

派生開発における継続的改善

～ XDDPとAgile開発手法によるハイブリッドプロセスの考案 ～

ソニーイーエムシーエス(株) 東海テック 湖西サイト
設計技術1部 設計1課 勝又 淳

Agenda

- **XDDP導入の効果**
- **開発プロセス改善の経緯と展開**
- **XDDPとAgile開発手法との併用実例**
- **まとめ／考察／今後の課題**

自己紹介

1/17

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) 東海テック湖西サイト
設計技術1部 設計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

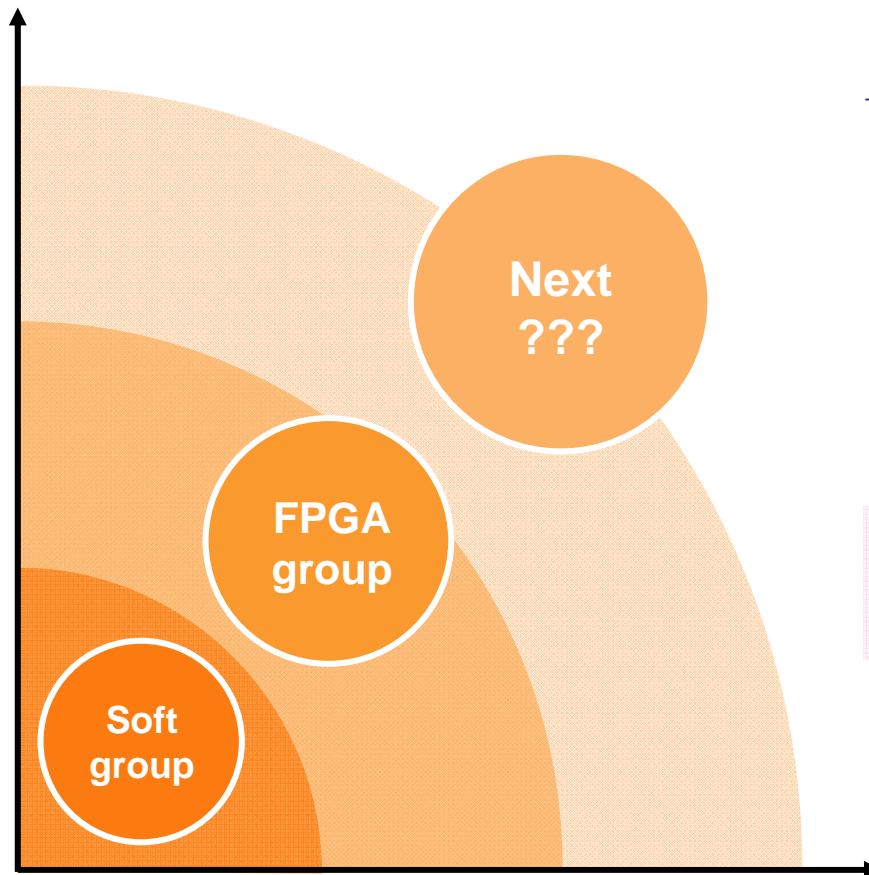


- 業務用組み込み機器の派生モデル設計 -

派生開発カンファレンス 2010で
発表させて頂きました。

[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

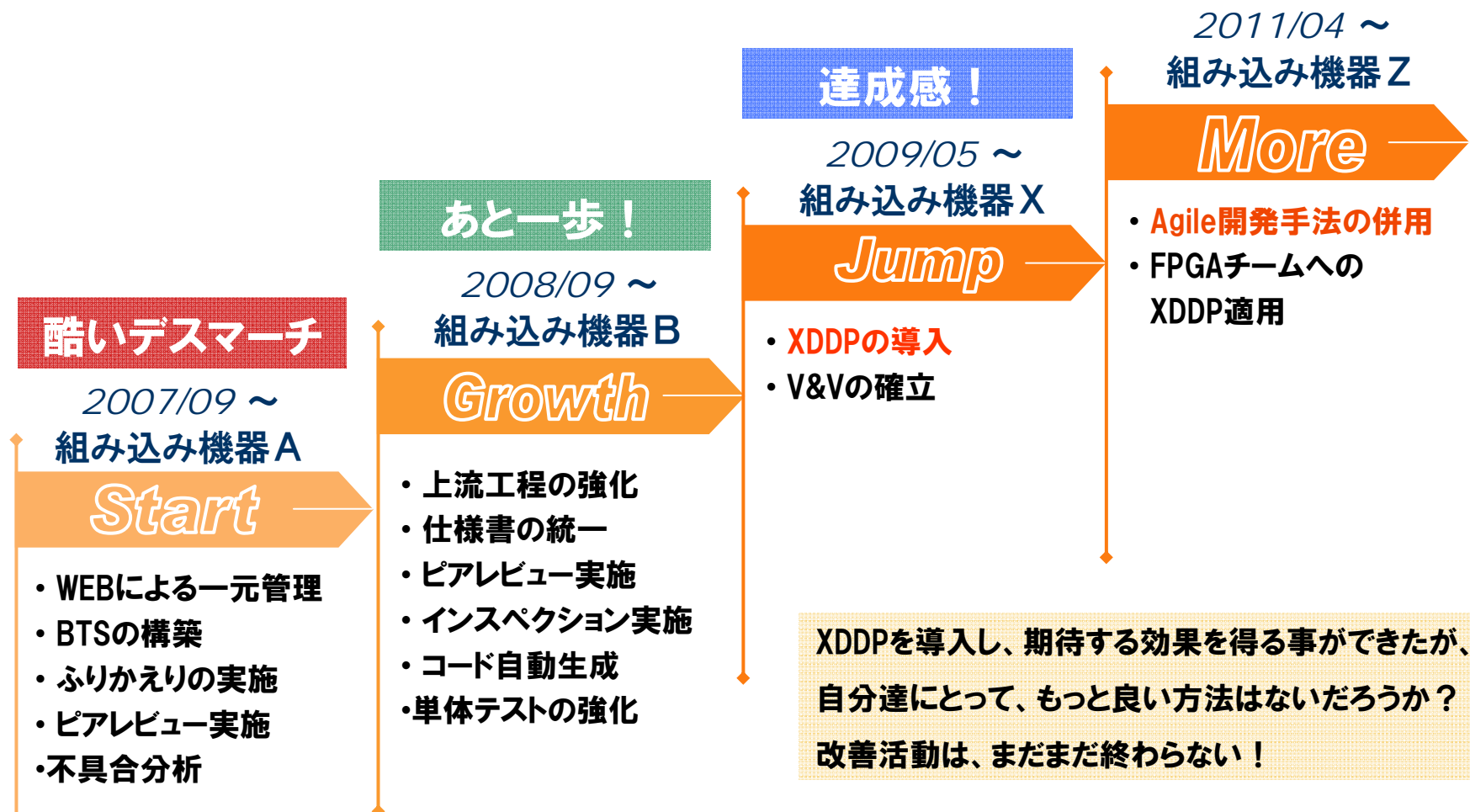


XDDPの展開

設計プロセスにおいて、XDDPの適用範囲を徐々に広げてきている。

- ▶ ソフトウェア設計への適用
- ▶ FPGA/PLD設計への適用

次の施策は？
基板設計への適用？ メカ設計への適用？
別のソフトGへの横展開？



※BTS:Bug Tracking System (不具合管理)

XDDPを導入し、期待する効果を得る事ができたが、自分達にとって、もっと良い方法はないだろうか？改善活動は、まだまだ終わらない！

課題①: 全体の設計工数の削減ができなかった

期待通りの設計工数の削減が実現できなかったのは、自分たちのXDDPの習熟度がまだまだ足りないからだと思うが、それとは別の視点からも改善策を検討したい。

課題②: H/Wチームへの検証用プロトタイプを提供

組み込みデバイスの開発なので、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。

課題③: 設計内テストチームのプロジェクト前半の稼働率が低い

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(動くモジュールのリリース後)に負荷がかかる。

課題④: 突発的な要求に迅速に応えたい

一部の機能だけ、どうしてもユーザーに見せたいというリクエストが数回あった。プロトタイプの作成等で対処したが、XDDPのルールからは逸脱していると思われる。

これらの課題を解決する為に、Agile開発手法を取り入れてみてはどうだろうか。

XDDPとAgile開発手法とのハイブリッドプロセスを構築してみよう！

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)
Agile開発手法を併用した事で品質を落としてしまっては本末転倒である。
品質を落とさないのが絶対条件である。

納期 (Delivery)

- 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: 100~120 LOC/H
ソフトウェア生産性は現状をキープすること。

設計工数削減 (Cost)

