

チームで極める派生開発

～ 3年半のXDDPの実践から見えてきたこと・感じたこと ～

ソニーイーエムシーエス株式会社 勝又 淳

Agenda

- **プロセス改善の経緯と歩み**
- **XDDPの導入効果**
- **XDDPとSCRUMの併用実例**
- **まとめ／考察／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) 湖西サイト

設計第2部門 設計1部 ※2012.05.01 時点

職務: 業務用放送機器ソフトウェア設計リーダー

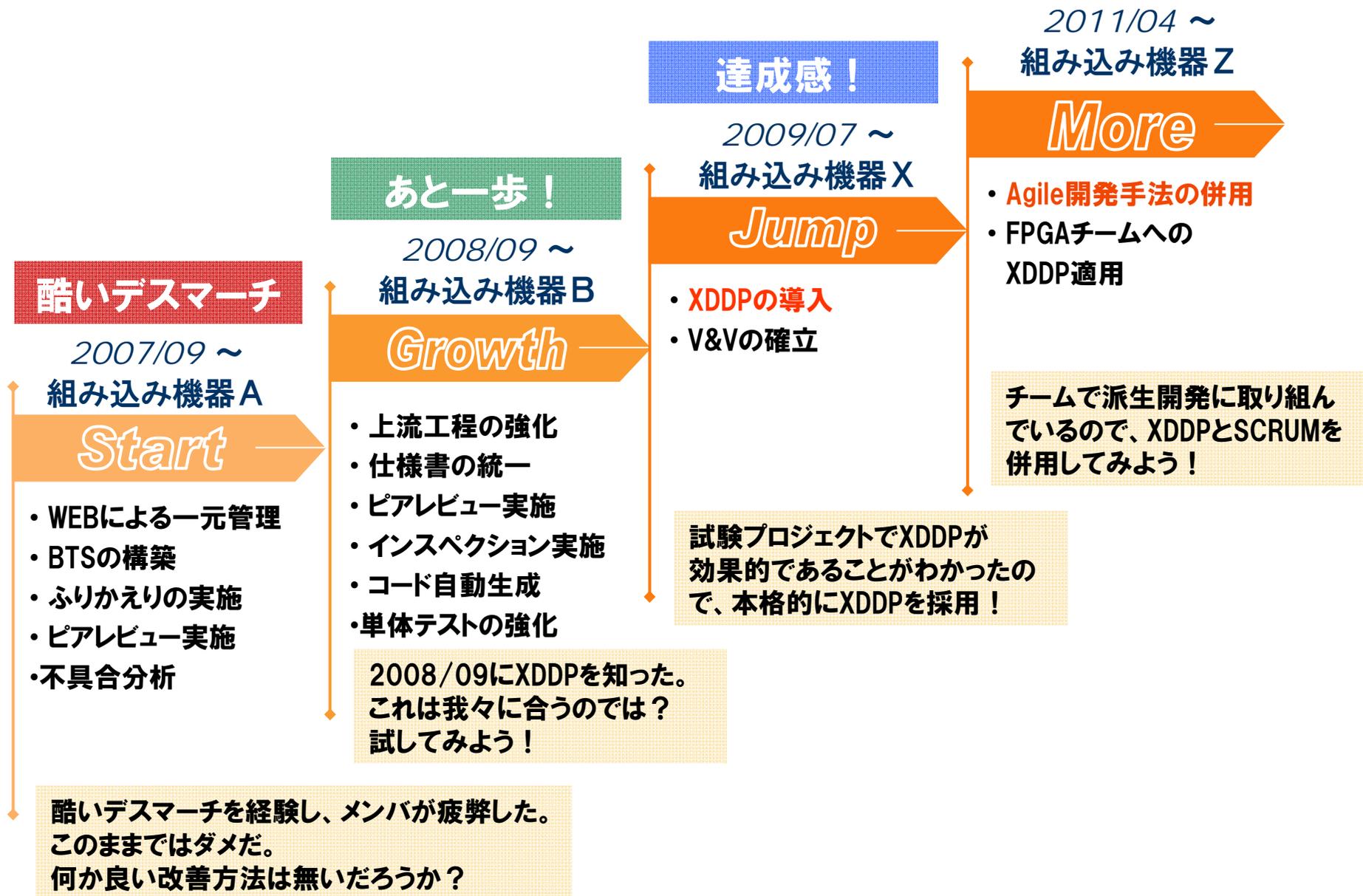


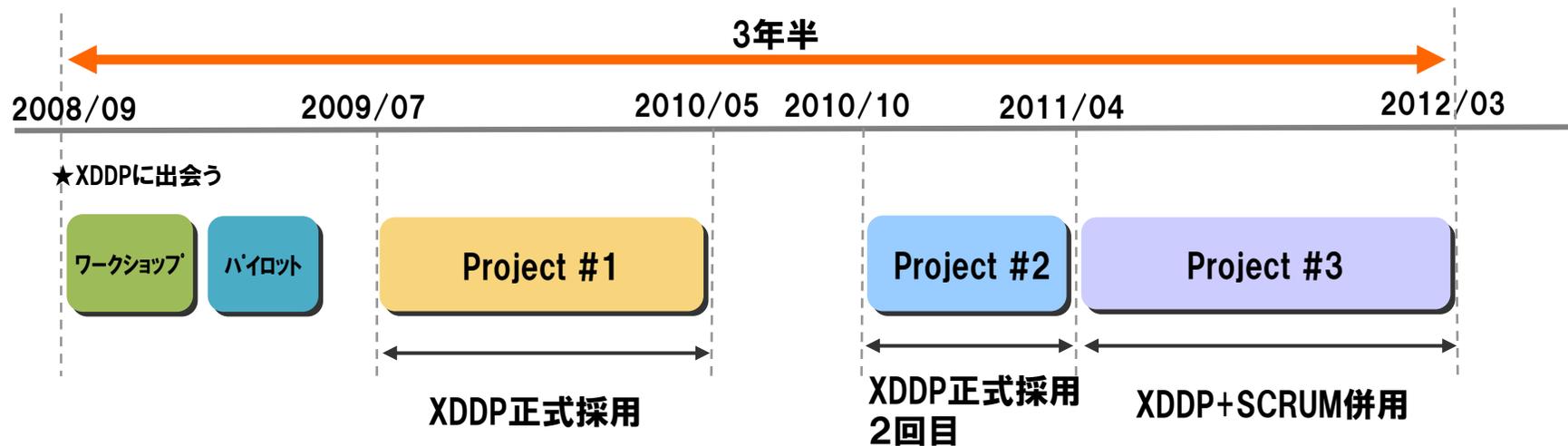
事業部で設計される機種の**派生モデル**の設計が多い



派生開発を極める事が有益である







プロジェクト	期間	ソフト人員	母体規模 (KLOC)	変更量 (KLOC)	要求数 (件数)	特徴
パイロットPrj	3ヶ月	6名	約650K	約2K	15	試験的に導入
Project #1	10ヶ月	6名	約650K	約15K	122	XDDP本格採用
Project #2	5ヶ月	4名	約650K	約15K	62	FPGAにも適用
Project #3	12ヶ月	7名	約500K	約50K	70	XDDPとSCRUMの併用

- ・いずれも当方が担当したプロジェクトにて適用
- ・3年半で3回のプロジェクトにXDDPを適用した
- ・固定メンバではない(毎回メンバが異なる)
- ・全て組み込みソフトウェアの派生開発が対象
- ・Project #3では、全く異なるドメインでの適用

1. XDDPに興味があった

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。

トップダウン型でXDDPを導入したのではなく、現場主導の改善案の一つとしてXDDPを導入した。

- ・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の導入効果 (Project #1)

7/19

案件	業務用組み込み機器の派生モデル開発 (Project #1)
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)

	項目	XDDP導入前	XDDP導入後	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

2つのプロジェクトでXDDPを実践して、QDC(特に品質面)において一定の改善効果を実感することができた。ここで一旦、ここまでの課題を抽出する。

課題①:H/Wチームへの検証用プロトタイプを提供

組み込み機器の開発なので、どうしてもH/Wとの兼ね合いが出てくる。本来は、H/Wチーム内で検証用ソフトを用意するべきだが、リソース的にそれを実現するのは難しい。XDDPとは別プロセスで検証用のプロトタイプを提供しているが、思いのほか工数が掛かるので何とかしたい。

