シフトレフトの
失敗事例から学ぶ次のアプローチ

JaSST23’Tokyo B7 テクノロジーセッション

2023年3月10日

株式会社AGEST

高橋 信弘

自己紹介

AGEST

名前

高橋信弘

所属

株式会社AGEST（AGEST, Inc.）

- お客様の品質課題を解決をする会社
- 2022年8月に事業承継の形で転籍
- 主にQA、テスト、改善をしている
- 平日子供との時間を取りたいとの思いから、残業を減らすべく業務効率化に関心大

≠

AGEST

AGEST

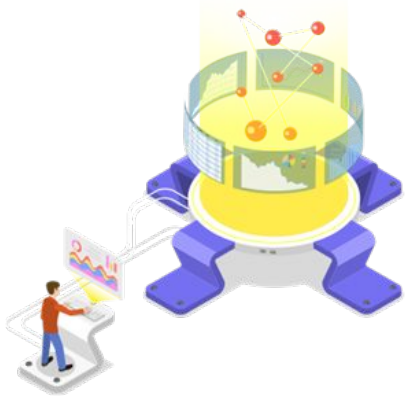
Agility / Agile
(機敏性 / アジャイル開発)

+

Test / Best
(ソフトウエアテスト / 最高品質)

— Mission

SAVE the
DIGITAL WORLD



— Vision

先端品質テクノロジーで、
すべてのDXに
豊かな価値と体験を

目次

1. シフトレフトできてますか？

2. シフトレフトとは

AGEST

JSTQBでは

3.早期テストで時間とコストを節約

早い段階で欠陥を見つけるために、静的テスト活動と動的テスト活動の両方をソフトウェア開発ライフサイクルのなるべく早い時期に開始すべきである。

早期テストは、シフトレフトとも呼ばれる。ソフトウェア開発ライフサイクルの早い時期にテストを行うことにより、コストを低減又は削減できる（**3.1**節を参照）。

https://jstqb.jp/dl/JSTQB-SyllabusFoundation_Version2018V31.J03.pdf

シフトレフトとは

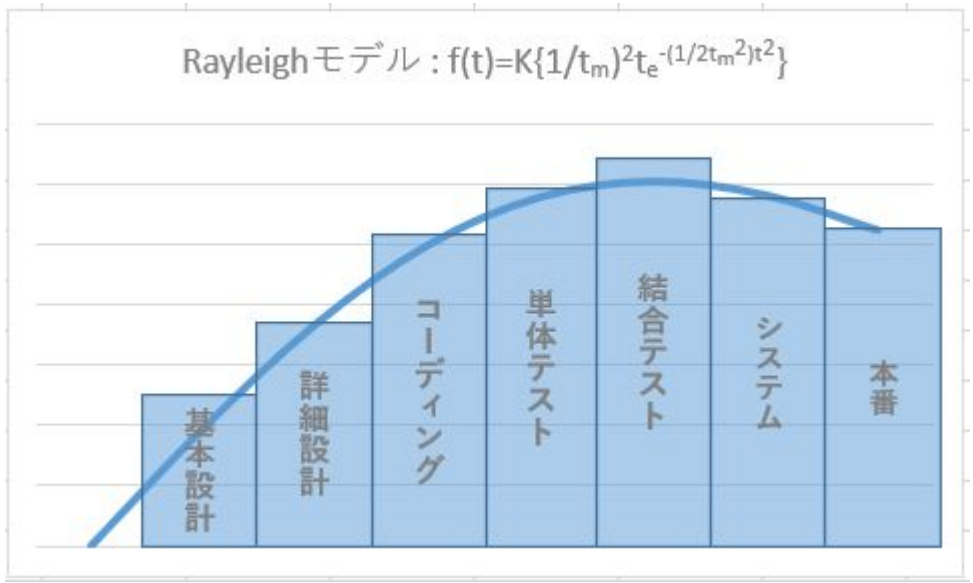
</Sqripts> (AGEST TecBlog)

