

# 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 ～

ソニーイーエムシーエス(株) 湖西テック  
商品設計1部1課 勝又 淳

# Agenda

- **XDDP導入の経緯**
- **XDDP導入への施策**
- **XDDPプロセスの管理**
- **XDDP導入の実際の効果**
- **独自の工夫**
- **まとめ/今後の課題**

名前： 勝又 淳（かつまた あつし）

所属： ソニーイーエムシーエス(株) 湖西テック  
設計部門商品設計1部1課

職務： 業務用放送機器ソフトウェア設計リーダー



事業部で設計される機種の**派生モデル**の設計が多い



**派生開発**を極める事が会社にとっても有益



## - 業務用組み込み機器の派生モデル設計 -

**期間:** 2009/07~2010/05

**人数:** 約20名(ソフトウェア6名)

**要求数:** 122件 (変更:110件 追加:12件)

**要求仕様数:** 676

**母体行数:** 約650 (KLOC)

**変更行数:** 約 15 (KLOC)



## 酷いデスマーチ

2007/09 ~

組み込み機器 A

Start

- WEBによる一元管理
- BTSの構築
- ふりかえりの実施
- ピアレビュー実施
- 不具合分析

※BTS: Bug Tracking System (不具合管理)

## あと一步!

2008/09 ~

組み込み機器 B

Growth

- 上流工程の強化
- 仕様書の統一
- ピアレビュー実施
- インспекション実施
- コード自動生成
- 単体テストの強化

2009/05 ~

組み込み機器 X

Jump

- XDDPの導入
- V&Vの確立

## 1. XDDPに興味があった

各シンポジウムやセミナーでXDDPの話聞き、派生開発がメインである私達には大変興味深い内容だった。いつか試してみたいと思っていた。

## 2. 過去のプロジェクトでの苦い経験

過去のプロジェクトで、安易なソースコードの変更により、問題を多発させてしまいプロジェクトメンバが疲弊した。同じ過ちを繰り返したくない。

## 3. 現場の改善意識

現状に満足せずに、常に改善しようという現場の雰囲気の後押しがあった。



プロジェクト開始時にXDDPを導入すると宣言し、承諾された。  
自分達で導入を決めたので、絶対に失敗したくない。

## 品質 (Quality)

→ 設計外への**不具合流出0件**(母体の残存不具合は除く)

## 納期 (Delivery)

→ ソフトウェア生産性が上がれば納期も短縮できるはず。  
水準以上の品質を保持して、1時間あたりに書くソースコード行数  
**ソフトウェア生産性:80 LOC/H**

