XDDPの概要について

（Vol.0.1）

2012年10月18日　佐藤 創

Copyright (C) 2012-2013 So Sato, All Rights Reserved.

更新履歴

版数	日付	内容	担当
0.1	2012/10/18	新規作成	佐藤 創

Copyright (C) 2012-2013 So Sato, All Rights Reserved.

XDDPとは?

Copyright (C) 2012-2013 So Sato, All Rights Reserved.

● XDDPとは？

●XDDP（eXtreme Derivative Development Process）

主に組込み系の派生開発の作り込み品質の向上を目的とした、開発プロセス

・作り込み品質の向上の結果　⇒　テスト工程での欠陥摘出・修正工数の削減

・作り込み品質の向上の手段　⇒　各種ドキュメントの作成と、ドキュメントを基にしたレビューの実施を可能にし、影響箇所の検討漏れ・誤りに気付きやすくする。

- （株）システムクリエイツ 清水吉男氏が開発経験を基に提唱。システム開発からコンサルティングに転じ、プロセス改善の経験と実績を通じて定義した開発プロセス。
- 派生開発推進協議会（AFFORDD）にて成果の活用、および研究会を推進。
  ⇒ http://www.xddp.jp/index.shtml

Copyright (C) 2012-2013 So Sato, All Rights Reserved.

XDDPとは？

- 組込みの世界を初め、現在では「派生開発（流用母体に対する変更・追加・削除・移植を行う開発）」が主流となっている。
- 派生開発は短納期や、自分で書いたソースコードではないなどの、新規開発にはない特徴がある。
- しかし、組織で定義している開発プロセスは新規開発のものがほとんど。
- 新規開発用のプロセスを派生開発に適用すると不都合あり。
- そもそも派生開発用のプロセスが定義されていないこともある。
- その結果、品質の低下、生産性の低下、開発者の慢性的な過負荷などの悪影響がある。
- 品質・生産性を向上させ、技術者が次のプロジェクトにむけた技術習得や学習に時間を割ける組織を目指し、技術者が楽しく仕事に取り組めることを目指す。

Copyright (C) 2012-2013 So Sato, All Rights Reserved.

XDDPが想定する派生開発の課題

Copyright (C) 2012-2013 So Sato, All Rights Reserved.

● XDDPで想定する「派生開発の課題・現状」

- 一般的に期間が短い
- 別の担当者が開発したソースコードを読む必要がある
- ドキュメントが不足しているためソースコード理解が難しい
- これまでの改修によって、当初の設計思想が崩れている
- 時間的にも「全体」を理解して取り掛かることが困難
- 期間が短く、１人プロジェクトになることがある
- 変更要求と仕様とのすみわけが曖昧

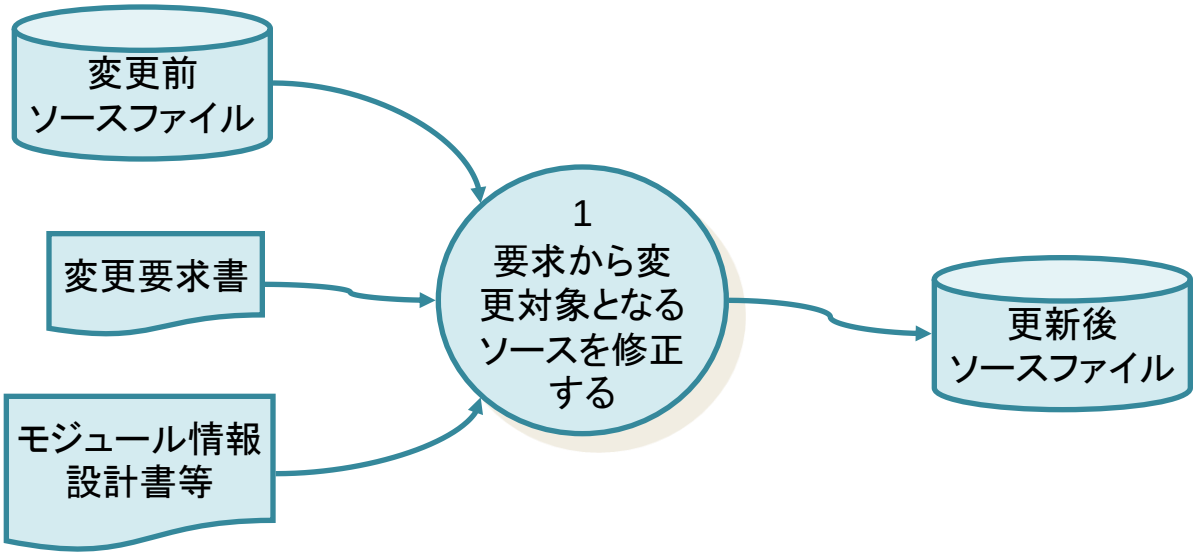
→ 「部分理解」しかできず、**影響範囲を漏らす**

- 派生開発独自の課題があるので、これらに対応した開発プロセスが必要
- XDDPで想定する現状の派生開発プロセスと、XDDPで提唱する開発プロセスを比較する

Copyright (C) 2012-2013 So Sato, All Rights Reserved.