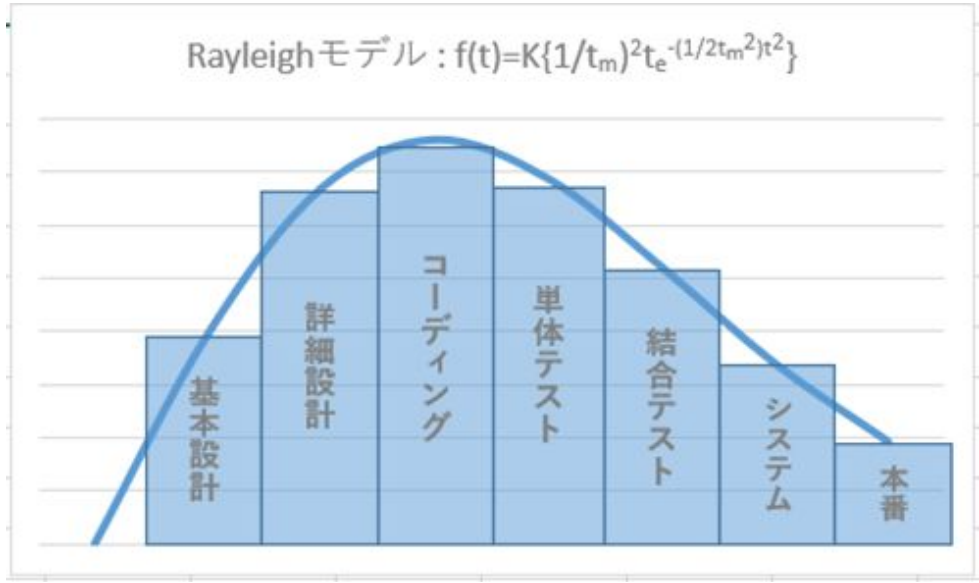
「いまさら「シフトレフト」について考えてみた」より

https://sqripts.com/2022/02/24/20438/

◆RayLeighモデル

障害除去を示した品質管理では有名な曲線



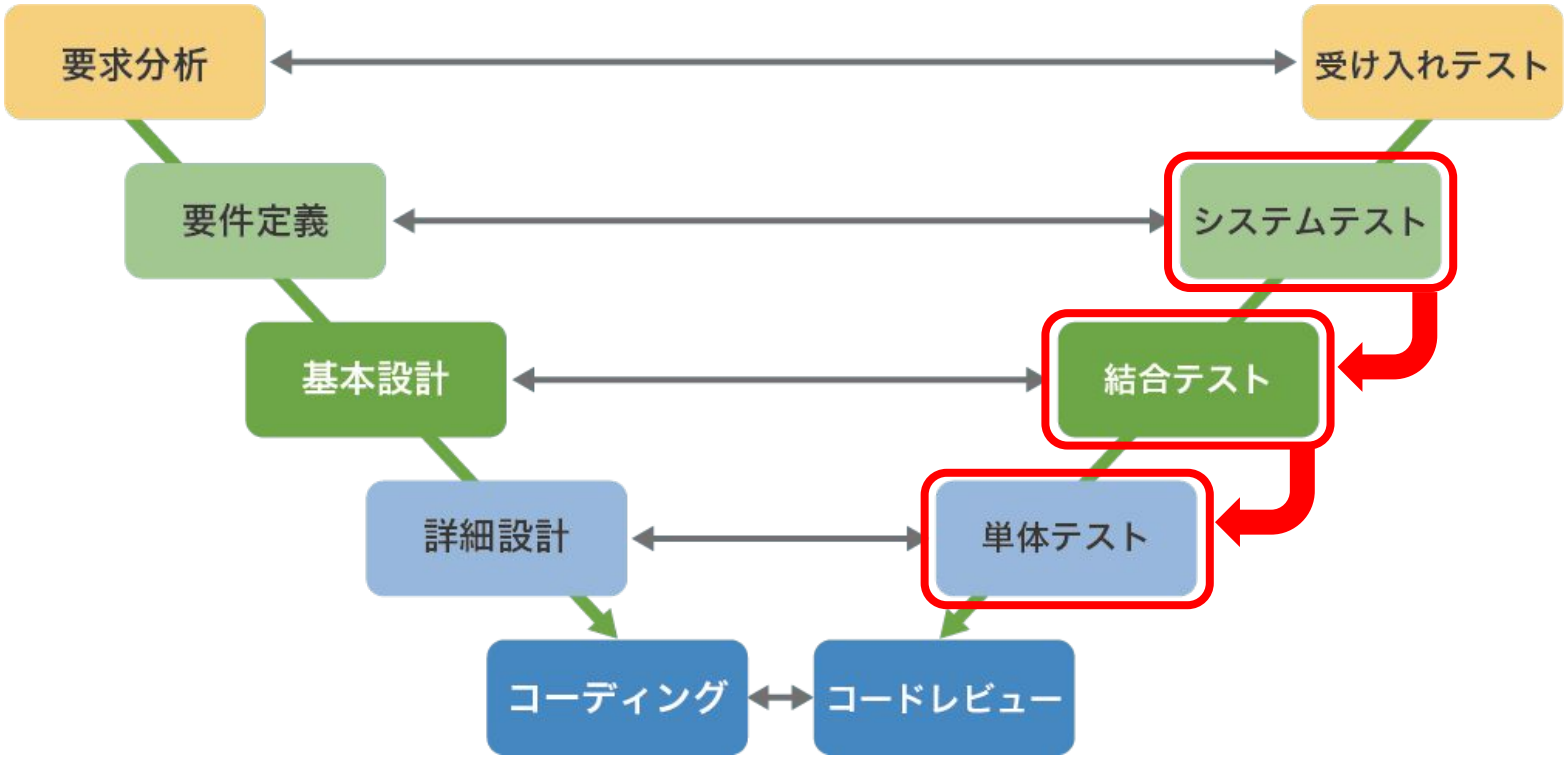


3. シフトレフト導入施策

アジャイル	ウォーターフォール

テストフェーズごとのテスト観点整理

導入難度：★★★　導入効果：★★



AGEST

背景・課題

1 大規模な現場

大規模開発で常時100名以上のメンバがいる現場

2 テスト観点が不明確

機能ごとに開発会社が分かれており、テストフェーズごとのテスト観点が明確でなかった

3 異なるテストを実施

A機能では結合テストで実施しているテストを、B機能ではシステムテストで実施していた
また機能ごとに実施しているテスト内容にかなり差があった

4 テスト関連資料が多数

現場にテストに関する資料が多々あり何を参照すべきか明確でなかった
これは会社に紐つくもの、部に紐つくもの、
誰が作ったかもわからない等、様々だった

導入施策

		テスト観点				
品質特性	テスト対象	テストタイプ	確認対象	テスト目的	テスト目的詳細	テスト内容例
機能適合性	画面	表示確認テスト	画面表示	画面レイアウトが設計書通りに表示するか確認する		
				画面の初期表示が設計書通りに表示するか確認する		
				画面の項目が設計書通りに表示するか確認する		
			帳票表示	帳票レイアウトが設計書通りに表示するか確認する		
				帳票の項目が設計書通りに表示するか確認する		
				印刷条件を指定して出力した帳票が設計書の仕様通りに表示するか確認する		
		操作性テスト	画面動作	画面が設計書の仕様通りに動作するか確認する		
		入力チェック		項目の属性が設計書の仕様通りに動作すること		

