

XDDPとUSDMで課題解決

本多 慶匡 / 的川 建史

東京エレクトロン ソフトウェア・テクノロジーズ(株) / 製品開発統括部

東京エレクトロン東北(株) / ソフト技術部



東京エレクトロン

Index

- XDDPとUSDMの導入
- 問題、障壁の解決
- USDM導入の効果
- Challenge
 - － 改善
 - － コードクローン
 - － Agile、PSP、XDDP
- まとめ

Key Word

- 組込み
- UI
- 保守開発
- 長期間

Motivation

- 不具合の低減
- 開発期間の短縮
 - 生産性の向上。
 - 要件定義が原因の手戻り時間を削減。



導 入



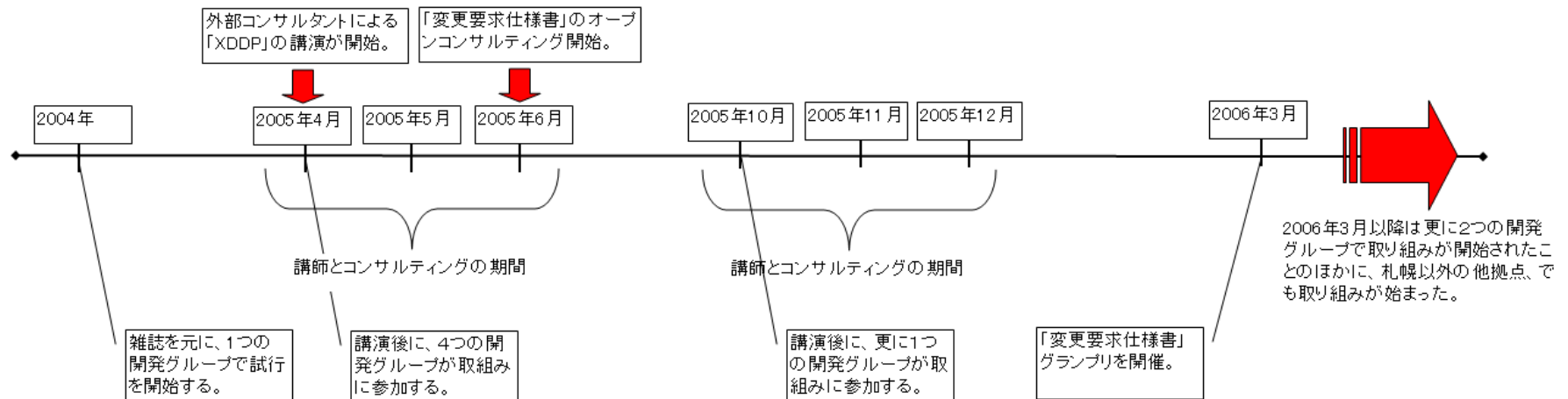
USDM

顧客による座席予約機能			備考欄	
カテゴリ名 (記号)	要求	RSV	操作画面から、指定された貸し会議室を予約する	振る舞い(動き)が分かるように動詞で表現する。
	理由		お客さんが希望する人数に応じた会場を操作端末を介して予約したい。	理由を書くことで、要求の履き違えをなくす。
	説明		端末はすでにネットワークに接続していて、ここではネットとの接続方法については扱わない。複数の端末から同時に同じ会議室を予約する可能性がある。	
階層化された要求	要求	RSV.01	会員番号の入力を求める	
	理由		この操作は、既に会員登録された人にも提供するもの	
	説明			
			<会員番号の入力>	
	<input type="checkbox"/>	RSV.01.1	会員番号の入力を求める	
	<input type="checkbox"/>	RSV.01.2	入力された会員番号が登録されていることを確認する	
	<input type="checkbox"/>	RSV.01.3	会員登録されていない場合は「その番号は登録されていません」を2秒間表示し、元の画面に戻る	
			<開始画面の表示>	
	<input type="checkbox"/>	RSV.01.6	会員であることを確認できれば予約操作が出来る画面を表示する	
	要求	RSV.02	開始画面から会議室を選択することで、予約状況を知ることができる	
理由		予約を急ぐかどうかの判断の材料を提供する		
説明		予約の対象とする会議室は指定のファイルに登録されている。予約状況は随時更新されている。		
<会議室一覧の表示>、<予約状況の表示>と操作と画面の挙動がイメージできるように時系列で表す。			<会議室一覧の表示>	
	<input type="checkbox"/>	RSV.02.1	「一覧」ボタンで会議室一覧画面を表示する	
	<input type="checkbox"/>	RSV.02.2	会議室ファイルを開いて、会議室一覧をファイルに登録されている順に表示する	
			<予約状況の表示>	
	<input type="checkbox"/>	RSV.02.7	会議室番号をキーにして、予約ファイルから予約状況を入手して画面に表示する	
