

デジタル複写機
のネットワーク
開発へのXDDP
適用事例

RICOH
imagine. change.

～ 製品開発をラクに続けていくために～

2013年5月24日
リコーITソリューションズ株式会社
エンベデッドソリューション事業部
宮浦 直人

IT for
Innovation

■ 自己紹介

- 宮浦 直人(みやうら なおと)
- リコーITソリューションズ株式会社
エンベデットソリューション事業部
東日本開発センター ES第1開発部
第1グループ リーダー
- リコーが販売するデジタル複写機の
ネットワーク制御部の機能開発及び、
市場保守を担当



■ アジェンダ

1. 背景

2. 導入

3. 試行と工夫

4. 取組みの効果 (データで分析)

5. まとめ

■ アジェンダ

1. 背景

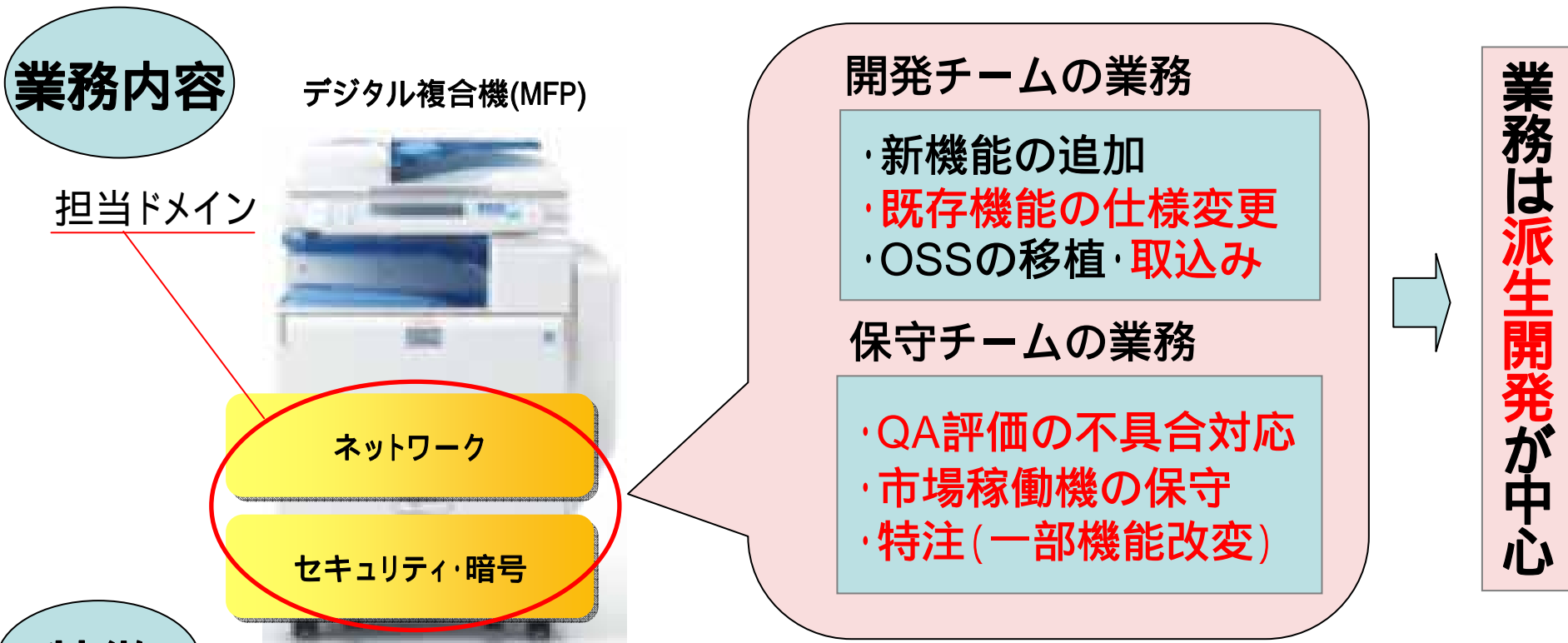
2. 導入

3. 試行と工夫

4. 取組みの効果 (データで分析)