● XDDPで想定する「これまでの派生開発のプロセス」その1
・最悪パターン

・開発プロセスがないパターン

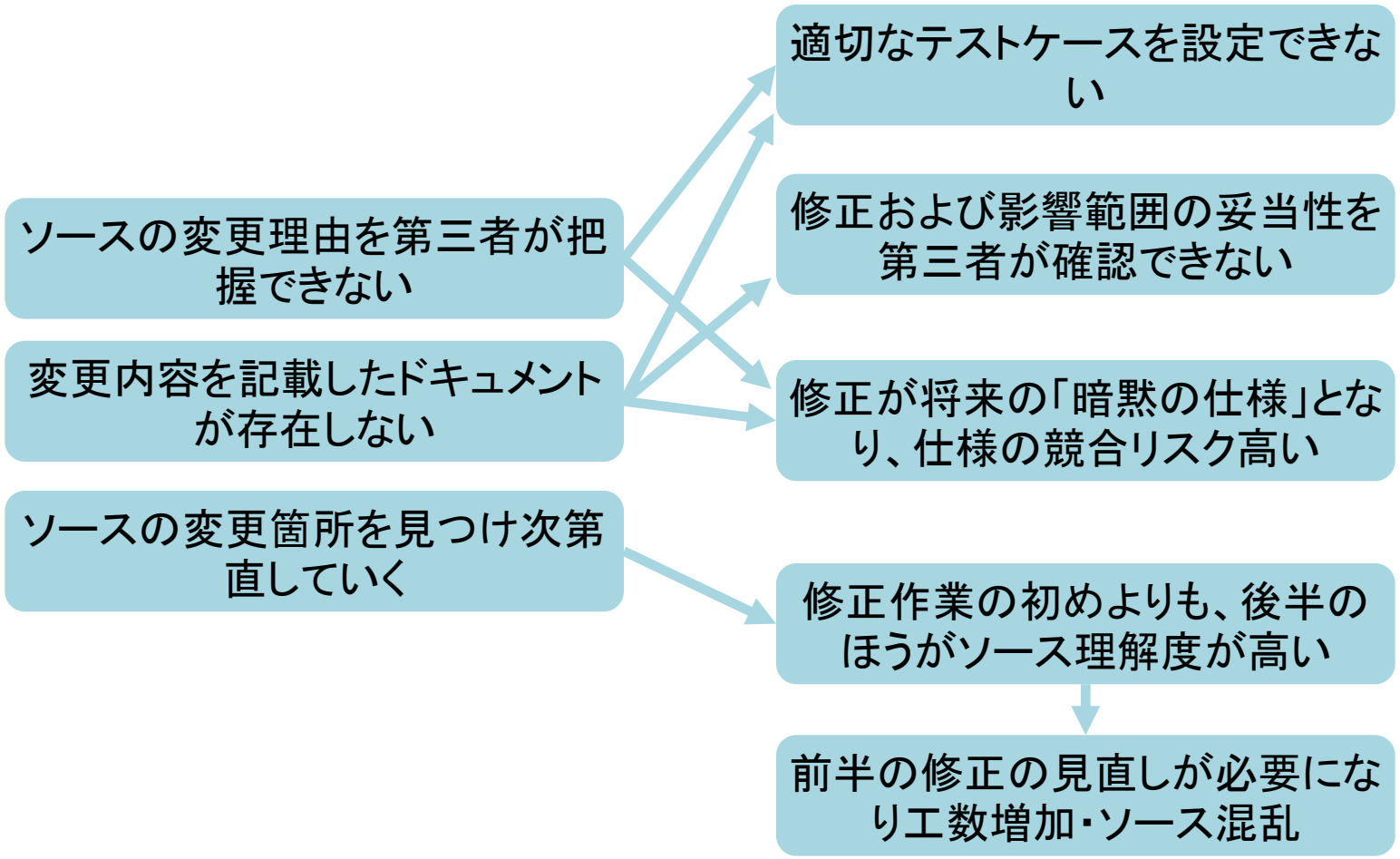


- 変更要求を見て、対応するソースを見つけ出し、見つけ次第にどんどん修正をしていく。
- 最終成果物はソースのみ。
- 場合によっては最終成果物として、設計書等も更新する。

Copyright (C) 2012-2013 So Sato, All Rights Reserved.

● XDDPで想定する「これまでの派生開発のプロセス」その1 「最悪パターン」プロセスによる悪影響

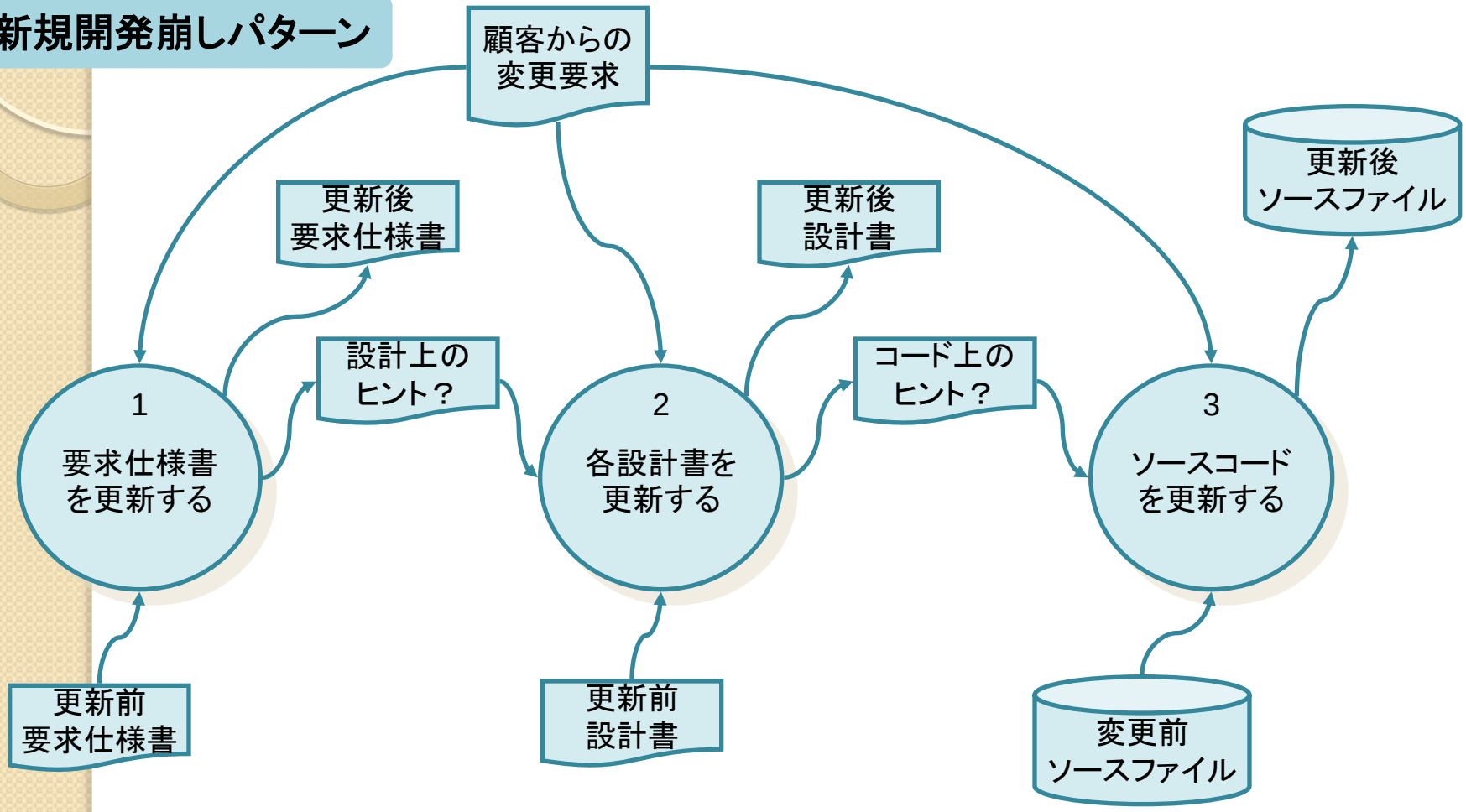
・開発プロセスがないパターン



Copyright (C) 2012-2013 So Sato, All Rights Reserved.

