

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株式会社（PFSC）

開発センター

厚田 鳴海

目次

- 1. PFSC 製品におけるソフトウェア**
- 2. 背景**
- 3. Stage1:派生開発WG活動開始**
- 4. Stage2:XDDPコンサルティング**
- 5. Stage3:XDDPの展開**
- 6. 今後の取り組み**

PFSC製品におけるソフトウェアの特徴

電子部品実装システムに求められるソフトウェアの特徴



製品の特徴

ソフトウェアの役割

カスタマー独自の生産形態に対応

カスタマ毎ソフトウェア・カスタマイズ
(ハードウェアは標準品でも
ソフトウェアはオーダー品)

コンシューマ商品のトレンドを
先取りした実装機能の提供

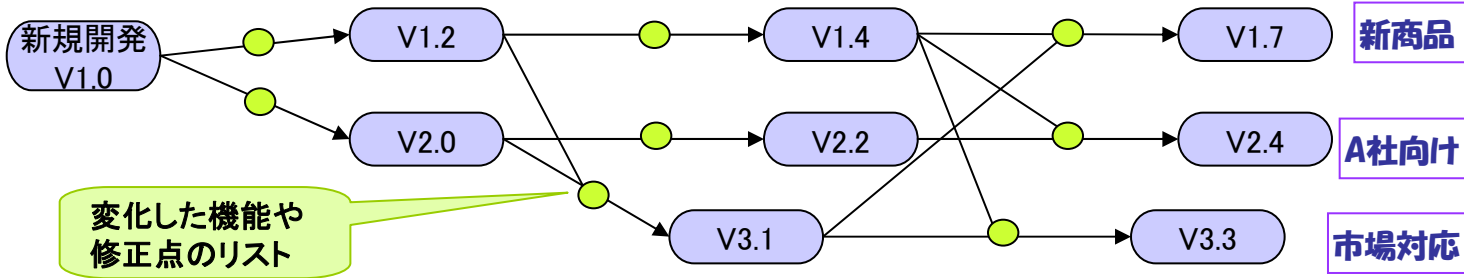
市場の要求変化を先取りする
短納期での開発

省スペース(スリム化)、
環境、コストダウン

ハードウェア依存から、
ソフトウェア依存へのシフト

XDDPの導入 -ほとんどの商品開発は派生開発

派生モデル開発のイメージ

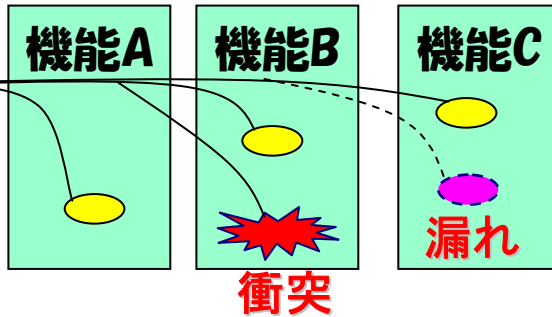


派生開発の特徴

- ・ベースモデルをもとに大量の仕様変更(不具合是正含む)
- ・新規より短い納期
- ・当初の設計思想が崩されている

現 状

機能追加



現象

- ・機能追加に伴ってトラブル多発
- ・時間不足、テスト不十分な状態でリリース
- ・今まで動いていた機能が障害発生

要因

- ・複雑な仕様の衝突
- ・見極められない影響範囲

派生モデル開発プロセス(XDDP)の導入

変更を適切にソフトウェアへ組み込む、派生モデル開発専用のプロセスが必要

XDDP
(eXtreme Derivative Development Process)

TM(トレーサビリティ・マトリクス)

