

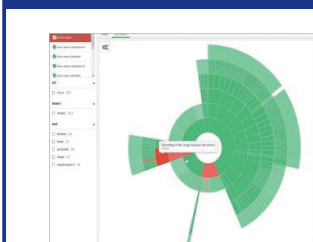
### 業界屈指のクロスプラットフォームテスト自動化対応

#### GUIテストの自動化と効率化

- ✓ テスト結果の一元管理とデータの可視化: 品質保証プロセス全般の見えるさを解決
- ✓ 自動化によりGUIテストの工数削減: 生産性の向上
- ✓ テストケースの再利用: テスト効率の向上
- ✓ コードカバレッジの測定: テストとコードの品質を向上

#### Test Center

テスト結果の一元管理



- 軽量のWeb UI
- テスト結果、カバレッジの一元管理
- 視覚的、統計的データの自動生成

結果のアップロード

品質の一元管理

結果のアップロード

#### Squish

GUIテスト自動化

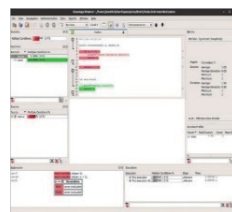


- 自動GUIテストの開発/実行
- クロスプラットフォーム
- 組込みデバイスでのテスト
- BDD(振る舞い駆動)対応
- AI拡張機能対応

カバレッジの  
取得

#### Coco

コードカバレッジ解析



- カバレッジ解析
- C、C++、QML、C#
- SystemC、Tcl
- MCDC、MCCまで対応
- 単体テスト、GUIテスト対応

#### お客様導入実績(一部抜粋)

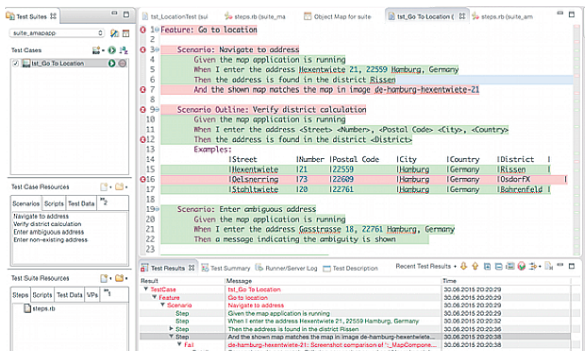


製品情報サイト

<https://www.qt.io/ja-jp/product/quality-assurance>

## Squish概要

### GUI機能を自動でテストするための クロスプラットフォームツール



### Squish IDE (GUI Tool)

- ✔ **直感的なGUI Tool (Squish IDE)**
- ✔ **テストケースのレコーディング**  
アプリ操作によるテストスクリプトを生成し記録
- ✔ **豊富な対応言語**  
Script対応言語: Python, JavaScript, Perl, Ruby, Tcl
- ✔ **検証ポイント(VP)の挿入**  
記録、またはスクリプトによる検証ポイント(VP)の挿入
- ✔ **画像ベースのテスト可能**  
プロパティベースに加えて画像ベースのテストが可能
- ✔ **AI拡張機能対応**  
テストコード改善、失敗原因解析、ログによるトラブルシューティング
- ✔ **MCP対応**  
外部AIエージェントからテストスクリプトを生成・改善、テスト実行、結果解析、ヘルプ購読 (Claude、Cursor、VS Code+GitHub Copilotなど)

### ターゲットプラットフォーム

- Windows
- Linux (Embedded Linux)
- Unix
- macOS
- iOS
- Android
- Web (HTML5)

### アプリケーション

- Qt (Qt Quick、QML、Qt Widget)
- Windows (MFC、WinForms、WPF)
- macOS app
- iOS app
- Android app
- Java
- Tk
- Web (HTML5)
- VNC

## Squishの仕組み

### Squish IDE

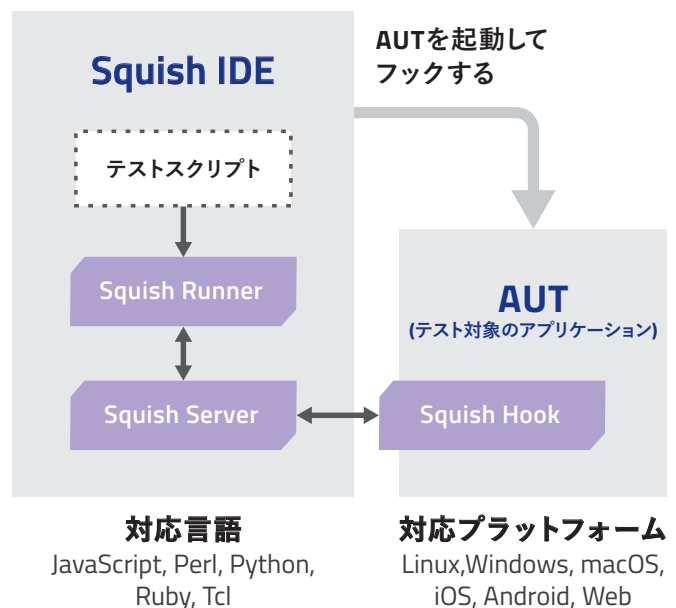
- アプリケーションのテストスイートを作成するための開発環境(Eclipse base)
- テストスイートは、スクリプト言語で記述された一連のテストケース