● XDDPで想定する「これまでの派生開発のプロセス」その2
・新規開発崩しパターン

・新規開発崩しパターン



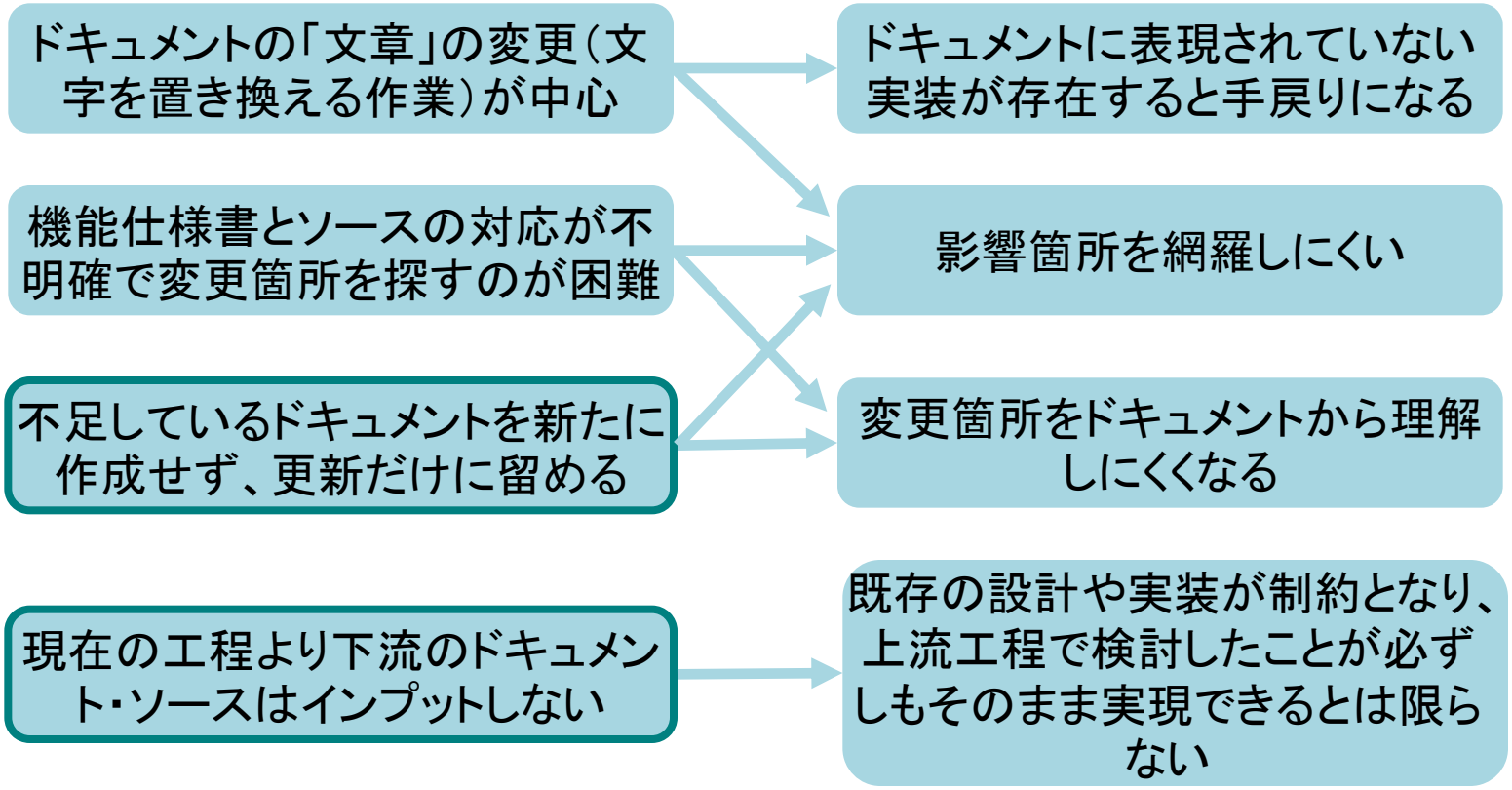
（出典：「派生開発」を成功させるプロセス改善の技術と極意，清水吉男，技術評論社，2007年，1.2.20 図1.9を参照）

- 既存のドキュメントだけを更新しながら上流から下流へと工程をすすめる。
- 工程間ではレビューをする場合もある。

Copyright (C) 2012-2013 So Sato, All Rights Reserved.

