

T4研究会「XDDPとテストプロセスの接続」 研究活動10年の軌跡とAI駆動開発への展開

派生開発推進協議会(AFFORDD)
T4研究会 堀川 透陽

2026/7/3 ソフトウェアテストシンポジウム2026関西 (JaSST' 26 Kansai)
AFFORDD+JaSSTコラボセッション
「要求とテストのMIRAI～生成AI時代の開発における要求とテストの新しい関係」

なぜT4研究会が生まれたか

「XDDP(派生開発プロセス)ではテストプロセスについて詳しく言及されていない」

XDDPの3点セット が生む障壁

3点セットとは・・・
変更要求仕様書(USDM)
/ TM / 変更設計書

- 差分情報がテスト技術者に理解しづらい
- 影響範囲が特定しにくい
- 正式文書更新が開発後半に集中
- テスト分析用の中間成果物がない

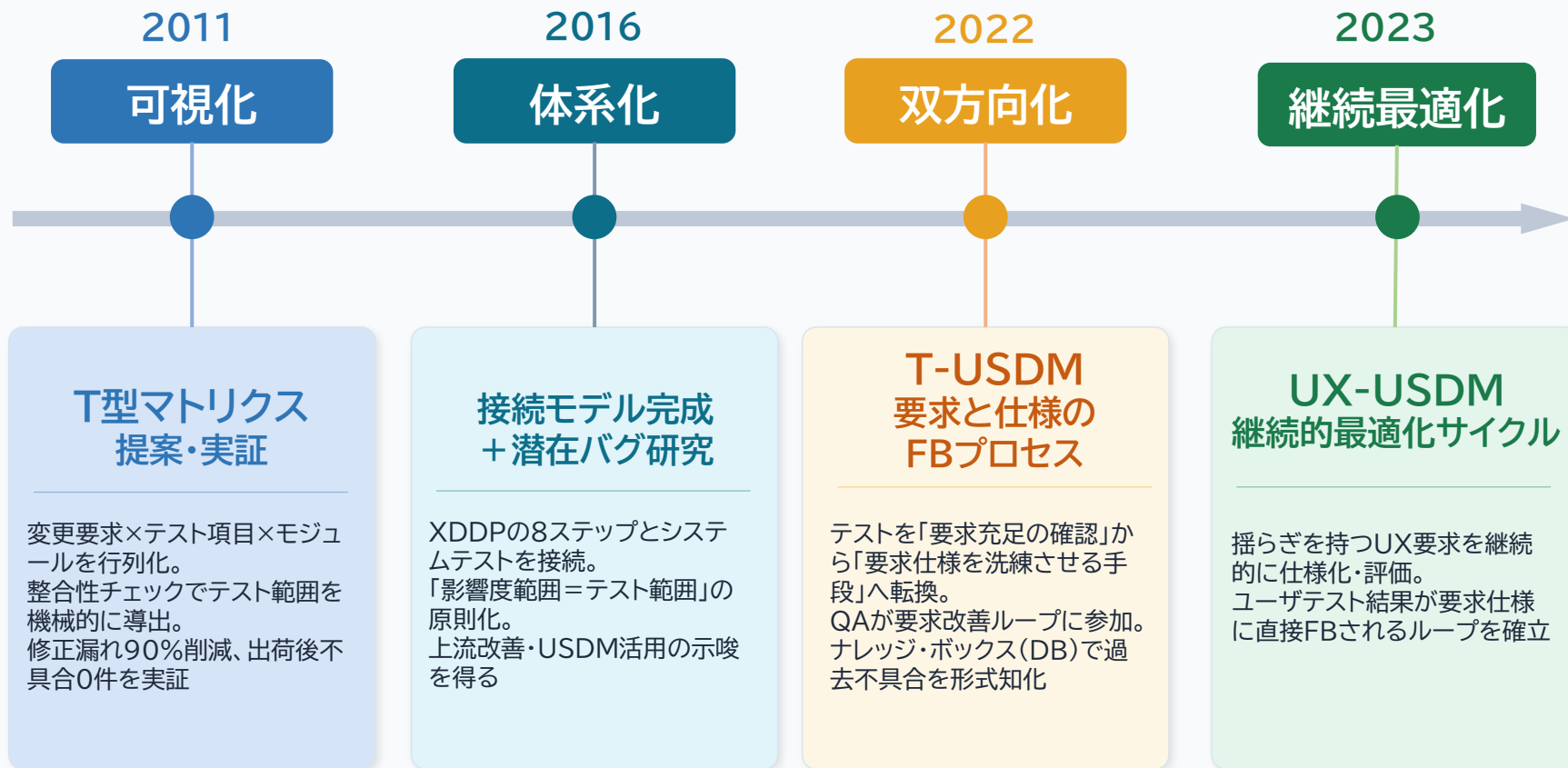
収集された不安・課題

- 【もの】変更点が見える化されていない
ドキュメントに反映されない
- 【やり方】開発側が変更の影響を把握できていない
テスト技術者が影響を説明できない
- 【人・組織】開発チームとテストチームの間に
壁がある
- 【全体】アジャイル開発でのテストの実行タイ
ミングが分からない

