

XDDP適用による 無知見プロジェクトのプロセス改善

株式会社デンソー
ITS技術2部

中井 栄次

1. 背景
2. 現状と課題
3. 課題の分析
4. 改善策
5. 実施結果
6. まとめと今後の進め方

自動車事業



インジェクタ

プラグ



ETC



エンジンECU



メーター



発電機



非自動車事業

環境機器・産業機器・情報機器



赤外線温熱機



自然冷媒
ヒートポンプ式給湯機



産業用ロボット



QRリーダー

開発対象製品



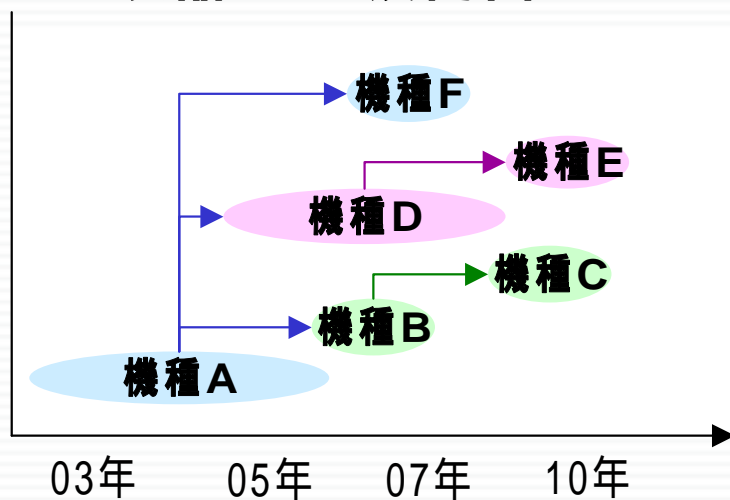
カーナビゲーション

カーナビゲーションシステムを開発

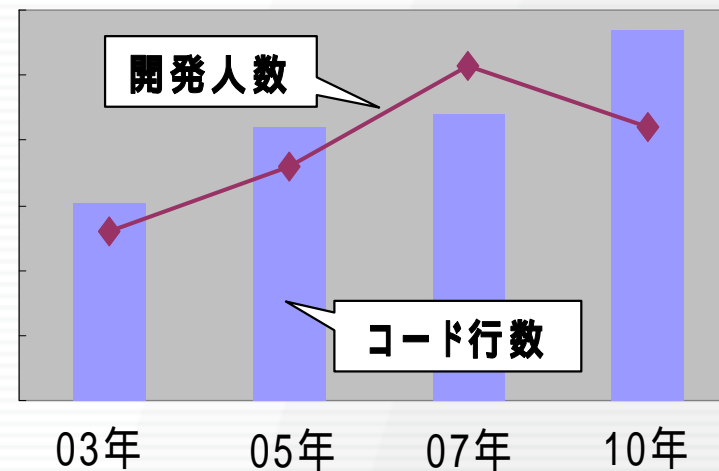
◆ カーナビ開発の現状

- 製品ソフトの大半が**派生開発**
- 多機能化に伴う**ソースコード行数の増加**、**開発人数は減少傾向**
- 海外活用、開発委託先の変更など、**開発体制の見直し**が進行

【製品ソフト展開図】



【コード行数と開発人数】



現在、XDDPを部内の全プロジェクトへ展開中

◆ XDDPとは？

・ 2つのプロセス

- ・ 「機能追加」と「変更」のプロセスに分離

・ 漏れのない仕様化

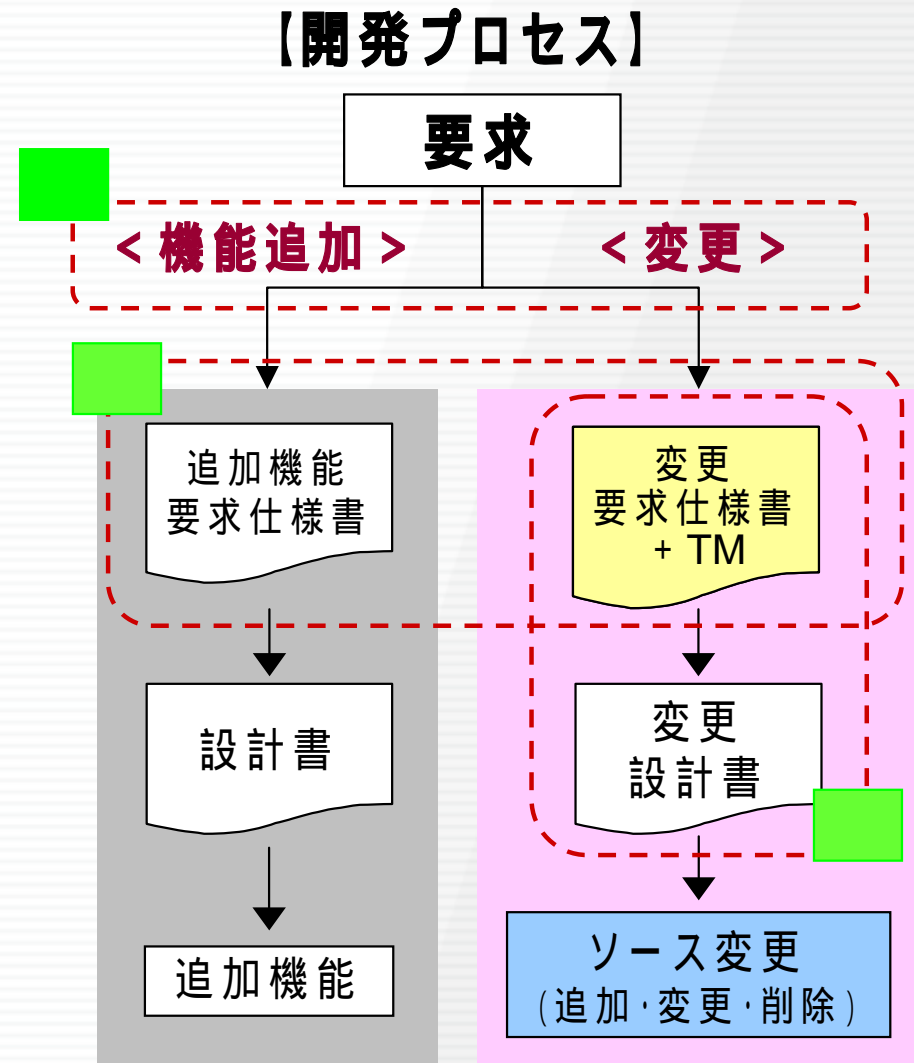
- ・ 追加機能仕様書 / 変更要求仕様書
- ・ USDM(*)を使用

(*) : Universal Specification Describing Manner

・ 変更を表現する3点セット

- ・ 変更要求仕様書 – 何を (what)
- ・ TM(*) – どこを (where)
- ・ 変更設計書 – どうやって (how)

