アジャイル開発におけるテスト分析による高速フィードバックプロセスの提案
～T-USDM による要求仕様の改善～

派生開発推進協議会 T4+T6研究会
株式会社ベリサーブ　堀川 透陽

背景 ～組み込み開発におけるテスト

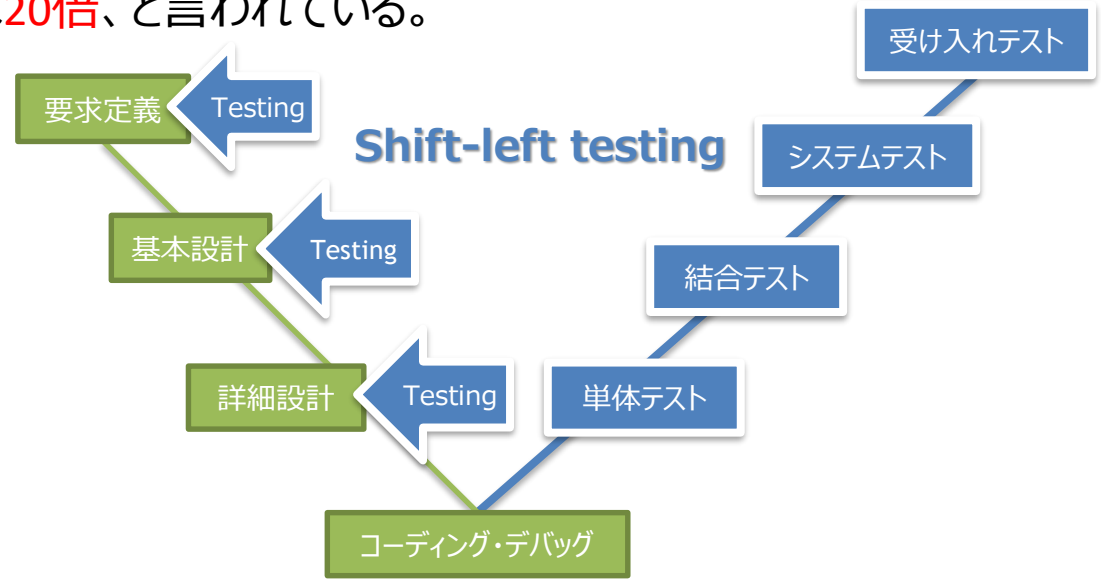
組み込み開発におけるテスト事情として、専門性と第三者の視点を持つテストベンダーを入れることで、
品質をテストから上げていくというケースがある。

そうした第三者検証のアプローチでは、様々なテスト技術やテストツールが駆使され、
より多くの不具合を検出するようになるが、反面、不具合の修正コストの増加も課題となる。

不具合の修正コストは**開発工程の早い段階であればあるほど抑えることができる**ことが知られている。
要求仕様での誤りを1とし、設計段階では5倍、テストによる検出では20倍、と言われている。

このコストの課題を解決すべく、
不具合を早い段階で見つけることを目的とし、
テスト活動を前倒しする**シフトレフトテスト**(Shift-left testing)の
アプローチが取り入れられ、浸透しつつある。

一方で、開発手法の主流が**アジャイル開発**に変化してきており、
テスト活動についても、手法に合わせたアップデートが求められている。



背景 ～アジャイル開発における要求仕様

AFFORDDでは、XDDPを活用したアジャイル派生開発において、ユーザストーリーを変更要求に落とし込む**A-USDM(アジャイルUSDM)**がT6研究会(「Agile開発との連携」)から発表された(2018年)。

図：A-USDM説明

特徴まとめ

- 「振る舞いの観点」のテンプレートに沿って要求を記述
- 観点に沿った要求記述全体を「上位要求」、観点の1つ1つを「下位要求」と捉え、要求の階層を表現
- １つ１つの下位要求に対して仕様を記述できる

