

# SQuBOK v3にみる不確実なDX時代の確実な 品質技術に向けて

- AI & 機械学習、アジャイル&DevOps&オー  
プンソース開発を中心に -

鷺崎 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  
国立情報学研究所 客員教授

システム情報 取締役(監査等委員)

エクスモーション 社外取締役



Twitter: @Hiro\_Washi [washizaki@waseda.jp](mailto:washizaki@waseda.jp)

<http://www.washi.cs.waseda.ac.jp/>

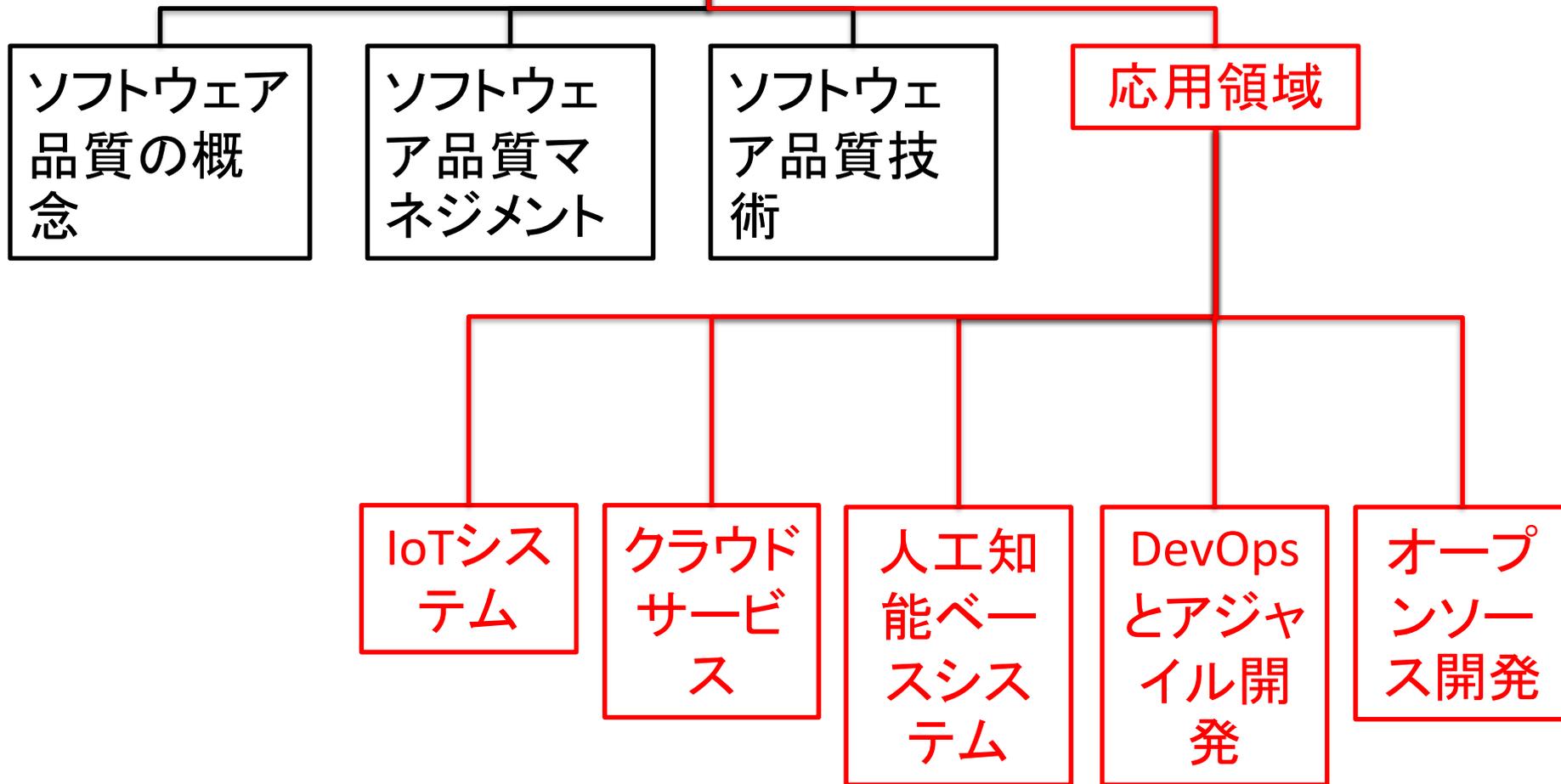


# デジタルトランスフォーメーション(DX)時代

- DX: デジタル化によりビジネスモデル変革
  - デジタル技術: IoT、クラウド、AI、ソーシャルほか
  - 変革レベル: プロセス→組織→ビジネス領域→社会
- 問題
  - 低リスクのオプションを選んで非常に少ない投資
  - デジタライゼーションが技術に限った話と誤解
  - 顧客が求めた時に直ちに対応できていない
  - 規約や標準へ厳格に従うばかりに変化やイノベーションを妨げ
- 解決
  - アジリティ、高速・適応的・仮説検証による不確実性の扱い
  - データ収集、エビデンスベース
  - アドホックから戦略的、複数を束ねるエコシステムへ
  - 階層化組織からアジャイル組織、ホラクラシーへ

