



# 要件管理と Visual Studio Team System

ホワイト ペーパー

2008 年 3 月

最新情報については、

<http://www.microsoft.com/japan/msdn/teamsystem/> を参照してください。



このドキュメントは暫定版であり、このソフトウェアの最終的な製品版の発売時に実質的に変更されることがあります。

このドキュメントに記載されている情報は、このドキュメントの発行時点におけるマイクロソフトの見解を反映したものです。マイクロソフトは市場の変化に対応する必要があるため、このドキュメントの内容に関する責任をマイクロソフトは問われな

いものとします。また、発行日以降に発表される情報の正確性を保証できません。

このドキュメントは情報提供のみを目的としています。明示、黙示または法律の規定にかかわらず、これらの情報についてマイクロソフトはいかなる責任も負わないものとします。

お客様ご自身の責任において、適用されるすべての著作権関連法規に従ったご使用を願います。このドキュメントのいかなる部分も、米国 Microsoft Corporation の書面による許諾を受けることなく、その目的を問わず、どのような形態であっても、複製または譲渡、あるいは検索システムに格納または公開することは禁じられています。ここでいう形態とは、複写や記録など、電子的な、または物理的なすべての手段を含みます。

マイクロソフトは、このドキュメントに記載されている内容に関し、特許、特許申請、商標、著作権、またはその他の無体財産権を有する場合があります。別途マイクロソフトのライセンス契約上に明示の規定のない限り、このドキュメントはこれらの特許、商標、著作権、またはその他の無体財産権に関する権利をお客様に許諾するものではありません。

© 2008 Microsoft Corporation. All rights reserved.

Microsoft、Excel、Outlook、Visio、および Visual Studio は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

その他、記載されている会社名、製品名には、各社の商標のものもあります。

## 目次

|  |    |
|--|----|
| 概要 .....   | 1  |
| 要件管理と要件ライフ サイクル.....                                 | 2  |
| 要件について .....   | 5  |
| Team System の要件ライフ サイクル ロードマップ .....                 | 6  |
| 要件に関するアクティビティと Team System ソリューション .....             | 7  |
| Team System の<br>次期リリース (コード名 Rosario) の要件関連機能 ..... | 23 |
| 要件関連パートナー .....                                      | 26 |
| 付録 A: Office Excel を使用した作業項目の分解 .....                | 31 |
| 付録 B: 要件品質レポート .....                                 | 32 |
| 付録 C: 要件進捗レポート .....                                 | 33 |

---

## 概要

多くのソフトウェア プロジェクトの遅延、超過、および失敗の根本的な原因は、ビジネス要件を定義、伝達、または検証できていないことにあります。業界では、組織のあらゆるレベルの要件を適切に策定することの重要性が認識されてきています。Microsoft Visual Studio Team System と Visual Studio パートナーのソリューションは、要件策定プロセスの改善に役立ちます。

このホワイト ペーパーでは、Visual Studio 2005 Team System と Visual Studio Team System 2008 を使用して、要件管理を成功に導く Visual Studio Team System の活用方法を説明することを目的としています。また、Visual Studio Team System の次期リリース (コード名 Rosario) で解決を目指している問題のいくつかについても説明します。

## 要件管理と 要件ライフ サイクル

アプリケーション ライフ サイクルから見た要件は、非常に簡単に定義できます (図 1 参照)。

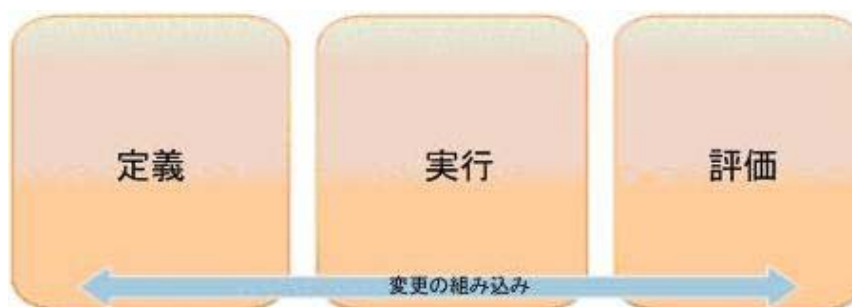


図 1. 要件ライフ サイクル

簡単に表すと、まず要件を定義します。ユーザーからの確な要件を引き出し、それを基にソリューションを開発し、作業の優先順位付け、洗練、分解、計画を行い、テストします。

次に、要件に従って作業を実行します。コードを記述し、テストを作成し、チェックインとビルドを行い、進捗状況と要件の質をトラックします。

最後に、運用環境に展開するかソリューションをリリースし、KPI、SLA、およびユーザーからのフィードバックを評価して、次の要件を洗い出します。

このプロセスの期間中には変更が発生します。たとえば、要件の変更、作業分解の変更、新しい作業の発生、作業予定時間の超過、スコーピングの発生、チームメンバの入れ替わりなどがあります。変更は避けられないため、要件のアクティビティは、一度きりのアクティビティであると捉えたり、厳密に順番どおり実施されるものであると考えることはできません。要件ライフ サイクルのどの時点でも変更が組み込まれるようにして、ライフ サイクルの各フェーズが繰り返されて、互いに影響を受けるに従い、変更がすべてのフェーズとアクティビティに波及できるようにします。

変更を組み込むためのソリューションとして、トレーサビリティが検討されることが多くあります。これは、変更管理の一環として、要件ライフ サイクルのすべてのフェーズとすべてのアクティビティに実装される必要があります。トレーサビリティは、各アクティビティの作業を意味のある形で関連付けるもので、通常、アプリケーション ライフ サイクル管理 (ALM) ツールでは、リンク リレーションシップによって表されます。トレーサビリティを実現するには、要件管理の各フェーズとアプリケーション ライフ サイクルの全期間を通じて使用される

各種ツールを統合して、関連する作業と成果物の正確かつ最新の状態が常に反映されるようにする必要があります。

変更を組み込むうえで他に必要なことは、要件ライフ サイクルの全期間で、頻繁にユーザーと連絡を取ることです。ユーザーのニーズの変化とソリューションの進化に合わせて、ソリューションがニーズを満たしているかどうかを定期的に検証するのは当然と言えます。成功しているプロジェクトでは、ユーザーと定期的に連絡を取り、作業とニーズの変化に常に対応できるようにしています。

Project Management Body of Knowledge Guide (PMBOK) をはじめ、いくつかの資料では、要件に関するアクティビティをより細かく定義しています。これらの資料で定義されているアクティビティは、要件ライフ サイクルに当てはめると、図 2 のようになります。

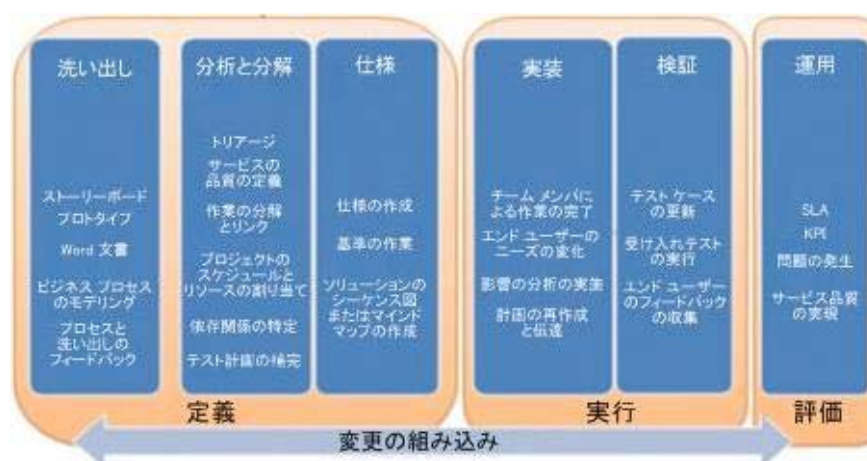


図 2. 要件に関するアクティビティ

すべてのプロジェクトや組織で、これらのアクティビティすべてが必要になるわけではありません。多くの場合、上記の分類では細かすぎるかもしれません。要件管理プロセスは、異なるチームやプロジェクトのさまざまなニーズに合わせて調整する必要があります。各アクティビティが変更を組み込み、他のアクティビティと連携して、アプリケーション ライフ サイクルを支える切れ目のない要件ライフ サイクルを実現することが重要です。

Visual Studio Team System は、要件ライフ サイクルの基礎となる統合プラットフォームとして開発されています。共通のデータに基づくチームの共同作業を実現する統合プラットフォーム、プロセスに含まれる作業のレベルの定義、およびすべての作業を統合するプラットフォームがあれば、要件ライフ サイクルとアプリケーション ライフ サイクルの全期間を通じて、一定のレベルで状況を把握し、変更を管理することが可能になります。

