

Microsoft®



Team Foundation Server 2010 では始める アジャイル開発

ホワイト ペーパー

株式会社テクノロジックアート 著

2011 年 10 月 11 日



著作権および、免責事項

このホワイト ペーパーに記載された内容は情報の提供のみを目的としており、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。 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、このホワイト ペーパーの内容に関する責任を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。

このホワイト ペーパーおよび、ソフトウェアを使用する場合は、適用されるすべての著作権関連の法律に従っていただくものとします。このホワイト ペーパーのいかなる部分も、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。ただし、これらは著作権法上のお客様の権利を制限するものではありません。

マイクロソフトは、このホワイト ペーパーに記載されている事項に関して、特許、申請中特許、商標、著作権、および他の知的財産権を所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がない限り、このホワイト ペーパーはこれらの特許、商標、著作権、または、その他の知的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。

別途記載されていない場合、このホワイト ペーパーで使用している会社、組織、製品、ドメイン名、電子メール アドレス、ロゴ、人物、場所、出来事などの名称は、架空のものです。実在する商品名、団体名、個人名などとは、一切関係ありません。

© 2011 Microsoft Corporation. All rights reserved.

Microsoft および Visual Studio は、マイクロソフト グループの商標です。 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

| | |
|--|------|
| 1. アジャイル開発とプロジェクト管理 | 1-1 |
| 1.1 ソフトウェア エンジニアリングの歴史とアジャイル開発 | 1-1 |
| 1.2 アジャイル開発とウォーターフォール型開発の比較 | 1-3 |
| 1.3 今、なぜアジャイル開発が必要か | 1-5 |
| 2. アジャイル開発手法「スクラム」の紹介 | 2-1 |
| 2.1 スクラムとは何か | 2-1 |
| 2.2 スクラムのプロセス フロー | 2-2 |
| 2.3 スクラムのプロジェクト体制 | 2-3 |
| 2.4 スクラムの作業項目管理 | 2-5 |
| 2.5 スプリント内での作業 | 2-7 |
| 2.6 スクラム チームの連携が成功の鍵 | 2-9 |
| 2.7 スクラムと Team Foundation Server | 2-11 |
| 3. Team Foundation Server ではじめるアジャイル開発 | 3-1 |
| 3.1 Team Foundation Server の概要 | 3-1 |
| 3.1.1 Team Foundation Server の機能 | 3-1 |
| 3.1.2 Team Foundation クライアント | 3-3 |
| 3.2 アジャイル開発の例題 | 3-5 |
| 3.2.1 プロジェクトの全体 | 3-5 |
| 3.2.2 スクラムにおける役割 | 3-6 |
| 3.3 Team Foundation Server によるアジャイル開発の実践 | 3-9 |
| 3.3.1 プロジェクトの開始 | 3-9 |
| 3.3.2 最初のスプリント計画会議 | 3-31 |
| 3.3.3 スプリント 1 の開始 | 3-45 |
| 3.3.4 バージョン管理とビルド管理 | 3-50 |
| 3.3.5 スプリント 1 の終了とスプリント 2 の準備 | 3-63 |
| 3.3.6 スプリント 3 からリリースへ | 3-68 |
| 4. まとめ | 4-80 |

1. アジャイル開発とプロジェクト管理

1.1 ソフトウェア エンジニアリングの歴史とアジャイル開発

ソフトウェア開発は、多くの知識を凝縮した極めて複雑な作業です。このために、ソフトウェア開発においては、納期の大幅な遅延、開発費の大幅な予算超過が起こりがちです。この困難を克服するために、ソフトウェア エンジニアリングにおいて、昔から多くの改善の努力がなされてきました。以下では、このようなソフトウェア エンジニアリングの改善の歴史を振り返ってみます。そして、その中で、アジャイル開発技術を位置付けていきます。

1950 年代、1960 年代

1950 年代、1960 年代のソフトウェア開発の創世記においては、コーディングをしてこれをデバッグする、という方法が基本でした。しかし、この方法は、ソフトウェアの開発規模が大きくなるにつれ破綻してしまいました。

1970 年代

そこで、1970 年代に登場したのが、ウォーターフォール（Waterfall）モデルに基づく開発方法（ウォーターフォール型開発）です。ウォーターフォール型開発では、コーディングをはじめの前に設計をきちんとして、どのようなソフトウェアを開発するかの方針や仕様をしっかりと定義して進めようとするものです。設計も、

- 1) 顧客の要求条件を正しく理解、定義するための要求仕様設計
- 2) ソフトウェアの骨格を定めるアーキテクチャ設計
- 3) ソフトウェアをモジュールに分解し、モジュールの作り方を定めるモジュール設計

など、目的に応じて技術が強化されてきました。ウォーターフォール型開発の導入による効果は劇的で、それまでよりもはるかに大規模なソフトウェアの開発を可能にしたのです。そしてまた、ソフトウェア設計技術の発展と普及をもたらしました。

そして、ウォーターフォール型開発は、現在でも多くのソフトウェア開発において依然として活用されています。

1990 年代

1990 年代に入ると世界の劇的な変化が起こり、ソフトウェア開発にも大きな影響を与えるようになります。情報技術面での変化は、PC の普及と、インターネットおよび WWW (World Wide Web) の普及です。まず、PC の普及はコンピューターパワーの劇的な増大をもたらしました。コンピューターの応用範囲は大きく広がり、それとともにソフトウェア開発への需要も大幅に拡大しました。一方、インターネットと WWW の普及は、同時に進展したグローバル化とともに、ビジネスの世界規模での分散化をもたらしました。世界規模でビジネスを同時進行させること、特に製造業では世界規模で水平分業を行うことが一般化し、また、ビジネスで生き残っていくための基本戦略の中に組み込まれるようになりました。ここで起こっていることは、世界規模で水平分業した複数のプロセスがネットワークで連携しながら、並行して実行されていくことです。そこで使用されるソフトウェアはさらに複雑度を増し、極めて短納期の開発を要求されます。

