

テスト自動化の成功を 支えるチームと仕組み

JaSST'22東北 招待講演
井芹 洋輝

この講演について

- テスト自動化の成功を支える工夫・留意点について解説します
- 資料は公開します
 - <https://speakerdeck.com/goyoki>
 - ディスコードの資料展開用チャンネルにリンクを展開します
- 対象とするテスト自動化：
 - テストレベル・テストタイプ：
 - 限定しません。
ユニットテスト～システムテストを含む、全般的な自動化が対象です
 - ドメイン：
 - ウェブ、組み込み、アプリといった特定のドメインに依存しない知見を解説します

自己紹介

- 現在
 - トヨタ自動車株式会社/テストチーム テックリード
 - テストの戦略立てや全体設計、テスト自動化推進を担当
 - JSTQB技術委員、テスト設計コンテストU30クラス審査委員長
- 主な著作・講演
 - 「システムテスト自動化標準ガイド」「androidアプリテスト技法」「ユニットテストの保守性を作りこむ」「テストを導くためのテストアーキテクチャの組み立て方」など
- これまで開発者・テスト担当者・コンサルタントなど様々な立場で、様々なテスト自動化の導入・運用を手掛けてきました。
今回はそこで得られた知見を整理して共有したいと思います

アウトライン

第一部

テスト自動化の成功

第二部

テスト自動化を支える活動

適切な
目的設定

テストバリエーション
確保

テストの
妥当性確保

テストの
内部品質確保

計画とアプ
ローチの工夫

テストの
活用拡大

自動化インフラ
への統合

監視とコントロー
ルによる改善

第三部

テスト自動化を支えるチームと仕組み

チーム

自動化
インフラ

プロセス

習慣化

テスト設計

テスト自動化
システム

テスト自動化を支えるチーム文化

能力の獲得
と向上

Whole
Team

責務の確立

価値の拡大

持続可能性
の維持

動機付け

テスト自動化の成功とは

「テスト自動化が成功した」とは

妥当な対価で目的を達成すること

テスト自動化の目的

テスト担当に閉じたの目的の例

- ・テストの効率性・有効性・正確性の向上
(手動で実現できないテストの実現など)

チームレベルの目的の例

- ・開発とテストのフィードバックサイクルの高速化
- ・Developer eXperienceの改善
- ・高度な品質や機能の実現

ビジネスレベルの目的の例

- ・ユーザのニーズ・シーズやクレームへの迅速な対応
- ・高度な製品価値の実現

目的達成を支えるテスト自動化の主要な効果

- テストの有効性の拡大
 - 手動では困難なテストを実現する
 - 本質的な目的達成に貢献する
- テストの効率性の改善
 - テスト活動にかかる時間やリソースを削減する
 - 同じコストでできるテストを増やす
- テストの正確性と信頼性の改善
 - 手動のミスを削減する
 - 誤差や不安定さに対応する
- テストの保守性の改善
 - テストの品質保証、維持、修正、再利用を容易にする
 - テストのインテグリティ（一貫性）やトレーサビリティの維持を助ける

テスト自動化の主要な対価

- 必要なリソース
 - 人：メンバー、スキル、組織
 - 物：機材、ツール、ライセンス、環境
 - 時間：学習工数、導入工数、運用・保守工数、管理工数
- 必要な労力の種類
 - 自動テストの実現
 - テスト自動化環境の構築・維持
 - 自動化テストの構築・維持
 - テスト自動化に付随する追加作業（例：複雑化、他テストの連携）への対応
 - テスト対象の改善/テストベースの確保
 - テスト自動化のためのテストビリティの構築・維持
 - 自動テストのためのテストベースの確保・維持
 - 自動化インフラの実現
 - 自動テストを支えるインフラの構築・維持
 - テスト自動化を支えるマネジメントの実現・維持