	<input type="checkbox"/>	RSV.02.8	予約状況の表示は、一覧表示されている画面の上から順に表示する	
	<input type="checkbox"/>	RSV.02.9	表示の種別は、 ・予約済み ・空きの2種類	グループ化することで考える範囲が特定されて、仕様の抽出がしやすくなる。
	<input type="checkbox"/>	RSV.02.10	予約状況表示のタイミングは、 1) 一覧画面を最初に表示したとき 2) 「更新」ボタンが押されたとき	

XDDP、USDМの導入

- 要求の仕様化技術への取り組み
 - 講師を招いてセミナーを開催。
 - オープンコンサルティングを開催。
 - 社内で教育者を育成し、教育を継続。

社内での「派生モデル開発」の展開の様子



Education

- 開発チームメンバー全員が受講
- 開発の依頼元、依頼先の両方で受講
- 社内向け講師を育成
 - XDDP、USDMの共通理解。
 - 講師による教育は8時間。
 - OJTで補足。



Writing

- 要求の階層化 ⇒ 上限(3)を設ける。
- 要求に対して展開する仕様数 ⇒ 上限(10)を設ける。
- 仕様のグループ化 ⇒ ⇒ グループ項目の固定化。



Transplant

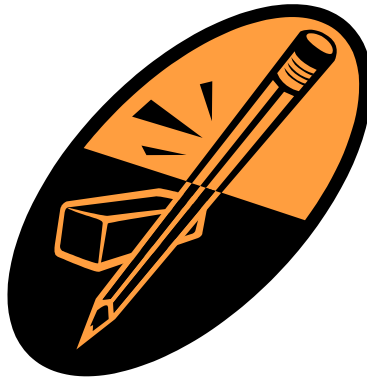
● 新規と変更?

- こたえは“移植”。



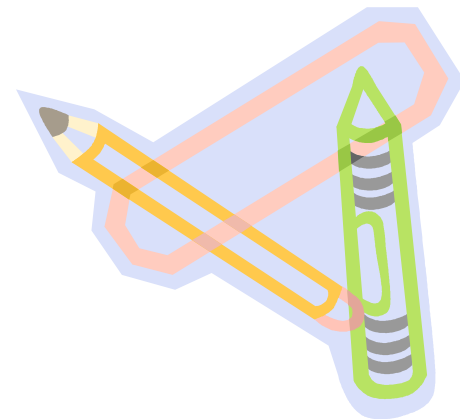
Maintenance

- 保守の為の文書は別に用意する。
 - コードの変更が伴わないことは記述しない。



Beautiful

- Color ダーク系よりもライト系。
- Sentence 長文にしない。
- Chart 文字より図。(使いこなす)



Color

カテゴリ名 (記号)	要求	(要求番号)	ここに要求を記述します。
		理由	要求の背景や理由について記述します。
		説明	要求について必要に応じて説明してください。仕様とは見做されません。セルを広げて、図を貼っても良いです。
	<<仕様分類名>>		主分割記号・全体を通して共通の分割基準として決める。(オプション) <<前提条件>><<内部処理>>
		<仕様分類名>	補助分割名・主分割の中に異なるテーマの仕様が混在する場合に補助分割記号を使って、純度の高い集合を作る
	<input type="checkbox"/>	(仕様番号)	上記の要求に含まれるべく要求仕様を記述します。必要に応じて“< >”でグループ分けしてください
	<input type="checkbox"/>	(仕様番号)	
	<input type="checkbox"/>	(仕様番号)	
		<仕様分類名>	
	<input type="checkbox"/>	(仕様番号)	
	<input type="checkbox"/>	(仕様番号)	