## 要件管理と要件の洗い出し

多くの人が、要件ライフ サイクル全体を要件管理と同じであると考えています。しかし、要件の洗い出しは、要件管理とは別のものとする人が増えてきています。この区別は、要件ライフ サイクルのソリューションを提供するうえで Team System が従うロードマップに、非常にスムーズに当てはめることができます。

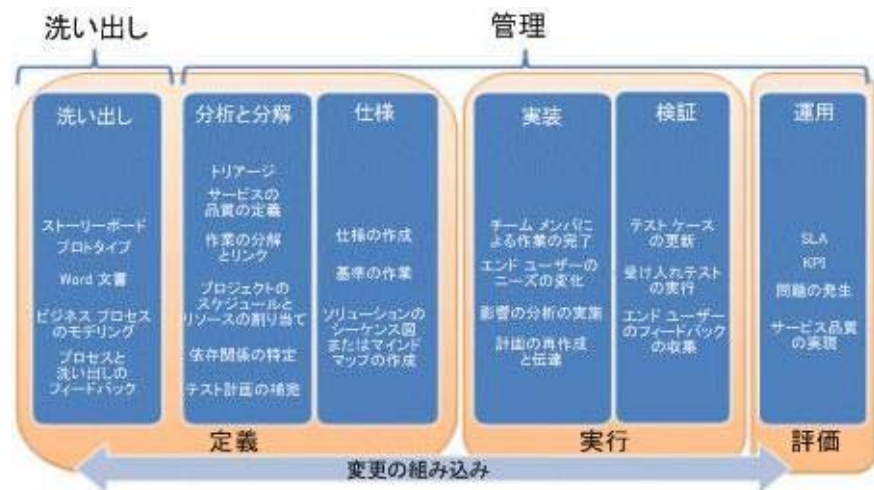


図 3. 洗い出しと管理

---

## 要件について

要件とは、エンド ユーザーのニーズまたはアプリケーションに求められるサービスのレベルを表す業界用語です。概して、要件の分類の細かさと意味はさまざまです。一般的なビジネスにおいては、要件は次のように分類できます。

- **ビジネス要件**：システムの全体的な目的や、システムがビジネスにどのように影響するか
- **ユーザー要件**：ユーザーのニーズと問題
- **機能要件**：システムの特定の動作とソフトウェア内部の働き
- **サービスの品質に関する要件**：システム全体に関するもので、ローカライズの可否、アクセシビリティ、パフォーマンス、スケーラビリティなどの "能力" を表します。

Team System を使用した要件管理の最初の作業は、要件の作業項目を作成することです。これにより、開発プロセスの全期間を通じて要件をトラックし、トレースできるようになります。

多くの組織では、ビジネス アナリストが、エンド ユーザーの公式な窓口となり、要件の洗い出し、洗練、分析、検証を行い、要件の進捗状況を報告する責を担っています。ただし、開発者、テスター、プロジェクト マネージャなど、さまざまな役割のスタッフが、要件の収集に関与していたり、その責任を負っていたりします。ソフトウェア エンジニアリング プロセスで要件がどのように定義されているか、だれが要件収集を担当するかに関係なく、Visual Studio Team System Team Foundation Server は、要件の作業項目を作成するときの要件を収集したり、保存したりするのに便利なツールです。



## TEAM SYSTEM の 要件ライフ サイクル ロードマップ

Team System は、要件ライフ サイクルのアクティビティのいくつかをサポートするほか、パートナーが連携し、スムーズに統合できるアクティビティを追加するための充実した環境を提供します。目標は、プロセスとプロセス テンプレートの定義に従って、このようなアクティビティが Team System プラットフォーム上で統合され、実施されるようにすることです。

Team System は、実装フェーズの問題に対処するために誕生しました。Visual Studio は開発者向けの実績ある生産性ツールです。Team System は Visual Studio と統合して、開発者やテスターが各自の仕事を完了するためのプロセスを実現します。要件ライフ サイクルに取り組む際の戦略は、まず実装フェーズから着手して、そこを起点にいくつかのリリースを経ながら発展させます。このホワイト ペーパーでは、まず Visual Studio 2005 Team System と Visual Studio Team System 2008 を使用して実現できることを説明してから、Visual Studio Team System のロード マップとパートナー製品について説明します。

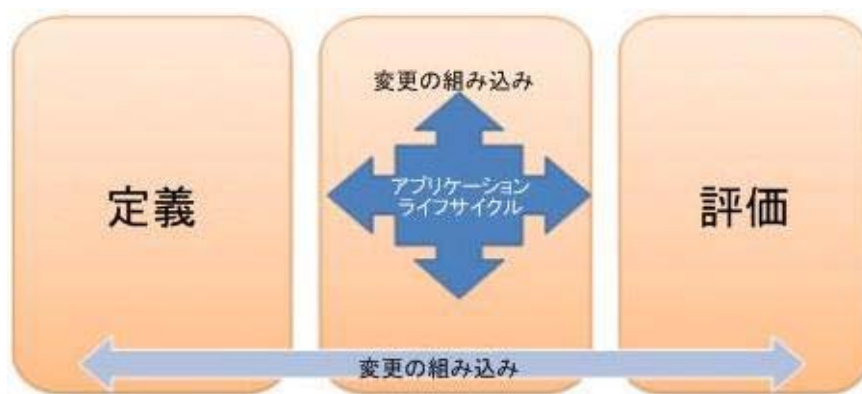


図 4. マイクロソフトの要件ライフ サイクル ロードマップ

---

## 要件に関するアクティビティ と TEAM SYSTEM ソリュ ーション

このセクションでは、要件に関する各アクティビティと、既にリリースされている Team System とパートナー製品を利用して、要件の洗い出し、洗練、指定、実装、検証を行う（アプリケーション ライフ サイクル中にこれらをトレースする）方法について、詳しく説明します。

### 要件の洗い出しと洗練

要件の洗い出しと洗練は、エンド ユーザーから要件を収集する作業です。成功するプロジェクトでは、定期的にエンド ユーザーと連絡を取り、プロジェクト ライフ サイクルの全期間を通じて要件を洗い出しています。通常、要件の洗い出しのアクティビティと洗練の大半は、プロジェクトのライフ サイクルの早い段階で行われますが、エンド ユーザーと継続して連絡を取ることで、最終的な製品のユーザー満足度を最大限に高めることができます。要件の洗い出しでは、ユーザーのニーズをすばやく察知し、ユーザーにソリューションのアイデアを説明し、優先順位とスケジュールの同意を得て、さらにソリューションのアイデアを再度説明します。このプロセスを通じて要件は追加、削除、または変更されますが、これは次のような理由から難しい作業です。

- **エンド ユーザーや顧客が、必要としているものをなかなか説明できない。**  
「どのようなものかはわからないのですが、見せていただければ、それが良いかどうかは言うことができます」などと言われたりします。
- **エンド ユーザーの説明は、ほとんどの場合不完全である。** エンド ユーザーはプロダクト デザイナーではないので、エンド ユーザーが提案したソリューションが完全にニーズを表していることは、まずありません。だれかが、デザインを隔々まで十分に検討する必要があります。
- **複数の異なるツールを使用して作業が行われる。** 要件の洗い出しに使用するツールや手法はいくつかあります。洗い出しのプロセスが進むにつれて、要件が複数の異なるツールに分散する可能性があります。そうになると、要件の優先順位や状態を一目で判断するのは困難です。

要件の洗い出しには、個人の好みに合ったさまざまなツールが必要です。また、洗い出しプロセスの進行に合わせて、適宜ツールを変更することも必要です。

| アクティビティ    | 目的/スケジュール                               | 適したツール  | パートナーのツール   |
|------------|---|---|---|
| 初期段階での記録   | ユーザーの説明に従って、すばやくニーズを書きとめることができる。        | テキストベースのツール (Office Word など)                    | Personify<br>Design<br>TeamSpec<br><br>Borland<br>CaliberRM<br><br>Techno<br>Solutions<br>TopTeam<br>Analyst  |
| 大枠のビジョンの定義 | プロジェクトの目的と、成功するための主な条件を記述する。            | テキストベースのツール (Office Word など)                    | Personify<br>Design<br>TeamSpec<br><br>Borland<br>CaliberRM<br><br>Techno<br>Solutions<br>TopTeam<br>Analyst  |
| 要件の関連付け    | 関連する要件をグループ化することで、問題やソリューションを検討する一助とする。 | データベースやスプレッドシートにおける階層<br><br>マインド マップ           | Borland<br>CaliberRM<br><br>Techno<br>Solutions<br>TopTeam<br>Analyst<br><br>Mindjet RM   |
| 全要件の確認     | 要件の一覧や要件間のリレーションシップを確認することで、要件を把握する。    | スプレッドシート<br><br>リレーションシップを示すグラフ<br><br>マインド マップ | Mindjet RM<br><br>Ravenflow<br>RAVEN<br><br>Borland<br>CaliberRM<br><br>Techno<br>Solutions<br>TopTeam<br>Analyst<br><br>Compuware<br>OptimalTrace<br><br>stpsoft stpBA |