- 無駄な後戻り工数を減らすことや早期のテストチームの着手による
設計工数全体の15%削減

[施策1] パイロットプロジェクトでのテスト導入

いきなりXDDPとAgile開発手法との併用を行うと混乱するので、まずは規模の小さいプロジェクトでテスト運用する。

[施策2] XDDPのルールを逸脱していなかチェックを行う

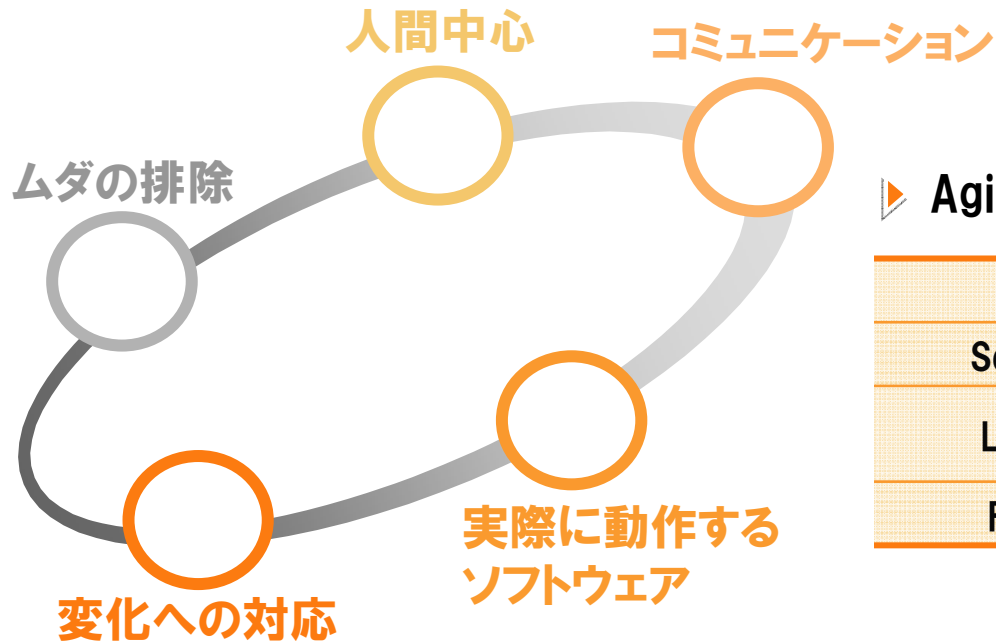
XDDPとAgile開発手法を併用することでXDDPのルールを崩してしまつては、併用する意味が無い。第3者によるチェックを受ける。

[施策3] Agileの共通認識

Agileという言葉が人によって捉え方が異なることが多い。

Agileの共通認識を図る為に、Agileに関する教育・意識合わせを実施する。

▶ Agile開発手法で重要と考える項目



▶ Agile開発手法例

XP	技術的なプラクティスを重視した手法
Scrum	プロジェクト管理に重点をおいたプロセス
Lean	開発プロセスのムダを取り除くことを主眼としたプロセス
FDD	feature (機能価値) から駆動して開発を行う



