージから倒れる技術者も・・・（社会問題）
- 技術者の悲鳴
 - 肉体的、精神的に常に高負荷を強いられ疲弊状態
 - 不具合と仕様変更の対応で技術習得、現状分析の時間が取れない
 - 技術力、モチベーションの低下

1.4 派生開発の現実（1）

■ 新規開発崩し

- 派生開発に**新規開発のプロセス**を適用している
- 要求仕様書から設計書、ソースコードを**完成させて開発**する
 - 変更点、影響範囲の抽出が困難なため抜け、漏れが発生

派生開発に必要な情報 = 派生元に対する**差分情報**

■ 時間が足りない

- 必要な**プロセスを省いて**ソースコードに手をつける
 - レビュー、テスト、ドキュメントの作成 等
- 変更箇所は**担当者任せ**になっている
 - 変更点はすべてドキュメントに記載されるわけではない
- 拙速なソースコードの修正
 - ソースコードに手をつけなければ間に合わないという**プレッシャー**
 - ソースコード上で**変更箇所を見つけ次第修正**する

- 「**思い込み**」と「**勘違い**」
 - 必ずしも ”**確信**” を持ってソースコードを変更していない
 - 担当範囲によっては、”**変更すべき箇所**” に全く**気づかない**時もある
 - 従来の手法では、適切に**変更箇所を抽出する方法**がない
 - 「ソースコードをすべて理解しなければいけない」という**呪縛**
- 「**部分理解**」が前提
 - システム全体を理解するような**時間はない**
 - ドキュメントに必要な仕様がすべて**記述されているとは限らない**
 - 結局、ソースコードを読んで理解するしかなくなる
 - **ソースコード**を理解するには**技術**が必要
 - **保守性**は無視され、**構造**が読み取れない
 - 「**無知見プロジェクト**⁽¹⁾」「**完全無知見プロジェクト**⁽²⁾」も存在する

無知見プロジェクト⁽¹⁾ : ソースコードに対する知見が不十分な状態での開発

完全無知見プロジェクト⁽²⁾ : ソースコード、開発対象の知識、開発ドキュメントが不十分な開発

2.1 *XDDP* とは？

2.9 導入の効果 – 品質・生産性の改善 –

2.2 *XDDP* を支える技術

2.10 担当者の声 – 導入した現場から –

2.3 従来方法との比較

2.11 まとめ

2.4 追加と変更のプロセス

2.5 変更プロセスの3点セット

2.6 *XDDP* のプロセス

2.7 品質を確保するメカニズム(1)

2.8 品質を確保するメカニズム(2)

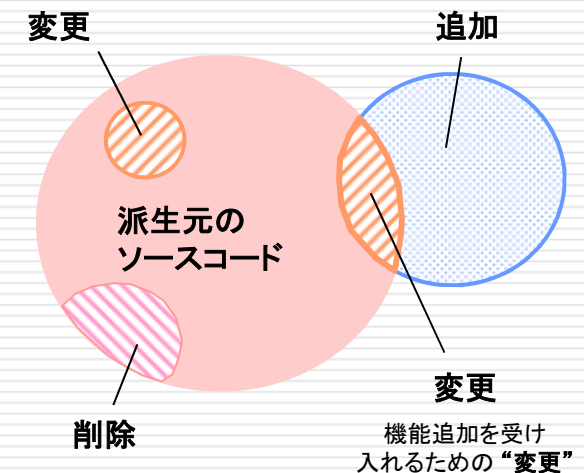


2.1 XDDP とは？

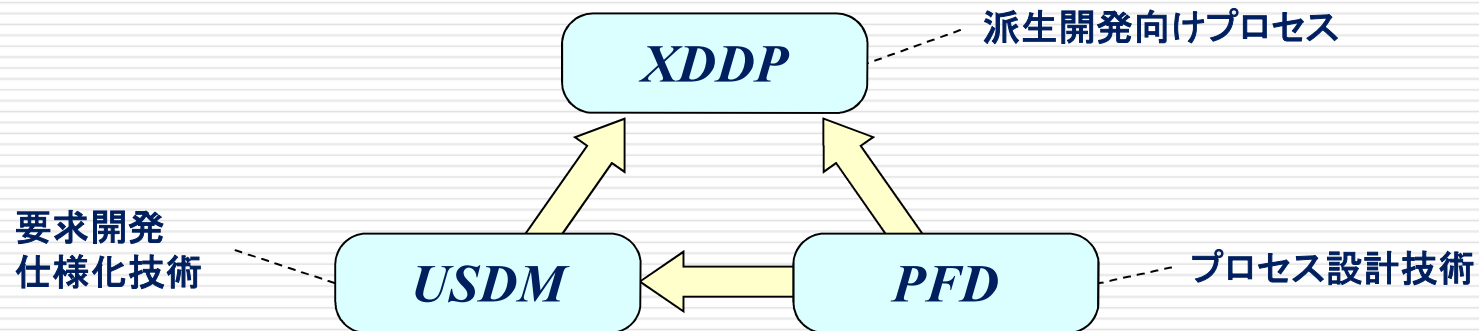
- **XDDP** : *eXtreme Derivative Development Process*
 - 派生開発の要求にあった合理的な開発プロセス
 - 清水吉男氏(システムクリエイツ)が提案
 - 「派生開発」を成功させるプロセス改善の技術と極意:技術評論社(2007)



- プロセスの特徴
 - 派生開発のための開発プロセス
 - 変更のプロセスと追加のプロセスを独立で扱う
 - 品質と *Agility* を追求したプロセス
 - 差分情報に基づいた開発
 - ムダの徹底排除と合理性の追求 (*Just in Time*)
 - 「必要な時に」「必要なことを」「必要なだけ」
 - 「思い込み」と「勘違い」を前提にする
 - 担当者の ”部分理解” をレビューでカバー
 - 適切な成果物を作成し、レビューを効果的に活用する