このような状況を反映して、1990 年代には反復型（イテレーティブ）開発プロセスが提唱されるようになりました。反復型開発では、要求されるソフトウェアの価値を段階的に繰り返し提供することにより、短期間で顧客の要求に応えようとしていました。MSF (Microsoft Solutions Framework), RUP (Rational Unified Process), 他がそれに該当し、基本的には「ミニ ウォーターフォール」になっていて、分割された小さなウォーターフォールのサイクルを繰り返すモデルになっています。

2000 年代

2000 年代に入ると、ソフトウェア開発のプロセスは、さらに「**アジャイル開発**」へと進展します。XP (eXtreme Programming), スクラム, FDD (Feature Driven Development) などのアジャイル開発プロセスでは、開発に参加するメンバー間のコミュニケーションを大切にします。具体的には、顧客と開発メンバー、開発メンバー相互のコミュニケーションを大切にします。そして、ドキュメンテーションを大幅に減らし、計画や設計にかかる時間を減らします。その代わりにテスト作業に力を入れて、「動くソフトウェア」を早期に提供して、顧客の要求に応えると共に、早期に顧客からのフィードバックを得るように努力します。

アジャイル開発の経験が積み重ねられると共に、次の二つの大きな流れが出てきます。

- 1) **アジャイル開発を大規模なプロジェクトに本格的に適用する**
- 2) **アジャイル開発の自動化ツールの発展**
- 3) **アジャイル開発の自動化ツールの連結、統合**

アジャイル開発は、当初は数名の一つのチームだけで開発する小規模プロジェクトに適用が限られていました。XP は、その典型でチームでのプログラム開発に焦点が絞られています。しかしながら、アジャイル開発へのニーズは、このように小さなプロジェクトに限られている訳ではありません。数十人、百人を超えるような規模のプロジェクトでもアジャイル開発を適用したいとのニーズは極めて高くなっていきました。そのようなプロジェクトでは、プログラミング作業をアジャイルに進めるだけではなく、既存のソフトウェアの再利用や、パッケージ、フレームワークの活用を前提としたシステム開発のアジャイル化への期待が強かったのです。スクラムはこのような、システム開発のアジャイル化への要請に応えるだけの柔軟性をもっていて、広く活用されるようになっていきました。

また、アジャイル開発は時間との戦いであり、一つのイテレーション サイクル（一般に 2 ～ 4 週間）の中に開発作業を納めなくてはなりません。ソースコードの変更履歴管理、ビルド、再テストなどの作業を、自動化支援ツールを活用して出来るだけ短時間の中に押し込める努力が行われました。アジャイル開発と自動化支援ツールの技術を連携させるようになっていったのです。

そして、これらの自動化支援ツールは個別に存在して働くだけではなく、相互に連携、統合されることによって更なる効果をあげていきます。例えば、変更履歴管理のツールと、変更に関連する機能の再テスト自動化ツールの連動等が挙げられます。また、自動化支援ツールをどのように使いこなしたとしても、全てが自動化される訳がなく、開発者たる人間系とツール群との連携も重要です。このためのマンマシン インターフェイスや、各種管理機能の充実が求められていきました。

このように、アジャイル開発の本格化とともに、これを支えるソフトウェア開発支援ツール、開発支援システムの充実への要請は非常に強くなってきていて、この要請に応えるための多くの努力が払われています。本書で紹介する Team Foundation Server (TFS) は、このような要請に応えるための総合的なソフトウェア開発支援システムです。小規模から大規模までのソフトウェアのアジャイル開発を、開発作業の支援、管理作業の支援の両面から支援し、関連する自動化支援ツール群を統合化して提供しています。

1.2 アジャイル開発とウォーターフォール型開発の比較

アジャイル開発とは何か、について記述した最も有名なものは、2001 年 2 月にアジャイル開発のキーパーソンが米国ユタ州に集まってまとめた次の宣言文（アジャイル ソフトウェア開発宣言）と言えましょう。

アジャイル ソフトウェア開発の宣言文

- ① プロセスやツールよりも個人との相互作用
- ② 包括的なドキュメントよりも動作するソフトウェア
- ③ 契約交渉よりもユーザーとの協調
- ④ 計画に従うよりも変化に対応する

以下、本節では、この宣言文も参考にしながら、アジャイル開発と従来型のウォーターフォール型開発を比較することによって、アジャイル開発の特徴を明らかにしていくことにしましょう。

繰返し型開発

ウォーターフォール型開発では、開発プロセスは一回だけで、工程は上流から下流へと Waterfall (滝) のように一方向に流れていきます。アジャイル開発では、開発のイテレーション (一般に 2 ～ 4 週間) を繰り返して開発を進めます。

ドキュメントよりも動作するソフトウェア

上記の短期間のイテレーションにより、動作するソフトウェアを早期に顧客に見て頂きます。そして、顧客からのフィードバックを頂き、それを次回以降のイテレーションに反映していきます。

契約交渉よりもユーザーとの協調

ウォーターフォール型開発では、プロジェクトを開始する前の契約がその後の全てを束縛しました。アジャイル開発では、プロジェクトの開始時点では、その後の全てを見通すことは出来ないと考えます。各イテレーションで優先的に開発すべき項目を顧客と相談の上で決め、また、イテレーションで得られた動作するソフトウェアを見て、その妥当性を評価して次回以降のイテレーションに反映していきます。このようにユーザーとの協調が仕様の決定の最も重要な要因となっています。

変化に柔軟に対応

ウォーターフォール型開発では計画、設計段階で決められたことを簡単に変えることはできません。しかし、開発すべきシステムの環境は、変化しがちです。

最近のソフトウェアでは、GUI (Graphical User Interface) の占める比率が極めて大きくなっています。このような GUI インターフェイスは、動かしてみないと見え方や操作性

が正しく理解できません。また、最近のソフトウェアの別の特徴として、全てをゼロから作るのではなく、既存のソフトウェアの一部を再利用、第三者パッケージの活用、オープン開発フレームワークの利用、等により開発の効率化、開発期間の短縮を図ります。このような複雑な組み合わせが、実際はどのように稼働するのか、動作させないと分かりません。動作させて、問題点を理解して、素早く改善していく、このような柔軟な対応がとても大切です。