5. まとめ

1. 背景 - チームの特徴



特徴

- 機能開発と保守の同時対応
- 扱うソフトウェアの規模増大、複雑化が進行
- 品質と生産性向上の定常的な要求

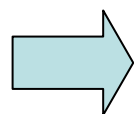
限定された
人員で対応

1. 背景 - チームの困りごと

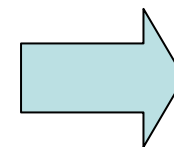
チームでは、常々こんな困りごとを抱えている

■ 開発工数の不足

QA評価で想定以上の不具合発生

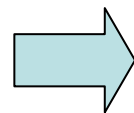


新規開発の工数で支援

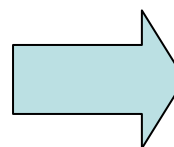


■ デグレードによる手戻り

ちょっとした仕様の変更で副作用が発生

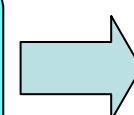


不要な手戻り工数が発生



■ 生産性の低迷

日々の業務に追われて開発者に余裕がない



改善活動に手が回らない



残業によるリカバリーで
開発者の負荷が増加

1. 背景 – 製品開発をラクにするためには

限られた人的資源で設定されたQCDを確実に達成し、製品開発を進めるには、**2つの課題の同時解決**が必須！

課題1.
不具合を出さない

課題2.
開発生産性のUp

従来のやり方を変えたい



偶然、社内の有識者に紹介してもらった
XDDPが課題解決の手段として使えそう

■ アジェンダ

1. 背景

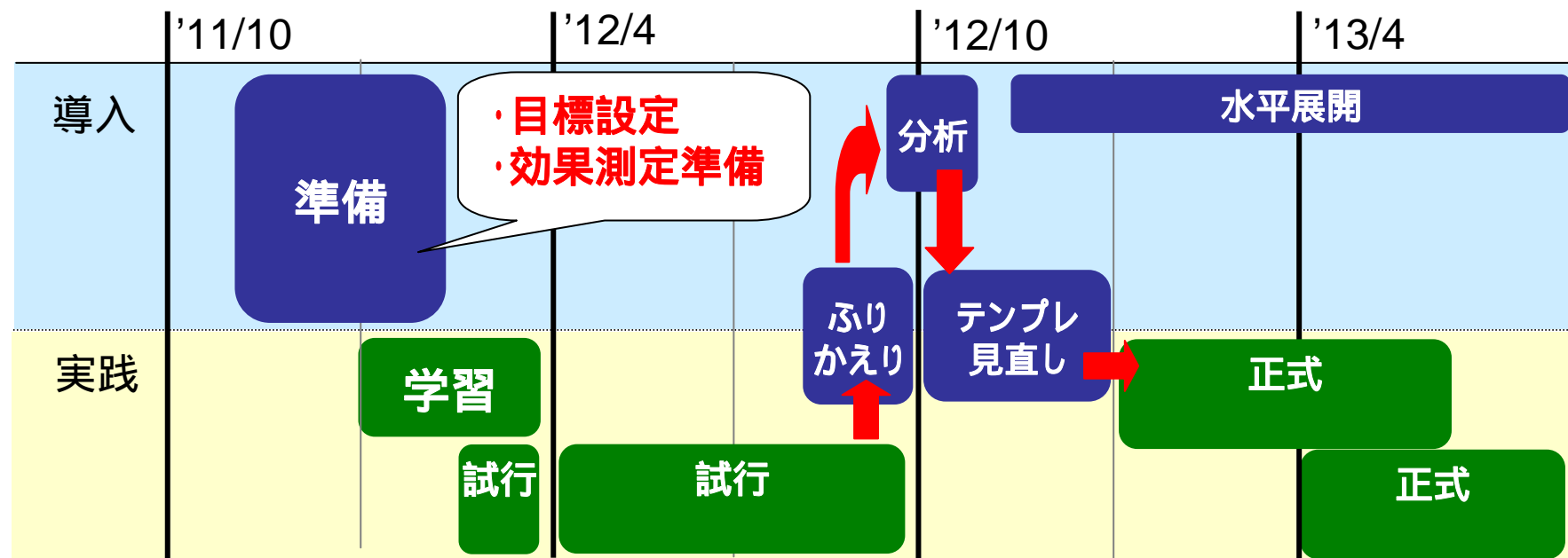
2. 導入

3. 試行と工夫

4. 取組みの効果 (データで分析)

5. まとめ

2. 導入 - 導入と実践の流れ



<試行プロジェクトの規模>

発表はここまで

プロジェクト	変更規模(KLOC) 括弧内は母体	変更要求数 (件数)	変更要求仕様数 (項目数)	開発期間	実施工数 (人月)	人員
試行	約0.3(30)	3	20	'12/02-03	2.0	1名
試行	約3.5(30)	32	196	'12/04-09	9.0	2名

■ 2. 導入 – 準備 (目標の設定)