● XDDPで想定する「これまでの派生開発のプロセス」その2 「新規開発崩し」プロセスによる悪影響

・新規開発崩しパターン



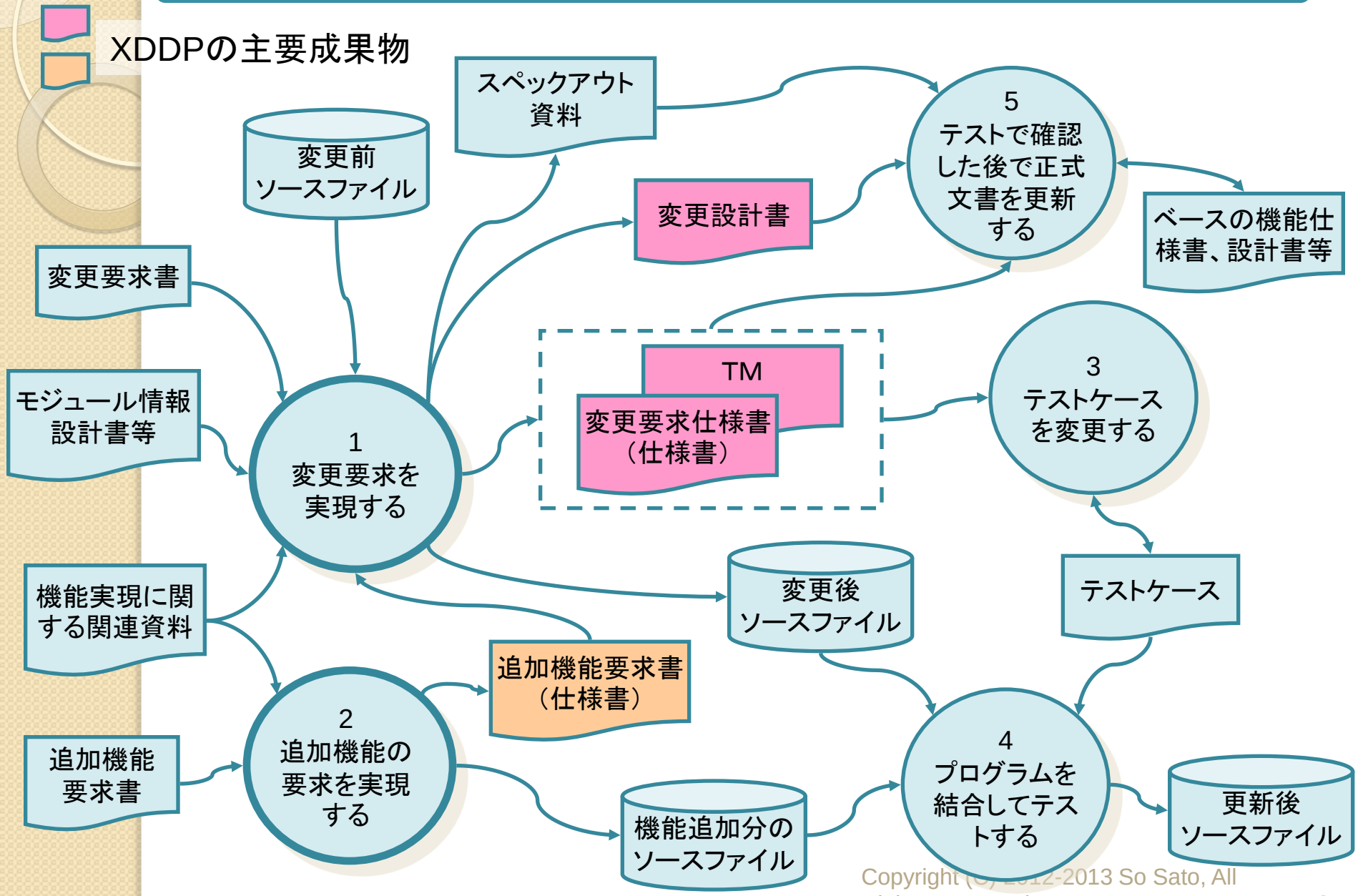
XDDPには記載されていないが当方が直面した問題・課題

Copyright (C) 2012-2013 So Sato, All Rights Reserved.

XDDPが提案する課題への対応

Copyright (C) 2012-2013 So Sato, All Rights Reserved.

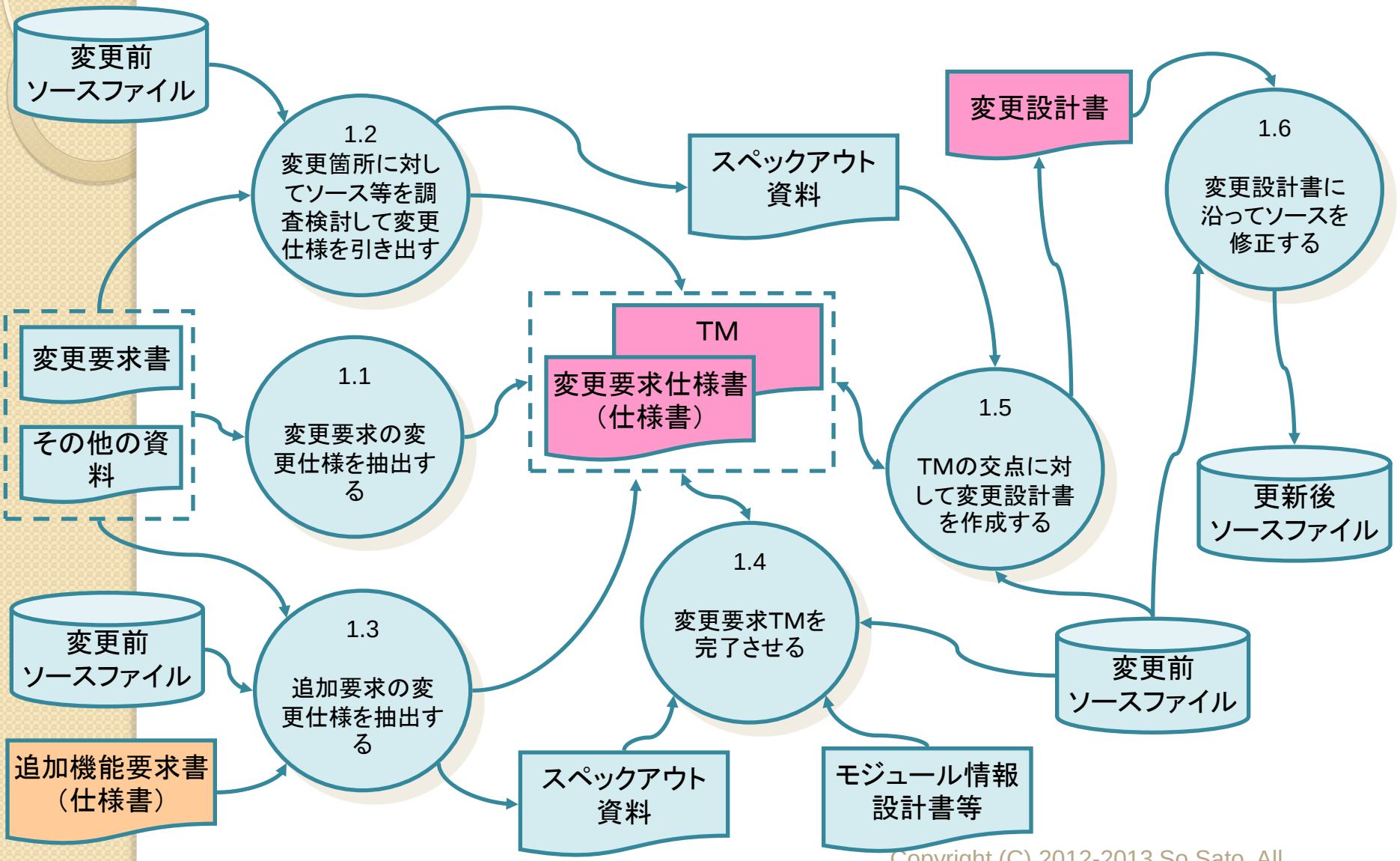
● XDDP全体のプロセス（変更と追加の混合）



（出典：「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，技術評論社，2007年，2.4 図2.3を参考に作成）

Copyright (C) 2012-2013 So Sato, All Rights Reserved.