シフトレフト導入施策（テストフェーズごとのテスト観点整理）

AGEST

導入難度 ★★★

苦労した点①	苦労した点②	苦労した点③
資料集めが難航	**上位文書作成の手間**	**テスト観点承認の難航**
あちこちからテストに関する資料が出てきて、人によりテストベースとしている資料が異なっていた。またベースとなる版から独自進化を遂げており、最新版がどれかわからない	テストフェーズごとにやるべきことを定義するためにテスト観点を整理すると、テストフェーズごとのテスト計画書、全体テスト計画書と上位文書も作成していくことになっていった	作成したテスト観点の承認も難航 PM、PL、プロジェクト管理、機能ごとのリーダ、機能ごとのメンバ・・・何度も説明を繰り返し、指摘件数は膨大になる。 また指摘が相反するものも多々あり、調整に手間がかかった

シフトレフト導入施策（テストフェーズごとのテスト観点整理）

導入効果 ★★

単体テストフェーズでの不具合検出率が増加
各テスト工程でも不具合検出漏れが減少した

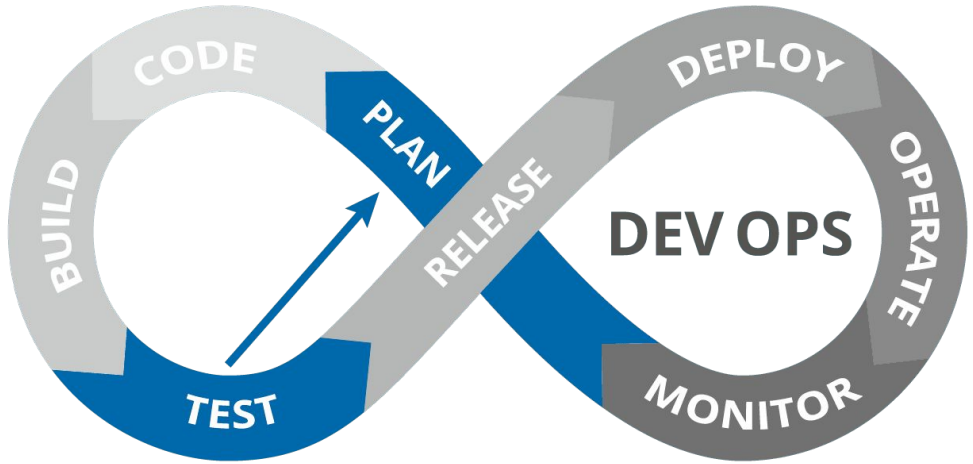
プロジェクトの規模が大きいので、全体的な効果（削減工数）が
大きかった

テストフェーズ内でのシフトレフトなので設計工程へまではシフト
レフトしていない

アジャイル	ウォーターフォール

QAによる仕様レビュープロセスの導入

導入難度:★　導入効果:★★



AGEST

背景・課題

1. レビュー**プロセス**は未定義
2. レビュー対象の**テストベース**は人により様々
   ⇒ストーリーボード、仕様資料、画面設計等
3. レビュー**タイミング**はテスト設計時
   ⇒レビュー指摘ではなく質問形式
4. 指摘内容は**個人のスキル**に依存
   ⇒ドメイン知識の有無、該当機能のテスト経験有無　等
5. 上流工程での**レビュー経験が少ない**メンバーがいる

AGEST

導入施策

施策①

sqip「レビューオリエンテーションキットを用いた、育成によるレビュー文化の形成」より、仕様レビュープロセスを導入

⇒レビュー対象、レビュー観点表、レビュー実施タイミング、
指摘の取扱い　等

施策②

仕様レビュー練習教材の作成

⇒既存不具合を仕様書に
埋め込み、練習教材とする

https://www.juse.or.jp/sqip/workshop/report/attachs/2011/SQiP3-1.pdf

シフトレフト導入施策（QAによる仕様レビュープロセスの導入）

導入難度　★

スムーズに導入出来た理由①	スムーズに導入出来た理由②	スムーズに導入出来た理由③
元々現場にシフトレフトの意識があった	**目的・効果の事前共有**	**開発・QAの協力体制**
プロセスはなかったが、効果的な仕様レビューが出来ているメンバーはいた また元々現場でシフトレフトの意識があったので、仕様レビュープロセスの導入に抵抗感がなかった	仕様レビューを導入したことで得られる目的、効果を事前に共有できていたことも要因の一つ	開発メンバへの説明もスムーズで、QAによる仕様レビュー導入後も、協力して設計工程での作りこみが出来ている