| アクティビティ                | 目的/スケジュール   | 適したツール   | パートナーのツール  |
|------------------------|---|--|--|
| トリアージ                  | 全要件の一覧を確認し、優先順位や記述などを調整する。  | スプレッドシート   |  |
| ユーザーによるソリューションの使用方法の定義 | <p>ユーザーの操作手順に従ってエクスペリエンスを記述する。</p> <p>ソリューション内で使用されるアクタとアクションを確認する。</p> | <p>ワープロ ツールを使用して記述したテキスト ベースのユーザー ストーリーやシナリオ</p> <p>UML によるアクティビティ図</p> <p>UML によるシーケンス図</p> | <p>Personify Design TeamSpec</p> <p>Borland CaliberRM</p> <p>Techno Solutions TopTeam Analyst</p> <p>Ravenflow RAVEN</p> <p>Compuware OptimalTrace</p> <p>TechnoSolutions Visual UseCase</p> |
| ユーザー エクスペリエンスの定義       | ユーザーが目にするエクスペリエンスとして、ソリューションを表す。  | <p>ストーリーボード ツール</p> <p>プロトタイプ</p>  | <p>Compuware OptimalTrace</p> <p>stpsoft stpBA</p>   |
| 契約の作成                  | 双方の同意が得られた一連の要件を記録する。   | テキスト ベースのツール (Office Word など)  | <p>Personify Design TeamSpec</p> <p>Borland CaliberRM</p> <p>Techno Solutions TopTeam Analyst</p>  |
| ビジネスプロセスのモデリング         | アプリケーションのビジネス ロジック、イベント、ルールのフローを表現する。                                   | モデリング ツール  | <p>stpsoft stpBA</p> <p>TechnoSolutions TopTeam Analyst/Visual UseCase</p> <p>Holocentric Modeler</p>  |

| アクティビティ            | 目的/スケジュール                         | 適したツール  | パートナーのツール   |
|--------------------|-----------------------------------|---|---|
| 技術的なソリューションの定義     | ソリューションの技術的な要件と、構築方法を定義する。        | プロトタイプ<br>データフロー<br>シーケンス図<br>マインドマップ                 | stpsoft stpBA<br>Mindjet RM<br>Ravenflow<br>RAVEN<br>TechnoSolutions Visual UseCase |
| エンドユーザーのフィードバックの使用 | 提案したソリューションに対するユーザーのフィードバックを洗い出す。 | 多数決ツール<br>ツールを利用した、ソリューションについてのコメント<br>バグレポート、フィードバック |   |

Visual Studio 2005 Team System と Visual Studio Team System 2008 は、ライフサイクルのどの時点でも、要件がシステムに作業項目として入力されるとすぐに、要件が反映されるようにします。要件の洗い出しには、いくつかの方法があります。1 つは、スプレッドシートを使用して要件を洗い出す方法で、Microsoft Office Excel の統合機能を利用して Team Foundation Server に要件を入力します。別の方法としては、チーム エクスプローラから Team Foundation Server 要件作業項目フォームに直接要件を入力することもできます。この情報を入力することは、すべての要件を把握し、要件をトリアーゼするうえで役立ち、アプリケーション ライフ サイクルの他のフェーズとも統合されます。

また、要件の洗い出しと Team Foundation Server への要件の入力を支援するパートナー ソリューションもいくつか提供されています。詳細については、このホワイト ペーパーの最後にある「要件関連パートナー」を参照してください。

Visual Studio Tools for Office 2005 Requirements Authoring Starter Kit では、Office Word と Team Foundation Server の基本的な統合機能を提供しています。このツールを使用すると、Team Foundation Server とのテキスト統合を実現できます。

---

## 要件の分析と分解

要件の分解は、ユーザーが抱える問題とニーズをソリューションと作業に変換する作業です。要件は、機能要件またはサービスの品質要件のどちらかに分類されます。これらは分解されて階層構造を形成し、サブ要件が作成されます。タスクが作成され、依存関係と受け入れテストが特定されます。要件の洗練を行う際には、次のような課題があります。

- **エンド ユーザーは最終的な製品（アプリケーション）を考え、開発チームはそれを構築するための途中の手順を考える。** ソフトウェアのエンドユーザーは、機能、コンポーネント、または開発チームが容易に理解できる成果物として要件を説明できるわけではなく、必要とするものをエンジニアリングの計画に変換するにはサポートが必要です。
- **開発チームは、最もコスト効率の良い計画を採用できるよう、ユーザーに対してソリューションを繰り返し提示する必要がある。** 開発チームは要件を解釈し、コストを計算する必要があります。また、その結果を複数の代替案と併せて、エンド ユーザーに提示する必要があります。エンド ユーザーに、ソリューションについて一通り説明して、コストとデザイン上のトレードオフが理解されるようにする必要があります。それにより、エンド ユーザーからソリューションに対するフィードバックを得ることができるようになります。ソリューションを効果的に説明し、エンド ユーザーがデザイン上のトレードオフによるコストを理解できるようにすることは難しい作業です。
- **複数の異なるツールを使用して作業が行われる。** 要件は複数の異なるツールで収集されるため、開発チームがすべての要件を入手しているという保証はありません。これについては、ビジネス アナリストに確認してもらう必要があります。

| アクティビティ | 目的/スケジュール  | 適したツール  | パートナーのツール   |
|---------|--|---|---|
| 全要件の確認  | 要件の一覧や要件間のリレーションシップを確認することで、要件を把握する。                 | スプレッドシート<br>リレーションシップを示すグラフ<br>マインド マップ   | Mindjet RM<br>Ravenflow<br>RAVEN<br>Borland<br>CaliberRM<br>Techno<br>Solutions<br>TopTeam<br>Analyst<br>Compuware<br>OptimalTrace<br>stpsoft stpBA<br>Personify<br>Design<br>TeamLook、<br>TeamSpec |
| トライアージ  | 全要件の一覧を確認し、優先順位、記述、カテゴリなどを調整する。                      | スプレッドシート  |   |
| 作業の分解   | 関連する要件をグループ化することで、問題やソリューションを検討し、サブ要件やタスクを作成する一助とする。 | データベース<br>における階層<br>スプレッドシート<br>における階層<br>プロジェクト計画<br>における階層<br>Office Word<br>における階層<br>マインド マップ | Personify<br>Design<br>TeamSpec<br>Mindjet RM<br>Ravenflow<br>RAVEN<br>Borland<br>CaliberRM<br>Techno<br>Solutions<br>TopTeam<br>Analyst<br>Compuware<br>OptimalTrace<br>stpsoft stpBA              |

| アクティビティ     | 目的/スケジュール   | 適したツール   | パートナーのツール  |
|-------------|---|--|--|
| 依存関係の特定     | 共有作業を特定し、作業の順序を付ける。                                 | リレーションシップを示すグラフ<br>マインド マップ<br>プロジェクト計画におけるリンク | Personify<br>Design<br>TeamSpec<br>Mindjet RM<br>Ravenflow<br>RAVEN<br>Borland<br>CaliberRM<br>Techno<br>Solutions<br>TopTeam<br>Analyst<br>Compuware<br>OptimalTrace<br>stpsoft stpBA |
| リソースの割り当て   | プロジェクト マネージャとリードにより、作業を完了するための適切な人材配置を行う。           | スプレッドシート<br>プロジェクト計画<br>カスタムのスケジュール アプリケーション   |  |
| 作業のスケジュール作成 | プロジェクト マネージャとリードにより、コストの見積もり、期間、プロジェクトのスケジュールを決定する。 | スプレッドシート<br>プロジェクト計画<br>カスタムのスケジュール アプリケーション   |  |
| テスト計画の補完    | テスト計画のテストケースを要件に関連付け、すべての要件が確実にテストされるようにする。         | スプレッドシート<br>テスト管理システム                          |  |
| 受け入れテストの策定  | ユーザー機能を検証するテストを列挙および定義する。                           | スプレッドシート<br>手動のテストツール                          |  |

Visual Studio 2005 Team System と Visual Studio Team System 2008 では、Team Foundation Server 内のデータとスプレッドシートまたはプロジェクト計画に含まれているデータを統合できます。データが要件作業項目として入



---

力される際、開発チームはこの統合機能を使用して要件を確認できます。ビジネスアナリストも開発チームも、要件の一覧を参照して、現在のすべての要件を確認でき、要件をトリアージしたり、要件がユーザーの現在の優先順位に従っていることを確認したりできます。