※初めてXDDPに取り組むチームでも、平均して80行/時間の生産性が出せるとのこと。

## 設計工数削減 (Cost)

→ 無駄な後戻り工数を減らすことで、**設計工数15%削減**

※XDDPを導入すると、従来比で2~3割の工数は削減できるとのこと。



## [施策1] XDDPワークショップの開催

XDDPに対してプロジェクト全員の理解度を揃える為にワークショップを開催した。

目的: XDDPの理解度の統一、現場のモチベーション向上

## [施策2] XDDPのテスト導入

変更規模の少ない部分でXDDPのテスト導入を実施し、効果を体感した。

目的: XDDP導入効果の体感による現場のモチベーションの向上

## [施策3] 要求仕様のデータベース化 (XDDP on web)

要求仕様をデータベース化し、WEB上での管理を行った。

他のデータ(不具合、レビュー、サイズ/工数見積もり)との連結を行った。

目的: XDDPの可視化、定量化、構成管理

## [施策4] XDDPのプロセス管理・プロセス分析

XDDPのプロセスが正常に行われているか、常にプロセスの監視・管理を行った。

目的: 定量的データによるXDDPのプロセス管理・プロセス分析

- ・グループ内で週2回×1.5時間、XDDPワークショップ(全15回)を開催
- ・XDDPに関して、ディスカッション/レビューを中心としたワークショップ形式

XDDP導入にあたり、特別なコンサルティングを受けてはいない。

「グループ内で協力し、全員が理解しながら習得する」という方針でワークショップを開催し、XDDPの技術を習得していった。

ワークショップ開催当初は、XDDPに関して半信半疑だったが、  
終了前には「上手くいきそう」という空気になってきた。



参考文献:  
「派生開発」を成功させる  
プロセス改善の技術と極意  
清水 吉男 著

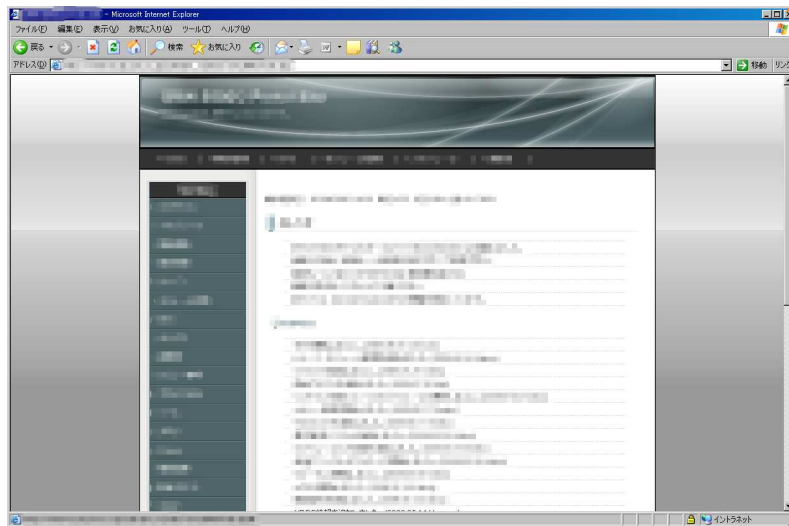


【XDDPワークショップの様子】

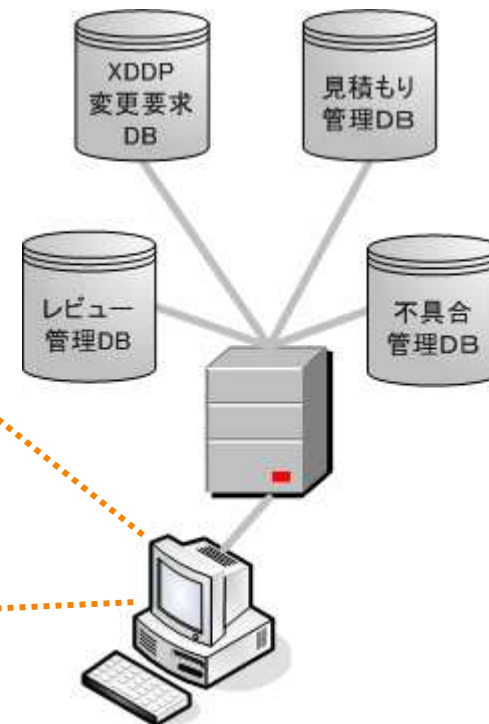
- XDDPでは要求をコード化する。コード化されたものを管理しない手はない。
- 追加機能／変更要求仕様をWEB上で登録し管理する。
- 要求コードを基軸として、データ連結を行う。

- ① 要求仕様管理データ
- ② 不具合管理データ
- ③ 見積もり管理データ
- ④ レビュー管理データ

XDDPを構成するデータ間で連結を行い、XDDPのプロセスが正常に行われているか、WEB上で常に管理・監視を行う。



XDDPのプロセスをポータルサイトで管理  
(PHP + MySQLで構築)



## 変更要求仕様書／追加機能仕様書の出力

- <XDDP on web の特徴>
- ①WEB上で要求・要求仕様を登録
  - ②変更要求仕様書を自動生成 (PDF)
  - ③要求に対応したドキュメントを表示
  - ④サイズ・工数見積もり管理
  - ⑤要求仕様を別の形式でExport
  - ⑥各要求の現況(ステータス)を表示

The screenshot shows the Requirements Management System interface. Key elements include:

- Navigation:** [Home] [プロセスモデル] [PFDD] [使い方] [スケジュール] [BTS] [レビュー]
- Buttons:** 要望新規登録, 要求追加登録, 要求仕様新規登録, 要求更新, 要求仕様更新, 要求仕様書出力, TM出力, 工数見積, エクスポート, TreeView, CSV.
- Table:** 各要求の進捗状況 (Progress Status of Each Requirement)