時間は有限、資金も有限

ウォーターフォール型開発では計画、設計段階で決められたことにしたがって開発が進められますが、開発期間や開発費用が予定を大幅に超過することが少なくありません。アジャイル開発では、イテレーションを基本単位として、時間（1 イテレーションは一般に 2 ～ 4 週間）も資金（プロジェクト従事人員数）も固定されています。この固定された時間と資金の範囲で優先順位の高い開発項目から開発を進めていきます。したがって、アジャイル開発では、現実的な開発力がどれだけあるかが常に配慮されています。

個人間の相互作用

アジャイル開発では、個人間の相互作用が極めて大切にされます。顧客と開発チームメンバーとの関係、そして、開発チームメンバー間の関係が重要です。お互いを尊重して、協業・連携していくことが大切です。 プロセスやツールもとても大切ですが、それ以上に個人間の相互作用が大切と考えられています。

1.3 今、なぜアジャイル開発が必要か

前節では、ウォーターフォール型開発との対比でアジャイル開発の特徴を述べてきました。つぎに、なぜ、今、アジャイル開発の必要性が強く認識されるようになったのかを考えてみましょう。

グローバル化に伴うビジネスの加速

グローバル化の進展とともに、ビジネスは益々、加速化されています。それに伴い従来以上に多くのビジネス プロセスがオンライン化、IT 化されるようになってきました。そして、それを実現するためのソフトウェアに対しては短期間での開発が求められてきています。

IT 技術の進歩

一方、IT 技術の面からみると、技術的に大きな革新が続いています。ネットワークも、コンピューターも大幅な性能改善が続いています。最近では、スマートフォンが、電話と言うよりも携帯型コンピューターへと変身しています。また、ソフトウェア面での進歩も急速に進んでいます。多様なソフトウェア部品やフレームワーク、アプリケーションパッケージが揃ってきて、ソフトウェア資産が蓄積されてきています。また、各種の自動化ツールやそれらの相互連携機能等、ソフトウェア開発支援技術も急速に進歩してきています。

これらの IT 技術の進歩が、アジャイル開発の実行可能性を大きくしています。

アジャイル開発は必然か？

このように、需要面からの変化、それに応える実現技術の革新を背景として、今後、アジャイル開発技術は急速に普及していくものと考えられます。

2. アジャイル開発手法「スクラム」の紹介

2.1 スクラムとは何か

アジャイル開発の代表的な手法を二つ挙げるとすれば、**XP (eXtreme Programming)** とスクラムとなります。XP は、その名が表すようにプログラミング工程に焦点をあてて、プログラミング作業を効率よく進めるための具体的な技術を定めています。一方、スクラムは、ソフトウェア開発プロセス全体に対する管理の枠組みを定めています。開発プロセスの大きなフレームワーク（枠組み）を定めて、その詳細については各々のプロジェクトでカスタマイズが出来るようになっています。

既に 1 章でも述べてきましたが、近年のソフトウェア開発は、全てをゼロから作り上げるものではありません。既存のソフトウェア、第三者によるパッケージ、オープンな部品群やフレームワークを活用しながら開発することが多くなってきています。開発の具体的な方法は極めて多様なものになります。必ずしもプログラミングが主体となる訳ではなく、構造の再設計、既存ソフトウェアの自動再テスト等などの多様な作業が含まれます。スクラムは、これらの多様な作業を包含しつつ、開発プロセス全体を管理していきます。極めて柔軟性が高く、適用範囲が広い開発管理技法と言えます。

スクラムは、ソフトウェア開発のライフサイクル全体のプロセスを定義します。具体的には、

- A. スプリントと呼ばれる短期の開発サイクル
- B. スプリントの中での作業管理項目の枠組み
- C. プロジェクト参加者の役割（ロール）

を定めています。開発に入る前にどのような会議を開き、そこで誰がどのような権限を持ち、何を決めていくか、というようなことを定めています。

スクラムでは、どのタイミングでどのような方法で作業項目を調整、管理していくのかに重点が置かれています。各開発サイクルの中で、具体的にどのようなプラクティスを用いて開発を進めるかは自由になっています。アジャイル開発手法のプラクティスを用いるのが一般的ですが、自由度が高く、これを各プロジェクト流にモディファイしても、極端な話としてウォーターフォール流のプラクティスを加えて従来型の繰り返し型開発のプロセス設計をすることも可能です。

スクラムは極めてシンプルな仕組みしか用意していません。それらの仕組みはすべて、開発の過程で何が起きているのかを明確に見えるようにすることを目的として導入されて

います。ソフトウェアのアジャイル開発においては顧客の積極的な参加が欠かせない条件です。スクラムは、それを実現するための手段であるとも言えます。

提供している仕組みはシンプルでプロジェクト管理の枠組みに特化しているために、スクラムは開発プロセスのフレームワークととらえることも出来ます。これは、必要に応じてプラクティスを変更、追加して、自分達に使いやすいようにカスタマイズ可能であることを意味しています。そしてその際には、開発の透明性を高める、というスクラム本来の目的を忘れないようにすることが大切です。

