[SQiP2012　経験論文]

XDDPとSCRUMによる チーム型派生開発の取り組み事例

ソニーイーエムシーエス㈱　湖西サイト　勝又　淳

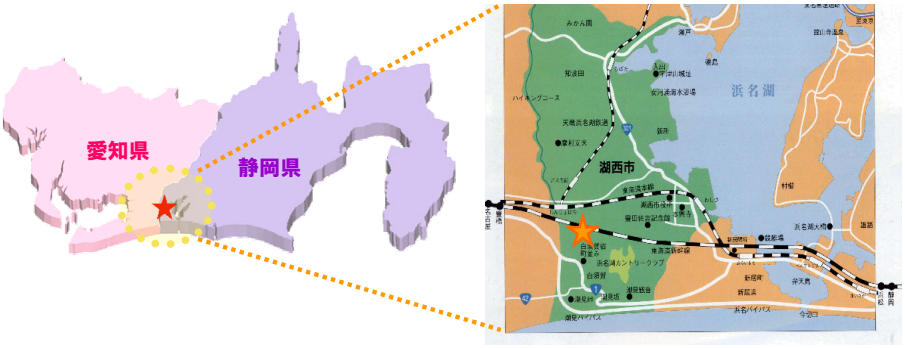
Agenda

1．自己紹介／背景／経緯
2．現状の課題
3．課題解決に向けた施策
4．適用プロジェクトと工夫した点
5．改善効果
6．考察／今後の課題
7．まとめ／主張したいこと

1．自己紹介／背景／経緯

1-1.自己紹介

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㈱　湖西サイト
　　設計第2部門 設計1部
職務：業務用放送機器ソフトウェア設計リーダー



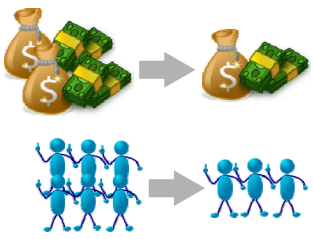
1-2.組織の位置づけ

[設計部門] 派生モデルの設計が多い

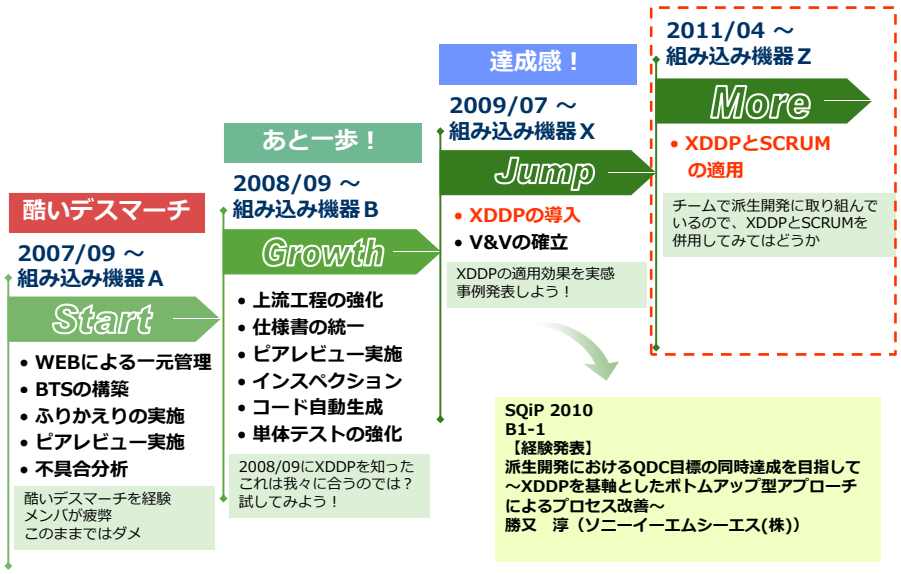
[ビジネス課題の１つ]
派生開発を短納期・高品質・低コストで行う

1-3.背景

長きに渡る景気低迷



1-4.改善活動の歩み



1-5.目標設定/目指すところ

派生開発において

QCD観点で成功し続ける

チームを作りたい

※派生開発
従来の製品に新しい機能を追加したり操作性を改善することで
既存システムから機能を拡張していく開発を意味する。

※QCD観点(品質，コスト，納期)

