



JaSST'13 Hokkaido

Tutorial

テスト自動化分析

- 頼れる大人のテストエンジニアリングと金勘定 -



○ 松木 晋祐 @snsk

株式会社ACCESS ソフトウェア・ソリューション本部

技術戦略課 課長 兼 東京リージョン品質保証責任者/テスト自動化研究会(STAR)お世話係/
JaSST東京実行委員/ASTER会員 ほか ※著作、出力は snsks.net にて。

派遣テスターから品証責任者までテスト、QAにまつわるひと通りのロールを経験。現在国内の品質番長及び“技術力”的向上をCEO、CTO指揮下にて推進中。



○ 後藤 香織

株式会社ベリサーブ 東日本第二事業部

大手Sierの常駐エンジニアを経て、システム検証エンジニアに転身。Web系のドメインを中心に活動中。



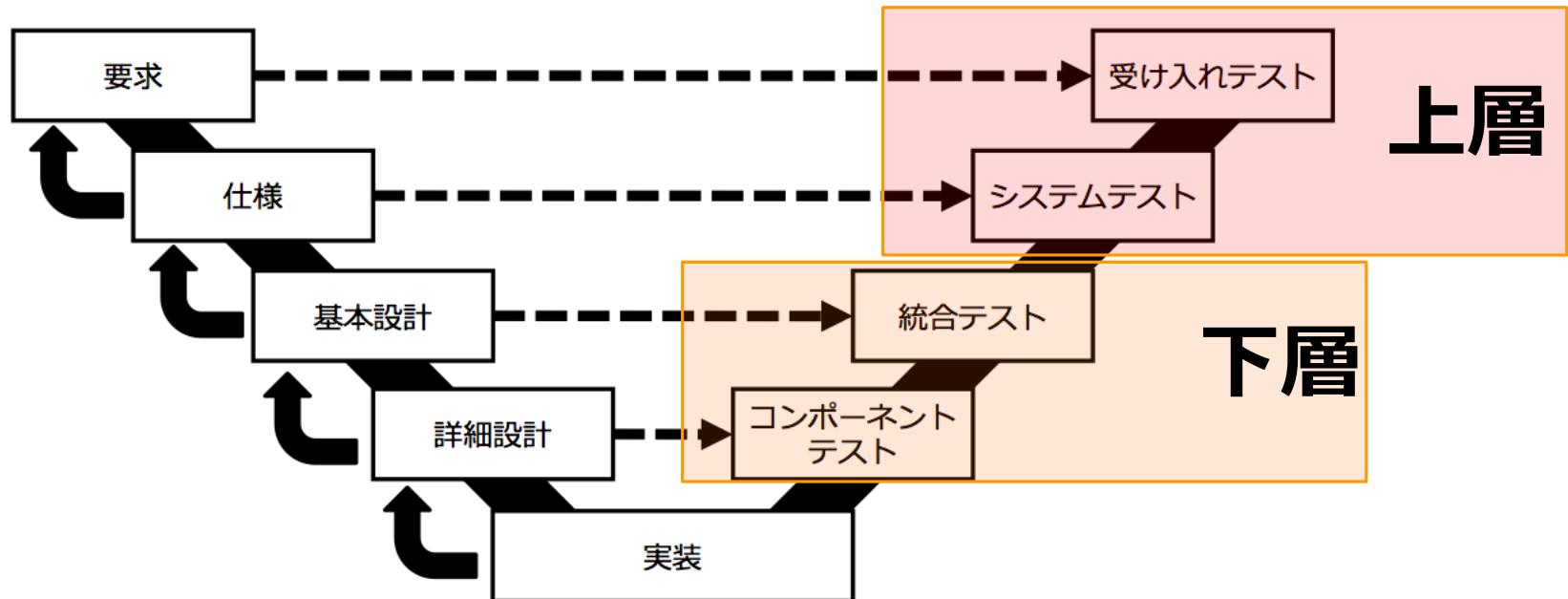
- システムテスト工程以降で利用できる自動テスト技術の解説(30分) ★いまここ
- ハンズオン(60分)
 - システム、プロジェクト期間、プロセスの状況、全部手動でやる場合のアーキテクチャ、テストタイプ、コストを提示
 - ハンズオン成果物は2つ
 - 全て手動で計画されているテストの自動化計画案
 - それによってどのコストがどのように変わるかの見解
- ハンズオン成果の発表(20分)
- 講評＆クロージング(10分)