■ 目標設定を実施

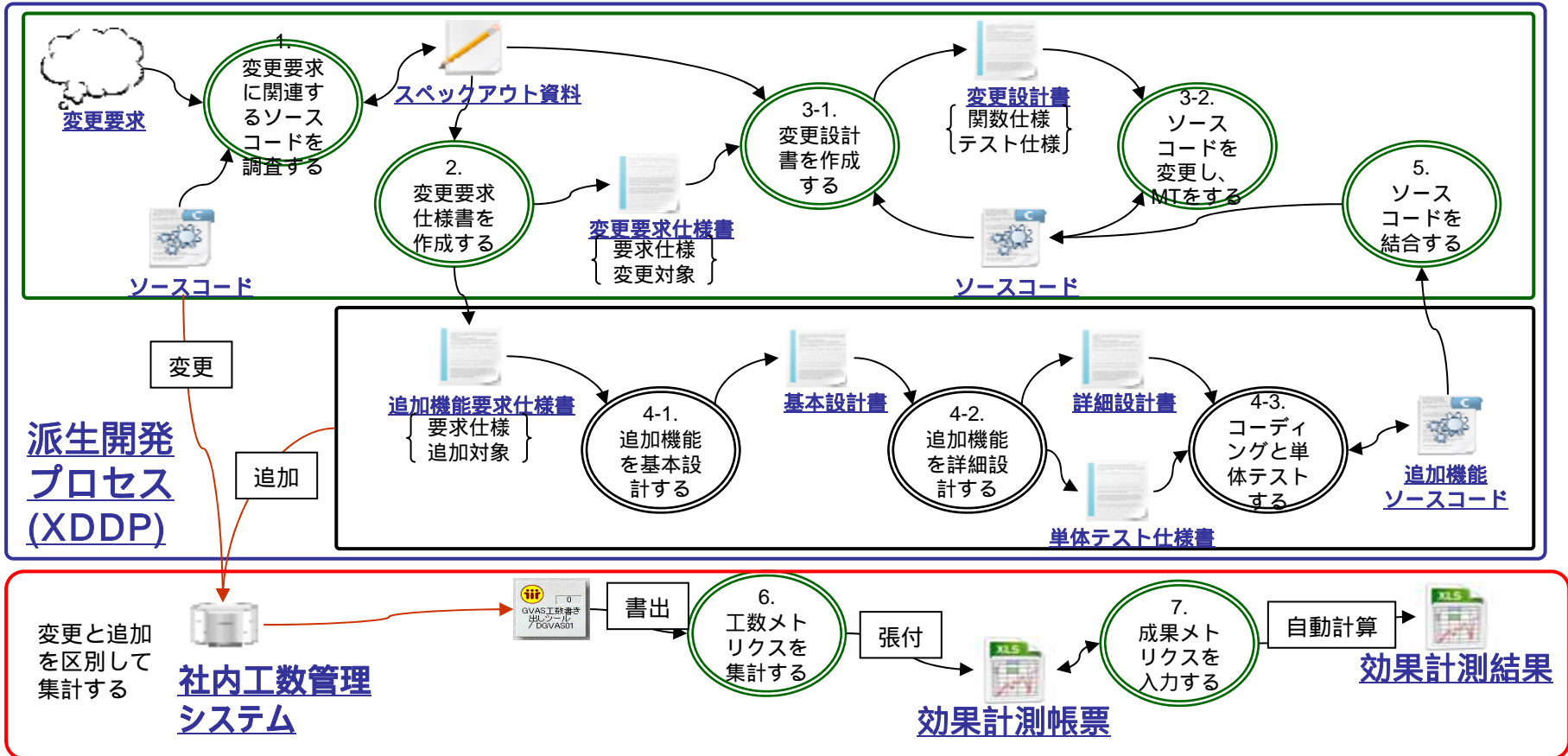
取り組み開始時に2つの課題と目標値を設定

課題	目標値
製品開発期間内で確実に品質を確保する	・障害密度: 0.18件/KLOC ・完了納期: 2W以内
開發生産性を向上させること	・コード生産性: 80L/h

■ 試行担当者とプロジェクト選定を実施

- ・担当者は、自ら興味をもって取り組んでくれる人を選定
- ・今後も継続が見込まれる開発案件を選択
- ・試行時の学習は、書籍と派生開発協議会のテキストを進めて、不明点は社内の有識者への相談会で解決

2. 導入 - 準備 (効果測定の流れ)



ツール(効果計測帳票)を使って実績を測定し、品質と生産性を定量で示すことができる

担当者が効果を実感

2. 導入 - 準備 (ツールの概要)

社内の工数管理システムから作業工程別の工数を抽出し、XDDPの3点セットに関連する生産性を自動計算

*****		***** 変更 *****	
XDDP 3点セットの データを それぞれ 参照可能	25.50	<変更要求仕様書>	
	6.25	変更要求仕様抽出時間(0)	25.00
	9.25	DF時間(0)	5.50
	9	DF指摘事項反映時間(0)	7.50
116	要求項目数	23	
		変更要求仕様項目数	80
	41.00	変更要求仕様に変更した工数(0) 38.00	