(*) : トレーサビリティ・マトリックス



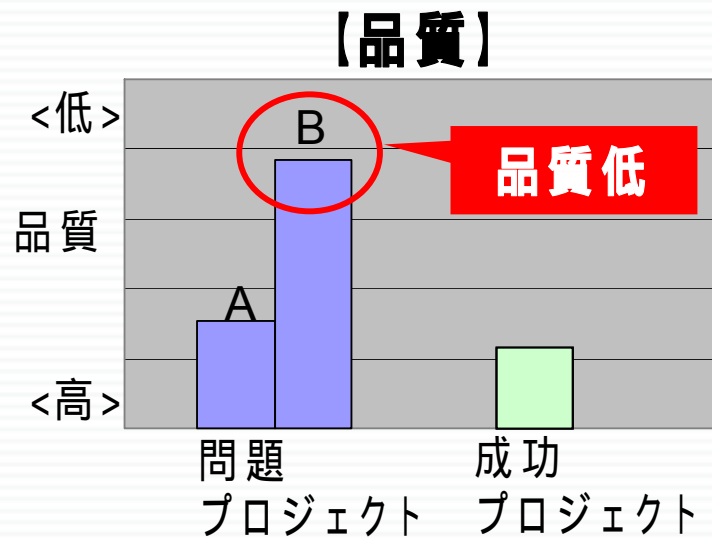
派生開発のための合理的な開発プロセス

◆ XDDP適用の現状

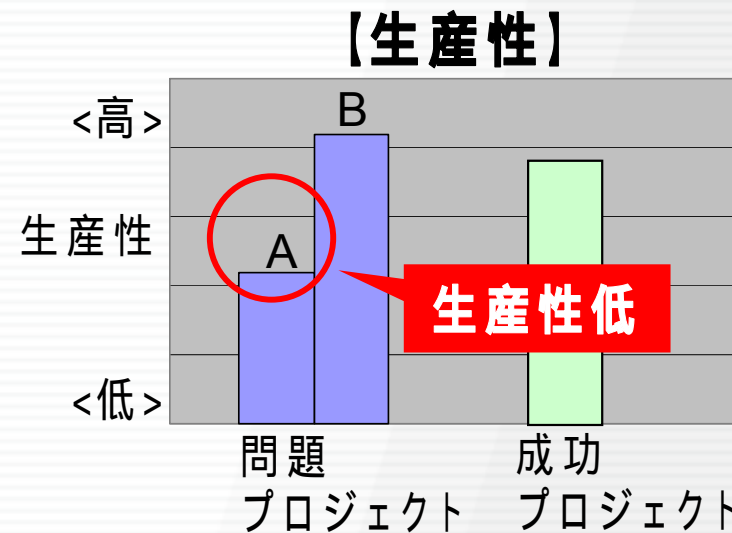
期待する**効果**が**得られない**プロジェクト(問題プロジェクト)が出てきた

■ 問題プロジェクトの評価

問題プロジェクトの代表事例 (A,B)を品質と生産性で評価



* 品質:不具合発生率[件/KLOC]



* 生産性:ソースコード行数 / 開発工数[LOC/H]

品質と生産性を両立できないプロジェクトが出現

◆ 問題の原因分析

■ 問題

- ・ ベースソフトの理解に大きな工数
- ・ 短納期のため、生産性を優先、検査工程で不具合発覚

■ 原因分析

- ・ 影響範囲が特定しにくい
- ・ ベースソフトの意図や理由が不明

なぜ？

担当者の**知見が不十分**

ベースソフトの**有識者が不在**

・ 成功しているプロジェクトと同じプロセスでよいのか？

“無知見プロジェクト”と名づけた

◆ “無知見プロジェクト”とは？

ベースソフトに対して、ソースコードレベルの知見を有したエンジニアが不在のプロジェクト

■ 無知見の原因

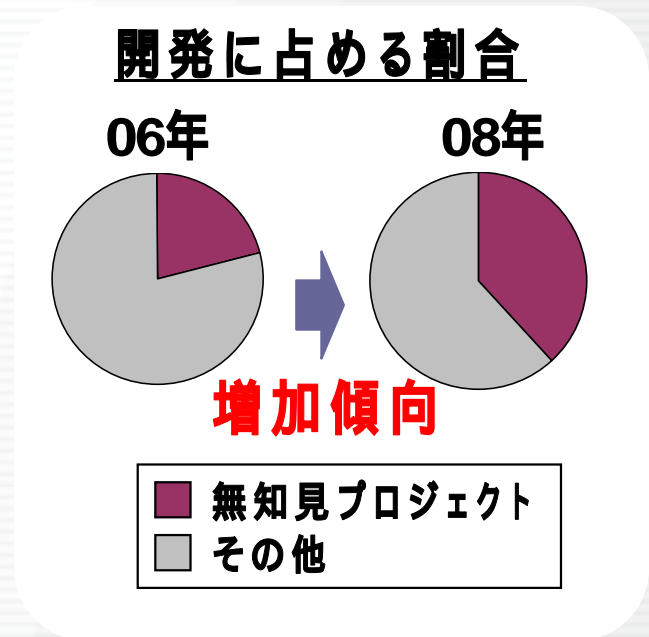
- ・ 担当者が既にいなくなった古いソフト
- ・ 外部委託先から引き戻したソフト
- ・ 開発委託先の変更(海外活用、オフショア)

