

AI-DRIVEN QUALITY ASSURANCE

# AI駆動品質保証

開発プロセスにAIを組み込んだ  
次世代の品質保証アプローチ

---

Qurated lab

2026

# AI開発の現状と課題

多くの企業がAI活用で直面している問題



## 品質の低下

AIが生成したコードのレビュー不足  
によるバグの混入



## 一貫性の欠如

開発者ごとにAIの使い方がバラバラ  
品質基準が統一されない



## 技術的負債の増加

「動くコード」の量産により  
保守困難なコードベースが膨張

# 73%

のエンジニアが

AI生成コードの品質に  
不安を感じている

※ 業界調査レポートより

# Qurated lab のAI品質保証プラットフォーム

AIを「品質保証の専門チーム」として体系化する独自アプローチ

01

## ルールベースの制約

7つの行動規範で  
AIの動作を厳密に定義

02

## 自動品質ゲート

JSTQB準拠のテスト設計と  
自動検証を全段階に組込

03

## 継続的な自己学習

成功/失敗パターンを蓄積し  
AIの判断精度が向上

AIを「書かせる」から「品質を担保させる」へ

全体像

# プラットフォーム構成

品質保証基盤を構成する5つの柱

30+

専門AI  
エージェント

8

自動品質  
評価基準

14

ライフサイクル  
フック

32

再利用可能  
ワークフロー

20

JIRA自動化  
ルール

コード生成 > 自動検証 > レビュー > 学習 > JIRA同期 > 品質向上

# 10の絶対ルール

AIの行動を厳密に制約する計算的に検証可能なルール体系

1 計画書がある作業は計画書に従う

2 設計判断はユーザーに確認してから決定

3 依頼外の「改善」をしない

4 計画書の変更はユーザー承認後に実施

5 タスク完了ごとに進捗を報告する

6 実装前に計画を提示して承認を得る

7 git push 前は必ずユーザー承認

8 ドキュメントの規約は全て遵守

9 イテレーション全ステップを実行

10 タスク完了時にステータスを更新

全ルールは違反検出時の自動対応フローを備え、品質の逸脱を未然に防止

# タスク実行イテレーション

全タスクで実行される6ステップの品質保証プロセス



## 全ステップ自動実行

各ステップの完了はフックにより自動検証。品質基準を満たさない場合は次に進めません。  
ISO/IEC 25010 の8品質特性を各ステップに対応付け、網羅的に品質を確保します。

# JSTQB 準拠のテスト設計

国際ソフトウェアテスト資格認定に基づく体系的テスト設計

## 同値分割法

入力を有効/無効クラスに分類し代表値でテスト

## 境界値分析

境界値 (0, 1, max, max+1) を重点的に検証

## デシジョンテーブル

複数条件の全組み合わせを網羅的にテスト

## 状態遷移テスト

状態間の遷移パスと禁止遷移を検証

## カバレッジ基準

一般コード	80%+
金融・決済ロジック	95%+
認証・セキュリティ	95%+

## ISO/IEC 25010 品質特性

- |          |         |
|----------|---------|
| - 機能適合性  | - 性能効率性 |
| - 使用性    | - 信頼性   |
| - セキュリティ | - 保守性   |
| - 互換性    | - 移植性   |

# 30+の専門AIエージェント

タスクの性質に応じて最適な専門家が自動的に介入

## コードレビューアー

品質・保守性チェック

即時

## アーキテクト

設計妥当性の検証

即時

## セキュリティレビューアー

脆弱性の検出

即時

## TDDガイド

テスト駆動開発の支援

即時

## テストプランナー

テスト設計・戦略策定

即時

## ビルドエラー解決

ビルド障害の自動修復

即時

## クロスレビューアー

別AIによる二重チェック

非同期

## DB設計レビューアー

データベース設計の検証

即時

## E2Eランナー

統合テスト実行

非同期

+ さらに多数の専門エージェント(言語固有スペシャリスト、ドキュメント更新等)

# マルチAIクロスレビュー

複数のAIモデルによる多角的なコード品質検証

## 第1段階: 詳細コードレビュー

- セキュリティ脆弱性の検出
- パフォーマンス問題の特定
- アーキテクチャ違反のチェック
- コーディング規約の遵守確認
- CRITICAL/HIGH/MEDIUM分類