他の手法との親和性が高いと言われているScrumを採用することを決めた。
XDDPとScrumのハイブリッドプロセスを構築する。

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

スクラムマスター



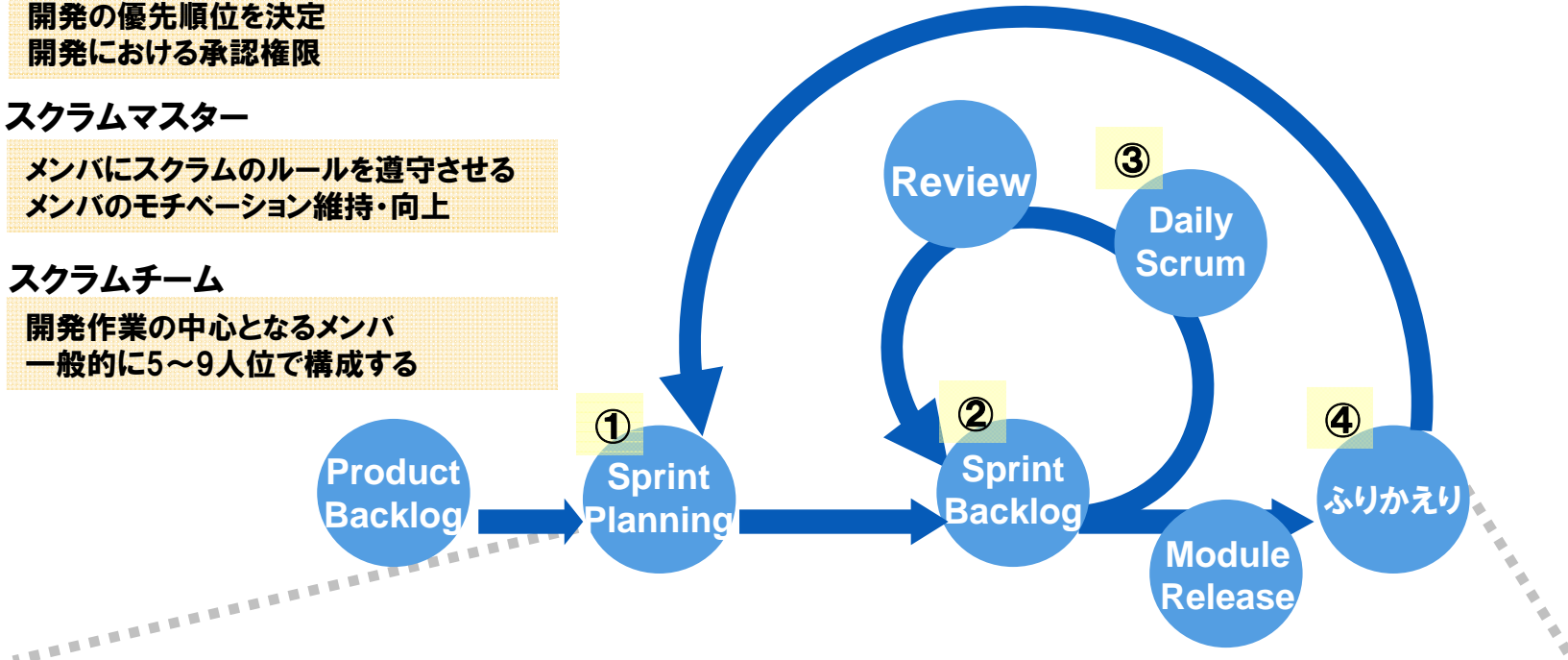
メンバにスクラムのルールを遵守させる
メンバのモチベーション維持・向上

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

TimeBox = 2 weeks と仮定



DAY 1	DAY 2	DAY 3	DAY 4	DAY 5	DAY 6	DAY 7	DAY 8	DAY 9	DAY 10
Daily Scrum	Daily Scrum	Daily Scrum	Daily Scrum	Daily Scrum	Daily Scrum	Daily Scrum	Daily Scrum	Daily Scrum	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

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。チームメンバは規範的な計画や一連のタスクに従うのではなく、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。

スプリント計画の実例

スプリント開始日の午前中に2～4時間かけて、当該スプリントで行う事を明確にし、合意を得る。

- ▶ バックログ項目の明確化
- ▶ タスクの振り分け
- ▶ タスクの見積もり

要求管理システム(Product Backlog Management System) 最終更新日:2011-04-21 14:50:00

要求分析シート表示(PDF) Export(Excel) HELP Back

要求分析シート実行

プロジェクト: [] リスト表示: [要求リスト] ソート: [ID降]

【要求ステータス】

登録待決	採択待決	否採択決	未採択決
0	0	0	0

【要求一覧表示】

新規登録

要求ID	件名	Sprint	要求コード	採択	更新者	更新日
05-0001	...			採用
05-0002	...			採用
05-0003	...			採用
05-0004	...		CHB01	採用
05-0005	...			採用
05-0006	...		DCP01	採用
05-0007	...			未決定
05-0008	...			採用
05-0009	...		BHN02	採用
05-0010	...		FMT01	採用
05-0011	...		SDR01	採用
05-0012	...		FMT02	採用

要求管理システムから当該スプリントのバックログを決定する。
テストチームからのフィードバックがあればそれも加味する。

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。

パイロットプロジェクトでの進め方

要求管理システム(Product Backlog Management System) 最終更新日:2011-04-21 14:59:09

要求分析シート表示(PDF) | Export(Excel) | HELP | Back |

要求分析シート発行

プロジェクト: [] リスト表示: 要求リスト ソート: [ID順]

【要求ステータス】

登録件数 採択件数 否採択数 未採択数

【要求一覧表示】

新規登録

要求ID	件名	Sprint	要求コード	採択	更新者	更新日
05-0001						
05-0002				採用		
05-0003				採用		
05-0004			CHB01	採用		
05-0005				採用		
05-0006			DCP01	採用		
05-0007				未決定		
05-0008				採用		
05-0009			BHN02	採用		
05-0010			FMT01	採用		
05-0011			SDR01	採用		
05-0012			FMT02	採用		