2.2 スクラムのプロセス フロー

スクラムでは、ソフトウェア開発において必要とされるシンプルなプロセスを定義しています。図 2.1 は、そのようなプロセスの流れを示したものです。

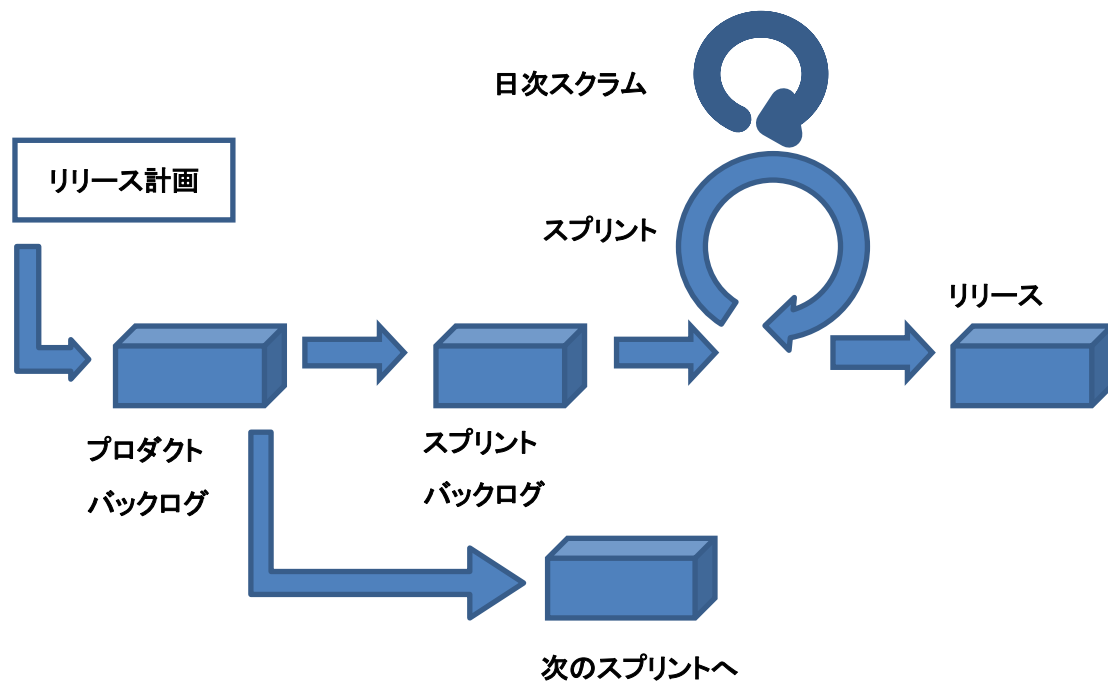


図 2.1 スクラムのプロセス フロー

スクラムの中心は、スプリントと呼ばれる 2 ～ 4 週間単位の開発期間の繰り返しから構成されます。このスプリントをどのように切り出し、繰り返し、回転していくかが、スクラムの要になります。

プロジェクトの開始に当たり、このプロジェクトの目標を定め、リリース計画を策定しま

す。そして、このプロジェクトとして開発しなくてはならない項目の集合としての「プロダクト バックログ」を作成します。次に、最初のスプリントをはじめるに当たって、スプリント計画会議を開催します。そこでは、顧客、開発者を含めた関係者が集まり、このスプリントで何を作るのかを確定します。具体的には、バックログの中から開発優先度の高い項目を選択して、最初のスプリントで開発すべき「スプリント バックログ」を選択します。「スプリント バックログ」に選ばれなかった項目は次回以降のスプリントで開発されることになります。

スプリント バックログが決まったら、スプリントを開始します。スプリントでは、毎日、日次スクラム ミーティングを開きます。そこでは、昨日までの作業の進捗状況を確認し、その日に取り組むべき作業を確認します。これを、毎日、そのスプリントが終了するまで繰り返します。

スクラムでは、日次スクラムとスプリントと言う二つの異なる長さのタイムボックスの中で作業を行います。そしてスプリントの終了時点で、スプリント レビュー会議を開催し、成果物をレビューします。

最後に、次のスプリント計画会議までの間に反省会ミーティングを行い、次回のスプリントに向けてチームで改良すべき点を話し合います。

スクラムでは、以上のスプリントの繰り返しを、プロジェクトの終了まで実施します。

2.3 スクラムのプロジェクト体制

スクラムのプロジェクト体制は、次の 3 つの役割を持つ人達から構成されます。

A. プロダクト オーナー

B. チーム

C. スクラム マスター

プロダクト オーナーは、作成するソフトウェアの所有者です。このソフトウェアが実現すべき要求仕様を管理する役割です。

チームも役割の一つです。このソフトウェアの要求仕様を実現する方法を考え、そして、このソフトウェアを実現する役割が与えられています。

そして、スクラム マスターは、プロジェクトのコーディネーターです。チームやプロダクト オーナーにスクラムのプラクティスを遵守させ、問題発生時にはこれを検知、対策し、プロジェクトが円滑に進むように調整する役割です。

以上の3つが、スクラムで定められている役割の全てです。スクラムの特徴は、このように役割が少なく絞られていてシンプルであることです。汎用的で本質的な役割に限定して定められています。

スクラムにおいては、開発プロジェクトにおける権利や責任は、基本的にプロダクト オーナーかチームのどちらかが担うことになります。スクラム マスターはプロダクト オーナーとチームが円滑にスクラムを遂行できるように調整、支援する、という一点のみににおいて責任を持ちます。

以上の 3 つの役割を束ねて「スクラム チーム」と呼びます。前述の「チーム」は開発の役割を持つもので、混同して使われがちですが、「スクラム チーム」とは別の概念です。

以下では、上述した 3 つの役割についてもう少し詳細に説明します。

プロダクト オーナー

スクラムにおいて重要なのがプロダクト オーナーです。プロダクト オーナーに誰になるのかは、開発形態により変わります。自社のプロダクト開発においては、プロダクトの仕様の決定権を持つマーケティング担当や開発部長などがなります。一方、受託開発においては、顧客（のプロジェクトの代表）がプロダクト オーナーになるでしょう。

アジャイル開発においては、顧客が積極的に開発に参加することが大切です。プロダクト オーナーは顧客の代表として、開発者と密接に連携をとり、仕様の明確化や詳細な意思決定を行うなど、プロジェクトで重要な役割を果たします。

スクラム マスター

スクラム マスターは、スクラムを円滑に進めるための調停者です。チームの代表としてプロダクト オーナーと話をするとともに、プロダクト オーナーの代わりにチームへ要望を伝えます。

スクラム マスターは専任で選ばれることがあります。また、チームのメンバーが兼務することもあります。複数の役割を兼務して、それを使い分けるのは一般的に困難なので、専任のスクラム マスターを配置する方が好ましいと言えます。

チーム

スクラムの特徴の一つとして、チーム リーダーやプロジェクト マネージャーと言った職責を定義していない点があります。基本的な考え方として、チーム内に階層はなく、自律的なメンバーが相互作用によりチームの意思を決定していくものと考えているからです。チームの自己組織化作用を考えると、リーダーの存在はむしろ阻害要因となる可能性

