

開発プロジェクトにおけるテスト マネジメントの基本と実践

13:35-14:35 25th Jan. 2019 JaSST'19 Hokuriku

湯本剛

自己紹介 湯本剛 (@yumotsuyo)

□経歴

1991年 工作機器メーカーで経理部に所属し、生産管理システムの受入テストに従事

1995年 メーカー系列ソフトハウスにてテストリーダーとしてアプリ開発に従事

- 財務会計のパッケージソフト、プリンタ、スキャナ、製造業受託開発（在庫管理、生産管理など）、顧客向けプロダクト販売Webサイト構築など

2003年 豆蔵にて、ソフトウェアテストコンサルタントとして、テストプロセス改善コンサル、テストツール導入支援、テスト教育に従事

2010年 日本ヒューレット・パカード(HP) ソフトウェア部門にて、ALM製品のプリセールスに従事

2014年 日本HPのサービス部門にて、デリバリー案件のテストマネジメント、QA組織立ち上げのコンサルティング&デリバリー

□現在（2019年1月）

ユーザー企業 開発部門のテスト専門部署

NPO法人 ASTER：ソフトウェアテスト技術振興協会 理事

JSTQB(認定ソフトウェアテスト技術者資格制度)技術委員

ISO/IEC JTC1/SC7 WG26（ISO29119：テストプロセス標準の策定）委員

□著書

- 書籍（6冊）：[現場の仕事がバリバリ進むテスト手法](#)、[ソフトウェアテストの基礎](#)、[JSTQBソフトウェアテスト教科書](#)、[基本から学ぶソフトウェアテスト](#)、[ソフトウェアテスト293の鉄則](#)、[TPINEXT（ビジネス主導のテストプロセス改善）](#)
- 学術論文：「データ共有タスク間の順序組合せテストケース抽出手法」電気学会 論文誌C Vol. 137 No. 7
- 国際学会：[A Test Analysis Method for Black Box Testing Using AUT and Fault Knowledge](#) など多数
- 雑誌、ネット記事：[日経ITPro](#)など多数

アジェンダ

開発におけるテストの役割

テストプロセス

テストプロセスの改善

テストの活動の組織化

まとめ

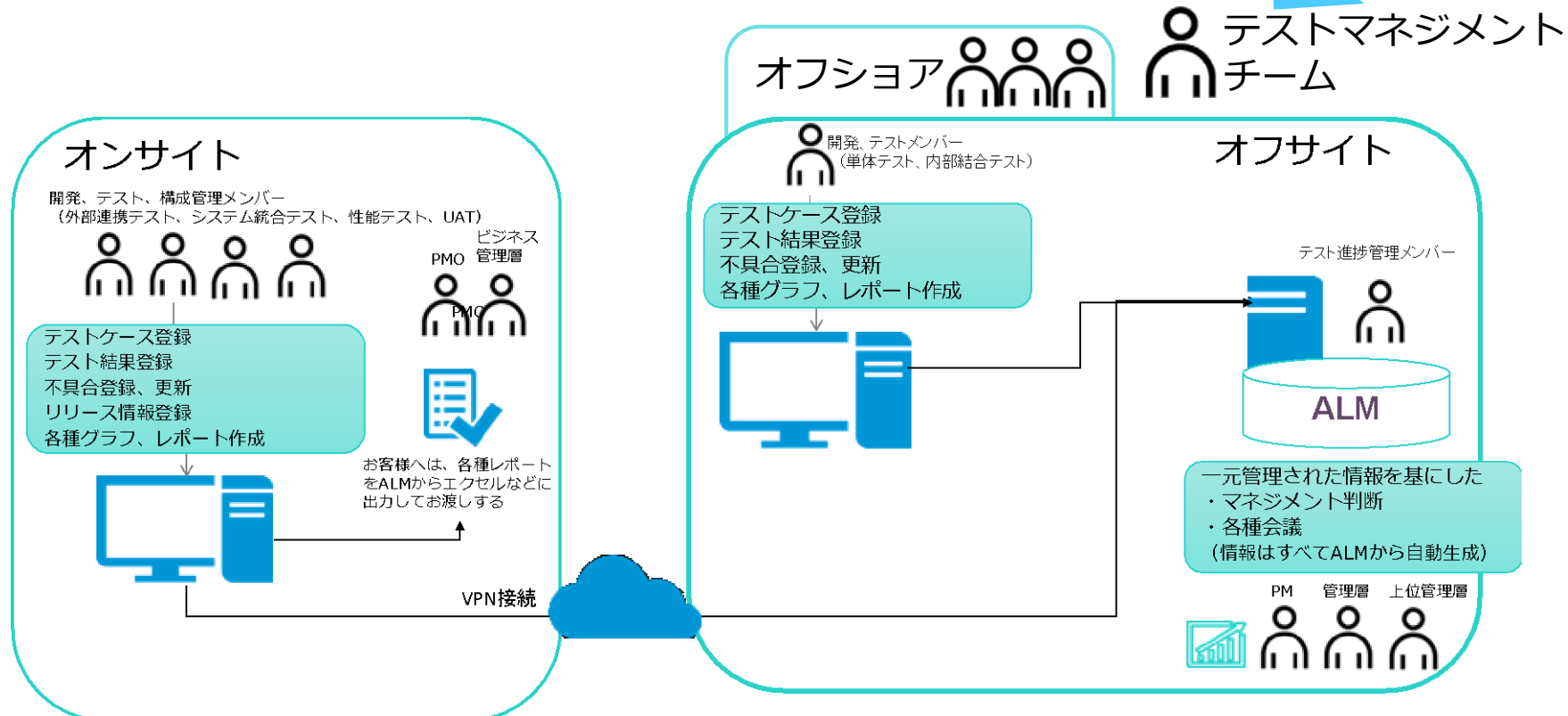
本日の講演のターゲット

テストマネジメントに携わる人たち

- ・ 管理工数を削減
- ・ 的確な判断と対処

※テストマネジメントを行う人たちが取り組むべきことを主題とします

この人たちが
ターゲット



開発におけるテストの役割

テストの活動を開発のクリティカルパスにしないこと

ソフトウェア開発において テスト工数はどの程度を 占めているでしょう？

A: 27%

B: 37%

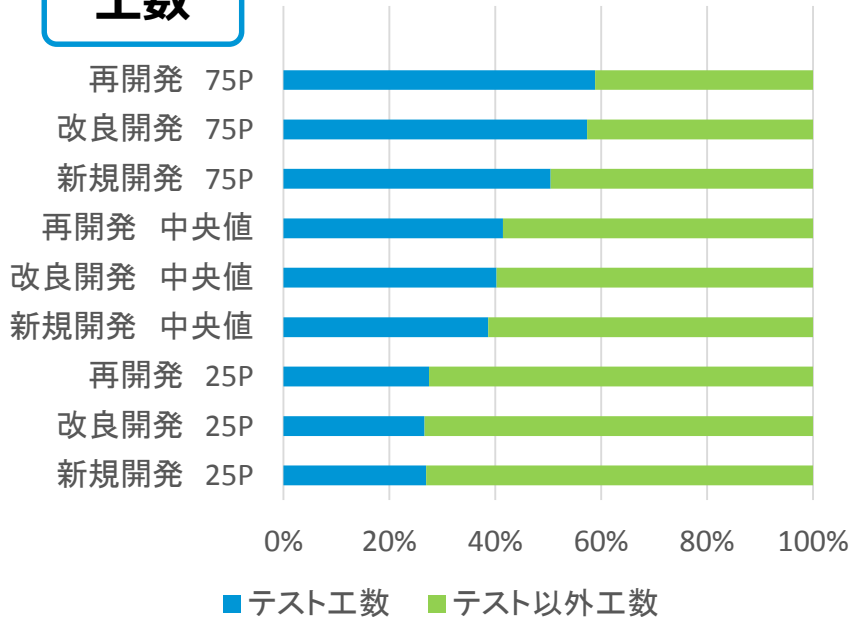
C: 40%

D: 58%

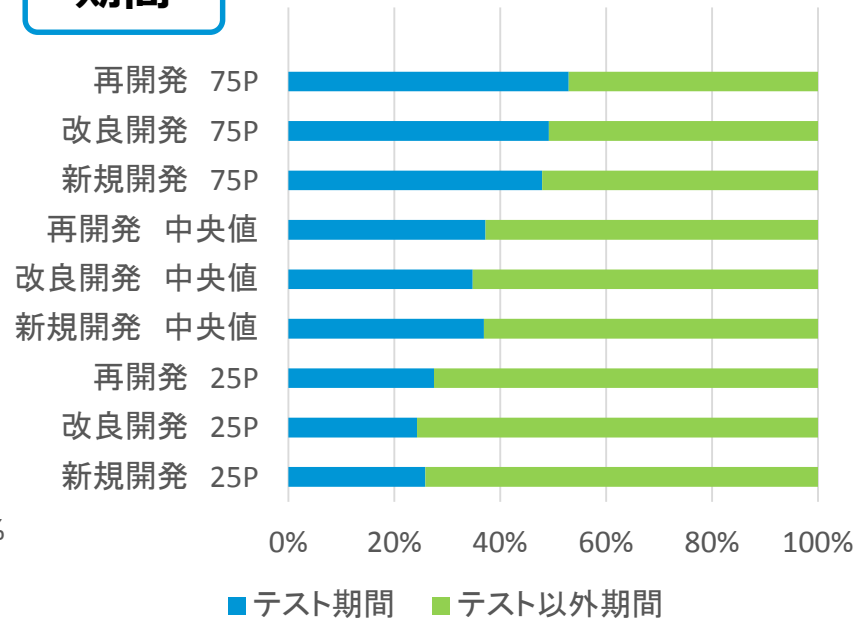
※IPA ソフトウェア開発データ白書2018 - 2019をベースに
開発5工程に要件定義も含めた工程比率で、再計算した値
テスト工数 = (製作の1/3を単体テストと仮定) + 結合テスト + 総合テスト

