当方が携わった派生開発のプロセス改善内容について

（Vol.0.1）

2012年10月24日　佐藤 創

更新履歴

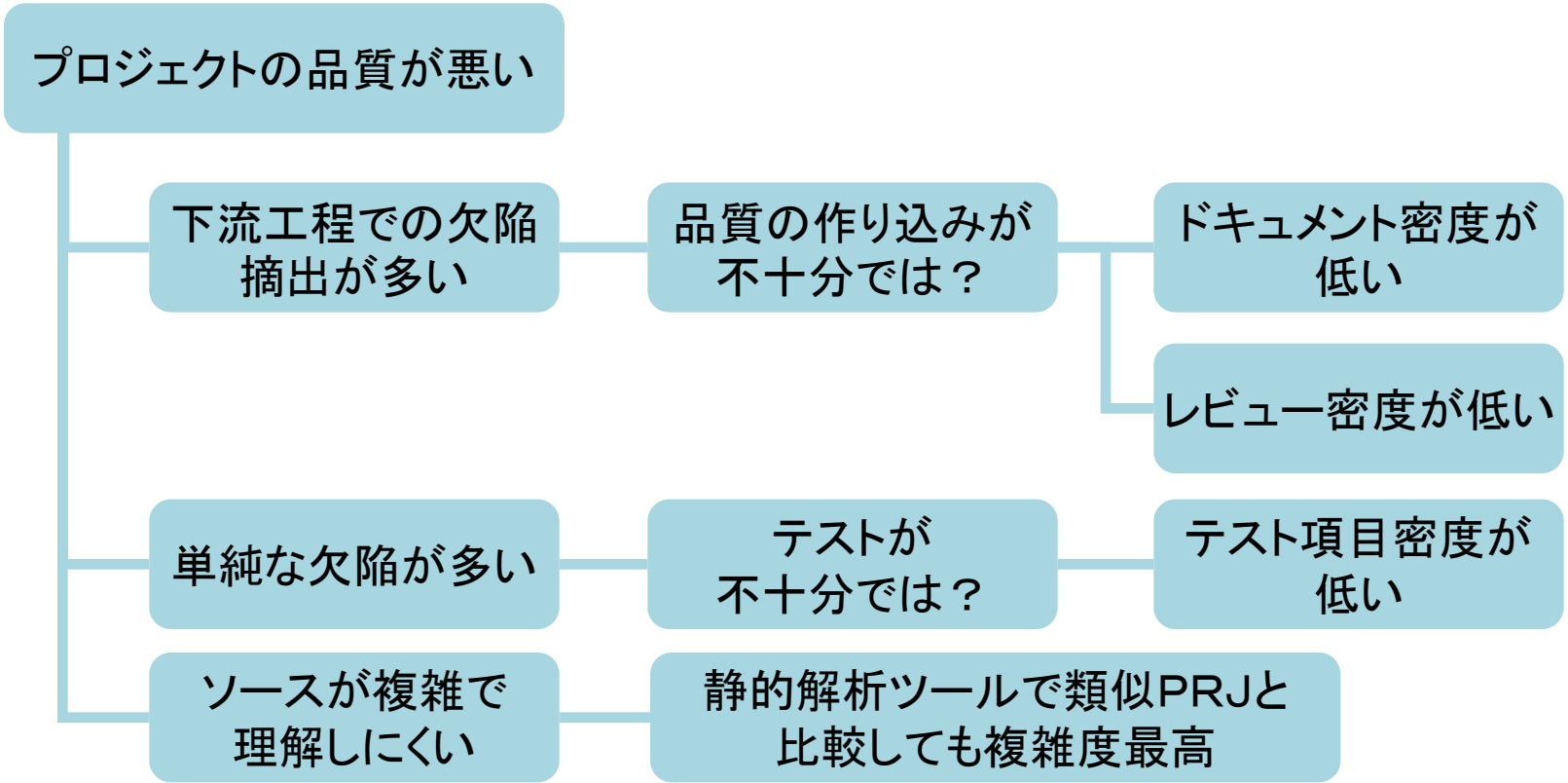
版数	日付	内容	担当
0.1	2012/10/24	新規作成	佐藤 創

PRJで遭遇した派生開発の問題

