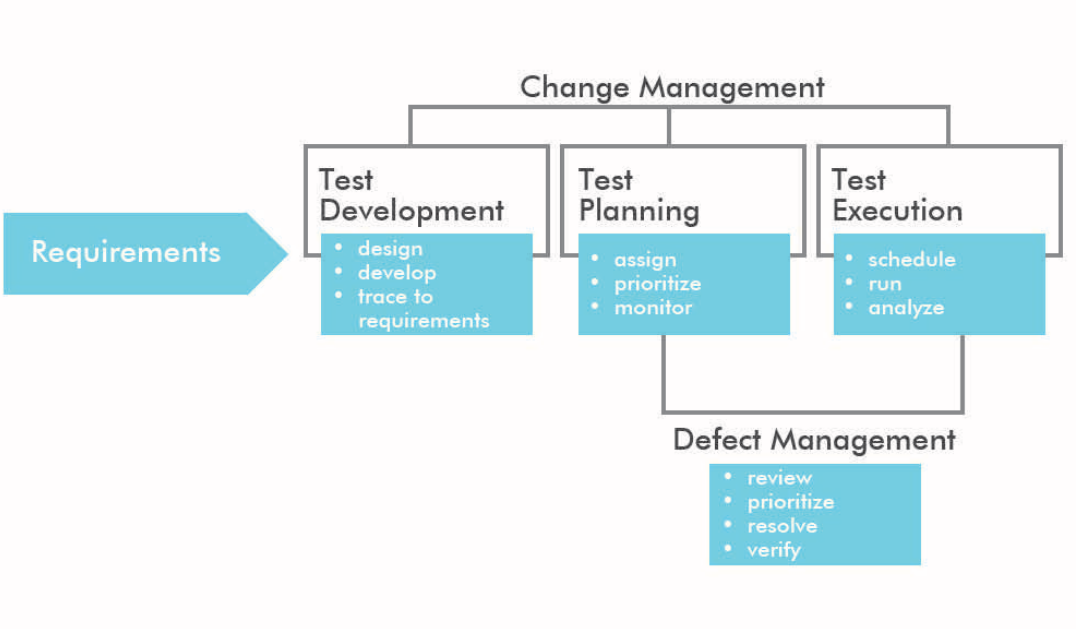
要件に基づいたテスト：トレーサビリティが実現する協調作業の促進

概要

ソフトウェア プロジェクトが失敗する原因の 50 ～ 70 % は、要件が正しく記述されていないことや周知されていないことにあります。これは、十分な資料に裏付けされた事実です。実際、ある業界調査でも、プロジェクトで確認されたエラーの 56 % は、要件策定の段階で根本原因が生じていることが判明しています。[1]

そこで今、正式な要件管理プロセスを導入して、要件に起因する失敗リスクを軽減する取り組みが企業各社の間で広まっています。この要件管理プロセスを担当するのは品質管理部門です。要件に基づくテスト手法を導入し、要件テストを網羅的に実施することで、要件管理の効率向上、失敗リスクの軽減、ソフトウェアの全体的な品質向上を実現できます。要件に基づくテストは、反復型やアジャイル型のソフトウェア開発手法を導入している企業でも広く利用されています。開発、ビルド、展開というサイクルが繰り返される中で変化していく要件を確実にカバーできるためです。

1 「What Is Requirements-Based Testing?」— ゲイリー E. モギョロディ (Gary E. Mogyorodi)、Bloodworth Integrated Technology, Inc.、STSC Crosstalk、2003 年 3 月



この技術白書では、アプリケーション ライフサイクル管理のための統一プラットフォームである PTC の Integrity が、いかにして要件に基づくテスト プロセスの導入を可能にし、次のことを実現するかを紹介します。

- テスト対象を把握して、プロジェクトのすべての要件について網羅的なテスト プランを策定、実施する
- 要件や仕様、テスト ケース、コードなど、アプリケーション ライフサイクルで作成されるすべての成果物の間にトレーサビリティを確立する
- テスト中に発覚した不具合の解決プロセスを管理する
- 指標や傾向をリアルタイムに確認できるレポートおよびダッシュボードを使用して、プロジェクトの進捗状況を把握する