● 1. 変更要求を実現する を詳細化したプロセス



（出典:「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，技術評論社，2007年，2.4.2 図2.4を参考に作成）

Copyright (C) 2012-2013 So Sato, All Rights Reserved.

● XDDPで作成する成果物の3点セットの概要と期待効果

- 成果物3点セットの概要

成果物名称	カバレッジ	記述内容	レビュー機会	
変更要求仕様書	What （Why）	何を変更するのか？ どのような振る舞いを変更するのか？ なぜ変更するのか？	○	○
トレーサビリティ・マトリクス	Where	変更する仕様がどこにあるのか？	○	
変更設計書	How	具体的な変更方法を記述する。	○	○

（出典：AFFORDD勉強会資料（セミナー版）派生開発アプローチ：XDDPの詳細，派生開発推進協議会，P30の図を参照）

- 成果物を作成することでレビュー機会を確保する。
- なぜ変更するのか、何を変更するのか、どこを変更するのか、どのように変更するのか、が見えるドキュメント作成を行うことで、レビューでの指摘が行いやすくなる。

Copyright (C) 2012-2013 So Sato, All Rights Reserved.

● XDDPで作成する成果物の3点セットの概要と期待効果

● 主要な成果物3点セット

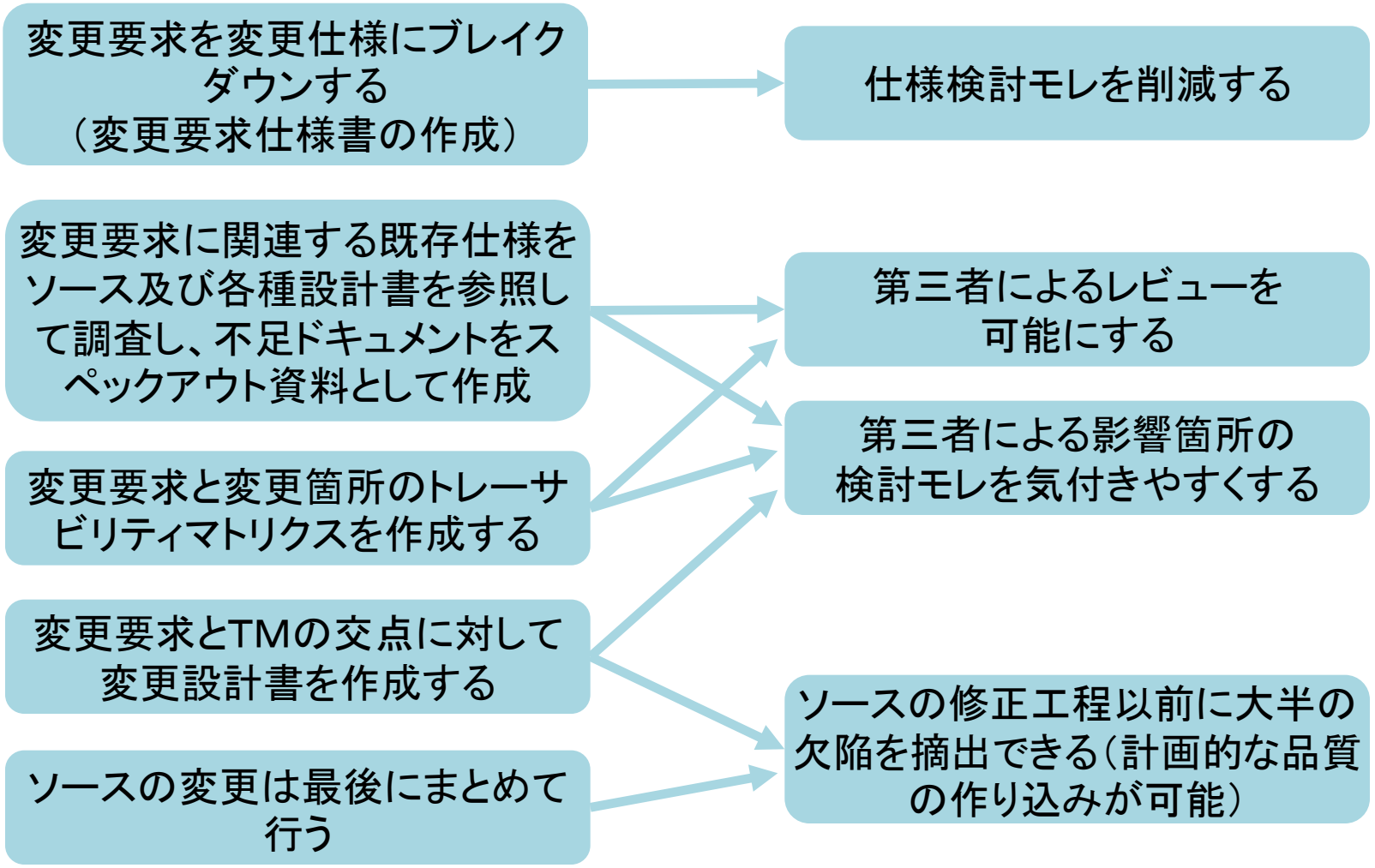
成果物名称	概要	期待効果
変更要求仕様書（USDM）	要求と仕様を区別し、要求から導き出される仕様を網羅したり、仕様から想定される要求を導き出して、要求を実現するための仕様を洗い出すツール。 要求と仕様を階層構造で表現したもの。	・要求と仕様を明確に区分し、漏れのない仕様の抽出をする ・関係者間での合意の形成を促進する
変更要求トレーサビリティ・マトリクス	変更要求－変更仕様－変更箇所（ソースなど）をつなぎ、要求や仕様がどの変更箇所で実現されるのかを示すツール。 縦軸に要求と仕様を記載し、横軸にソースファイルを記載し、マトリクスで変更箇所をトレースできるようにしたもの。	・要求－仕様－変更箇所が、漏れなく落としこまれていることを第三者が判断できる ・変更箇所の漏れがあった場合に、第三者がレビュー等で気づきやすくなり、影響箇所の検討漏れを予防できる
変更設計書	変更要求トレーサビリティ・マトリクスにおいて、変更箇所の1つ1つに対応して作成される、変更差分の設計書。	・具体的な変更内容を事前にレビューすることができる