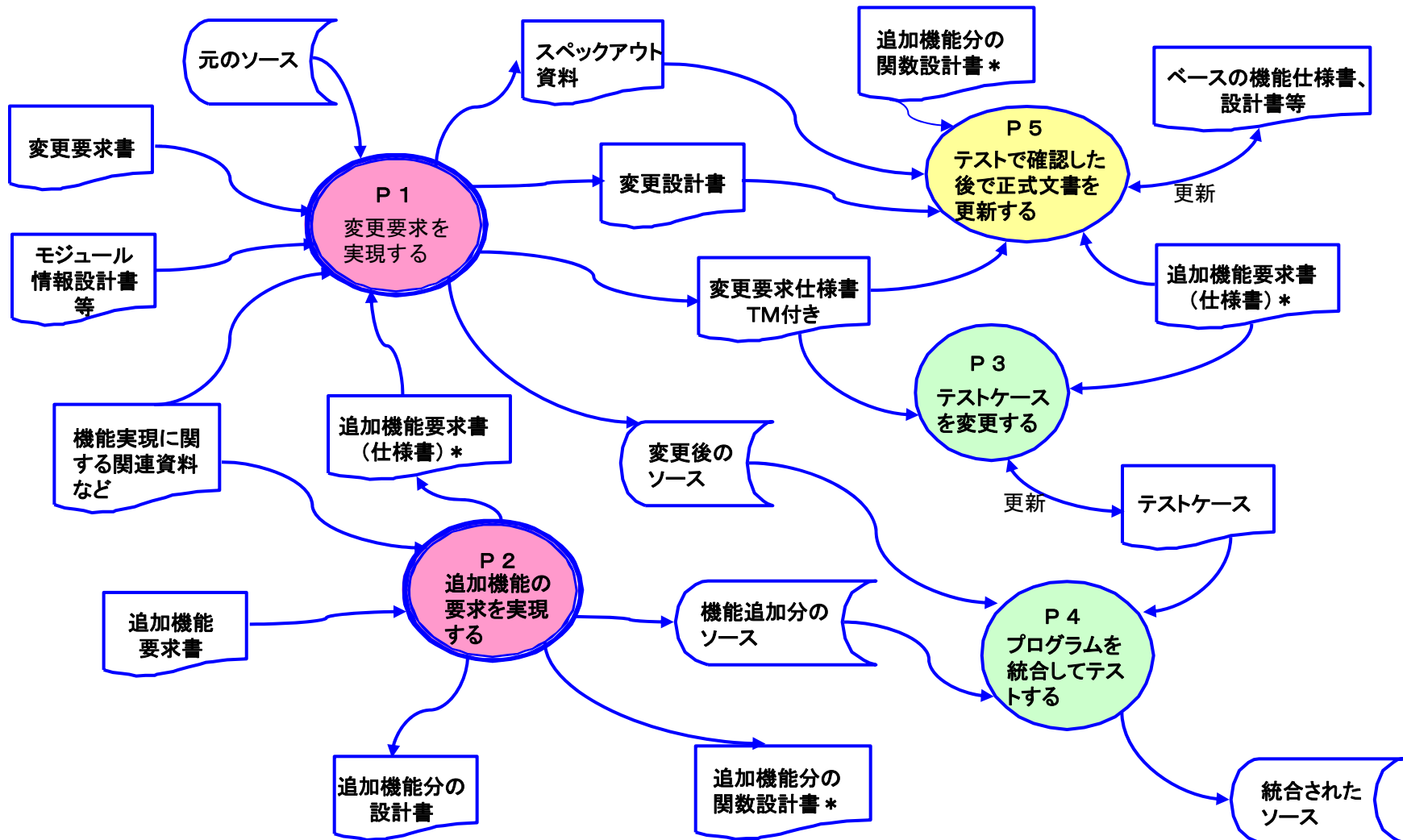
#	変更要求仕様	A	B	C	D	E	F	G	H
5	画面に通信記録の表示を追加する								
5.1	接続状況の表示の大きさを●に変更する				●				
5.2	表示用メモリの配置を●に変える						●		
5.3	受信時データの区切りにコードを挿入する								○

変更要求仕様

変更
設計書

XDDPの基本プロセス

- ・「追加」と「変更」では要求が違う → プロセスも異なる
- ・それぞれ「①変更用PFD」「②追加用PFD」として下位のPFDで展開 (PFD:Process Flow Diagram)