コード	要求名称	SF	ステータス	要求元	種別	区分	ドキュメント
ADJ01			未着手	湖西・生産技術	変更	不具合	TM
ADJ02		△	完了	湖西・生産技術	変更	効率化	[S][C][T][H][V]
ADJ03			完了	湖西・製造	変更	効率化	[S][C][T][H][V]
ADJ04			完了	湖西・製造	変更	効率化	[S][C][T][H][V]
ADJ05			未着手	湖西・製造	変更	効率化	[S][C][T][H][V]
ADJ07			完了	湖西・設計	変更	不具合	[S][C][T][H][V]
APR01		◎	完了	湖西・設計	変更	コスト削減	[S][C][T][H][V]
APR02			完了	湖西・設計	変更	ディスコン	[S][C][T][H][V]
APR03		◎	完了	湖西・設計	変更	品質向上	[S][C][T][H][V]
APR04			完了	湖西・製造	変更	効率化	[S][C][T][H][V]
APR05		○	完了		変更	不具合	[S][C][T][H][V]
APR06		△	完了	湖西・設計	変更	品質向上	[S][C][T][H][V]
APR07		△	完了	湖西・設計	変更	品質向上	[S][C][T][H][V]
APR08			単体テスト済み	湖西・設計	変更	不具合	[S][C][T][H][V]
APR09			単体テスト済み	湖西・設計	変更	不具合	[S][C][T][H][V]

要求単位で  
管理

### 各要求の進捗状況

### 変更要求仕様書

### 変更設計書

各要求に対するドキュメントを  
ワンクリックで表示



## 【要求仕様入力フォーム】

要求 新規登録 - Microsoft Internet Explorer

変更 or 追加機能を選択

登録

ROOT要求: ADJ02.ALL\_DLのFAIL発生時にFAIL発生箇所が特定できるようにしてほしい

親要求: 000:ROOT直下

グループ名称:

要求仕様名称:

要求種別: **変更** (登録者: 協栄) | コード(ゼロ詰め2桁:01-99):

要求を選択

要求のグループ化・階層化

要求仕様コード

## 【見積もり入力フォーム】

要求仕様を記入

要求仕様内容

TM(SS-95)  PLD1  PLD2  PLD3  PLD4  HAF

TM(APR-62)  PLD1  PLD2  PLD3  PLD4  PLD

TM(VPR-79)  PLD1  PLD2  PLD3  PLD4  PLD

TM(EQ-94)  PLD1  PLD2  PLD3  PLD4  HAF

TM(soft)  SYS1  SYS2  CTRP  SRVO

合意状況: **未着手** | ステータス: **未着手**

ページが表示されました

工数見積もり管理 - Microsoft Internet Explorer

Requirements Management System

初期: 要求仕様書作成時の見積もり  
調整: 変更設計書作成時の見積もり  
実績: 実際に費やしたサイズ・工数

初期・調整・実績の  
サイズ・工数見積もりを登録

ブロック選択: **CTRP** | フェーズ選択: **ALL**

コード	要求名称	初期		調整		実績		単体テスト	
		LOC	H	LOC	H	LOC	H	見積り	実績
ADJ02.01		20	2	15	1	18	0.25	0	0
ADJ02.02		5	1	39	1	26	0.5	0	0
ADJ02.03		4	1	8	1	8	0.25	0	0
ADJ02.04		5	0.25	5	0.25	5	0.25	0	0
ADJ02.05		0	0	0	0	0	0	0	0
APR03-02.08		0	0	0	0	0	0	0	0
APR05.01		10	1	2	1	2	0.25	0	0
APR05.03		40	2	62	2	70	0.5	0	0

## 要求/要求仕様データベースから変更要求仕様書(PDFファイル)を自動生成

The screenshot shows a web browser window displaying a table of change requirements. The table is organized into a hierarchical structure. Annotations with orange arrows and brackets highlight specific features:

- root要求**: Points to the top-level requirement row (IRC02).
- 要求コード**: Points to the 'Requirement Code' column.
- 要求の階層化**: A bracket on the left side indicates the hierarchical structure of the requirements.
- 変更要求仕様**: A bracket on the left side indicates the detailed specifications for a change requirement.
- USDM表記\***: A bracket on the right side indicates the use of USDM notation for requirements.