シフトレフト導入施策（QAによる仕様レビュープロセスの導入）

導入効果　★★

仕様不明確や、レビュー観点表に沿ったドメイン指摘を狙った指摘が検出出来ており、設計工程での不具合検出が増加している

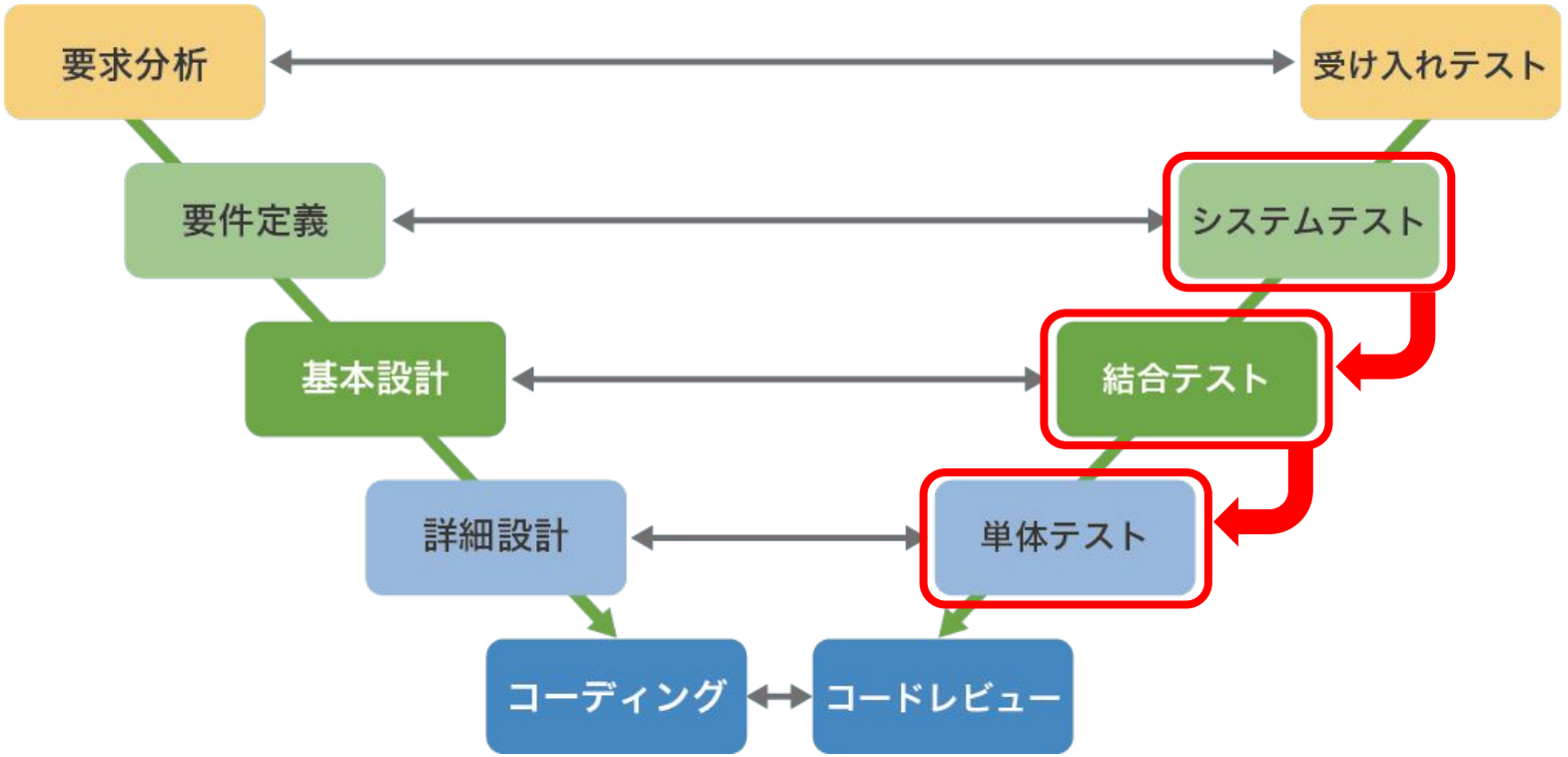
WBSに組み込んでいることもあり、確実に仕様レビューが実施される仕組みが出来ている

AGEST

GQM法による品質特性毎のメトリクス設定　※導入失敗

導入難度：★★★★★　導入効果：ー

アジャイル	ウォーターフォール



シフトレフト導入施策（GQM法による品質特性毎のメトリクス設定）

AGEST

背景・課題

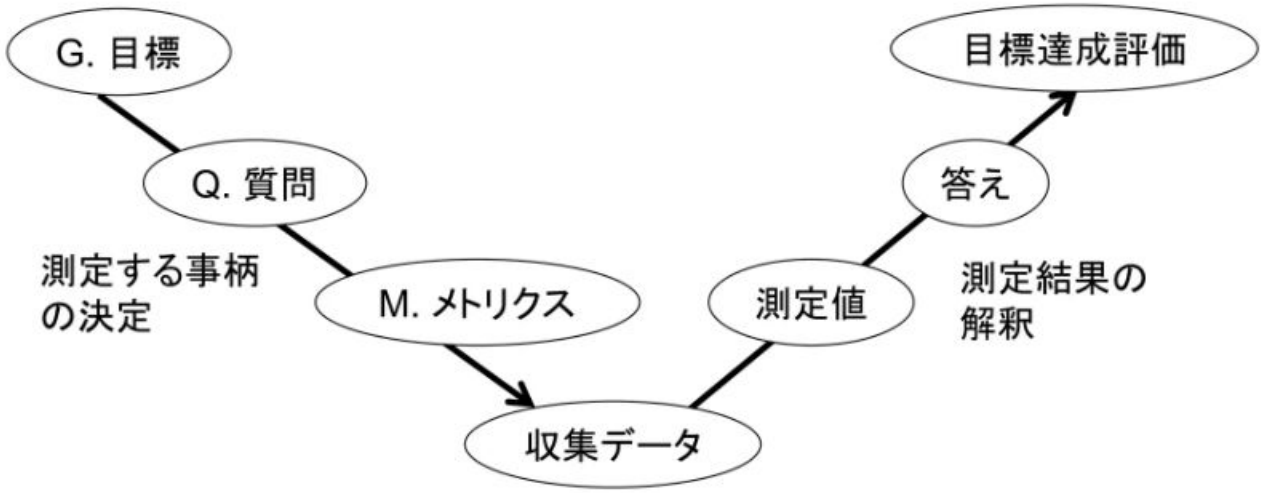
1. 大規模開発で**常時100名以上**のメンバがいる現場
2. テスト工程では**品質特性を利用した品質保証**を導入しており、
   品質特性を使用して品質向上に繋げる活動に理解がある
3. 設計工程では**レビューによるメトリクス**を取っている
4. システム刷新により**インフラとアプリ両方**で適用可能な
   品質保証の施策が求められていた
5. とにかく**ステークホルダ**が膨大