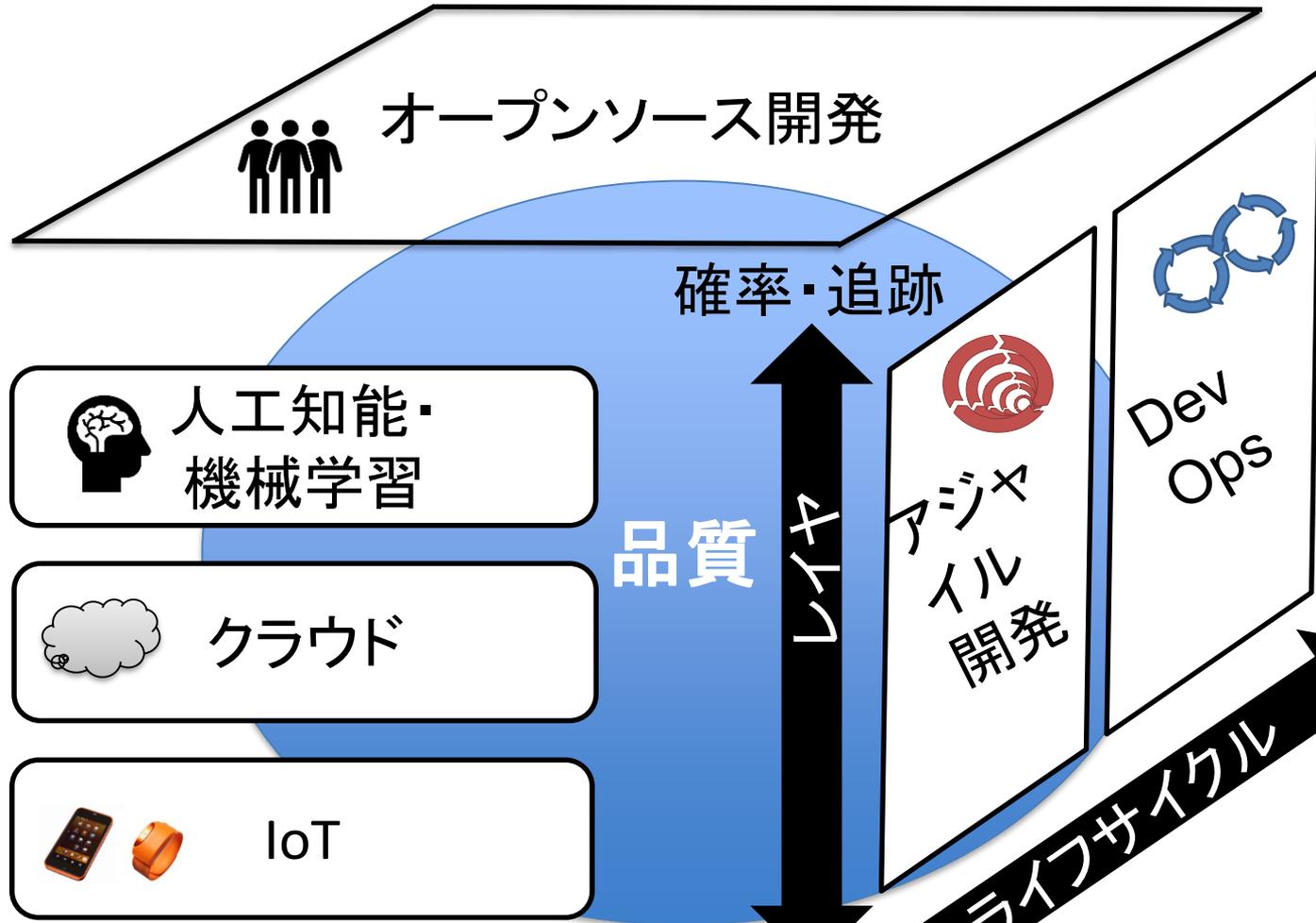


# SQuBOKガイド 2020 (V3)



応用領域担当: 鷲崎 弘宜(取りまとめ)、石川 冬樹、鄭 顕志、松崎 和賢、竹之内 隆夫、長久 勝、伊原 彰紀、森田 純恵ほか

# SQuBOK応用領域と品質



# 人工知能ベースシステムの品質

## 人工知能システムにおける品質の概念

- 訓練・テストデータ品質
- 性能指標