2．現状の課題

2.現状の課題

[課題１]後戻りに関する課題

・終盤でのハードウェア不具合検出による変更リスクの回避
・ハードウェア検証ソフトウェアの提供時期と方法
★できるだけ後戻りを削減したい

[課題２]ソフトウェア開発チームに起因する課題

・毎回異なるメンバ
・コミュニケーション重視，人的要因トラブルの削減
★個人・チームが成長できるチームビルディングをしたい

[課題３]ソフトウェアリリースタイミングに起因する課題

・プロジェクト終盤での突発的な
デモンストレーション要求，製造テストラン要求
・ビジネスに直結することも多いので迅速に提供したい
★突発的なリリース要求に対応したい

3．課題解決に向けた施策

[施策１] XDDP※1の適用

派生開発で起こりやすい様々な問題を解決
・課題１(後戻り削減)に対する具体的な施策

[施策２] SCRUM※2の適用

継続的改善，定期的リリースへの仕組みづくり
・課題２(チームビルディング)
・課題３(突発的なリリース要求)
に対する具体的な施策

※1 XDDP(eXtreme Derivative Development Process) とは、ソフトウェアコンサルタントの清水吉男氏が考案された、「派生開発※に特化した合理化された開発プロセス」である。
※2 SCRUMとは、アジャイル開発における方法論のひとつであり、プロジェクトマネージメント要素を含んでいる。

3-1.課題解決に向けた施策(1)XDDP

[施策１] XDDPの適用

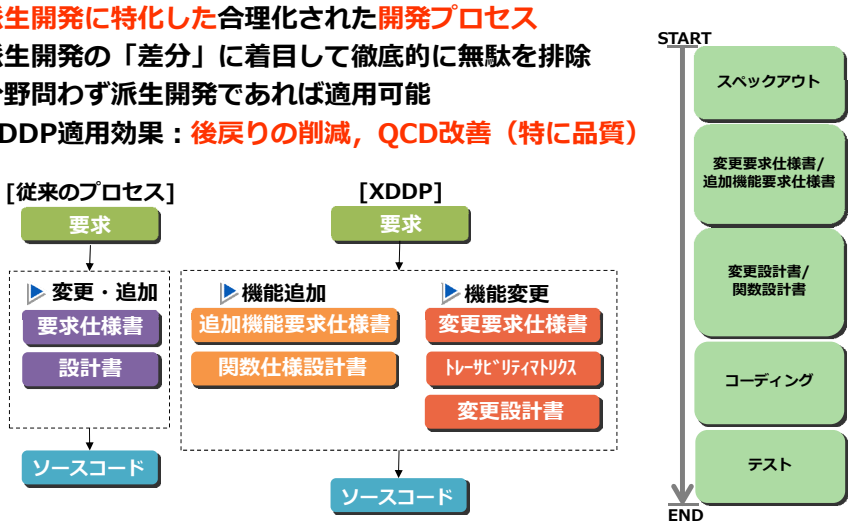
～ 派生開発で起こりやすい様々な問題を解決 ～

・課題１(後戻り削減)に対する具体的な施策

3-1-1.XDDPとは

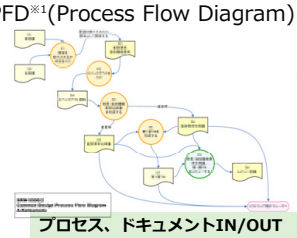
<XDDP(eXtreme Derivative Development Process) の特徴>

・ソフトウェアコンサルタントの清水吉男氏が考案
・派生開発に特化した合理化された開発プロセス
・派生開発の「差分」に着目して徹底的に無駄を排除
・分野問わず派生開発であれば適用可能
・XDDP適用効果：後戻りの削減，QCD改善（特に品質）