■ XDDPトライアングル



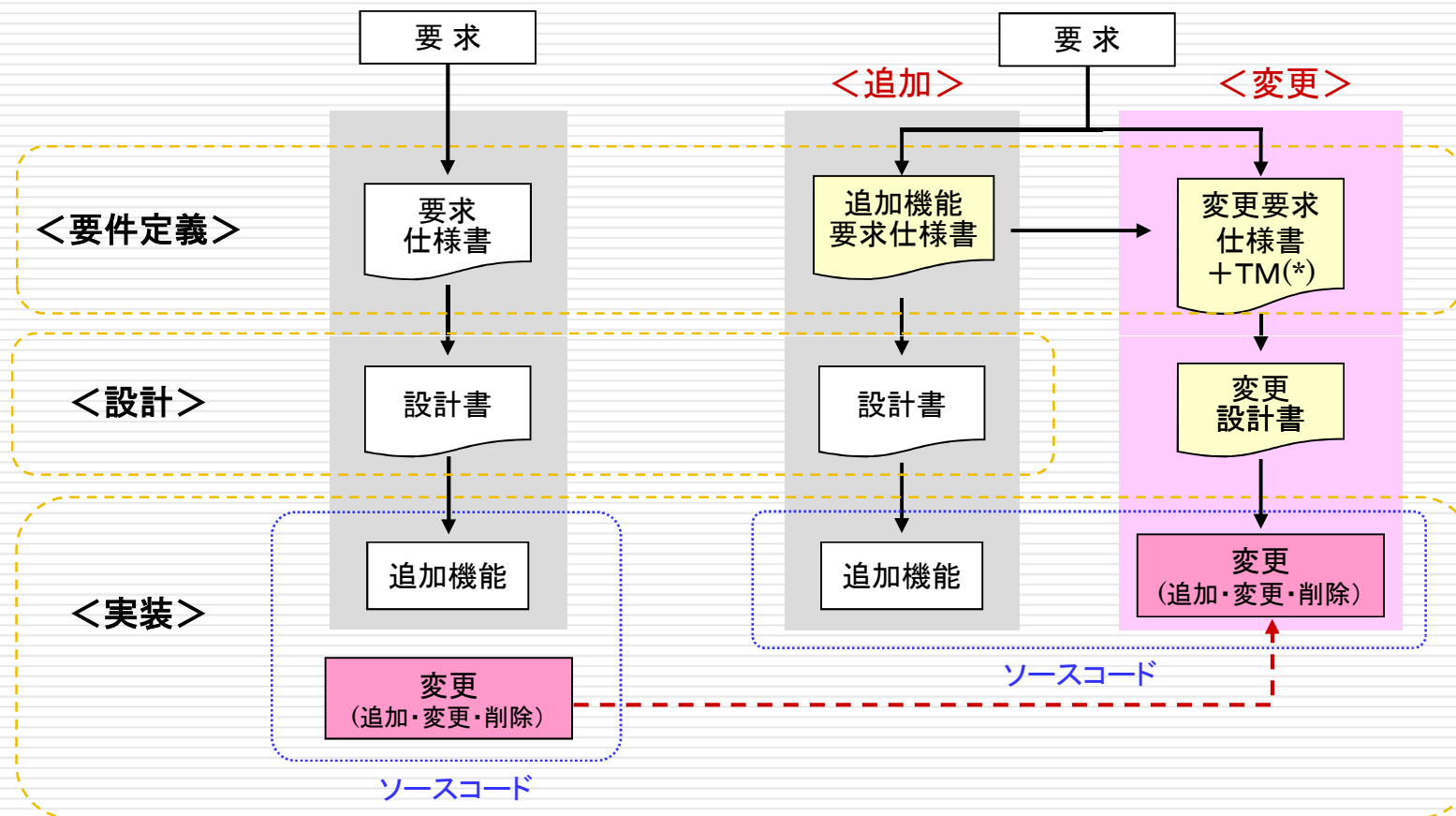
- **USDM** (*Universal Specification Describing Manner*)
 - 要求 (追加/変更)を適切に表現し、必要な仕様を引き出す方法
 - 要求と仕様を階層関係の中で捉え、仕様が漏れにくい構成
 - XDDPでは、変更要求仕様書、追加機能要求仕様書を USDM で作成
- **PFD** (*Process Flow Diagram*)
 - 要求を実現する合理的な“成果物とプロセスの連鎖”を設計する技術
 - 開発途中の変化に対して、開発アプローチの「別案」を考え出すツール

2.3 従来方法との比較

13 / 36

【 従来の派生開発 】

【 XDDP 】



TM(*) : トレーサビリティ・マトリックス

2.4 追加と変更のプロセス

14 / 36

■ 追加のプロセス

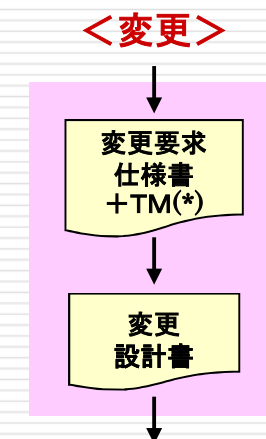
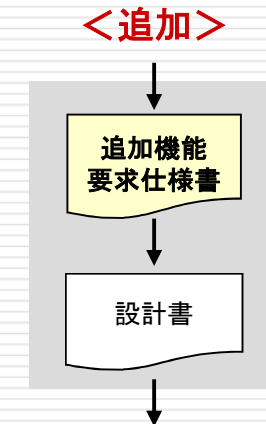
- 一般の**新規開発のプロセス**に準拠したプロセス
- 通常は「**アーキテクチャ設計**」のプロセスが存在しない
 - 但し、既存のアーキテクチャの制約は受ける
- 追加機能要求仕様書は**新規開発**と基本的に同じ