- 頑健性
- 説明可能性
- 機械学習システムの品質

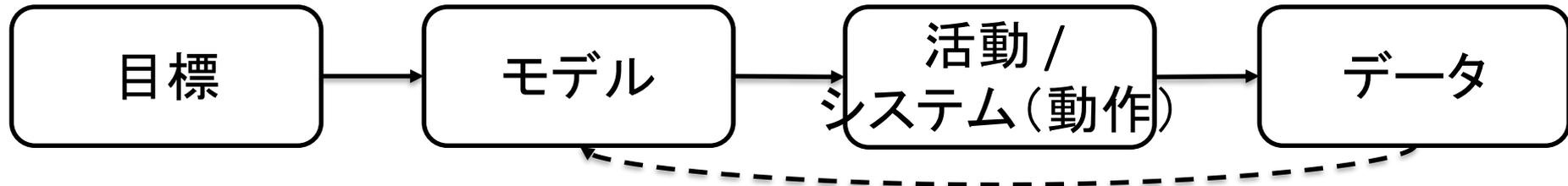
## 人工知能システムの品質保証マネジメント

## 人工知能システムの品質保証技術

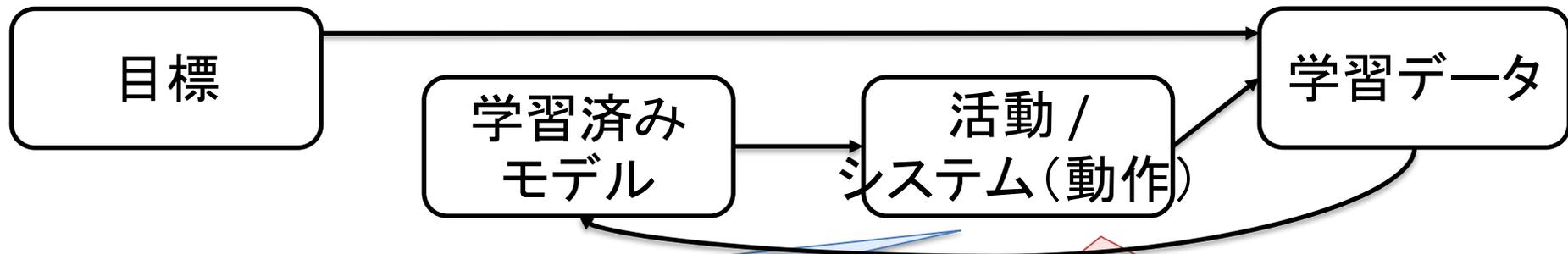
- 疑似オラクル
- メタモルフィックテストイング
- 頑健性検査
- ニューロンカバレッジ
- 説明生成