→H/Wのプロセスとムダなく合わせるには、どうしたらよいのか？

課題②:テストチームのプロジェクト後半の負荷が高い

テストチームにも変更要求仕様書レビューから参加してもらっているが、それでもプロジェクトの後半に負荷がかかる。

課題③:もっとチーム全体の生産性向上を図りたい

メンバ全員でXDDPに取り組んでいるので、個の生産性は上がってきているが、チーム全体での生産性は改善の余地があるのではないだろうか。

チームの生産性を疎外する要因を取り除き、改善を繰り返していけば、更なる生産性向上が図れるのではないか。

→チーム全体の生産性向上を図るには、どうしたらよいのか？

課題を解決する為に、Project #3では、XDDPとアジャイル開発手法の一つである**SCRUM**を併用することにした。SCRUMはプロジェクト管理要素が強いので、XDDPとの親和性は高いと思われる。

案件	業務用組み込み機器の派生モデル開発 (Project #3)
期間	2011/04 ~ 2012/03(約1年)
人数	約15名(ソフトウェア7名, FPGA2名, テスト2名)
要求数	約70件
母体行数	約500KLOC
スプリント回数	24回 (TimeBox=2週間)
特徴	<ul style="list-style-type: none">・コアLSI を変更、H/W ブロックの大幅変更・初めてXDDP/SCRUM を実践するメンバーが多い・組織変更により、今までと異なるドメインの派生開発 条件の厳しいプロジェクトだが、このプロジェクトで成功することができれば、チームで行う派生開発の成功パターンの1つであることが証明できる。
プロジェクト成功判定	<ul style="list-style-type: none">・外部不具合流出件数 (Quality観点)・納期遅延日数 (Delivery観点)・予算超過無し (Cost観点)・メンバーの充実感(プロジェクト終了後のアンケート)

[SCRUMの特徴]

- アジャイル開発手法の一つ
- プロジェクトマネジメントを重視
- チーム全体の生産性向上を目指す

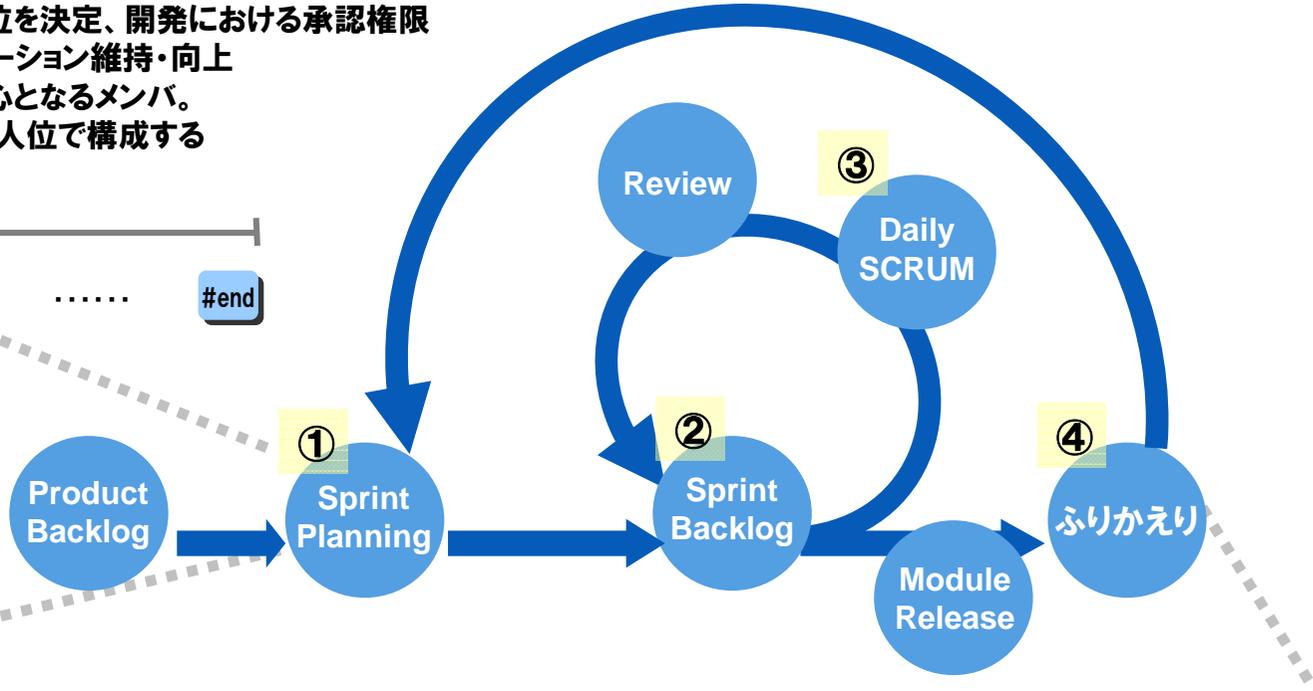
プロダクトオーナー: 開発の優先順位を決定、開発における承認権限
 スクラムマスタ : メンバのモチベーション維持・向上
 スクラムチーム : 開発作業の中心となるメンバ。
 一般的に5~9人位で構成する