要件は、Visual Studio 2005 Team System と Visual Studio Team System 2008 と連携して Office Excel または Office Project を使用して分解できます。Office Excel を使用する場合は、要件のタイトルのフォーマットを変えて、子の要件であることを示すことができます。分解を示す作業項目の種類を切り替えてタスクにすることもできます。この関連を示す情報は .xls ファイルにのみ保持され、Team Foundation Server では保持されません。作業のリレーションシップを理解し、作業分解の最新の状態を確認するには、.xls ファイルを開き直してデータを更新する必要があります。付録 A には、この方法で Office Excel を使用して実行された作業分解の例を掲載しています。

ただし、Office Project を使用して作業項目をインデントすることで作業分解を行う方が、より一般的です。この場合も、関連を示す情報は .mpp ファイルにのみ保持され、Team Foundation Server には保存されません。

依存関係の管理も、Visual Studio 2005 Team System と Visual Studio Team System 2008 で同様に実行できます。Office Project または Office Excel に保持されている、作業項目間の依存関係を示すデータは、Team Foundation Server で "関連" リンクが手動で入力されない限り、.xls または .mpp ファイルにしか保持されません。Team System レポート ウェアハウスからレポートを生成するには、Team Foundation Server でリンクを手動で入力する必要があります。最新の依存関係情報は、.xls または .mpp ファイルを再度参照して、データを更新することで、確認できます。

Visual Studio 2005 Team System と Visual Studio Team System 2008 と連携して、リソースを割り当て、作業をスケジュールするツールとしては、Office Project か Office Excel を使用できます。作業項目は、Team Foundation Server クライアントから直接割り当てすることもできます。

要件がテストされていることを確認するには、テスト結果を要件と関連付ける必要があります。これは、テストの実行または作業項目フォーム自体から直接処理できます。この処理が済むと、すべての要件に対してテストが用意されているかどうかと、テスト結果を示すレポートを生成できるようになります。

---

付録 B では要件のテスト対応状態を示すレポートの例を、付録 C には要件の進捗レポートの例を紹介しています。Team System の将来のリリースでは、このプロセスがよりスムーズに実行できるようにしたいと考えています。

## 要件の指定

ソリューションを記述するプロセスは、要件の指定と呼ばれます。このプロセスのツールとアクティビティは、「要件の洗い出しと洗練」のセクションに記載されているものと多くが同じです。要件の指定では、同意が得られるソリューション、またはたたき台となる基準のソリューションを策定することに重点が置かれます。

| アクティビティ                | 目的/スケジュール  | 適したツール  | パートナーのツール  |
|------------------------|--|---|--|
| ユーザーによるソリューションの使用方法の定義 | ユーザーの操作手順に従ってエクスペリエンスを記述する。<br><br>ソリューション内で使用されるアクタとアクションを確認する。 | ワープロ ツールを使用して記述したテキストベースのユーザーストーリーやシナリオ<br><br>UML によるアクティビティ図<br><br>UML によるシーケンス図 | Personify<br>Design<br>TeamSpec<br><br>Mindjet RM<br><br>Ravenflow<br>RAVEN<br><br>Borland<br>CaliberRM<br><br>Techno<br>Solutions<br>TopTeam<br>Analyst<br><br>Compuware<br>OptimalTrace<br><br>stpsoft stpBA |
| ソリューションの指定             | ソリューションを記述する。  | テキストベースのツール (Office Word など)  | Personify<br>Design<br>TeamSpec<br><br>Ravenflow<br>RAVEN<br><br>Borland<br>CaliberRM<br><br>Techno<br>Solutions<br>TopTeam<br>Analyst<br><br>Compuware<br>OptimalTrace  |

| アクティビティ        | 目的/スケジュール                           | 適したツール                                  | パートナーのツール  |
|----------------|-------------------------------------|---|--|
| 契約の作成          | 双方の同意が得られた一連の要件を記録する。               | テキスト ベースのツール (Office Word など)           | Personify<br>Design<br>TeamSpec<br><br>Ravenflow<br>RAVEN<br><br>Borland<br>CaliberRM<br><br>Techno<br>Solutions<br>TopTeam<br>Analyst<br><br>Compuware<br>OptimalTrace  |
| 技術的なソリューションの定義 | ソリューションの技術的な要件と、構築方法を定義する。          | プロトタイプ<br>データ フロー<br>シーケンス図<br>マインド マップ | Personify<br>Design<br>TeamSpec<br><br>Mindjet RM<br><br>Ravenflow<br>RAVEN<br><br>Borland<br>CaliberRM<br><br>Techno<br>Solutions<br>TopTeam<br>Analyst<br><br>Compuware<br>OptimalTrace<br><br>stpsoft stpBA |
| 基準のソリューションの策定  | ある時点の要件と提案されたソリューションのスナップショットを作成する。 | データベースのスナップショットとバージョン管理                 |  |

Visual Studio 2005 Team System と Visual Studio Team System 2008 では、Visual Studio チーム エクスプローラにおいて Office SharePoint のドキュメント ライブラリとの統合を実現しています。このため、Visual Studio 環境を離れずに、容易に指定の内容を参照できます。また、作業項目内から要件を指定

---

しているドキュメントや他のファイルへのリンクを設定できます。これにより、作業から作業の記述への基本的なトレーサビリティが提供されます。ただし、変更管理、ファイル内のテキストや他のコンテンツとの統合機能は提供されません。Office Word とのより密接な統合については、パートナー製品が多くの領域に対応しています。詳細については、このホワイト ペーパーの最後にある「要件関連パートナー」を参照してください。

## システムの分析とデザイン

要件を洗練して、実装用の統合されたアーキテクチャとコード スケルトンを作成できるように、このプロセスを支援するツールとダイアグラム サーフフェイスがいくつかあります。チームでは、ソリューションがエンド ユーザーのニーズを満たしていることを確認するために、このプロセスにおいて要件のトレースが必要になる場合があります。ただし、このホワイト ペーパーでは、要件管理のこの部分については取り上げません。

## 要件の実装

ビジネス アナリストが、要件の進捗状況についてレポートし、アプリケーション ライフ サイクルの全期間を通じて要件の変更を管理します。要件がチームの作業として実装されるまでのトレーサビリティは、ユーザーと作業に対する変更の影響を管理するうえで重要です。これは、チーム全体で変更の影響を評価するか、要件管理ツールを使用してリンク リレーションシップを確認して影響を評価することで、手動で行うことができます。作業が完了したら、開発チームとテスト チームが適切なリンク リレーションシップを作成および更新して、要件のトレーサビリティを維持することが重要です。このプロセスは、次のような理由から、非常に困難なプロセスです。

- **要件は変化するもので、開発チームは速やかに、かつ適切に調整する必要があります。** 完全で明確な要件を最初に定義できたとしても、プロジェクトは常に変化しています。ユーザーのニーズはビジネスの変化に合わせて変化し、開発チームの処理能力も変化し、デザインも変化します。これらすべてが同期した状態を保つことは難しく、古い状態のままであるものが出てくる可能性が十分にあります。
- **開発作業が変化しても、ユーザーに状況が伝わらない。** 作業分解は開発チームが理解できる形で処理されていますが、ビジネス アナリストがユーザーの要件に対するその影響を評価して伝えることは困難です。
- **トレーサビリティを提供するツールは、手動による処理が多く、保守が難しい。** 要件と関連する作業とテスト間のリンクを十分に考えて設定しても、変

更の評価は手動による処理が多いプロセスです。リンクが古くなったり、変更の評価担当者にとって意味がない場合は、リンクの作成にかけた投資は、そのコストに見合うものではありません。要件管理ツールの多くは、リンクリレーションシップを容易に作成および保守できないことが原因で使われなくなっています。

| アクティビティ            | 目的/スケジュール   | 適したツール   |
|--------------------|---|--|
| 作業の完了              | タスクの更新、コードの記述、テストの実行、バグの記録、タスクまたは要件のソースコードへのリンクを行う。 | 作業項目管理とソースコード管理の統合ツール  |
| 要件の更新              | ユーザーのニーズの変化に合わせて、要件の範囲、優先順位などを更新する。                 | スプレッドシート<br>データベース エントリ<br>要件の洗い出しツールに直接入力された情報                                    |
| 作業分解の更新            | 作業分解、スケジュール、作業割り当てに対する要件の変更の影響を評価する。                | データベースにおける階層<br>スプレッドシートにおける階層<br>プロジェクト計画における階層<br>Office Word における階層<br>マインド マップ |
| 依存関係の更新            | 依存関係に対する要件の変更の影響を評価する。                              | リレーションシップを示すグラフ<br>マインド マップ<br>プロジェクト計画におけるリンク                                     |
| ユーザーとチームメンバへの変更の伝達 | 開発チームとテストチームが取り組んでいるタスクが適切であることを保証する。               | プロジェクト計画<br>Office Excel スプレッドシート<br>作業キュー   |

Visual Studio 2005 Team System と Visual Studio Team System 2008 では、どちらでも作業項目管理、ソースコード管理、Office Excel、および Office Project 間の統合機能を提供しています。この統合機能により、実装アクティビティと作業計画間のトレーサビリティが実現されます。開発者またはテスターが作業をチェックインするときに、タスクの完了をマークできます。このタスクと