■ 従来の対応

- ・ 無知見のリスクを考慮せず、**XDDPをそのまま適用**

■ 課題

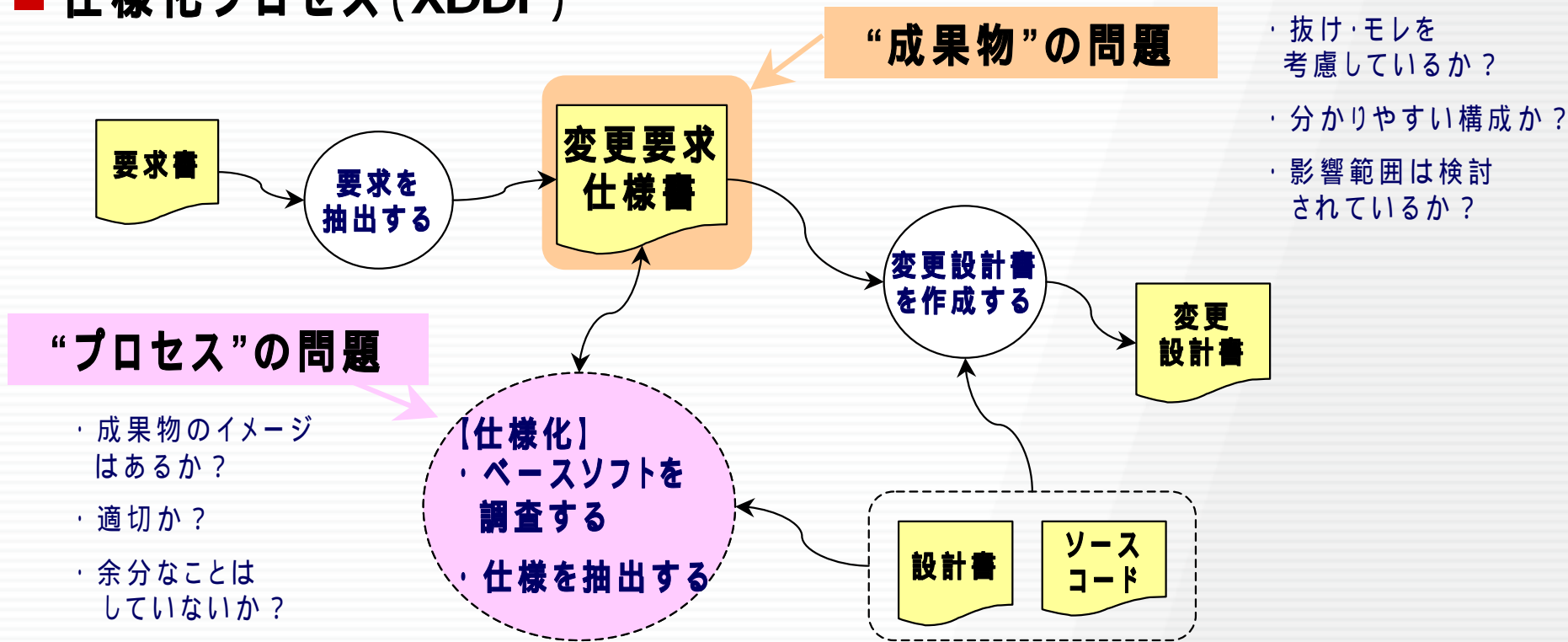
無知見プロジェクトの品質・生産性の両立



◆ 問題の抽出

品質と生産性の両立に向け、XDDPの仕様化プロセスに着目

■ 仕様化プロセス(XDDP)