■ 変更のプロセス

- 派生元のソースコードの**差分情報**に着目したプロセス
 - 追加のプロセスより複雑で厄介な作業が強いられる
 - ソースコードに対する**変更点のみ**に関する**成果物**を作成する

<変更の3点セット>

- (1) 変更要求仕様書
- (2) *TM* (トレーサビリティ・マトリックス)
- (3) 変更設計書



TM(*) : トレーサビリティ・マトリックス

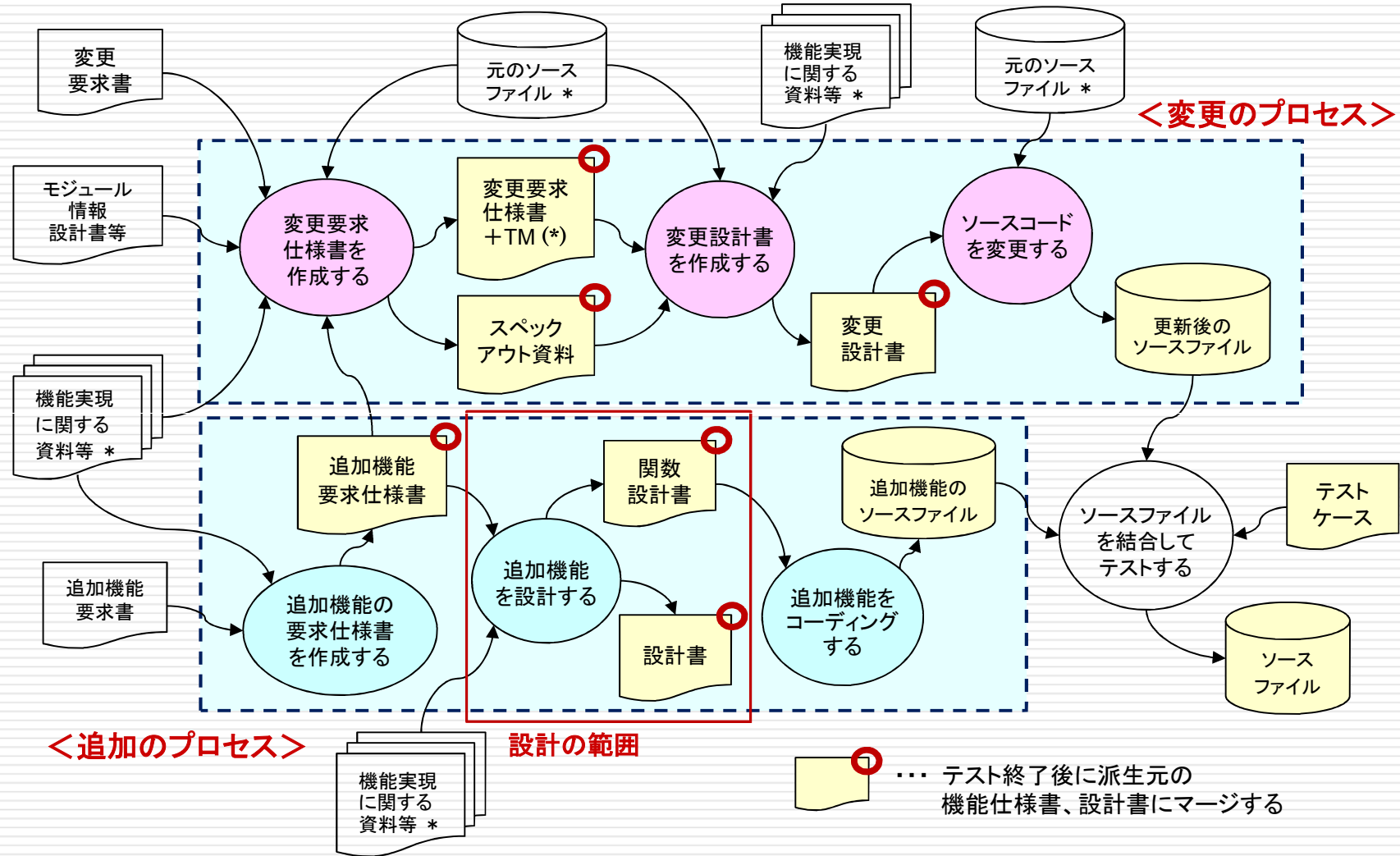
2.5 変更プロセスの3点セット

- **XDDP の変更プロセスでは3つの成果物を作成する**
 - 変更要求仕様書 / **TM** (トレーサビリティ・マトリックス) / 変更設計書
 - **作成するタイミング**が異なるので**効果的なレビュー**の機会を確保できる
 - 担当者の「部分理解」をレビューでカバーする
 - 3点セットは開発終了後に**正式文書にマージ**する
 - 各成果物は、変更箇所の記録として保存する

成果物	カバー範囲	内容	レビュー(*)	
変更要求仕様書	What Why	何をなぜ変更するか どのような振る舞いをするか	○	○
TM	Where	変更仕様がどこにあるか	○	
変更設計書	How	変更仕様をどのように修正するか	○	○