テストにはどの程度のばらつきがあるのか？

工数



期間



※IPA ソフトウェア開発データ白書2018 - 2019をベースに
開発5工程に要件定義も含めた工程比率で、再計算した値
テスト工数 = (製作の1/3を単体テストと仮定) + 結合テスト + 総合テスト

上記比率を使って見積もってみると

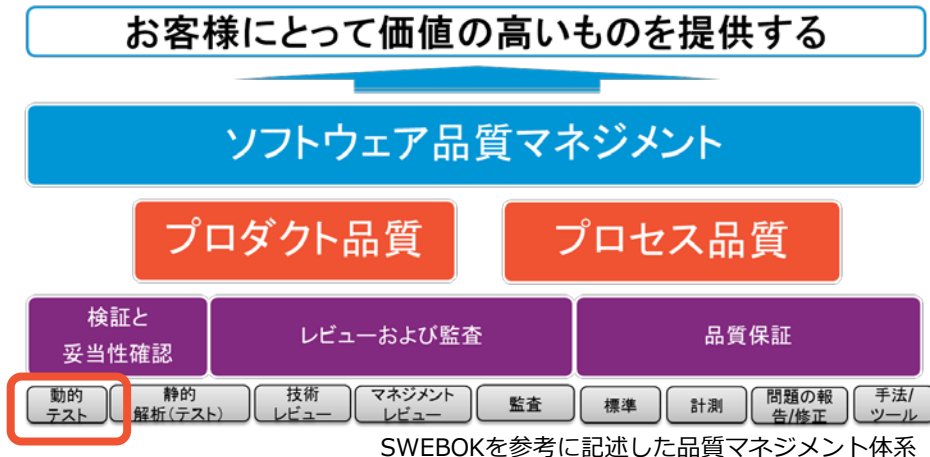
★1ヶ月 5人月の小規模改良開発

テストは、「 」日間「 」人日のテストが必要と見積れる

★1年間 500人月の大規模プロジェクト

テストは、「 」ヶ月間「 」人の人が行うと見積もれる

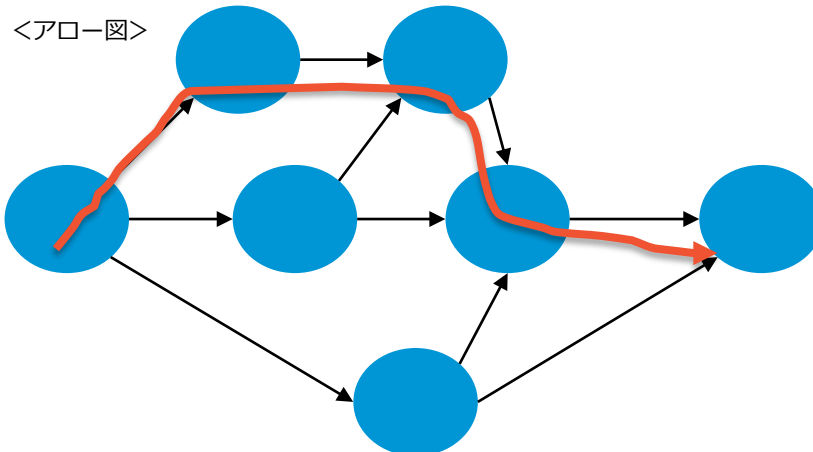
テストには2つの大きな役割



品質の良いものを世の中に 送り出すための大事な手段

- 品質計測のインプット情報として鮮度がよく綺麗な情報を渡すこと

どれだけ正しく写す鏡か？



開発ライフサイクルの主要な活動

- 技術業として、決められたQCDを達成できるように仕事をする

どれだけ効率的にできるか？

上手いテストをするとは

テストの活動をクリティカルパスにしないこと

テストは品質マネジメントの一手段

- 他の品質マネジメント手段との住み分けをしてスコープを明らかにする
- 関係する人たちへの「迅速に」「正しい」情報提供

テスト技術次第

- テスト対象の確認をより少ない量で確実に行う技術（テスト設計、テスト戦略）
- 作業スピードを上げる技術（仮想化技術、自動化技術）
- テストでのアウトプットを再利用する技術（上記技術の適用、テスト品質、構成管理）