T4研究会のコンセプト

- 差分情報とテストの関係を可視化する
- 開発者とテスト技術者の共通言語を作る
- テスト分析～設計をXDDPに組み込む
- 上流からの改善でバグを根絶する

「要求とテスト」の関係モデル ～10年の変遷



「要求とテスト」の関係モデル ～10年の変遷

2011

可視化

T型マトリクス 提案・実証

変更要求×テスト項目×モジュールを行列化。
整合性チェックでテスト範囲を機械的に導出。
修正漏れ90%削減、出荷後不具合0件を実証

T型マトリクス用いた変更要求の提案

(2011年)

■T型マトリクスを用いた(要求—機能—構造)の明確化

「列」に開発要求(左)、モジュール(右)および「行」に機能項目(=テスト項目)を配置することにより、機能との関連を明確にし、開発者とテスト技術者の認識を合わせる

今回開発で対応する開発要求
(開発者視点)

変更要求仕様書のTMに同じ
(開発者視点)

開発要求1	開発要求2	開発要求3	開発要求4	..	開発要求	モジュール	モジュールA	モジュールB	モジュールC	モジュールD	モジュールE	..	モジュールX
○					機能A	●		●					
	○	○			機能B		●			●			
	○				機能C		●				●		
					...								
			○		新規機能X	●					●		
				○	新規機能Y					●			●

2つのマトリクスを結合

①開発要件—機能マトリクス
※製品仕様上の関連を把握

②機能—モジュールマトリクス
※ソフト構造上の関連を把握

既存のテスト項目から抽出した機能項目 (テスト技術者視点)

「要求とテスト」の関係モデル ～10年の変遷

2022

双方向化

T-USDM 要求と仕様の FBプロセス

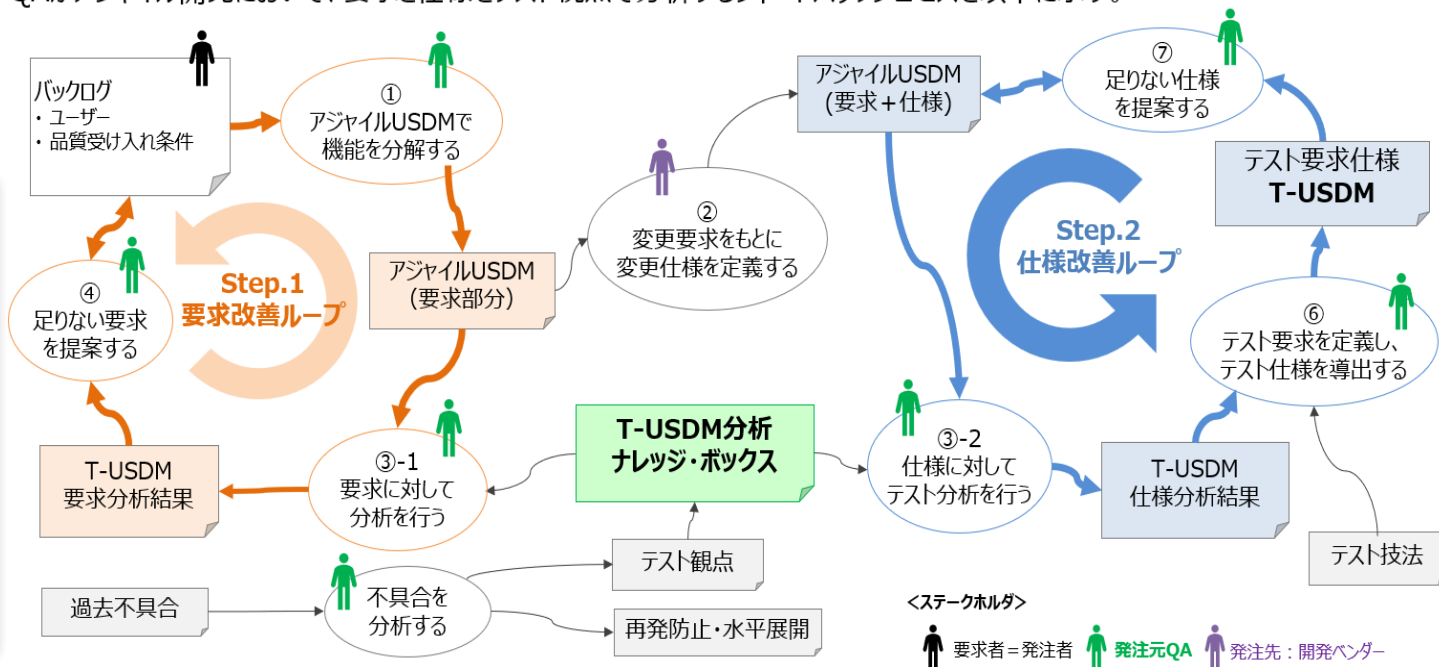
テストを「要求充足の確認」から「要求仕様を洗練させる手段」へ転換。
QAが要求改善ループに参加。
ナレッジ・ボックス(DB)で過去不具合を形式知化