変更要求	要求コード	要求元	理由	説明
変更要求	IRC02			
変更要求	IRC02-01			
変更要求	IRC02-02			

※USDM (Universal Specification Description Manner) :XDDPにおける要求を仕様化する技術

## 【不具合入力フォーム】

新規登録日 : 2010-03-30 13:08:10

件名	
登録者	未選択 ※選択必須
フェーズ	2nd_step_量産 ※選択必須
モジュール	2nd_step_α+5版 ※選択必須
Version	SYS1 ROM : 8.720 CP PLD : 1.000 EQ PLD1 : 4.000 HPR SYS2 ROM : 8.750 SYS PLD1 : 4.000 EQ PLD2 : 4.000 HPR CP ROM : 8.740 SYS PLD2 : 4.000 EQ PLD3 : 4.000 HPR SV ROM : 8.720 SYS PLD3 : 4.000 EQ PLD4 : 4.000 HPR EQ ROM : 3.500 SYS PLD4 : 4.000 FC ROM : 3.701
テストレベル	未選択
変更要求	ADJ02: 発生箇所が不明なため、発生箇所が特定できないため
テストケース	未選択
深刻度	ADJ03: 発生箇所が不明なため、発生箇所が特定できないため
接続	ADJ04: 発生箇所が不明なため、発生箇所が特定できないため ADJ05: 発生箇所が不明なため、発生箇所が特定できないため ADJ07: 発生箇所が不明なため、発生箇所が特定できないため APR01: 発生箇所が不明なため、発生箇所が特定できないため APR02: 発生箇所が不明なため、発生箇所が特定できないため APR03: 発生箇所が不明なため、発生箇所が特定できないため APR04: 発生箇所が不明なため、発生箇所が特定できないため
設定/条件	

## 【レビュー入力フォーム】

レビュー管理システム

HOME | TOP PAGE | 変更要求仕様書レビューポイント | 変更設計書レビューポイント

レビュー実施日	2010-03-29	工数	H
レビュー名	VPR12変更設計書レビュー		
カテゴリ	インスペクション		
変更要求	VPR12		
レビュー資料	VPR12変更要求仕様書 VPR12変更設計書		
資料種類	変更設計書	資料ページ数	
レビューアー			
担当者			

Sheet-1

ランク	C	指摘内容	

不具合管理/レビュー管理のデータに、変更要求コードを登録することで、  
変更要求・不具合・レビューの因果関係の分析が容易となる。

## 【XDDPプロセス分析(要求別)】

XDDPを構成する各要素をWEB上で管理することで、不具合情報やレビュー情報等とのリンクが可能となる。それにより、プロセスの状況をリアルタイムに知る事が可能となる。

分析したい  
要求コードを選択

要求コードを選択: [IRC04]

### XDDP プロセス分析 (要求別)

XDDPのプロセスに関するマトリクスを要求別に表示します。

[要求仕様数]

要求仕様数合計	変更要求仕様数	追加機能要求仕様数
43	43	

[不具合情報]

不具合件数	単体テストで検出	設計内テスト
2	2	0

[レビュー指摘件数]

合計	Sランク	Aランク
204	12	42

[レビューマトリクス]

ページ数	所要時間	欠陥密度(件/頁)
312	7.5	0.65

[不具合検取状況]

識別	レビューにて検取	単体テスト完了時
工程検出率	99.03%	0.97%
累計検出率	99.03%	100%

[工数・サイズ見積もり情報]

BLOCK	初期見積もり		調整見積もり		実績		見積もり偏差(乖離)	
	サイズ(L)	工数(h)	サイズ(L)	工数(h)	サイズ(L)	工数(h)	サイズ	工数
SYS1	3515		1699		1389		-1.53	
SYS2	5935		5190		5756		-0.03	
CP	3920		5276		5523		0.29	
CV	1240		1300		1326		0.06	
合計	14610		13463		13994		-0.04	

## 【XDDPプロセス分析(全体)】

### XDDP プロセス分析

XDDPのプロセスに関するマトリクスを表示します。

[要求数]

要求数合計	変更要求数	追加機能要求数
118	106	12

[要求仕様数]

要求仕様数合計	変更要求仕様数	追加機能要求仕様数
655	580	75

[工数・サイズ見積もりの情報]

BLOCK	初期見積もり		調整見積もり		実績		見積もり偏差(乖離)	
	サイズ(L)	工数(h)	サイズ(L)	工数(h)	サイズ(L)	工数(h)	サイズ	工数
SYS1	3515		1699		1389		-1.53	
SYS2	5935		5190		5756		-0.03	
CP	3920		5276		5523		0.29	
CV	1240		1300		1326		0.06	
合計	14610		13463		13994		-0.04	