● 対象の派生開発PRJの特徴

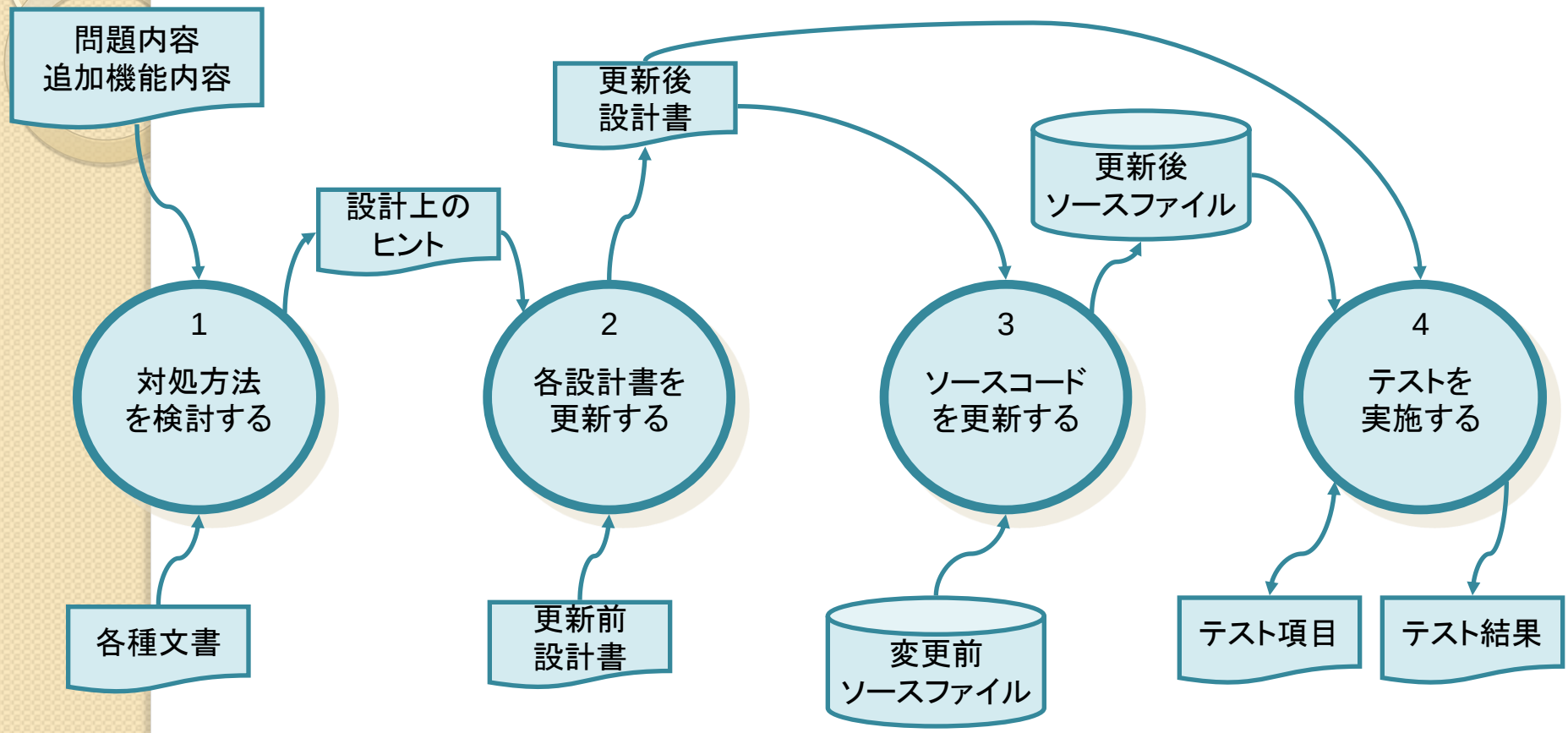
●対象となる派生開発PRJの概要

某組込み系システムの改修・保守プロジェクト。他の組織で開発された本稼働直前の総合テストからシステムを引き継ぎ、総合テストの実施、および機能追加、残留欠陥の是正を実施するプロジェクト。