3-1-2.XDDPにおける成果物

PFD※1(Process Flow Diagram)



プロセス、ドキュメントIN/OUT

①スペックアウトドキュメント

リバースエンジニアリング

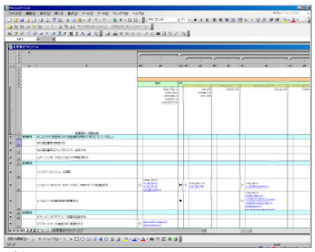
派生開発に最適な
ムダの無いドキュメント

要求/仕様の漏れ・衝突を
防いで後戻りを削減

②変更要求仕様書

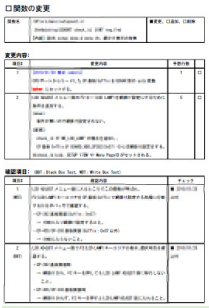
要求/仕様の明確化
USDM※2形式で表現

③トレーサビリティマトリクス



要求/仕様の衝突防止

④変更設計書



コード変更、単体/結合テスト

※1(注意) PFD(Process Flow Diagram)は、XDDPの成果物ではない。
※2USDM(Universal Specification Describing Manner)
システムクリエイツの清水吉男氏が提案した表記法。
仕様の抽出を容易に、かつ抽出漏れを少なくすることを目的としている。

3-2.課題解決に向けた施策(2)SCRUM

[施策２] SCRUMの適用

～ 継続的改善・定期的リリースへの仕組みづくり ～

・課題２(チームビルディング)
・課題３(突発的なリリース要求)

に対する具体的な施策

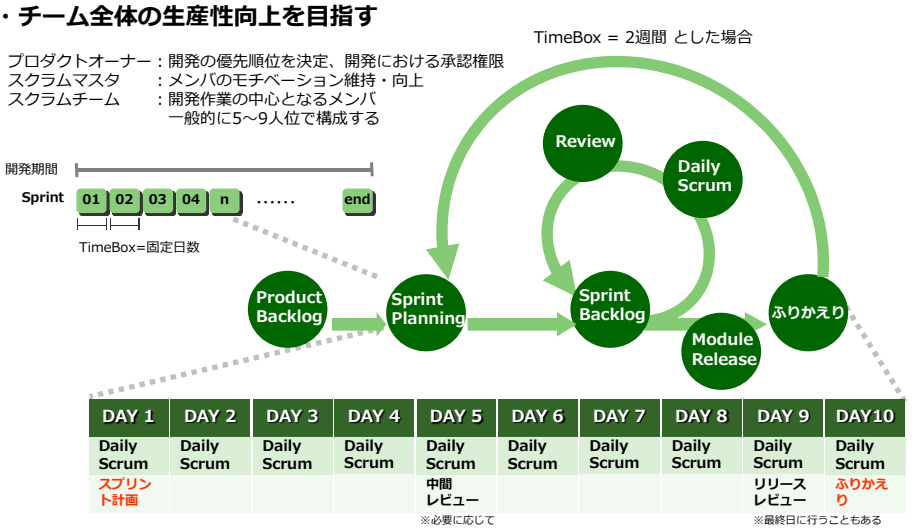
3-2-1.SCRUMの概要

10/26

[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**

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。
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。

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

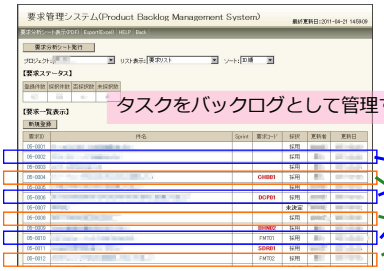


DAY 1	DAY 2	DAY 3	DAY 4	DAY 5	DAY 6	DAY 7	DAY 8	DAY 9	DAY10
Daily Scrum	Daily Scrum	Daily Scrum	Daily Scrum	Daily Scrum	Daily Scrum	Daily Scrum	Daily Scrum	Daily Scrum	Daily Scrum
スプリント計画				中間レビュー				リリースレビュー	ふりかえり
				※必要に応じて				※最終日に行うこともある	