### Squish Runner / Server

- AUTのオブジェクトにクエリを実行するか、Squish Hookを介してコマンドを実行することにより、テストケースを実行する

### Squish Hook

- AUTのオブジェクトにアクセスし、Squish Serverと通信できるようにするためのライブラリ



お問合せ先

Qt Group日本オフィス

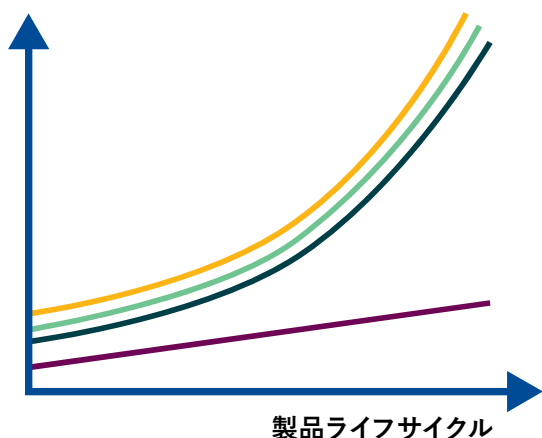
〒100-0005 東京都千代田区丸の内3-3-1 新東京ビル4F

Tel:03-6264-4500 Mail:japan@qt.io Web:https://www.qt.io/ja-jp/



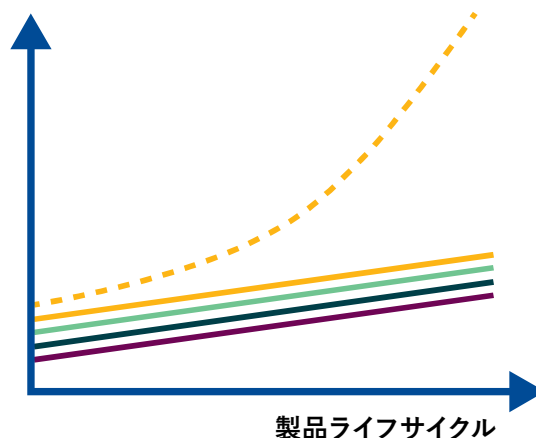


### ソフトウェア品質が改善されない原因は複雑度の上昇です



- 複雑度
- 変更あたりにかかる時間
- エラー率
- 機能数

Axivion



- 複雑度
- 変更あたりにかかる時間
- エラー率
- 機能数

アーキテクチャとコード実装の  
一致を担保する **アーキテクチャ検証**

+

リスクの高い不具合をピンポイントに  
且つ低い誤検知率で検出する **静的解析**

この2つを組み合わせることで常にクリーンなソースコードを  
保つ継続的リファクタリングのアプローチ

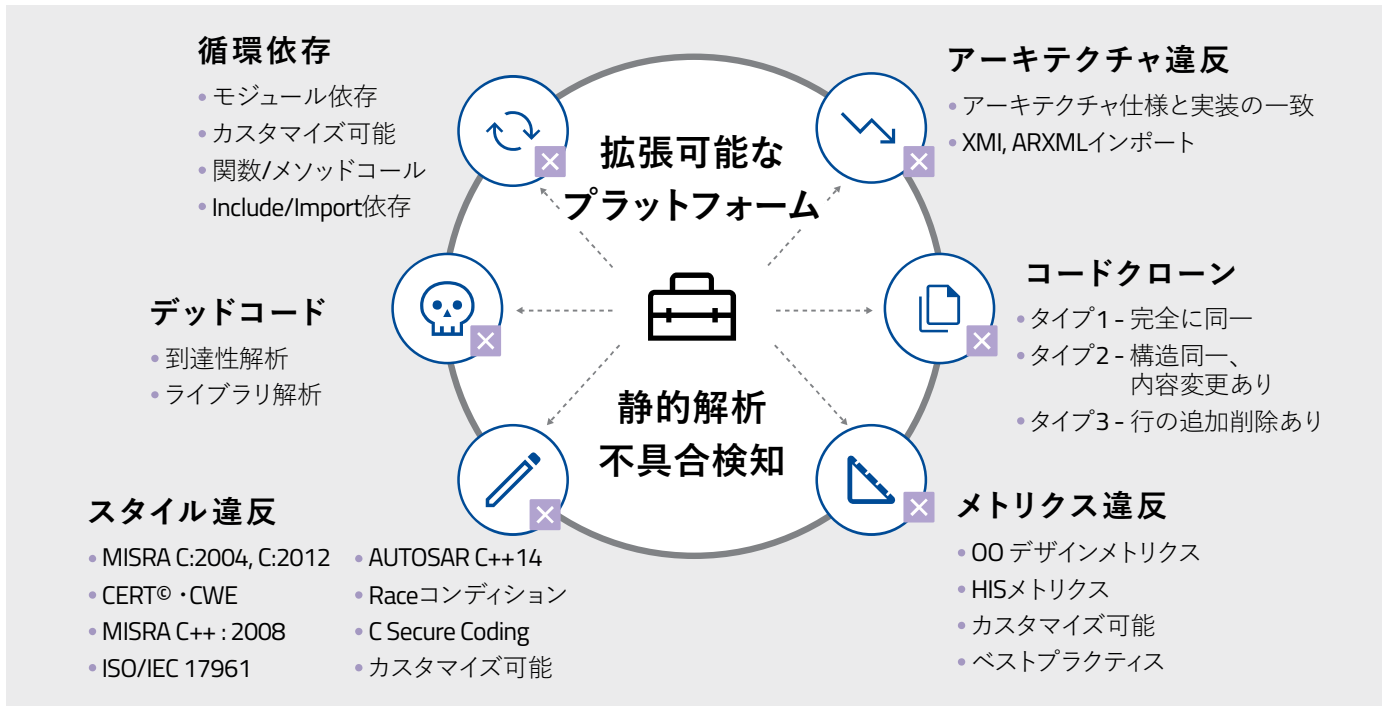
- ✓ C言語、C++、C#に対応
- ✓ ソフトウェアの複雑度を上げる6つの要因をカバー
- ✓ 豊富なAPIを提供し柔軟なカスタマイズが可能
- ✓ 独自のコーディングルールとして設定可能