スクラムに欠かせない重要なポイントは、
 チームメンバが自律的・主体的に動かなければならないこと。
 各自が役割に責任をもって自分たちで進め方を考え、実施していく。
 最初はスプリントの目標に基づき、その後は毎日のデイリースクラムを通じて、
 日次ベースで自ら組織的な調整を行う。

TimeBox = 2週間 とした場合



TimeBox: 固定日数
 通常2週間~6週間



DAY 1	DAY 2	DAY 3	DAY 4	DAY 5	DAY 6	DAY 7	DAY 8	DAY 9	DAY 10
Daily SCRUM									
スプリント計画				中間レビュー				リリースレビュー	ふりかえり

※必要に応じて

※最終日に行うこともある

デイリースクラムの実例

昼休み後、ホワイトボードの前でスタンドアップ形式で行う。
最大でも15分程度で終了する。

<デイリースクラムの様子>



全員が以下の3点についてのみ話す。

- ▶ 昨日から今日までにやったこと
- ▶ 今日これからやること
- ▶ 問題点

! <デイリースクラムを行って感じたメリット>

- ・毎日決まった時間にミーティングを行うので、リズムが体に刻まれ、自然と定刻に同じ場所に集合するようになる。個々のリズムがチームのリズムと調和してくる。
- ・人に話すことで頭が整理される。また、無駄な作業のチェックにもなる。
- ・この場では、1日単位のことだけを考えるので、全力で目の前のタスクに取り組むことができる。
- ・問題が発生したら(発生しそうになったら)、直ぐに別途解決の場を設けることができる。早期に問題発生を予防できる。(最速の改善)

デイリースクラムによって、自分達の立ち位置を共有し、安心して1日のタスクに取り組む事ができる。
→ チームの生産性向上に繋がる。

ふりかえりの実例

各スプリントの終了時に、KPT形式でふりかえりを行う。

- ▶ **Keep:**
今回のスプリントで良かったこと
次回のスプリントでも続けたいこと
- ▶ **Problem:**
今回のスプリントで良くなかったこと
次回のスプリントでは改善したいこと
- ▶ **Try:**
Problemの具体的な解決策
新たに挑戦したいこと