開発ライフサイクル次第

- テストの成熟度を高めようとするほど、開発の状況が前提条件となる

テストプロセス

上手いテストをするには、テスト実行以外にもやることがある

テストプロセス

- ・ テスト実行だけでは、「何故その行為をするのか？」が不明のまま
- ・ それに、仕組みや仕掛けを入れ込んでおかないと、実行するのもままならない

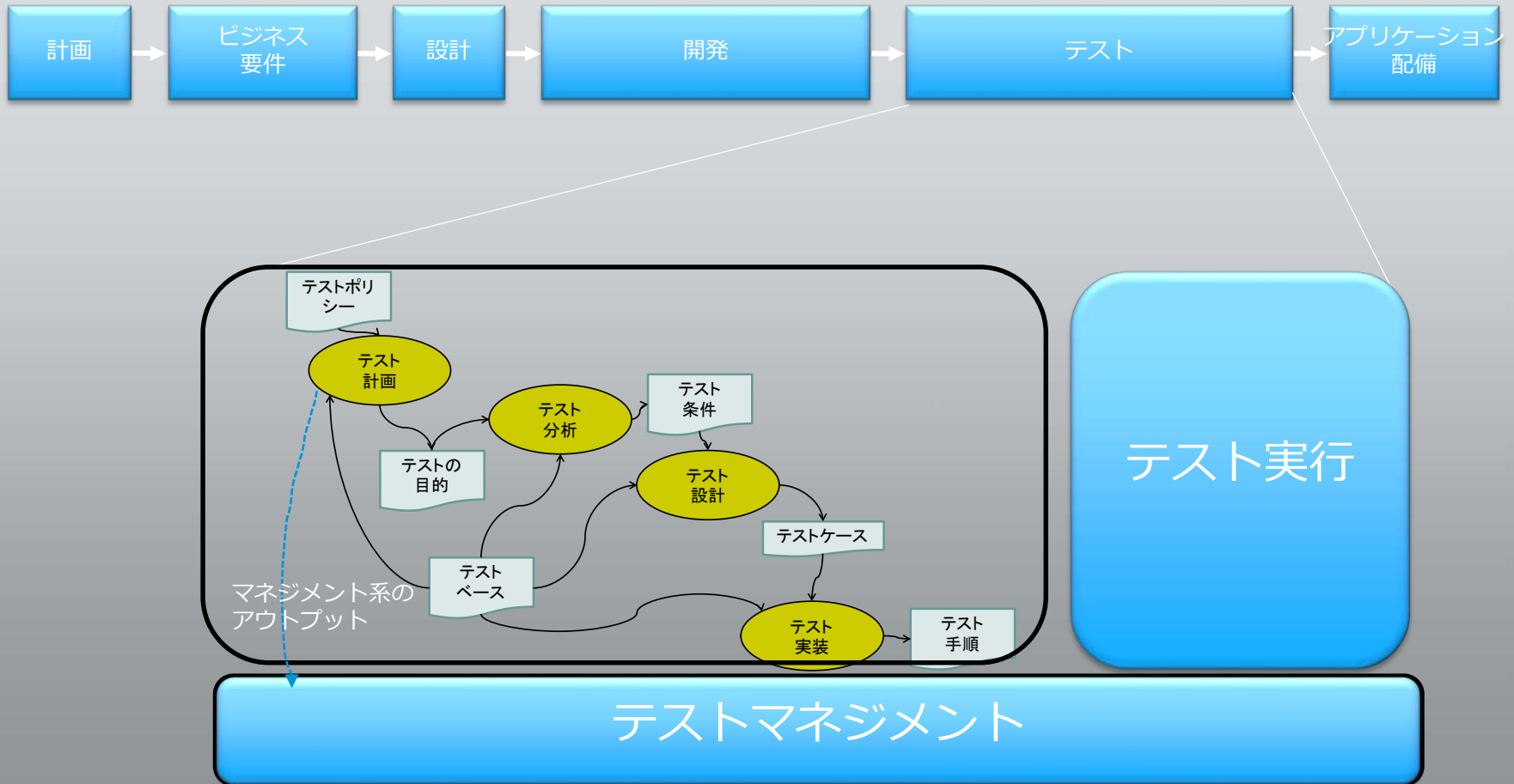
開発ライフサイクル



テストプロセス

- ・ テスト実行だけでは、「何故その行為をするのか？」が不明のまま
- ・ それに、仕組みや仕掛けを入れ込んでおかないと、実行するのもままならない

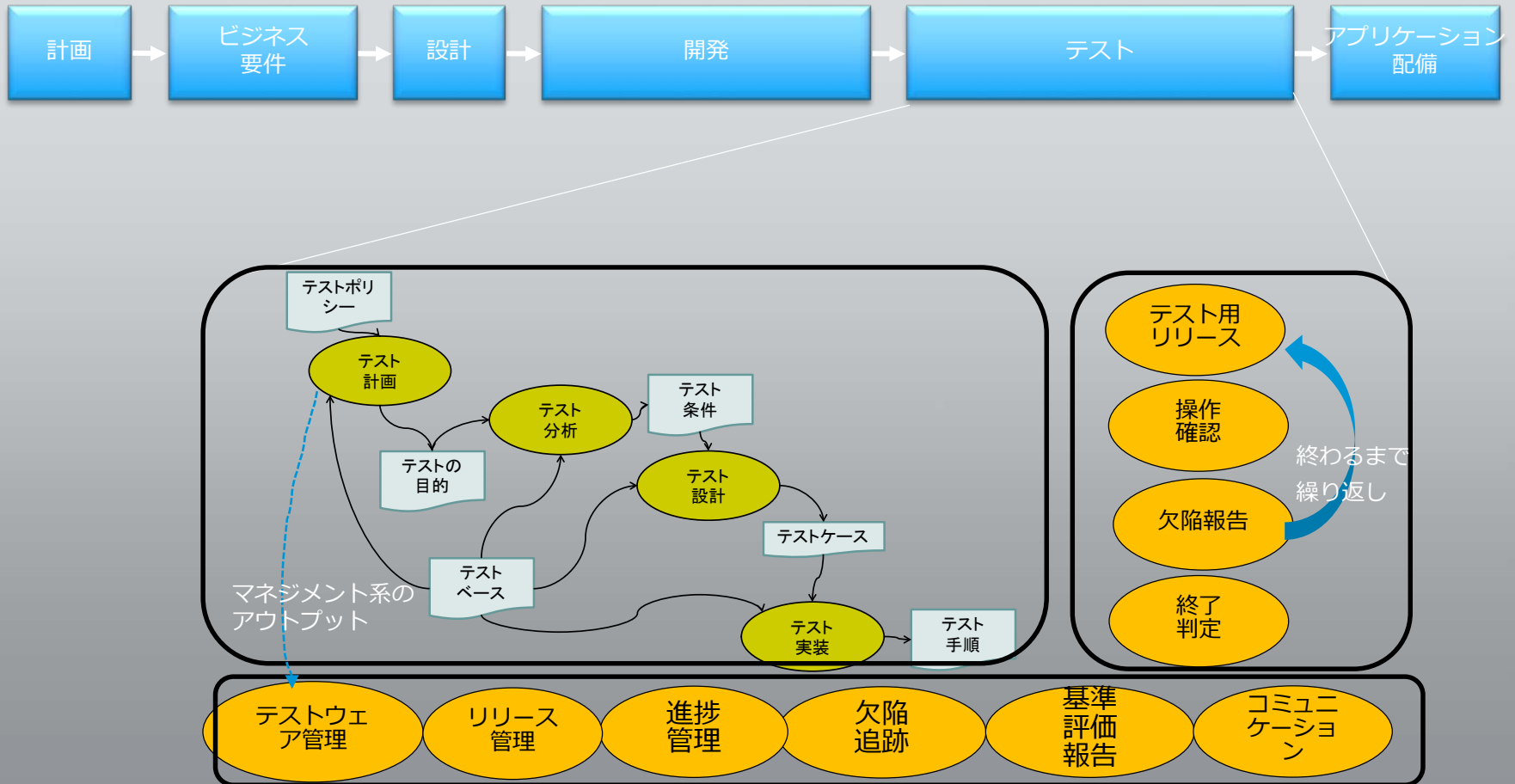
開発ライフサイクル



テストプロセス

- ・ テスト実行だけでは、「何故その行為をするのか？」が不明のまま
- ・ それに、仕組みや仕掛けを入れ込んでおかないと、実行するのもままならない

開発ライフサイクル



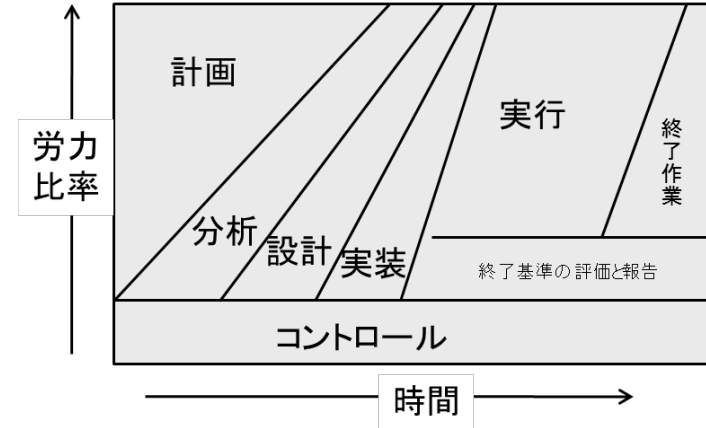
ISTQBが定義するテストプロセス

<http://jstqb.jp/>

ISTQBでは、テストプロセスは、以下の活動に分けて考えるとしている

- 計画
- コントロール
- 分析
- 設計
- 実装
- 実行
- テスト終了基準の評価と報告
- テスト終了作業

テスト開発プロセス



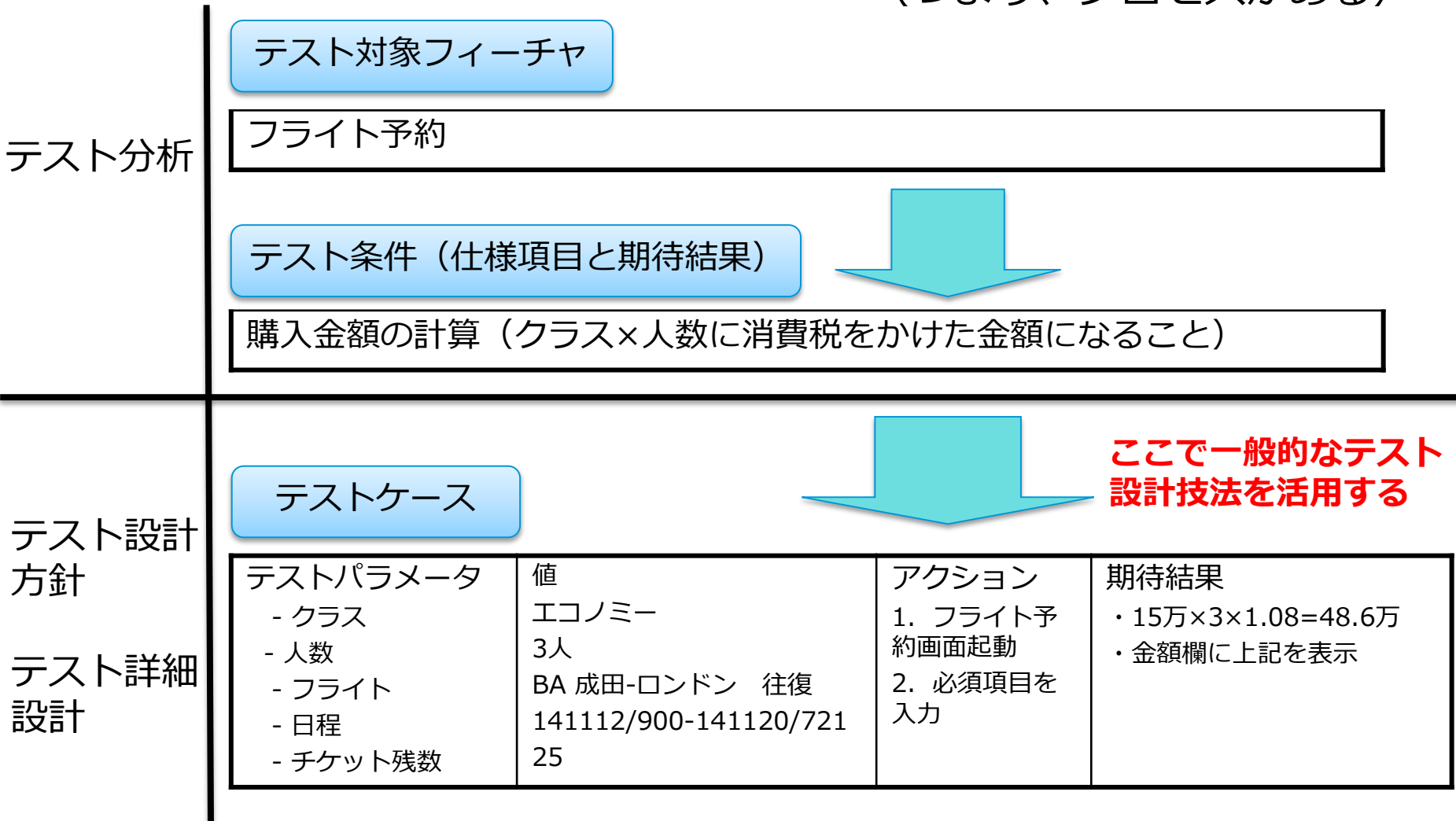
これらの活動はひとつずつ順番に行われるとは限らず、重なったり、並行したり、繰り返されたりする

テスト開発プロセスの三つの活動

- (分析する) テストを作れるようテスト対象を理解する
- (設計する) 目的に沿ったテストを作る
- (実装する) テストを実行できるよう用意する
 - (手動なら) 手順を考える (自動なら) テストコードを書く
 - 環境構築、テストデータ作成

テスト開発プロセス

本来のテストケースは以下の構造となり、段階的に成果物を作っていく
(つまり、プロセスがある)