カテゴリ名 (記号)	要求	(要求番号)	ここに要求を記述します。
		理由	要求の背景や理由について記述します。
		説明	要求について必要に応じて説明してください。仕様とは見做されません。セルを広げて、図を貼っても良いです。
	<<仕様分類名>>		主分割記号・全体を通して共通の分割基準として決める。(オプション) <<前提条件>><<内部処理>>
		<仕様分類名>	補助分割名・主分割の中に異なるテーマの仕様が混在する場合に補助分割記号を使って、純度の高い集合を作る
	<input type="checkbox"/>	(仕様番号)	上記の要求に含まれるべく要求仕様を記述します。必要に応じて“< >”でグループ分けしてください
	<input type="checkbox"/>	(仕様番号)	
	<input type="checkbox"/>	(仕様番号)	
		<仕様分類名>	
	<input type="checkbox"/>	(仕様番号)	
	<input type="checkbox"/>	(仕様番号)	

Performance

● Excelマクロの活用

- “要求仕様書” から“テスト仕様書” を作成するマクロの活用。
- “狙い” は、

ハイレベルな効率と品質の両立

- ・ テスト仕様書・チェックリスト作成時間の短縮
- ・ テスト項目の作成漏れ防止

導入した結果 . . .

効率 3倍！

品質 2倍！



Performance

● Excelマクロの活用

－ 副次的な効果として、

- ・「要求仕様の書き方」が安定した。



仕様漏れ、記述ミスが減少した。

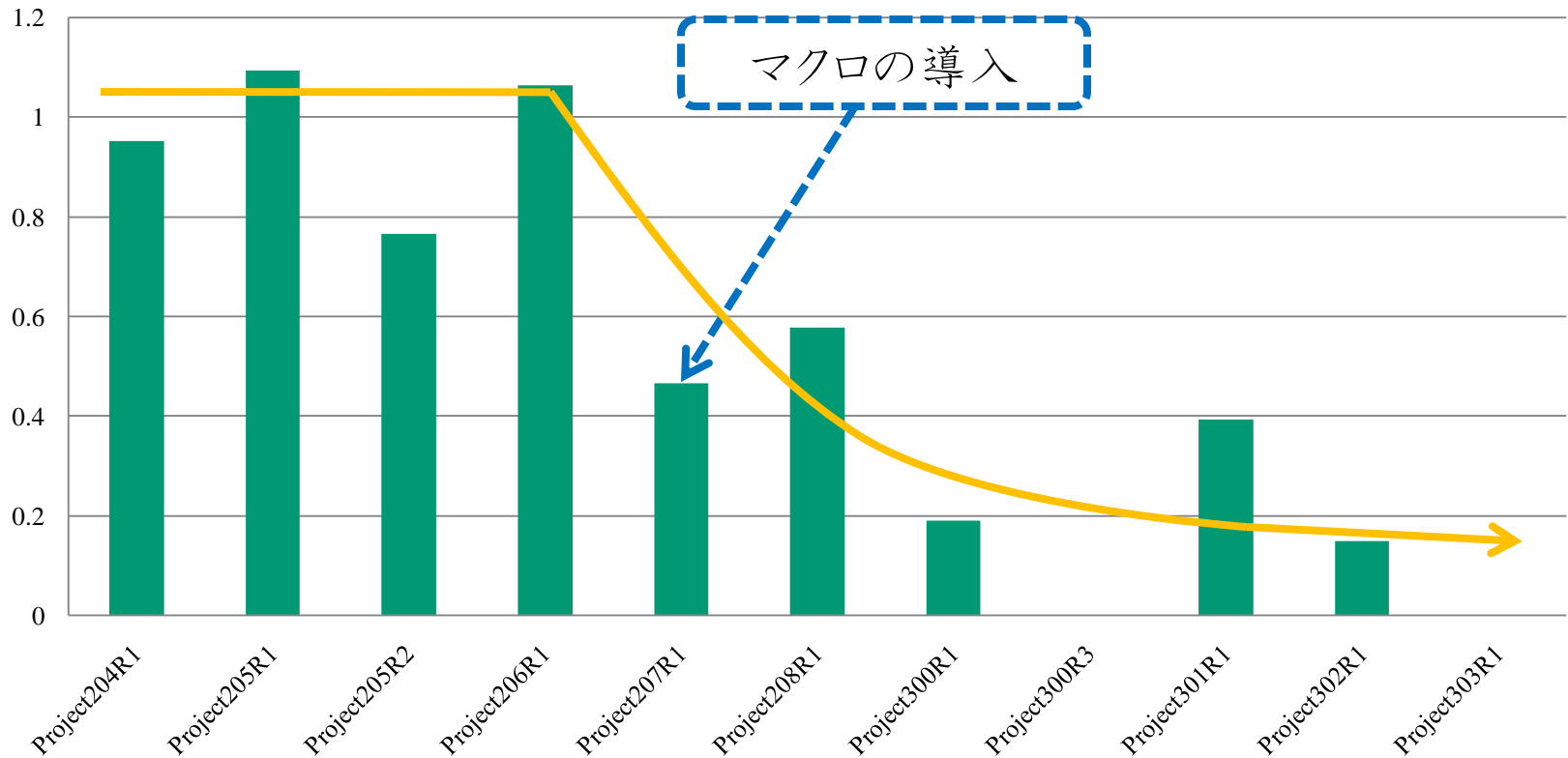
(次ページ参照)