[不具合情報] ※母体の残存不具合は除く

不具合件数	単体テストで検出	設計内テストで検出	QAで検出(不具合流出)
46	39	6	1

[生産性・信頼性]

生産性(LOC/H)	不具合含有率(KLOC)
123.2 LOC/H	0.5 件/KLOC

[レビュー指摘件数]

合計	Sランク	Aランク	Bランク	Cランク
643	23	92	192	336

[レビューマトリクス]

ページ数	所要時間	欠陥密度(件/頁)	指摘密度(件/h)
3175	132.25	0.2	4.86

[不具合検取状況]

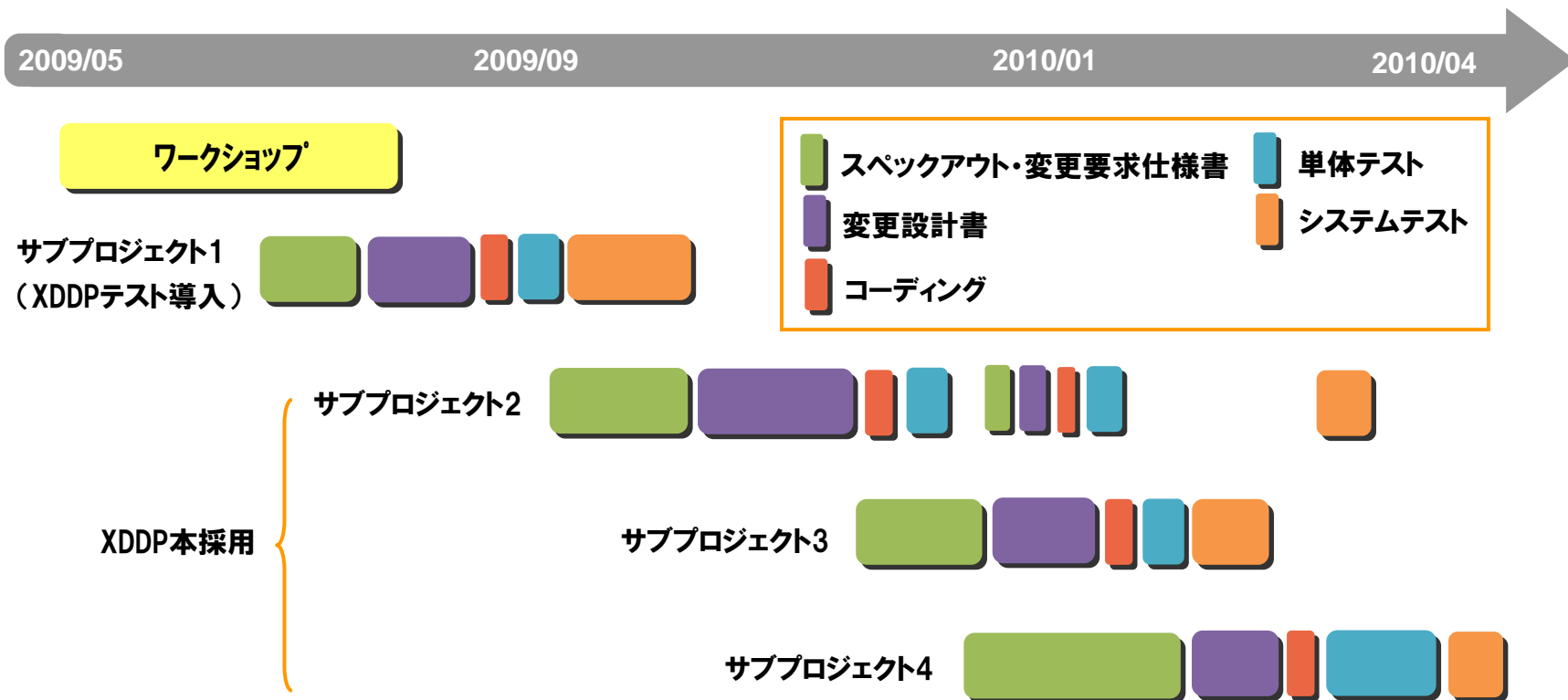
識別	レビューにて検取	単体テスト完了時	設計内テスト完了時	不具合流出
工程検出率	93.32%	5.66%	0.87%	0.15%
累計検出率	93.32%	98.98%	99.85%	--