レビュー(*) : 変更要求仕様書 と **TM** は一緒にレビューをしてもよい

2.6 XDDP のプロセス

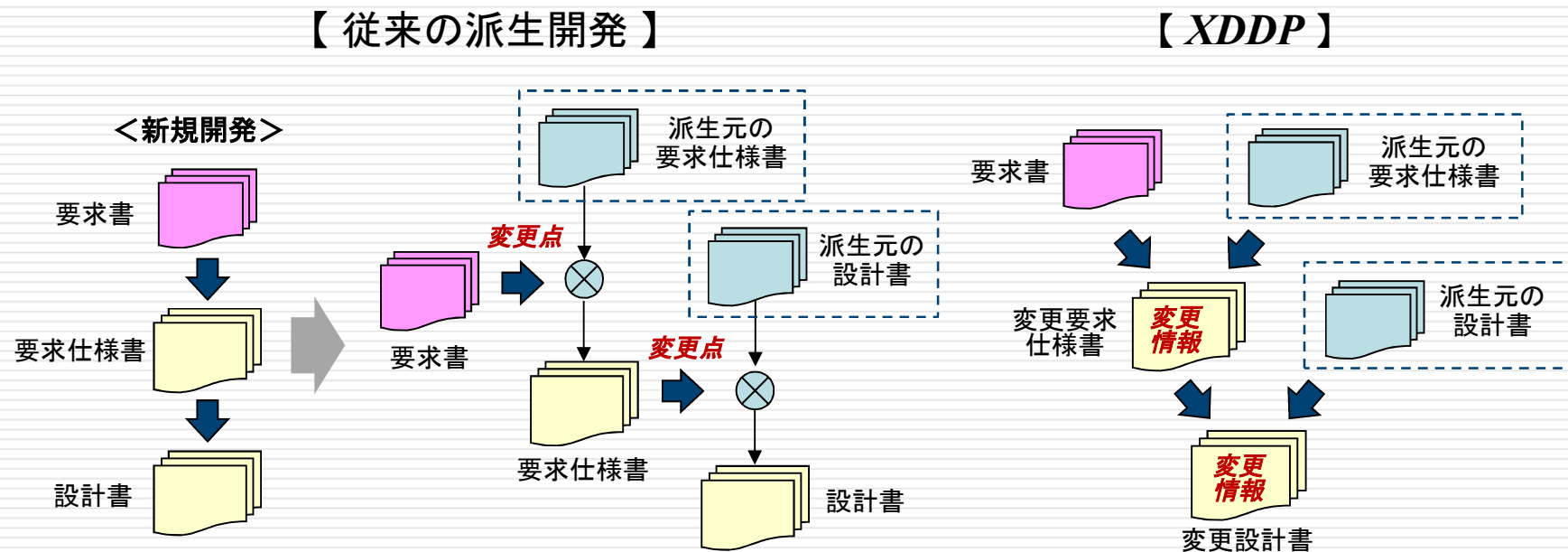


TM(*) : トレーサビリティ・マトリックス

2.7 品質を確保するメカニズム (1)

17 / 36

- **差分情報**を表現した**成果物**が品質、生産性を確保する
 - 派生開発で扱うのは派生元のソースコードに対する**差分情報**
 - 完成後の仕様書、設計書では**変更点**、**影響範囲**の抽出が困難
 - ドキュメントを**完成させること**に設計者の注意が払われる
 - 新規開発のプロセスでは、**変更点の反映と抽出を繰り返している**

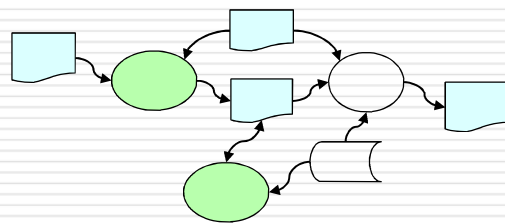


■ プロセス、仕様、ソースコードの観点から変更点を抽出する

- (1) 最適な開発プロセスの設計 (PFD)
- (2) 抜け漏れのない要求の仕様化 (USDM)
 - 追加機能要求仕様書／変更要求仕様書
- (3) ソースコードの変更点の抽出
 - TM (トレーサビリティ・マトリクス)
- (4) ソースコードの変更方法の確認
 - 変更設計書

→

- ・ 全変更点の文章化
- ↓
- ・ 効果的なレビュー



(1) プロセス設計 (PFD)

(2) 要求の仕様化 (USDM)

要求	Req.1	
	理由	
	説明	
	<Group 1>	
要求	Req.1-1	
	理由	
	説明	
仕様	<グループ A>	
	Req.1-1-1	
	Req.1-1-2	
	<グループ B>	
	Req.1-1-3	
	Req.1-1-4	
	<Group 2>	
要求	Req.1-2	
	理由	
	説明	
仕様	<グループ C>	
	Req.1-2-1	
	Req.1-2-2	
	<グループ D>	
	Req.1-2-3	
	Req.1-2-4	

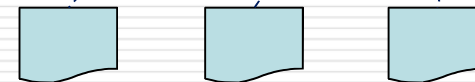
- ・ 追加機能要求仕様書
- ・ 変更要求仕様書

(3) ソースコードの変更点の抽出 (TM)

#	変更要求・仕様	A	B	C	D	E	F	G	H
5	画面に通信記録の表示を追加する		●						
5.1	接続状況の表示の大きさを・・に変更する							●	
.	...					●			
5.4	表示用メモリの配置を・・に変える								
5.5	受信時データの区切りにコードを挿入する								