3-2-2.スプリント計画

11/26

スプリント計画の実例



(例)
- **ゴール（ユーザーストーリー）：**
  Sprint #Xでは、再生機能を実装する
- **再生機能に関する要求からタスクを抽出**
  （バックログとして管理）
- **チーム内で役割を決める**

<スプリント計画のポイント>
- **各スプリントの最初に行う**
- **当該スプリントで達成するゴールを決める**
- **ゴールを達成する為に必要なタスクを抽出**
- **チームメンバ自らが役割を決める**

チームの自律を促進する（自己組織型チームの形成）

3-2-3.デイリースクラム 12/26

デイリースクラムの実例

毎日，定刻，スタンドアップ形式
最大でも１５分程度で終了

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**

<デイリースクラムの様子>

<デイリースクラムのメリット>
・リズムが刻まれる。個のリズムとチームのリズムの調和
・人に話すことで頭が整理される。無駄な作業のチェック
・1日単位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)、別途解決の場を設ける
⇒早期に問題発生を予防できる。(最速の改善)

チームの生産性向上に繋がる

3-2-4.ふりかえり 13/26

ふりかえりの実例

各スプリントの最後にKPTふりかえりを行う

- **Keep:**
  今回良かったこと、次回も継続したいこと
- **Problem:**
  今回良くなかったこと、次回改善したいこと
- **Try:**
  具体的な解決策、新たな挑戦

<ふりかえりの様子>

<KPTふりかえりのメリット>
・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
・スプリント毎に行うので常に改善意識を保つことができる
・継続的な改善意識を保つ事で、次のSprintも頑張ろうという気になる

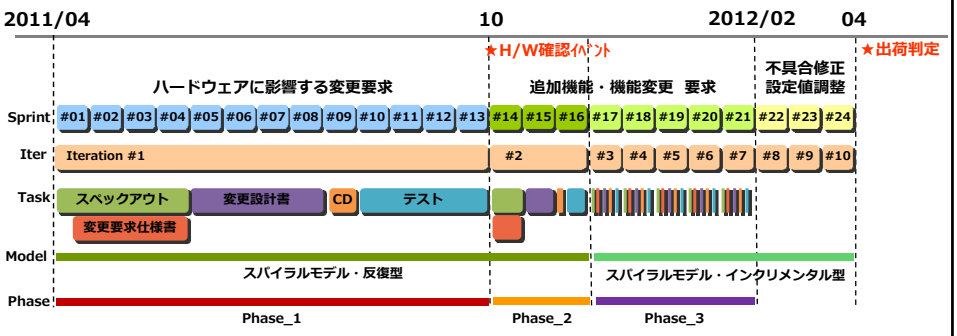
チームの生産性向上に繋がる

4．適用プロジェクトと工夫した点

4-1.改善事例対象プロジェクト概要

案件	業務用組み込み機器の派生モデル開発
期間	2011/04 ～ 2012/03（約１年）
人数	約15名（ソフトウェア ６名(平均), FPGA2名, テスト2名）
要求数	約70件
母体行数	約500ＫLOC
スプリント数	24回(TimeBox=2週間)
特徴	・コアLSIを変更、H/Wブロックの大幅変更 ・初めてXDDP/SCRUM を実践するメンバが多い ・組織変更による異なるドメインでの派生開発
プロジェクト成功判定	・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) ・メンバの充実感（プロジェクト終了後のアンケート）