AGEST

導入施策

Goal-Question-Metric (GQM) パラダイム

- 明確に目標を据えて、目標に対して必要なメトリクスを対応付けるゴール指向（目標指向）な枠組み
- 目標（Goal）：測定上の目標
- 質問（Question）：目標の達成を評価するための質問
- メトリクス（Metric）：質問に回答するために必要な定量的データを得るための主観的・客観的メトリクス



R. van Solingen, E. Berghout, “The Goal/Question/Metric Method” McGraw-Hill Education, 1999; ISBN 0-07709-553-7.

2014年12月18日 SQiP研究会 演習コースI

メトリクスの基礎とGQM法によるゴール指向の測定

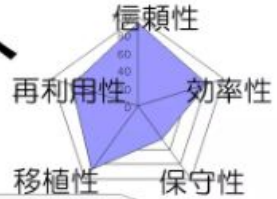
鷲崎　弘宜
早稲田大学グローバルソフトウェアエンジニアリング研究所
washizaki@waseda.jp
Twitter: @Hiro_Washi

https://www.slideshare.net/hironoriwashizaki/gqm20141218

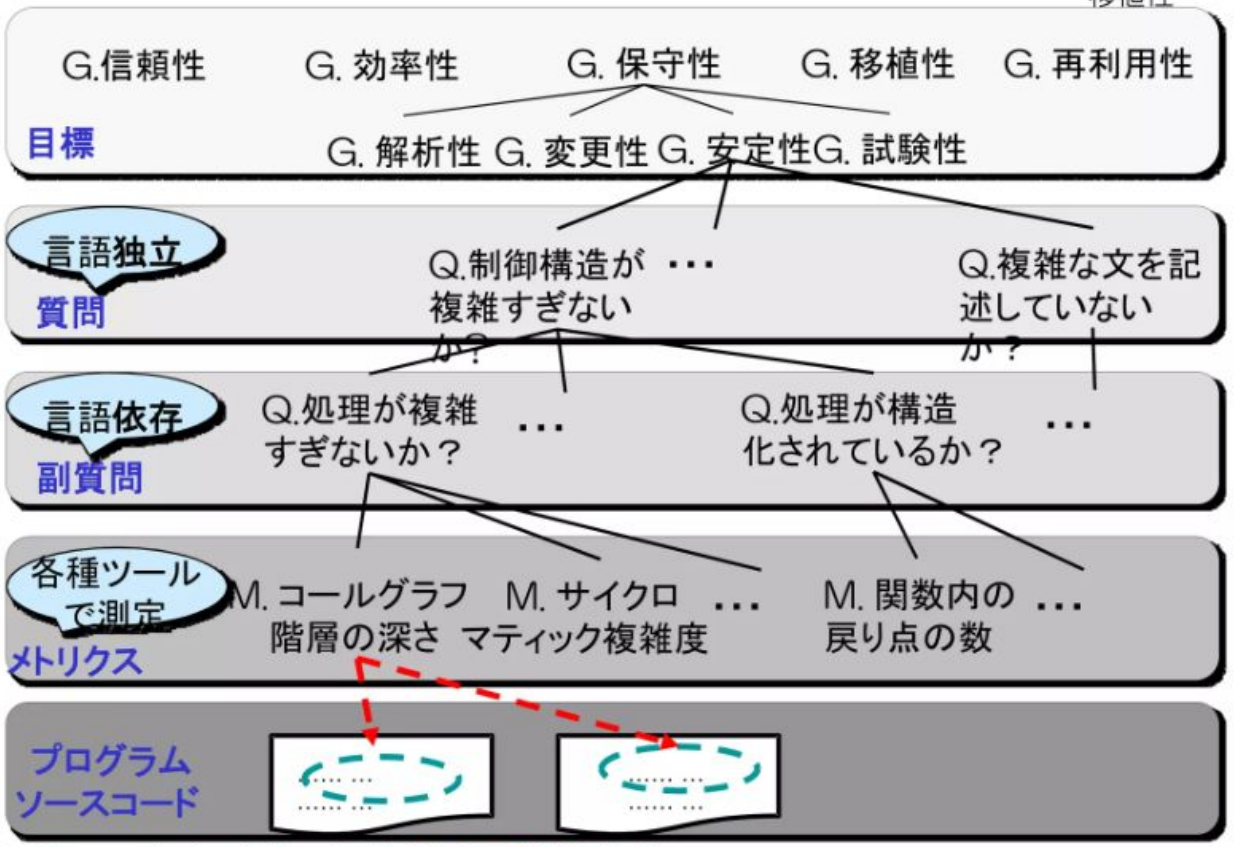
AGEST

導入施策

【応用】規模や複雑さから品質診断へ



- Goal-Question-Metric 法による段階的マッピング



[Adqua] http://www.ogis-ri.co.jp/product/b-08-000001A6.html

鷲崎弘宜,田邉浩之,小池利和,ソースコード解析による品質評価の仕組み,日経エレクトロニクス 2010/1/25

2014年12月18日 SQiP研究会 演習コースI

メトリクスの基礎とGQM法によるゴール指向の測定

鷲崎　弘宜
早稲田大学グローバルソフトウェアエンジニアリング研究所
washizaki@waseda.jp
Twitter: @Hiro_Washi

https://www.slideshare.net/hironoriwashizaki/gqm20141218

シフトレフト導入施策（GQM法による品質特性毎のメトリクス設定）

AGEST

導入難度　★★★★★

失敗した理由①	失敗した理由②	失敗した理由③
導入後のイメージの共有不足	**対象範囲が広かった**	**マネージメント層とリーダ層の齟齬**
汎用的なGQMを提案したが、具体的な部分まで検討出来ておらず、導入後のイメージを共有できなかった	システム刷新プロジェクトだったので対象範囲がインフラ、NW、共通API、機能、外部システム等と広かったため、 説明するステークホルダが多く時間がかかった	課題分析と施策検討をQAメンバのみで実施していたため、 マネージメント層やリーダ層と課題感や施策規模に齟齬があった