テストプロセスの改善

現場で上手いテストをしていくための方法

テスト設計技法の適用箇所

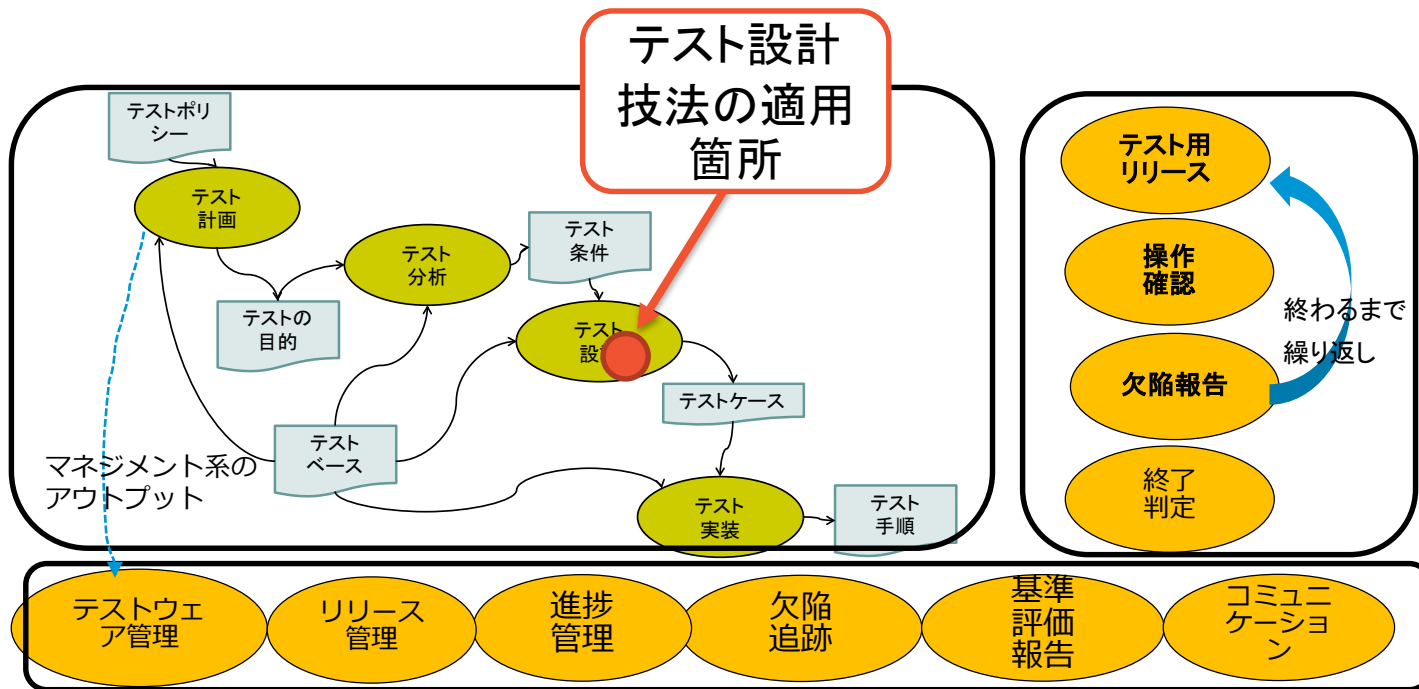
テストプロセスの改善

テストプロセスとして全体を考慮した改善の必要性

テストプロセスはソフトウェア／システム開発の多くを占めている

テストを良くするためには技法、手法だけで対応できない

- 多くの依存関係、想定外の対応を余儀なくされる



テスト設計技法に課題があるときの影響（1）

テストプロセスの改善

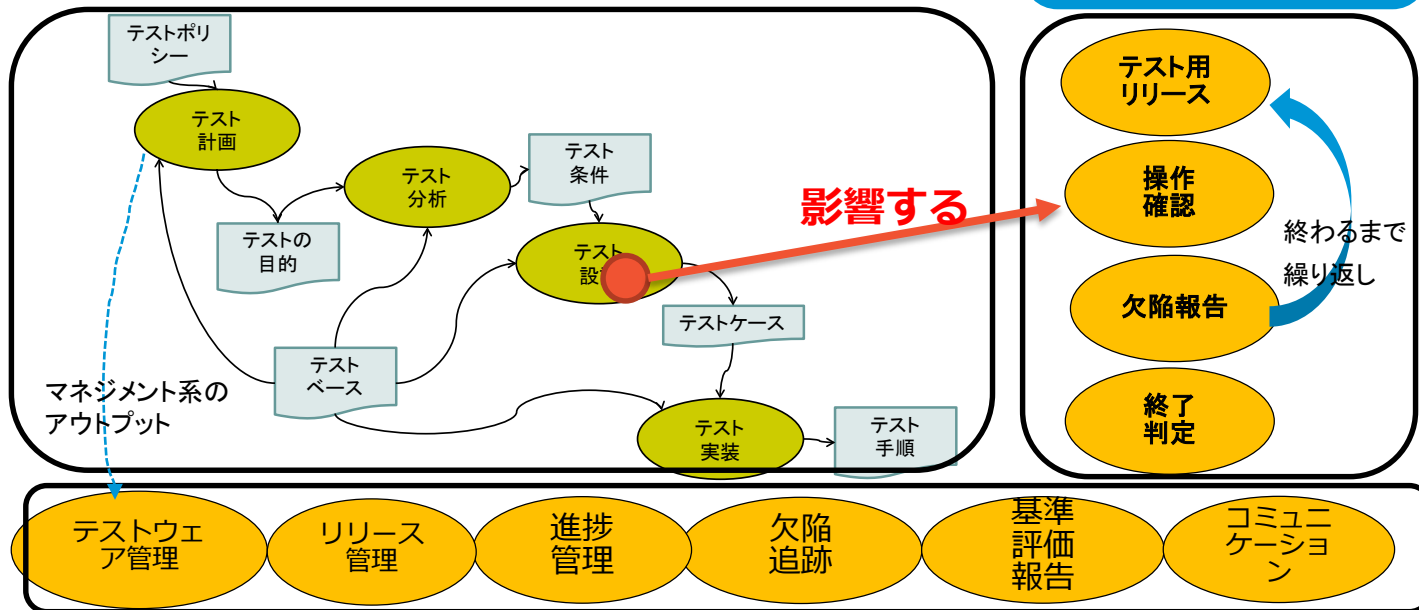
テストプロセスとして全体を考慮した改善の必要性

テストプロセスはソフトウェア／システム開発の多くを占めている

テストを良くするためには技法、手法だけで対応できない

- 多くの依存関係、想定外の対応を余儀なくされる

テストが無駄に多いと…



テスト設計技法に課題があるときの影響（2）

テストプロセスの改善

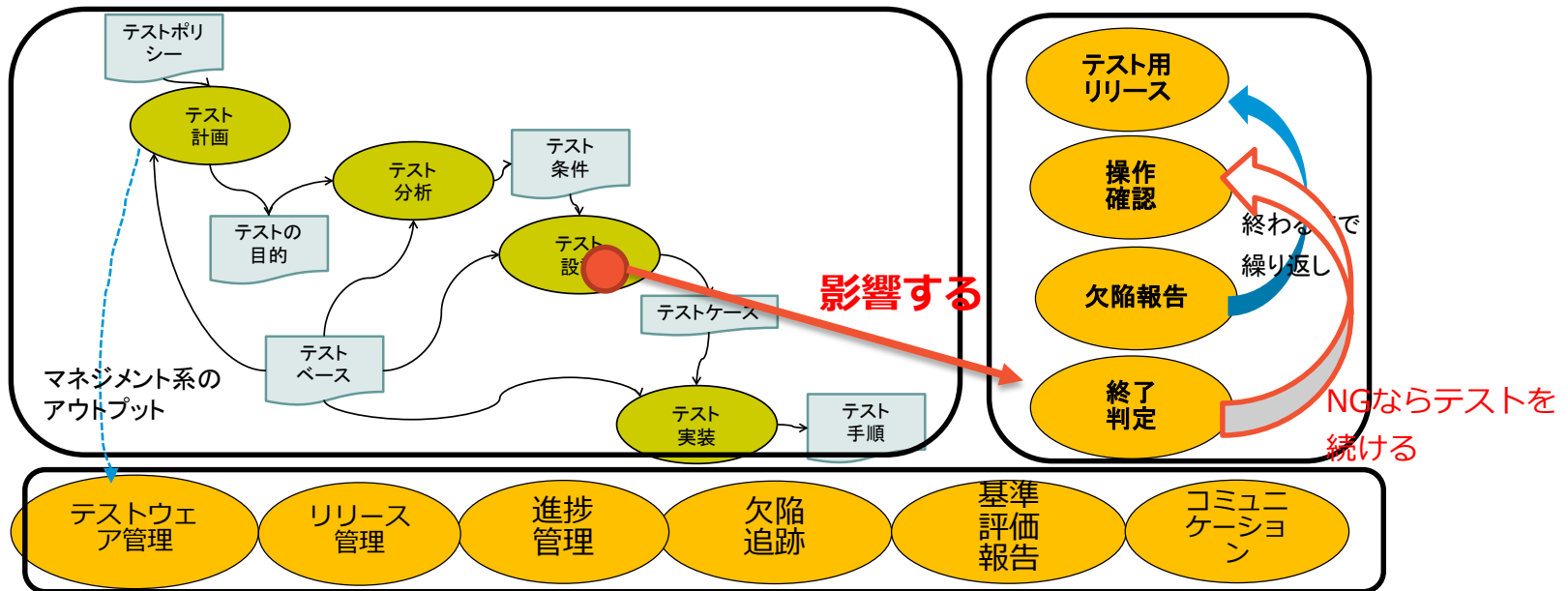
テストプロセスとして全体を考慮した改善の必要性

テストプロセスはソフトウェア／システム開発の多くを占めている

テストを良くするためには技法、手法だけで対応できない

- 多くの依存関係、想定外の対応を余儀なくされる

テストがあきらかに足りないとき…



テスト設計技法に課題があるときの影響 (3)