「テスト自動化が成功した」とは

妥当な対価で目的を達成すること

テスト自動化の成功を支える活動

テスト自動化を成功させるためには？

妥当な(対価)で(目的)を達成すること

1. 適切な目的をたてる

2. 費用対効果を改善する

テスト自動化の適切な目的を立てる

第一部

テスト自動化の成功

第二部

テスト自動化を支える活動

適切な
目的設定

テストバリエーション
確保

テストの
妥当性確保

テストの
内部品質確保

計画とアプ
ローチの工夫

テストの
活用拡大

自動化インフラ
への統合

監視とコントロー
ルによる改善

第三部

テスト自動化を支えるチームと仕組み

チーム

自動化
インフラ

プロセス

習慣化

テスト設計

テスト自動化
システム

テスト自動化を支えるチーム文化

経験的に多いテスト自動化失敗要因: 不適切な目的や目標の設定

チームの成功につながる目的を選ぶ

- テストのニーズ・シーズ・制約、チームの状況に応じて：
達成できる目的が変わる
テスト自動化のリスクや費用対効果が変わる
- その時のチームにとって有益で実現可能な目的を設定するのが重要

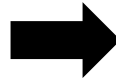
例：

【対象】 サーバクライアントシステムのスマートフォンアプリ開発

【ニーズ】 サーバサービスのリグレッションを検出したい

【制約】 UIは未確定で変更が多い

【チーム保有の技術】 サーバの実行環境は仮想化対応見込み



【有効で成功が見込める自動化】
サーバAPIのリグレッションの継続的検出

【失敗しやすい自動化】
UIに依存するEnd to Endのふるまいの品質保証のコスト削減

広い活動スコープで目的を設定する

- テスト自動化活動のスコープを広げると：
より本質的・効果的な目的を選択できる
目的達成に使える手段や関係者が増える

テスト担当に限定した視点

【取りえる目的の例】特定のテスト実行工程の費用対効果の改善

【取りえる手段の例】テスト担当の作業

開発+テストに広げた視点

【取りえる目的の例】開発の迅速化、Developer eXperienceの向上、全体のQAの最適化

【取りえる手段の例】テストバリエーションの向上、テスト・開発の連携協力