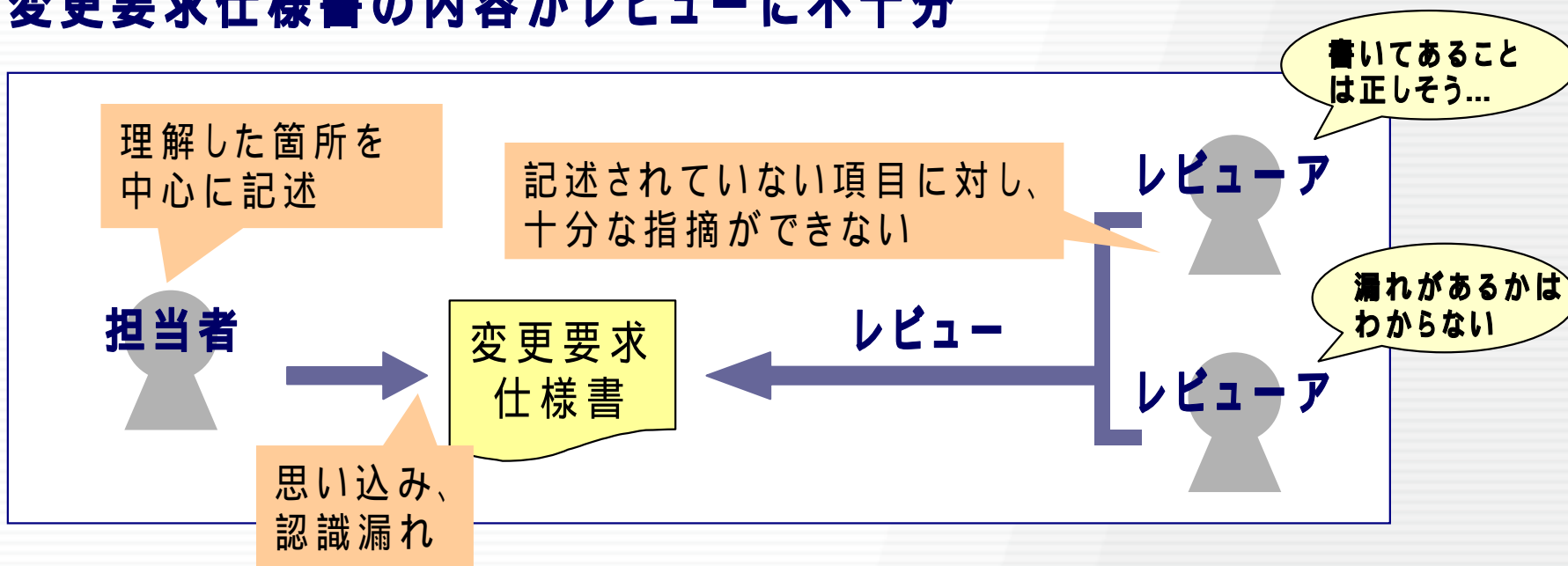
問題は“成果物”と“プロセス”にある

◆ 成果物の問題

仕様に対する**思い込み**、**認識漏れ**がレビューで指摘されない

■ 原因

変更要求仕様書の内容がレビューに不十分



■ 分析結果

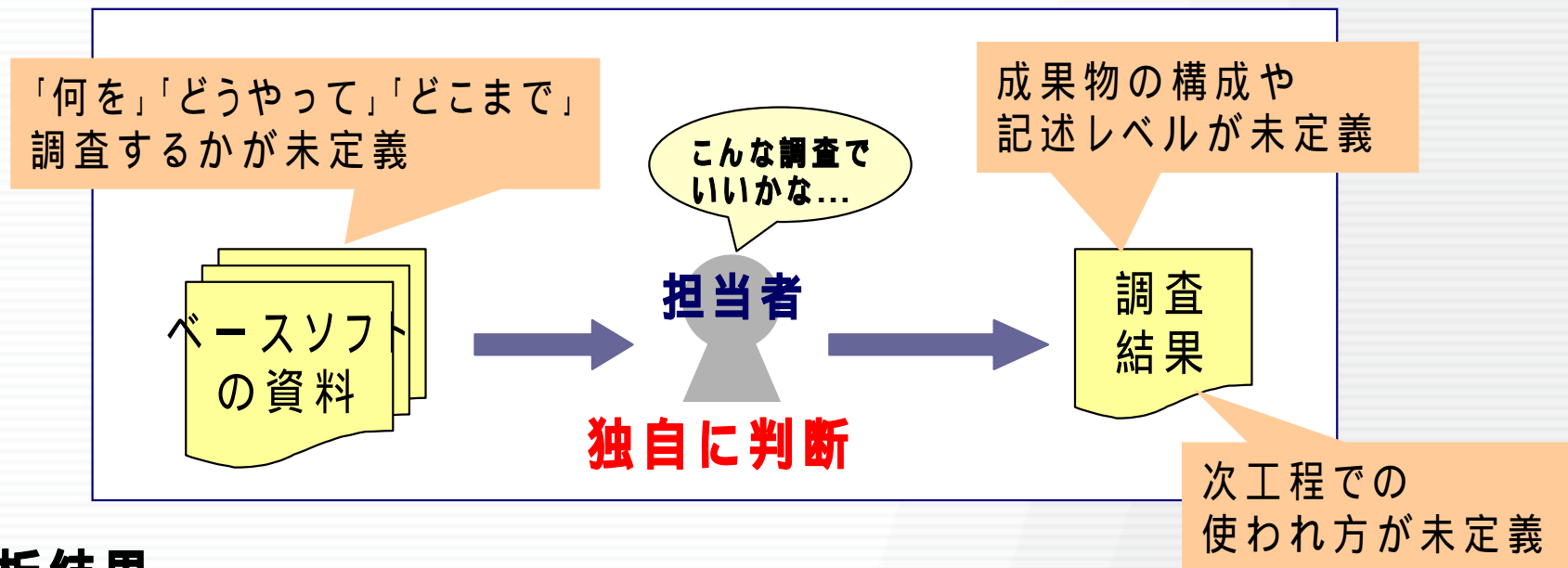
変更要求仕様書の構成が担当者の理解度に依存

◆ プロセスの問題

- ・ 調査不足による**変更箇所の抽出漏れ**が発生
- ・ 余分な調査を行うことにより、**工数が増加**、納期を圧迫

■ 原因

- ・ 調査のやり方があいまい



■ 分析結果

調査プロセスが担当者の独自の判断で実施

■ 問題

仕様書の構成が担当者に依存



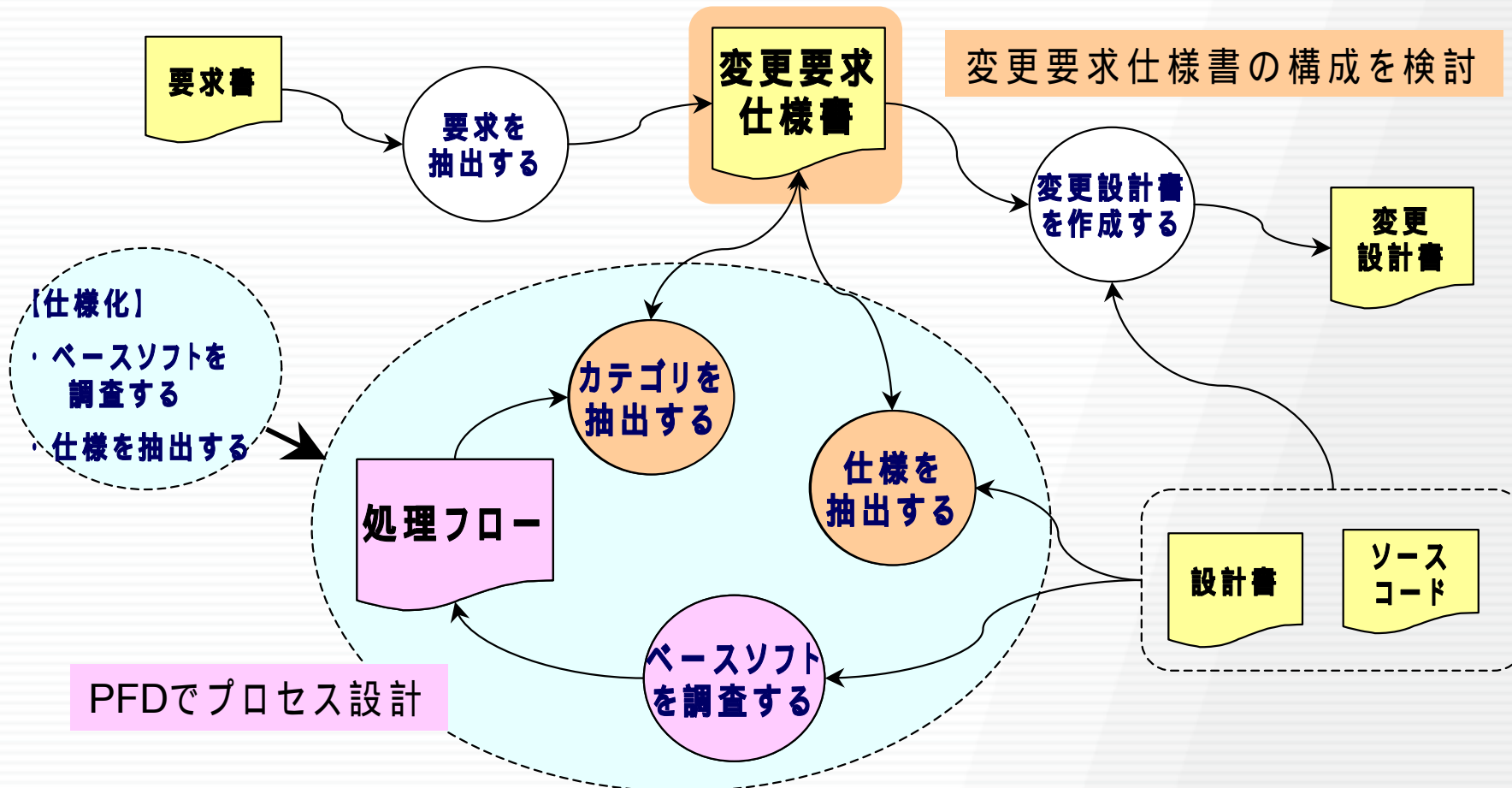
担当者が独自に調査を実施



■ 改善策

仕様書の構成定義

調査プロセスの設計



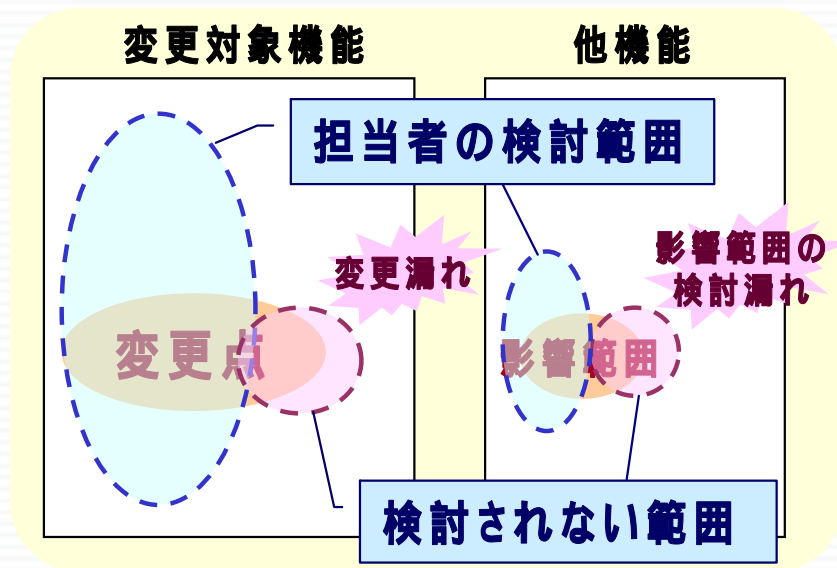
◆ 改善策 仕様書の構成定義 - USDМのカテゴリの構成と分類方法 -

■ ねらい

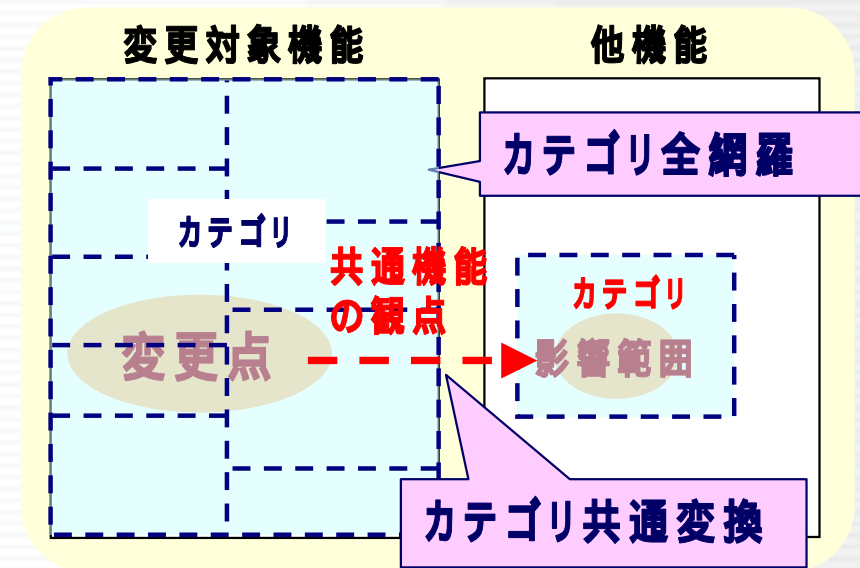
- ・カテゴリ分類方法をあらかじめ定義
- ・漏れ・ダブリのないカテゴリで構成

- ➡ 担当者の検討漏れを防止
- ➡ レビューアの気づきを誘発

【従来】



【今回】

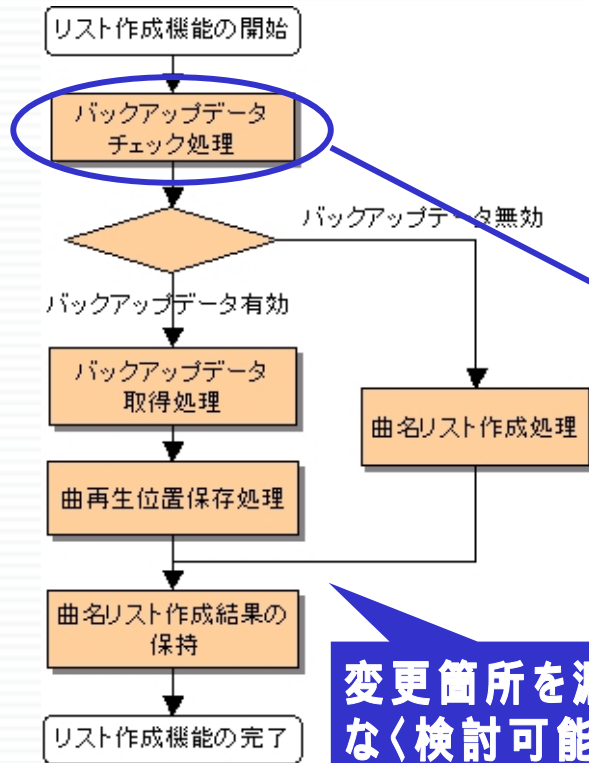


カテゴリ全網羅に加えカテゴリ共通変換も実施

■ カテゴリ全網羅

処理フローを作成し、**時系列**のカテゴリを**漏れなく**検討

【処理フロー】



【手順】

処理フローを作成
 処理フローからカテゴリを抽出(時系列分割)
 カテゴリの範囲で変更要求、変更仕様を抽出
 変更がないカテゴリはその旨を明記

【変更要求仕様書】

要求	オーディオ制御	電圧低下状態からの復帰時、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。