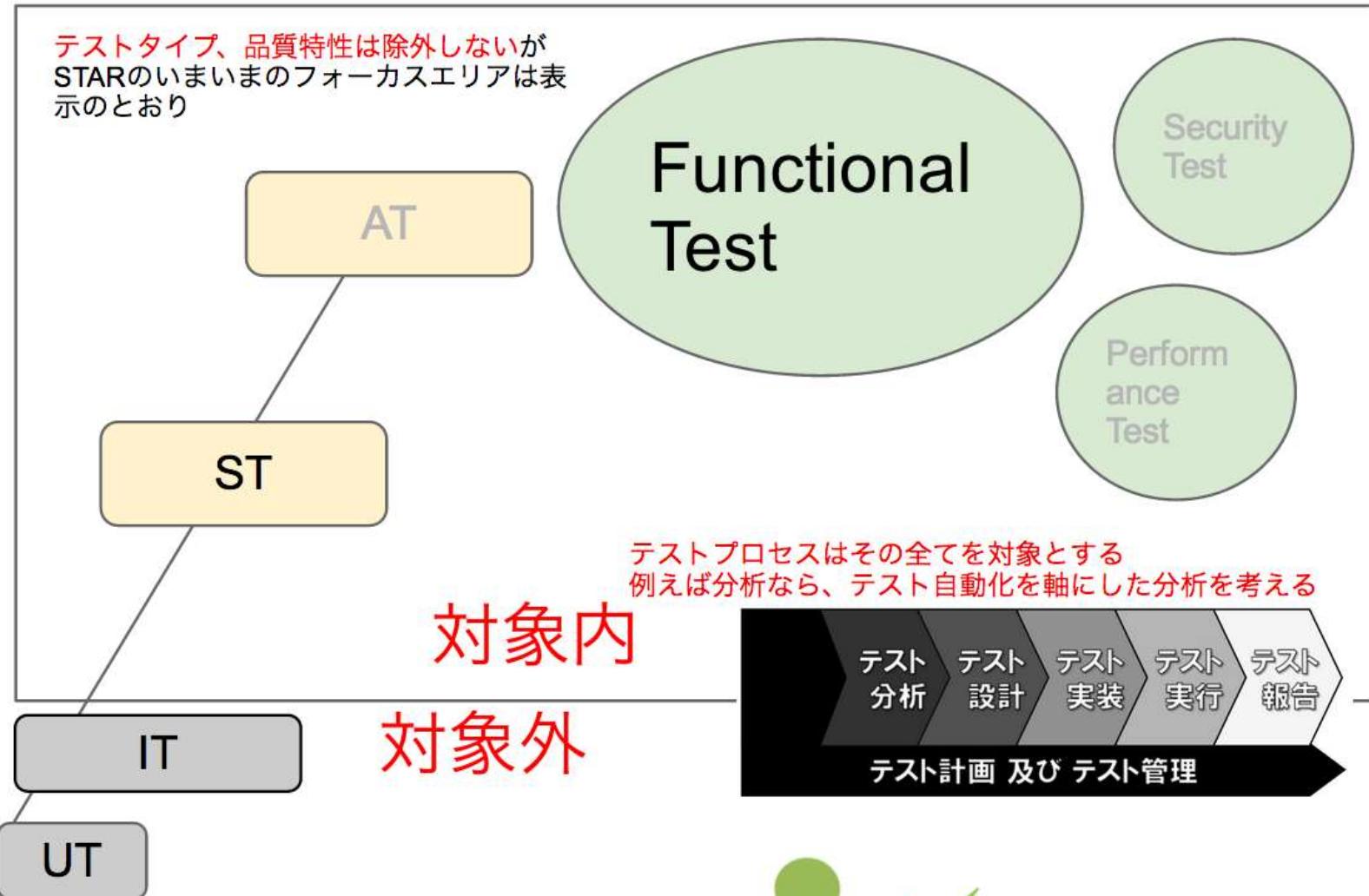
- プロジェクト概要書
- 画面モックアップ
- スケジュール・開発&テストプロセス
- テスト分析＆計画
- 計算用紙