ラベル　振る舞いの観点　上位要求　⟷ 下位要求を1文にすると、上位要求になる

ラベル	要求仕様	(だれが)どんな状況で	(だれが)どんなとき	(だれが)どうなったら	(だれに)どうなる	理由	説明
		荷物の集配依頼を受けて、担当地域の配達員の端末に住所や氏名等の情報を送る					
集荷依頼	要求	オペレータが集荷受付中	配達員が配達しているとき	オペレータが荷物の集配依頼を受けたら	担当地域の配達員の端末に住所や氏名等の情報を送る	配達中の配達員に情報を送ることで、配達ルートの中で荷物を集めたい	依頼者からの電話を受けて、集荷先の氏名や住所、メールアドレスなどを聞き出して入力したデータが画面上に表示している
		下位要求	下位要求	下位要求	下位要求		
	仕様			<集荷情報入力画面の表示> <担当配達員の検索> <配達員情報の表示>	<集荷指示データの編集> <集荷担当者への送信>		
				仕様	仕様		
		荷物に貼られた配送シートの情報を読み取って、配送データを生成して配送DBに新規登録する					
荷物受取	要求	配達員が配達中	集荷指示を受け取った後	配達員が荷物を受け取ったら	荷物に貼られた配送シートの情報を読み取って、配送データを生成して配送DBに新規登録する	一旦受け取った荷物は、できるだけ早くシステムに登録しておきたい	
	仕様						

各下位要求に対し、仕様を記述できる

※参照 「モデルベース開発におけるアジャイルXDDPの適用～スクラムXの提案と実践」
https://affordd.jp/wp-content/uploads/conference2018/affordd_conference2018_p5.pdf

背景 ～アジャイル開発におけるテストの現状

アジャイル開発におけるテストのガイドラインとしては『アジャイルテストの4象限』が知られている。
このQ1～Q4のバランスを取るということは、要求と仕様がV&V(検証と妥当性確認)の視点で確認されていることを意味する。

アジャイル・テストと役割分担

<Q1>
主に開発メンバーが実施
<Q2>
開発メンバー or QA
<Q3>
主にQAが実施
<Q4>
QAが実施？
または専門スキルを有する外部に委託するなど

※テストと絡めた開発手法

<単体テスト>
- TDD(テスト駆動開発)

<単体～結合テスト>
- BDD(振る舞い駆動開発)

<受け入れテスト>
- ATDD(受け入れテスト駆動開発)

課題と解決方針

これまでの背景を踏まえ、派生開発／アジャイル開発におけるテストをテーマにした課題と、解決方針を以下に策定する。

- ❖ アジャイル派生開発におけるQAのプロセスを、テストの視点から深く考えたい
- ❖ これまでのシフトレフト・テストはアジャイル開発にどう適用すべきか
- ❖ シフトレフトを適用した場合、アジャイル開発における品質の作りこみはスピードに影響しないか