工数・サイズ見積もり

生産性・不具合含有率

レビュー実施状況





## <ポイント>

- H/W側の都合により、プロジェクトを4つのサブプロジェクトに分けイテレーションを行った。
- スペックアウト／変更要求仕様書／変更設計書の作成に大部分の工数を費やしている。
- 五月雨的開発にも適用可能である。
- 外部的な変化に柔軟に対応できている。

<変更規模:15KLOC、要求仕様数655件 時のデータ>

	項目	前回PRJ (XDDP導入前)	今回	改善効果
C	ソフトウェア生産性	37.4 LOC/h	123.2 LOC/h	大幅up
Q	設計内テスト不具合数	98 件	6 件	大幅改善
Q	設計内不具合含有率	8.2 件/KLOC	0.5 件/KLOC	大幅改善
Q	QA不具合件数	9 件	1 件	大幅改善
D	納期遅延日数	0日	0日	遅延無し

このようにXDDPを基軸としたプロセス改善を実施し、  
XDDPのテスト導入プロジェクト(変更規模:2KLOC)で期待する効果が確認できたので、  
次のプロジェクト(変更規模:15KLOC)より本格的にXDDPの導入を行った。  
スペックアウトに時間を要した為、設計工数15%削減とまではいかなかったが、  
品質(Q)、納期(D)、コスト(C)目標の同時達成が実現できた。

## 1. XDDP導入による改善効果

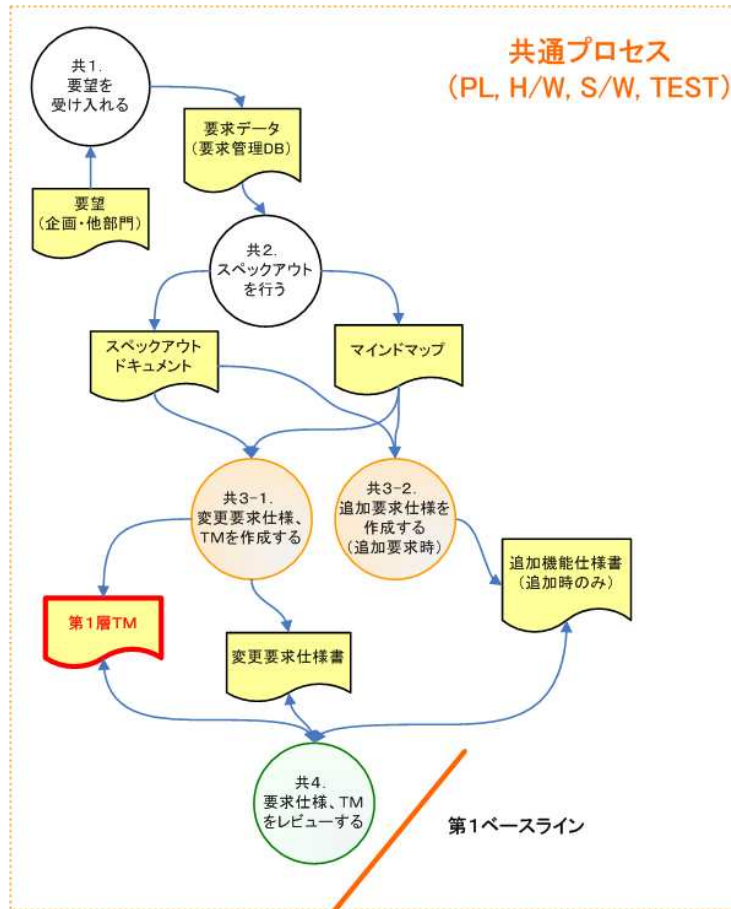
このように派生開発にマッチしたプロセス (XDDP) を導入し、忠実に実行することで、上流工程から品質が確保でき、無駄な後戻り作業を減らすことが可能となった。

## 2. ボトムアップ型でのXDDP導入成功

メンバ全員がXDDPを学習/理解し、XDDPのプロセスを忠実に実行し、プロセスの管理・監視を行うことで、ボトムアップ型のXDDP導入でも十分対応できることが確認できた。