● その他の関連成果物

成果物名称	概要	期待効果
スペックアウト資料	仕様の検討時点で、不明確な機能や動作についてリバースエンジニアリングで仕様をまとめた資料。	・不明確な動作仕様を明らかにでき、第三者のレビューを可能にする ・既存ドキュメントに不足している資料を追加することで今後の保守に役立てる

Copyright (C) 2012-2013 So Sato, All Rights Reserved.

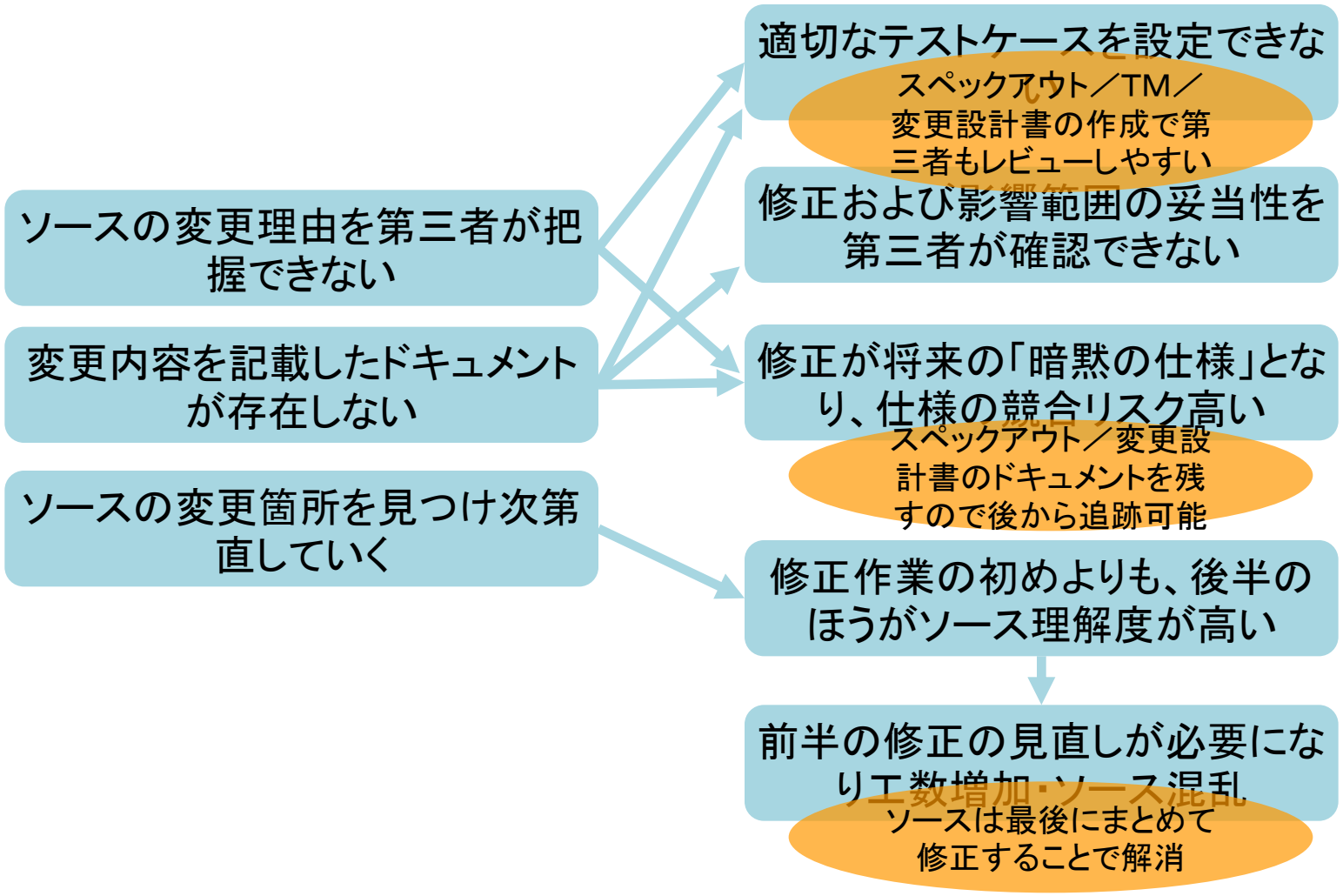
● XDDPで解決を目指すこと



Copyright (C) 2012-2013 So Sato, All Rights Reserved.