テストプロセスの改善

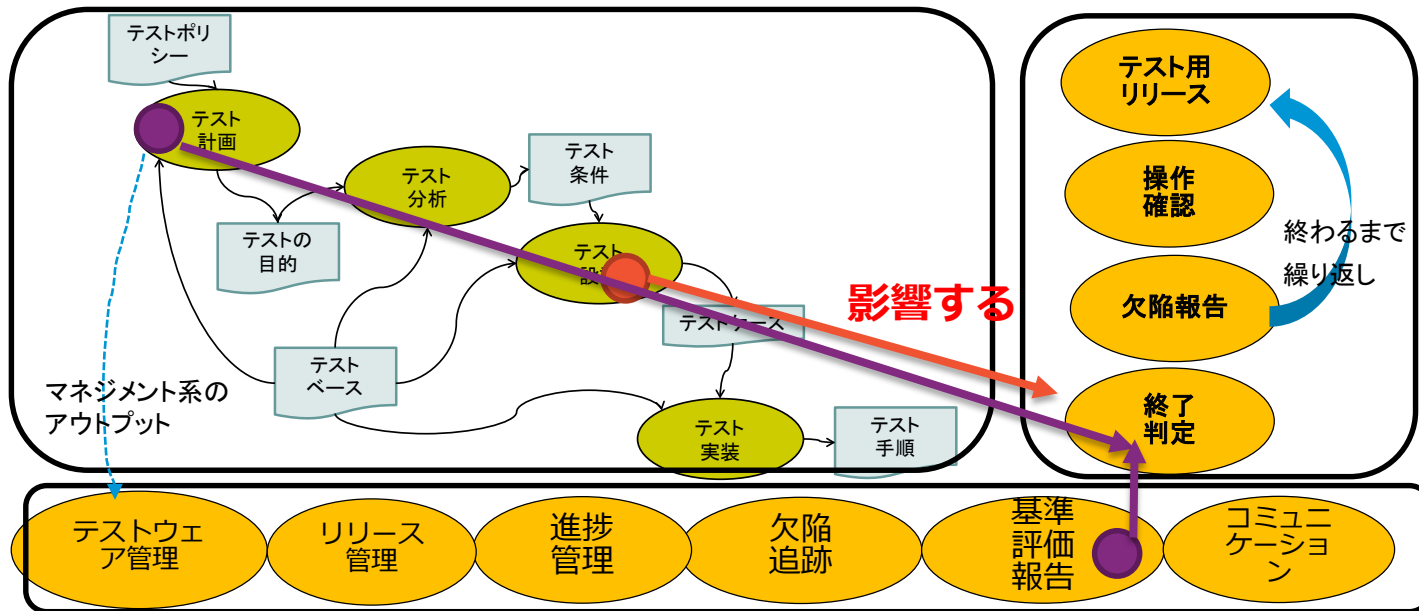
テストプロセスとして全体を考慮した改善の必要性

テストプロセスはソフトウェア／システム開発の多くを占めている

テストを良くするためには技法、手法だけで対応できない

- 多くの依存関係、想定外の対応を余儀なくされる

更に計画時に判断基準を誤ると…



市場トラブル

テスト設計技法を習得しても解決しない問題（1）

テストプロセスの改善

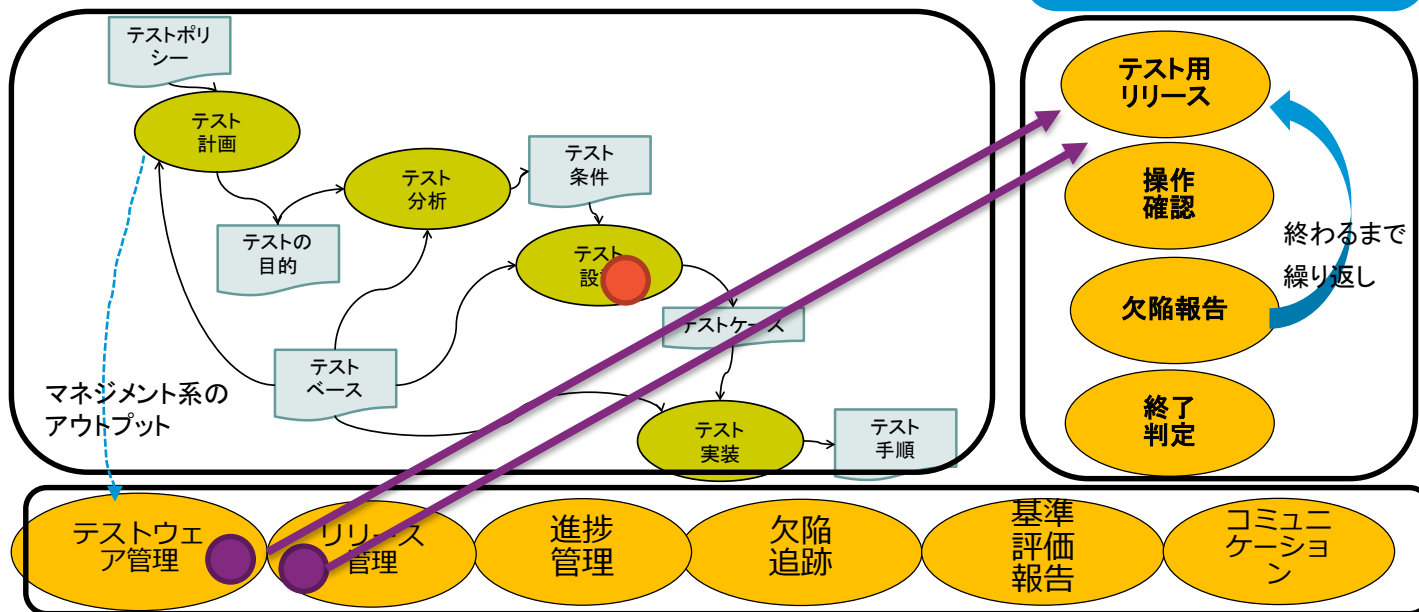
テストプロセスとして全体を考慮した改善の必要性

テストプロセスはソフトウェア／システム開発の多くを占めている

テストを良くするためには技法、手法だけで対応できない

- 多くの依存関係、想定外の対応を余儀なくされる

リリースにもた
ついてどんどん
遅れる



テストウェアマネジメント、リリースマネジメントを怠ると

テスト設計技法を習得しても解決しない問題（2）

テストプロセスの改善

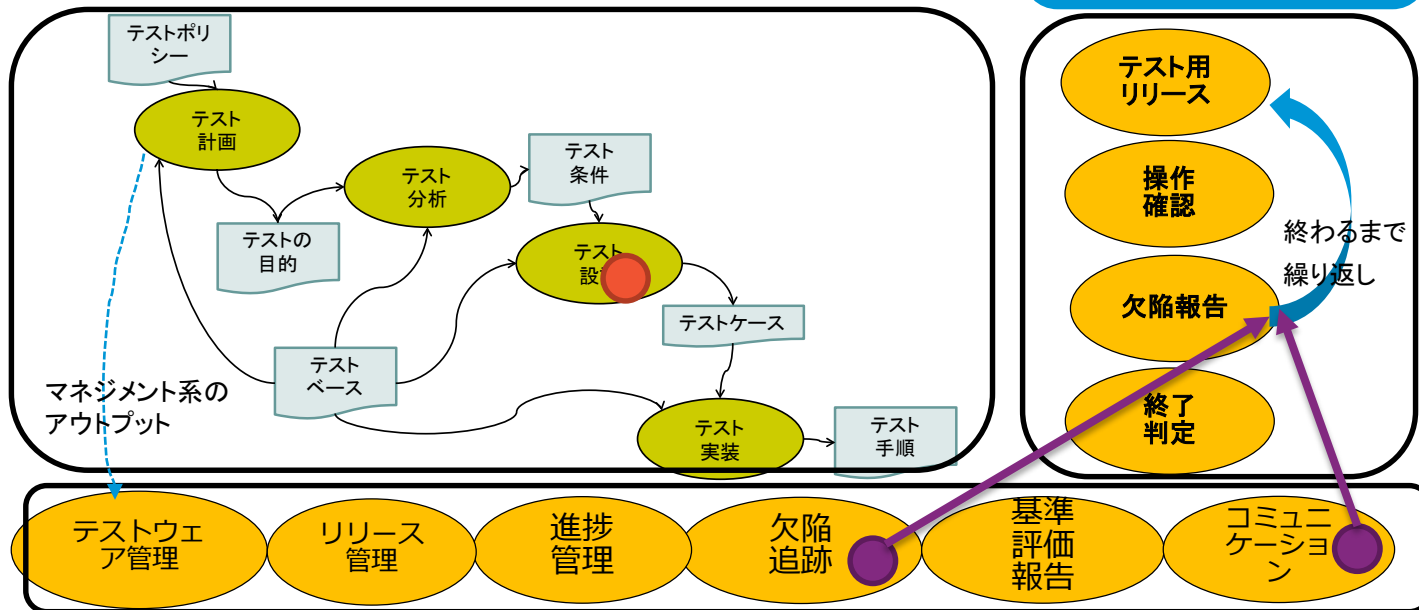
テストプロセスとして全体を考慮した改善の必要性

テストプロセスはソフトウェア／システム開発の多くを占めている

テストを良くするためには技法、手法だけで対応できない

- 多くの依存関係、想定外の対応を余儀なくされる

欠陥が直らない
ので進まない



欠陥追跡を怠ると

世の中に公開されているテストプロセス改善モデル

段階的モデル

TMMi (テスト成熟度モデル統合...CMMIを基にしたテストプロセス改善モデル)
ISO/IEC/IEEE33063 Process assessment model for software testing

連続的モデル

TPI NEXT (TPIの後継版として2009年に開発された)

CTP (クリティカルテストイングプロセス)

STEP(体系的テストと評価プロセス)

※段階的モデル、連続的モデルの分類はISTQBテスト技術者資格制度
Advanced Level シラバス 日本語版 テストマネージャ Version2012.J03より引用

**テストを改善するにはテストに特化した
参照モデル（改善の道具）利用が近道**

TPI NEXTとは

テストプロセス改善に特化した成熟度モデル

“自己評価”とそれに基づく“自己改善”

アセッサにより“認定”したり,“認定”されたりしない

評価軸(通常横軸)は単なるプロセスではない

独自に選定した16個のキーエリア (KA)

人の要素、技術要素を明示している

ビジネス主導 [BD (ビジネス主導) のTPI]