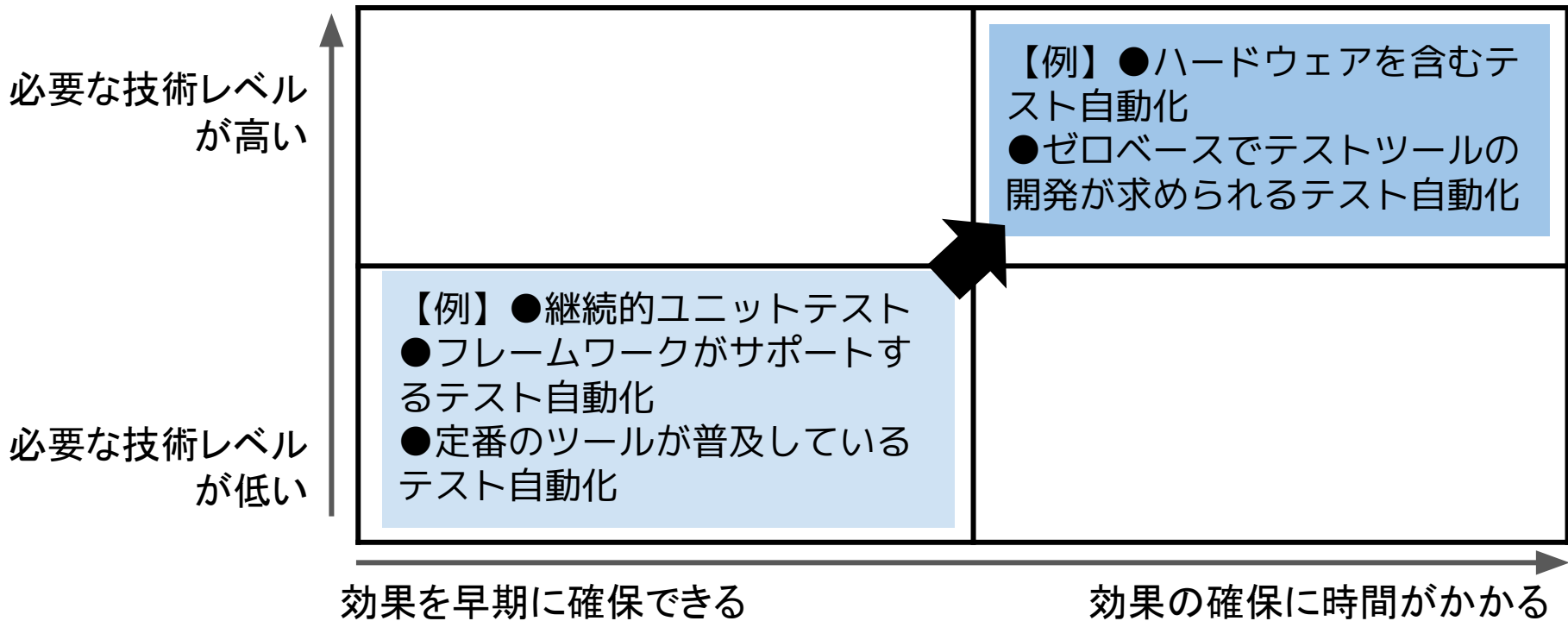
ビジネスを巻き込んだ視点

【取りえる目的の例】ビジネス要求への迅速な追従、高度な製品価値の実現

【取りえる手段の例】ユーザとの連携、プロダクトライフサイクルを通じた継続的改善

ステップバイステップで目的達成を目指す

- テスト自動化活動を継続し、テスト自動化の成功や技術を蓄積すると、より難しい目的に対応できるようになる。持続可能性が重要



テスト自動化を成功させるためには？

妥当な(対価)で(目的)を達成すること

1. 適切な目的をたてる

2. 費用対効果を改善する

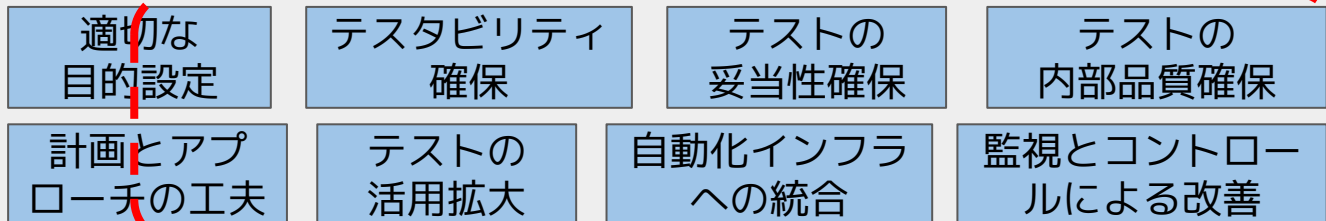
費用対効果を改善する

第一部

テスト自動化の成功

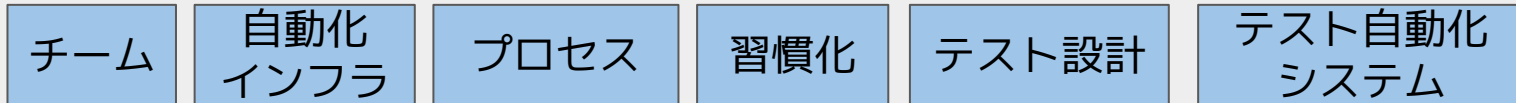
第二部

テスト自動化を支える活動

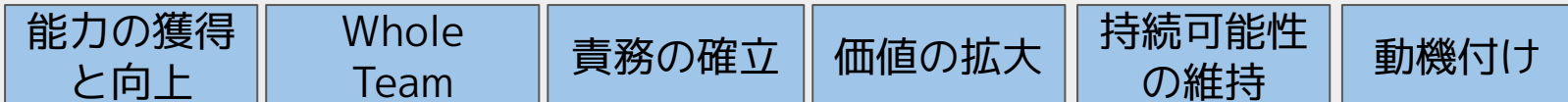


第三部

テスト自動化を支えるチームと仕組み



テスト自動化を支えるチーム文化



自動テストにとってのテストビリティ（試験性）を高める

品質特性	テスト自動化にとっての特性	実装例
実行円滑性	円滑に自動化された処理を実行できる	シンプルなセットアップ手順
観測容易性	自動テストがテスト対象の出力(間接出力、副作用含む)を観測しやすい	エラー認識手段の充実、適切なログ設計、十分な副作用対策(ステートレスなど)、観測用デバッグ機能
制御容易性	自動テストがテスト対象の入力(間接入力含む)を操作しやすい	使いやすいAPI、間接入力の制御手段(Stubなど)、デバッグサポート
分解容易性	対象から自動テスト可能なテスト対象を切り出しやすい。自動テストの障害を分離しやすい	適度に配置された接合部(DI、Dependency lookup)、適切な結合性設計
単純性	テスト対象の仕様や構造、実行方法が単純	冗長性の少ない仕様やコード
安定性	自動テストの支障となるテスト対象のバグや不安定さ(変更が頻繁など)が少ない	テスト対象のバグの少なさ、仕様変更の少なさ、IFの不変さ
理解容易性	テスト対象の仕様や構造、実行方法が分かりやすい	理解しやすい仕様やコード、適切なドキュメンテーション