● 対象の派生開発PRJで遭遇した問題・課題（1／4）
1. 当該PRJの開発プロセス（1／2）

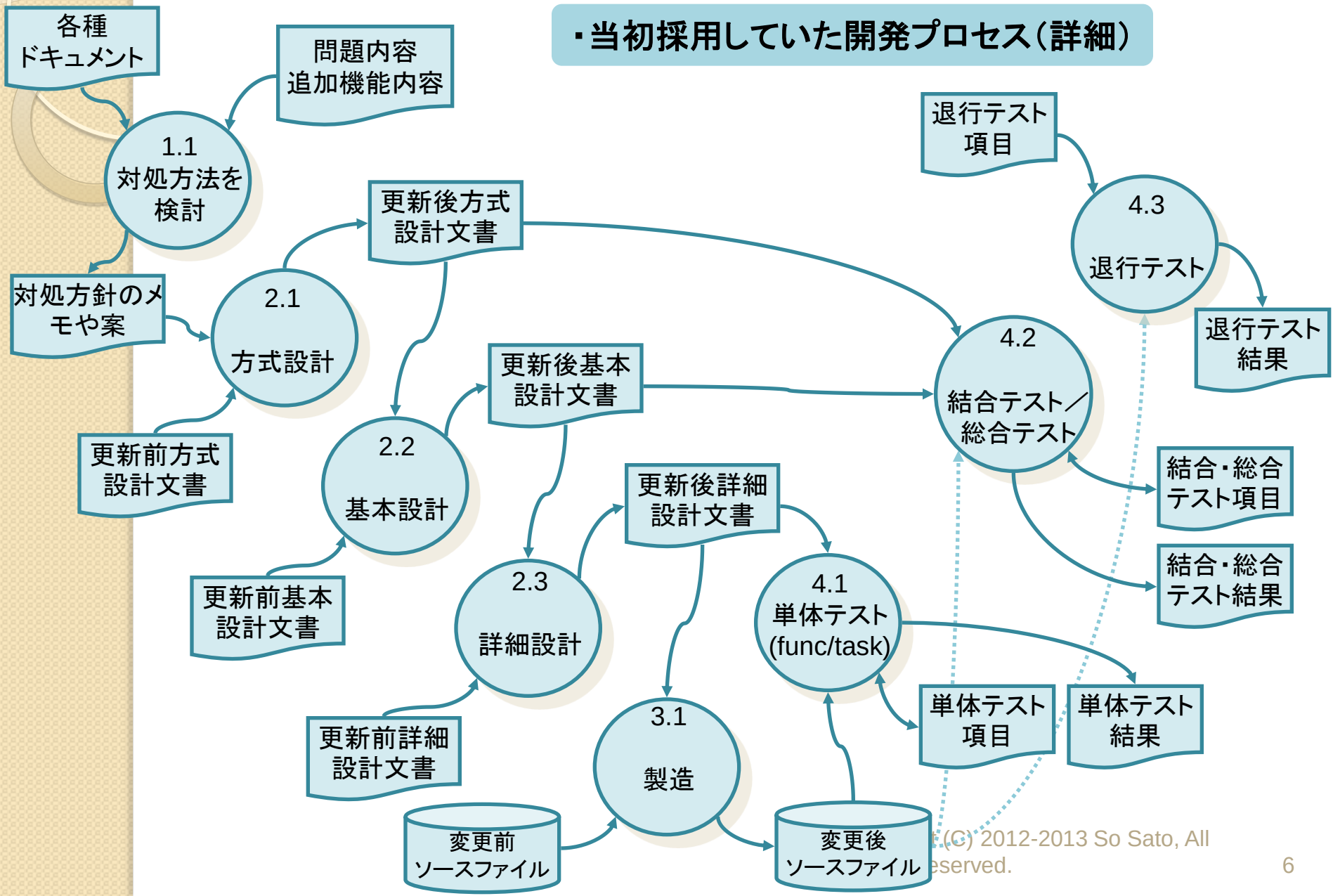
・当初採用していた開発プロセス（概要）



- 既存のドキュメント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必ず行う。
- 基本的にはウォーターフォールのV字モデル開発。

● 対象の派生開発PRJで遭遇した問題・課題（2／4）
1. 当該PRJの開発プロセス（2／2）

・当初採用していた開発プロセス（詳細）



● 対象の派生開発PRJで遭遇した問題・課題（3／4）
2. 当該PRJで遭遇した問題・課題（1／2）

PRJ序盤での問題

問題対処に伴う動作仕様の変更を実施

結合テスト工程で、欠陥やデグレが発生

品質の作り込みができていなかった

調査・検討不足
ドキュメント不足

レビューで品質確保できない

原因となった作業プロセス

不足しているドキュメントを新規作成せず、更新だけに留めている

開発工程より下流のドキュメントはあまりインプットせずにトップダウンで決定している

その結果

既存ドキュメントの品質に左右される

ドキュメント間のトレーサビリティが見えない

ドキュメントから読み取れない「暗黙の制約条件」が後からバグとなって表出

レビューへのインプット・ドキュメントが不足していて品質確保できない

得た教訓

- 上流工程でソースまで全て調査してから設計しないと、既存の実装の制約を把握できない
- 既存ドキュメントは不足しているので、不足内容を補いながら設計を行う必要がある