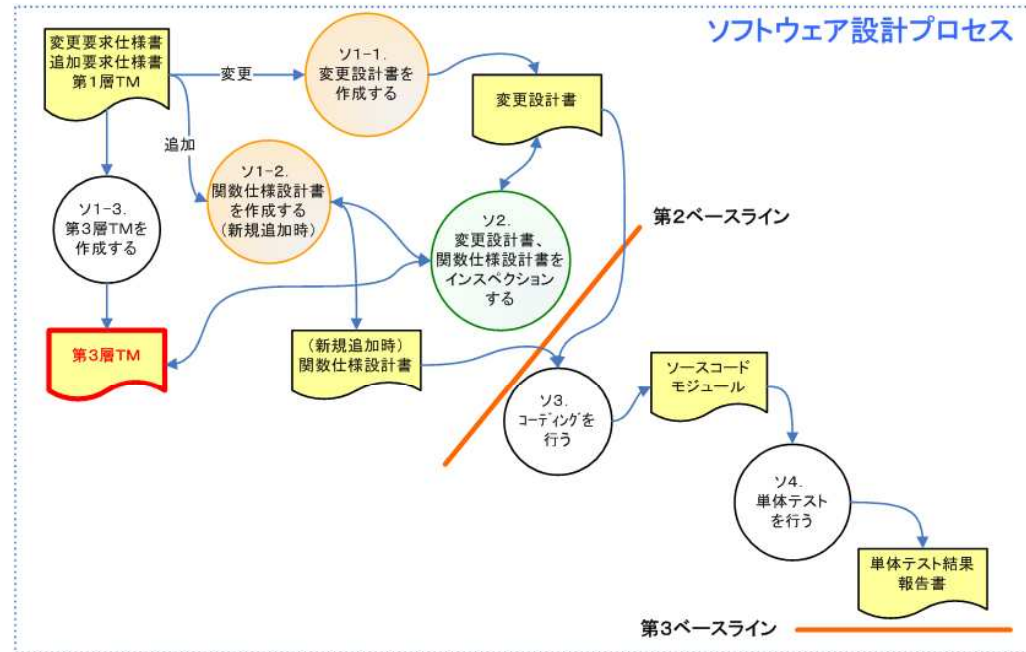
## 3. 設計者の意識向上

ボトムアップ型でXDDPを導入したことで、設計者の品質への意識、ムダな工数の削減に対する意識が格段に向上した。



## 第3層TM:S/W専用 (通常のTMに相当)

列要素:関数、ファイル

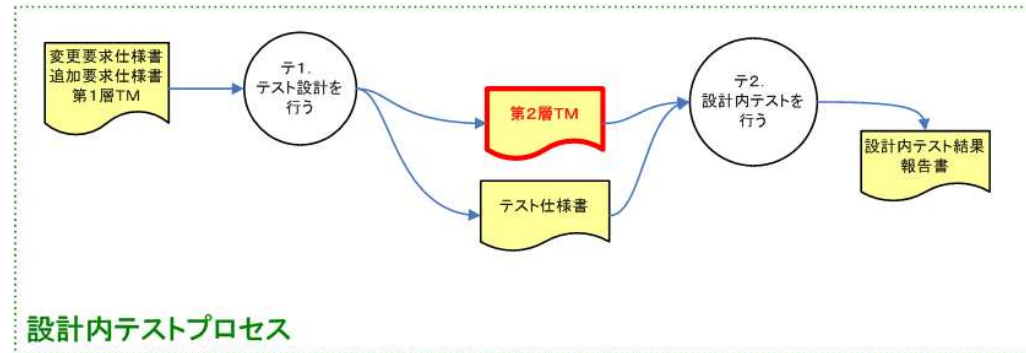


## 第1層TM:S/W, H/W, テスト共通

列要素:ブロック、大機能

## 第2層TM:テスト専用

列要素:機能



## ● ピアレビュー

### <対象物>

- ・ 変更要求仕様書
- ・ 追加機能仕様書
- ・ 第1層TM

### <参加者>

- ・ PM
- ・ ハードウェアG
- ・ ソフトウェアG
- ・ テストG

## ● インスペクション

### <対象物>

- ・ 変更設計書
- ・ 関数仕様設計書
- ・ 第3層TM

### <参加者>

- ・ ソフトウェアG
- ・ テストG

※ドキュメントの特性と参加者の層の違いで分けた。

[レビューメトリクス]

ページ数	所要時間	欠陥密度(件/頁)	指摘密度(件/h)
3175	132.25	0.2	4.86

[不具合検取状況]