	4.55	1Hあたりに抽出した要求仕様項目 3.20	
	0.35	要求仕様項目1件あたりに変更した時間 0.48	
	0.00	<TM仕様書>	
	0.00	TM作成時間(0)	62.25
	0.00	DF時間(0)	5.50
	0	DF指摘事項反映時間(0)	10.00
		TMのセル交点数	246
	0.00	TMに変更した工数(0) 77.75	
	0.00	1Hあたりに抽出したセル交点 3.95	
	0.00	<変更設計書>	
	77.75	変更設計書作成時間(0)	
		詳細設計作成時間(0)	117.75

<狙い>

- 生産性を定量化することで、推進担当者や開発者が導入効果や実施上の問題点に気付くことができる
- プロジェクトを評価するときの準備工数を減らすことができる

■ アジェンダ

1. 背景

2. 導入

3. 試行と工夫

4. 取組みの効果 (データで分析)

5. まとめ

■ 3. 試行と工夫 – 試行メンバの構成

試行プロジェクトは、3名で構成。
役割を決めて、試行毎に変えながらすすめた。

- プロジェクト管理者 : 私(経験:13年)
全体推進、進捗管理、レビュー
- 担当者 : Hさん(経験:9年)
勉強会推進、変更要求仕様書、実装、モジュールテスト
- 担当者 : Oさん(経験:5年)
変更設計書、実装、モジュールテスト