要件に基づくテストとは

要件に基づくテスト、要件管理から始まります。要件管理ではまず、ビジネス アナリストやビジネス ユーザー、製品マネージャなどのビジネス担当者が、システムに対するニーズを正確かつ網羅的に記述した要件や仕様をドキュメントにまとめます。その後、品質管理部門がこれらの要件を土台にしてテスト プロセスを計画、実施します。品質管理部門では、要件に直接照らしてテスト プランとテスト ケースの実施状況を追跡しながら、プロジェクトの進展に合わせてテストの進捗状況を把握し、すべての要件を検証します。そしてすべての要件を徹底的にテストし、テスト中に見つかった不具合を解決したところで、晴れて正式リリースの運びとなります。

要件に基づくテストを実施すると、アプリケーション ライフサイクルの早い段階で問題を発見できるため、開発コストを大幅に削減できる可能性があります。ソフトウェア エラーの修正コストは、プロジェクトの工程が進むにつれて一気に上昇します。要件策定段階でエラーを発見した場合の修正コスト比を 1:1 とすると、コーディング段階では 7:1、製造段階では 40:1 ～ 100:1 にもなります。[2]

2 B. ベーム (B. Boehm)、V. バジーリ (V. Basili)、「Software Defect Reduction Top 10 List」 — IEEE Computer

要件の不具合を発見、修正するためのコスト	フェーズ	不具合の修正に必要な平均コスト	正規化した値
	要件策定	$139	1
	設計	$455	3
	コーディング	$1,000	7
	システム テスト	$7,000	50
	製造	$14,000	100

現在のテスト手法の課題

プロセスが手動

品質管理部門が手作業でテストを管理しようとすると、次のような多数の課題に直面します。

- プロセスや標準をすべてのプロジェクトに強制適用できないため、品質が安定しない、不具合があっても監査を通る、などの問題が生じる
- 手動でのレポート作成やテストのステータス評価に時間がかかる
- テスト ケースへのダイレクトリンクがないため、すばやくアクセスして要件の妥当性を確認できない
- プロジェクトに悪影響を与える要件の変更を影響分析で見抜けない
- 要件に対するテストの範囲が不完全で、エラーや漏れが生じる
- プロジェクト ライフサイクル中に行われた要件の変更を把握できず、適切なテストを行えない

複数のポイント製品やサイロ化

多くの環境では、チーム固有のニーズに応えるために、ポイント製品が場当たり的に導入されていきます。要件管理やテスト管理、問題追跡のための製品を複数導入していくと、分断した情報のサイロができあがり、協調作業に関する次のような問題が生じます。

- アプリケーション ライフサイクルの各段階を横断した経営幹部向けのレポートを単一のシステムで作成できない
- ビジネス ルールを実装し、複数の領域にわたってプロセスを強制することが困難
- 1 つの領域で行った変更が別の領域に反映されない

Integrity による要件に基づくテスト

Integrity をはじめとする、アプリケーション ライフサイクル管理のための統一プラットフォームを導入すると、品質管理をアプリケーション ライフサイクルのほかの段階と連携させ、プロセスを手動で実施している場合やポイント製品を複数導入している場合の課題を解消することができます。

変更管理

アプリケーション開発を成功に導くには、ライフサイクル全体にわたって効果的に変更管理を行える必要があります。Integrity プラットフォームのプロセス制御機能を使用すると、変更の実施を特定のユーザーや特定のタイミングだけに制限できます。また、包括的な影響分析機能により、変更を実施した場合のリスク評価を行うことも可能です。変更を実施した場合は、影響を受ける可能性のあるアイテムに要確認のフラグが付けられるため、必要に応じて修正を行うことができます。