# 機械学習によるパラダイム転換

従来のエンジニアリング: **演繹的** (モデルが最初に与えられる)



機械学習ベース: **帰納的** (モデル・アルゴリズムがデータで決まる)



プロセス > プロダクト 

機械学習ベース・データ駆動エンジニアリング

実行 

機械学習システム

# 人工知能システムの品質の概念

- データの品質
- モデルの性能
  - 正解率、混同行列
  - ROC、AUC
  - 未学習、過学習
- モデルの頑健性
- モデルの説明可能性
- システムの品質
  - KPI
  - 仮説検定

		予測	
		陽性 Positive	陰性 Negative
正解	陽性	真陽性 True positive	偽陰性 False negative
	陰性	偽陽性 False positive	真陰性 True negative

未学習

過学習

<https://www.slideshare.net/dsuket/ss-57488780>

# 人工知能システムの品質保証マネジメント

- 試験的・探索的と顧客協力
- 監視と性能劣化検出
- オンライン学習と検証
- 成果物・版管理
  - データ、プログラム、ハイパーパラメータ

バッチ学習

オンライン学習

# 人工知能システムの品質保証技術

- 疑似オラクル
  - 別の実装や古いバージョンとの比較
  - サーチベースドテストティング
- メタモルフィックテストティング
  - 入力の変化により出力を予想可能な関係によるテスト
  - 例: ノイズを追加しても判定が変わってはいならない
- 頑健性検査
- ニューロンカバレッジ
- 説明生成

# 説明生成: 判断根拠を説明できるか

- 背景: 日本 AI利活用原則、EU GDPR、US DARPA XAI

分類	概要	例
大域的説明	近似モデル	決定木 (Born Again Trees) ルール (defragTrees)
局所的説明	根拠	特徴量 (LIME) データ (influence)
深層学習モデルの説明	特に画像認識モデルの根拠	注目領域のハイライト (SmoothGrad) 説明文生成 (Visual Explanations)
説明可能なモデル設計	説明可能モデル	シンプルなモデルの採用 ルールセット (Interpretable decision sets)

原 聡, 私のブックマーク: 機械学習における解釈性, 人工知能 33(5), 2018

原 聡, 私のブックマーク: 説明可能AI, 人工知能 34(4), 2019

AI・機械学習 グループ, アドバンス・トップエスイー 最先端ソフトウェアゼミ成果発表, 2019

<https://www.topse.jp/images/%E6%A9%9F%E6%A2%B0%E5%AD%A6%E7%BF%92%E3%82%BC%E3%83%9F.pdf>

# 大域的の例: defragTrees

## 説明可能モデルの例: Interpretable decision sets

Making Tree Ensembles Interpretable: A Bayesian Model Selection Approach, AISTATS 2018

## 局所的の例: LIME

“Why Should I Trust You?” Explaining the Predictions of Any Classifier, KDD 2016

## 深層学習モデルの説明例: SmoothGrad

Interpretable Decision Sets: A Joint Framework for Description and Prediction, KDD 2016