<ふりかえりの様子>

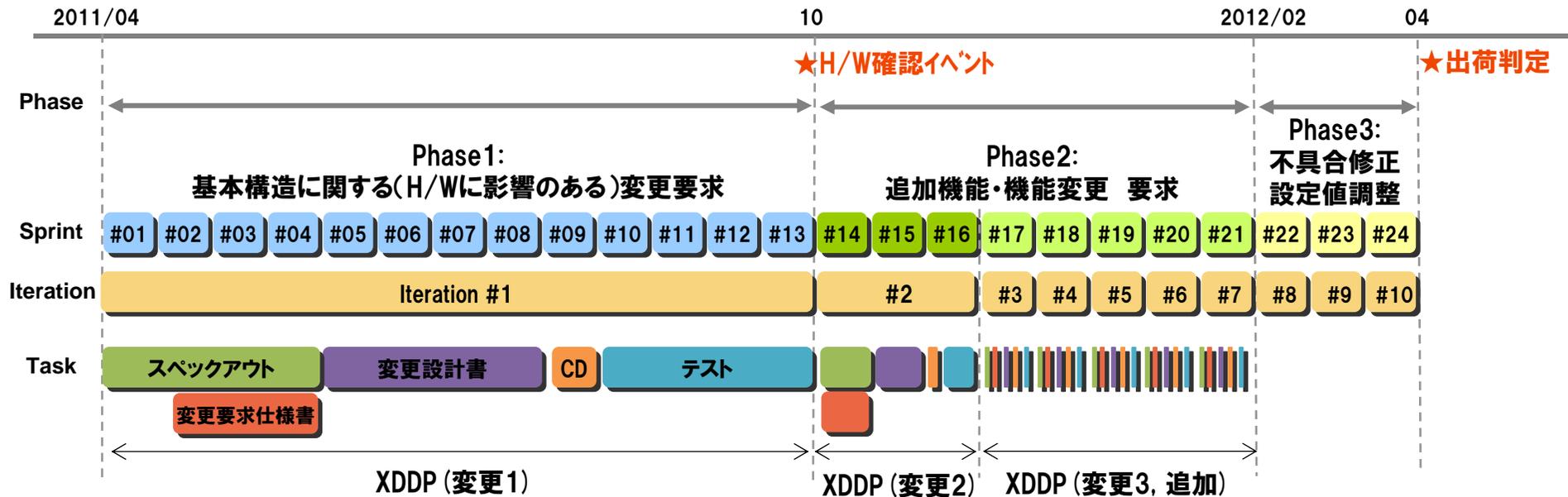


<ふりかえりを行って感じたメリット>

- ・スプリント毎にふりかえりを行うことで、常に改善意識を保つことができる
- ・付箋紙に書くことで率直な意見が沢山出てくる
- ・チームにリズムが出てくる。チームの一体感を感じる。

ふりかえりによって、継続的な改善意識を保つ事ができる。次のスプリントも頑張ろうという気になる。

→ チームの生産性向上に繋がる。



基本構造 (アーキテクチャ) やハードウェアに影響のある要求を抜粋し、XDDPのプロセスを適用して前半に品質を造りこむ。

10月のH/W確認イベントでの品質が非常に良かった。
H/Wが早期に安定したことで、後半の追加機能に集中することができた。
また、プロトタイプを作成するムダも省けた。

テストチームも早期に稼動することができた。

モジュールが無い段階からドキュメントを成果物としてSCRUMを適用した。
早期からSprintのリズムを共有することで、チーム内のモチベーションが高まった。
デイリースクラムやふりかえりにより継続的改善意識を最後まで保つ事ができた。



XDDPを採用したプロジェクトの規模とQDC結果

プロジェクト	期間	人員	母体規模 (KLOC)	変更量 (KLOC)	要求数 (件数)	生産性 (LOC/h)	品質※ (指摘数)	納期 (遅延日)
パイロット	3ヶ月	6名	約650K	約2K	15	91.2	0	0
Project #1	10ヶ月	6名	約650K	約15K	122	123.2	1	0
Project #2	5ヶ月	4名	約650K	約15K	62	118.5	0	0
Project #3	12ヶ月	7名	約500K	約50K	70	150.3	0	0

※ここでの生産性は、1時間あたりのコード行数を示す。
※品質は、正式版以降のQA指摘不具合件数を示す。
※Project#1では、軽微な不具合指摘が1件あった。
※Project#3では、大量のデータ変更/コード削除が含まれているので、生産性が若干高めの数値となっている。