テスト時の注意点や
要求仕様の補足を
記述する

	<履歴機能>	
	<データ保持状態表示機能>	
<input type="checkbox"/>	401-1	現在のデータ保持状
<input type="checkbox"/>	401-5	現在のデータ保持状
*	説明	現状の[Server Data :
*	説明	Newest Date)は表示し また、トレースファイル

OutPut

仕様漏れ・ミス数低減
(100仕様あたり)

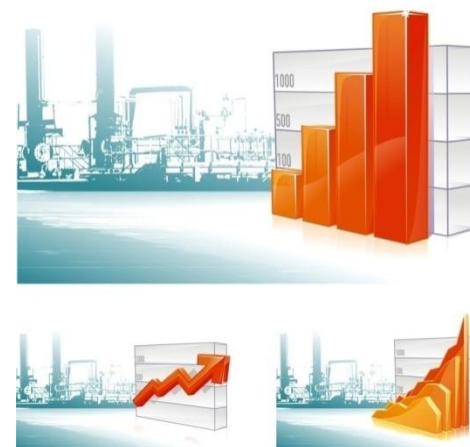


OutPut

- 仕様が原因の不具合が減少
- 開発期間の短縮
 - 要求元と開発者間で仕様合意までの時間が短縮。



XDDP導入後の“要件定義”の割合変化



見える化



ソースコードの生産性を比較

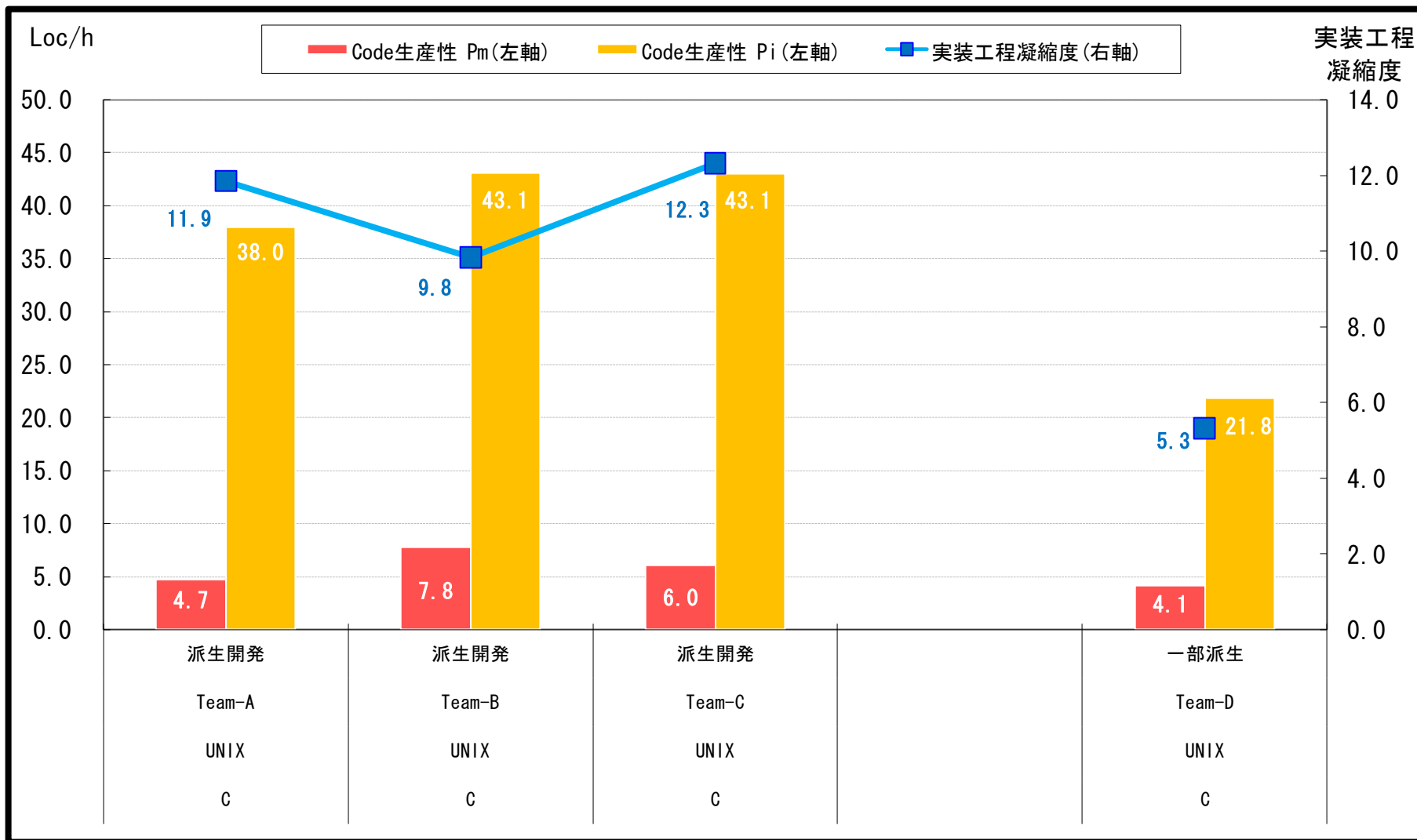
- USDMを採用している開発チームと採用していない開発チームでソースコードの生産性を比較



ソースコードの生産性

- 2種類のソースコードの生産性データを活用
 - 全期間に於ける生産性
 - ・ $P_m = \text{生成行数} / \text{プロジェクト期間の全時間数}$
 - 実装工程に於ける生産性
 - ・ $P_i = \text{生成行数} / \text{実装工程の全時間数}$