↓

変更設計書



(4) ソースコードの変更方法の確認

2.9 導入の効果 - 品質・生産性の改善 -

19 / 36

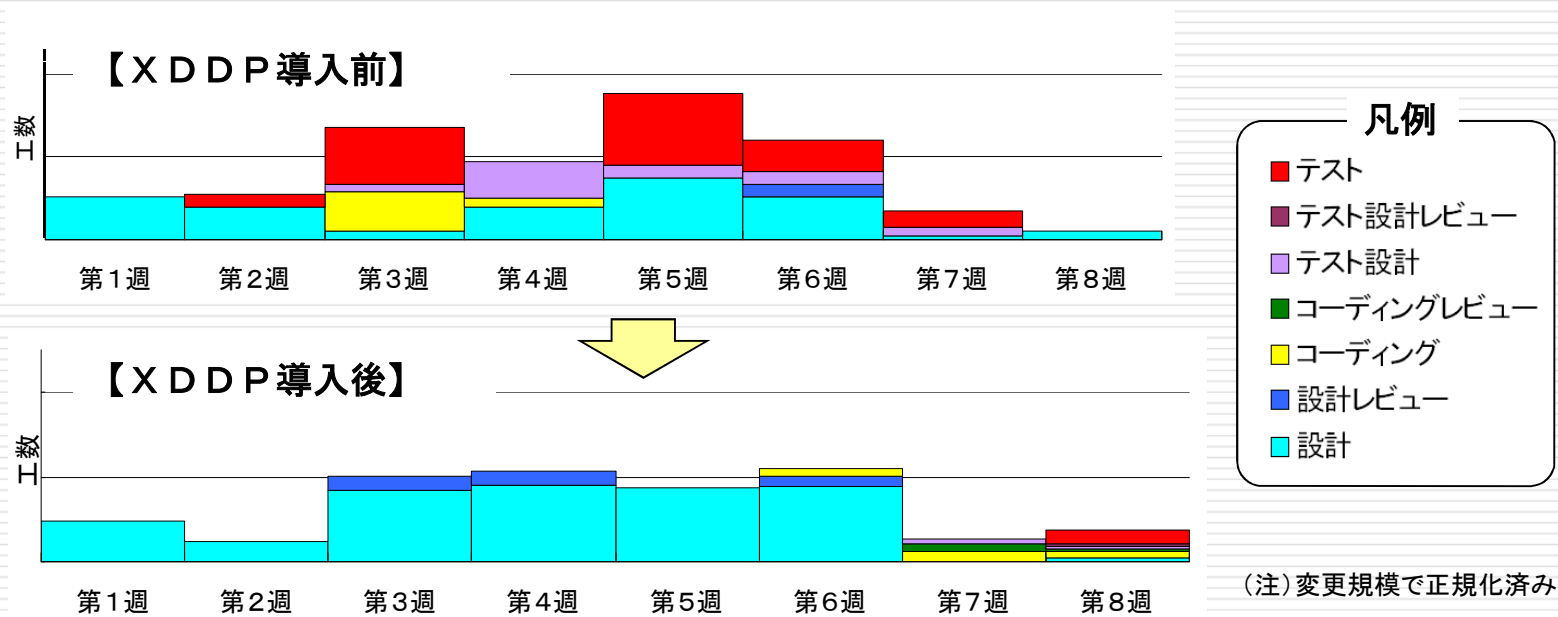
■ 開発事例

対象	ハードウェア制御
変更規模	約 1KLOC (母体:数 KLOC)
担当者	1名
期間	2ヶ月

■ 結果

生産性	全体	1.26 倍
	実装	1.99 倍
品質		流出不具合 0 件

■ 工数分布



■ 技術者

- プロジェクトの先が見えるので安心して開発を進められる
- レビューでハードウェアグループとのコミュニケーションが図れた
- メンバー相互の信頼感が増し、担当するドメインの知識が向上した

■ 管理者 / リーダー

- あらかじめ成果物が定義されているので、計画/進捗が管理しやすい
- 変更がまとまって表現されていて効率的にレビューできる
- レビューした設計ドキュメント(3つの成果物)を次の開発に残すことができた

■ 推進者

- XDDP 導入前は下を向いていた担当者の表情が明るくなった

- 取り組みやすい – 親和性 –
 - これまでの開発と基本的に**実施することは変わらない**
 - **実施順序**を整理し、変更点を記述する**成果物を追加**しただけ
 - 「現状のスキル + α (*USDM*、*PFD* の習得)」で対応できる
 - 設計やテストの技法は影響を受けない
- すぐ効果が出る – 即効性 –
 - 派生開発に**必要小限の成果物とプロセス**で開発を進める
 - 必要なプロセスを適切なタイミングで実施している
 - テスト工程で不具合が出なくなり、**後戻り工数が激減**する
- まだ工夫の余地がある – 恒久性 –
 - *XDDP* は基本原則であり、**他の技術との組合せ**が可能
 - **現場の工夫**により、更なる**改善効果**が期待できる

3. 開発現場での取り組み

22 / 36

3.1 派生開発カンファレンス 2010

3.2 ソニーイーエムシーエス (1)

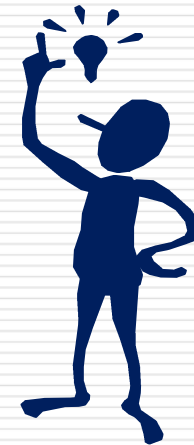
3.3 ソニーイーエムシーエス (2)

3.4 デンソー (1)

3.5 デンソー (2)

3.6 日立情報制御ソリューションズ (1)

3.7 日立情報制御ソリューションズ (2)



- 派生開発カンファレンス 2010
 - 派生開発推進協議会主催のカンファレンス（年1回実施）
 - 2010年6月18日（金）@横浜開港記念会館
 - 258人の参加、**大盛況**のうちに終了

- 発表内容
 - (1) XDDPとUSDMMでプロジェクトの課題解決 - 東京エレクトロン東北
 - (2) **派生開発におけるQDC目標の同時達成目指して** - ソニーイーエムシーエス
 - (3) **XDDPによる無知見プロジェクトのプロセス改善** - デンソー
 - (4) エンプラ系におけるユーザコミュニケーションへのUSDMMの活用 - TIS
 - (5) **ソフトウェアの改造で悩んでいませんか？** - 日立情報制御ソリューションズ
 - (6) XDDPによる派生開発ソフトウェア品質向上の取り組み
 - パナソニック ファクトリー ソリューションズ