<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	2.1	曲名リストを作成するタイミング、曲名リストの結果と、曲名リストの...
	要求仕様2.1.1	バックアップデータチェック...
	要求仕様2.1.2	曲名リストが「作成完了...
<バックアップデータ取得処理の変更>		
要求	3.1	変更なし
<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>		

検討範囲が限定され、仕様漏れを防止

レビューが変更漏れを指摘

検討漏れを防止し、レビューの効果を向上

■ カテゴリ共通変換

共通機能のカテゴリを作成し、**異なる観点**で影響範囲を再検討

【共通機能】

複数の機能が使用する共通処理

- 例 ・ファイルアクセス処理
- ・タイマ処理
- ・通信処理
- ・共通データ管理

【手順】

共通機能のカテゴリを作成
 共通機能のカテゴリへ変換
 異なる観点で再度検討

影響範囲を検討

【変更要求仕様書】

要求	オーディオ制御	電圧が低下からの復帰後、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。
<バックアップデータチェック処理の変更>		
要求	1.1	バックアップデータにサイズチェックを追加する。
要求仕様	1.1.1	サイズ取得に失敗した場合、返却する戻り値を「サイズ取得失敗」に変更する。
要求仕様	1.1.2	要求仕様3.1.1を参照
<曲名リスト作成条件の変更>		
要求	2.1	バックアップデータチェックが正常終了である場合に、曲名リストを作成するよう変更する。
要求仕様	2.1.1	バックアップデータチェック結果が「正常以外」の場合、曲名リストの作成処理を追加する。
要求仕様	2.1.2	バックアップデータチェック結果が「作成完了以外」の場合、曲名リストの作成処理を追加する。
<バックアップデータ管理の変更>		
要求	3.1	バックアップデータの管理を、ファイル名からの管理からファイル名とサイズの管理に変更する。
要求仕様	3.1.1	サイズを正常に取得した場合、返却する戻り値を「OK」から「サイズ取得OK」に変更する。
要求仕様	3.1.2	ファイルが存在しない場合エラーを返却していたが、サイズの有無も確認するように変更する。