ソースコード生産性の比較



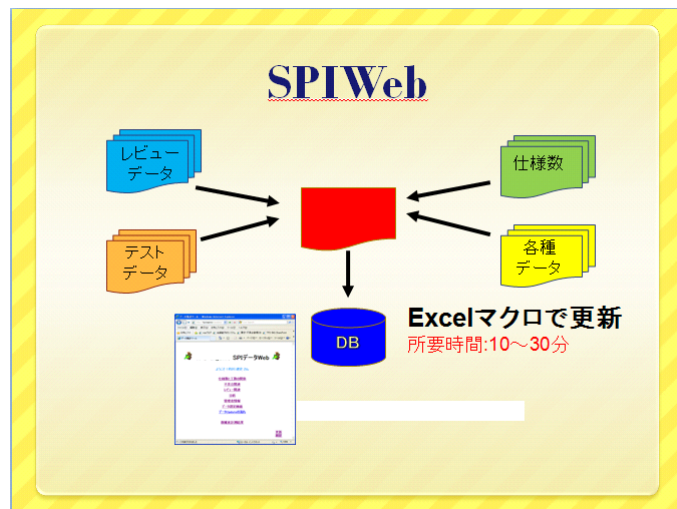
改善



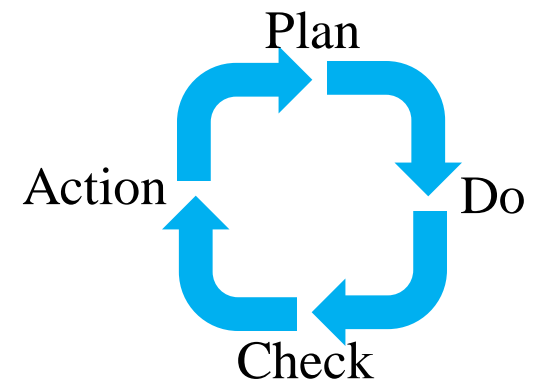