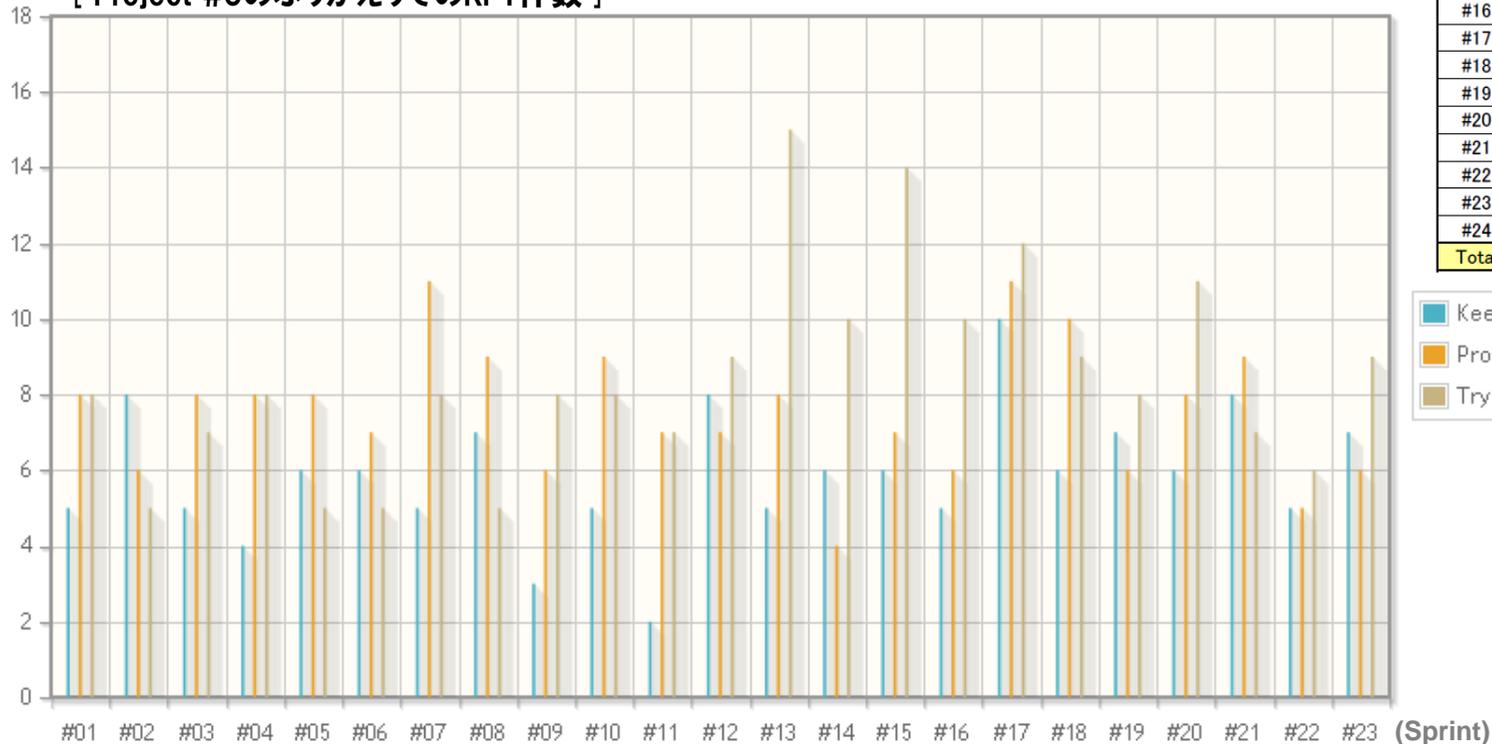
XDDPを適用した全てのプロジェクトにおいて、QDC観点で成功プロジェクトであったと言える。

SCRUM適用効果 (Project #3)

SCRUMの適用効果を定量的に表すのは難しいので、「チームの成熟度up」を「チームの生産性up」とみなす。後半にTryの数が増えていることは、チームのモチベーションが高く改善意識が継続していることを意味する。