派生開発推進活動の経緯

2007年度

2008年度

2009年度

Stage. 1

派生開発WG活動開始

1. WG活動

- ・メンバー 9名

2. 活動内容

- ・XDDPテンプレート作成
- ・メンバーによる試行

3. 研修受講

- ・代表者によるPC本社
セミナー受講

Stage. 2

XDDPコンサルティング

1. WG活動

- ・メンバー 15名

2. 活動内容

- ・オープンコンサルティング
- ・事例集作成
- ・XDDPガイドライン作成
- ・活動HP公開
- ・機種展開

3. オンサイトセミナー

- ・「派生開発セミナー」
- ・「プロセス改善の勘所」

Stage. 3

XDDPの展開

1. WG活動

- ・メンバー 16名

2. 活動内容

- ・XDDP公開レビュー
- ・XDDP勉強会の実施
- ・保守プロセス(XDDP)
ガイドライン作成

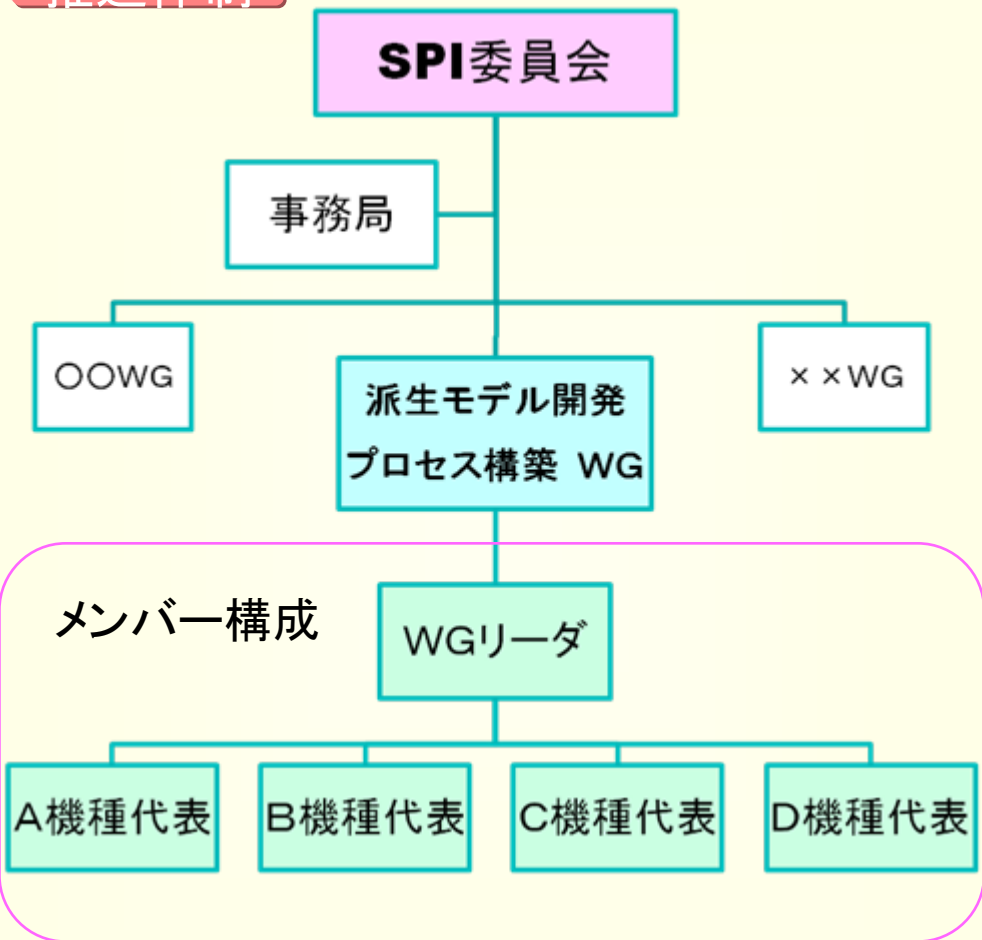
3. XDDPの展開

- ・見積もり時にUSDMを使用。
- ・メカ、ハード、ソフトとの連携に
USDMを試行。
- ・構成管理ツールとの連携

Stage. 1 派生開発WG活動の開始

Stage. 1: 派生モデル開発プロセス構築WG (2007/6~2008/3)

推進体制



取り組み内容

1. XDDPの検討
 - ・変更要求仕様
 - ・TM(トレーサビリティマトリクス)
 - ・変更設計書
2. 派生モデル開発プロセスの構築と実践
3. 関連メンバー(WG)との連携

しかし、

正しくできているのか？

TMのモジュールが
列のMAX超えちゃ
うよ

ここまで書いたらコ
ーディングしたほ
うが早いんじゃない
い

追加と変更
分ける必要
ないよね

要求と仕
様って何が
違うの



オンサイトセミナーでのアドバイス

活動1年後、オンサイトセミナーにてディスカッション

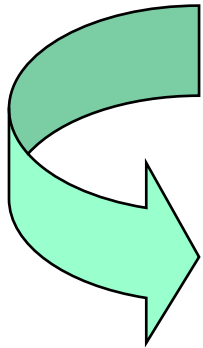
しかし…

やってはいけない「工夫」をやっていた！



正しいやり方で実施しないと、効果は得られない

セミナーでの
アドバイス



- × 変更要求仕様書に要求の内容しか書けていない
…従来の習慣で、機能仕様書に書いていた粒度で変更内容を記載。
before/afterになっていない。仕様になっていない(要求レベル)
- × 変更要求仕様書(TM付き)の「行」にモジュールを書く
…変更仕様とモジュールが別シートになるとTMの気付きが得られない。
- × 変更設計書の変更内容に変更仕様が記載されている
…従来の習慣で、何を直すかを変更設計書に記載。どう直すかの記載はなし。コーディング時に迷うため、生産性が落ちる。