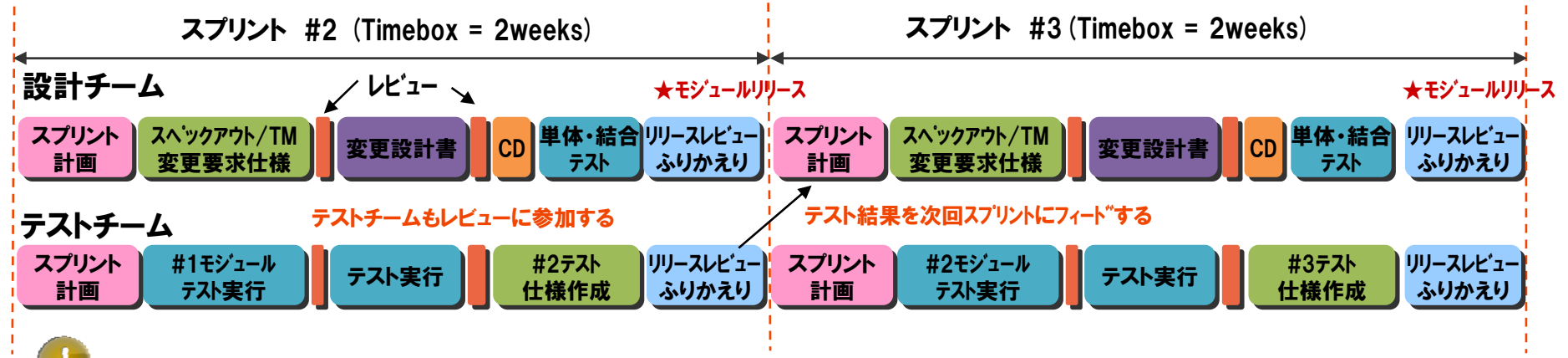
要求をバックログとして管理する

sprint #1

sprint #2

- 業務用組み込み機器の派生モデル バージョンアップ設計 -
- 【期間】 2009/02~2010/04
 - 【人数】 約7名(ソフトウェア,FPGA 5名)
 - 【要求数】 16件
 - 【母体行数】 約650 (KLOC)
 - 【変更行数】 約 4 (KLOC)
 - 【スプリント数】 4回 (Timebox=2weeks)

スプリント間で要求の関連性が疎になるように、ユースケース単位でスプリントバックログを選定する。



関連性の高い要求をユースケース単位でスプリントバックログとして括ることで、XDDPの特徴をできるだけ損ねないようにする。
 ⇒スプリントバックログの括り方を誤ると効果が出ないので、ここは慎重に検討する。
 設計チームとテストチームが連動することで、テストチームも早期にテストを実施することができる。

パイロットプロジェクトでScrumに慣れ、ますますの感触を得ることができたので、次のプロジェクトで本格的にXDDPとAgile開発手法とのハイブリッドプロセスを採用する事を決めた。