● XDDPで想定する「これまでの派生開発のプロセス」その1
プロセスによる悪影響が解消されるか？

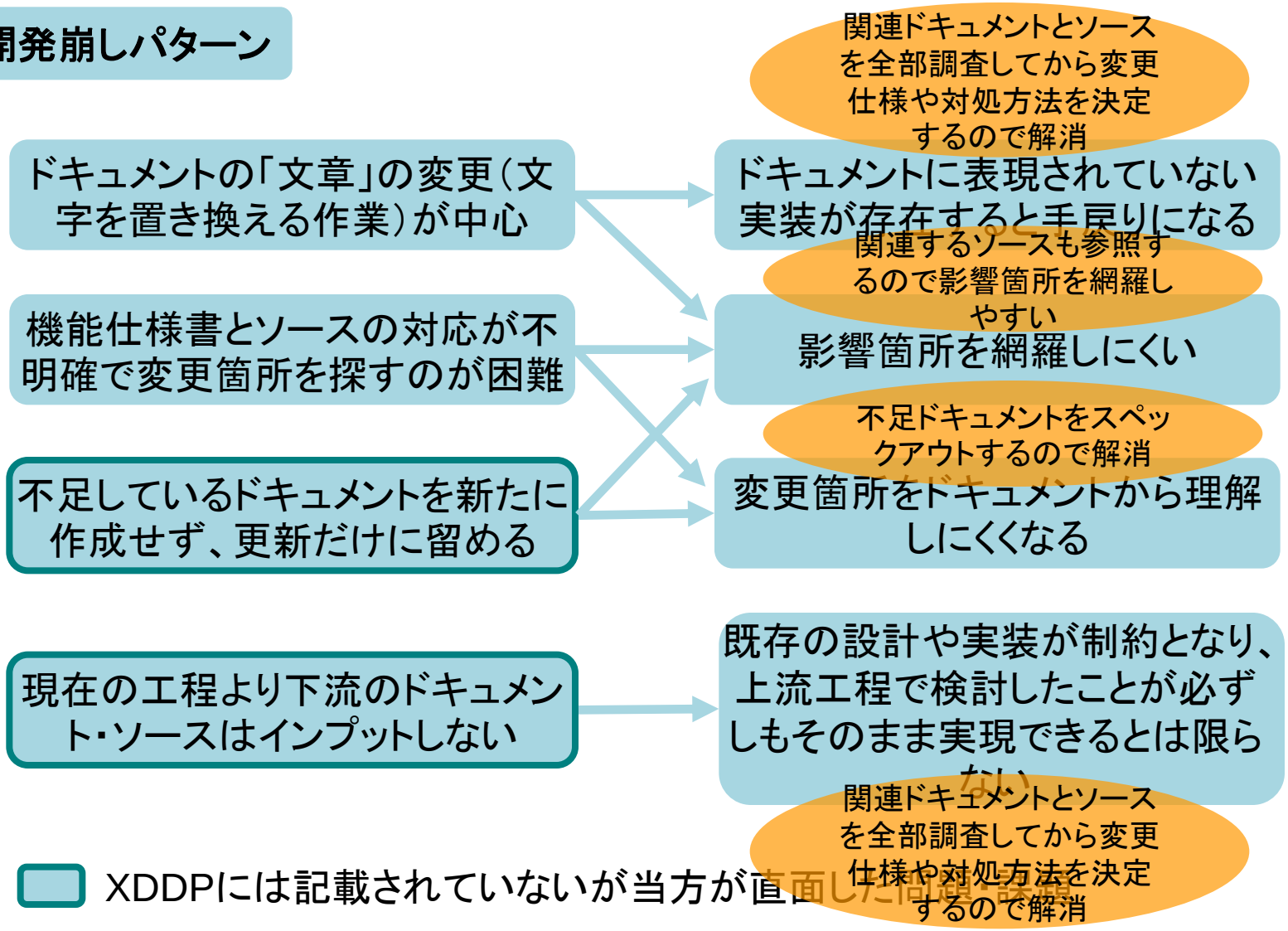
・開発プロセスがないパターン



Copyright (C) 2012-2013 So Sato, All Rights Reserved.

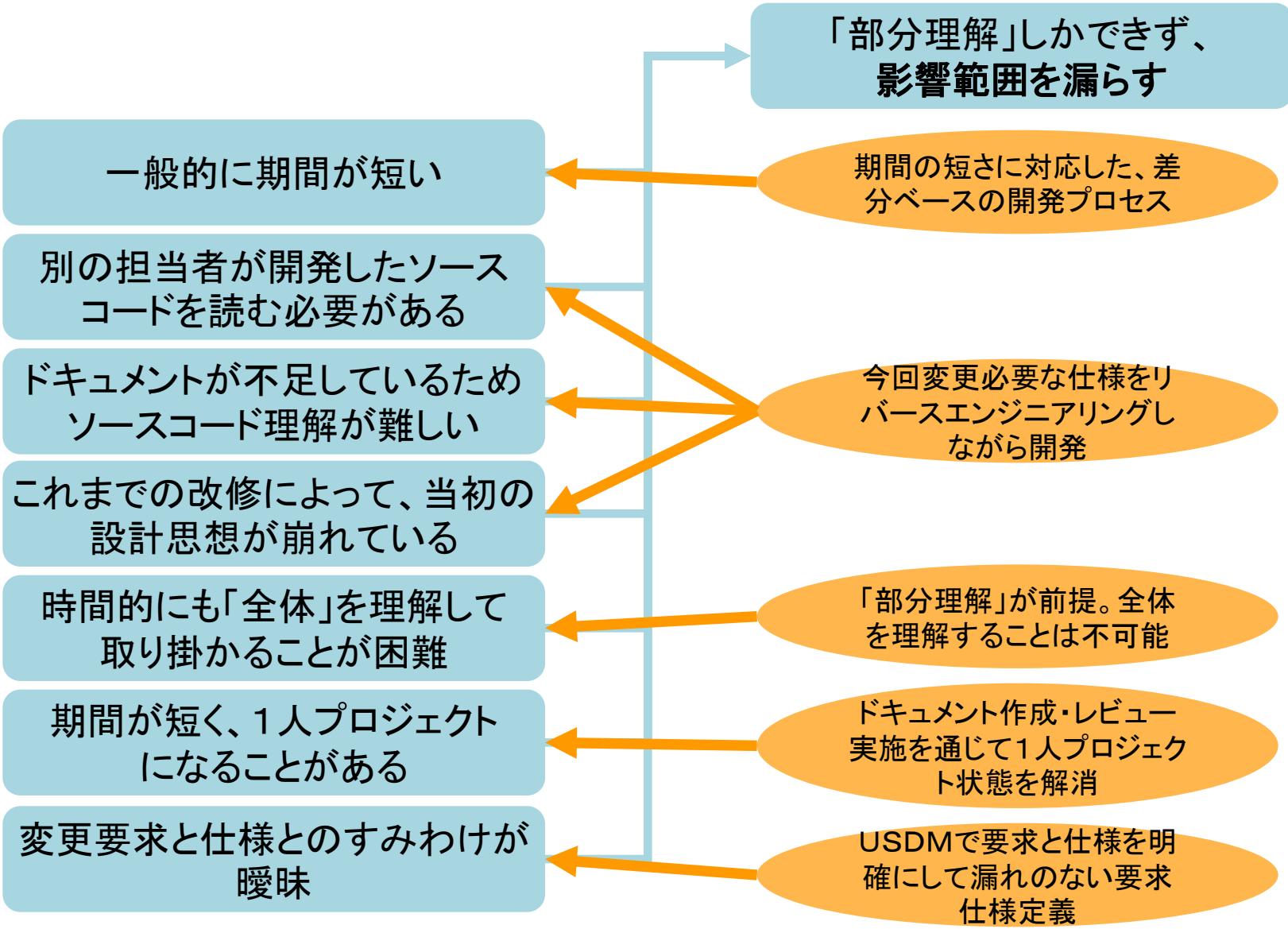
● XDDPで想定する「これまでの派生開発のプロセス」その2 プロセスによる悪影響が解消されるか？

・新規開発崩しパターン



Copyright (C) 2012-2013 So Sato, All Rights Reserved.