要件フェーズと下流のテスト作業の連携

ほとんどの企業では、要件定義に Microsoft Word や Excel、Project などのオフィス ツールを使用しています。Integrity は、これらのツールで作成した要件ドキュメントも容易にインポートすることができます。また、ドキュメント作成用のインタフェースを使って、Integrity システムで直接要件を作成することも可能です。

要件をインポートまたは作成すると、Integrity のワークフロー機能により承認サイクルが開始されます。要件が承認された後、テストの作成担当者は、開発者が要件に直接リンクした機能仕様を作成するようのと同様に、要件にリンクしたテスト ケースを作成できます。また Integrity では、開発プロセスと品質管理プロセスが連携しているため、要件の変更リクエストがあった場合に、その変更が及ぼす影響を分析したり、要件に変更が加えられた場合に、変更内容を下流のプロセスに伝達したりすることができます。

Integrity が備える要件管理機能の詳細については、技術白書「An Innovative Approach to Managing Software Requirements」をご覧ください。

要件を考慮に入れたテスト プランの策定

ソフトウェア アプリケーションの要件確定後、品質管理部門は、その要件を直接の土台としてテスト プロセスを作成、実装し、システムが想定どおりに動作するかどうか、ビジネス ニーズに応えられるかどうかを検証します。このステップでは、まずテストを作成します。テストの作成担当者は、最終成果物で要件が満たされていることを確認するためのテスト ケースを要件ごとに 1 つ以上作成します。そしてこのテスト ケースを基に、テスト エンジニアが実行するテスト ステップの詳細、想定される結果、テストの評価に使用する合格 / 不合格の基準を定義します。テストを自動実行する場合は、外部の自動テスト実行ツールのスクリプトへのリンクも設定する必要があります。

要件とテスト ケースを結び付けるメリットは、ビジネス アナリストと品質管理担当者との間にフィードバック ループが確立されることです。このフィードバック ループを活用することで、要件を練り上げ、さらに製品全体を改善することが可能になります。たとえば、テストが必要な項目を細部まで確認することにより、要件の誤りや不足している要件に気付く場合があります。

自動テスト、手動テストの実施

テスト ケースが完成し、論理的なグループ分けを行ったら、テストの実行準備が整ったことになります。この段階で、品質管理マネージャがテスト エンジニアにテスト作業を割り当てます。テストを自動化している場合、Integrity は自動テスト実行フレームワークを通じて、HP Quick Test Professional や ITKO Lisa など外部のテスト実行ツールと連携できます。

Integrity は、テストの実行に必要な情報を外部システムに提供し、テスト終了後に実行結果を受け取ります。テスト結果は、Integrity で出力して内容を確認できます。テストを手動で行う場合は、テスト結果エディタを使用します。テスト結果エディタでは、実行する必要のあるテスト ステップ、想定される結果、合格 / 不合格の基準を専用のビューで確認できます。また、結果についての詳細なコメントを記録することも可能です。自動テストの結果も、テスト結果エディタで参照できます。