※品質特性定義: ロジャー・S・プレスマン『実践ソフトウェアエンジニアリング(日科技連出版)』

経験的に多いテスト自動化失敗要因: テスタビリティ不足

テストの妥当性を高める

- 自動化する価値のあるテストケースを設計し、その価値を維持する
- **価値のある妥当な**テストケースを実現するための要点：

【テスト設計の保守】

テストの要求や制約が変化する開発ライフサイクルの中で、自動テストが有効であり続けるようにテスト設計を保守する

【テスト設計】

適切なテスト設計アプローチで、責務を効率的に充足する自動テストのテストケースを設計する

【テストレベル設計/テスト基本設計/テストアーキテクチャ設計】

自動テストが価値を発揮できるように、テスト戦略を構築し、テスト全体の責務設計を工夫する

【テスト要求分析】

広くステークホルダ(ユーザ、チーム、PjM、PdMなど)と協力し、テストについてのニーズ・シーズ・制約を広く掘り出し、実現すべき要求を識別する

自動テストの内部品質を作りこむ

- 一般的なソフトウェア開発のエンジニアリングやノウハウの活用でテストシステム・テストコードの品質を作りこむ。要点となる品質：

品質特性	確保すべき特性
保守性	自動テストの保守を容易にする ※監視容易性(問題の監視が容易)、試験性(正しいか・妥当かの確認が容易)、解析性・修正性(問題発生時の特定・是正が容易)、可読性(TaDとして運用可能)
信頼性	偽陽性/偽陰性の削減、安定性の確保を容易にする ※継続的自動テストはFragile Test/Flaky Testとの戦いに直面する
性能効率性	フィードバックを迅速化する。並行実行や頻繁な変更時実行が可能なように必要リソースを削減する
拡張性	少ない対価で自動テストのカバレッジを拡張可能にする 例:パラメータ化テスト、データ駆動テスト、環境仮想化、モデル駆動テスト
使用性	チームにとっての使いやすさや魅力を向上する 例:フィードバックの手軽さ、結果の見やすさ

経験的に多いテスト自動化失敗要因: 内部品質不足による保守コスト悪化

テスト自動化の計画とアプローチを工夫する

- 見積もりと戦略策定・計画づくりで、テスト自動化の適切な段取りを構築する。方向づけ・コントロールで段取りの実行を推進する
 - リソース（人・物・金・時間）の選定と確保
 - 適時に自動テストを投入できるように収集
 - テスト自動化のリスクのコントロール
 - プロトタイピングや反復開発による早期の学習機会の確保
 - 関係者の巻き込みと連携の管理
 - テスト自動化活動のマネジメント

経験的に多いテスト自動化失敗要因：段取りの悪さ

自動テストの活用の頻度と幅を広げる

- 自動テストの資産は様々な関係者が様々な用途で活用できる
 - 例：自動化に成功した結合テストの用途
 - for 開発者
 - ダブルループのテスト駆動でプログラミングを主導
 - CIに組み込み、開発中のリグレッション発生を監視
 - PR受け入れやパイプライン通過の受入基準に利用
 - for 後工程のテスト担当者
 - スモークテストや、自身のテストの補完手段として利用
 - for 後続プロジェクト
 - Test as Documentationとして保守に利用
- テスト自動化の活用者を広げ、様々な場面で自動テストを活用することで、テスト自動化の費用対**効果**にレバレッジをかけることができる
 - 前提：利用者を広げる自動テストの魅力確保が必要
 - 例：納得感の向上、分かりやすいフィードバック