テスト自動化技術概論



- おおまかにV字モデル図上の下層と上層に分かれる
- 下層のテストはFWが充実しており、特に意識しなくても自動で実行される
- 本日のターゲットは上層



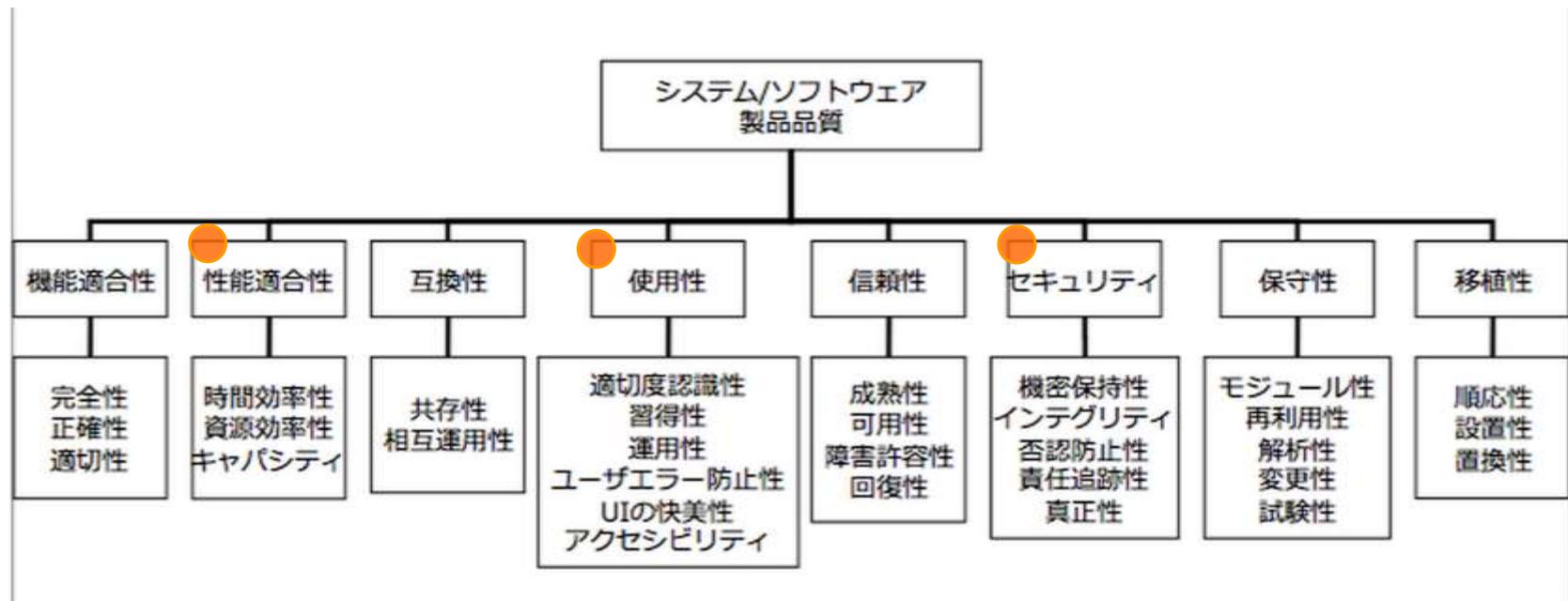


上記はサービス、価値の完成度で定義されるテストレベルを指す。
ST以降は、サービス、価値が提供可能な対象を扱うレベルである。

○ 上層のソフトウェアテスト技術自体の特徴

- 扱うべき品質特性の幅が下層と比べて大きく拡がる
- 検出できるのは「故障」
 - エラー → 欠陥 → 故障 → インシデント
 - 根本の欠陥の修正までもう1ステップ掛かる

● は特にシステムテスト以降でテストすべき特性



○ テスト自動化技術特有の特徴

- 「チェック」まで厳格化したテストケースのみが自動化できる
- テスト「実行」工程のコストが大幅に下る
 - が、テストタイプに拠って下がりだすサイクルが異なる
- テスト「実行」「報告」工程の正確さが格段に向上する
 - コストにフォーカスすると見えにくいが大きな特徴

● コストが下がるのはこの領域のみ



○ テスト自動化技術特有の課題

- 上層のテストは近年FWが整いつつあるが、まだまだ技術的ハーダルは高い
 - プログラミング技術が事実上必須となるため、素養を持たないスタッフの転換に多少コストが掛かる
 - 一方、キーワードのみでテスト記述を構成し、プログラミングが必要な駆動部分と分離することでこれを解決するアプローチもある
 - キーワード駆動、アクションベースド
- UIの変更に弱く、メンテナンスコストが長年課題とされてきたが、オブジェクトマッピング技術が浸透するに従い解決しつつある