## 第2段階: クロスチェック

- 別AIモデルによる独立検証
- 第1段階の見落としを検出
- 代替アプローチの提案
- 品質スコアの二重確認
- 最終品質判定

単一AIモデルのバイアスを排除し、多角的な視点で品質を担保

# ライフサイクルフック

開発の各段階に自動的に介入する品質ゲート

● セッション開始	前回状態の復元、学習パターンの読み込み	開始時
● ツール実行前	セキュリティチェック、危険操作ブロック、インジェクション検出	毎回
● ツール実行後	アーキテクチャ違反検出、テストヘッダー検証	毎回
● 応答完了時	パターン抽出、学習サイクル実行、スナップショット保存	毎回
● コンパクト前	重要状態のスナップショット保存(タスク・エラー・決定事項)	自動

## フェイルセキユア設計

セキュリティ関連フックはエラー時にブロック(Fail-Secure)。情報提供フックはFail-Openで開発を妨げない。

# 自己学習システム

セッションを重ねるごとにAIの判断精度が向上する仕組み

256+

蓄積された  
学習パターン

## 成功パターン

信頼度0.7以上で記録  
Phase完了、バグ修正等

180+件

## 失敗パターン

回避すべきアンチパターン  
TDDスキップ、レビュー省略等

50+件

## 観測データ

メトリクス時系列  
テスト結果、ビルド状態等

1.4MB

信頼度が特定の閾値を上回ったパターンは「本能」に昇格し、自動適用される判断基準となる

# Git × JIRA 完全自動連携

Git操作だけでJIRAが完全同期。手動ステータス更新ゼロ。

R1

ブランチ作成 + push

> To Do > In Progress

R2

PR作成(非ドラフト)

> In Progress > In Review

R3

PRマージ

> In Review > Done

R4

全子タスク完了

> 親Epic > 自動Done

## 品質ゲート連動

PR作成前に品質ゲート(静的解析件・全テストパス・カバレッジ基準 CRITICALレビュー0件)を通過必須。  
JIRAチケットにはAIが品質エビデンスを自動コメント。プロダクトの数百のチケットを体系的に管理。

# Figma 連携・ドキュメント駆動開発

デザインからコードまで一貫した品質管理体制

## Figmaデザイン連携

- Figmaデザインと連携し、仕様書のベース資料として活用
- デザインシステムとコンポーネント設計の一致を検証
- UI実装時にFigmaとの差分をレビューで確認
- アクセシビリティ要件をデザイン段階で定義

## ドキュメント駆動開発

- |                |                            |
|----------------|----------------------------|
| <b>実装計画書</b>   | 全機能に事前承認制の計画書を作成           |
| <b>技術別ガイド</b>  | React/Go/Flutterなど各技術の詳細規約 |
| <b>バグ監査書</b>   | 体系的な監査レポート(69件管理)          |
| <b>品質メトリクス</b> | ISO 25010準拠の測定基準を文書化       |

# マルチスタック統一品質管理

同一の品質基準をモバイル・Web・バックエンド・インフラに統一適用

## モバイル

Flutter / Dart

iOS / Android

WebView連携

### 品質ルール

- Widget テスト
- Riverpod状態管理規約
- Secure Storage

## フロントエンド

React / TypeScript

Next.js (App Router)

Tailwind CSS

### 品質ルール

- Zodバリデーション
- XSS/CSP対策
- コンポーネント設計

## バックエンド

Go (Gin)

AWS DynamoDB

JWT認証

### 品質ルール

- 入力検証必須
- インジェクション対策
- エラーハンドリング

## インフラ

AWS (EC2/ECS)

GitHub Actions

IaC

### 品質ルール

- 暗号化必須
- 最小権限原則
- セキュリティスキャン

## SUMMARY

# 日本企業で最も先進的な AI 駆動開発を実践

- 30+の専門AIエージェントによる自動品質保証
- JSTQB / ISO 25010 準拠の体系的テスト設計
- マルチAIクロスレビューで人的バイアスを排除
- Git × JIRA完全自動連携で手動管理ゼロ
- Figma連携・ドキュメント駆動で設計品質を担保
- Flutter / React / Go のマルチスタック統一品質基準

---

**Qurated lab**

quratedlab.com