・今までの習慣でやってはだめ。これを変えないといけない。
・まずは、私のテンプレートでやってみて。私と同じ景色が見えるようになってから工夫するように。

XDDPを効果的に実施するために

XDDPで効果をあげるための条件

- ◆ TOPの支援が得られること
- ◆ オープンコンサルティングを実施できること
- ◆ プロセス設計の仕方を合わせて学ぶこと

Stage. 2 XDDPコンサルティング

Stage. 2: XDDP WG (2008/4~2009/3)

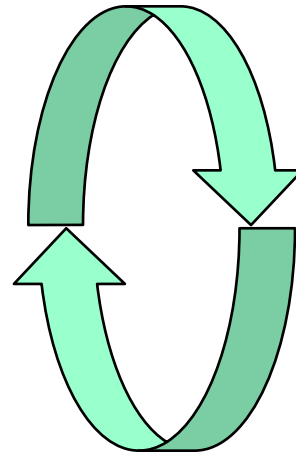
XDDP WG

【取り組み】

- ・発表者の支援、コンサル内容のフォロー
- ・ノウハウのまとめ、ガイドライン事例集の作成、
- ・メンバー教育

【効果】

- ・自主性の発揮
- ・取り組みの中での不安や疑問の解消
- ・仲間同士での議論の場の提供
- ・組織の中での指導者の育成



オープンコンサル

- ・個人または開発チームとコンサルタントのやりとりを聴衆に見せる方法。
- ・自由参加、強制はしない。
- ・参加者は、誰でも意見、質問可能
- ・取り組みに課題を持っている人は、アドバイスを自分の問題として聞くことができる

WGとコンサルティングの併用で短期間での習得、組織展開を狙う

オープンコンサルティング

■ 期間:

- 2008年7月～2009年3月 9ヶ月(ほぼ1回/月オンサイト)

■ コンサルタント:

- (株)システムクリエイツ 清水吉男氏

■ 取り組み:

- XDDPを中心にした派生開発におけるQCDの改善活動

① PFDを使ったプロセスの設計

② USDMによる要求仕様(変更要求仕様)の作成

③ XDDPによる派生開発アプローチ

■ 「オープンコンサルティング」の効果:

- 能動を誘う方法で副次効果が大きい
- 組織の意思が機能したことで、多くの参加者を得た
- 取り組みがオープンになることで安心して取り組める

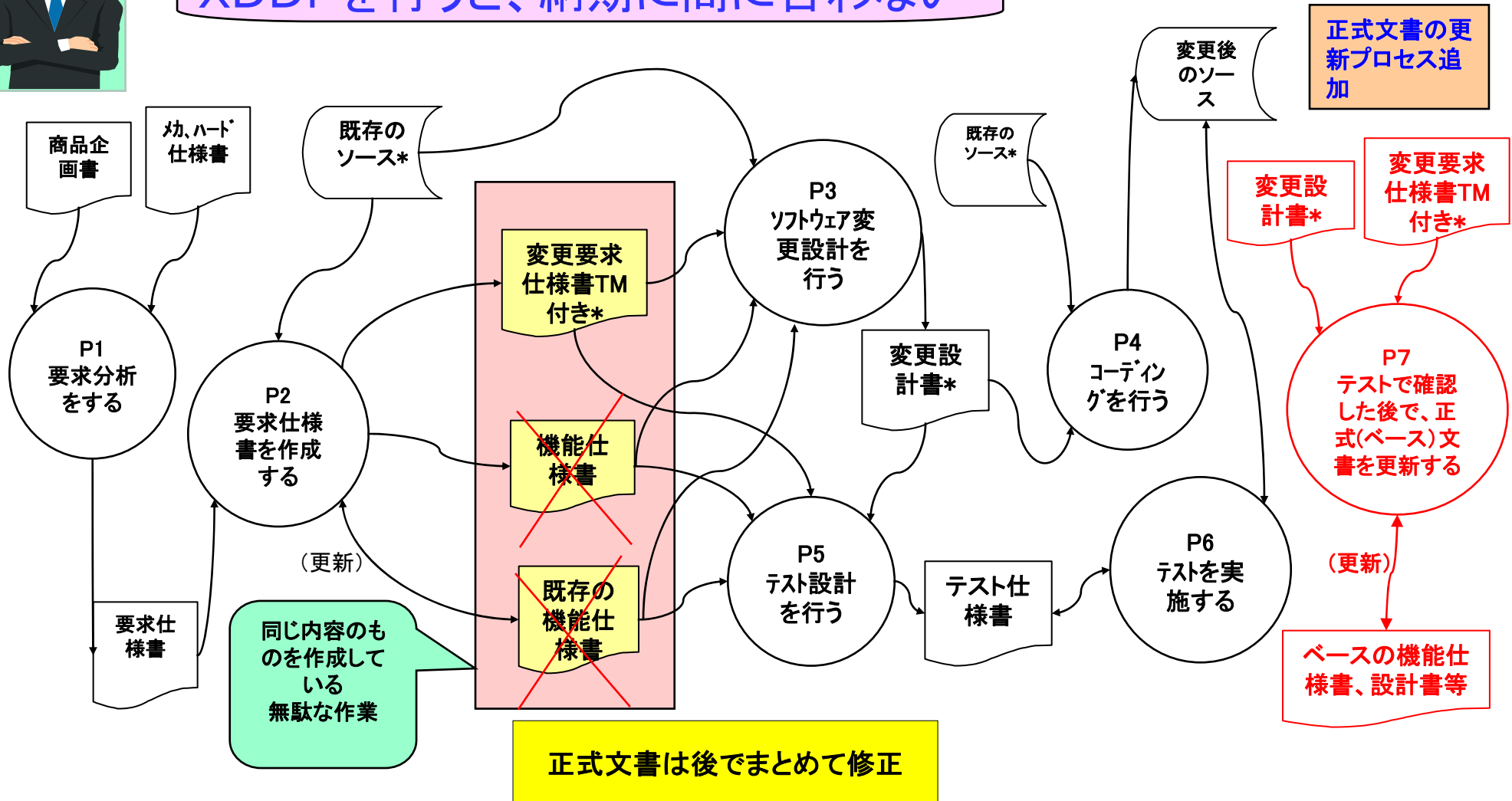


・参加は自由
・マネージメント層も出席

・WGとの連携
・組織展開の加速

プロセス改善事例①

XDDPを行うと、納期に間に合わない

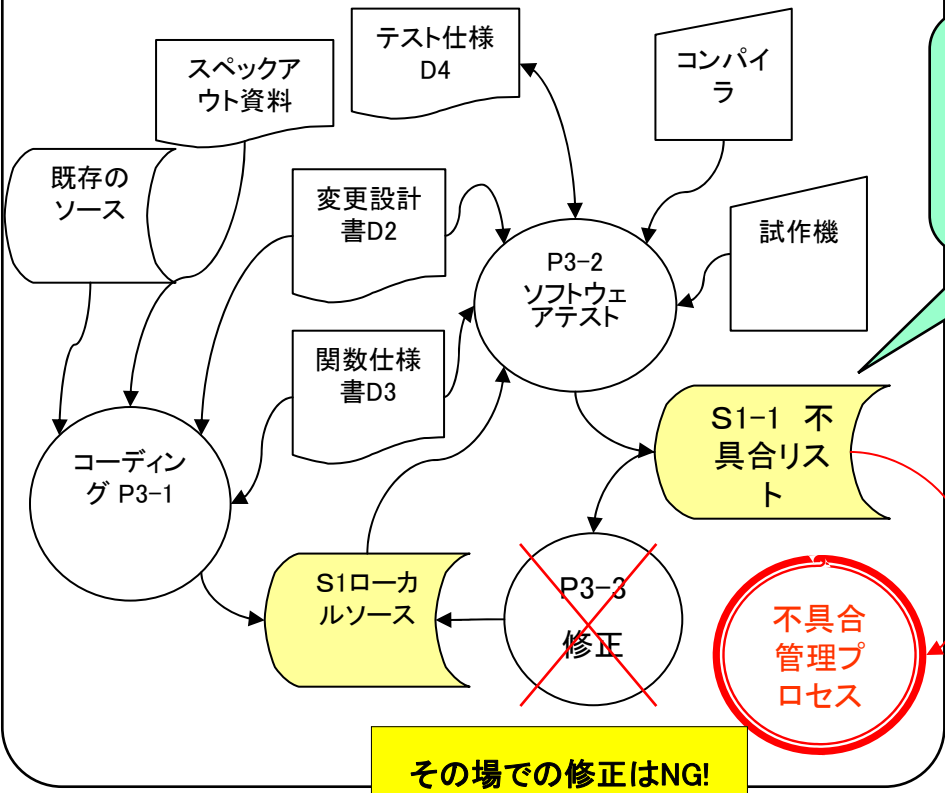


プロセス改善事例②



XDDPを行っても品質が改善しない

プログラム製造プロセス

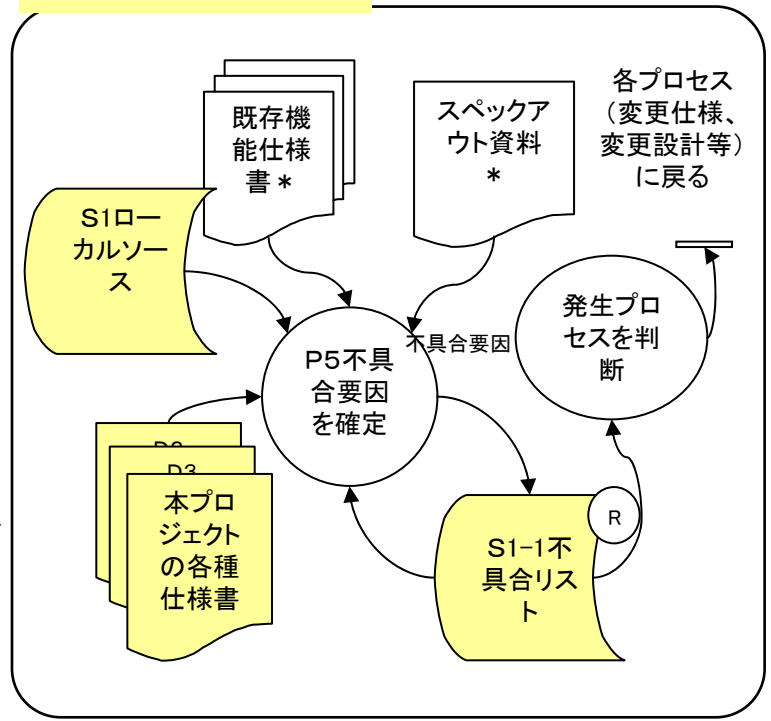


個人が判断すると、勘違いしたまま修正する可能性がある

不具合管理プロセスを通す

その場での修正はNG!

不具合管理プロセス



バグを出したプロセスでバグを修正しない！！

組織展開における抵抗の克服