① フロントローディング強化

② ドキュメントのトレーサビリティ強化

● 対象の派生開発PRJで遭遇した問題・課題（4／4）
2. 当該PRJで遭遇した課題（2／2）

PRJ中盤での問題

	原因となった作業プロセス	その結果
レビューアが、設計内容の妥当性を検証することが難しい 設計工程で調査のやり直し、調査観点の漏れが発生	結論に至る過程が述べられていない（結論のみのドキュメント）	レビューで「どうやって調査したのか」をヒアリングして効率悪化 どんな思考過程を経て結論が出たのかが分らない（結論が正しいことを証明できない）⇒調査し直し
結合テストにおいて、設計時点では検討していなかった処理を試験範囲を広げてテストしたところ、関連する欠陥が多発した 最初に想定した対処の有効範囲が限定的。水平展開漏れで手戻り発生	対処内容を決定する時点で、「対処スコープ」に関する議論が不足	「個別の問題に対する対策」に焦点が当たり、水平展開が漏れる 対処の有効範囲が明らかにされず、影響範囲の調査モレが発生する

得た教訓

- 結論までの過程もドキュメントに全て表現せよ（頭の中を全てアウトプットする）
- そもそもの対処スコープが変われば、設計・実装のインパクトも大幅に変わる（対処スコープを明確にしてから設計を開始）