# DevOpsとアジャイル開発における品質

## DevOpsとアジャイル開発における品質の概念

- DevOpsにおける品質特性と品質維持技法
- アジャイル開発の品質指標

## DevOpsとアジャイル開発の品質保証マネジメント

- 伝統的な品質保証(QA)からアジャイル品質(AQ)への転換
- アジャイルスキル体系

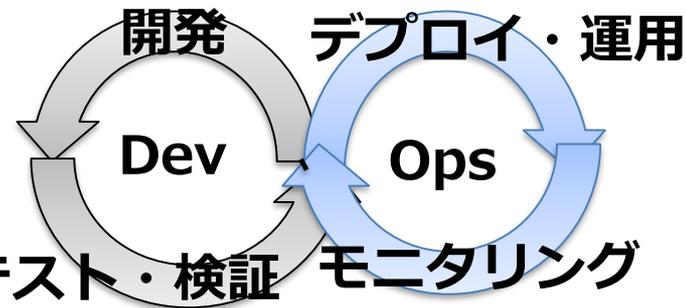
## DevOpsとアジャイル開発の品質保証技術

- 品質ダッシュボード
- ツール類

### テスト技術

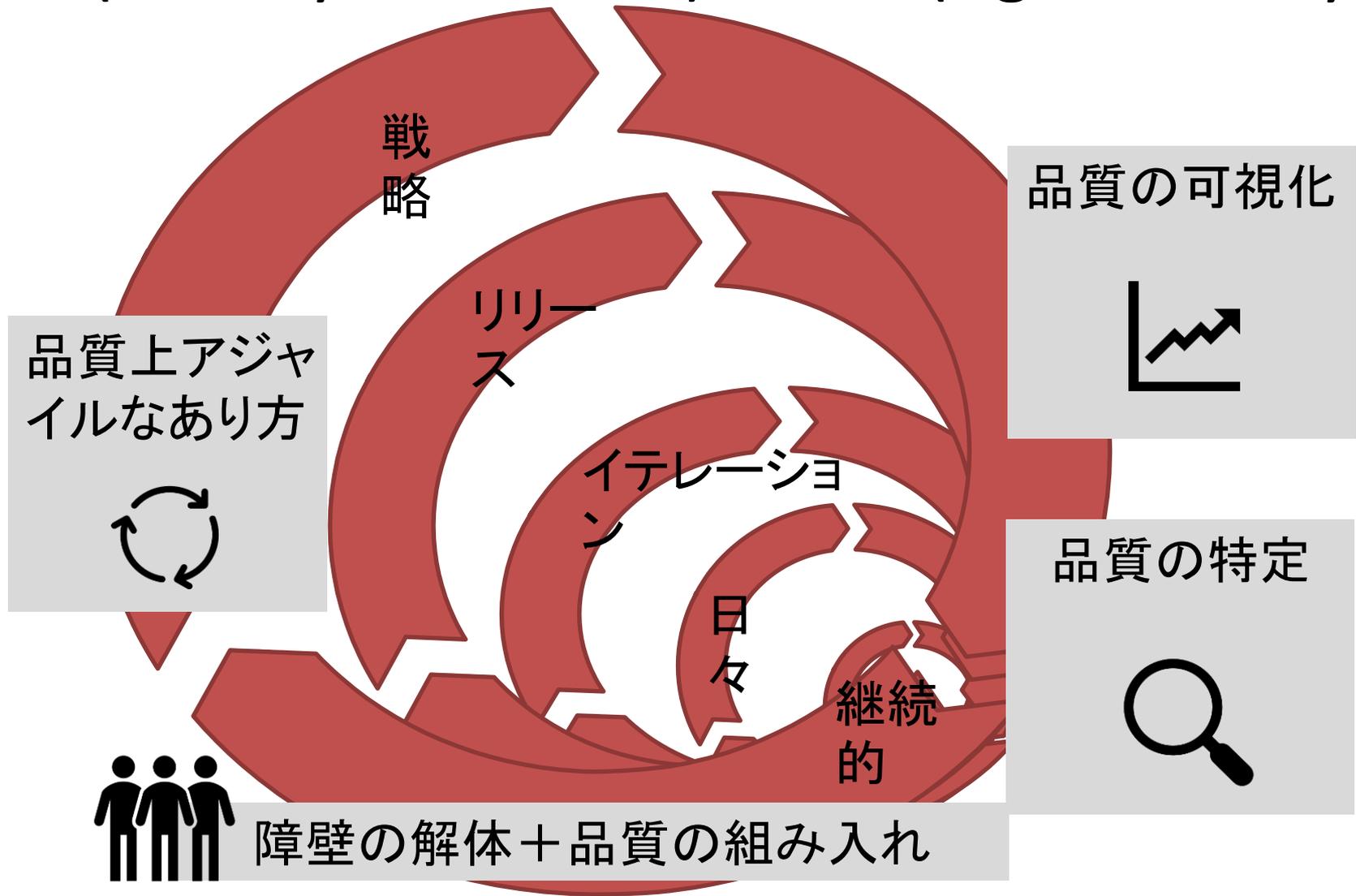
- アジャイルテスト
- 継続的テスト
- シフトレフトテスト
- シフトライトテスト
- カナリアテスト