XDDP導入ハードルを下げるための施策

ハードル

施策

“XDDP”に対する不信

マネージャからメンバーへの声かけ。“工数が倍になっても、品質がよかったら元が取れる。まずは、納期を心配せずにやってみて。”

新しい取り組みに対する抵抗

オープンコンサルの実施により、ポイントをつかめるようにする。ガイドライン、事例集を整備し、始めるときの参考にする。

仲間(サポート)がいない
ことに対する不安

不具合修正も変更要求ととらえ、XDDPで修正する。変更量少なく、小さく始められる。デグレードが無くなり、成功体験が自信になり、継続する。

WGメンバーを中心とした仲間づくり。
周りがやっていると安心できる。わからないことも聞くことができる。

効果－生産性

生産性： XDDP適用による生産性は非適用の**0.95倍**

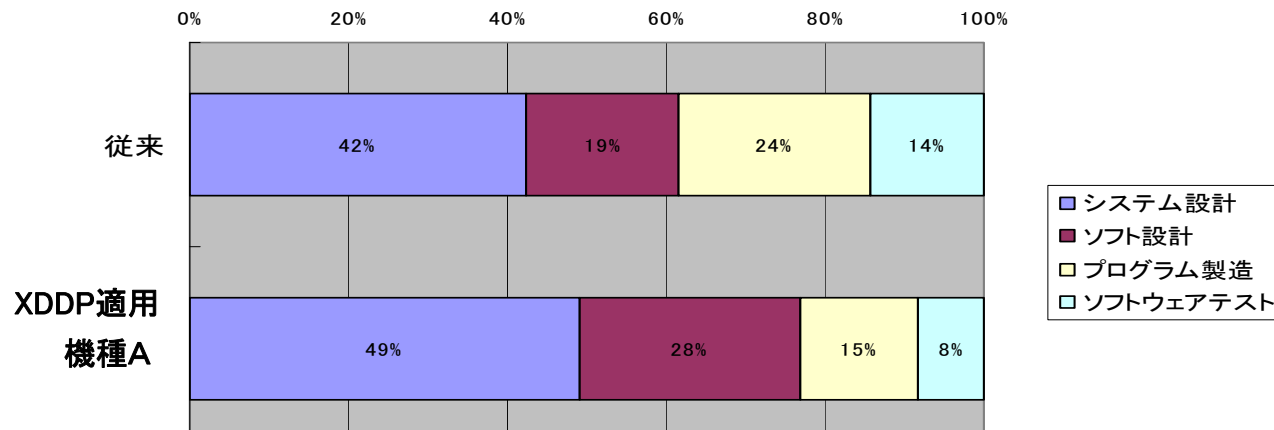
(但し、開発Step数100以上は**1.06倍**)