---

スケジュール情報を保持しているプロジェクト計画またはスプレッドシートは、Team Foundation Server の Office 統合機能を使用して簡単に更新できます。作業分解構造を保持している .xls または .mpp ファイルは、定期的に関いて、現行の Team Foundation Server データを使用して更新し、要件と開発作業の同期が保たれるようにします。また、Team System レポート ウェアハウスで要件の進捗状況についてのレポートを生成するには、ユーザーが手動でタスクを要件に関連付ける必要もあります (付録 C 参照)。

アラートを設定して、要件の作業項目の変更を通知できます。このアラートでは、受信者に要件の進捗状況と作業分解シートを確認するよう促し、ユーザーのニーズの変化を常に把握できるようにします。チームが実行している作業も、常に変化しています。チームの進捗状況も監視して、要件作業分解のレポートやビューのデータを定期的に更新し、要件への影響を特定する必要があります。

開発者またはテスターがコードをチェックインしたら、このチェックインは関連するタスクに関連付けられて、適切なタスクがクローズされる必要があります。チェックイン ポリシー機能は、Visual Studio 2005 Team System と Visual Studio Team System 2008 のどちらにも備わっています。ポリシーを設定すると、各チェックインに関連する作業項目が必要になり、関連する作業項目がない場合は、チェックインが失敗します。このポリシーは、開発者やテスターがリアルタイムで進捗状況をレポートするうえでも、適切なソース コードの成果物がタスクと要件にリンクされるようにするうえでも便利です。また、これを基に、プロジェクト計画やスプレッドシートを容易に更新できます。

統合ツールと作業項目/成果物間のリレーションシップがリンクされたら、レポートを生成して、ビジネス目標に対する進捗状況を継続して監視できるようになります。また、チーム メンバ全員が、作業と関連する要件を理解できるようになります。付録 B には要件品質レポートを、付録 C には要件進捗レポートを掲載しています。これらのレポートは、ある 1 つの要件にリンクされているすべての作業の進捗状況を示しています。作業に依存関係や作業分解項目があるかどうかは示されていません。Visual Studio 2005 Team System と Visual Studio Team System 2008 では、異なるリレーションシップを除外することは困難です。この問題は、コード名 Rosario の Visual Studio Team System リリースで解決される予定です。

## 要件の検証

ビジネス アナリストは、ソリューションの任意のビルドを使用して、手動の受け入れテストを実行し、エンド ユーザーの要件を検証します。ただし、次の点に注意してください。

- **テスト チームが以前の一連の要件をテストしている間に、要件が変更される。**  
テスト チームは要件が変更されたことを知らず、テストが要件を十分にカバーしていないことを認識していません。ビジネス アナリストの手元には、変更された要件に基づく受け入れテストを行うための最新の手動テストがありません。
- **要件をいつテストできるようになるかわからない。** 要件が確定するタイミングやどのビルドを使用すべきかを判断することは、困難です。

| アクティビティ             | 目的/スケジュール                         | 適したツール   | パートナーのツール                       |
|---------------------|-----------------------------------|--|---------------------------------|
| プロトタイプ              |                                   |  |                                 |
| テスト ケースの更新          | テスト ケースが最新の要件を確実にカバーするようにする。      | テスト オートメーション ツール<br>スプレッドシート<br>テスト ケースや要件についてのレポート    |                                 |
| 受け入れテストの実行          | 必要なエンド ユーザー機能を検証する。               | スプレッドシート<br>手動のテスト ツール                                 |                                 |
| エンド ユーザーのフィードバックの使用 | 提案したソリューションに対するユーザーのフィードバックを洗い出す。 | 多数決ツール<br>ツールを利用した、ソリューションについてのコメント<br>バグ レポート、フィードバック | Personify<br>Design<br>TeamLook |

Visual Studio 2005 Team System と Visual Studio Team System 2008 で提供されるトレーサビリティとアラート機能は、作業の進行に合わせてチームの共同作業を支援します。チームがタスクを更新し、タスクが要件にリンクされたときには、要件についてのクエリやレポートによって、これらのタスクの進捗状況を確認できます。クエリまたはレポートでは、どの要件が確定したかを確認できます。

Team Foundation Server のコンポーネントである Team Foundation ビルドでは、ビルド番号、作業項目の変更セット、関連付けられている作業項目の情報



---

が記載されたビルド レポートが作成されます。このようなレポートを使用すると、どのビルドにどのタスク、バグ、または要件がチェックインされているかを容易に確認できます。そのため、テスターやビジネス アナリストが適切なビルドを選択できるようになり、処理された要件、タスク、バグを把握できます。また、ビルドについてのアラートを設定して、ビルドがリリースされたらすぐに、テスターがテストを開始できるようにすることもできます。

このプロセスが実装されると、要件をチェックイン後すぐにテストし、直ちに修正することが可能になります。テスターの生産性が向上し、正しいテストを実行でき、古い要件や不適切なビルドをテストすることがなくなります。全期間を通じて、要件の進捗レポートと品質レポートを生成して、エンド ユーザーに伝達できます。このような最新データが揃っていれば、ビジネス アナリストは開発チームの現在の作業状況の把握に費やす時間を短縮し、ユーザーとの連絡、要件の検証と更新、要件の変更に関する共同作業により多くの時間をかけられるようになります。

---

## TEAM SYSTEM の 次期リリース (コード名 ROSARIO) の要件関連機能

### 要件の洗練

Visual Studio Team System コード名 "Rosario" では、要件の洗練プロセスが強化される予定です。たとえば、Team Foundation Server クライアントから子要件やタスクを分解する手段を提供して、作業分解プロセスが強化される予定です。また、作業項目の階層が作成しやすくなり、容易に依存関係を示せるようになります。プロセス テンプレートでは、採用した方法 (親子、先行タスクと後続タスク、依存関係など) で利用するリレーションシップ リンクの種類を定義し、必要なリンクや作業項目をまとめるフィールドなどのルールを指定できるようになります。これにより、非常に柔軟に作業のリレーションシップを管理したり、作業がどのように関連するかを記述したりすることが可能になります。

別の作業リレーションシップが作成されると、クエリを特定の種類のリンクに対して実行して次のような情報を確認できます。

- 対応する作業のない要件はどれか
- このタスクが対応する要件はどれか
- 要件から派生していないタスクはどれか

Team System レポート ウェアハウスのレポートでも、同じ情報を生成できます。このウェアハウスでは、ある期間における傾向を見ることもでき、次のような疑問を解消できます。

- 過去 1 ヶ月の間に、この要件の範囲はどの程度変更されたか
- この要件についてはどの程度のペースで作業を進めているか

Office Excel と Office Project で、要件 (および他の作業項目階層) を完全に編集できるようになる予定です。Team Foundation Server に保存されている親子のリンクが、Office Excel または Office Project のツリー階層内の適切な場所に表示されるようになります。Microsoft Office ツールで行った変更は、データが発行されると Team Foundation Server に保存されます。Office Project では、先行タスクと後続タスクなど、他のリンク リレーションシップの編集や更新も可能で、Team Foundation Server と Microsoft Office の両方に同じデータが反映されます。作業リレーションシップが変更されると、整合性が保たれた形でシステム全体が最新の状態に更新され、チーム メンバが直ちに各自の作業コンテキストと依存関係を認識します。

Team Foundation Server コード名 Rosario では、Office Project Server と統合して、複数のプロジェクトで共有しているリソースを管理することも可能になります。Office Project Server は、個人の予定表、休暇、利用可能時間情報が収集されるエンタープライズ リソース共有元を提供しています。人材が複数の

---

プロジェクトに割り当てられるときも、全プロジェクトにおける完全な稼働状況を把握できます。チーム メンバ間で負荷を分散でき、すべてのプロジェクトに対するメンバの貢献を考慮できるようになりました。

また、手動テストの作成と実行を容易にするシステムを作成する予定です。テストと要件をごく簡単にリンクし、手動テストも自動テストも含めて、要件の品質を示すレポートの実行セットを生成できるようにします。

これらのすべての機能強化により、要件の洗練プロセスは、ユーザーによる作業を多く必要としない、適切な作業リレーションシップが自然に設定されるスムーズなプロセスになります。スタッフは、生産性が向上するツールで作業を実行できます。マネージャとビジネス アナリストは、要件に関する作業の進捗状況と品質の状況を容易に確認できます。また、システム全体でトレーサビリティ機能を使用できるので、作業のコンテキストと変更の両方を把握できます。