# DevOpsにおける品質 特性と品質維持技法



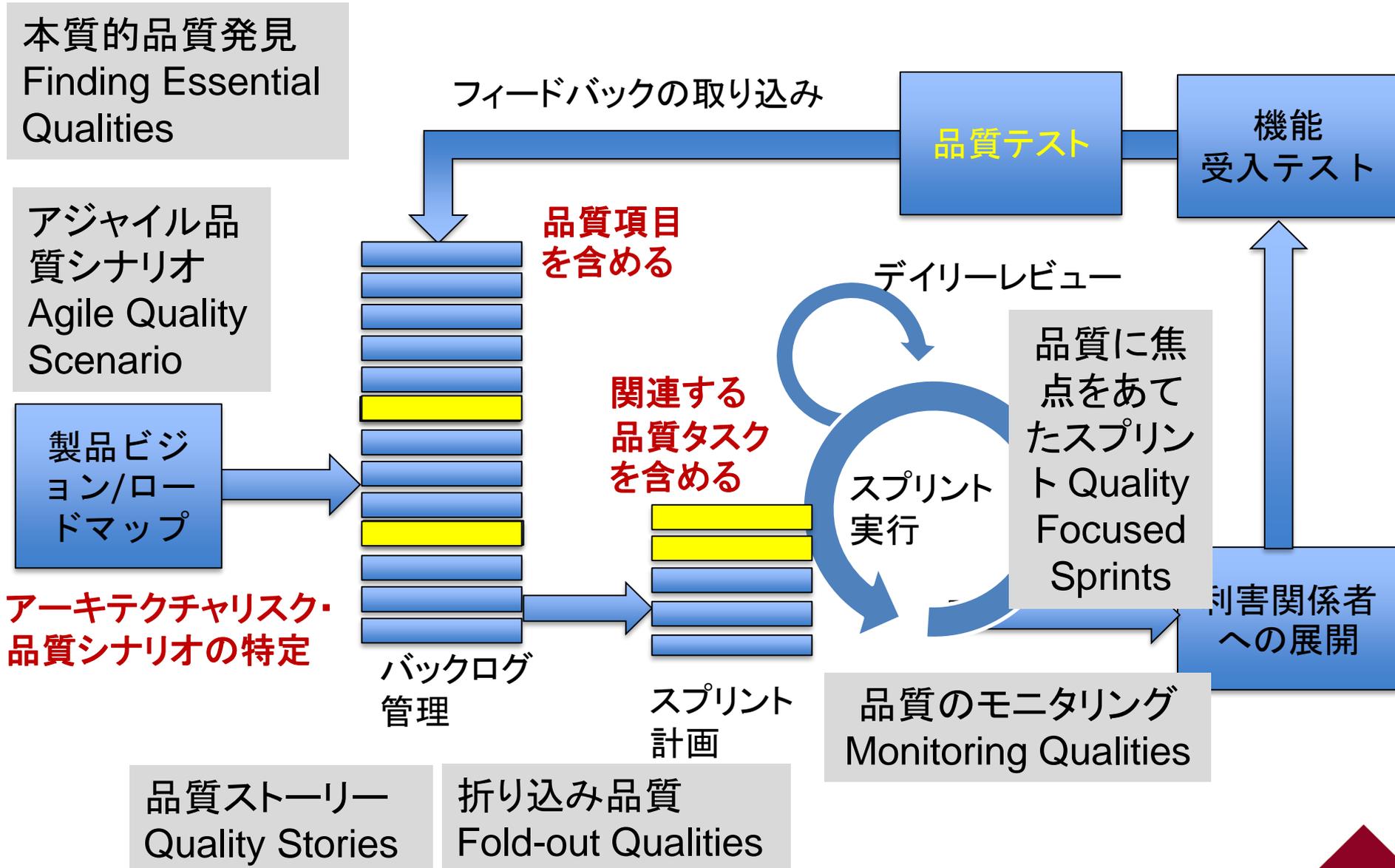
特性	品質を高めるためのテクニック
反復可能性	アクティビティのトレースを残す。すべてのもののバージョンを管理する等。
処理性能(パフォーマンス)	プロセスのボトルネックを明らかにするための計測を行う。使っていない環境を解体する等。
信頼性	さまざまなサービスのエラー率を明らかにする。エラー率の高いサービスをミラーリングする等。
回復可能性	スクリプトに例外処理を組み込む。モニタリングサービスに情報を提供する。適切な診断を生成してデバッグ作業が早く終わるようにする。
相互運用性	安定したインタフェースを柔軟なスクリプト機能をもつツールを選ぶ。パイプラインのさまざまなフェーズデータモデルの一貫性を保つ。
テスト可能性	専用ツールのために単体、インテグレーションテストを行う。テストケースとモニタリング規則を調和させる。
変更可能性	ツールの予想される変更に基づいてスクリプトをモジュール化する。アクティビティを小さなモジュールにカプセル化し、モジュール同士の結合は疎になるようにする。