自動化インフラに自動テストを統合する

- 開発を支える自動化インフラに自動テストを組み込む。
自動化インフラで自動テストの保守や改善効果を底上げする
 - デプロイメントパイプラインへの統合による開発のインテグリティの確保。
自動テストの責務の確立
 - 自動テストの構築・維持に有用なインフラ機能の活用
 - 自動テストのテストウェアの管理
 - テストシステムやテストコードの構成管理
 - 自動テストの品質保証
 - テストコードの動的解析・静的解析、テストシステムのテスト
 - 自動テストの実行
 - 開発イベントに合わせた、効率的な自動テストの実行管理
 - 変更ごとの実行/並列実行
 - 自動テストの監視と問題是正のサポート
 - カバレッジ計測、偽陽性・偽陰性の監視、Flaky Testの監視

監視とコントロール、保守で自動テストを維持・改善する

- 自動テストの品質や成果を監視し、問題検出と是正を迅速化して、効果を確保する
自動テストを保守し、自動テストの効果を維持・改善するように務める
- 問題検出手段
 - 事前の問題検出
 - プロトタイピング/トライアル、反復開発
 - レビュー/シナリオウォークスルー
 - バグをどのように検出するか/仕様項目をどう保証するか
 - テスト構築時の問題検出
 - レッド→グリーン of テスト駆動サイクル
 - バグの注入（フォールトインジェクション）
 - 運用中の監視
 - 擬陽性の継続的監視
 - テストカバレッジの評価
 - 欠陥流出の評価
 - 事後の評価
 - 目的達成指標ごとの評価
 - 例：欠陥流出率/カバレッジ/費用対効果の事後評価/ステークホルダの主観評価

テスト自動化を支える活動

第一部

テスト自動化の成功

第二部

テスト自動化を支える活動

適切な
目的設定

テストバリエーション
確保

テストの
妥当性確保

テストの
内部品質確保

計画とアプ
ローチの工夫

テストの
活用拡大

自動化インフラ
への統合

監視とコントロー
ルによる改善

第三部

テスト自動化を支えるチームと仕組み

チーム

自動化
インフラ

プロセス

習慣化

テスト設計

テスト自動化
システム

テスト自動化を支えるチーム文化

能力の獲得
と向上

Whole
Team

責務の確立

価値の拡大

持続可能性
の維持

動機付け

テスト自動化の成功を支えるチームと仕組み

テスト自動化の成功を支えるチームと仕組み

- テスト自動化の成功要素の実現には、チームと仕組みによる下支えが不可欠
- 例：テストバリエーションの確保
 - 【チーム】
 - 開発とテストがお互いのために連携・協力する体制
 - テストバリエーションを実現・維持するためのチームの基礎開発力
 - テストバリエーションを実現するためのパターンやイディオムの実践
 - テストバリエーションを維持・改善するためのリファクタリングや監視
 - 【プロセス】
 - 設計段階でもテストを考慮するプロセス
 - テスト担当者によるレビュー支援
 - テストから開発へのテストバリエーションのフィードバックを確保するプロセス
 - 【開発インフラ】
 - テストバリエーションの確認・維持を行うインフラ
 - テストバリエーションを支えるルールの静的解析

テスト自動化のチーム・仕組みを支えるチーム文化

チームと仕組み

マネジメントシステム、プロセス、パイプライン、
インフラ・支援サービス、テストシステム、ルールと体制

チーム・仕組みを生み出す



文化を誘導する

チーム文化

ビジョン・目的、価値観、リテラシー、習慣、環境、人、関係性、学習と進化

テスト自動化を支えるチーム文化

第一部

テスト自動化の成功

第二部

テスト自動化を支える活動

適切な
目的設定

テストバリエーション
確保

テストの
妥当性確保

テストの
内部品質確保

計画とアプ
ローチの工夫

テストの
活用拡大

自動化インフラ
への統合

監視とコントロー
ルによる改善

第三部

テスト自動化を支えるチームと仕組み

チーム

自動化
インフラ

プロセス

習慣化

テスト設計

テスト自動化
システム

テスト自動化を支えるチーム文化

能力の獲得
と向上

Whole
Team

責務の確立

価値の拡大

持続可能性
の維持

動機付け

テスト自動化の成功を支えるチーム文化の志向性

1. チームのテスト自動化の能力を確保し高める
 - 効率性を高める。有効性を高める。ブレークスルーに必要な力をためる
2. Whole Teamでテスト自動化を推進する
 - 開発者やドメインエキスパート、マネージャも連携して自動化する
 - 例：テストバリエーションの確保
3. チームでの自動テストの責務を確立する
 - 全体の開発活動の中での自動テストの責務を確立し、責務を果たす
4. チームにとっての自動テストの価値を高める
 - チームでテスト自動化のニーズ・シーズをつかみ実現する
 - チームでテスト自動化の価値を維持し・高める
5. チームで自動テストの持続可能性を保つ
 - 保守・拡張・進化が容易な状態を維持する。
習慣的に技術的負債を返済する
6. チームでテスト自動化の動機付けを行う
 - テスト自動化の価値を示し、主体的・自発的なテスト自動化を促す