Sprint	Keep	Problem	Try	Total
#01	5	8	8	21
#02	8	6	5	19
#03	5	8	7	20
#04	4	8	8	20
#05	6	8	5	19
#06	6	7	5	18
#07	5	11	8	24
#08	7	9	5	21
#09	3	6	8	17
#10	5	9	8	22
#11	2	7	7	16
#12	8	7	9	24
#13	5	8	15	28
#14	6	4	10	20
#15	6	7	14	27
#16	5	6	10	21
#17	10	11	12	33
#18	6	10	9	25
#19	7	6	8	21
#20	6	8	11	25
#21	8	9	7	24
#22	5	5	6	16
#23	7	6	9	22
#24	40	43	31	114
Total	175	217	225	617

(件数) [Project #3のふりかえりでのKPT件数]



※sprint #24のふりかえりでは、全体のふりかえりをおこなったので、グラフから外しています

プロジェクト終了時にチームメンバーに対して以下のアンケートを実施した。

設問	YES	NO
Q1. XDDPの適用効果はあったと思いますか？	6	0
Q2. 次の派生開発プロジェクトでもXDDPを適用した方が良いと思いますか？	6	0
Q3. SCRUMの適用効果はあったと思いますか？	6	0
Q4. 次のプロジェクトでもSCRUMを適用した方が良いと思いますか？	6	0
Q5. 今回のプロジェクトは成功だと思いますか？	6	0
Q6. 今回のプロジェクトを終えて充実感がありますか？	4	2

<コメント抜粋>

- A1. 大きなシステムの場合、部分的な把握で変更しなければならず、理解したつもりで、すぐに実装を開始すると、必ず思い違いがあって、後戻りが発生するが、XDDPの採用により、後戻りが少なかったと思います。
- A2. 我々なりにテラーリングする必要がある気もしています。
個人的には、スペックアウトと変更要求仕様書についてなんかやれることがある気がしています。
- A3. 2週間毎のスプリントで区切ること、デイリースクラム等リズムができる(慣れてから実感する)。
設計内テストチームが早い段階でテストできることで、不具合を多く検出できる
- A4. 気づきの場になっていると思うので有効だと思います。
また、スクラムでの情報から、チームとしての早期対応策を練ることが出来るのではないのでしょうか。
- A5. スケジュールや流出バグ数の結果からみて概ね成功だったと思います。
ただし失敗だった部分や課題はたくさんあったと思います
- A6. これまでXDDPのみの時は(担当部分が明確に分かれていたせいもあるが)「個人での設計」という色が強かったが、SCRUMも適用することでよりチームを感じたから。
- A6. 数値的な指標では合格点かもしれないが、まだまだ改善すべき点が多いため、充実感はありません。

アンケート結果からも、今回のXDDPとSCRUMの併用は、メンバーにとっても非常に有意義な取り組みであったことが伺える。

1. XDDPとSCRUMの併用効果

派生開発において、派生開発に特化した手法（XDDP）を適用することは、品質面、納期面、開発コスト面において効果がある。

また、アジャイル開発手法の一つであるSCRUMと併用することで、チーム内に一定のリズムが生まれ、改善意識を最後まで保つことができる。

個人の生産性向上施策：XDDP

チームの生産性向上施策：SCRUM

今回実践したXDDPとSCRUMの併用は、現状の私達にとって最良の方法である。

2. 開発プロセスを設計する

全てのソフトウェア開発において、XDDPとSCRUMの併用が有効であるとは限らない。規模・リソース・納期等の条件に応じて、最適な開発プロセス・開発手法を自分達で選択／構築していく事が重要であると考える。

<注意点>

※プロセスの併用は、両者の良い所を打ち消してしまう危険性も含んでいるので、十分な注意が必要である。

※XDDPを数回重ねて安定して結果を出せるようになった後に、他のプロセスとの併用を考えた方が良いと思われる。

理に適った手法を採用し、チームメンバが主体となって改善を繰り返しながら開発に取り組むことで、プロジェクトを成功に近づけることができると考える。

チームで派生開発を行う際に
QDC目標同時達成するという課題に対して、
自分達にとって1つの解を見出すことができたが、
まだまだ極めたとは言えない。

いつか**派生開発を極めた**と言える日がくるまで、
今後も継続的改善を行っていきたい。

XDDPという素晴らしい手法を公開して頂いた清水さんに
改めてお礼申し上げます。

ご清聴ありがとうございました