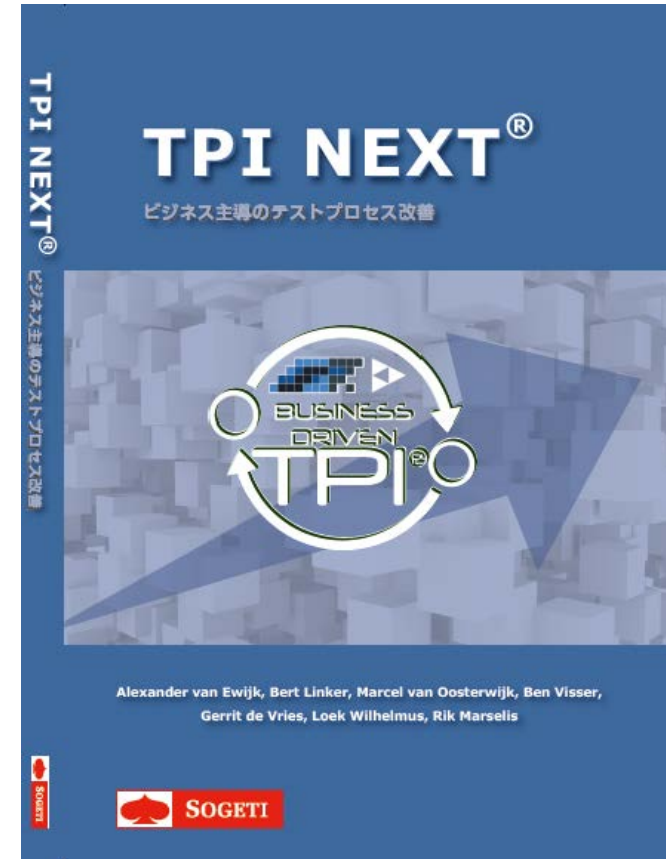
TPI NEXTでいう「ビジネス」とは

「テスト業務が実現すべきこと」を指している

- 適切な市場投入タイミング順守
- コスト削減
- テスト対象の有効性の向上、など

達成度の因果関係を示す「クラスタ」

- ビジネス要因を加味したクラスタのカスタマイズ



英語版

- 時期: 2013年出版
- 出版社: UTN Publishers
- 著者: Alexander van Ewijk, 他6名

日本語版

- 時期: 2015年出版
- 出版社: トリフオリオ
- 訳者: 薮田和夫、湯本剛、皆川義孝

改善の対象となるキーエリアのグループ

3つのグループの特徴

利害関係者のコミットメント
関与の度合い
テスト戦略
テスト組織
コミュニケーション
報告

利害関係者との関係

- ・ 開発プロジェクト全体から情報収集が必要
- ・ 改善対象がテストと関連する他組織にひろがる

テストマネジメント

テスト業務の専門性

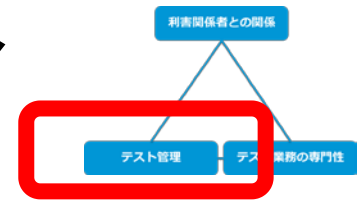
手法の実践
テスト担当者のプロ意識
テストケース設計
テストツール
テスト環境

- ・ テストマネジメントする人たちからの情報収集が必要
- ・ KA全般的に改善対象に関わる人が同じ

- ・ テストを実際に実行する人たちからの情報収集が必要
- ・ KA毎に改善対象に関わる人が異なる

テストプロセス管理
見積りと計画
メトリクス
欠陥管理
テストウェア管理

キーエリアグループ：テストマネジメント



	定義	コントロールレベル	効率化レベル	最適化レベル
テストプロセス管理	テストプロセス管理は、与えられた時間とコストとリソースの中で、テストに課せられた任務の遂行を最大化する。	・テスト計画の立案 ・テスト計画の合意 ・モニタリングと調整	・計画逸脱を予測し事前協議 ・調整内容の可視化 ・人員含めた再配置権限	・テストプロセスの改善サイクル
見積りと計画	見積もりと計画の技法を適切に利用すると、テストプロセスの計画や見積もりが予測可能で信頼のおけるものになる。	・各活動の工数見積り実施 ・活動の依存関係明確化	・見積りにFPなどの技法やメトリクスを活用 ・テストビリティをレビューする工数を計画	・組織全体で見積り用データ収集し、利用 ・見積り方法の保守 ・テストウェア再利用のための工数を計画
メトリクス	メトリクスを使った定量的な観察は、客観性を与える。	・メトリクスの定義と利用 ・収集データの管理 ・データの精度を監視	・メトリクスのための工数を得られたメリットごと計測 ・メトリクスの活用についての改善サイクル	・情報ニーズと対応するメトリクスを連動し、ニーズの変化にも迅速に対応
欠陥管理	欠陥管理とは、個人レベルとグループレベルの両方で、根本原因分析やガイドラインを利用して欠陥に対処することである。	・欠陥追跡ライフサイクルと記録内容の定義と運用 ・対応責任者明確化 ・関係者間での同一データへのアクセス保証	・欠陥情報が即時活用可能 ・プロジェクトレベルで欠陥データへのアクセス保証／権限コントロール ・傾向の特定が可能	・組織レベルで欠陥予防に活用できるだけの分析とフィードバックルール運用
テストウェア管理	テストウェアの管理によって、テストの成果物間や、テストの成果物と関連する設計文書間の一貫性が確保できる。	・テストに関わる成果物へのアクセス保証 ・テストに関わる成果物のバージョン管理 ・テストケースのもとになるテストベースが明確	・テストベース、機能とテストウェアのトレース確保 ・要件とのトレース確保 ・テストウェアを保管するための論理的構造の定義と運用	・再利用ルール ・保守フェーズへの引き渡し ・再利用の程度を計測

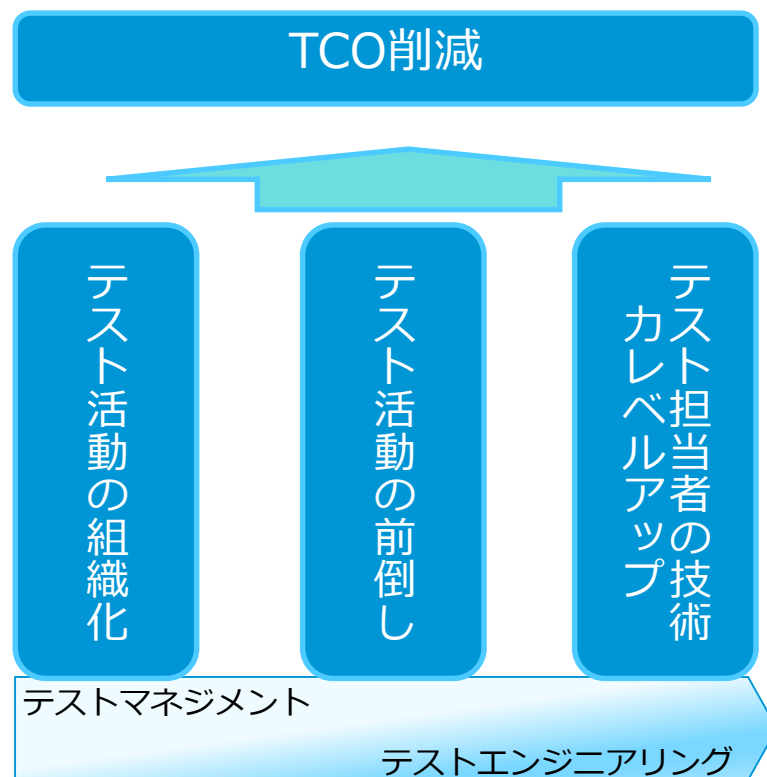
テストの活動の組織化

テストマネジメントから始める施策

テストをボトルネックにしない施策

これまでの経験から言える、テストマネジメントから始める施策の優先順位

1. テスト活動の組織化（情報一元化でテストを可視化して的確な判断を支援）
2. テスト活動の前倒し（要件定義段階での品質確保）
3. テスト担当者の技術カレベルアップ（テスト設計、自動テスト）



施策とテストプロセス改善の関係

3つのグループの特徴

テスト施策

テスト活動の組織化

改善対象キーエリア：テストプロセス管理

↓
テストウェア管理

↓
報告、コミュニケーション

テスト活動の前倒し

改善対象キーエリア：テストプロセス管理

↓
計画と見積もり

↓
関与の度合い

テスト担当者の技術力 レベルアップ

改善対象キーエリア：手法の実践

↓
プロ意識

↓
テストケース設計、テスト環境

マネジメント視点でのテストプロセス改善の内容

テスト設計、テスト結果、不具合などのテストの成果物を標準化する（ガイド、テンプレート提供）
PJ支援時にテンプレートの利用を確認し、必要に応じて是正する
→標準化した**テストウェアをデータベースで一元管理**する
関係者全員にとっての正確な情報源／迅速な意思決定に貢献

※**テストマネジメント主導で始められる**
（プロジェクトの行く手を照らすヘッドライトとなる）

テスト活動の関与の時点が早くなるよう開発プロセスを変更する。
見積もり時にリソース配分、スキル所有者アサインを調整する。
早期にマスターテスト計画を作成し、全員が早期にテストを開始する計画であることを合意する。

※**局所的な現場主導の取り組みはできるところから始めるのがよい**
開発プロジェクトメンバー全体の合意が必須
（テストマネジメントとして開発メンバーをフル活用する）

該当する活動に必要な技術、手法の利用ガイドを作成する。
ターゲットPJでの改善推進時には、ガイドを使った教育（育成）、技術の利用推進、課題協議を行う。
→テストを設計する手法の導入（ゆもつよメソッドなど）
→自動テストでの網羅率向上を組織的に展開