市場不具合が削減できれば、市場対応含めたトータルの生産性は向上できる。

■派生モデル開発適用、非適用での生産性比較(2008年度XDDP適用分)

	PG生産性	開発生産性
派生モデル開発 非適用 (過去生産性)	1	1
派生モデル開発 適用	4.1倍	0.95倍
開発Step数100以上	4.5倍	1.06倍

■従来プロセスとの工数比率比較

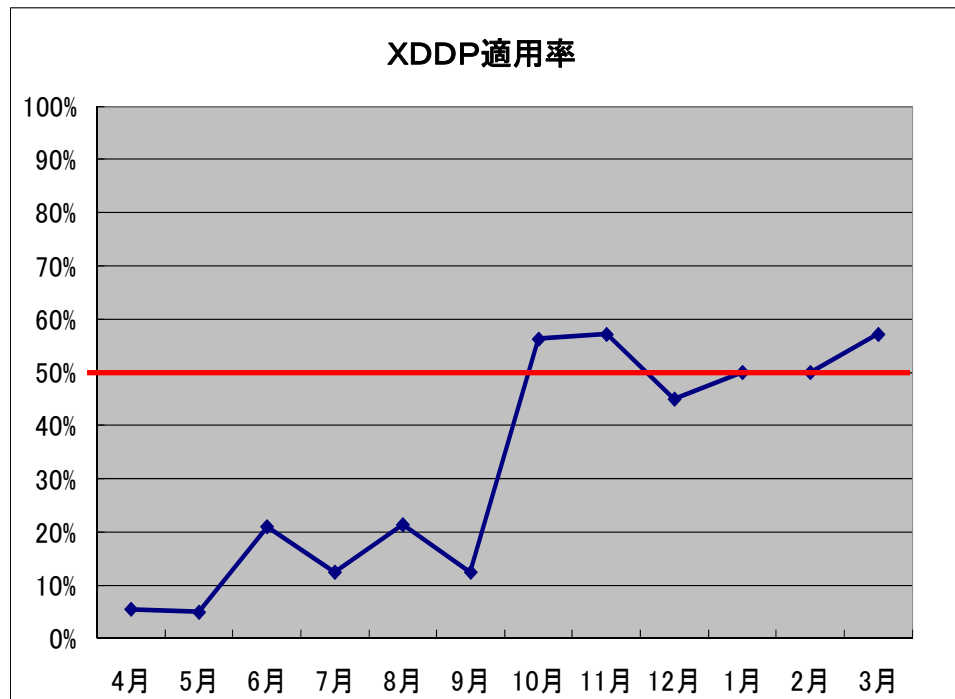


システム設計、ソフト設計に時間をかけプログラム製造を一気に行う。ソフトウェアテストもスムーズに進み時間短縮となっている。

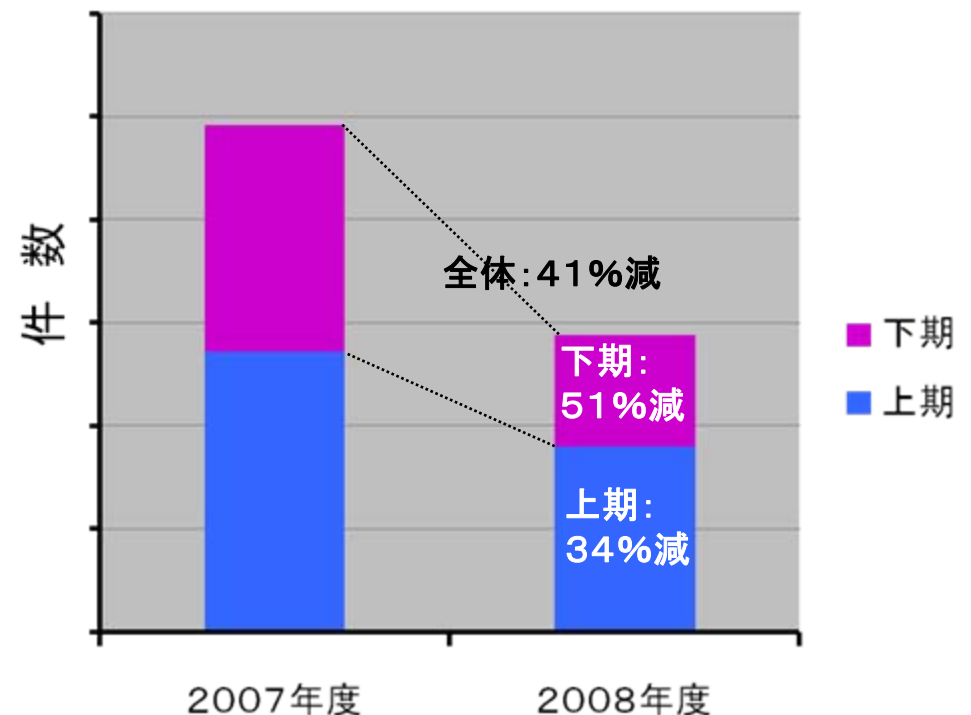
効果一品質

- ・2008年度下期のXDDP適用率は**50%以上**をキープ
- ・2008年度の不具合発生状況は2007年度の**41%減**
(下期のみの比較では、**51%減**)

XDDP適用率(2008年度)



不具合発生状況



メンバーの感想

工程	内容	よかったこと
変更要求 仕様書	機能仕様書は最後に書く	最終的に固まった仕様で綺麗に書ける
	変更内容をbefore/after形式で書く	「〇〇を●●にしたい」と書くことで、変更方法がイメージできる。 現仕様を確認できる。
	要求に対して理由を書く	仕様が的を得ているか判断しやすい
	階層構造で書く	概要と詳細がつかみやすい
TM	関数レベルで変更点を抽出する	ソースを調査するので、変更漏れが発見できる場合がある 仕様とソースのつながりが分かる
変更設計書	説明は文章で書く ソースコードを書いてはならない	・変更内容が文章で書かれている為、内容が理解しやすい ・変更方法が分かる為、テスト内容も考えやすい
コーディング	ぎりぎりまでソースコードを変更しない	・コーディング前なので、修正して欲しい箇所を遠慮なく言える ・仕様の内容を間違ってもコーディング前で作業がむだにならない

Stage. 3 XDDPの展開

Stage. 3: XDDPの展開実施中(2009/4 ~)

項目	課題	取り組み内容