- **XDDP 導入の施策**
 - ワークショップ、XDDP のテスト導入
 - 要求仕様のデータベース化 (*XDDP on web*)
 - XDDP のプロセス管理・プロセス分析
- **導入効果**

項目	導入前	XDDP
生産性	37.4 LOC/H	123.2 LOC/H
設計内テスト不具合数	98 件	6 件
設計内不具合含有率	8.2 件/KLOC	0.5 件/KLOC
QA 不具合件数	9 件	1 件
納期遅延日数	0日	0日

QCD の目標の同時達成が実現できた

- 担当者の声
 - 開発途中での仕様変更が以前より減った
 - 何を変更するかが関係者間で共有できているので仕様漏れや実装漏れが軽減している
 - レビューやインスペクションで指摘を受けやすくなった
 - 仕様の漏れや手戻りが減らせた
 - 修正箇所やその周辺の処理が事前に明確になった
 - 複数人での処理が確認できた
 - ソースコード修正前に誤りや抜けを見つけることができた
- XDDP を導入して大変だったこと
 - 資料作成、調査に時間がかかる ← 時間をかけるだけの価値がある
- 次のプロジェクトもXDDP を続けたいですか？
 - はい・・・6名 いいえ・・・0名

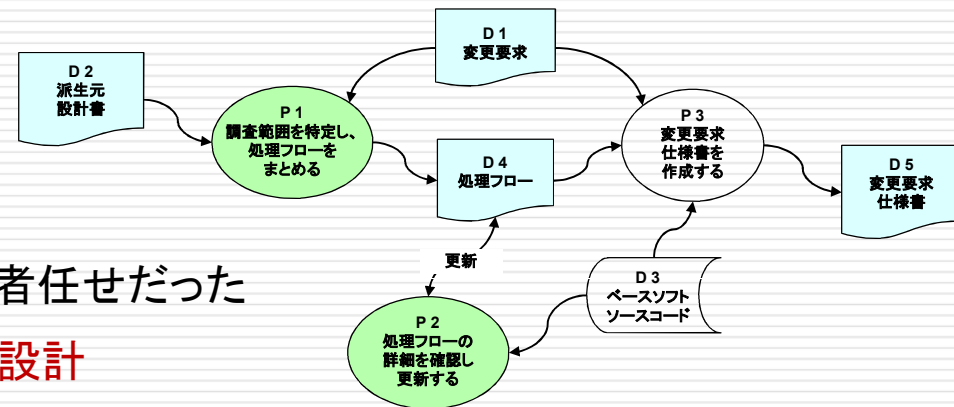
■ 無知見プロジェクト

- 派生元のソフトウェアに対してソースコードの知見を有した**担当者が不在**、もしくは**担当者の知見が不十分**なまま開発せざるを得ないプロジェクト

■ 現場担当者の工夫

● 調査プロセスを定義

- 調査プロセスの標準はない
- これまで調査プロセスは担当者任せだった
- **PFD** を使って調査プロセスを**設計**
 - **変更要求仕様書**を作成するプロセスを設計

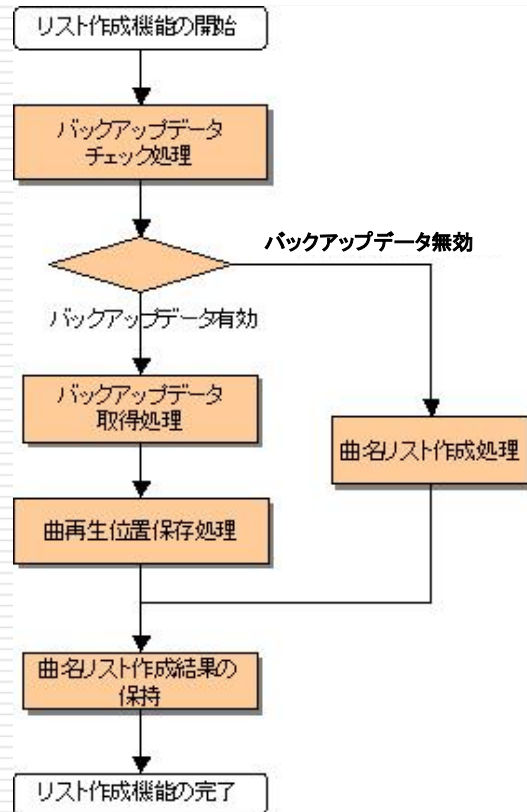


【調査プロセスの PFD】

● 変更要求仕様書の構成を厳密に規定

- **USDM** のカテゴリの**構成と分類方法**を規定 → 担当者に依存しない構成へ
- 変更漏れをなくすために変更機能の**処理フロー**を活用する
- **共通処理**の変更は新たなカテゴリとして扱う