テスト自動化の成功を支える仕組み

第一部

テスト自動化の成功

第二部

テスト自動化を支える活動

適切な
目的設定

テストバリエーション
確保

テストの
妥当性確保

テストの
内部品質確保

計画とアプ
ローチの工夫

テストの
活用拡大

自動化インフラ
への統合

監視とコントロー
ルによる改善

第三部

テスト自動化を支えるチームと仕組み

チーム

自動化
インフラ

プロセス

習慣化

テスト設計

テスト自動化
システム

テスト自動化を支えるチーム文化

能力の獲得
と向上

Whole
Team

責務の確立

価値の拡大

持続可能性
の維持

動機付け

テスト自動化の能力を持つ人とチームの確保

- テスト自動化の能力を持つ人材を確保
 - テスト自動化人材のスキルモデルやスキル評価手段の確立
テスト自動化能力を持つ人材の獲得
 - 主要スキル：
開発技術、開発インフラ技術、テスト技術、**テスト自動化技術**
- チームを構築
 - 要となるロールの確保
 - テストアーキテクト/テストアナリスト
 - 品質保証・テスト全体を視座に、あるべきテスト自動化へ先導
 - SET/SETI
 - 開発技術を活かして、テスト自動化およびテスト自動化インフラに注力
 - 必要なコラボレーション手段（例：開発者とテスト担当者）の仕組み確保
- チームの能力の維持・向上の仕組み化
 - プロトタイピングといった学習サイクル確保

経験的に多いテスト自動化失敗要因：能力不足（特に開発力）

【テスト自動化の成功を支えるチームと仕組み】

自動化インフラの整備

- 自動テストの統合先として、開発の自動化インフラを構築
 - 開発全体のデプロイメントパイプラインを自動化
 - パイプラインへの自動テストの統合
- 開発インフラで、テスト自動化活動をサポート可能にする
 - プロセスのサポート：自動化された一貫性あるプロセスのパイプライン化
 - テstabiリティ確保のサポート：
各種静的解析やテストカバレッジ評価手段の提供
 - 自動テストの運用のサポート：
テスト結果の監視と報告
Flaky Testなどの自動テストの問題検出と復帰
 - テスト構築のサポート：
モデルベーステストや、仮想化などで自動テストをサポート

【テスト自動化の成功を支えるチームと仕組み】

テスト自動化のプロセスの整備

- テスト自動化の良いアプローチや要点をプロセス化してチームに定着させる
 - テスト自動化の能力を伸ばす学習機会の確保
 - リスクをコントロールするためのフィードバックサイクルの確保
 - プロトタイピングや反復開発
 - 適切なテスト自動化アプローチのプロセス化
 - 適時・適切なツール選定など、必要作業のプロセス化
 - Whole Teamアプローチの活動のプロセス化
 - Wモデルなど各工程でのテストの考慮
 - 必要なテスト自動化活動のクライテリア化
 - テストコードの品質基準の設定

【テスト自動化の成功を支えるチームと仕組み】

テスト自動化の良い習慣の定着

- 作業・チェックの自動化、作業のルール化、クライテリア運用を通して、テスト自動化を支えるグッドプラクティスをチームの習慣として定着させる
- 自動テストコードの良い習慣の例：
 - FIRSTの原則
 - Fast、Independent、Repeatable、Self-Validating、Timely
 - xUTP12の原則
 - 「Googleのソフトウェアエンジニアリング」ユニットテストの原則
 - 変化しないテストを目指せ。
 - 公開API経由でテストせよ。
 - 相互作用ではなく、状態をテストせよ。
 - テストを完全かつ簡潔にせよ。
 - メソッドではなく、挙動をテストせよ。
 - 挙動に重点を置いてテストを構成せよ。
 - テスト対象の挙動にちなんでテストに命名せよ。
 - テストにロジックを入れるな。
 - 明確な失敗メッセージを書け。
 - テスト用コードを共有する場合、DRYよりDAMPに従え。