テスト自動化10の誤解



テスト自動化10の誤解



#	内容
1	テスト自動化ツールで壊れた(開発)プロセスを修復できる。
2	自動化テストで多数の新たな欠陥が見つかる。
3	自動化テストは、手動テストを置き換える。
4	レコード&プレイバックは効果的なテスト自動化のアプローチである。
5	一つのツールですべてのテストフェーズを統合できる。
6	テスト自動化は、100%の自動化が、テスト自動化成功を意味する。 (→100%自動化すべきである)
7	テスト自動化ツールは直ちにテストスケジュールを短縮できる。
8	自動化のコスト=ツールソフトウェアのコスト+テストスクリプト開発コスト (メンテナンスコスト等がぬけている)
9	単位時間あたりのテスト実施時間が極端に短くなるわけではない。場合によっては遅くなる場合もある。
10	テストの自動化は、テストスクリプトを実装することだけだと考えている。 (それだけを工数に含める)

- 自動化できるのは“チェック”まで落とし始めた項目のみ
 - チェックを規定する課程で大抵の大きな不具合は発見できる
 - これはテスト設計の時点で大抵の不具合が見つかるのと同じ
- 自動化されたテストが新たに発見する不具合の多くは変更によるデグレード(=機能低下)
 - これは欠点ではなく、**変化し続けるソフトウェアが備えていなければならぬ必須の装置である**ことを示す重要な特性
- とは言え、テストデータの大量生成と投入に拠る網羅性の向上とそれにまつわる新たな不具合の発見には期待できる

○ レコード&プレイバック=キャプチャ & リプレイ

- 操作を記録、再生し、一定のポイントで検証を行うアプローチ
- 大抵のケースではメンテナンスコストを抱えきれなくなる

テスト自動化の3世代(出典: TABOK)

第3世代

キーワード駆動 / モデルベースドテスト

2013年
時点では
この辺

第2世代

機能分割 / データ駆動

第1世代

Liner Script / レコード&プレイバック

#9 単位時間あたりのテスト実施時間が極端に短くなるわけではない。場合によっては遅くなる場合もある。

- オブジェクトマッピングを用いた場合でも「実際の操作」は発生するため、実行の速度は充分に熟練したテストオペレータのそれと大差ない
 - むしろプログラミングされたテストであるがゆえの融通の訊かなさの方が目立つ
- ゆえに機能テストにおける1日あたりのテスト実行“項目数”がオペレータの3倍以上になることは考えにくい
- 機能テスト以外の特性についてはこの限りではない

#10 テストの自動化は、テストスクリプトを実装することだけだと考えている

- テスト設計までのフェーズは通常のテスト工程とまったく同じ
- 通常のテスト工程に加えて下記の工程が必要になる
 - (1)駆動、判定、報告系のFW導入、運用
 - (2)テストスクリプトプログラミング
 - (3)テストスクリプトのメンテナンス