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。



「要求とテスト」の関係モデル ～10年の変遷

2023

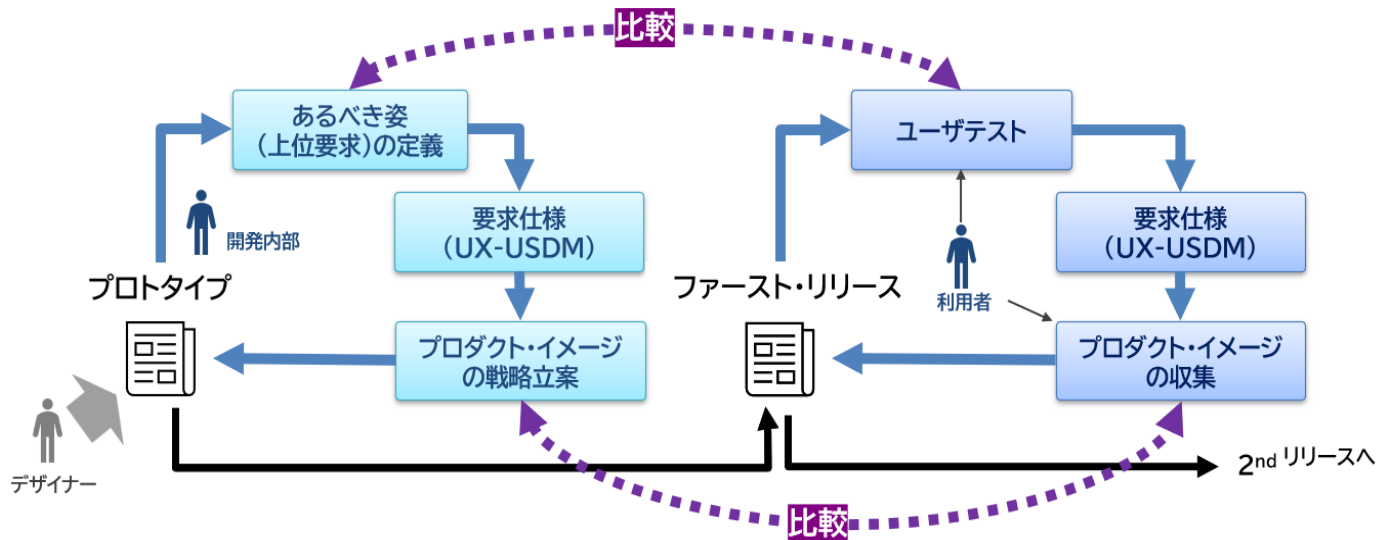
継続最適化

UX-USDM 継続的最適化サイクル

揺らぎを持つUX要求を継続的に仕様化・評価。
ユーザテスト結果が要求仕様に直接FBされるループを確立

ユーザビリティを最適化し、UXを評価するプロセス(全体)

ユーザビリティ/UXの品質を作るプロセスにはウォーターフォールモデルは不向きであるため、アジャイル開発を前提としたプロセスを検討する。
また、本提案のプロセスは「ファーストリリースまで」と「ファーストリリース以降」に分けて説明していく。



「要求とテスト」の関係変化(Before/After)