※**局所的な現場主導の取り組みはできるところから始めるのがよい**
スキルを持ったテスト担当のアサインが必須
（テスト技術者の技術力をフル活用する）

マネジメントからみた一元管理情報の三つの活用例

状況の可視化によって、正しい判断を行う。そのための情報を得るのに工数や時間を最小限にする。そうすると最新情報を頻度高く得られる。

- ✓ プロジェクト進行中のテスト実行進捗把握の効率化
- ✓ プロジェクト進行中の不具合追跡の効率化
- ✓ 定期的な上位管理者層への進行中状況のレポート作成
- ✓ 素早い品質判断情報入手

事例

小規模 ▲

大規模 ◎

状況の可視化を正確に行うため、基礎情報の登録が正しいかを確認する作業を効率化する。

- ✓ テスト結果記載状況確認作業の効率化
- ✓ 不具合記載情報記載状況確認作業の効率化

小規模 ○

大規模 ◎

テスト結果から品質分析を行い、傾向分析やプロジェクト間の比較、未来のプロジェクトへのフィードバックを行う。

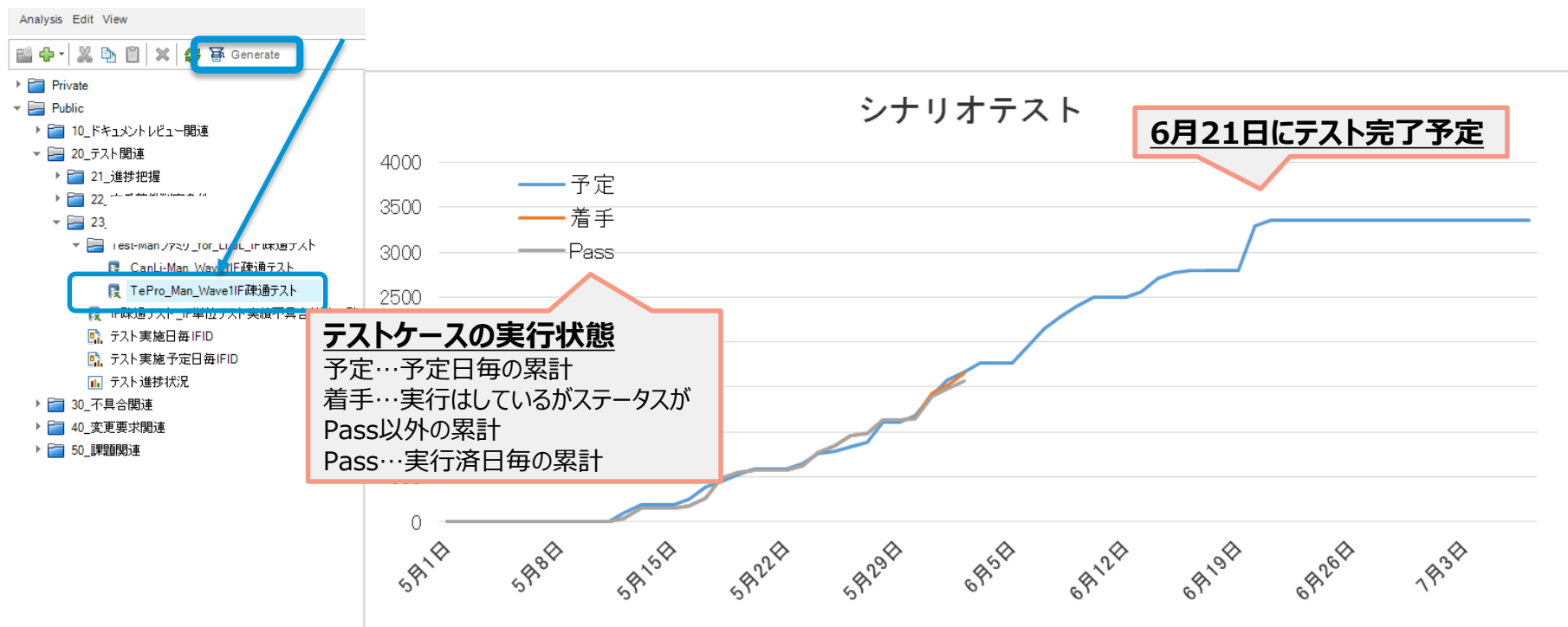
小規模 ◎

大規模 ◎

テスト実行進捗把握をリアルタイムに見る例

テスト実行の進捗状況グラフを機能領域別、テストフェーズ（リリース）別に
テストケース単位、テストシナリオ単位で簡単に把握

テスト管理データベースにテスト結果を日次で登録する運用にしておき、いつでも進捗背を確認
したいときにボタンを押すと進捗がわかるエクセルが生成される（このようなツールを埋め込む）



Test execution Progress Management Tool

Test-Man Family

当時テストマネジメントチームにいた長谷川さん 現)ベリサーブ が作成したツール

テスト実行進捗把握をリアルタイムに見る例

週単位の実行数集計

予定…予定日毎の累計
着手…実行はしているがステータスが
Passed以外の累計
完了数…実行済日毎の累計

シナリオ単位の実行数集計

シナリオ
名のリスト

		予定数	実行数	合格数	完了フラグ	予定開始日	予定完了日	着手日	完了日
T00684	得意	0	0	0	未			未登録	未登録
T00685	得意	0	0	0	未			未登録	未登録
T00689	仕	0	0	0	未			未登録	未登録
T00697	仕	0	0	0	未			未登録	未登録
T00699	品	2	2	2	完了	10月13日	10月13日	10月13日	10月13日
T00706	品	2	2	2	完了	10月13日	10月13日	10月13日	10月13日
T00707	品	0	0	0	未			未登録	未登録
T00710	品	3	3	3	完了	10月13日	10月13日	10月13日	10月13日
T00712	品	2	2	2	完了	10月13日	10月13日	10月13日	10月13日
T00751	品	0	0	0	未			未登録	未登録
T00402	在	0	0	0	未			未登録	未登録
T00403	在	0	0	0	未			未登録	未登録

■進捗状況全体 (テストケース単位)

			実施						
20 週			予定数	着手数	完了数	完了予実	着手累計	完了累計	
1	10月10日	10月16日	9	9	9	0	9	9	
2	10月17日	10月23日	3	6	6	-3	15	15	
3	10月24日	10月30日	6	0	0	6	15	15	
4	10月31日	11月6日	0	0	0	0	15	15	
5	11月7日	11月13日	0	0	0	0	15	15	

週単位のシナリオ単位実行 数合計

■進捗状況全体

20 週			予定数	着手数	完了数	完了予実	着手累計	完了累計	
1	10月10日	10月16日	4	4	4	0	4	4	
2	10月17日	10月23日	1	2	2	-1	6	6	
3	10月24日	10月30日	2	0	0	2	6	6	
4	10月31日	11月6日	0	0	0	0	6	6	
5	11月7日	11月13日	0	0	0	0	6	6	

日単位の実行数集計 (テストケース)

			予定数	実行数	合格数	失敗数	予定累計	着手累計	合格累計	失敗累計
10月10日	月	1	0	0	0	0	0	0	0	0
10月11日	火	1	0	0	0	0	0	0	0	0
10月12日	水	1	0	0	0	0	0	0	0	0
10月13日	木	1	9	9	9	9	9	9	9	9
10月14日	金	1	0	0	0	0	9	9	9	9
10月15日	土	1	0	0	0	0	9	9	9	9
10月16日	日	1	0	0	0	0	9	9	9	9
10月17日	月	2	3	0	0	0	12	9	9	9
10月18日	火	2	0	6	6	6	12	15	15	15
10月19日	水	2	0	0	0	0	12	15	15	15