AGEST

導入効果　ー

マネージメント層への提案が通らなかったため導入失敗

マネージメント層の承認が通ってたとしても、有効な効果をあげられていたかは疑問が残る

シフトレフト導入施策

導入して効果が出た施策もあるが、効果が出なかった施策もある

それに施策を導入することにとても労力、気力、時間がかかり
シンドかった・・・

世間のQAの皆さんはこんな大変な思いをしてシフトレフト
をしているの？調べてみたら・・・

4. よりよいシフトレフトを求めて

よりよいシフトレフトを求めて

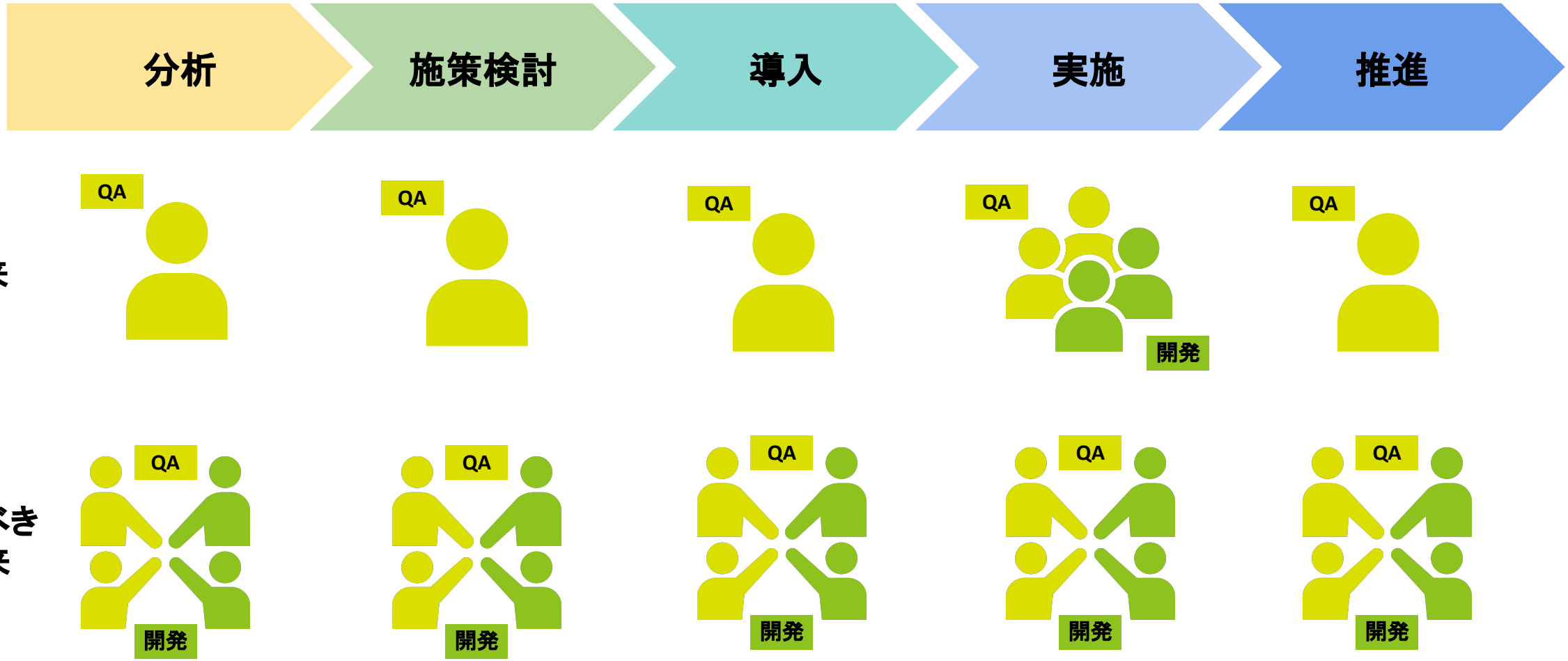
Jasst nanovol.9「シフトレフトって何をシフトするのなの？」より

- 要するに、
  - 開発の初期からちゃんと後工程(テストなど)について
  - 考えたり
    - QAやテストエンジニアと開発者が一緒に考えたり
    - 開発者自身が考えるようになったり
  - 実際に作業したりして
    - QAが要求や設計のレビューに加わったり
    - テスト設計をより上流で行って(コード無しで)要求や設計のバグを見つけたり
    - 上流のモデルベース/モデル駆動化を進めて(コード無しで)要求や設計のバグを見つけたり
    - トレーニングしたりTotTしたり
  - 後回しにする嫌なことを
    - 工数やコストが大幅に増えたり
    - 技術的負債ができたり
    - エンジニアが疲弊してモチベーションが下がったり
  - 防ぐことなの！