があります。

2.4 スクラムの作業項目管理

スクラムの特徴の一つに作業項目（タスク）の管理方法が挙げられます。非常にシンプルな仕組みですが、開発の透明度を高めるために必要な方法が整備されています。具体的には、バックログと呼ばれる「作業項目の一覧」を管理する一連の方法です。

バックログの種類

バックログには、次の3種類があります。

- A. プロダクト バックログ**
- B. スプリント バックログ**
- C. リリース バックログ**

プロダクト バックログには、プロダクトが満たすべき要求項目とその優先度、規模が書き込まれています。いわば、要求仕様のリストと捉えることが出来ます。通常の要求仕様リストと異なるのは、様々な粒度の要求項目が並んでいる点です。この要求項目のことを、「プロダクト バックログ項目」と呼びます。

この表にプロダクト バックログ項目を追加することは、誰にでも出来ます。プロダクトに機能を追加したい人ならば、プロダクト オーナーはもちろん、開発者、テスターなど、誰でも追加することが出来ます。そして、追加する人により、項目の粒度は異なるでしょう。

プロダクト バックログ項目の優先度と規模を見積もるのはプロダクト オーナーの仕事です。通常は、規模の見積もりはチームからの申告をそのまま使うことになるでしょう。しかし、見積もりはプロダクト オーナーの仕事であり、自らの責任において管理するとの姿勢が大切です。プロダクト オーナーは管理責任を負いますが、プロダクトにどのような機能を組み込むかの決定権を持っています。

次に、スプリント バックログは、スプリントを開始するに当たって作成するリストです。これには、スプリントで実行すべき作業項目の一覧が載っています。スプリント バックログの作業項目は、プロダクト バックログと異なり、開発者が実行可能な粒度にまで分解されています。そして、チームによる見積もり作業時間が入っています。このリストの作業項目を、「スプリント バックログ項目」と呼びます。

スプリント バックログ項目を作成し、見積もるのはチームの仕事です。

最後に、リリース バックログは、プロダクト バックログのうち、今回のリリースに含めるものを抽出したものです。したがって、リリースをどのように行うかによって、リリース バックログに含まれる項目が変わってきます。スプリント毎にリリースを行う場合には、スプリント バックログにブレイクダウンするもととなったプロダクト バックログが、リリース バックログになります。しかし、リリースを複数のスプリントをまたいで設定した場合には、これらのスプリントを包含した大きな要求項目の集合になります。

プロジェクト開始時点でのプロダクト バックログ作成

プロジェクトを開始する際には、最初に、チームの結成やプロダクト オーナーの任命などの体制作りを行います。それができあがったら、スプリントを開始するために、プロダクト バックログの洗い出しを行います。

プロジェクト開始時のプロダクト バックログ項目の洗い出しは、プロダクト オーナーとチームが共同して行う最初の作業です。スプリント期間の長さにも依存しますが、スプリントを 2, 3 回行なうに十分な程度の量を洗い出す必要があります。

洗い出しが完了したら、プロダクト オーナーは、プロダクト バックログにプロダクト バックログ項目をまとめます。その際に優先度付けがなされ、優先度が高いものについてはその詳細が分析されていなくてはなりません。分析の粒度は、チームがスプリント内で実施可能かどうかを十分に判断できる程度の細かさになります。

スプリント計画会議でのプロダクト バックログの選定

スプリントを開始します。開始に当たっては、毎回スプリント計画会議を開催します。これは、表 2.1 に示すように、前半後半、それぞれ 4 時間の計 8 時間で構成されるミーティングです。

スプリント計画会議の前半では、プロダクト オーナー、スクラム マスター、管理者などの関係者が集まり、どの要求項目をスプリントに盛り込むかを決定します。チームは、プロダクト オーナーに質問し要求項目を理解します。そして、スプリントに盛り込むことが可能かどうかについて、プロダクト オーナーに助言します。最終的に、どのプロダクト バックログ項目を盛り込むかどうかを決定するのはプロダクト オーナーです。スクラム マスターは、この会議が円滑に進むように調整役を担当します。スプリント バックログに盛り込むべき要求項目が決定した段階で、スプリント計画会議の前半は終了です。

大切なことは、プロダクト オーナーが自らプロダクト バックログの選択を行うこと、そして、チームとプロダクト オーナーは、意思決定を行うに足る情報を相互に提供し合うこと、です。

スプリント計画会議でのスプリント バックログ作成

スプリント計画会議の後半では、スクラム マスターとチームが要求項目を作業項目に分解し、スプリント バックログを作成する作業を行います。作業項目は通常 8 ～ 16 時間程度で実施可能な粒度に揃えられます。意味上の分類よりも、作業項目の規模で揃えるのが特徴です。

また、このタイミングで分析を進めるうちに、要求項目がスプリントに収まらないことが判明することがあります。この場合には、スクラム マスターがプロダクト オーナーを招へいし、再度、協議を重ねることで調整します。慣れないうちは、そのスプリントで実施すべき要求項目を減らさざるを得ないこともあるでしょう。

こうして、スプリント計画会議が終了すると、このスプリントで実施すべきスプリント バックログが完成します。

表 2.1 スプリント計画会議

| | 時間 | 参加者 | 成果物 | 内容 |
|----|-----------|---------------------------------------|----------------------------|---------------------------------------|
| 前半 | 30 分～4 時間 | 関係者 プロダクト オーナー スクラム マスター チーム | スプリントに含めるプロダクト バックログ項目のリスト | 今回のスプリントに含めるべきプロダクト バックログ項目を決定する |
| 後半 | 30 分～4 時間 | スクラム マスター チーム | スプリント バックログ | プロダクト バックログを分析してスプリント バックログ項目にブレイクダウン |

2.5 スプリント内での作業

日次スクラム ミーティング

スクラムでは、「日次スクラム ミーティング」と呼ばれるミーティングを毎日行います。そして、日次スクラム ミーティングから、次の日次スクラム ミーティングまでの一営業日を「日次スクラム」と呼びます。その一日を、全員一丸となって走り抜けるというラグビーのスクラムをモチーフとした比喻表現です。