<考察>

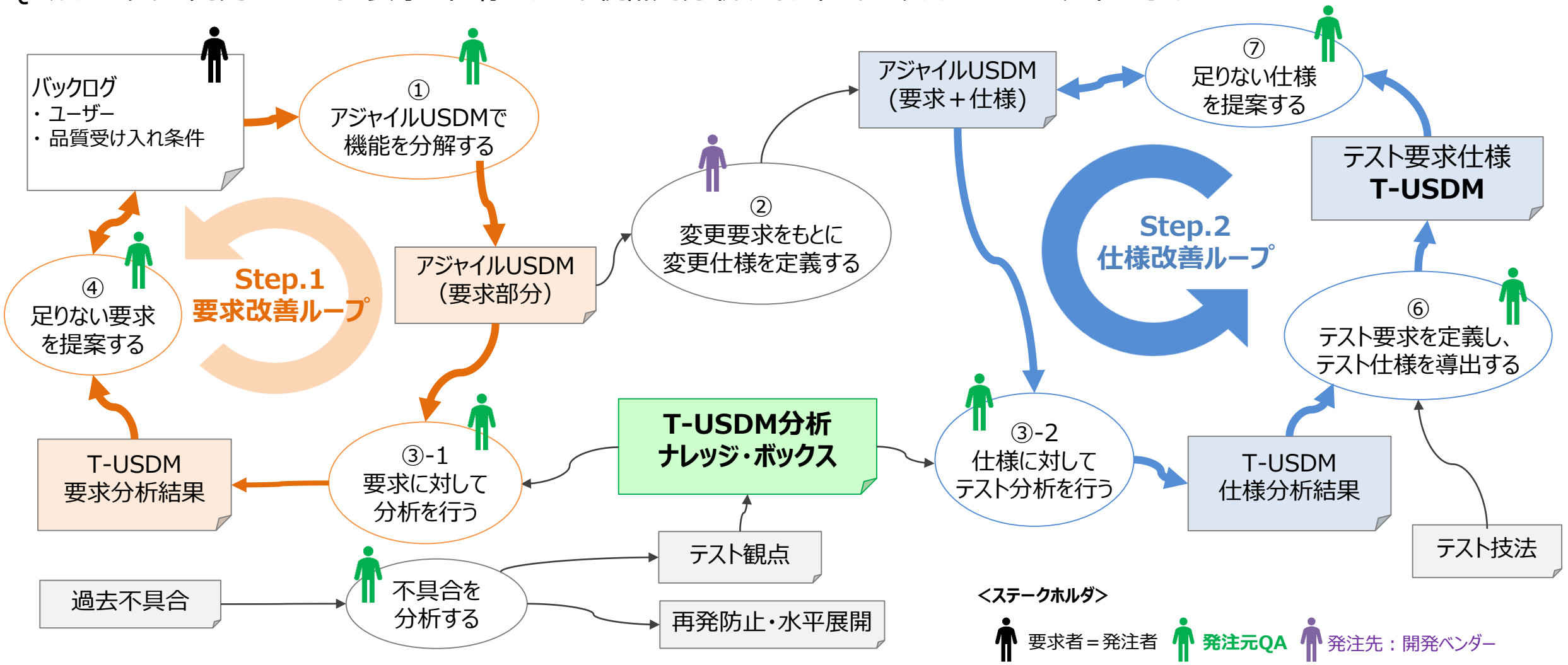
- ❖ 品質の作りこみが甘い＝潜在バグを抱えるシステムは、常に不具合を生み出すリスクを抱えている
  - ➡ ファーストリリースは早くても、いずれのリリースでしわ寄せ＝スピードの低下が来てしまう
- ❖ アジャイル開発においても不具合はいずれ対応しなくてはならない
  - ➡ 不具合の素は事前に潰しこんでおくに越したことはない

- ❖ QAが要求仕様に対してテスト視点でレビューすると、不具合を根元から断つことができるのではないか
  - ➢ 要求分析にはA-USDMを活用する
  - ➢ テスト視点を引き出すための仕掛けも必要