# QA (Quality Assurance) to AQ (Agile Quality)



- J. Yoder, R. Wirfs-Brock, A. Aguilar, "QA to AQ: Patterns about transitioning from Quality Assurance to Agile Quality," AsianPLoP 2014
- J. Yoder and R. Wirfs-Brock, "QA to AQ Part Two: Shifting from Quality Assurance to Agile Quality," PLoP 2014
- J. Yoder, R. Wirfs-Brock, H. Washizaki, "QA to AQ Part Three – Tearing Down the Walls," SugarLoafPLoP 2014
- J. Yoder, R. Wirfs-Brock, H. Washizaki, "QA to AQ Part Four - Prioritizing Qualities and Making them Visible," PLoP 2015
- J. Yoder, R. Wirfs-Brock, H. Washizaki, "QA to AQ – Part Six – Being Agile at Quality," PLoP 2016

# プロセスへの品質の組み入れ例

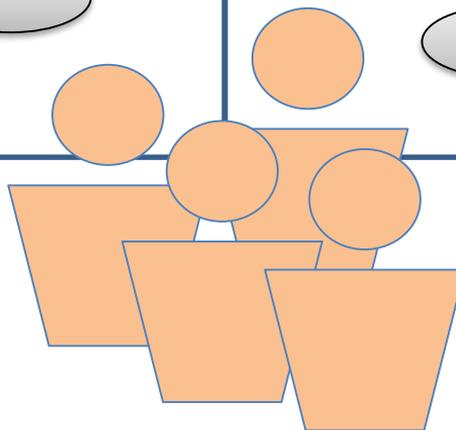




# 可視化: バックログ上の品質検討

## Qualify the Backlog

Product	To Do	Doing	Done
	 セキュアに… 	 	
			 技術的負債を…



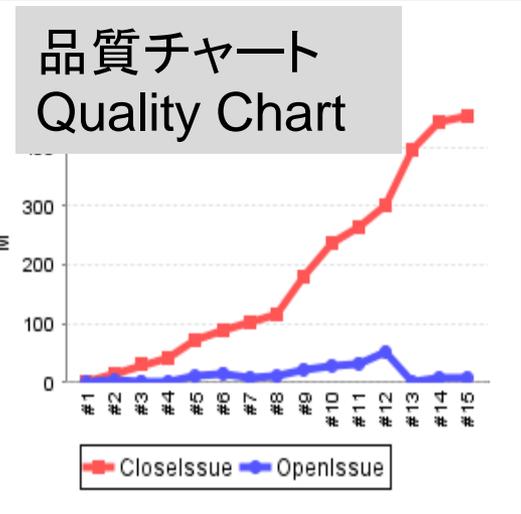
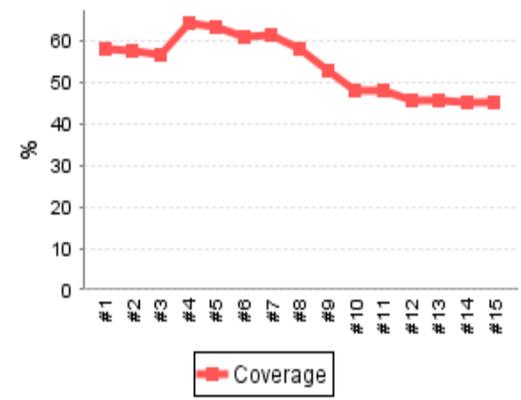
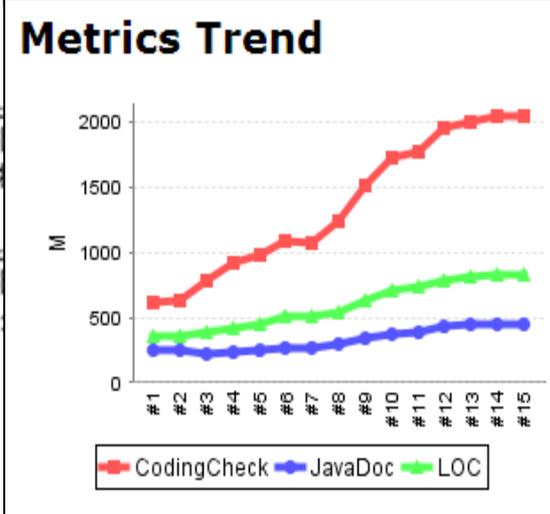
品質シナリオのバックログ項目を追加  
(システムの品質に関連するアーキテクチャの作業も含めて)



# 品質ダッシュボードと品質ラジエータ

**G:** 一定の機能性を確保できている

SubGoal	Question	SubQuestion	Metric
テストが十分できている	テストに人ではらつきはないか		テストカバレッジ(人)