マネージメントへの展開	<ul style="list-style-type: none">顧客要件に対する工数見積もりの誤差が大きい。	<ul style="list-style-type: none">見積もり時に、USDMを活用。変更点を詳細に列挙、変更仕様別に工数見積もりを行うように変更。
メカ、ハード協調への展開	<ul style="list-style-type: none">メカ、ハードを伴うシステム変更時、開発後半での仕様変更多発。	<ul style="list-style-type: none">プロジェクト開始時に変更ユニット毎に、レビュー時にUSDMを活用。各担当毎に詳細を合意ののち、個別の開発開始。
構成管理への展開	<ul style="list-style-type: none">機能ごとにXDDP帳票(3点セット)が多数作成されるため、全体の関連が見えない。	<ul style="list-style-type: none">Web上で、誰が、どの件名を、どこまで対応しているか把握できるツールを作成。(市販ツールカスタマイズ)構成管理ツールとの連動実施。

今後の取り組み

■ 大規模システムにおける影響範囲の見える化

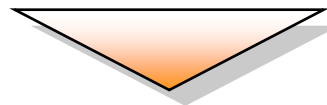
規模が大きくなると、変更に対する影響範囲が広がってくる。影響範囲を抑えて、変更要求を適切にソフトウェアに組み込むための仕組みが必要になる。

■ 不具合の分析とフィードバックプロセスの確立

ソフトウェア構造の複雑化等の理由により、不具合が漏れることがある。派生開発の場合、ベースが同じため、同種の不具合が再発する可能性が高くなる。発生した不具合を分析し、開発プロセスの脆弱性を適切に改善するプロセスを確立する必要がある。

■ 派生開発をスムーズに行うための新規開発プロセスの見直し

新規開発したシステムもすぐに派生開発へ移っていく。新規開発時にしか取り扱わない内容もあるので、その点を考慮し、スムーズに派生開発へ移れる新規プロセスを再構築する必要がある。



継続的な改善と横展開の推進で「QCD同時達成」を目指す！