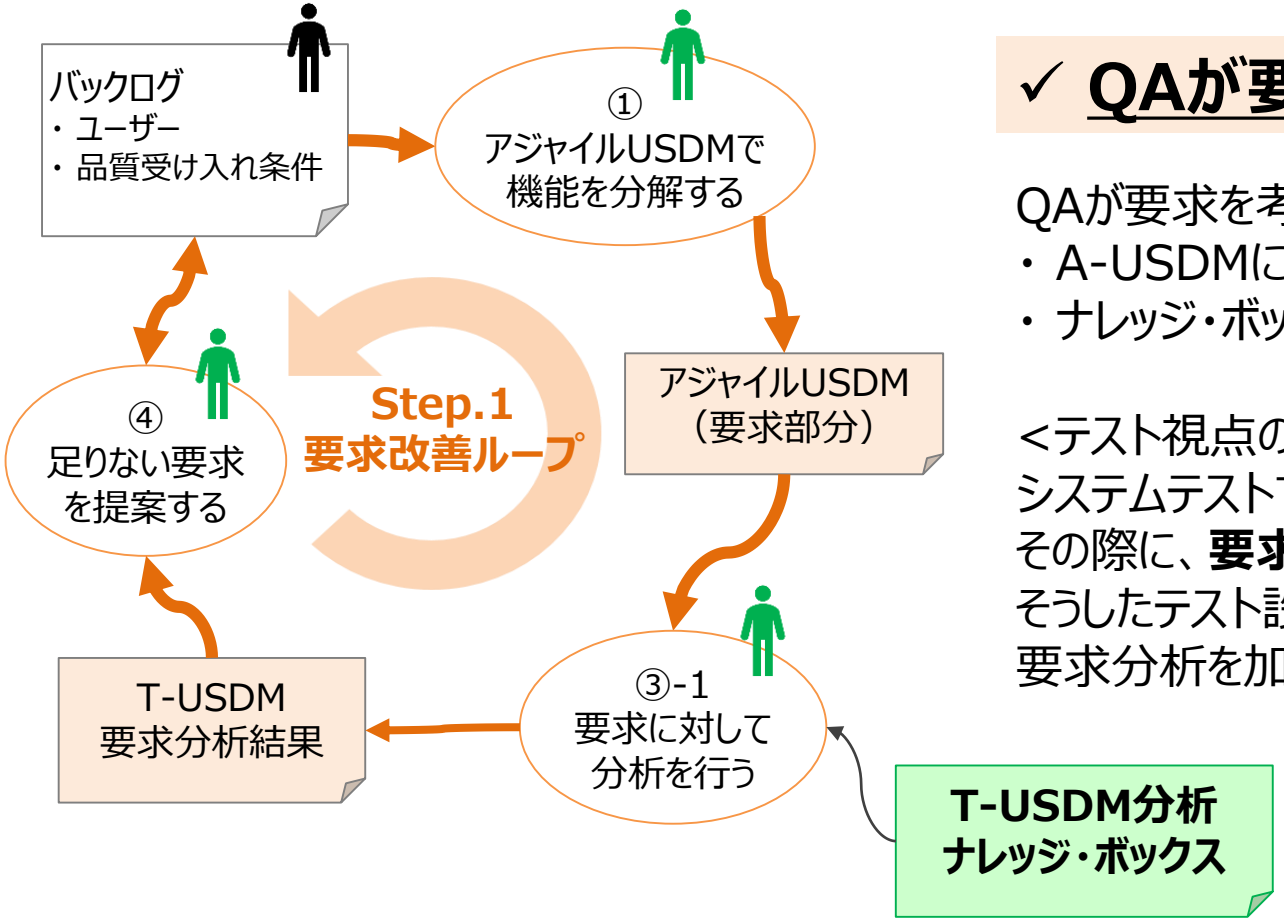
要求と仕様のフィードバックプロセス

T-USDM(Test USDM)
テスト要求仕様をUSDMで整理し、記述したもの

QAがアジャイル開発において、要求と仕様をテスト視点で分析するフィードバックプロセスを以下に示す。



要求改善ループの特徴

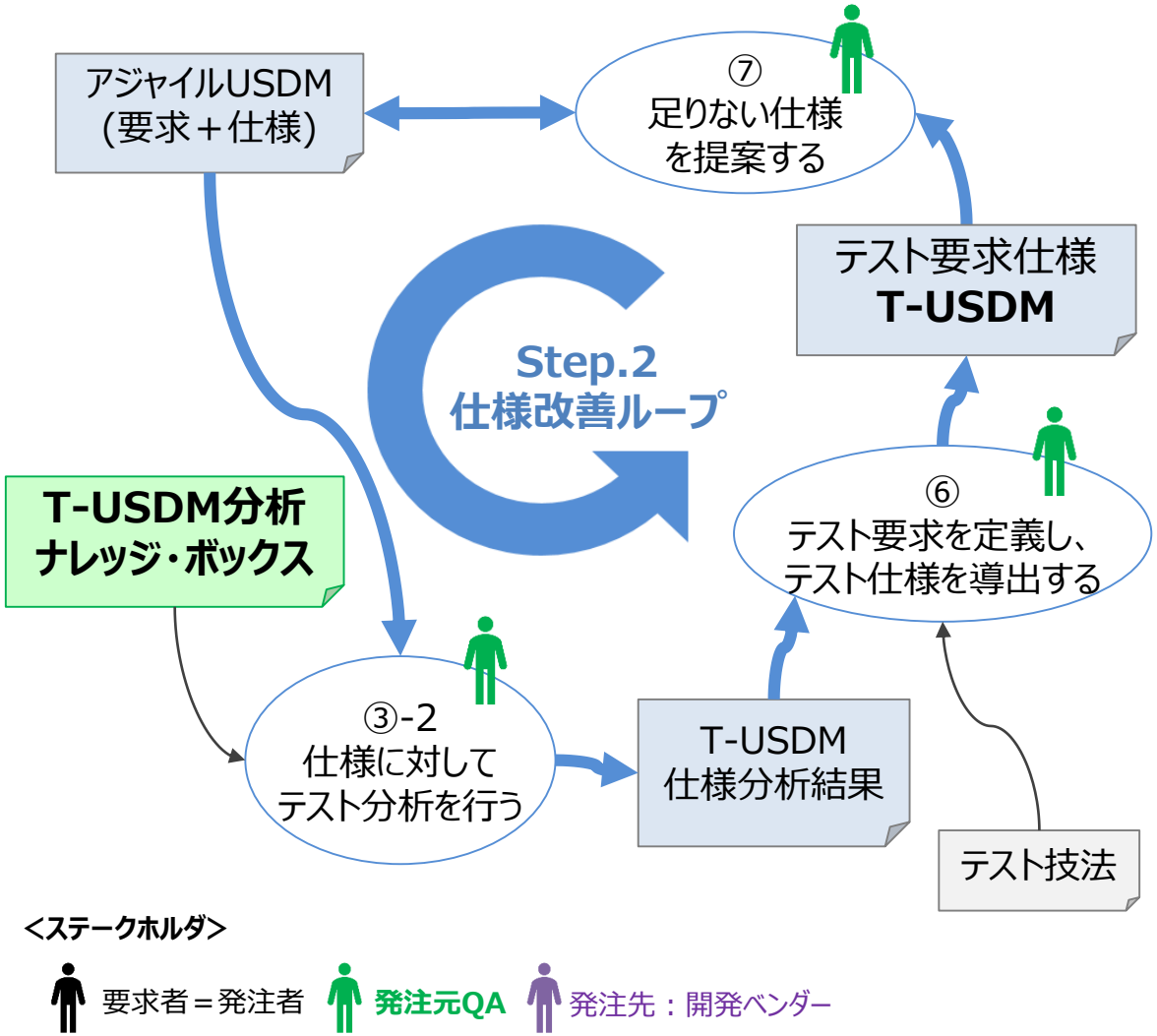


✓ QAが要求を考えつくすプロセス

QAが要求を考えつくすため、2段階の分析を用意する。
- A-USDMによるユーザストーリーの落とし込み
- ナレッジ・ボックスによるテスト視点での要求の補完

<テスト視点の分析について>
システムテストでは、要求の記述を基にテストシナリオを考えることがあるが、
その際に、**要求の記述から条件やパターンを広げている**ことが多い。
そうしたテスト設計における思考をナレッジ・ボックスにまとめておき、
要求分析を加えることで、足りない要求を補完できる。