4-2.XDDP/SCRUM適用の実際の進め方と工夫した点 15/26



[工夫点①]
ハードウェアに影響のある変更要求のみを抜粋してXDDPを適用 (Phase_1)
課題１に対して
終盤でのハードウェア変更リスクを回避

[工夫点②]
ドキュメントを動くモジュールと見立てて#01よりSCRUMを開始　(スプリント0)
課題２に対して
SCRUMの理解度を深める，リズムを刻む

[工夫点③]
Phase_3では主に機能追加
終盤は1sprint=1iteration
課題３に対して
突発的な要求に応えるため
・デモンストレーション要求
・製造テストラン要求

5．改善効果

5-1.改善効果1(QCD観点)

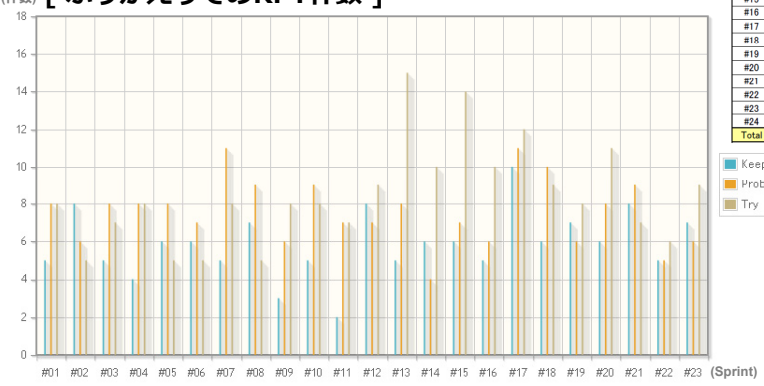
	項目	改善前 XDDP/SCRUM未適用	今回 XDDP/SCRUM適用	改善効果
	概要	[案件] 業務用組み込み機器の派生モデル開発 [期間] 2008/09 ～ 2009/05（約10ヶ月） [人数] 約12名（ソフトウェア5名） [要求] 約50件 [母体] 約650KLOC	[案件] 業務用組み込み機器の派生モデル開発 [期間] 2011/04 ～ 2012/03（約1年） [人数] 約15名（ソフトウェア5名） [要求] 約70件 [母体] 約500KLOC	----
Q	QA不具合指摘件数※1	9 件	0 件	大幅改善
C	ソフトウェア生産性※2	37.4 LOC/人h	150.3 LOC/人h	大幅up
D	出荷遅延日数※3	0日	0日	遅延無し

※1:正式版以降のモジュールにおけるQA検査による不具合指摘件数を示す。
※2:ここでのソフトウェア生産性は、純粋な1時間あたりのコーディング行数(LOC/人H)を示す。
※3:ソフトウェア自体は予定よりも1週間程前倒しできたが，他との兼ね合いで出荷日は予定通りの日程となった。
※ :改善前と今回で、人的リソース・規模が異なるので、単純比較はできないが、QCD観点で改善傾向にあることはわかる。

5-2.改善効果2(継続的改善)

<効果>
・全体を通じて活発な意見が出ている
・全体を通じて継続的な改善意識が見受けられる
・後半のSprintでTry数が多くなっている

[ふりかえりでのKPT件数]

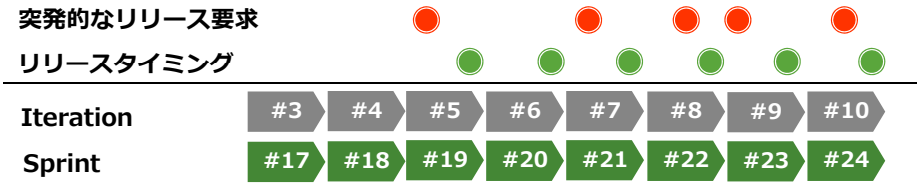


Sprint	Keep	Problem	Try	Total
#01	5	8	8	21
#02	8	6	5	19
#03	5	8	7	20
#04	4	8	8	20
#05	6	8	5	19
#06	6	7	5	18
#07	5	11	8	24
#08	7	9	5	21
#09	3	6	8	17
#10	5	9	8	22
#11	2	7	7	16
#12	8	7	9	24
#13	5	8	15	28
#14	6	4	10	20
#15	6	7	14	27
#16	5	6	10	21
#17	10	11	12	33
#18	6	10	9	25
#19	7	6	8	21
#20	6	8	11	25
#21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