https://www.slideshare.net/YasuharuNishi/what-should-you-shift-left

よりよいシフトレフトを求めて

従来も「**目的**」は立てていた、「**方法**」も検討していた、ただ「**共に考える**」発想がなかった！

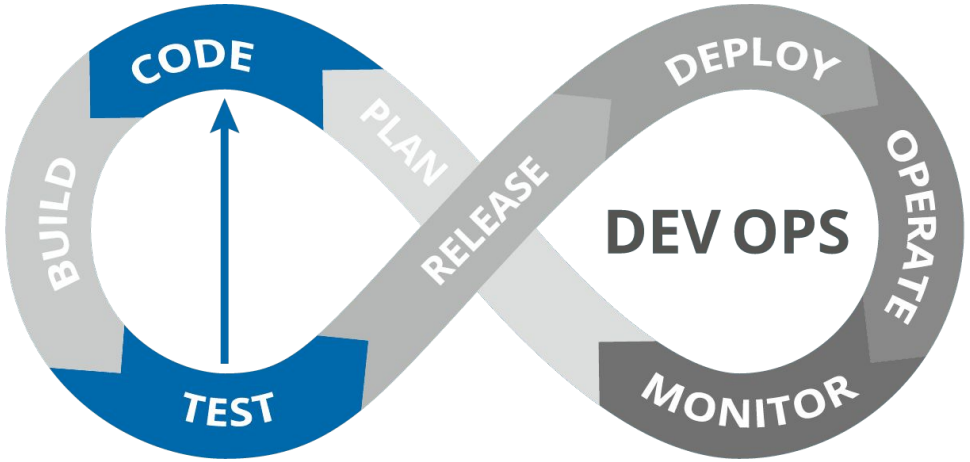


AGEST

開発テストの最適化

アジャイル	ウォーターフォール

導入難度：★★　導入効果：ー



AGEST

導入現場

SMS 株式会社エス・エム・エス

カイポケ 介護ソフト「カイポケ」

AGEST

背景・課題

1 小規模な現場

アジャイル開発、チームは開発QA合わせて15人程度

2 現場の品質意識が高い

開発者の品質に対する意識が高く、また開発とQAのコミュニケーションも活発

3 開発とQAのテスト境界線が曖昧

QAは開発者が実施しているテスト内容や範囲を正確に理解しておらず

開発者が実施するテストとQAが実施するテストの境界線が曖昧

AGEST

単体テストの理解

分析 → 施策検討 → 導入 → 実施 → 推進

テスト目的

JSTQBでは

2.2.1 コンポーネントテスト

コンポーネントテストの目的

コンポーネントテスト（ユニットテストまたはモジュールテストとも呼ばれる）は、個別にテスト可なコンポーネントに焦点をあてる。コンポーネントテストの目的は以下の通りである。

- リスクの軽減
- コンポーネントの機能的/非機能的振る舞いが設計および仕様通りであることの検証
- コンポーネント品質に対する信頼の積み上げ
- コンポーネントに存在する欠陥の検出
- 欠陥がより高いテストレベルまで見逃されることの防止

要約すると**テスト目的とは**

- プログラム（コンポーネント）に依存する欠陥[1]を検出する
- 単体テストフェーズで検出可能な欠陥を見逃さない

テスト対象

JSTQBでは

テスト対象

コンポーネントテストの典型的なテスト対象の例：

- コンポーネント、ユニット、またはモジュール
- コードとデータ構造
- クラス
- データベースモジュール

- コンポーネント、ユニット、モジュールはだいたい同じ、「部品単位」と理解した
  - 参考はコチラ

ただわかりづらい・・・

コチラがイメージしやすい　※赤線箇所が単体テスト対象

要約すると**テスト対象は**

- 画面を構成する最小単位（コンポーネント等の部品単体、クラス、DBモジュール等）
- 最小単位を組み合わせた画面単体

テスト種類

書籍「JSTQB　Foundation第３版」より　※↑のpdfには記載がなかった

テストの種類：
・機能テスト
・非機能テスト（性能など）
・構造テスト（デシジョンカバレッジなど）
・リソース動作テスト（メモリーリークの検出など）
・ロバストネステスト

- ロバストネストテスト：堅牢性を検証するテスト＝コンピュータシステムで実行中のエラーや、誤った入力に対処できる能力のこと　参考はコチラ

要約すると**テスト種類は**

テスト種類	テスト手法	備考
機能テスト	動的/静的テスト	期待通りの動作をすること(レビューを含む)
非機能テスト	動的/静的テスト	性能、セキュリティ、ユーザビリティ等。もう少し理解した方がよいかも
構造テスト	静的テスト	コードカバレッジ（C0,C1等）の計測
リソース動作テスト	静的テスト	静的解析ツールによるメモリーリーク等の検出
ロバストネステスト	動的テスト	異常値テスト(異常となる値を入力してエラーとなることのテスト)