そのミーティングには、チームとスクラム マスターが参加します。それ以外のメンバーが参加することもあるでしょうが、あくまでもオブザーバーとして参加します。発言は認

められていません。

ミーティングでは、チームの一人一人が次のことについて簡潔に報告します。

- ① 前回の日次スクラムから今までに行ったこと
- ② 今回の日次スクラムで行うこと
- ③ 困っていること

以上の三点のみです。

スクラム マスターは、チームが十分に自律性を発揮し、作業項目の消化を進めているかを確認し、スクラムのプロセスが円滑に進むように働きかけます。

各メンバーは、自発的に優先度の高い作業項目を消化していきます。もし誰も必要な作業項目を選ばなかった場合にはどうしたら良いでしょう。スクラム マスターが作業項目を割り当ててはいけません。スクラム マスターはチームに作業項目の必要性を説明して、チームに解決策を考えさせることが大切です。常に、自らが解決策を考えることで、チームに自律性が芽生えてきます。

この日次スクラム ミーティングは、チーム内の透明性を確保するために必要なプラクティスです。スクラム マスターはこの会議を通じて、チームの雰囲気やメンバーの表情を観察することで、形式的な報告には表れない情報を引き出して、開発プロセスの透明度を高めなくてはなりません。

スプリントの途中でのバックログの修正

作業項目への分割はスプリント計画会議の後半（4 時間）の時間内で実施しなくてはなりません。そのために、作業項目への分割が不十分になってしまうことがあります。これを解決するために、スプリントの途中で作業項目の再分割や調整を随時実施する必要があります。

チームは、スプリント内で実施すると約束した要求項目を達成する責任を負います。しかし、当初の見積もり違いや想定違いで、どうしてもそのスプリントに収まりきれないことがあります。そのような場合は、機能レベルを縮小することで要求項目を一回り小さくするように、プロダクト オーナーと調整します。この際に調整役となるのがスクラム マスターです。

また、機能レベルでの調整では済まない場合には、優先度の低い要求項目を次回スプリントに持ち越す形で調整することもあります。特にプロジェクトの最初のスプリントでは、

ソフトウェアの性格やチームの特質が不明確なので、作業効率を正確に見積もることは困難です。2 ～ 3 回のスプリントを終わるまでは、このようなスプリント バックログの調整が多発することがあります。

スプリント バックログの調整は、作業効率の見積もりの正確性が上がった後でも起こります。プロダクト オーナーに何が起きているのかを正確に伝えるためには、このようなバックログの改訂作業が欠かせません。

2.6 スクラム チームの連携が成功の鍵

以上に述べてきたように、スクラムは、極めてシンプルな仕組みで運用されています。プロダクト オーナーとチームの役割が密接に連携し、スクラム マスターの役割がそのバランスを調整することで、変化に適応していきます。

プロダクト オーナーの二つの役割

「スクラムにおいては、プロダクト オーナーが大変である」とよく言われます。そのプロダクト オーナーの役割を再確認しておきましょう。一言でいえば、「チームに対する顧客の代表として、要件の整理を行い、ソフトウェアの価値を決定する。」となります。これを分解すると次の二つに整理されます。

A. 顧客の代表としての側面

B. 要件を整理して、ソフトウェアに付加する価値を決定する側面

プロダクト オーナーの役割を理解する上での一つ目のポイントは、顧客の代表であるという点です。つまり、チームから見た場合は、プロダクト オーナーが顧客です。品質とは顧客のニーズへの適合度であるという考え方にに基づき、プロダクト オーナーの要件を可能な限り満たすようチームは振る舞います。

反対に、このソフトウェアのステークホルダー（利害関係者）からすると、プロダクト オーナーは自分たちの代表です。このソフトウェアを望ましい形にするためには、プロダクト オーナーに自分の要件を理解させて、このソフトウェアに反映してもらう必要があります。

一般的に、顧客と開発者との間のギャップが要件の取り違えを引き起こし、顧客の望むソフトウェアと違ったものを開発してしまう要因となります。スクラムでは、プロダクト オーナーという役割を設けることでこの問題に対処しています。

このように、プロダクト オーナーは、仕様決定におおきな影響力を持つため、多方面か

ら仕様に関する調整を依頼されることになります。ソフトウェア開発を成功させるために、重要な役割を担っています。

プロダクト オーナーの役割の二つ目のポイントは、要求項目の整理を行うという点です。プロダクト オーナーは、プロダクト バックログ項目としてまとめ、各々に優先度を付ける権利を持っています。また、各スプリントで、どの機能を実装するのかを決めることが出来ます。

それには、チームに要求を正確に伝達し、正確な見積もりが行えるように十分な情報を提供しなくてはなりません。また、機能を実装する際に必要な費用と要因について正確に理解し、チームが実現可能な範囲で機能を選択するという意思決定を行う必要があります。

この意思決定において求められる能力と才覚は非常に高いものになるので、スクラム マスター等の支援を必要とするのが通常です。また、プロダクト オーナー チームのような合議体に役割を与えることで、負担の軽減を図る試みもなされています。このような試みも興味深いものがありますが、基本的には、意思決定主体の明確化という意味で個人に役割を割り当てる方が好ましいと考えられます。

プロダクト オーナーは品質の要

通常、多数の関係者がいると仕様の取りまとめに苦労します。ソフトウェアへの要件がまとまらなかったり、矛盾する要件が出てきたりするからです。スクラムでは、プロダクト オーナーの役割を用意することで、プロジェクトをこのような混乱から守ります。プロダクト オーナーに任命された人は、ソフトウェアへの要件を整理し、何を実装するかを決定する権限を持ちます。また、チームと密に連携をとり、チームが正しくソフトウェアへの要件を理解し、実装するように説明する責任を持ちます。

このようにスクラムでは、開発者がソフトウェアの要件を適切に理解し実装できるような仕組みを設けているのです。しかし、この調整は「多くの利害関係という複雑性」そのものを解消したわけではありません。プロダクト オーナーという顧客側の代表を決めることで、チームと顧客の間にあった溝を、プロダクト オーナーと顧客の間に移動させたに過ぎません。すなわち、プロダクト オーナーが、顧客を含めたステークホルダーの要件を適切に吸収し、解釈することが出来なければ、利用者が本当に望んでいるソフトウェアを作ることは出来ません。ソフトウェアの品質の要となる役割がプロダクト オーナーなのです。