※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

5-3.改善効果３(リリースタイミング)

18/26



Sprintごとにリリースすることで，
突発的なリリース要求に迅速に応えることができた

5-4.アンケート結果

19/26

プロジェクト終了時にチームメンバに対して以下のアンケートを実施

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. 我々なりにテー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ら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。
→(課題)効率の良いリバースエンジニアリング方法の追求、レビューでの発言者の偏り　等の各タスク毎の課題
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、充実感はあまりありません。

アンケート結果からも、今回のXDDPとSCRUMの適用は、メンバにとっても非常に有意義な取り組みであったと思われる.

5-5.Episode

Episode Ⅰ
最速の出荷判定会議

Episode Ⅱ
チームメンバに緊急事態発生！その時に…

6．考察／今後の課題

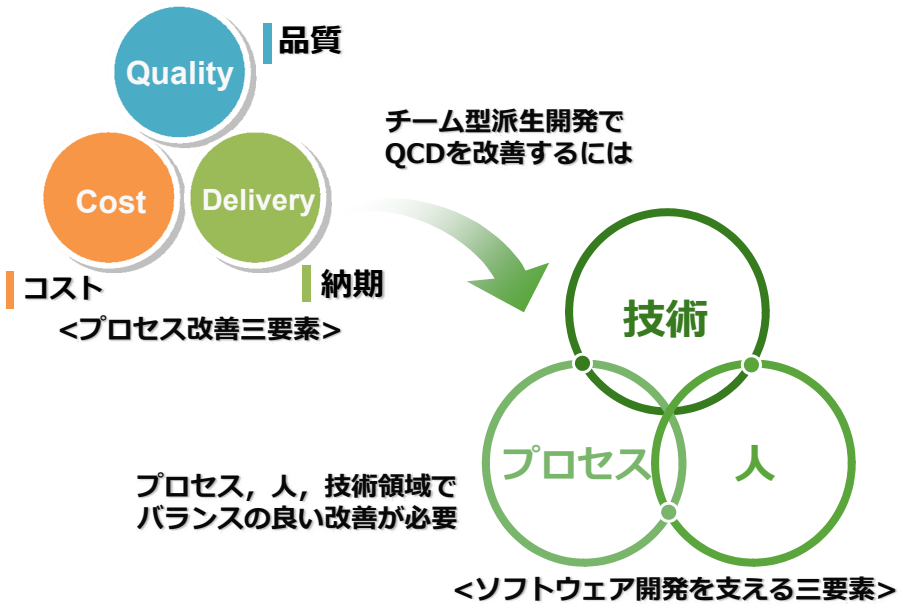
6-1.考察１:現状の課題解決とXDDP/SCRUMの相関 21/26

XDDPとSCRUMを併用することで現状の３つの課題をすべて解決

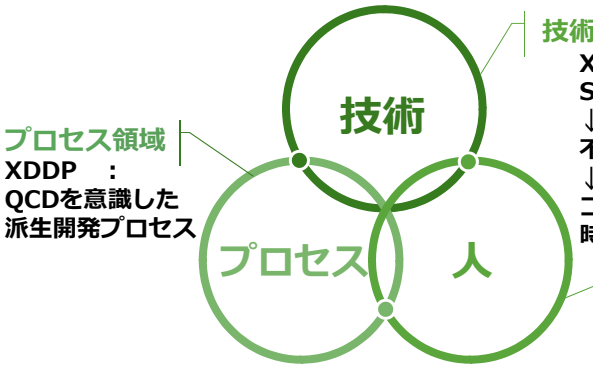
課題	課題解決に有効な手法
1. 後戻りに関する課題(終盤でのH/W変更リスク)	XDDP
2. 人とチームが成長するチームビルディング	SCRUM
3. ソフトウェアのリリースタイミング	SCRUM