トレーサビリティ
による対応付け

Before

テストは「要求充足を確認する後続活動」

After

変更要求→TM→テスト項目→テストケース
の一方向トレーサビリティを確立

▶ T型マトリクス: 整合性チェックでテスト修正漏れを機械的に検出

双方向
フィードバックへ

Before

「テストは要求仕様を検証するもの」

After

「テスト視点は要求仕様を完成に導くもの」
QAが要求改善ループの参加者に

▶ T-USDM: 要求とテストを同一フォーマット(USDM)上に統合。二重ループ構造

継続的
最適化サイクルへ

Before

「要求は一度完成したら固定」という前提

After

テストの役割は、
要求を「更新し続ける」パートナーとなる

▶ UX-USDM:
要求(あるべき姿)をテスト観点として、リリース毎にユーザテスト結果を要求にFBする閉ループ

「要求とテスト」の研究に共通する4つの原則

01 要求はテスト視点により補完される

テスト分析(Who/When/Where/How)を通じて
要求の抜け漏れが明らかになる

02 仕様化(形式化)が接続の鍵

T型マトリクス・T-USDM・UX-USDM
いずれも「形式化・構造化」によって
要求とテストを繋いでいる

03 ナレッジの組織資産化

過去不具合の活用(2016年)→
ナレッジ・ボックス(2022年)へ。
属人知をチーム共有可能な形式知に変換

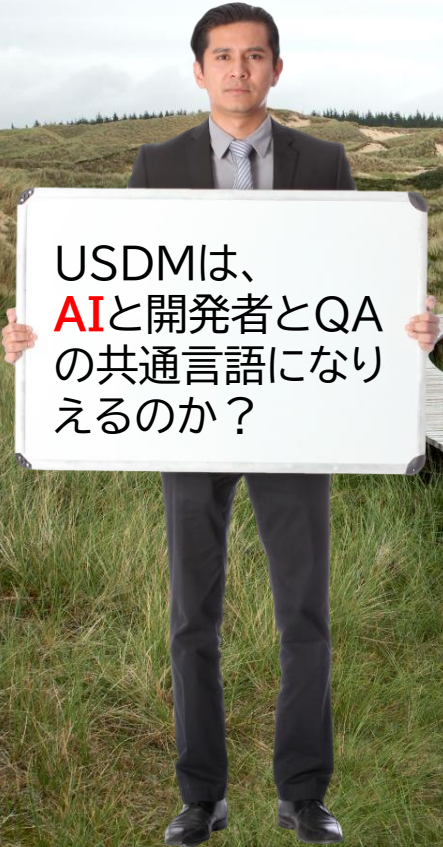
04 テスト活動の上流へのシフト

テスト活動の起点を
下流(実行)→上流(要求レビュー)へ
継続的に引き上げることが一貫した方向性

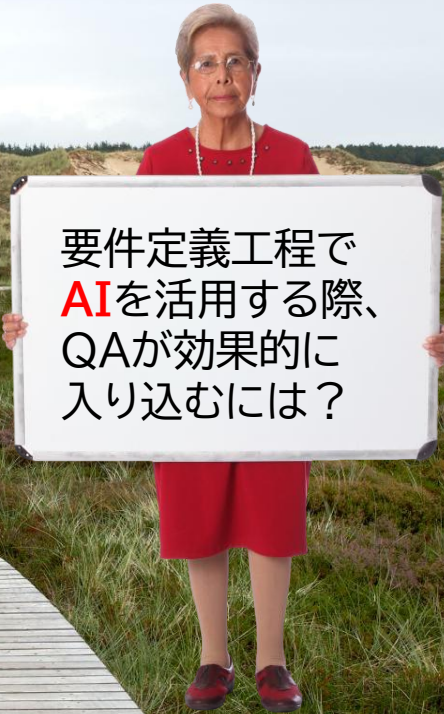
「要求とテスト」とAI適用

研究会アプローチ	変化の方向性	AI適用による変化
T型マトリクス 整合性チェック	→ 自動化	AIエージェントによる仕様書の抜け漏れ自動検出が 実用段階。
T-USDM:ナレッジDB 過去不具合の形式知化	→ 強化	GitHubへの成果物集約+AIの自動検索・参照。 「過去不具合の活用」がRAGナレッジベースとして実 装可能
T-USDM:要求改善ループ テスト視点での要求補完	→ 加速	AIエージェントによる仕様レビュー(Figma MCP等)。 Who/When/Where観点の自動チェックが実用段階。 最終判断は人間
T-USDMの構造 (テスト要求→テスト仕様)	→ 協調	T-USDMの構造がAIへの指示書(プロンプト構造) と高い親和性。テストケース自動生成のINPUTとし て活用可能
UX-USDM 継続的最適化サイクル	→ 部分代替	プロダクト・イメージのモニタリング自動化は可能。 ただし「UXのあるべき姿」の定義(上位要求)は人

ディスカッションは続く —————



USDMは、**AI**と開発者とQA
の共通言語になり
えるのか？



要件定義工程で
AIを活用する際、
QAが効果的に
入り込むには？