

# CA Agile Requirements Designer



## 概要

アプリケーション・エコノミーの現在、組織はアプリケーションを継続的に提供するだけでなく、優れた顧客エクスペリエンスも達成しなければならないというプレッシャーの下に置かれています。ただし、従来のテストは時間がかかりすぎ、手作業が多すぎるため、アプリケーション品質が損なわれることが多々あります。ソフトウェアの欠陥が見逃されてしまうと、コストのかかる手直しが発生し、最終的にプロジェクトが失敗に終わることもあります。

CA Agile Requirements Designer は最小限のテスト回数で最大限の機能テストを行い、要件の変更に応じてテストの設計と実行を自動化することで、組織は迅速にアプリケーションの品質を確保でき、商品化に必要な時間とリソースも大幅に削減できます。

## 主なメリット / 成果

**テスト・コストの削減:** 不具合を早期に検出することで、修正に要する時間、リソース、コストを削減します。

**品質向上:** 可能な限り少ないテスト回数で機能を100% 網羅したテストを行うことで、テストを最適化します。

**アプリケーション・デリバリの加速:** テストの設計と実行、データ割り当て、変更の実装を自動化することで、手作業を不要にします。

## 主な特長

**優れた要件の提供:** 要件をアクティブなフローチャートで可視化し、要件のあいまいさを90%<sup>1</sup>、ソフトウェアの不具合を56%<sup>2</sup> 削減します。

**テスト・ケースの設計:** 機能を100% 網羅できる最少回数のテスト・ケースを自動的に作成し、適切なデータと予想される結果にリンクすることができます。

**テストの最適化:** テスト・ケースをインポートし、重複があればそれを取り除いて、テスト・サイクルを30% 短縮します。

**変化する要件の管理:** 変更による影響を自動的に特定し、テスト・ケースを数分以内に更新します。

**アクティブな自動化:** アプリケーションを徹底的にテストするために必要なすべてのテストを生成します。

**難易度の分析:** プロジェクトと変更依頼への対応を開始する前に時間、コスト、難易度を正確に測定します。

**データの検出と作成:** 考えられるあらゆるテストを実行するために必要な、適切なテスト・データのプロビジョニングとリンクを行います。

## ビジネス上の課題

従来型のテストは DevOps には適していません。継続的デリバリーに必要なスピードや多数の変更に適していないからです。さらに欠陥の56%<sup>2</sup> は要件が十分に定義されていないことに起因しており、これがソフトウェアの不具合とそれに伴う顧客エクスペリエンスの低下を招いています。

**あいまいな要件:** 不明確な要件を統一されていないフォーマットで保存することで、ソフトウェアの不具合、さらには顧客エクスペリエンスの低下が発生します。ビジネスと IT の間のコミュニケーション不足にもつながり、プロジェクトの遅れが発生することがあります。

**不十分なテスト範囲:** テスト・ケースを手作業で設計する方法は一般に、体系的なものではなく、時間もかかります。その結果、機能の10~20%しか網羅できず、ソフトウェアの不具合が発生してコストが増加する可能性があります。

**無駄な過剰テスト:** 4 回程度のテストであれば通常範囲内ですが、機能によっては18回も過剰にテストすることもあります。

**自動化のフレームワークに関するジレンマ:** ほとんどの自動化フレームワークは手作業のスクリプト作成に依存しており、手間がかかります。また、増加するスクリプト・ライブラリを手作業で保守するためにさらに無駄な時間が費やされています。

**日々発生する変更:** 要件が変化しても、テスト担当者はシステムへの影響を調べる方法がありません。テスト・ケースとスクリプトを1つずつ手動で確認、更新するのが一般的ですが、これでは時間がかかり、非効率です。

## ソリューションの概要

テスト自動化ソリューションの CA Agile Requirements Designer を使用すると、組織は低コストで高品質のアプリケーションを構築して、迅速な商品化が可能になります。

この製品ではフローチャート・モデルで要件が可視化されるため、組織はあいまいな要件と、それに伴う高額な不具合を解消できます。標準装備のコストと手間の分析機能を使用して、開発に要する時間とコストを算定し、予算超過や遅れに伴うリスクを軽減できます。