### 要件の実装

Visual Studio 2005 Team System と Visual Studio Team System 2008 があれば、チーム メンバは作業が関連していることは理解していますが、どのように関連しているかは理解していない可能性があります。Visual Studio Team System コード名 Rosario では、要件とシステム内の他の作業間で意味のあるリンク リレーションシップを格段に容易に作成できるカスタマイズ可能なリンクを提供する予定です。チーム メンバは、特定のスプレッドシートやプロジェクト計画を参照しなくても、作業の依存関係や作業が要件からどのように派生しているかについて、より詳しく理解できます。たとえば、要件を構成するタスクや、依存関係のある他の要件を確認できるようになります。作業の相互関係を示すレポートやビューを提供し、各自の作業のコンテキストや変更の影響をより詳しく理解できるようになる予定です。

また、リンクについてのルールも提供し、作業をまとめたり、依存関係の受け入れもできるようになる予定です。これらのルールにより、正しいリンク リレーションシップが設定され、リンクの保守を最小限に抑えられます。ルールは、カスタマイズや拡張したりできるようにし、より容易にプロセスをツールに組み込めるようにします。

さらに、サブスクリプションと通知システムも強化し、ユーザーが各自にとって意味のあるアラートをより簡単にサブスクライブでき、不要な "作業項目の変更" アラートを受信しなくて済むようにします。

---

その結果、Visual Studio Team System コード名 Rosario では、アプリケーション ライフ サイクルの全期間を通じて、容易にプロセスに従い、作業リレーションシップを保守できるようになります。ビジネス アナリストとエンド ユーザーは、どちらも最新で開発チームの作業が正確に反映されているレポートによって、開発チームの作業をこれまでより詳しく理解できるようになります。また、チーム メンバは、要件の変更をすぐに認識し、より簡単に変更による影響を確認できます。

### **要件の検証**

Visual Studio Team System コード名 Rosario では、手動テストのトラックと実行、およびテストと要件とのリンクが容易になります。ビジネス アナリストには、ビルドに対して手動の受け入れテストを実行し、結果をトラックできるツールが提供されます。結果は Team System レポート ウェアハウスで参照することが可能で、すべてのテスト ケースについての完全な要件品質レポートを作成できるようになる予定です。

---

## 要件関連パートナー

### Borland CaliberRM

Borland® CaliberRM™ は、コラボレーション、影響の分析、コミュニケーションを支援するエンタープライズ ソフトウェア要件管理ツールです。ソフトウェア チームは、このツールを使用すると、非常に正確かつ予測可能な形で主要なプロジェクト マイルストーンを守ることができます。

CaliberRM は、Team System とシームレスに統合し、要件の保存と表示に階層型のツリー ビューを使用して要件管理を支援する豊富な機能を備えた要件管理環境を提供します。

Team System と CaliberRM を併用すると、要件が更新されたことが確実に当該チーム メンバに伝達され、リンクされているコード、テスト、および開発資産が変更に従って更新されるようにすることで、より品質の高い製品をより短い時間で提供できるようになります。

[会社 Web サイト](#) | [CaliberRM の概要](#) | [デモ/チュートリアル](#) (英語) | [Visual Studio Team System との連携使用についてのホワイト ペーパー](#) (英語)

### Compuware Optimal Trace

Compuware Optimal Trace™ は、視覚的なストーリーボードとトレース可能なリレーションシップ機能を備えており、プロジェクト ライフ サイクルの全期間を通じて、ユーザーの観点からビジネス ニーズの観点に至るまで、ソフトウェア要件をキャプチャします。

Optimal Trace Enterprise Edition は Team System と統合可能で、Team System ユーザーは Optimal Trace の機能を利用して、正確で完全なビジネス要件とシステム要件を策定できます。この結果、Team System のすべてのデザイン アクティビティと QA アクティビティを要件までさかのぼってトレースしたり、構想から開発に至るまでプロジェクト ライフ サイクルの最初から最後まで完全なトレーサビリティが提供されます。

Optimal Trace との統合により、Team Foundation Server 内の Team System "作業項目" の自動設定が可能になります。手動での入力を不要にし、完全で正確な要件を基盤としてプロジェクトのアクティビティが作成されるようにします。また、この統合により、Optimal Trace で実行された要件の変更を Team Foundation Server プロジェクトに反映できます。この反映処理により、作業項目に関連付けられている他のデータの整合性 (作業見積もりなど) が維持され、ライフ サイクル中の変更管理が全体的に容易になります。

[会社 Web サイト](#) | [OptimalTrace の概要](#) | [デモ/チュートリアル](#) (英語) | [Visual Studio Team System との連携使用についての Web キャスト](#) (英語)

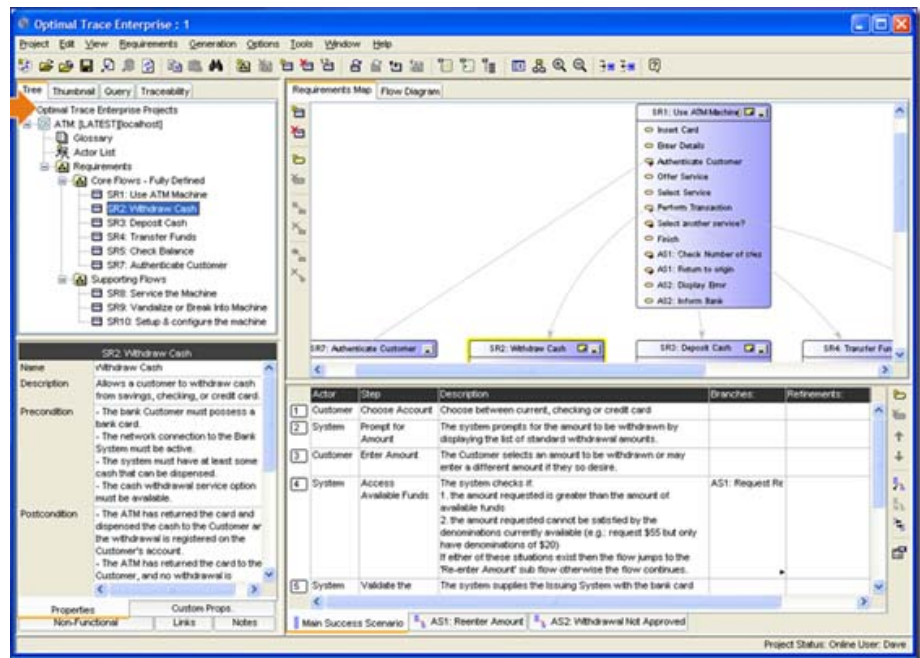


図 5. Compuware Optimal Trace Enterprise Edition の画面例

## Mindjet Requirements Manager

Mindjet Requirements Manager for Team System では、マインド マップ図を使用して、開発者が要件をすばやく簡単に収集する方法を提供します。要件を収集したら、Team System で作業項目を作成し、プロジェクトの進捗状況をトラッキングできます。この処理は、要件の洗い出しと、会議中で臨時にすばやく要件を収集できるように最適化されています。Mindjet Requirements Manager は Mindjet MindManager のビジュアル インターフェイスを基盤としたもので、要件をキャプチャ、整理、管理でき、Team System との双方向のリンクにより、作業項目を簡単にトラッキングできます。

[会社 Web サイト](#) (英語) | [Requirements Manager の概要](#) (英語) | [デモ/チュートリアル](#) (英語)

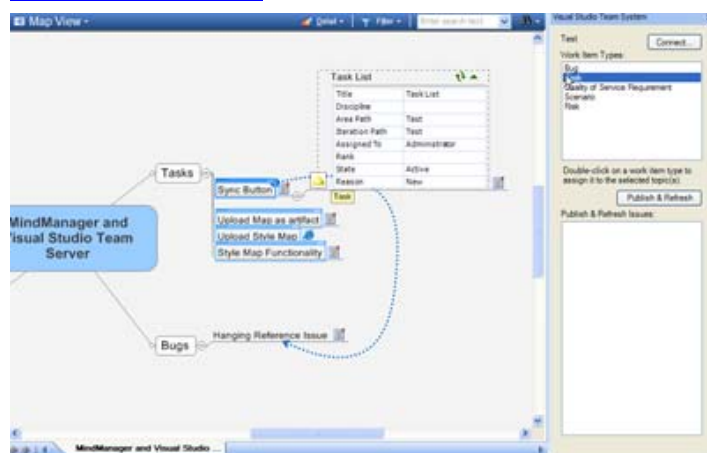


図 6. Mindjet MindManager のビジュアル インターフェイスの例

## Personify Design TeamLook と TeamSpec

Personify Design TeamSpec™ は、Office Word を使用して策定した要件を Team Foundation Server に関連付ける手段を提供します。TeamSpec を使用すると、Team Foundation Server プロジェクトの成果物 (変更要求、バグ、シナリオ、タスクなどの作業項目) から構造化された Office Word 文書を作成したり、既存の Office Word 文書からコンテンツを Team Foundation Server にプッシュしたりできます。