3. 試行と工夫 – プロジェクトの特徴

サードベンダからの要求仕様公開を受けて
デジタル複写機の組込み機能の一部として取り込む

1. 仕様公開から機能搭載までの期限が存在する
(前仕様の有効期限が設定)

確実な納期達成が
MUST

2. 曖昧な要求記載が見られ、解釈誤りが生じる
可能性を含んでいる

要求分析と仕様の
落とし込みが重要

3. 開発完了後にサードベンダによる認証評価が行われ、
問題発生時の工数影響が大きい

手戻りを発生させない
品質の作りこみ

3. 試行と工夫 – 気づいたこと

試行が終わった段階で、担当者を交えてふりかえりを実施
成果物ごとに気づいた点を整理した

1. 変更要求仕様書

変更要求数が多い場合に、レビューで検出モレが起きた
追加機能要求を仕様化する際に、変更と追加の要求仕様の
関連を誤ったため、不具合が混入した
TMを作るのが大変だった

2. 変更設計書

「変更内容」と「確認項目」の関連付けがわかりづらく、
テスト項目が不足していたことに気づけなかった
例外条件の評価項目が漏れていた(テストの網羅性)



いずれも仕様書フォーマットの工夫で、何とかかなりそう...

3. 試行と工夫 – 要求仕様書

1	2	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1																
2																
3						TT/GVAS連携機能										
4		ユーザケース			UC.1	初期設定を行う										
5		□□□	要求		RFP.1	処理に必要な情報を初期設定する										
6			新規		理由	開発作業が実施している、TTとGVASの二重入力作業負担を軽減するため										
7					説明	起動時の日付指定により、Windowsタスクスケジューラ等で自動実行する際に初期設定で保存された情報を使用して処理を実行することを可能とする。										
9			□□□	要求	RFP.1.1	初期設定を行う										
10				新規	理由	処理に必要な情報を初期設定するため										
11				説明		起動オプションには、開始日と終了日を指定することが可能。指定がないとき、当日が処理対象となる。日付の形式は、「yyyy/mm/dd」とする。										
12			□□□	仕様		<起動オプションの判定>										
13						本ソフトウェアを起動したときに渡された引数（開始日と終了日）をチェックする。										
14						・引数=0（なし）のとき、当日のみを処理対象とする。										
19						・引数=2のとき、開始日（第1引数）、終了日（第2引数）の範囲を処理対象とする。										
23						・引数の形式がイリーガルな場合、<オプションエラー>を実行する。										
24						・開始日よりも終了日が前の場合、<オプションエラー>を実行する。										
25						※引数の形式は、「yyyy/mm/dd」とする。										
26			□□□													
27			□□□	仕様		<初期設定項目の入力>										
28			□□□	仕様		<初期設定項目の保存>										
29			□□□	仕様		<オプションエラー>										
29			□□□	RFP.1.1.7		オプションエラーを警告表示する。										
29			□□□	RFP.1.1.8		本ソフトウェアを終了する。										
29			□□□													
29			□□□	要求	RFP.1.2	初期設定を読み込む										
29				新規	理由	初期設定ファイルの内容を読み出し、変数の値として保持するため。										
29				説明		初期設定ファイルのシングルユーザーIDとパスワード、ノートパスワードをそれぞれ別の変数に値設定する。本バージョンでは、本アプリケーション利用前に、上記内容について手動にて設定を記述しておく必要がある。										
78																
79		ユーザケース			UC.2	TTの実績をGVASへ反映する										
80		□□□	要求		RFP.2	TTに入力済の実績を、TTより取出しGVASへ入力する										

工夫1
グループ表示可能とすることで、レビューでの見やすさを向上させた

工夫2
要求分類を新規・追加・変更に分類し、1シートで一元的に表現するようにした。



機能分類
tt2gvas_main.js
tt2gvas_sub.js
tt2gvas_tt.js
tt2gvas_ie.js
tt2gvas_excel.js
tt2gvas_file.js
tt2gvas_notes.js