共通処理

時系列
カテゴリ

カテゴリ作成

変換

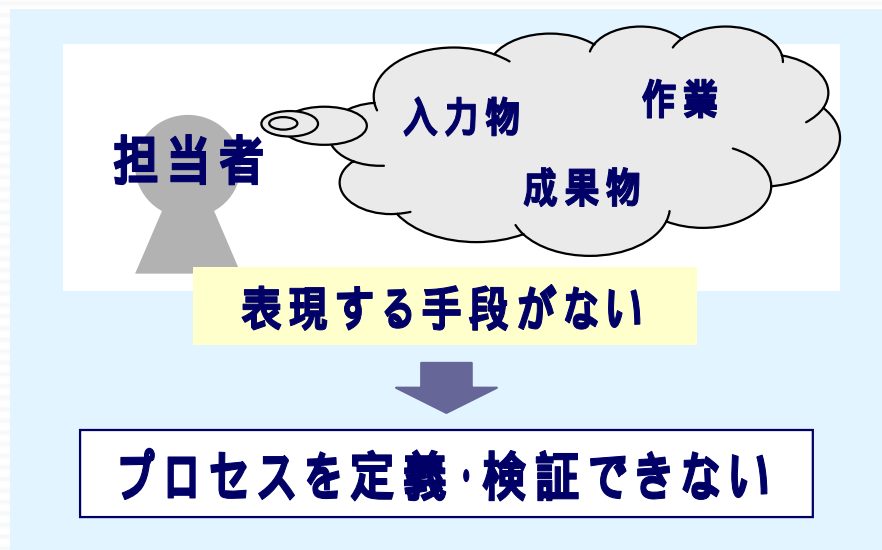
カテゴリ共通変換により、変更誤り、デグレードを防止

◆ 改善策 調査プロセスの設計

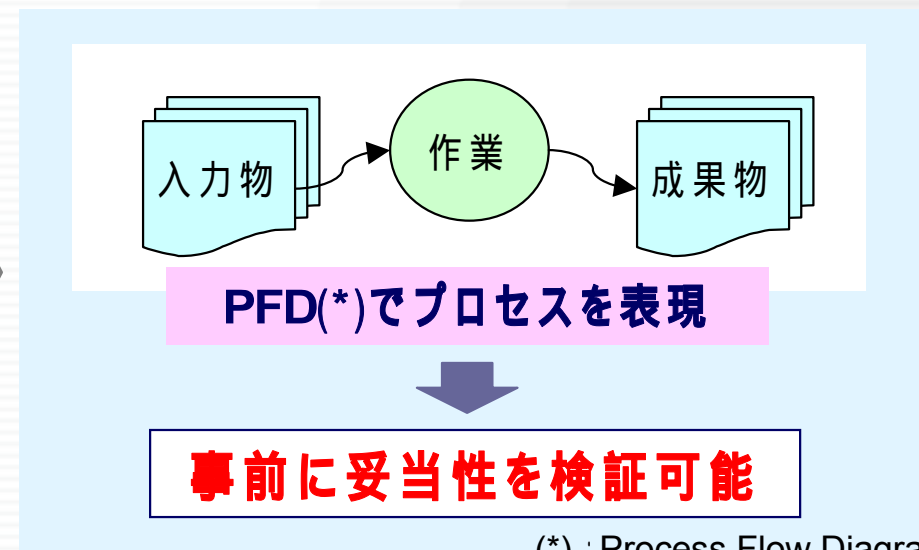
■ ねらい

- ・ 今回の調査の目的に合わせ成果物を定義 ➡ 余分な作業を防止
- ・ 入力物と作業内容を明確に定義 ➡ 調査漏れを防止

【従来のプロセス】



【今回のプロセス】



(*) : Process Flow Diagram

調査漏れと余分な作業の発生を事前に防止

■ プロジェクトの状況

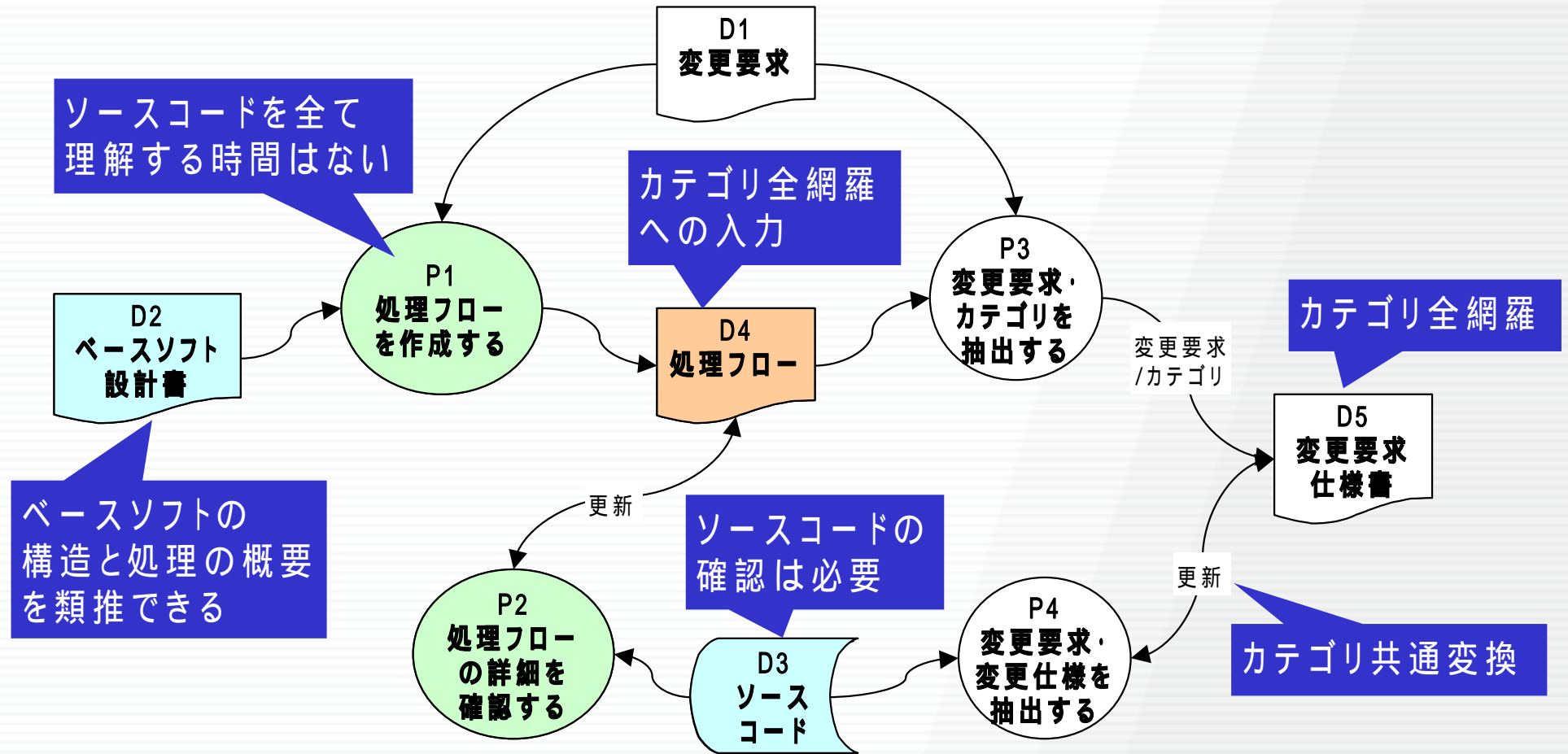
無知見の要因	外部発注ソフトの引き戻し
ドキュメント	設計書あり(処理と構造の概要が理解できるレベル)
担当者の知見レベル	ベースソフトの仕様・構成をある程度理解
レビューアの知見レベル	なし
納期	1ヶ月(短納期)