Personify では TeamLook™ も提供しています。これは、ソフトウェア開発プロジェクトの関係者全員が Office Outlook から Visual Studio Team System 2005 プロジェクトを介してコミュニケーションや共同作業を行うことができるようにする便利なソフトウェア プロジェクト ダッシュボードです。

[会社 Web サイト](#) (英語) | [TeamSpec の概要](#) (英語) | [デモ/チュートリアル](#) (英語)

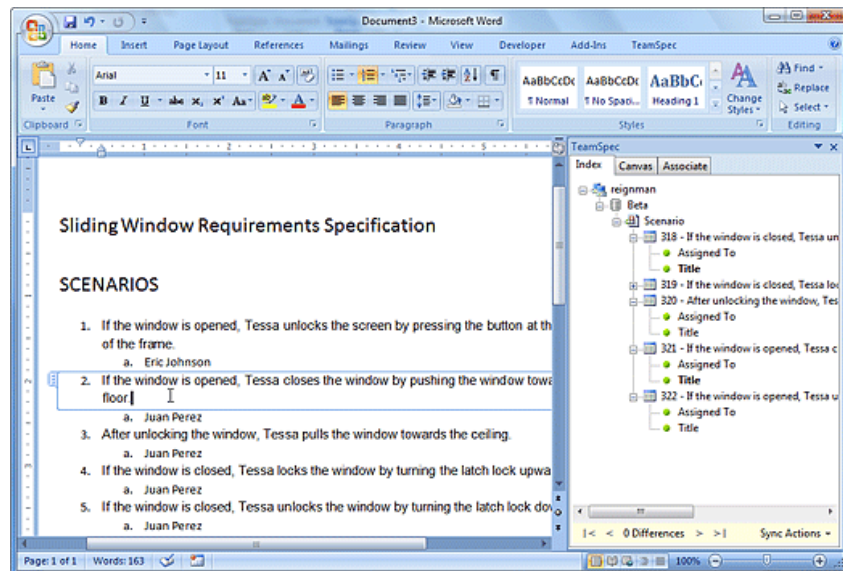


図 7. Personify Design TeamSpec と Office Word の画面例

## Ravenflow RAVEN

Ravenflow は、ビジネス アナリストの要件の洗い出し、分析、指定、検証を支援する RAVEN™ と呼ばれるソリューションを提供しています。RAVEN は構文解析ソフトウェアで、平易なビジネス英語テキストを解析し、自動的にアクティビティ図と職責図を生成します。すぐにグラフィカルなフィードバックが得られるので、要件の誤りを検出したり、アプリケーション フローを把握したりできます。

Ravenflow では Team System との統合により、Ravenflow でキャプチャされた要件を Team Foundation Server の作業項目として発行できます。



[会社 Web サイト \(英語\)](#) | [RAVEN の概要 \(英語\)](#) | [デモ/チュートリアル \(英語\)](#)

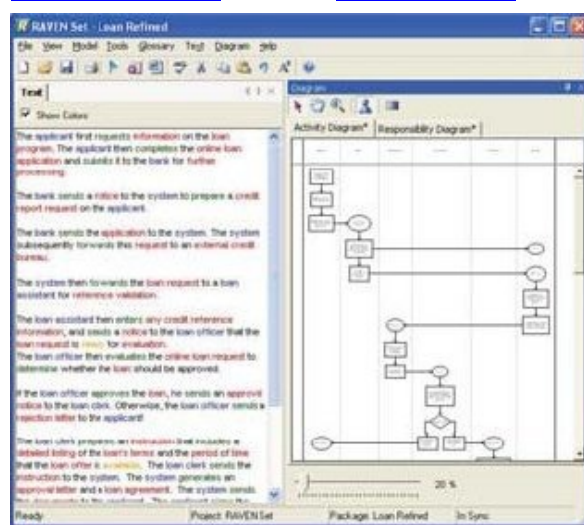


図 8. 図が表示されている RAVEN の画面例

## stpsoft stpBA と Quew

Visual Studio Team System 2005 対応の stpsoft stpBA ストーリーボードを使用すると、ビジネス アナリストやアナリスト開発者が、Microsoft Visio と互換性のあるグラフィックを使用して、GUI からチーム プロジェクトの要件とシナリオをキャプチャ、改良、および検証できます。このツールは Team System プロセス テンプレートとシームレスに統合し、画面フロー図、HTML ストーリーボード、UI 仕様、機能仕様、Team System 作業項目、テスト スクリプトを生成します。stpsoft は、[stpsoft Quew](#) という名前の軽量のドキュメント ベースの要件ツールも提供しています。

[会社 Web サイト \(英語\)](#) | [Visual Studio Team System 用 stpBA の概要 \(英語\)](#) | [デモ/チュートリアル \(英語\)](#)

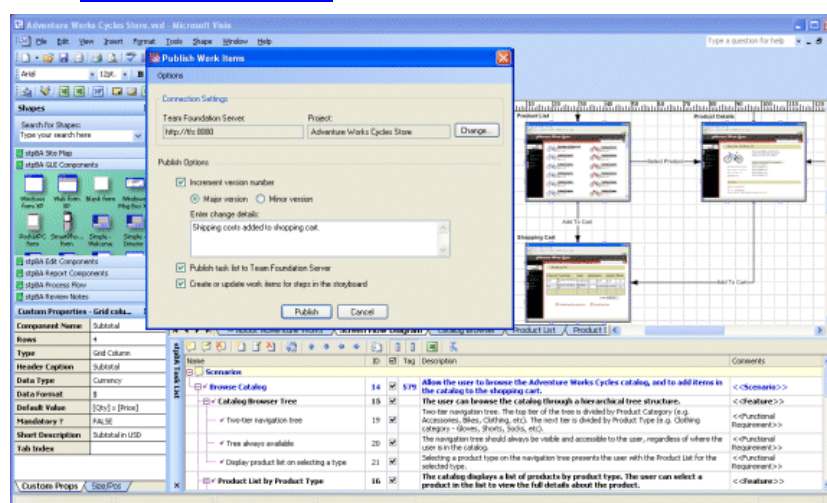


図 9. stpBA の [Publish Work Items] ダイアログ ボックスの画面例



---

## Techno Solutions Visual Use Case と TopTeam Analyst

Techno Solutions では一連の ALM ツールを販売しています。そのうち、2 つのツールが要件開発に使用できます。どちらも Team Foundation Server と統合します。

Visual Use Case™ は、ユース ケースのイベント フローの作成とユース ケース図の描画に使用するツールです。TopTeam Analyst™ は、要件を文書化、優先順位付け、および管理するツールを提供する要件管理ツールです。このツールは階層型の要件構造を使用し、リッチ テキスト編集機能と、トレーサビリティ ダイアグラム ツールおよびエディタを提供しています。

[会社 Web サイト](#) (英語) | [Visual Use Case の概要](#) (英語) | [TopTeam Analyst の概要](#) (英語)

## Holocentric Modeler

以下は、Holocentric Modeler 5.2 のパンフレットからの引用です。Holocentric Modeler 5.2 を使用すると、ビジネス プロセスの非効率な箇所を特定し、より優れたプラクティスをデザインできます。プロセスに含まれる各段階の期間やラグ タイム、頻度と量、月給とスキル レベルなどのデータをすべて含む、現実的なシナリオをキャプチャできます。表現も重要です。別のダイアグラム記号を使用して、スイムレーンやフリー フォームなどユーザーに合わせて表現を調整し、より詳細なダイアグラム ビューで期間とラグ タイムを表示します。最適な結果が得られるように、プロセスを分析したり、変更をシミュレーションしたりすることができます。シナリオを強調表示すると、モデルの分析中のパスが表示されます。

[会社 Web サイト](#) (英語) | [Holocentric Modeler 5.2 のパンフレット](#) (英語) | [デモ](#) (英語)

## 付録 A: OFFICE EXCEL を使用した作業項目の分解

Title 列のインデントは、スプレッドシート内でしか維持されません。つまり、スプレッドシート内で示されている作業項目の親子のリレーションシップは、Team Foundation Server には保存されません。したがって、.xls ファイルは破棄しないでください。要件の作業の進捗による影響を確認するには、このスプレッドシートを定期的に更新する必要があります。

要件をタスクに関連付けるには、[リンクと添付ファイル] ダイアログ ボックスを表示して、手動で要件をタスクに関連付けます。これにより、同じ情報が Visual Studio Team System レポート ウェアハウスにもレポートされるようにすることができます。

| Project: Rosario   Server: http://vstfdevdivir:8080   Query: All Deliverables   List type: Flat |  |             |             |
|---|--|-------------|-------------|
| ID  | Title  | Column1     | Assigned to |
| 1357  | the system must accept at least one character per second | Requirement | Lori Lamkin |
| 1358  | input screen   | Task        | Lori Lamkin |
| 1359  | fix existing performance issues                          | Task        | Lori Lamkin |
| 1694  | the system must have a Web interface                     | Requirement | Lori Lamkin |
| 2145  | logon screen   | Task        | Lori Lamkin |
| 2395  | welcome screen   | Task        | Lori Lamkin |
| 2444  | order screen   | Task        | Lori Lamkin |
| 2456  | inventory browsing                                       | Task        | Lori Lamkin |
| 2556  | users should be able to place orders                     | Requirement | Lori Lamkin |
| 2767  | back-end database  | Task        | Lori Lamkin |
| 2926  | update inventory   | Task        | Lori Lamkin |