③ ドキュメントのアカウンタビリティ強化

④ 超上流の強化（対処スコープ検討）

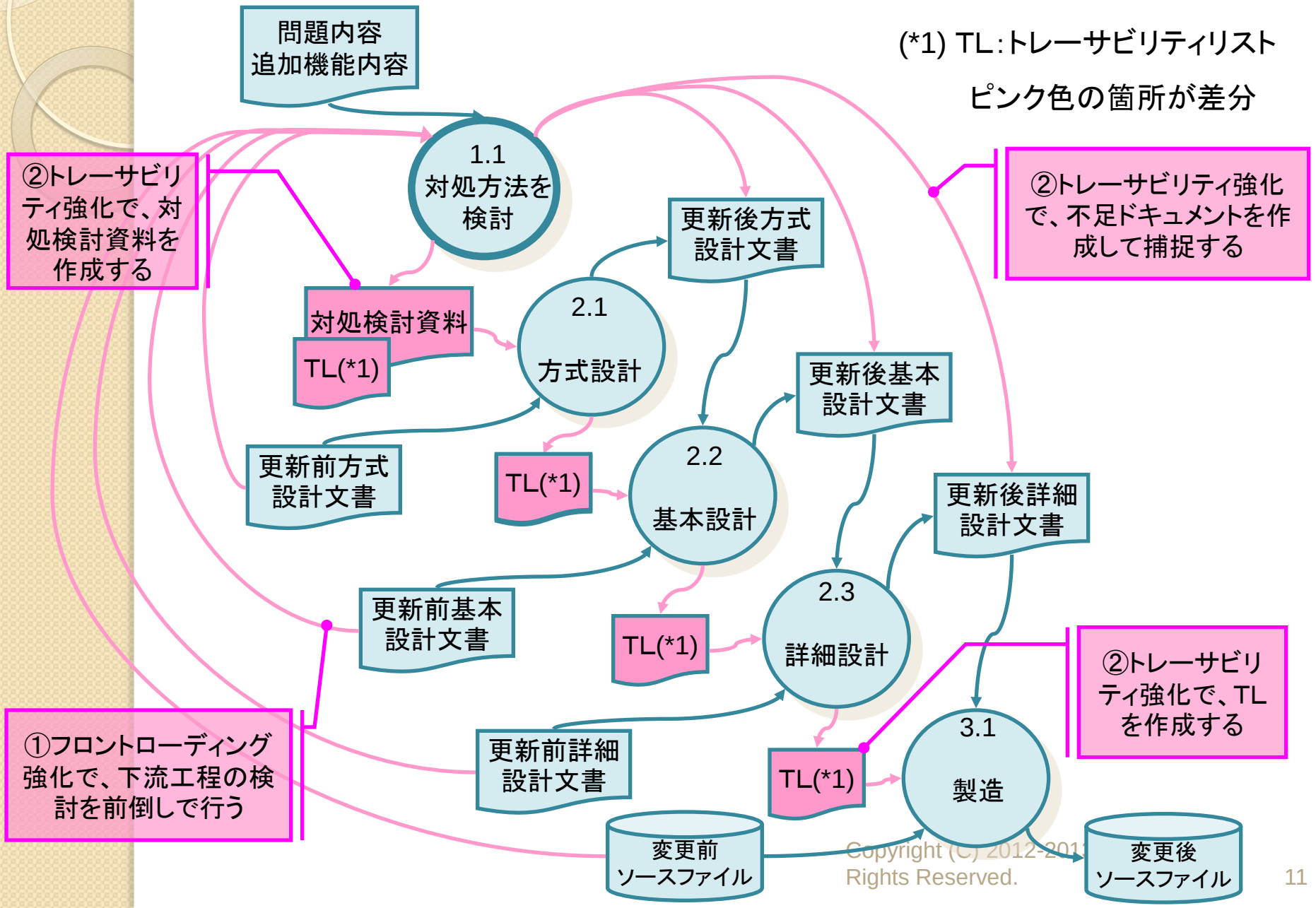
PRJで実施した問題への対応

● 施策の一覧

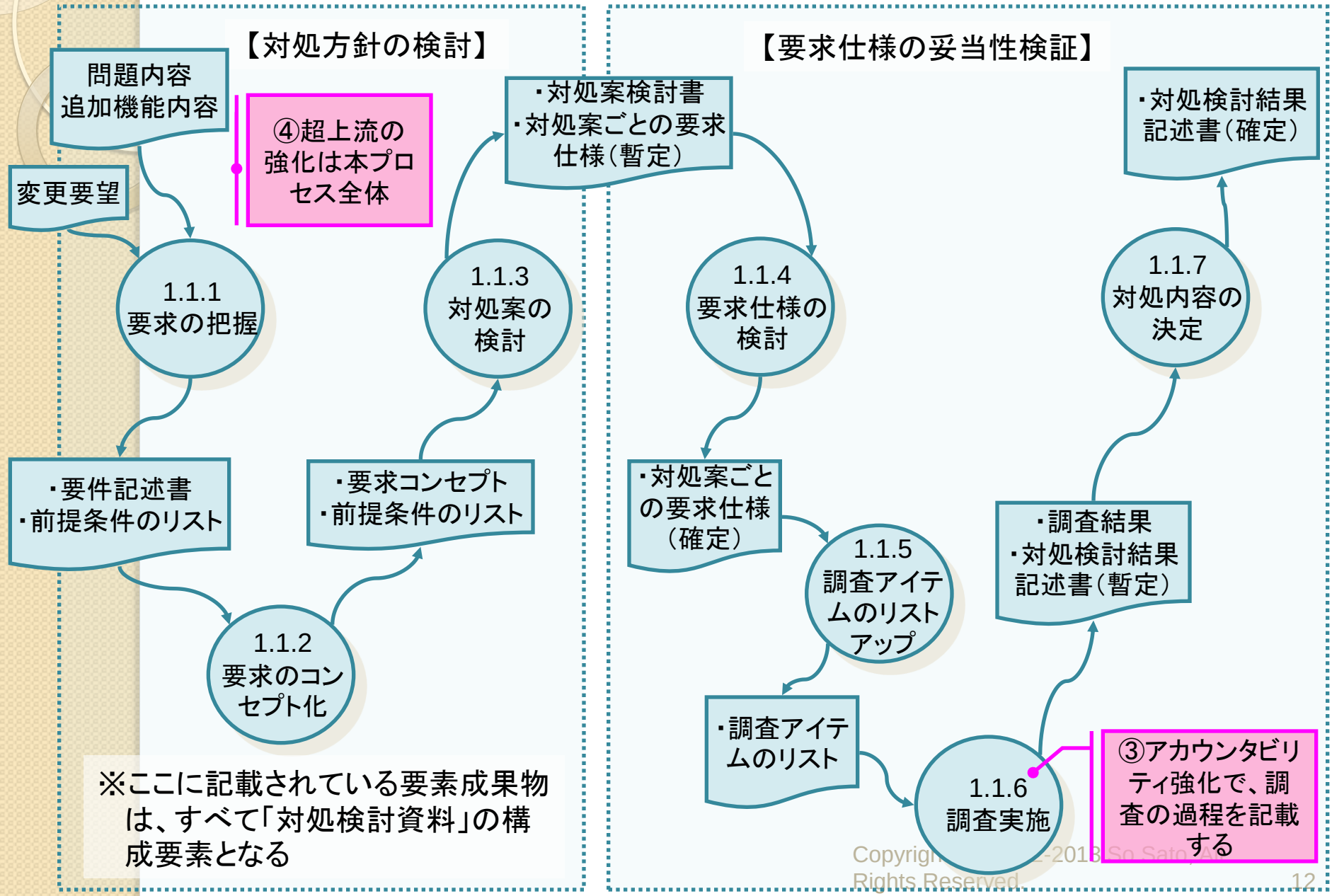
● 施策の一覧

施策名称	概要	期待効果
① フロントローディング強化	●対処が妥当と判断できるまで全工程の調査を前倒し実施する。 ●対処方法を検討する際は、各種設計書とソースコードの全てを参照して、下流工程の調査も前倒しをする。 ●成果物に限らず実機を用いての動作確認を加えても構わない。	●上流工程で検討した対処案が、下流工程に至ってから問題ないのかを確認するのではなく、上流工程で妥当性を検討できる。 ●既存のシステム動作や実装を踏まえた上で対処を検討できるので、手戻りが少なくなり、計画的なスケジューリングが可能になる。
② ドキュメントのトレーサビリティ強化	●基本設計－詳細設計－ソースまでの成果物を参照し、「対処検討資料」として、全工程に横串を通した対処検討資料を作成する。 ●ドキュメントが不足している既存仕様があれば、「対処検討資料」に現状の動作仕様を調査したものを添付するなどして、不足ドキュメントを補う。 ●各工程の成果物の変更箇所をマトリクスで示し、変更箇所を追跡可能にする。	●対処内容を「対処検討資料」に、全工程を通じて記載することで、工程間の壁に阻まれずにドキュメントを記載できる。 ●不足しているドキュメントを補うことで、レビューの際にレビューアが理解しやすくなる。 ●変更箇所マトリクスを作成することで、レビューの際にレビューアが理解しやすくなる。