製品情報サイト

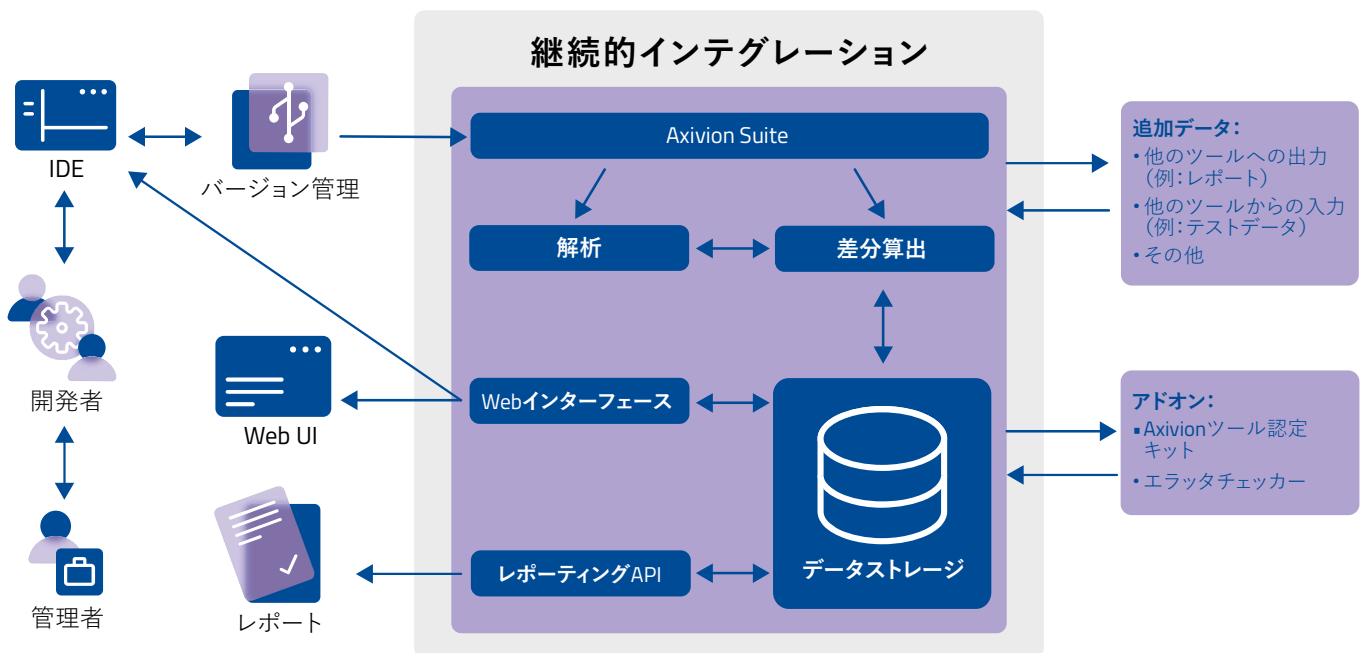
<https://www.qt.io/ja-jp/product/quality-assurance/axivion-suite>

## ソフトウェアの複雑度を上げる6つの要因



- ▶ 上記6つの要因以外にも、コーディングルール違反だけでなくランタイムエラー系のエラーも検出可能 (例:リソースリーク、NULLの間接参照、未初期化変数、など)
- ▶ アーキテクチャモデリングツールGravis (グラビス) を搭載し、アーキテクチャ仕様がない状態でもアーキテクチャモデルを作成可能

## Axivion Suiteの技術的セットアップ - 開発環境とプロセスへの統合



ご購入後はQtのサポート担当者がトレーニングとセットアップをご提供!  
 すぐに効果をご実感いただけます!

お問合せ先

Qt Group日本オフィス  
 〒100-0005 東京都千代田区丸の内3-3-1 新東京ビル4F  
 Tel:03-6264-4500 Mail:japan@qt.io