Test execution Progress Management Tool

Test-Man Family

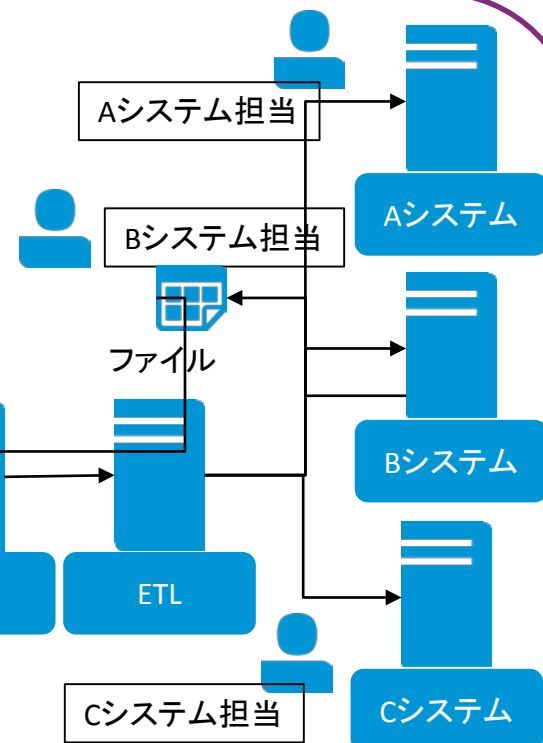
当時テストマネジメントチームにいた長谷川さん 現)ベリサーブ が作成したツール

システム統合テストのテストシナリオと テストケース作成の実例

データ準備チーム



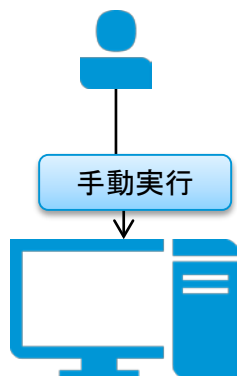
他システム



ユーザー



テストチーム

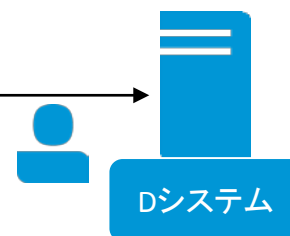


バッチ実行



SaaS

D担当



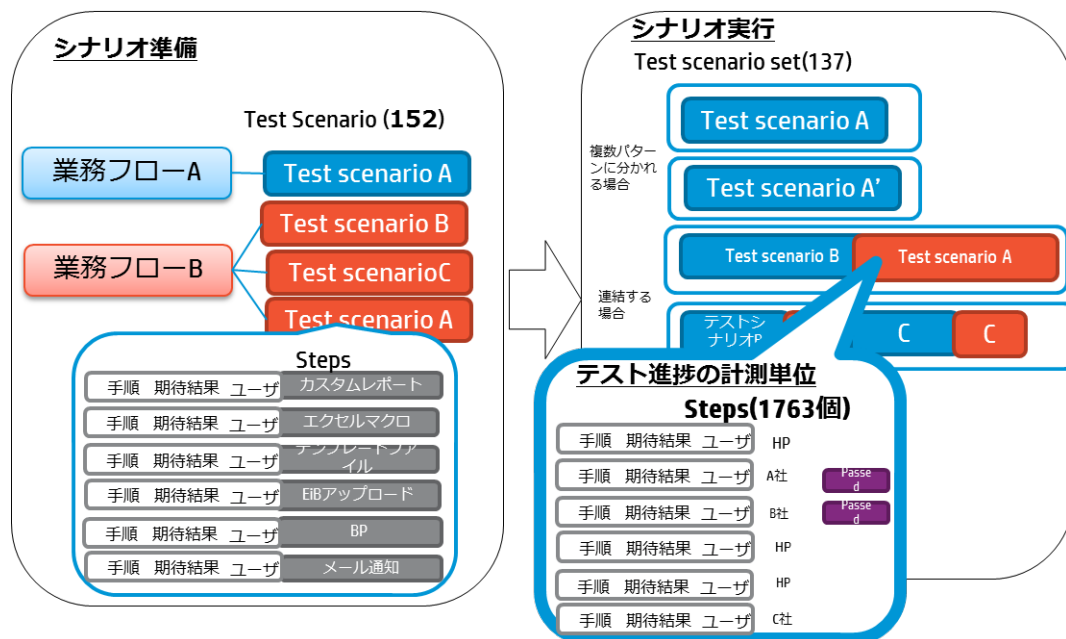
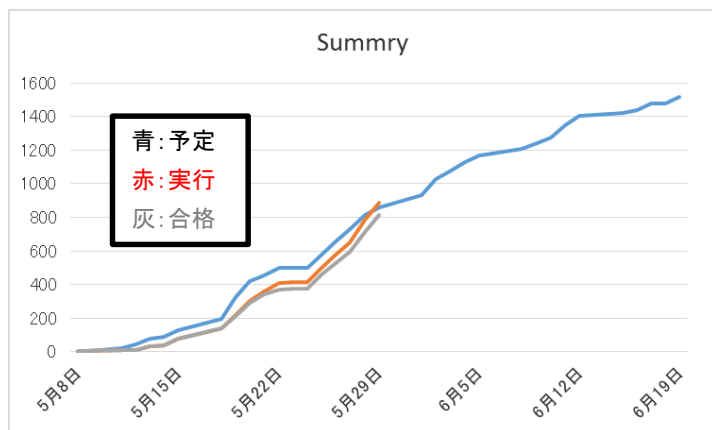
システム統合テストのテストシナリオと テストケース作成の実例

シナリオ作成と実行スケジューリングを分離して管理

各業務単位でテストシナリオを作成（業務知識のあるメンバーが実施）

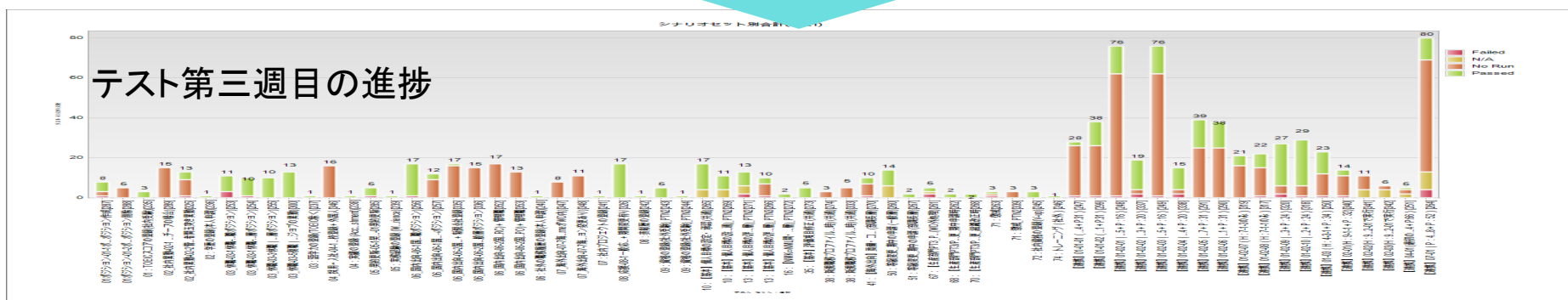
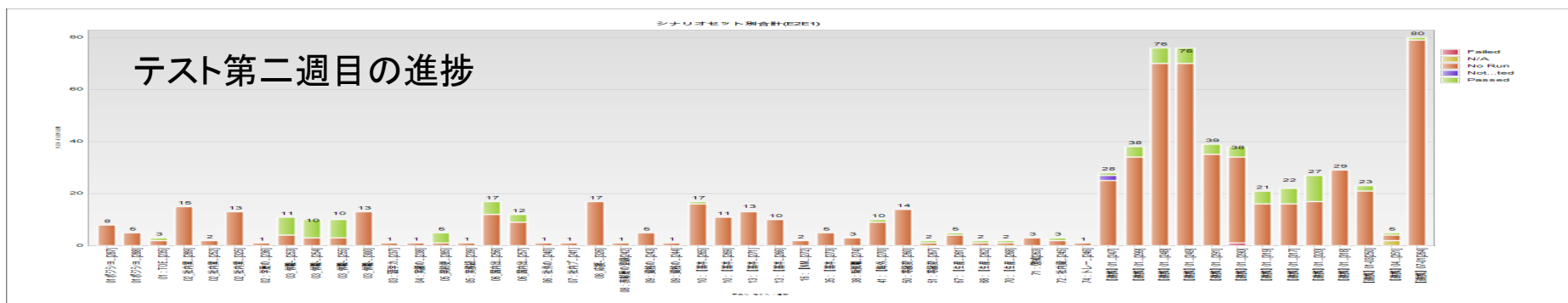
日程に合わせてフレキシブルにシナリオを組み立てて実行（テストチームにてシナリオをステップ単位に分割してテスト実行を行った）

- 1ステップ = 1つのシステム行う1つのテストケースとして管理した
- 実行可能なテストを選んでいき、想定スピードどおりにテストの進捗がでるように管理



システム統合テストのテストシナリオと テストケース作成の実例

日程に合わせたフレキシブルなテスト実行を把握するグラフの例
(テスト管理ツールより自動生成)



まとめ

開発におけるテスト役割

- テストは開発の2割~5割をしめるが、やり方次第
- テストを開発のクリティカルパスにしないのが上手いテスト
(テストが品質に貢献することは大前提！)

テストプロセス

- 活動を具体的にあきらかにして共有可能にするのがプロセス
- テストもプロセスなので、実行前に目的（とその詳細化）をして、仕掛けを入れ込む

テストプロセスの改善

- 何を改善すべきかは、テストプロセス全体で考える

テストの活動の組織化

- テストマネジメント主体でできる改善施策

最後に（次のページ）

- 「適切なこと」を行ってこそ「適切な方法」の効果が出る

最後に

テストをクリティカルパスにしないための三大要素

ちゃんと仕様理解する

- 問題が起きたら必ず仕様書を読み返す。
- テストケースに該当する仕様書がどこにあるかわかるようにしておく。

細かく可視化する

- テストケースを細かく用意して（1期待結果で1ケース）、その消化状況、一日の消化目標数、不具合とのリンクを常に全員が見れるようにする。
- 上記の「共通認識」になる情報を日次確認できる時間を作る。（5分でよい）

迅速に不具合を直す（よう働きかける）

- スタックしている不具合を、関係者達の目の前で転がして動けるようにする。
- 修正後は、できるだけ早く再テストを行いレポートをクローズする。

これらは愚直にやらないといけない。
技法、ツールはこれらを支援しないといけない。

最後に

阻害要因を乗り越えるために大事なこと

実現させるモチベーションのために必要な5つのこと

目的の共有と合意形成

- お先真っ暗にしないため「この光に向かうんだ！」

パトロール

- 放置しないため「無視されていない！」

駆け込み寺

- 途方に暮れないため「なるほど！」



ダッシュボード

- みんなが同じものを見て仕事するため「9回裏2アウト満塁！」
 - 計測とか、報告とか、一見つまらない仕事は、ダッシュボードのためにある

機械化（ツールを使った効率化）

- つまらない仕事を減らすため「刺し身たんぽぽはやだ！」
 - 標準化とは、つまりは機械化するための中間過程である

参考文献

IPA/SEC「ソフトウェア開発データ白書2018-2019」
(IPA/SEC)2018

IEEE Computer Society「SWEBOKソフトウェアエンジニアリング知識体系」
(オーム社)2005

湯本剛「ソフトウェアテストPressVol.10 特集1 今こそ聞きたいテストの上流設計」
(技術評論社) 2010

ISTQB「ISTQBテスト技術者資格制度Foundation Levelシラバス日本語版Version2011.J01」
(NPO法人ASTER ,JSTQB)2011

ISTQB「ISTQBテスト技術者資格制度Advanced Level シラバス日本語版TM Version2012.J03」
(NPO法人ASTER ,JSTQB)2013

Alexander van Ewijk 他 (Sogeti)「TPINEXT ビジネス主導のテストプロセス改善」
(トリフォリオ) 2015

ご清聴

ありがとうございました