仕様改善ループの特徴



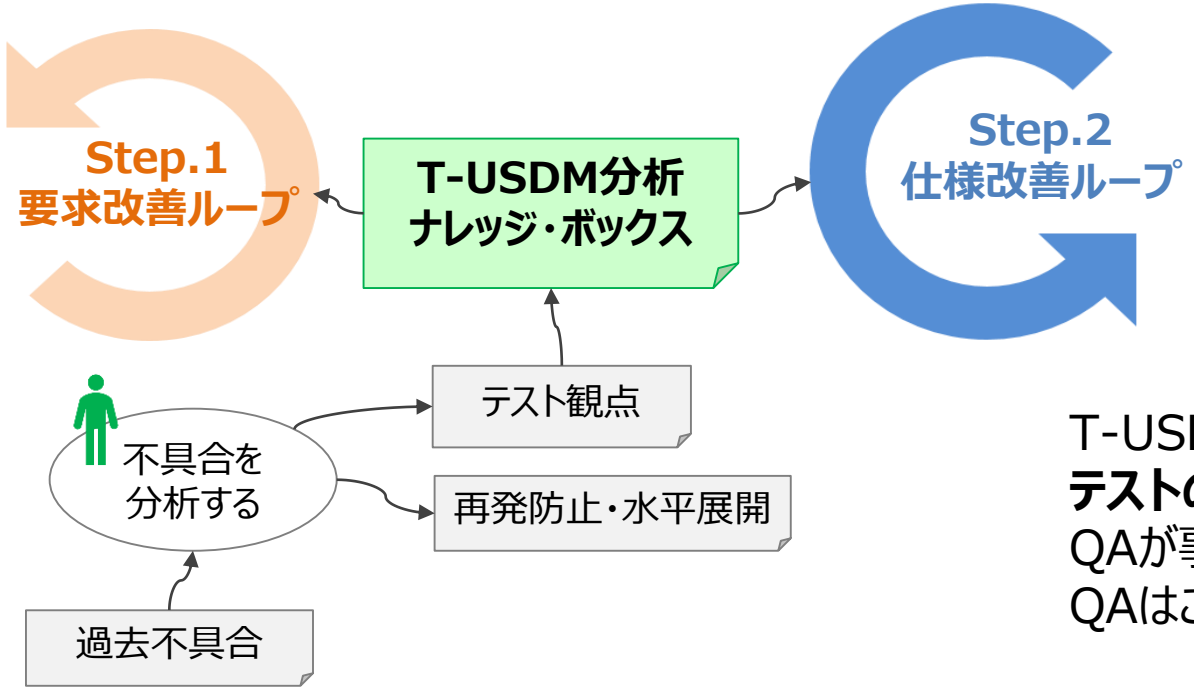
✓ QAがテスト視点で仕様を補完する

このプロセスでは、QAが変更仕様に対し**テスト視点で分析**を行い、足りない仕様を補完する。

テスト技法や、過去不具合の傾向をナレッジ・ボックスにまとめておき、変更仕様に対するテスト要求仕様をT-USDMとして記述し、レビューを実施する。

開発メンバーはテスト要求仕様の記述に対して、仕様の考慮漏れがあると事前に対策を取ることができるため、その後のテスト実施では**不具合は発生せず、修正コストが削減できる。**

ナレッジ・ボックス(knowledge-box)



<ステークホルダ>

要求者＝発注者　発注元QA　発注先：開発ベンダー

✓ QAの頭脳～知と技の集約

T-USDM分析 ナレッジ・ボックスは、
テストの知見と経験を集約し整理されたドキュメントで、
QAが事前に準備しておく必要がある。
QAはこのナレッジ・ボックスをプロジェクトを横断して使用する。

以下の情報をチェックリストやモデリングで整理しておく。
- システム特有のテスト観点
- 過去不具合から、不具合を多発させるテスト観点、条件
- シナリオのアンチパターン
- ラルフチャートや状態遷移テストといったテスト技法のアプローチ

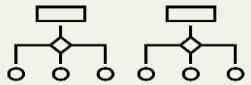
ナレッジ・ボックス(knowledge-box)

ナレッジ・ボックスは**テストの知見・経験を集約した共通ドキュメント**で、QAにおる要求仕様の分析をサポートするツールである。
プロジェクト／スプリントの終了後は、レトロスペクティブと併せて内容をブラッシュアップする（＝**ナレッジ・ボックスは成長し続ける**）。

Step.1 要求分析 ・・・ 変更要求が出てきた際に実施するバリデーションチェック

Who	When	Where	What	How
ユーザ／システム ☑ アクターはすべて抽出したか？ ☑ 権限の違いを考慮しているか？ ☑ 悪意のあるユーザを想定しているか？	**時／タイミング／順序** ☑ 動作直前の状態を抽出しているか？ ☑ 動作のタイミングを考慮できているか？ ☑ 前後の動作を考慮できているか？	**場所／空間／シチュエーション** ☑ 環境(HW/SW)を考慮しているか？	**タスク／サービス** 仕様で記述する項目	**手段／状態**

Step.2 仕様分析 ・・・ 変更仕様が出てきた際に実施するテスト分析

要求ベースのアプローチ	テスト技法アプローチ	テスト観点アプローチ
<Who> ☑ 権限のないユーザによるアクセス **<When>** ☑ 操作の順序を入れ替える **<Where>** ☑ 旧バージョンでの動作	**ラルフチャート** ☑ 入力パターンの組み合わせ ☑ 動作中の割り込み処理 ☑ 動作中の悪意のある割り込み：意地悪テスト観点 **状態遷移** ☑ ダイレクトアクセスで遷移する ☑ 入力後のロールバック	**テストモデル：**NGT、クラシフィケーションツリーなど  **過去不具合の分析からのインプット** ☑ ODC分析から得た設計モレの傾向