	適用効果	不足しがちな要素
XDDPのみ適用	・QCD意識（特に品質） ・後戻り削減	・チームの一体感 ・継続的改善意識 ・定期的なリリース
SCRUMのみ適用	・チームの一体感 ・継続的改善意識 ・定期的なリリース	・QCD意識 ・後戻り削減
XDDP/SCRUMの適用	・QCD意識（特に品質） ・後戻り削減 ・チームの一体感 ・継続的改善意識 ・定期的なリリース	不足要素なし XDDPとSCRUMの 併用デメリットは無し

6-2.考察２:ソフトウェア開発を支える三要素① 22/26



6-2.考察２:ソフトウェア開発を支える三要素② 23/26



技術領域
XDDP　:プロセス領域改善
SCRUM :人領域改善
↓
不具合や人的要因問題が軽減
↓
コアである技術領域に
時間を費やすことができる

プロセス領域
XDDP　:
QCDを意識した
派生開発プロセス

人領域
SCRUM :
継続的な改善意識をもった
自律型チームの形成

<ソフトウェア開発を支える三要素>

6-3.今後の課題 24/26

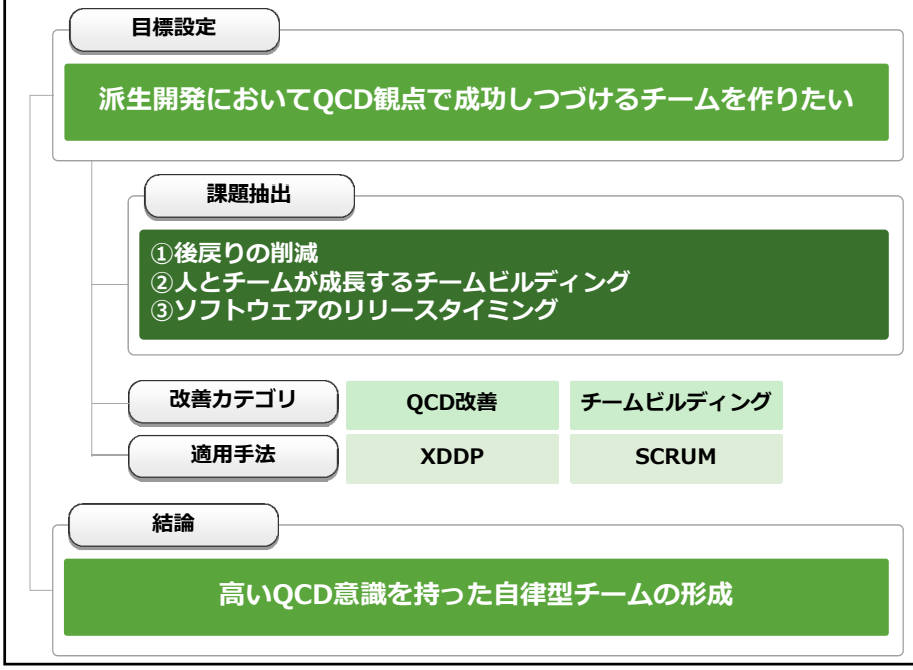
課題点

1．システムテストの重複
・複数回のIterationに分かれる為、システムテストでの重複項目が発生
→リグレッションテストの自動化、バックログの分け方等の改善

2．開発規模が大きくなった場合
・10人以上のチームになった場合，分割などの対処が必要

7．まとめ／主張したいこと

7-1.まとめ



7-2.さいごに（主張したいこと）

チームでソフトウェア開発を成功させるためには

高いQCD意識　チーム主導 自律型チーム　継続的改善

(例)XDDP　(例)SCRUM

個人の成長
チームの成長

あらゆるムダを削減し
技術を磨く時間を増やす

ご清聴ありがとうございました