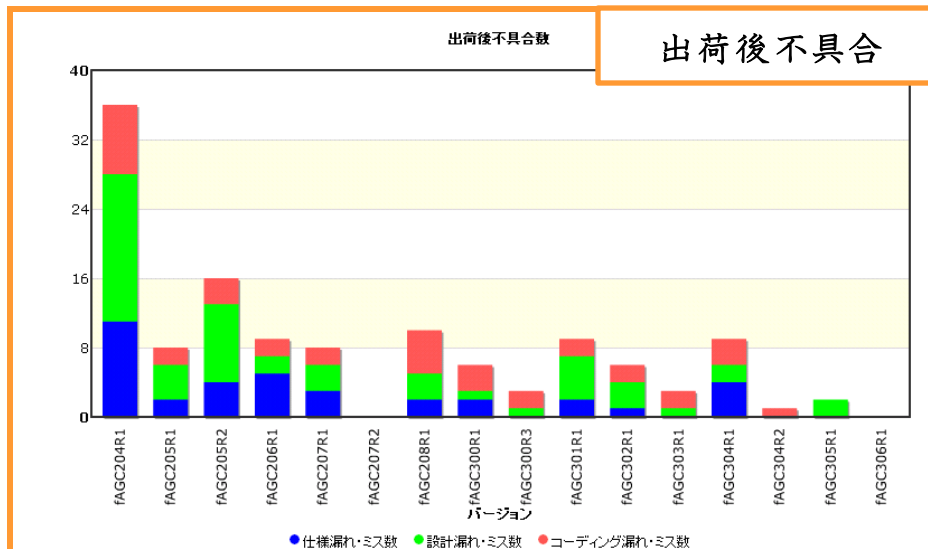
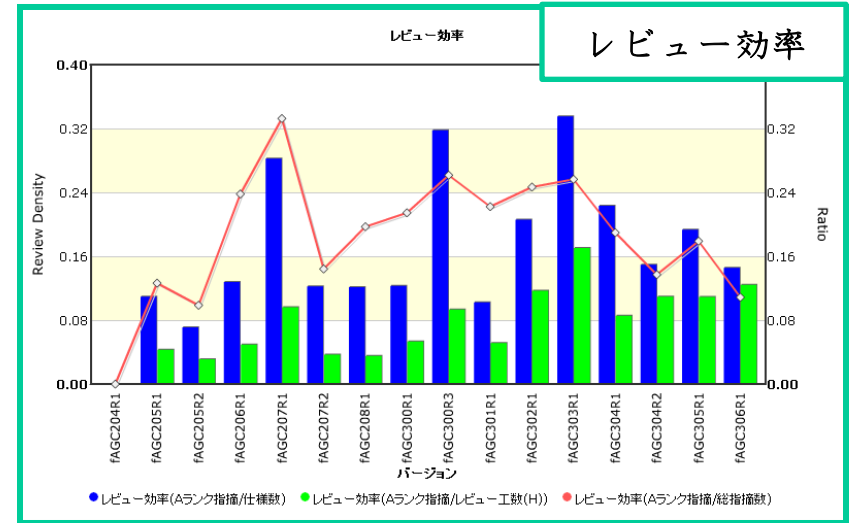
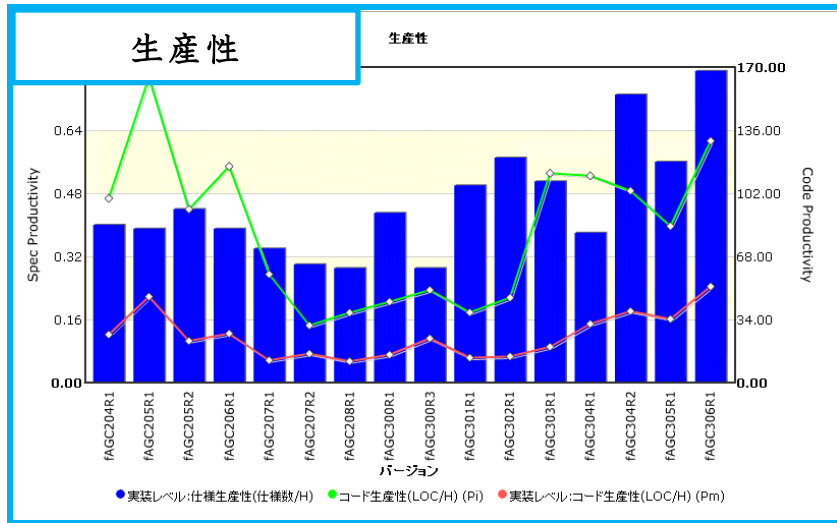
開発データを利用してプロセス改善