【 処理フロー 】



カテゴリ
抽出

【 変更要求仕様書 】

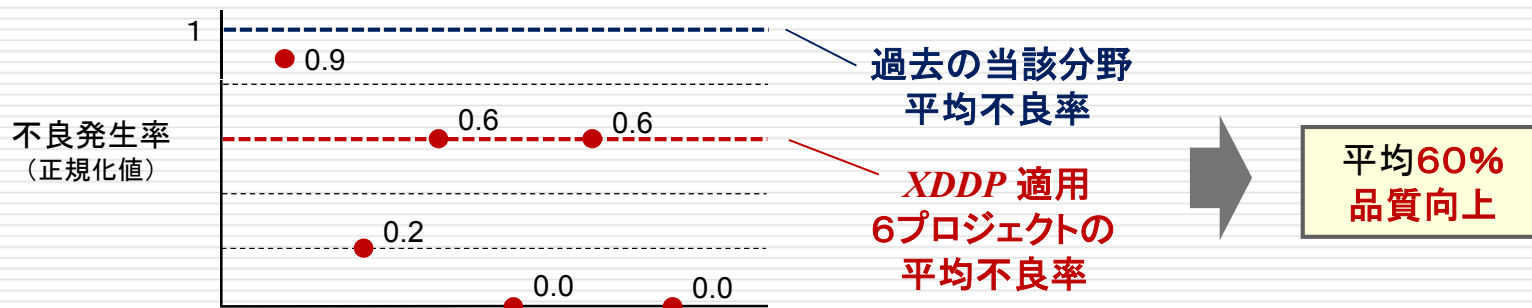
要求	オーディオ制御	オーディオ再生中かつ曲名リスト作成中に電圧が低下した場合、復帰時、曲の先頭から再生する。本仕様を、電圧低下検出直前の再生位置から、再生するように変更する。
理由		電圧低下状態からの復帰処理で、曲名リスト作成が完了した後は、直前の再生位置から再生出来るが、曲名リスト作成完了前でも、直前の再生位置から再生できるほうが好ましいため。
説明		電圧低下により、ソフトがリセットする可能性があるため。
* <バックアップデータチェック処理の変更>		
- <曲名リスト作成条件の変更>		
要求	オーディオ1	曲名リストを作成するタイミングを、バックアップデータチェックの結果と、曲名リストの作成結果を合わせて判断する。
理由		-
説明		-
オーディオ1-1		バックアップデータチェック結果が「正常以外」の場合、曲名リストを作成する。
オーディオ1-2		曲名リストが「作成完了以外」の場合、曲名リストを作成する。
- <バックアップデータ取得処理の変更>		
要求	オーディオ2	変更なし
理由		-
説明		-
* <曲再生位置保存処理の変更>		
* <曲名リスト作成処理の変更>		
* <曲名リスト作成結果の保持の変更>		

■ XDDP の適用評価

種別	開発規模	不良発生率 (件/Ks) の比較				
		設計	単体テスト	組合テスト	結合テスト	顧客先
通常	1	1	1	1	1	1
XDDP 適用	1.3	1.54	0.77	0.31	0.56	0.16

■ ある分野での品質向上効果

仕様の確認不足による不具合が減少



■ 支援ツール SagePro/eXM

- 変更要求仕様書作成支援 ... 変更要求、変更仕様から変更要求仕様書を作成
- 変更TM、変更設計書作成支援 ... 変更TMの作成、変更設計書との自動連携
- 効率的なレビュー実施支援 ... レビュー時のプログラム閲覧、見たい情報の参照等

■ XDDP の有用性 (8つの仮説の実証)

- (1) 開発者でなくても派生開発が可能になるのではないかと
→ 担当者が過去の経験者でなくても派生開発が可能
- (2) 情報共有可能なレベルでの変更要求の明確化が実現できるのではないかと
→ 変更要求仕様書で明確化
→ 情報共有も可能となり第三者チェックがスムーズになる
- (3) 変更の影響範囲をシステム全体にわたり確定できるのではないかと
- (4) 変更漏れを無くすことができるのではないかと
- (5) 組織的なレビュー、情報の共有を徹底できるのではないかと
→ 変更要求仕様書、トレーサビリティマトリックス
→ システム全体にわたり影響範囲を確定できる
& 第三者を含めた組織的なレビューができる
- (6) テストケースの作成、テスト作業の効率化を図れるのではないかと
→ 変更点が見える様になり、変更設計書で抽出できる
- (7) 上流工程強化により、後戻り作業を削減することができるのではないかと
→ 上流でのレビューにより不具合を抽出できることから後戻り作業が減少
- (8) 大きな品質向上を望めるのではないかと
→ 不良発生率 60% 減少

4. 最新の技術動向

30 / 36

4.1 *SPLE* との連携

4.2 *SPLE* と *XDDP*

4.3 *SPLE* と *XDDP* の関係



■ SPLEとは？

- **SPLE** : *Software Product Line Engineering* (ソフトウェア製品系列開発)
 - ソフトウェア製品系列開発とは、プラットフォームと大量個別生産を用いてソフトウェアアプリケーション(ソフトウェア集約システムおよびソフトウェア製品)を開発するパラダイムである。

■ XDDPとSPLE

- 「XDDP」も「SPLE」も共に「派生開発」の世界を扱う方法である
- 「SPLE」は現実的な作業方法を示していない
 - SPLE は製品系列開発のパラダイム / XDDPは現実の作業方法の1つ

■ SPLEを始める前提

- SPLEに耐えるアーキテクチャ技術はあるか？
- 再利用性の高いモジュール技術はあるか？
- 劣化しにくい最初からの設計技術はあるか？
- 劣化しにくい変更技術はあるか？
- 計画通りに作業を進めることができるか？



XDDP はSPLE の前提をクリアする**一つの手段**

- **XDDP** によって手に入るもの
 - 秩序ある派生開発のアプローチ
 - ソースコードの**劣化防止** / 劣化を遅らせる
 - バグが少なく**最小限の工数** / **コスト**で対応できる
 - リファクタリングによってソースコードの**延命**を図る
 - 余った工数で「**次の準備**」ができる