③ ドキュメントのアカウンタビリティ強化	●「対処検討資料」では、検討した結果だけでなく、結論に至るまでの過程もドキュメント化する。 ●「対処検討資料」をストーリーのあるドキュメントとして作成する。	●結論を導く過程（プロセス）もレビューで検証できるので、レビューアも理解しやすく、また考え方のミスなどを摘出しやすくなる。 ●結論が変更になった場合も、結論を導いた過程を遡ることで、どこに影響が及ぶのかをロジカルに判断することができ、変更に強くなる。
④ 超上流の強化	●検討した対処案の有効範囲（スコープ）を検討したり、個別の変更要求の裏に潜んでいる「要求」を洗い出したりして、対処のスコープを検討する。	●スコープを検討することで、対処の有効性（限界）を理解した上で、対処の採用・非採用を判断できる。 ●類似原因による問題への対処漏れ（水平展開の漏れ）が減少する。

● 施策①と②を実施した場合のプロセス



● 施策③と④を実施した場合のプロセス（プロセスを新規追加）



● 施策で作成する成果物の3点セットの概要と期待効果（1／2）

- 成果物の位置づけ

成果物名称	カバレッジ	記述内容	対応すると思われるXDDP成果物
対処検討資料	Why What How	なぜ変更するのか？ 何を変更するのか？ どのように変更するのか？	変更要求仕様書
トレーサビリティ・リスト	Where	どこを変更するのか？ 上流工程の変更が、下流工程のどこに反映されているのか？	トレーサビリティ・マトリクス
不足ドキュメントを補う各種設計書（1.で作成）	How	現状はどのような動作になっているのか？	スペックアウト資料
各種設計書、ソースファイル（2./3.で作成）	How	どのように変更するのか？	変更設計書