- 仕様数で開発規模を語る
- データをWebで誰でもいつでも参照できる

➡ 個人レベル・チームレベルで
改善のモチベーション向上



開発データを利用してプロセス改善



次の取組み

知見とツールの活用

- コードクローン解析ツールの導入
 - 影響範囲の特定を支援するツールとして期待。
 - 知見者に依存する割合が小さくなることを期待。

 **AIST CCFinderX** + **USDM**

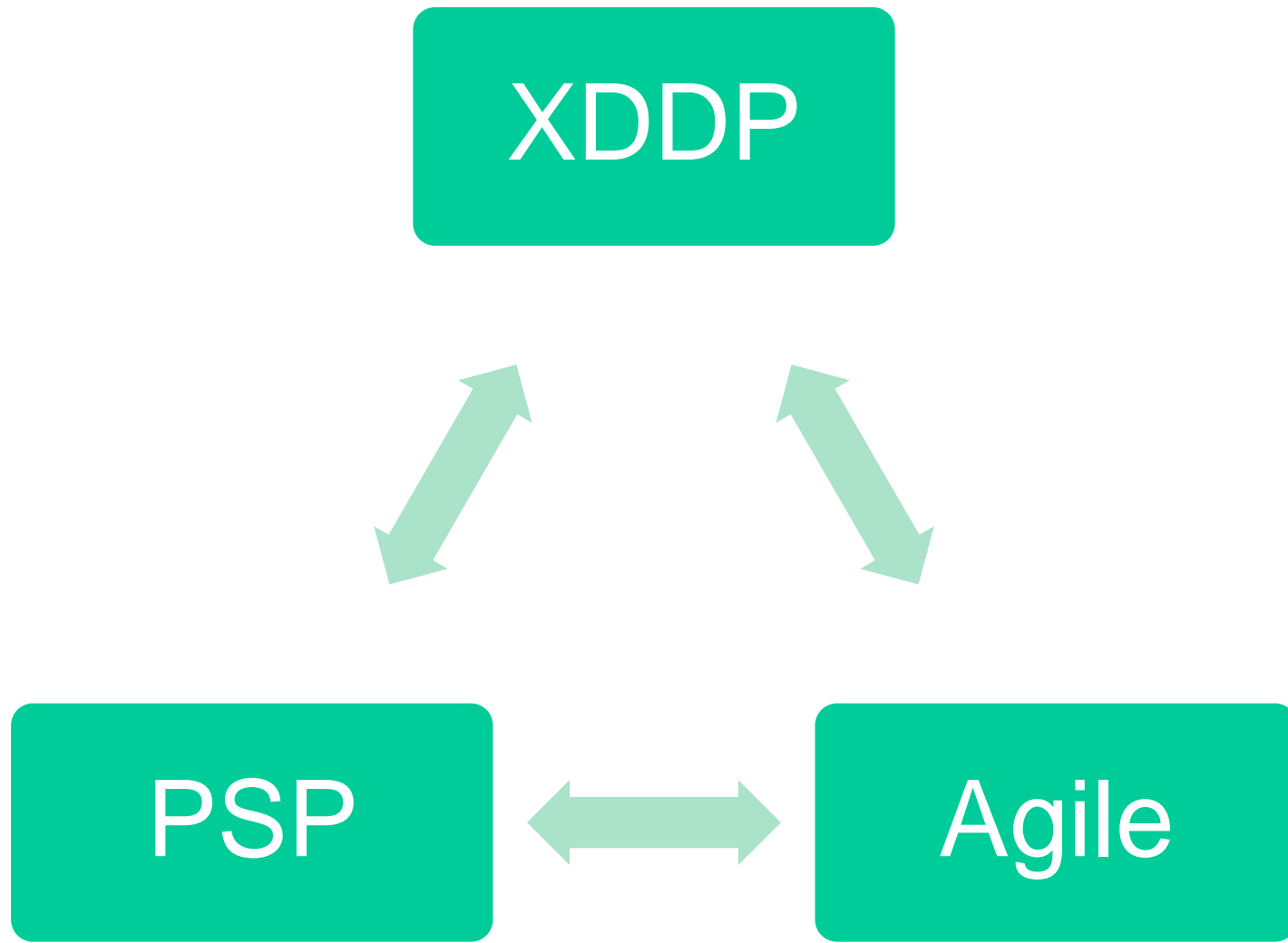
Agile



ソフト開発対象の変化

- 多人数でレガシーコードを保守する開発から、少人数で新規ソフトを開発するように変化した。

A Discipline for Software Engineering



XDDPとPSP

- PSPをまなぶとPFDの理解が深まる。
 - プロセス設計
 - プロセス定義
 - チャンク化



XDDPとAgile

- XDDPを経験した開発者は、Agileを正しく理解する。
 - XDDPの要素はAgileの中でも機能している。
 - XDDPをAgileでも続ける。
 - 1イテレーションの中でXDDPを実施。



一口サイズ見積りとチケット駆動開発

- 全体計画を2週間単位の計画に分解する。
- 1イテレーションをタスクに分割。
- 一つのタスクは1日(6h)以内。
- 15分単位で追跡する。
- XDDPの要素“一口サイズ見積り”、“3段見積り”が活きている。



プロセス設計

- 朝のスタンドアップミーティングでタスクの移動、タスクの実施方法を報告。
 - タスクとタスクの繋ぎ(チャンク)を検討し、チームメンバーに伝える。
- 毎朝がプロセス設計の場。

XDDP or Agile

- これらは別のものではない。
- Agile開発の中でも『USDM』は使われている。
- イテレーション毎に『サイズ見積り』。
- イテレーション毎に『プロセス設計』。
- 1イテレーションの中でXDDPが実施されている。
- XDDPで成功した開発者はAgileでも成功を掴む。

まとめ



社内に根付いたXDDP

- 社内に根付いたものは、
 - USDMで書かれた要求仕様書。
 - ・ USDMは新規開発でも活用されている。
 - ・ 新規でも記述様式が変わらないので、依頼元が理解しやすい。
 - ・ 一番の効果は、依頼元と開発者間で正しく理解されること。
 - サイズ見積もり。
 - プロセスと成果物の連鎖。
 - 改善活動。

XDDP + Challengeで“成長”と“変化”を続ける。



おわり