	重要なモジュールに関して、テストが十分できているか	高ファンアウトで十分できているか	テストカバレッジ ファンアウト
	重要度が中・低程度のモジュールに関してテストが十分できているか	中・低ファンアウトで十分できているか	テストカバレッジ ファンアウト
変更・追加が想定外の影響を与えにくくなる	変更・追加が想定外の影響を与えていないか	Bug Fixによる影響か	修正後欠陥数 影響分析結果



修正済み欠陥数
潜在欠陥数

# テスト技術の拡充

- アジャイルテスト
  - 顧客・プログラマ・テスト対話、イテレーション単位
- 継続的テスト
  - コード変更の都度、テスト自動実行
- シフトレフトテスト
  - 早い段階(仕様、設計など)でのテスト
- シフトライトテスト
  - 本番稼働後のテスト
  - 例: カオステスト、カナリアテスト
- カナリアテスト
  - 本番環境やユーザの一部でテストしてから展開

# オープンソース開発における品質

## オープンソース開発における品質の概念

- OSSの定義
- OSS開発の特徴
- OSS開発における品質の概念

## オープンソース開発利活用の品質マネジメント

- プロジェクトの組織管理
- コミュニケーション管理
- 再利用OSS管理

## オープンソース開発の品質保証技術

- プルリクエスト駆動開発
- OSS健全性評価メトリクス

# CHAOSS (Community Health Analytics Open Source Software)

- 多様性の組み入れ Diversity-Inclusion
  - OSS組織の多様性、組織に参加する個人の多様性、組織体制の観点で公平な協調作業を実践しているか
  - 例: 参加者性別、年齢、スキル、貢献者人数、昇格
- 成長-成熟-衰退 Growth-Maturity-Decline
  - 不具合解決、プログラム実装、組織の成熟度、また、組織の発展過程、衰退傾向
  - 例: 不具合解決数、解決時間、変更数、参加者数
- リスク Risk
  - 人的要因、ライセンス、脆弱性の観点で組織が負うリスク
  - 例: ライセンスの内容、脆弱性数
- 価値 Value
  - 組織に参加する価値
  - 例: 市場価値、ダウンロード数、製品への導入数

# SQuBOK 2020 乞うご期待 & ご意見歓迎！

- 技術・プロセスのポータル
- 用語整理 & 標準化
- 文献ガイド