T-USDM(Test USDM) によるテスト要求仕様

ナレッジ・ボックスにより得られたテストアプローチやテスト観点をもとに、テスト要求とテスト仕様をUSDM形式で作成する。

■ **T-USDM記載例**

Pot-273-T01	タイマ沸騰機能の状態は常に正しく示される。	【テスト要求】 テストで証明したいこと
	Pot-273-T01:01	タイマ沸騰中は[wait]が点滅し続ける。
	Pot-273-T01:02	タイマ沸騰中に水位センサがoffしても[wait]が点滅し続ける。 【コメント】仕様ではそうなるが、ユーザには優しくない
Pot-273-T02	タイマ沸騰機能はポットの状態に影響されない。	
	Pot-273-T02:01	沸騰中の、沸騰タイマ機能の設定([沸騰]＋[タイマ]キー同時押下)はできない(反応なし、とする)。
	Pot-273-T02:02	タイマ作動中に沸騰タイマ機能を設定すると、元のタイマがリセットされ、沸騰タイマは指定時間後に沸騰動作を開始する。
Pot-273-T03	タイマ沸騰待ち状態の割り込み操作は沸騰機能に影響しない。	
	Pot-273-T03:01	タイマ沸騰待ち状態で[沸騰]キーを押下すると、沸騰行為が開始されるが、沸騰タイマ機能は解除する。
	Pot-273-T03:02	タイマ沸騰待ち状態で[給湯]キーを押下すると、給湯は行われるが、水位メータが第一水位センサ以下になると、タイマは作動しない。

【テスト仕様】テスト要求を満たすために
必要な実行内容＝テストケース

QA/テストと開発の対話

QAが作成したT-USDMは、テスト実行前に開発とレビューを行い、これにより考慮が不足していた仕様が補完される。
これは言わば、**テストケースを事前公開**することであり、テストチームは通常しない行為である。
しかし、アジャイル開発においてはスピードを重視している。不具合による手戻りを防ぐためには、この手段が最も効率的だと考える。

アジャイルソフトウェア開発宣言

私たちは、ソフトウェア開発の実践
あるいは実践を手助けをする活動を通じて、
よりよい開発方法を見つけだそうとしている。
この活動を通して、私たちは以下の価値に至った。

プロセスやツールよりも**個人と対話**を、
包括的なドキュメントよりも動くソフトウェアを、
契約交渉よりも顧客との協調を、
計画に従うことよりも変化への対応を、価値とする。