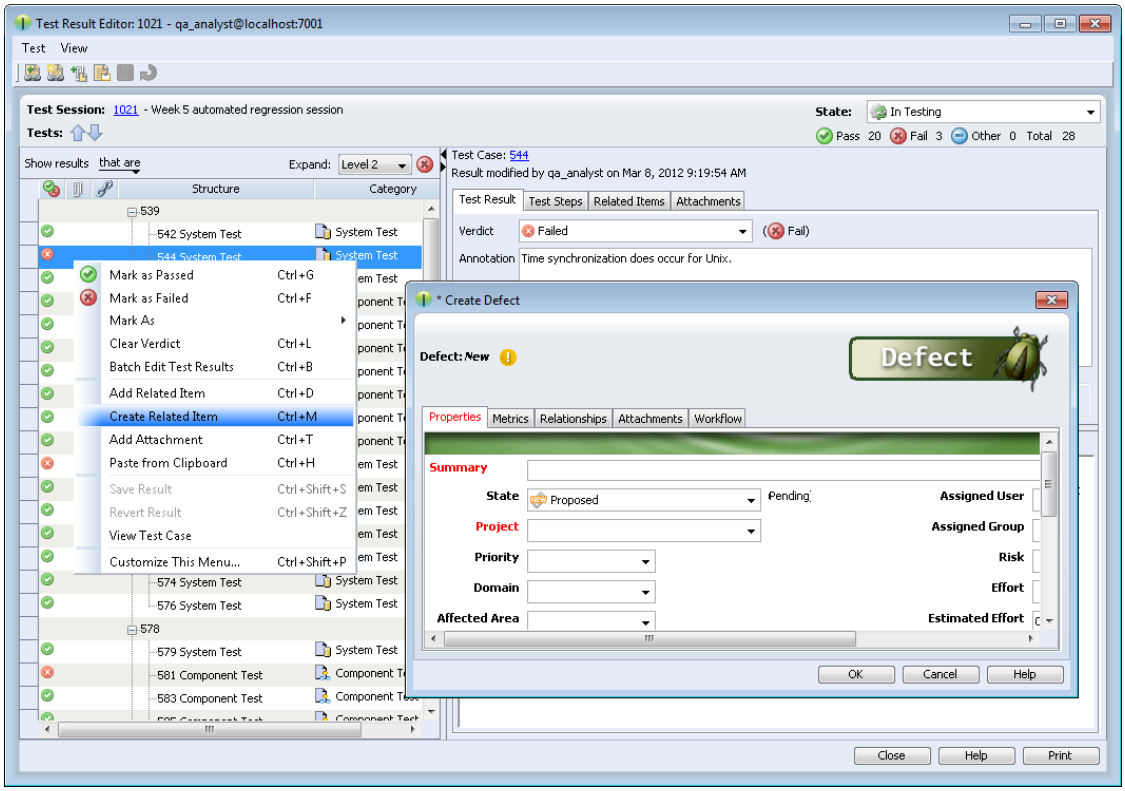
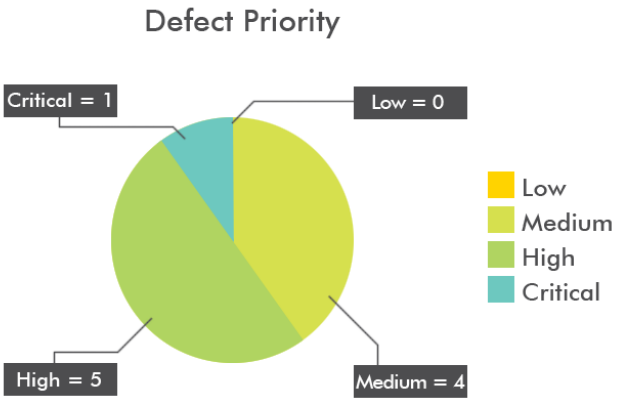


図 1：不合格となったテスト ケースと該当の要件にリンクした不具合項目の作成

不具合の修正とプロジェクト コストの削減

製品のリリース準備状況を測る重要な指標に、見つかった不具合の数と深刻度があります。先にも述べたように、不具合の発見と修正は、開発ライフサイクルの早い段階であるほど対応コストが少なくて済みます。不具合が見つかった場合は、優先順位を設定して担当の開発者に修正作業を割り当てます。このとき Integrity では、不具合の修正のために開発者が行ったすべての変更と不具合との間にトレーサビリティが自動設定されます。これにより、問題の要件から、テスト時に見つかった不具合を修正するために変更されたコードの行までを完全に追跡することが可能となります。