プロダクト オーナーが有効に機能しない場合

プロダクト オーナーが有効に機能しない場合にはどうしたら良いのでしょうか？ スク

ラムでは、スクラム マスターという調整役を用意しています。スクラム マスターは、プロダクト オーナーがスクラムのプラクティスを適切に実践できないと判断した場合は、プロダクト オーナーの作業を共に実施し、問題を解消する義務があります。時には、プロダクト オーナーの上層部に掛け合い、プロダクト オーナーの交代を調整することもあり得ます。

スクラム マスターはプロジェクト マネージャーにあらず

スクラム マスターの位置付けは、プロジェクトマネージャーと同じと誤解されがちです。立ち位置が非常に似ているからです。しかし、実態は別物です。

スクラム マスターは、あくまでも、スクラムが円滑に進むように支援する役割であり、主体的な責任は負いません。チームとプロダクト オーナーとが円滑にスクラムを進めることが出来るように、あらゆる局面で補助を行う役割です。

また、ここが大切なポイントなのですが、スクラムにおいてはチームの自律性を重要視しています。そのため、プロジェクト マネージャーやチーム リーダーといったチームを代表して管理するような役割を想定していません。実践に当たって、このような役割を追加することは可能ですが、通常はチームの自律性に悪影響を与えると考えられますので、プロジェクト マネージャーやチーム リーダーという役割を設定しないことが望ましいでしょう。

2.7 スクラムと Team Foundation Server

この 2 章でこれまでに説明してきたように、スクラムはアジャイル ソフトウェア開発プロセスの大きな枠組みを管理していきます。そこでは、開発プロジェクトに参加する人々の役割を明確にし、相互の協力関係を大切にします。

一方、3 章以降に説明する Team Foundation Server (TFS) は、その名の通りに、ソフトウェア開発チームが相互に協力しながらアジャイル開発を推進することを支援する統合的なシステムです。したがって、スクラムとの相性はとても良く、スクラムが方法論、技法、TFS が支援システムの役割を分担して、うまく連携出来るようになっています。具体的には、3 章を読みながら、この 2 章のスクラムの話と対応付けて頂ければと願っています。

3. Team Foundation Server では始めるアジャイル開発

アジャイル ソフトウェア開発宣言（1 章参照）には、「プロセスやツールよりも、個人とその相互作用」に価値を置くとあります。これは「プロセスやツール」を使わないということではなく、「プロセスやツール」が「個人とその相互作用」を助けるものであるべきということを言っています。優れた開発ツールは、アジャイル開発に開発のリズムと関係者同士の調和をもたらすものになるでしょう。

Microsoft Visual Studio Team Foundation Server（以下、TFS）は、作業項目をはじめ、ソース、ビルド、テスト、進捗状況など、開発に必要なあらゆる項目を一元的に管理することができ、開発プロジェクトの中核として機能します。スクラムなどの開発プロセスを実践する上では、強力なツールとなります。

本章では、アジャイル開発プロセスのひとつであるスクラムを採用したプロジェクトの例を示します。また、実際にプロジェクトを進める中で、TFS がどのようにプロセスを助けるかを見ていきます。

3.1 Team Foundation Server の概要

ここでは、TFS によるアジャイル開発の実践を見ていく前に、TFS に関する前提知識を説明します。TFS は、Microsoft Visual Studio 2010 におけるサーバーアプリケーションで、Visual Studio の対応領域を開発フェーズだけでなく、プロジェクトの計画フェーズから、テスト、リリースフェーズにまで広がっています。さらに、タスク管理や継続的インテグレーションのためのソース、ビルド管理など、アジャイル開発プロセスを支援する機能も充実しているのが特徴です。

3.1.1 Team Foundation Server の機能

TFS はバージョン管理システムである Microsoft Visual SourceSafe（以下、VSS）とは異なり、ソースコードだけでなく、要件表やテスト ケース、バグ レポートなど開発プロセスの各工程で分散して管理されていたリソースを一元化して、包括的に管理しています。TFS は、いわばプロジェクトの中核をなす統合データベースとして機能し、これによりあらゆる情報を追跡し、進捗や統計情報など最も適したビューを提供することを可能としています。

さらに、TFS には Visual Studio 専用のインターフェイスだけでなく、Eclipse やコマンドライン、Web ブラウザなど、ユーザーが普段使っている慣れたツールから接続できるよう設計されています。これを活用することにより、情報の共用を進め、チームのコラボ

レーションを向上させることができます。

TFS の主な機能と Visual Studio のエディションの対応は次のようになります。

表 3.1 TFS の主な機能

| | |
|--------------------|--|
| 作業項目の追跡 | 作業項目は TFS の作業管理の最小単位で、ユーザー ストーリー、タスク、テスト ケース、バグ、懸案事項のような項目があります。作業項目同士やコードの変更情報（変更セット）などとのリンクが設定でき、プロジェクトのあらゆる情報を追跡することができます。また、作業項目を検索するクエリを設定することで、目的に応じた作業項目一覧を抽出することができます。 |
| レポート | 作業項目のデータからプロジェクトの進捗や統計情報をグラフ化するなど、最適なビューを提供します。作業項目から直接生成されるので、資料用に新たにグラフを作成する必要がなく、リアルタイムにプロジェクトの状況を把握することができます。 |
| バージョン管理 | ソースコードとその変更の履歴をリポジトリというデータベースで管理します。チーム内でのソースコードの共有はもちろん、変更内容に作業項目を関連付けて、変更に至った経緯なども合わせて記録します。また、リリース後のメンテナンスバージョンなど、複数バージョンの系列を管理することもできます。 |
| ビルド管理 | ビルド作業を自動化し、日次ビルドやソースコードの変更を契機にして自動ビルドを実施する(継続的インテグレーション)などのスケジューリングができます。ビルドと同時にテストも実施すれば、バグの早期発見にもつながります。 |
| プロジェクト ポータル | プロジェクト ポータルにより、プロジェクト全体の状況がどうなっているかを、チーム メンバーやプロジェクト リーダーがいつでも参照できるようになります。 |
| テスト管理 | テスト計画からテスト ケースの構成、テストの実施、自動化、バグ管理まで、テストのライフサイクルを統合的に管理します。テスト対象となるユーザー ストーリーや、テスト対象のビルドバージョン、発生したバグとその修正情報など、テストに関するあらゆる情報が追跡可能となります。 |