CA Agile Requirements Designer を使用することで、時間のかかる手作業を大幅に削減できます。テスト担当者は機能を100% 網羅できる最少回数のテスト・ケースを自動的に作成し、適切なデータと予想される結果にリンクすることができます。既存のテスト・ケースをインポートおよび最適化して、自動テストとして実行することで、平均的なテスト・サイクルを30% 減少することができます。

最適化されたテスト・ケースをプッシュし、Rally または HP ALM などの既存のツールでテスト担当者に割り当てることができます。自動化されたテストとして実行することもできるので、手作業でのスクリプト作成が不要になり、アプリケーションの詳細なテストをオンタイムかつ予算内で行えます。

CA Agile Requirements Designer は要件変更に伴って正常に機能しなくなったテストを自動的に検出および修復し、100% 網羅するために必要な新しいテストを自動的に作成します。これによって手作業でのテスト保守で発生していたボトルネックが解消され、変化するユーザ・ニーズに早期に低コストで対応できる、優れたソフトウェアを提供することができます。

[1] CA Technologies の実装経験で得たメトリクス

[2] ベンダー・レポート: 要件ベースのテスト

## 主な差別化要因

**最適化— テスト・ケース設計を超える機能**：可能な限り最少回数のテスト・ケースで最大範囲を自動的に網羅できる機能によって、品質向上とテスト・コストの削減が可能になります。範囲と難易度の分析により、100%の機能をテストできていることを確認できます。

**再利用を念頭に設計**：一元化されたライブラリに適切なデータと期待される結果をリンクし、そこからテスト・ケースの保存、共有、再利用が可能です。分散したテスト・チームは、進化するシステムを完全にテストするために必要なテストをすぐに修正し、実行することができます。

**利用性とコラボレーション**：ユーザとビジネス・アナリストはアクティブなフローチャートを使ってすぐに要件をモデル化して、あらゆるテストや開発作業でその単一のモデルを参照できます。

**オンデマンドでテスト・データを提供**：テスト・ケースの自動作成時にテスト・データが自動生成されるため、CA Agile Requirements Designerでは開発ライフサイクルで最も手作業が多く、時間もかかる2つの作業が不要になります。

**統合されたソリューション**：各種の要件定義、テスト自動化、実行用ツールを統合することで、CA Agile Requirements Designerへの投資回収率（ROI）が高まります。

## 関連製品

**CA Test Data Manager**：CA Agile Requirements Designerで作成したテスト・ケースと直接照合して、テストに必要なすべてのデータを作成します。

**CA Service Virtualization**：CA Agile Requirements Designerでモデル化した仮想エンド・ポイントに基づき、テストに必要なサービスを作成します。

**Rally**：CA Agile Requirements Designerに要件をエクスポートして、適切な一連のテスト・ケースに変換および最適化します。

## お客様の成功事例

**大手金融機関**：たった30分でテストの網羅率を5%から100%に改善<sup>1</sup>。

**多国籍銀行**：変更依頼の実装サイクル時間を99%短縮し、7.5時間から2分で実装<sup>1</sup>。

**世界的なサービス企業**：18回もの過剰なテストが不要になり、テスト実行回数を大幅に減少<sup>1</sup>。

**多国籍金融サービス・プロバイダ**：テスト・ケースの作成時間を95%短縮。

詳細については、[ca.com/jp](https://ca.com/jp)をご覧ください。

CA Technologies (NASDAQ: CA) は、企業の変革を推進するソフトウェアを作成し、アプリケーション・エコノミーにおいて企業がビジネス・チャンスを獲得できるよう支援します。ソフトウェアはあらゆる業界であらゆるビジネスの中核を担っています。プランニングから開発、管理、セキュリティまで、CAは世界中の企業と協力し、モバイル、プライベート・クラウドやパブリック・クラウド、分散環境、メインフレーム環境にわたって、人々の生活やビジネス、コミュニケーションの方法に変化をもたらしています。詳細については[ca.com/jp](https://ca.com/jp)をご覧ください。

[1] CA Technologies の実装経験で得たメトリクス

[2] ベンダー・レポート：要件ベースのテスト