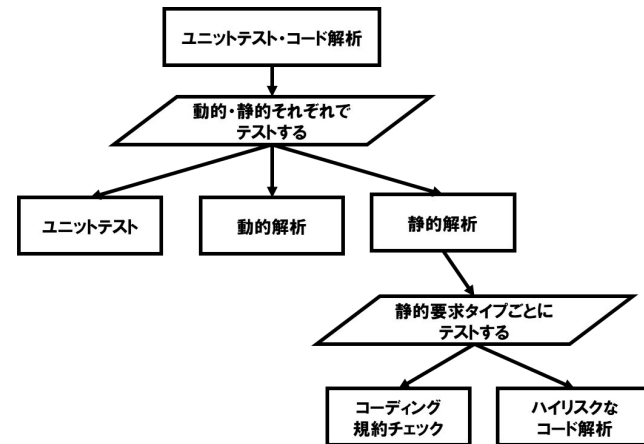
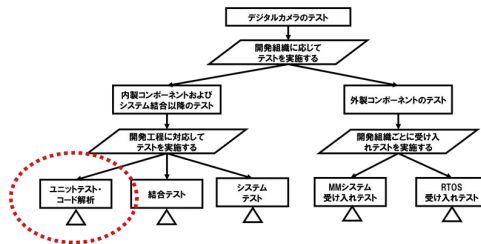
「Googleのソフトウェアエンジニアリング」(オライリー、Titus Winters他)

【テスト自動化の成功を支えるチームと仕組み】

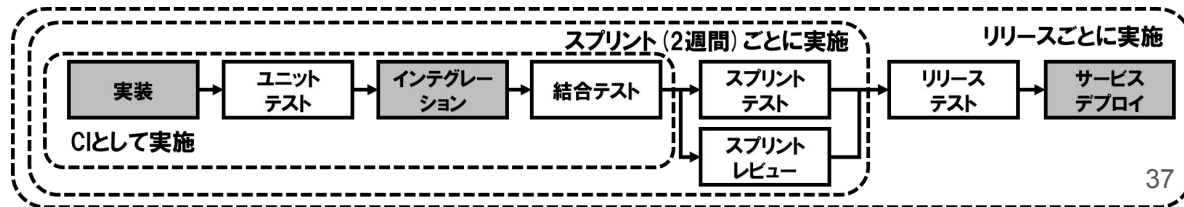
品質保証・テストの全体設計

- 全体の品質保証・テストの設計構造を構築・維持し、開発ライフサイクルの中で自動テストのあるべき責務を先導する
 - ある仕様項目の実現はだれが確認するか、あるバグの検出はだれがやるか

品質保証・テスト活動の責務設計



品質保証・テスト活動の構造設計



テスト自動化システムの作りこみ

- テスト自動化システムを整備し、テスト自動化活動を支える
 - テストスクリプトの実装支援
 - PythonやJavaScriptなどのエコシステム利用の実現
 - GherkinなどのBDDサポート
 - データ駆動テスト、パラメータ化テストといった構造化支援
 - xUnitやSpec系といったテストスクリプトの基本構造の指定
 - 自動テストを支援するユーティリティ機能の提供
 - 監視、レポート、テストダブル（スパイ手段など）、セットアップ・後始末
 - テスト自動化のアーキテクチャやフレームワークの指定
 - テストウェアやテストデータの効率的な管理手段の提供

テスト自動化の成功を支えるチームと仕組み

- チームと仕組みを整備することで、テスト自動化を成功させる活動が容易になる。チームと仕組みによる基礎作りを継続することで、テスト自動化の成功可能性を高められる

まとめ

第一部

テスト自動化の成功

第二部

テスト自動化を支える活動

適切な
目的設定

テストバリエーション
確保

テストの
妥当性確保

テストの
内部品質確保

計画とアプ
ローチの工夫

テストの
活用拡大

自動化インフラ
への統合

監視とコントロー
ルによる改善

第三部

テスト自動化を支えるチームと仕組み

チーム

自動化
インフラ

プロセス

習慣化

テスト設計

テスト自動化
システム

テスト自動化を支えるチーム文化

能力の獲得
と向上

Whole
Team

責務の確立

価値の拡大

持続可能性
の維持

動機付け