■ プロセス設計の方針

- ・カテゴリ全網羅を実施するため、処理フローを成果物とする
- ・短納期のため、設計書で概要をつかみ、ソースコードで確認する

プロジェクトの状況からプロセス設計の方針を決定

■ 設計した調査プロセス



今回のプロジェクトに最適な調査プロセスを設計

◆ 対象プロジェクト

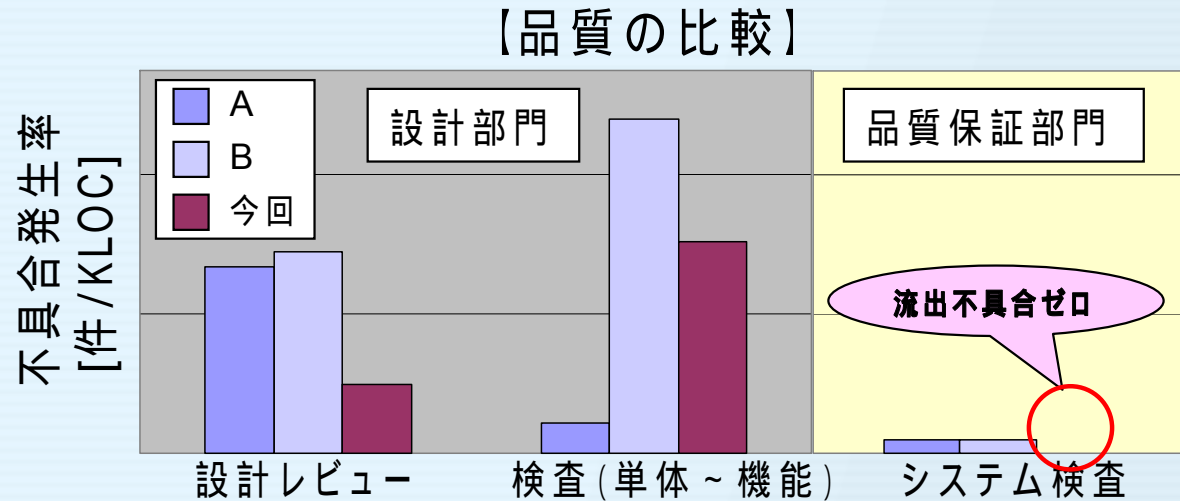
	今回プロジェクト	比較プロジェクト	
		プロジェクトA	プロジェクトB
開発手法	無知見プロジェクト用XDDP	XDDP	XDDP
開発概要	ミドルウェア(オーディオ制御)	ミドルウェア (オーディオ制御)	マイコン制御
変更規模	約300 [LOC]	約3,000 [LOC]	約1,000 [LOC]
ベース規模	約26,000 [LOC]	約36,000 [LOC]	約1,100 [LOC]
開発期間	1ヶ月	7ヶ月	4ヶ月
知見レベル	・外部発注ソフトの引き戻し ・ドキュメントあり	・開発委託先の変更 ・ドキュメントあり	・開発委託先の変更 ・ドキュメントあり

◆ 結果の比較

品質

**設計部門からの
流出不具合ゼロ**

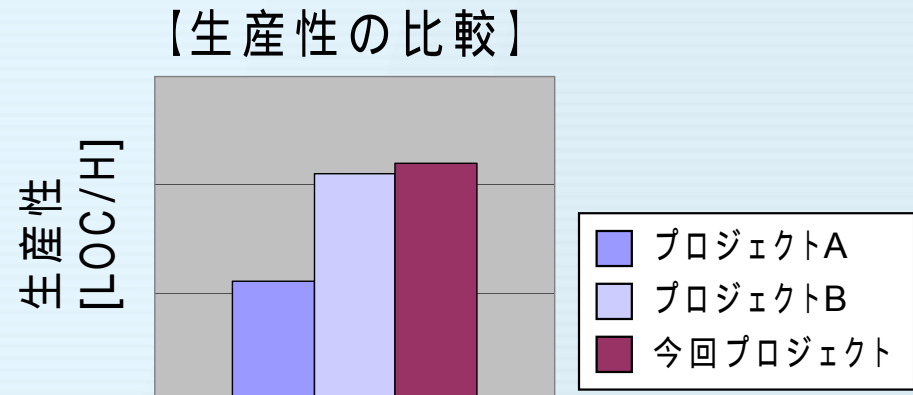
仕様書の構成定義、検査
により仕様漏れ、誤りを防止



生産性

他プロジェクトに比べ生産性向上

PFDにより作業漏れや
余分な作業を防止できた



プロジェクトA, B に比べ、品質・生産性をともに改善

◆ 考察

- 無知見プロジェクトの成否は、**調査・仕様化**で決まる
 - ・ 調査プロセスと仕様書の構成定義が品質と生産性を改善する
 - ・ 無知見プロジェクトでも、流出不具合ゼロ、納期遅れなし
- **仕様書の構成**を工夫することで抜けモレを防止できる
 - ・ カテゴリ全網羅とカテゴリ共通変換により、仕様の抜けモレを抑えた
 - ・ 開発着手時には想定していなかった変更箇所も抽出できた
- 無知見プロジェクト以外にも適用可能である
 - ・ 特に担当者のスキル、経験が不足するプロジェクトに有効である

◆ まとめ

- 無知見プロジェクトに対してXDDPの適用方法を工夫し、**品質・生産性**を改善した
 - ・ 変更要求仕様書の構成を定義し、変更漏れ・変更誤りを防止
 - ・ 調査プロセスを事前に設計し、作業漏れ・余分な作業を防止

◆ 今後の進め方

- 仕様化技術のレベルアップ
 - ・ 検査工程で検出された不具合を仕様化の段階で防止
- 無知見プロジェクト自体の削減
 - ・ XDDPによる変更箇所・変更理由の明文化
 - ・ ベースソフトのリファクタリング