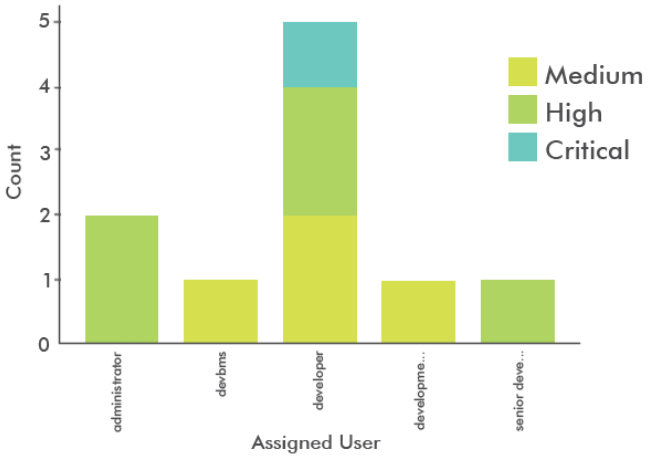


図 2、3 : 重要なテスト指標をリアルタイムで示すグラフ

グラフ、レポート、ダッシュボード

品質管理マネージャやプロジェクト マネージャ、経営幹部は、リソースの準備や確保、問題の迅速な解決、スケジュールおよび予算の管理などを的確に行うため、正確な最新情報にアクセスできる必要があります。しかし多くの環境では、レポート作成は手動で行われており、情報をリアルタイムに入手することができていません。レポート作成に人間が介在していると、結果に主観が紛れ込むおそれもあります。また、製品ごとにデータ リポジトリが独立しているポイント製品を複数使用している場合、アプリケーション ライフサイクルを一元的に把握することは困難です。

Integrity には、詳細なグラフとレポートを作成できる高度な機能が用意されています。職務別に用意されたダッシュボードでは、それぞれの担当者が自身の作業を行うために必要な情報をリアルタイムに把握できます。フィルタ機能も用意されているため、1 つのプロジェクトの情報だけを表示することも、自身が関係する複数のプロジェクトの情報をまとめて表示することも可能です。グラフやレポートの背景にあるライフサイクルの詳細を知りたい場合は、ワンクリックでデータを掘り下げることができます。たとえば、優先順位の高い不具合が 3 つ残っていることが判明した場合は、グラフの該当部分をクリックしてすぐさま問題の詳細を確認し、速やかに修正を行うことができます。

まとめ

Integrity を利用して要件に基づく自動テスト プロセスを導入すると、スケジュールや予算、ビジネス要件を順守しながらソフトウェア プロジェクトを完遂できるようになります。これは、次のことを可能にすることによって実現されます。

- プロジェクトの要件について網羅的なテスト プランを策定、実施する
- 要件の変更に起因するプロジェクトの混乱を回避する
- 一元化されたリアルタイムの情報にアクセスして、プロジェクトの進捗状況を把握する
- プロセスや標準をすべてのプロジェクトに強制適用する
- 開発ライフサイクル全体にわたって、すべての成果物の間にトレーサビリティを確立し、高い可視性を確保する

Integrity 事業所

北米 +1 800 613 7535

英国 +44 (0) 1252 453 400

ドイツ +49 (0) 711 3517 750

アジア太平洋地域 +65 6830 8338

日本 +81 3 5422 9503

integrityinfo@ptc.com

詳細については、PTC.com/products/integrity をご覧ください。

© 2012, Parametric Technology Corporation (PTC). All rights reserved. PTC 製品およびサービスに対する唯一の保証は、該当する製品およびサービスに付随する明示の保証書において定められています。本書のいかなる内容も、追加の保証と解釈されてはなりません。本文中で言及されている顧客事例は、単一のユーザー エクスペリエンスおよびかかる顧客の証言に基づいています。PTC 製品およびサービス、または PTC が関係している市場に関するアナリストなどによる将来の見通しは、そのアナリスト自身の見通しであり、PTC はその根拠または正確性について何ら言明いたしません。PTC、PTC ロゴ、およびすべての PTC の製品名およびロゴは、米国およびその他の国における PTC またはその子会社、あるいはその両方の商標または登録商標です。その他の製品名または企業名はすべて、各所有者の商標または登録商標です。新製品や新機能のリリース時期は予告なく変更されることがあります。

7513-Integrity: Requirements-Based Testing-WP-JA-0312