識別	レビューにて検取	単体テスト完了時	設計内テスト完了時	不具合流出
工程検出率	93.32%	5.66%	0.87%	0.15%
累計検出率	93.32%	98.98%	99.85%	--

※前回プロジェクト(XDDP未導入)でのレビュー検取率は約65%

**XDDPの成果物を上流工程で関係者全員でレビューを行うことが非常に効果的である。**

XDDPの導入により、不具合件数が大幅に削減できた。  
しかし、「変更要求仕様漏れによる不具合」はまだ発生している。

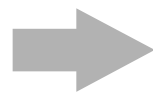
スペックアウトにもっと時間を掛けていれば...。  
トレーサビリティマトリックス(TM)をもっと細かい粒度で作成していれば...。  
果たして、本当にそうだろうか。

スペックアウトに時間をいくら費やしても、TMの粒度をいくら細かくしても、  
「変更要求仕様漏れによる不具合」を0件にすることは難しいと思った。

そこで、スペックアウトの段階で、変更要求をセントラルイメージとした  
マインドマップを作成し、マインドマップによる新たな気づきを活用した。

左脳利用 : スペックアウト

右脳利用 : マインドマップ



マインドマップ活用後、「明らかな漏れ」による  
不具合は発生していない。

※スペックアウトにどのくらい時間を掛けるのが最適か。(今後の課題)



## Q1. XDDPを導入して良かったことは何ですか？

- ・途中での仕様変更が以前より減ったと感じています。
- ・他ブロック (H/W含む) が何を変更するかが関係者間で共有できていることで、仕様漏れや実装漏れが軽減していると思います。
- ・レビューやインスペクションで指摘を受け易くなった。
- ・修正箇所やその周辺の処理が事前に明確になり、複数人での確認ができ、ソースコード修正前に誤りや抜けを見つける事ができた。明らかに仕様の漏れや手戻りは減らせたと思う。

## Q2. XDDPを導入して大変だったことは何ですか？

- ・資料作成、調査に時間がかかる。でも時間をかける価値がある。
- ・スペックアウト→変更要求仕様書→変更設計書と進んだところで変更が入り、再度スペックアウトからとなると、資料作成に費やす時間が期限に対して重い。

## Q3. 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

- ・次回改善したいと思う点がいくつかあります。是非、続けていきたいです。
- ・XDDPのプロセスに慣れた時、より効果が実感できると思うので続けたいです。何回かやってみた後でないと本当の判断が出来ないと思っています。

## 1. 関係者全員の合意を得る・共通認識を図る

- ・XDDPに対してプロジェクト全員の足並みを揃える。全員合意の上で導入する。
- ・グループでの学習が効果的。脱1人プロジェクト。

## 2. XDDPのルールを忠実に守る(プロセスの管理・監視)

- ・XDDPのルール(書籍に書かれていること)を忠実に守ることで効果が発揮される。
- ・XDDPのプロセスが守られているか常に管理・監視を行う。
- ・偽装工夫厳禁！

## 3. XDDPを信じて最後までやり通す

- ・最初は変更規模が少ない案件でパイロット導入を行い、手ごたえを実感する。
- ・導入すると決めたら最後までやり通す。
- ・個人的には、XDDPの一部(ex.USDMのみ)を導入するよりも、開発の1ブロックにXDDPを完全適用した方が効果的だと思う。
- ・一度XDDPの効果を体感したら、以前のプロセスにはもう戻れない！



## 1. XDDPを「個」から「群」へ（トップダウンアプローチとの融合）

今回はプロジェクト単体でのボトムアップ型でXDDPの導入を始めたが、導入効果を示す事ができたので、今後はトップダウン型と融合して組織全体にXDDPを広めていきたい。また、H/W設計にもXDDPを導入していきたい。

## 2. XDDPのルールを遵守しているか定期的な確認を行う

XDDPのルールに逸脱している箇所が無いか定期的に見直しを行う。  
→ XDDPプロセス監視者／管理者を設ける。

## 3. 削減した工数で今後への準備を行う

今回、納期遅延は発生しなかったが、工数15%削減という目標には至らなかった。今後、工数を要した箇所を改善し、削減できた工数で新技術の導入検討や、更なるプロセス改善等の今後への準備に投資できるようにしたい。

**ご清聴ありがとうございました**