● 施策で作成する成果物の3点セットの概要と期待効果（2／2）

- 成果物の概要と期待効果

成果物名称	概要	期待効果
対処検討資料	●対処内容について、上流から下流工程までを調査した結果を記述し、対処内容の妥当性を証明する文書。 ●対処案を作成し、その案の妥当性を検証するには、どんなことを調査する必要があるのかを列記し、1つ1つ調査した過程と結論を明記する。 ●調査の過程で既存処理の動作仕様が不明であるなど、ドキュメントが不足している場合は、新規に作成して補完する。	●対処案を実現するにはどんなことを調査する必要があるのかをゼロベースで検討できるので、検討漏れや検討誤りを指摘しやすい。 ●検討結果が変更になったとしても、結論を導いた過程が明確なので、遡ることで変更が他に与える影響をトレースできる。 ●上流から下流までの既存ドキュメントの不足を補える。 ●結果的にレビューで理解しやすいドキュメントとなるため、上流工程で品質を作り込みやすい。
トレーサビリティ・リスト	●上流の変更内容が、下流の成果物のどこに反映されているのかをトレースできる表形式の文書。	●レビュー時に変更の漏れや誤りに気付きやすい。 ●変更箇所を漏れなくレビューできたのかを確認しやすい。 ●次工程の着手前にTLを作成することで、変更規模を見積ることができ、計画的な作業が可能になる。 ●別の担当者に作業を引き継いでも変更箇所をトレースしやすい。
不足ドキュメントを補う各種設計書（1.で作成）	●対処検討資料を作成するために必要な既存ドキュメント。	●リバースエンジニアリング等を行いながら、不足しているドキュメントを補えるので、レビュー効率化、今後の保守の効率化が図れる。
各種設計書、ソースファイル（2./3.で作成）	●各工程で作成する成果物	●今までと位置づけは同じ。

● 施策で解決を目指すこと

施策①
既存の実装がどうなっているのかを早期に把握した上で、対処を検討する

施策②
既存ドキュメントだけでは動作仕様が明確でない場合は、新規にドキュメントを起こす

施策③④
対処内容ありきで変更箇所を探すのではなく、対処案（仮説）を調査しながら検証していくというアプローチを取る

施策③④
仮説を検証するために必要な調査アイテムを、自分の理解度に応じてゼロベースで洗い出す

施策②
変更する仕様がどのように設計⇒ソースへと落とし込まれていくのかをトレースできる

早期に母体の実装方式や、母体欠陥を摘出でき、計画的なスケジューリングが可能になる

第三者によるレビュー理解を促進し、レビュー効率を上げる

「対処案は本当に妥当か？」というクリティカルシンキングが行いやすいプロセスなので、仮説の検証が促進される

自分の理解度をベースに、調査作業を漏れなくピックアップでき、調査漏れの防止、計画的スケジューリングを可能にする

変更が下流工程にどのように反映されているのかを追跡可能であり、変更漏れを防止できる

施策の限界

● 施策の効果に関する疑問点や限界

- 疑問点や限界

疑問点	回答
対処検討資料を作るために、既存のドキュメントに不足している内容も作成するので、とても工数がかかるように思える。	●本プロセスを採用すればこれまで以上の工数を必要とする。 ●上流工程が一番重い作業となり、下流工程に行くに従って（事前に検討しているために）、少ない工数で対応できるようになる。 ●テスト工程での欠陥摘出による手戻りを最小限に食い止めることで、トータルの工数削減を図るのが狙い。 ●また上流でおおよその作業規模感やリスクを把握できるので、計画的なスケジューリングが可能である点もメリットである。
変更対処内容を証明するために調査するときの、有効な手法はあるか？	●マトリクス表を用いて網羅的に調査することが最も有効な手段。 ●このあたりは幾つかの手法を提示可能。（今後対応する）
「ストーリーのある対処検討資料」というコンセプトだが、具体的にはどのように作成したらよいのか？	●決まりきった型があるわけではないが、検討を進めるプロセスや、検討すべき内容のリストアップはできる。（今後対応する）
影響範囲を漏らさないようにするための工夫はどのあたりか？	●1つ目はフォワードローディングを行うという点。 ●2つ目は対処検討資料を充実させて、ロジカルに構成することで、検討の漏れを減らす、というアプローチ。問題対処ありきで変更する箇所を調べるアプローチではなく、対処案という仮説を検証するアプローチで、ゼロベースでの検討を可能にする。 ●3つ目は、対処検討資料の作成におけるスコープ定義によって、要求コンセプトを明確にすることで、水平展開の漏れを予防する。