AGEST

開発者テストの理解

分析 → 施策検討 → 導入 → 実施 → 推進

AGEST

開発者と認識合わせ

分析 | 施策検討 | 導入 | 実施 | 推進

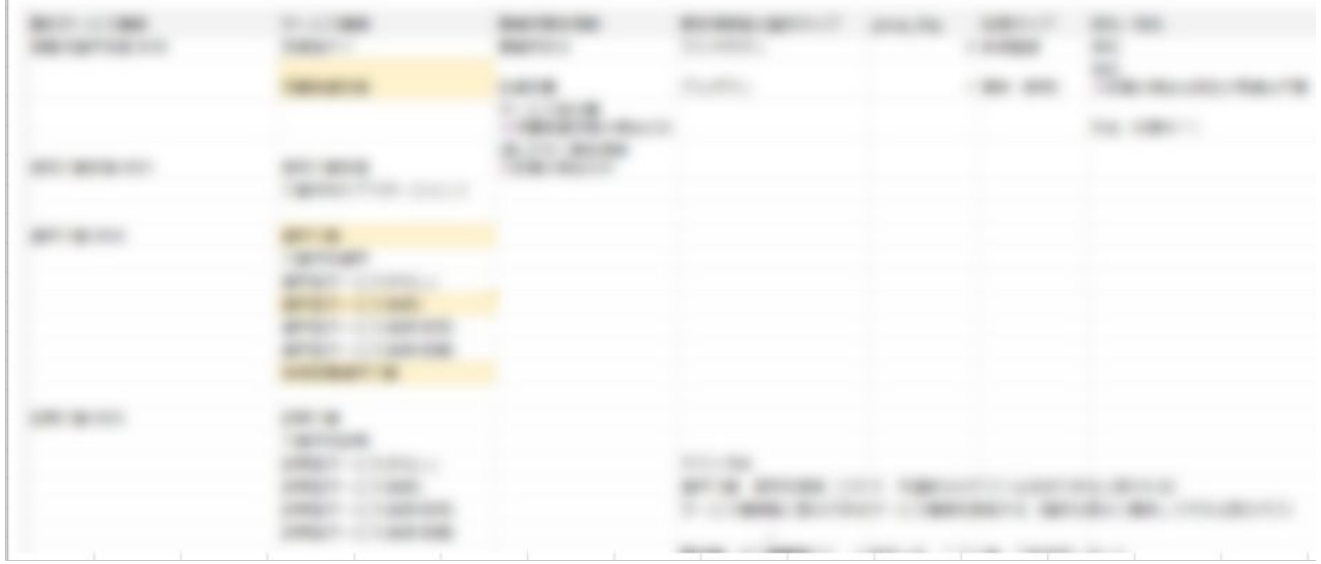
開発者テスト内容を理解

テスト目的

一般的なテスト目的	開発者テスト
プログラム（コンポーネント）に依存する欠陥を検出する	○
単体テストフェーズで検出可能な欠陥を見逃さない	○

- 🐼 プログラム（コンポーネント）に依存する欠陥を検出する
  - コード（ロジック）に対するホワイトボックステストを実施している

- 🐼 単体テストフェーズで検出可能な欠陥を見逃さない
  - 因子、水準を洗い出している
  - サービス種類毎に差分があるサービス種類を確認している

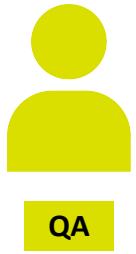


よりよいシフトレフトを求めて（開発テストの最適化）

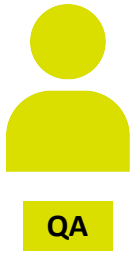
AGEST

分析

単体テストの理解



開発者テストの理解



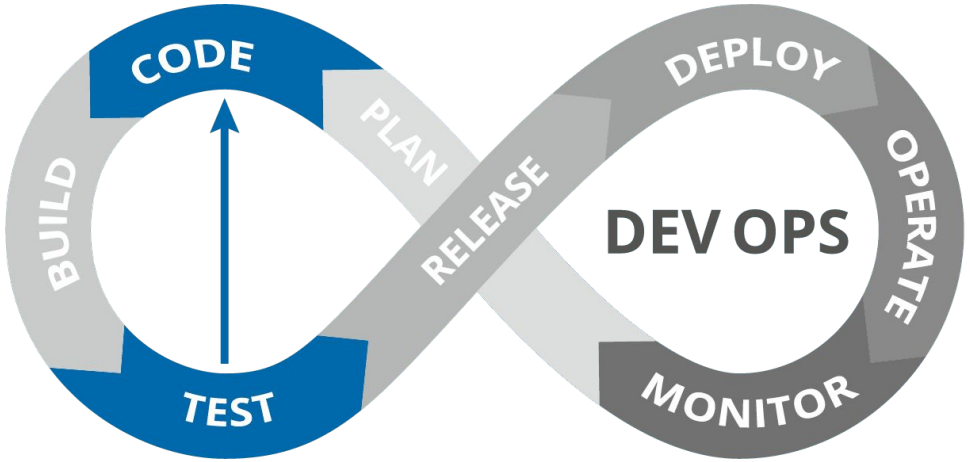
開発者と認識合わせ

開発テストの最適化改め

モブテスト（MobTesting）の導入

アジャイル	ウォーターフォール

導入難度：★　導入効果：-



AGEST

導入施策

- 単体テスト時に開発とQAで一つの画面を見ながら探索的にテストする
- まずはプレとしてあまりルール（セッションやロール決め）を設定せずに実施する
- QAは仕様理解と改修箇所の過去不具合を調査した上で参加する

AGEST

導入難度 ★

分析	施策検討	導入	実施	推進
				
単体テストの理解 開発者テストの理解 開発者と認識合わせ	予定施策の効果予測 開発側からの新提案	モブテストの導入	プレ実施	プレ実施の フィードバック

開発テストの最適化 → モブテスト（MobTestting）の導入

5. 品質支援アプローチについて

If you want to go fast, go alone.
If you want to go far, go together.

早く行きたければ、ひとりで行け。
遠くまで行きたければ、みんなで行け。

Mission
ミッション

SAVE the DIGITAL WORLD

Vision
ビジョン

先端品質テクノロジーで、

すべてのDXに豊かな価値と体験を

Value
バリュー

Agility - 俊敏な行動
Professionalism - プロであること
Simplicity - シンプルさ
Welcome change - 変化を味方につける
As a team - チームとして働く