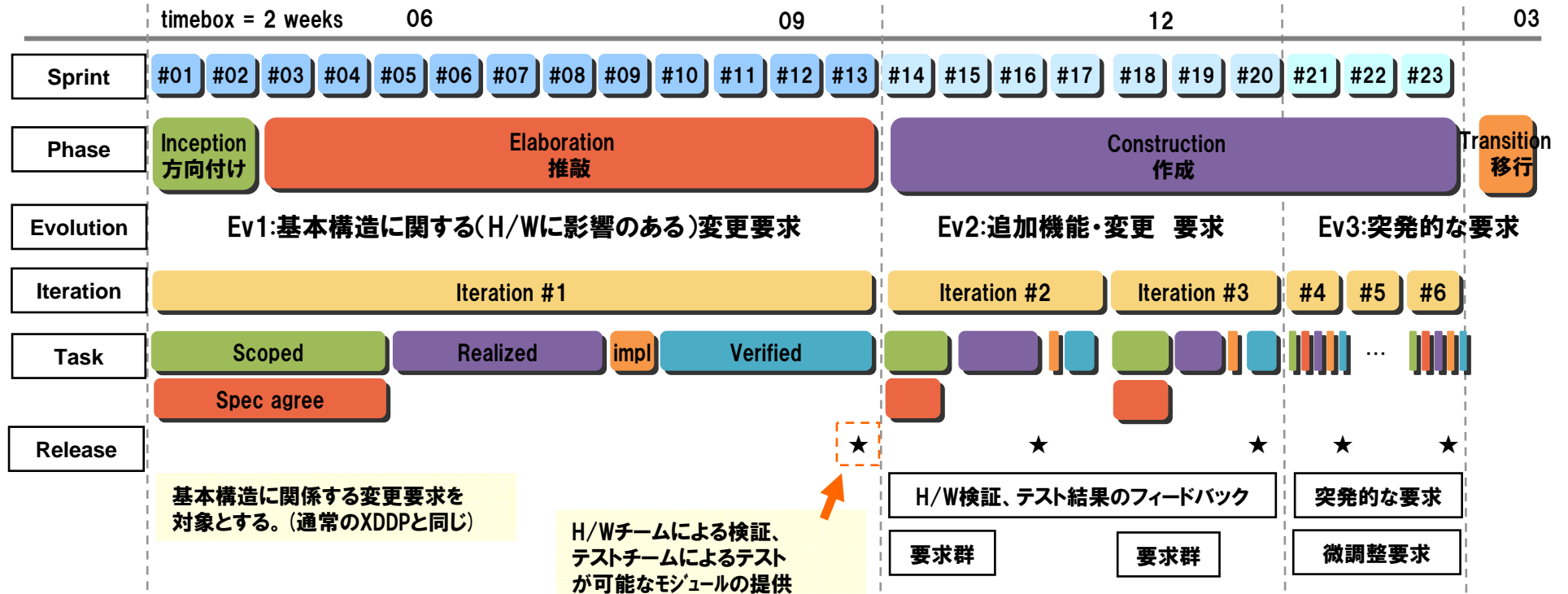
- 業務用組み込み機器の派生モデル開発 -

[期間]	2011/04 ~ 2012/03
[人数]	約15名(ソフトウェア5名, FPGA2名, テスト2名)
[要求数]	約70件
[母体行数]	約500 (KLOC)
[変更行数]	約 50 (KLOC) ※初期見積もり
[スプリント数]	23回予定 (timebox = 2weeks)

※組織変更に伴い、今までとは別領域ドメインでの派生開発

本プロジェクトでの開発プロセス

15/17



基本構造に関する変更要求を対象とする。(通常のXDDPと同じ)

H/Wチームによる検証、テストチームによるテストが可能なモジュールの提供

H/W検証、テスト結果のフィードバック

突発的な要求

要求群

要求群

微調整要求

<主な実行タスク>

Scoped	要求分析シート、スペックアウトドキュメント、第1層TM(列粒度:ブロック)、初期見積もり
Spec Agreed	変更要求仕様書、追加機能要求仕様書、第3層TM(列粒度:ファイル)、上記ドキュメントのレビュー、調整見積もり①
Realized	変更設計書、関数設計書、上記ドキュメントのレビュー、調整見積もり②
Impl	コーディング
Verified	単体テスト、結合テスト

要求群間の依存関係が低くなるように括りを工夫する。(パイロットプロジェクトで行った形式:ユーザーケース単位の括り)

プロジェクト終盤には、微調整要求やデモ要求等の突発的な要求が発生しがちなので、iteration回数を増やす。

(注)開発中プロジェクトの為、現時点での所感

1. XDDP自体がAgileである

そもそもXDDP自体が、派生開発において無駄を排除し合理化を追求した Agileなアプローチであるので、Agile開発手法との併用に違和感は感じなかった。

2. リズムを刻む

Scrumと併用したことで、従来のXDDP単独時には感じなかったリズムが生まれた。個人のリズム、チームのリズムが調和することで、好循環が生まれてくる。

3. 継続的改善

デイリースクラムやTimeBox終了時の「ふりかえり」で、直ぐに現状の問題点を改善する事ができる為、以前よりも、改善し続けるという意識が増してきた。

! XDDPとScrumの融合で、「人」に関する部分が更に良くなってきた事を実感した。また、Scrumを通じて**主体的・自律的**な行動が増えてきた → チーム力の向上

1. XDDPとAgile開発手法との融合

派生開発において、XDDPとAgile開発手法を併用することは可能であり、特にScrumとの親和性は高く、チーム力の向上に効果があると考えます。基本動作(アーキテクチャ等)の変更部分は従来のXDDPで行い、機能追加・調整部分をAgile開発手法の特徴である反復型開発を適用することで、両者の良い所を得ることが可能であると考えます。

2. 開発プロセスを設計する

全ての派生開発プロジェクトにおいて、XDDPとAgile開発手法とのハイブリッドプロセス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を自分達で設計する事が重要であると考えます。

<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。

(偽装工夫にならないように注意)

※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。

ご清聴ありがとうございました