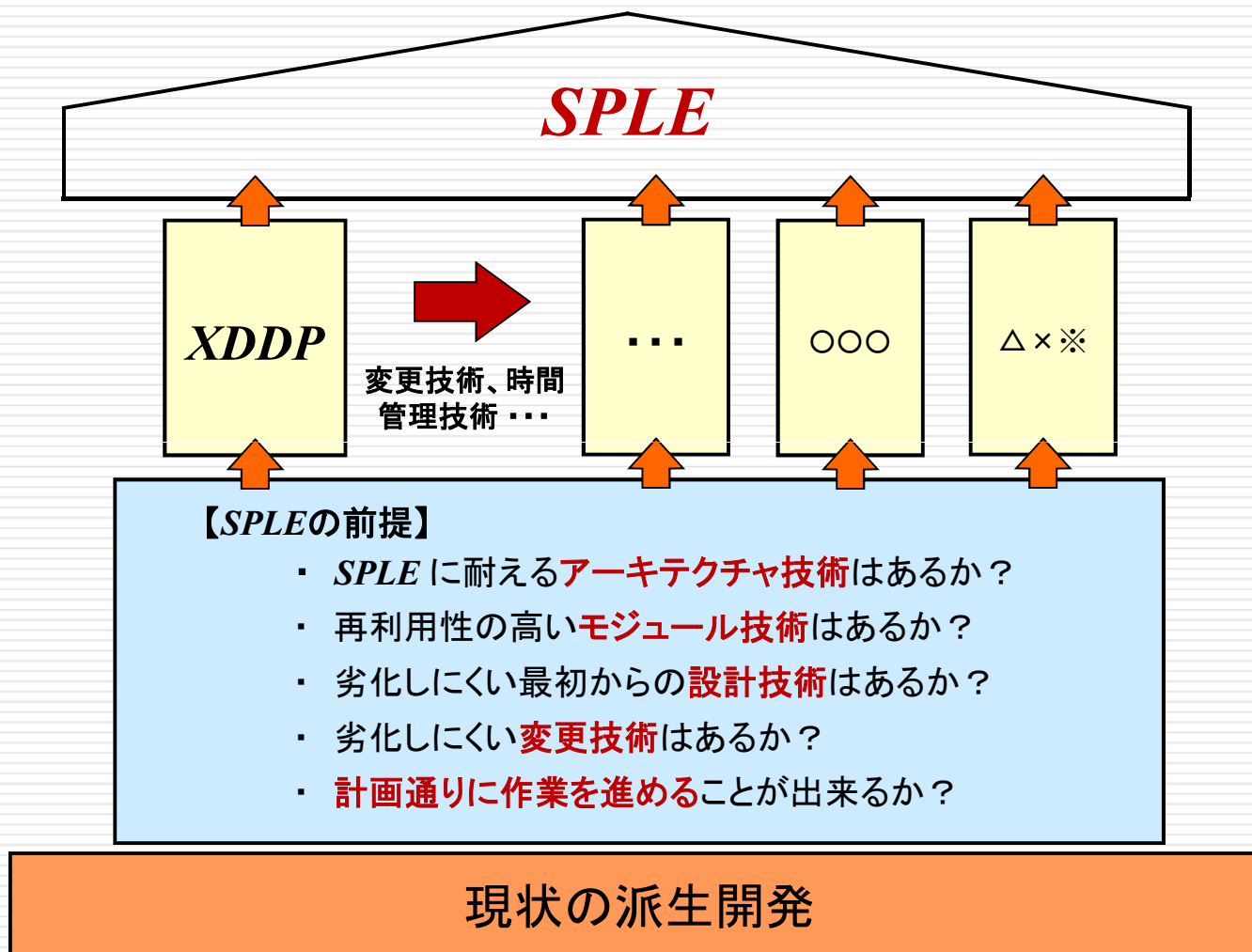


SPLE の前提条件
(技術、管理、準備 ...)

■ **SPLE と XDDP の比較**

SPLE	XDDP
パラダイム	メソドロジ
全体理解 を志向	部分理解 を容認
組合せを志向 (変化点の抽出が重要)	すり合わせの容認 (変化点とその影響が重要)
計画駆動	変化駆動
アーキテクチャ志向 (全体を通して変わらない)	アーキテクチャ希薄 (派生元のソフトウェアに依存)

4.3 SPLE と XDDP の関係



- 日本のソフトウェア開発
 - 日本のソフトウェア開発のほとんどが派生開発
 - 派生開発への対応が今後の成長を左右する

- ***XDDP***
 - 派生開発のための開発プロセス
 - 2つのプロセス(追加+変更)で開発を進める → 品質、生産性の改善
 - ***XDDP***トライアングル = ***XDDP*** + ***USDM*** + ***PFD***

- 開発現場での取り組み
 - 派生開発カンファレンス 2010
 - ソニーイーエムシーエス、デンソー、
 - 日立情報制御ソリューションズ

- 最新の技術動向
 - ***SPLE***と***XDDP***
 - ***SPLE***、***XDDP***共に「派生開発」の世界を扱う方法



- 古畑慶次: プロセス改善の最新動向と実践,
ESEC 第14回組込みシステム開発技術展専門セミナーテキスト, 2011.5.11
- *AFFORDD* ホームページ
<http://www.xddp.jp/>
- 「派生開発カンファレンス 2010」開催報告
<http://www.xddp.jp/conference2010.shtml>
- 勝又淳: 派生開発におけるQDC目標の同時達成を目指して
—XDDPにおけるチームビルディングとプロセス管理, 派生開発カンファレンス 2010
http://www.xddp.jp/conference2010/xddp2010_P3.pdf
- 中井栄次: XDDP適用による無知見プロジェクトのプロセス改善
派生開発カンファレンス 2010
http://www.xddp.jp/conference2010/xddp2010_P4.pdf
- 伯田誠: ソフトウェアの改造で悩んでいませんか?
～派生開発の品質と効率の向上を目指して～, 派生開発カンファレンス 2010
http://www.xddp.jp/conference2010/xddp2010_P6.pdf

6. 参考文献 (2)

- クラウスボール他:ソフトウェアプロダクトラインエンジニアリングーソフトウェア製品系列開発の基礎と概念から技法まで
エスアイビーアクセス, 2009
- 清水吉男:*XDDP...SPL*の現実解としての一つの方法
ES-Kyushu プロダクトライン推進部会(*QPL*)セミナー講演資料