図 10. Office Excel を使用した作業分解の例

## 付録 B: 要件品質レポート

テスト結果は、作業項目フォームでテストをリンクするか、任意のテストの実行結果を任意の要件に関連付けることで、要件と関連付けることができます。この関連付けを作成したら、Visual Studio Team System キューブからテスト結果パースペクティブを取得します。これは、.rdl ファイルの作成時か、Office Excel にデータ ソースをバインドすることで処理できます。

図 11 は、Office Excel にバインドされたキューブと、要件およびテストの実行結果についてのレポートを示しています。

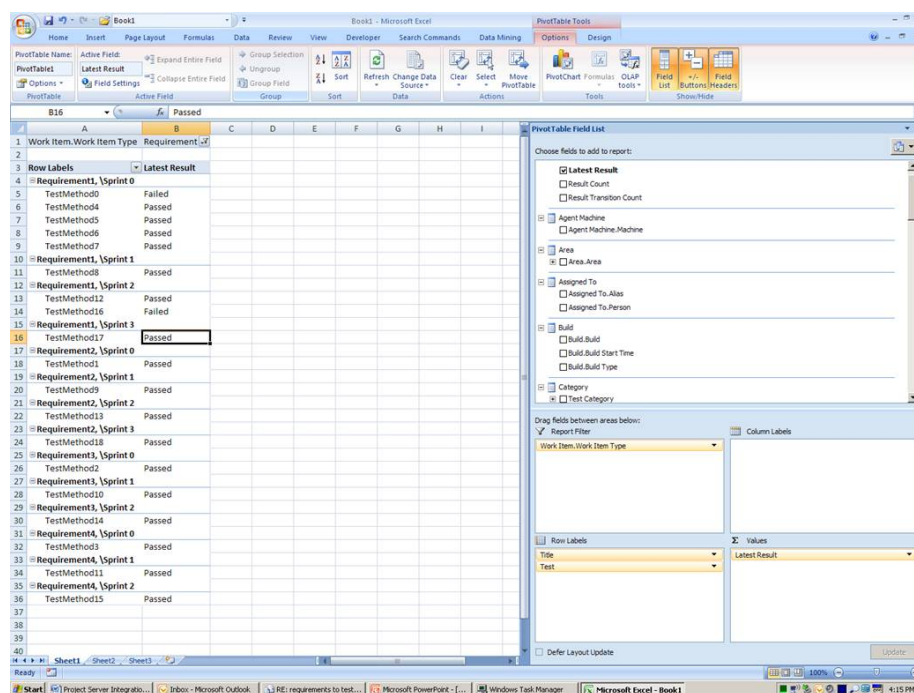


図 11. 要件品質レポートと Office Excel にバインドされたキューブの例

## 付録 C: 要件進捗レポート

図 12 は、サンプル データを使用して要件とタスクの関連付けを示したもので、要件ごとに何件のタスクが残っているかを示しています。このようなレポートを作成するには、Team Foundation Server の作業項目フォームで、要件とタスク作業項目が関連付けられているかどうかをユーザーが確認する必要があります。このデータは、"関連作業項目" キューブから取得されたもので、これには主属性と関連属性があります。

図 12 の例では、"作業項目の種類" が "要件" でフィルタされ、"関連作業項目の種類" が "タスク" でフィルタされて、他の作業項目の関連付けが表示されないようにしています。

|                                       |              |            |   |        |
|---------------------------------------|--------------|------------|---|--------|
| Work Item Type\Work Item Type         | Requirement  |            |   |        |
| Related Work Item Type\Work Item Type | Task         |            |   |        |
| Related Project.Project               | Formal_620_2 |            |   |        |
| Project.Project                       | Formal_620_2 |            |   |        |
| Related Count                         |              |            |   |        |
| Title                                 | Identifier   | Identifier | Title   | State  |
| Requirement1, \Iter 1-0               | 2            | 10         | Task 1, Scenario=Scenario_0_1, Iteration=\Iter 1-0  | Active |
|                                       |              | 17         | Task 8, Scenario=Scenario_0_1, Iteration=\Iter 1-0  | Active |
|                                       |              | 20         | Task 11, Scenario=Scenario_0_1, Iteration=\Iter 1-0 | Active |
|                                       |              | 24         | Task 15, Scenario=Scenario_0_1, Iteration=\Iter 1-0 | Active |
|                                       |              | 27         | Task 18, Scenario=Scenario_0_1, Iteration=\Iter 1-0 | Active |
|                                       |              | 43         | Task 34, Scenario=Scenario_0_1, Iteration=\Iter 1-0 | Active |
| Requirement2, \Iter 1-0               | 3            | 13         | Task 4, Scenario=Scenario_0_2, Iteration=\Iter 1-0  | Active |
|                                       |              | 18         | Task 9, Scenario=Scenario_0_2, Iteration=\Iter 1-0  | Active |
|                                       |              | 22         | Task 13, Scenario=Scenario_0_2, Iteration=\Iter 1-0 | Active |
|                                       |              | 23         | Task 14, Scenario=Scenario_0_2, Iteration=\Iter 1-0 | Active |
|                                       |              | 32         | Task 23, Scenario=Scenario_0_2, Iteration=\Iter 1-0 | Active |
| Requirement3, \Iter 1-0               | 4            | 28         | Task 19, Scenario=Scenario_0_3, Iteration=\Iter 1-0 | Active |
|                                       |              | 37         | Task 28, Scenario=Scenario_0_3, Iteration=\Iter 1-0 | Active |
|                                       |              | 44         | Task 35, Scenario=Scenario_0_3, Iteration=\Iter 1-0 | Active |
| Requirement4, \Iter 1-0               | 5            | 25         | Task 16, Scenario=Scenario_0_4, Iteration=\Iter 1-0 | Active |
|                                       |              | 42         | Task 33, Scenario=Scenario_0_4, Iteration=\Iter 1-0 | Active |
| Requirement5, \Iter 1-0               | 6            | 12         | Task 3, Scenario=Scenario_0_5, Iteration=\Iter 1-0  | Active |
|                                       |              | 14         | Task 5, Scenario=Scenario_0_5, Iteration=\Iter 1-0  | Active |
|                                       |              | 15         | Task 6, Scenario=Scenario_0_5, Iteration=\Iter 1-0  | Active |
|                                       |              | 16         | Task 7, Scenario=Scenario_0_5, Iteration=\Iter 1-0  | Active |
|                                       |              | 21         | Task 12, Scenario=Scenario_0_5, Iteration=\Iter 1-0 | Active |
|                                       |              | 31         | Task 22, Scenario=Scenario_0_5, Iteration=\Iter 1-0 | Active |
|                                       |              | 34         | Task 25, Scenario=Scenario_0_5, Iteration=\Iter 1-0 | Active |
|                                       |              | 36         | Task 27, Scenario=Scenario_0_5, Iteration=\Iter 1-0 | Active |
|                                       |              | 40         | Task 31, Scenario=Scenario_0_5, Iteration=\Iter 1-0 | Active |
| Requirement6, \Iter 1-0               | 7            | 11         | Task 2, Scenario=Scenario_0_6, Iteration=\Iter 1-0  | Active |
|                                       |              | 19         | Task 10, Scenario=Scenario_0_6, Iteration=\Iter 1-0 | Active |
|                                       |              | 35         | Task 26, Scenario=Scenario_0_6, Iteration=\Iter 1-0 | Active |
|                                       |              | 39         | Task 30, Scenario=Scenario_0_6, Iteration=\Iter 1-0 | Active |
| Requirement7, \Iter 1-0               | 8            | 26         | Task 17, Scenario=Scenario_0_7, Iteration=\Iter 1-0 | Active |
|                                       |              | 30         | Task 21, Scenario=Scenario_0_7, Iteration=\Iter 1-0 | Active |
|                                       |              | 33         | Task 24, Scenario=Scenario_0_7, Iteration=\Iter 1-0 | Active |
|                                       |              | 38         | Task 29, Scenario=Scenario_0_7, Iteration=\Iter 1-0 | Active |
|                                       |              | 41         | Task 32, Scenario=Scenario_0_7, Iteration=\Iter 1-0 | Active |
| Requirement8, \Iter 1-0               | 9            | 29         | Task 20, Scenario=Scenario_0_8, Iteration=\Iter 1-0 | Active |
| Grand Total                           |              |            |   |        |

図 12. 要件進捗レポートの例