◆ カテゴリ分割の効果

- 上位要求に対して、検討する範囲を限定

➡ 変更漏れ、変更誤りを起こしにくい
生産性の向上

- 分割の基準を統一

➡ 漏れ・ダブリが発生しにくい

分割基準の例

- 時系列分割
- 構成分割
- 状態分割
- 共通分割

変更要求仕様書

要求		
	理由	
	説明	
<カテゴリ>		
要求		
	理由	
	説明	
	要求仕様	
	要求仕様	
<カテゴリ>		
要求		
	理由	
	説明	
	要求仕様	
	要求仕様	
要求		
	理由	
	説明	
	要求仕様	
	要求仕様	

カテゴリの適切な分割が重要

プロジェクトの**状況、背景**に合わせて、**プロセスを毎回変化させる**

■ プロセスが毎回設計されてこなかった理由

- ・毎回異なる作業（調査など）は標準プロセスとして定義できない
- ・プロセスを自在に表現するツールがない

➡ PFDによるプロセスの“見える化”により、開発のシミュレートが可能

■ 固定された標準プロセスの弊害

- ・標準プロセスの強要によるやらされ感
- ・プロセスの形骸化

➡ プロセスの定義により、自らプロセスを考え、改善する意識を醸成

開発のシミュレートとプロセス改善意識の向上

成果物定義書

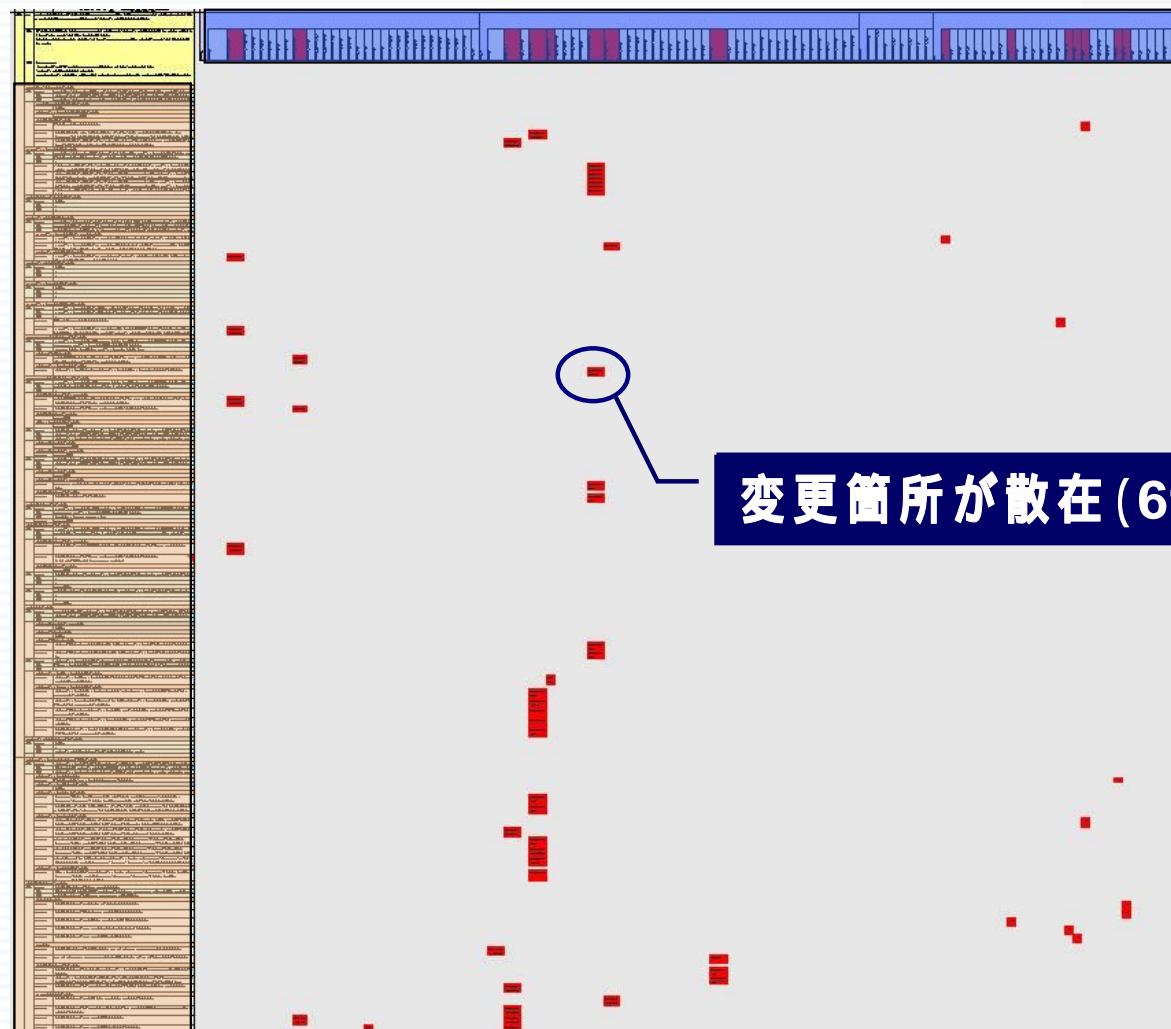
文書番号	成果物名 ・ ファイル名 / 構成と説明	作成日
D4	処理フロー	
成果物の目的	変更要求仕様書のカテゴリを抽出するために作成する。	
編集方針	カテゴリの抽出が目的であるため、必要以上に詳細を記述しないこと。 設計書を参照した情報は、ソースコードで設計書の記述が正しいことを確認すること。 他機能と依存関係がある場合は、コメントを残すこと。	
成果物の構成と簡単内容	1. タスク構成 タスク構成図、タスク責務一覧 2. 各処理のフローチャート ・リスト作成処理 ・バックアップ書き込み処理 分岐のマトリックスも作成すること 3. バックアップ状態管理 状態遷移図	

プロセス定義書

プロセス番号	プロセス名	作成日
P1	調査範囲を特定し、処理フローをまとめる	
入力情報	D1 変更要求 D2 ベースソフト設計書	
出力物	D4 処理フロー	
作業内容		作業担当者
プロセス実施条件：D1、D2を入手した時点 必要スキル：ベースソフトの機能、構成の知見を有すること 担当者： 作業内容： 1. 変更対象機能を抽出する 変更要求から、今回の変更に関係する機能(変更機能)に当たりをつける 2. 設計書から処理フローを作成する 変更機能について、ベースソフト設計書を調査し、処理フローを作成する 時系列の図がある場合は、参考にすること プロセス終了判定： 処理フローについてのDRを実施し、指摘事項の反映が完了した時点		

■ 変更要求仕様書の全体図

変更仕様



ソースファイル

変更箇所が散在(61箇所)