● 再掲：コストが下がるのはこの領域のみ

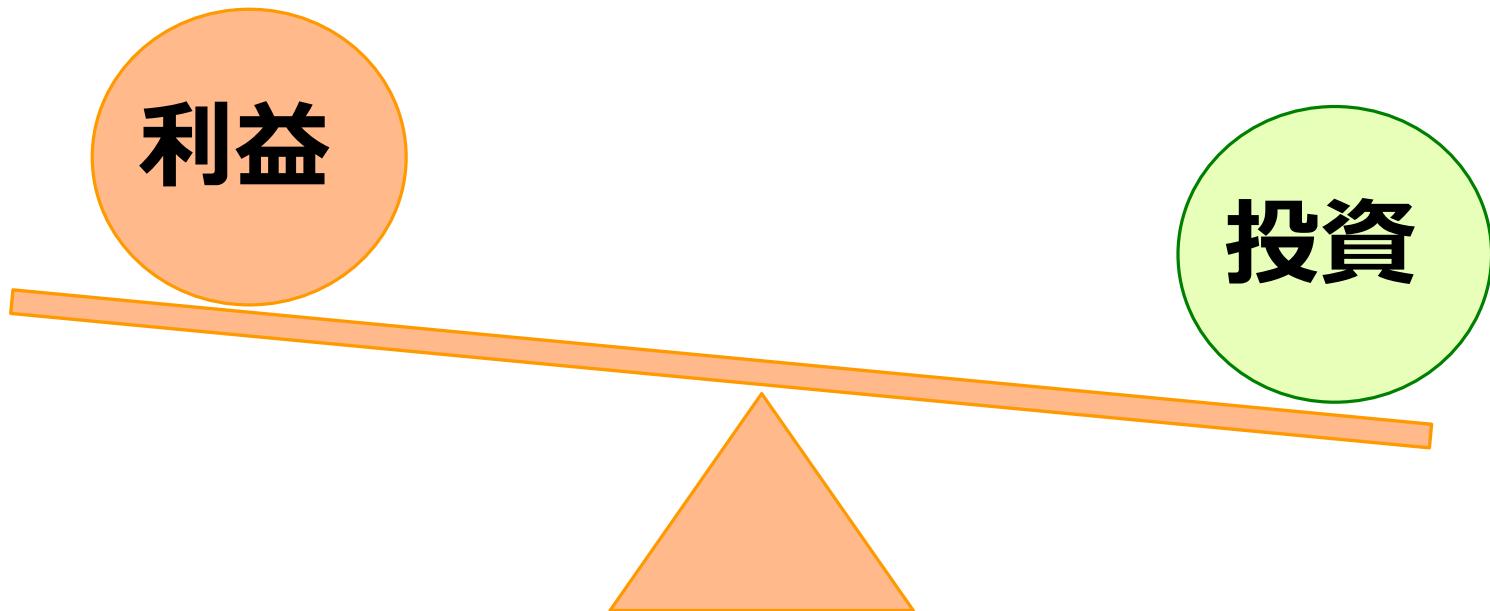


テスト自動化のROI



- ROI = Return On Investment

- 投資収益率
 - 投資によってどれだけの利益が生まれたか？



- ツール/テストプロセス/人/プロダクトについて、時間軸で考える

	導入前～導入プロジェクト	導入後～ソフトウェアライフサイクル全般
ツール	ツール導入のコスト	サポートやライセンスなど保守契約に要するコスト
テストプロセス	スクリプト作成の負荷	欠陥修正後の確認テスト実行にかかる時間
人	導入に必要なスキルのトレーニングコスト	スクリプトのメンテナンスのための技術伝承のコスト
プロダクト	自動テスト可能なつくりにするためのコスト	新技術の導入等により利用技術の変化に対応するコスト

出典: TABOK 1.3 Automation Return-on-Investment(ROI)

© 2013 ACCESS CO., LTD. All rights reserved.

- 効果はテストの活動においてどのような便益が得られそうかを考える

テスト効果	内容
テスト効率性 (=テスト費用)	通常人間の3倍稼働可能であることによる俯瞰的な速度向上(単位時間ではない)、コスト削減
テスト網羅性、 実現性	データ駆動以降の仕組みの導入に拠る現実的に実行可能なデータの幅の拡張、及び単純に人手では不可能な規模でのテスト実行が可能になる
テスト正確性	コンピュータが実行することによるテスト手順実施誤り、 テスト結果記入誤りの撲滅が可能になる ※ただし、手順定義 자체を誤る可能性も看過できない

- 費用、時間、リスクの3つの観点で考える
- 費用
 - 投資と利益の費用、つまり費用の面から評価
 - 自動化前後を比較
 - ⇒カバレッジ向上などの費用以外の利点を見落とす
- 時間
 - 時間の投資と利益から評価、つまり効率向上の面から評価
 - 自動化前後を比較
 - ⇒手動テストでできず、自動化されたことではじめてできるようになったテストを計上しにくい
- リスク
 - 自動化しないことのリスクとリスクが損失に変わった場合のコストから評価
 - ⇒前後の比較が必要なく、他の2つの問題をカバーするが、精緻なリスク分析、損失計算が必要

出典: TABOK 1.3 Automation Return-on-Investment(ROI)

© 2013 ACCESS CO., LTD. All rights reserved.

- テスト自動化のROIはどの観点を使っても一長一短あるため、決定版、定石と呼べるものは無い
- そのため期間を区切り選択的、戦略的に便益を狙っていくことが重要
- テスト自動化のROIはいずれかひとつ、またはその組み合わせに絞って最大化し、観点は定期的に見直す必要がある

では、いよいよ実践です