工夫4
ソースコードの構造をグループで階層化するマクロを組み込みTM生成を簡単にした

工夫3
ブランクの要求仕様を作成、追加するマクロを組み込んだテンプレートを準備した

3. 試行と工夫 - 変更設計書

変更設計書では、主にテスト項目のモレを防ぐ工夫を実施

変更要求仕様	確認者	□確認	
#sample01.1.1	見積り行数	2	見積り時間
	変更行数		作業時間

工夫1
変更内容に対する確認項目を併記するようにした



変更内容:

項目#	変更内容	予想行数	<確認> 左から 基本・境界・ 限界・エラー
1	sample()で扱うパラメータAの上限値を100から200に変更する。	1	□■□■
	確認項目#		
	1-1	パラメータAを200に設定して、戻り値がXXXになること	□■□□
	1-2	パラメータAを201に設定して、戻り値がYYYになること	□□□■
2		1	■□□■
	確認項目#		

工夫2
必要なテスト条件が網羅されているかが一目でわかるようにした。

工夫3
社内標準プロセスで規定されている、エビデンスとして使用可能とした。

3. 試行と工夫 – 工夫点の狙い

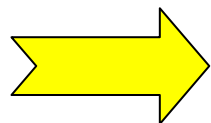
それぞれの工夫点の狙いを整理する

1) 変更要求仕様書

- レビュー時の気付きを向上させる(工夫点1, 2)
- ドキュメント作成の効率化を図る(工夫点3, 4)
- 利用範囲の拡大(派生開発以外で新規開発も意識)(工夫点1)

2) 変更設計書

- レビュー時でのテスト項目のモレ、誤りに対する気付きを向上させる(工夫点1, 2)
- 社内標準プロセスとの統合(工夫点3)



工夫点を反映したテンプレートは、直ちにXDDP取組み中のプロジェクトへ展開

■ アジェンダ

1. 背景

2. 導入

3. 試行と工夫

4. 取組みの効果 (データで分析)

5. まとめ

4. 取組み効果 - 品質向上

XDDPに取り組むきっかけとなった品質課題に対する効果を効果計測帳票ツールの出力結果から確認した

テーマ名	実施年度	変更(追加含)ステップ数(KLOC)	単体テスト		結合テスト		機能テスト	
			障害件数	障害密度	障害件数	障害密度	障害件数	障害密度
従来開発	2011	2.5	26	10.4	13	5.20	8	3.20
XDDP試行(1)	2011	0.3	12	40	1	3.33	0	0.00
XDDP試行(2)	2012	3.5	28	8	5	1.42	1	0.30

障害発生件数の減少という形で
確実に効果が出たことを実感出来た！

目標値
障害密度: 0.18
結果: ×

4. 取組み効果 – 開発工数

XDDPを用いると工数が増えるという声もあったが、実際にどのような効果が得られたか

目標値
納期: 2W以内
結果:

社内標準 マイルストーン	計画 (人日)	完了日 (人日)	結果 (計画差異)	XDDP
システム設計	10	10	○(±0)	-
モジュール設計	22	35	×(+13)	1)変更要求 2)TM 3)変更設計書
プログラミング	11	4	○(-7)	-
モジュールテスト	20	16	○(-4)	-
結合テスト	10	10	○(±0)	-
結果	73	75	○(+2)	-

・曖昧要求の明確化が必要だった
・仕様書記載の増加や不慣れ

・実装が明確だった(実装が捗った)
・不具合が少なく手戻りが減った

プログラミング以降の工程から上流工程への工数シフトが見られ、トータル工数は目標値に収めることができた

4. 取組み効果 - 生産性

各フェーズにおける生産性に対する効果を検証

目標値
コード生産性: 80
結果:

従来開発と試行プロジェクトとの生産性比較

テーマ	要件定義				設計				実装			
	工数 (h)	規模 (件数)	生産性 (件数/H)	テーマ比 (%)	工数 (h)	規模 (件数)	生産性 (件数/H)	テーマ比 (%)	工数 (h)	規模 (LOC)	生産性 (LOC/H)	テーマ比 (%)
2011年特注A	1.0	-	-	n/a	10.0	8	0.80	100	4.5	30	6.6	100
2012年機種開発B	1.0	-	-	n/a	20.0	5	0.25	31	3.0	50	16.6	251
試行(1)	13.0	20	1.54	n/a	5.0	19	3.80	475	1.0	95	95	1439 (572)
試行(2)	50.5	196	3.88	n/a	117.0	115	0.98	122	4.0	452	113	1712 (680)

※実装テーマ比の括弧内の数値は、機種開発Bを100とした時の比率



要件定義: USDM導入で**工数は従来より増加したが、大幅に質が向上**
設計: 開発完了後のマージ工数を含め、**従来と同等以上の成果**
実装: 変更点の明確化で**従来よりも大幅に生産性が向上**

4. 取組みの効果 - 拡がりの兆し

その他の効果として、社内発表をはじめ情報共有を進めた結果、現在は**社内でXDDPの拡がり**が起きてきた段階にある



- MFP製品のネットワーク機能の保守で実践開始
- MFP製品のシステム制御部で試行
- 無線制御ソフト開発で試行
- プリンタ製品のGUI開発で実践
- エンタープライズ系開発で実践

■ アジェンダ

1. 背景

2. 導入

3. 試行と工夫

4. 取組みの効果 (データで分析)

5. まとめ

5. まとめ – 試行により得られたこと

■ 品質・納期・生産性に対するXDDPの効果

試行の結果として、「4. 取組みの効果」で述べたXDDPの有効性を示す定量的なデータを自ら確認できたことは大きい。

また、今回の試行で常にデータを測定しそれに基づくメトリクスを確認しながら進める開発スタイルに変わったことも重要なポイントと考えている。

■ トータル開発工数への影響

工程毎の工数配分が大きく変わり、上流への工数シフトが特にQD面で良い影響を生み出すことがわかった。

一方で、上流工程が狙い通りに行われていないと従来よりも工数が増えてしまう懸念がある。

そのため、今回工夫をしたテンプレートを更に改良しながら、**確実に上流工程で品質を作りこむことが必要**と考えている。

■ 5.まとめ - 今後の課題

■ プロセスの課題

- 社内標準開発プロセスと乖離した部分があるため、社内標準プロセスのテーラリングを行いXDDPを取り込む

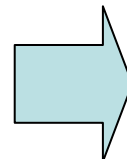
■ 成果物の課題

- 派生開発の規模が大きくなると正式ドキュメントへのマージ作業が増大する、またWord/Excel形式ではドキュメントの変更管理が容易でないため、ドキュメントを差分管理できるシステムを導入し、マージ作業をなくす
- 複数コンポーネントを含む開発では要求仕様TMが肥大化するため、システムレベルとコンポーネントレベルの要求仕様TMに分割する

5. まとめ – 最後に

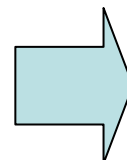
XDDPに取り組んだ結果、ラクになったのだろうか？

やりたかったこと
「不具合出さない」

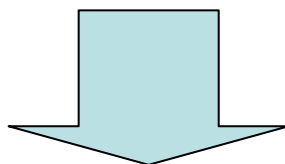


確実に減った

やりたかったこと
「開発生産性Up」



実感がない...



実感がわからないのは、余裕ができて
も火を吹いている別プロジェクト応援に
駆りだされてしまうため。

やはり、チーム全体への定着が
不可欠！ (現在展開中)



ご清聴ありがとうございました

RICOH
imagine. change.