すなわち、左記のことがらに価値があることを認めながらも、
私たちは右記のことがらにより価値をおく。

“個人と対話”について、
テストの視点で、開発と対話があっただろうか？

仕様書に書いていない条件を実行し、
不具合が見つかったことはないか？
なぜそれは事前に開発と対話しなかったのか？

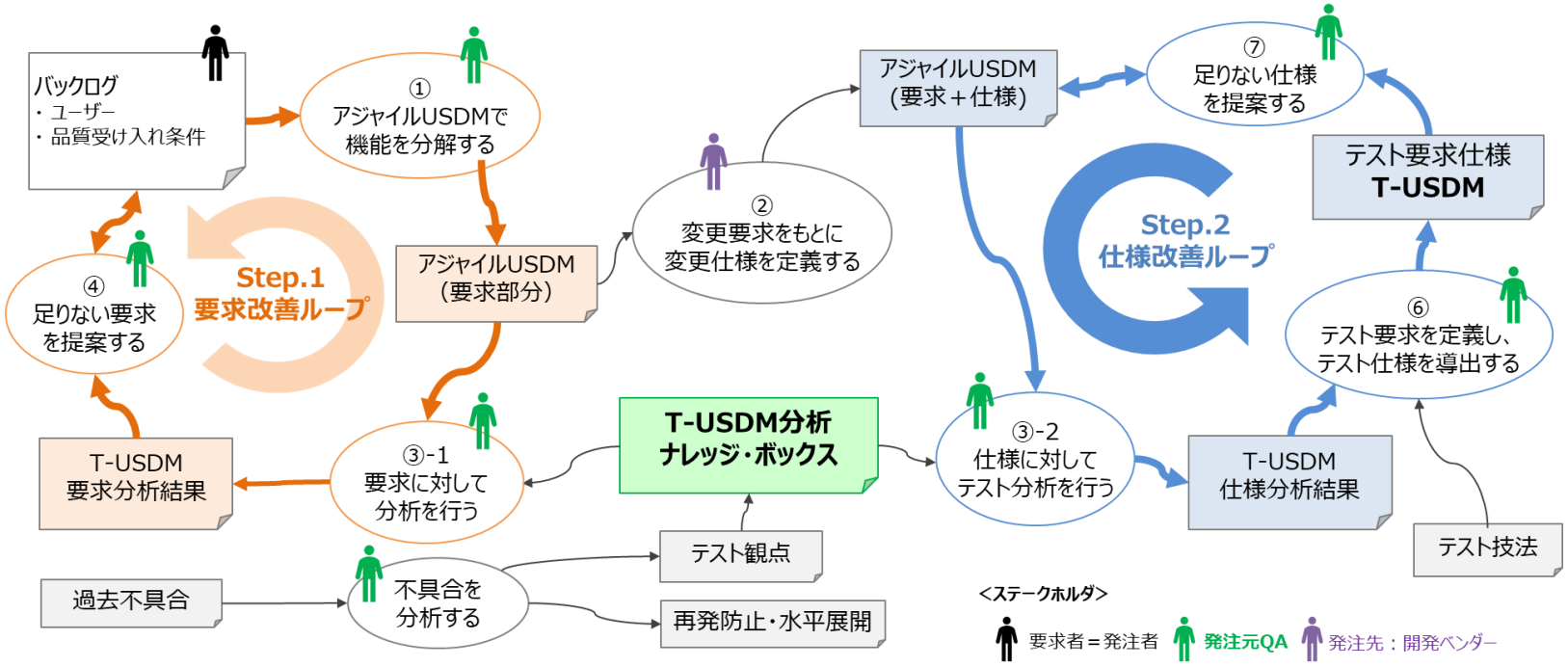
不具合を出すことが目的となっていないか？
本当に重視すべきは何か？

まとめ

アジャイル派生開発におけるQAは、要求仕様をテスト視点でレビューすることで不具合の出ない品質を作る。

要求と仕様のフィードバックプロセス

- ✓ QAは事前にテスト観点や過去不具合の傾向をまとめた**ナレッジ・ボックス**を用意する
- ✓ 要求改善ループでは、ユーザ視点(**A-USDM**)とテスト視点(ナレッジ・ボックス)で要求を考えつくす
- ✓ 仕様改善ループでは、テスト視点(**T-USDM**)でテストパターンを仕様として実装する



期待される効果と課題

❖ **最初から品質の高いリリースが期待できる**
要件と繋がるシステムテスト、要求と繋がる受け入れテストは、
要求改善ループの効果により、基本的に不具合は発生しない、ただの確認作業となりコストカットとなる。

❖ **テストによる”過剰品質”をコントロールできる**
事前にテスト仕様を開発とレビューするため、重要度の低いテスト
(重箱を隅をつつく様なテスト、レアケースの条件)は事前に除くことができる。
スピードと品質のトレードオフをコントロールできる。

❖ **不具合の件数が成果ではなくなる**
不具合を出さないことを目的としているため、不具合件数はメトリクスとして使えない。
QAの成果をメトリクスでどう判断するかは検討が必要。

本発表の共同執筆者：T4,T6研究会メンバー

本発表にあたり、提案及び内容のディスカッションに関わった研究会メンバーを以下に記載する。

T6研究会 「Agile開発との連携」

- 永田敦（サイボウズ）
- 斎藤賢一（エクスモーション）
- 本田英稔（構造計画研究所）
- 葛西孝弘（NECプラットフォームズ）
- 伊藤建一（クレスコ）
- 久保宏志
- 工藤寛（NECプラットフォームズ）
- 中村勝志（アンリツ）
- 星野充史（アンリツ）
- 加藤康之（アンリツ）
- 松山輝樹

T4研究会 「XDDP」とテストプロセスとの接続

- 堀川透陽（ベリサーブ）