● XDDPで想定する「派生開発の課題・現状」に対応したプロセス



Copyright (C) 2012-2013 So Sato, All Rights Reserved.

XDDPへの疑問点

Copyright (C) 2012-2013 So Sato, All Rights Reserved.

● XDDPの効果に関する疑問点

- 疑問点と調査結果（成果物に関連すること）

疑問点	調査結果
変更要求TMで影響箇所のモレをなくす、ということだが、そもそもXDDPは「部分理解」が前提である。部分理解しかしていない状態で、どのように影響範囲を特定するのか？そこの方法論は何？	●具体的には変更要求TMと、レビューにて影響箇所のモレを見つける。 ⇒他の担当者の「気付きを誘発」することが主眼であり、方法論としては存在しない。 ●補足資料として、「リソース間のつながりを示す資料」、「機能間の依存関係を示す資料」などがあると、より影響範囲を特定しやすい、との提言はあるが、具体性および実証成果に欠ける。
TMは、横軸にファイル名を用いるということだが、それではレビューの際に詳細すぎて、ソースコードを熟知した人がいないと「気付きが誘発」されない。	●ソースコードで粒度が小さすぎるのであれば、別の機能単位などの有効な基準を用いる。 ●横軸にどのような基準を採用するかは、プロジェクト特性に応じて判断が必要。
差分設計書とは具体的にどんなフォーマットを想定している？	●今回の変更箇所が明確に判断でき、変更仕様が実現されていることがわかるものであればなんでもよく、規定はしていない。サンプルとしては関数差分仕様書や、フローチャートのようなものを想定している。 ●変更前と変更後が明確なのであれば問題なく、特にフォーマットは規定していない。
スペックアウトするときに有効な技法や、記述フォーマットは何か？	●特に規定はしていない。プロジェクトに応じて適宜選択が必要である。

Copyright (C) 2012-2013 So Sato, All Rights Reserved.

● XDDPの効果に関する疑問点

- 疑問点と調査結果（プロセスに関連すること）

疑問点	調査結果
変更箇所を見つけ次第にソース修正して、ドキュメントを残さないという最悪プロセスを想定しているので、すでにドキュメントを残しているプロセスに対しては大きな効果はないのでは？	●新規開発崩しでの課題解決にはつながる。 ●元々採用しているプロセスによっては、大きな効果を得られない可能性がある。
正式ドキュメントへの変更内容の反映は、テストが終わってから、とあるが、では、なにをインプットとしてテストケースを抽出するのか？XDDPの主要な成果物はすべて差分なので、影響範囲を含めたテストケースの抽出ができるのか？	●どのようにすれば、影響範囲を含めた漏れのないテストケースを抽出できるのかは規定していない。今後の課題である。
規定されているプロセスが大きすぎる。例えば「変更箇所に対してソース等を調査検討して変更仕様を引き出す」プロセスで、ソースを調査して、要求仕様として抽出するときに、どんなことに留意してスペックアウトすると良いのか？ここでの検討・調査漏れがあれば、変更仕様も漏れてしまう。	●特に詳細な手順は定めていない。変更要求に関連すると思われるソースやドキュメントを調査して、必要と思われる仕様をスペックアウトしながら、理解した内容を、変更要求仕様書にまとめていく、という大枠のみを示している。
レビューにおいてXDDP成果物のみだと、差分のみが示されており、直接変更した箇所以外の機能や動作について、ドキュメントから確認することができない。これに気付けるのはTMのみだが、粒度がソースファイル単位であり、気付きを得にくい。	●レビューではTMをベースとして、影響箇所の漏れや誤りについて「気付きを促す」ことを主眼としている。 ●ソースファイル単位で気づきを得られない場合は、適切な粒度の単位を用いるか、他の補助資料、補助プロセスを適宜加える。
レビューによって品質を確保することを重視しているプロセスであると考えられるが、プロジェクトによっては有識者が不在でレビューアも母体システムについて理解が乏しいケースも想定される。そうした場合の品質確保はどのように行われるのか。	●有識者が不在のケースについては言及されていない。 ●明確に定めた成果物を作成する時点で、設計者自身が設計誤りに気付きやすくなるため、設計時点での品質の作り込み効果も多少は期待できると想定される。

Rights Reserved.

● XDDPで行われている研究活動

研究会のテーマ	
障壁の克服方法	上位の要件開発技法と「USDM」の連携
「USDM」の入門	ソフトウェア品質要求の定義
「XDDP」の入門	「USDM」のリスク管理への応用
「XDDP」とテストプロセスとの接続	SPLと「XDDP」の連携
影響箇所の気付き	「USDM」の支援ツール
Agile開発との連携	「XDDP」の支援ツール
ハードの派生開発への適用	「PFD」の支援ツール
大規模システムへの効果的対応	USDMと形式言語との接合における曖昧表現の克服
ビジネス領域での「XDDP」の活用	派生開発におけるスペックアウトの仕方
「USDM」とユースケースの連携	「XDDP」とモデル駆動開発の融合

- まだまだ周辺領域の研究は始まったばかりなので、今後の検討課題は多い。

Copyright (C) 2012-2013 So Sato, All Rights Reserved.

参考文献

[1] 「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，技術評論社，2007年

[2] 【改訂第2版】[入門＋実践] 要求を仕様化する技術・表現する技術　～仕様が書けていますか？，清水吉男，技術評論社，2010年

[3] 派生開発アプローチ：XDDPの詳細　～派生開発にマッチした開発アプローチ AFFORDD勉強会資料，派生開発推進協議会，2012年

[4] 派生開発推進協議会のサイト，http://www.xddp.jp/

Copyright (C) 2012-2013 So Sato, All Rights Reserved.