表 3.2 TFS のエディション別の機能比較

| 機能 | Ultimate | Premium | Professional | Test Professional |
|-------------|----------|---------|--------------|-------------------|
| 作業項目の追跡 | ○ | ○ | ○ | ○※3 |
| レポート | ○ | ○ | ○ | ○※3 |
| バージョン管理 | ○ | ○ | ○ | |
| ビルド管理 | ○ | ○※1 | ○※2 | |
| プロジェクト ポータル | ○ | ○ | ○ | ○ |
| テスト管理 | ○ | | | ○ |

※1 デバッグ情報の収集など、一部不可

※2 単体テストの自動実行のみ

※3 テスト計画から実施、管理を目的した機能

3.1.2 Team Foundation クライアント

さまざまな Team Foundation クライアントから TFS を操作することができます。Visual Studio からの接続はもちろん、Excel や Project など Microsoft Office 製品、Eclipse、コマンドライン、Web ブラウザなどから接続が可能となっており、プロジェクトの環境や状況に合わせて適切なクライアントを選択することができるようになっています。

利用者は、普段使い慣れたツールから接続できるため、たとえば、プロジェクト マネージャーは Excel をそのまま利用すればよいでしょう。従来の Excel シートによる管理だとファイルに情報が蓄積され、データの更新や閲覧にタイムラグや欠落が発生する恐れがありました。これでは、本来のマネジメント、開発作業に注力すべきなのに、ツールの運用に注意を払わなければなりません。しかし、Excel の TFS 連携機能を使えば、チームで統一リポジトリに情報が蓄積されているため、Excel をビューとして最新の情報を閲覧し、意思決定をおこなうことができるようになります。一方、開発者は Excel を開くことなく Visual Studio や Eclipse などから自分のやるべきタスクを知り、更新を行えばいいのです。

Team Foundation クライアントには次のようなものがあります。

■ Visual Studio (チーム エクスプローラー)

TFS の標準クライアント。チーム プロジェクトの作成など、すべての操作が可能。

■ Test Manager

TFS のテスト計画と実行用クライアント。テストライフサイクル管理のために必要。

■ Web ブラウザ (Team Web Access)

Team Web Access により、ブラウザ上で作業項目の作成など、チーム エクスプローラーと同等の機能を提供。

■ Microsoft Office (Excel、Project)

Office ソフトウェアからのアクセスが可能。複数の作業項目の一括操作ができ、プロジェクト管理作業に便利。

■ Power Tools (Windows エクスプローラー拡張や付加機能の提供など)

Windows エクスプローラーや MSSCCI 対応の統合開発環境から接続可能。

■ コマンドライン (チーム エクスプローラー、Team Explorer Everywhere)

コマンドラインからのアクセスによって、Windows OS だけでなく、Windows 以外の OS (Linux、Mac OS X、HP-UX、Solaris、AIX など) から利用可能。

■ Eclipse および、Eclipse ベースの統合開発環境 (Team Explorer Everywhere)

Team Explorer Everywhere により、統合開発環境 Eclipse からのアクセスが可能。
Visual Studio と同様、チーム エクスプローラーを備える。

さらに TFS は、オープンな ALM プラットフォームというコンセプトを持っています。必要なツールを独自に作成するための SDK が提供されており、さまざまなオープンソース、商用のユーティリティが作られています。とくにオープンソース プロジェクトのためのコミュニティ CodePlex にて多くのユーティリティが開発、提供されています。

CodePlex

<http://www.codeplex.com/site/search?query=TFS>

3.2 アジャイル開発の例題

本章では、TFS を使ったスクラムによる開発を例題として取り上げます。ここでは、例題のプロジェクトの全体像を示し、次節の実践編を読み進めるための参考資料としての役割を果たします。

- プロジェクトの全体 (3.2.1)
- スクラムにおける役割 (3.2.2)

3.2.1 プロジェクトの全体

例題のプロジェクトは、ソフトウェア製品のためのコミュニティ サイトの構築プロジェクトです。コミュニティ サイトでは、製品ユーザー同士がフォーラムなどを使って情報交換をし、製品情報を閲覧します。コミュニティ サイトで、積極的にコミュニケーションが行われることによって、製品へのフィードバックが活発になり、より良い製品へつながることを期待します (表 3.3)。

プロジェクトは、コミュニティ事務局が取りまとめた、コミュニティ メンバーからの要望を元に、サイトを新規に開発します。開発プロセスはスクラムをベースに、XP (eXtreme Programing) のプラクティスなどを取り入れたものを想定します。

開発環境は TFS を中心に、バージョン管理やタスク管理、バグ管理を行い、Visual Studio や Microsoft Office、Eclipse、Web などのクライアントと連携して作業を進めます (表 3.4)。

表 3.3 プロジェクトの概要

| プロジェクト名 | 製品向けコミュニティ サイト構築 |
|----------|---|
| プロジェクト概要 | 製品のためのコミュニティ サイトを構築し、製品に対する情報交換の中で、製品に対するフィードバックを得る |
| 顧客 | コミュニティ事務局 |
| ユーザー | 一般閲覧者、製品ユーザー、コミュニティ メンバー、コミュニティ事務局 |
| 開発プロセス | スクラム、XP |

表 3.4 開発環境の概要

| | |
|-------------|--|
| 開発サーバー | Team Foundation Server 2010 |
| 開発クライアント | Visual Studio 2010 Ultimate Test Manager 2010 Excel 2010 Internet Explorer 8 Eclipse 3.5 (Team Explorer Everywhere 2010 SP1) |
| バージョン管理システム | Team Foundation Server 2010 のソース管理 |
| タスク管理システム | Team Foundation Server 2010 の作業項目 |
| バグ管理システム | Team Foundation Server 2010 の作業項目 |

3.2.2 スクラムにおける役割

例題のプロジェクトでは、スクラムにおける各役割を表 3.5 のように割り当て、読者である「あなた」がプロジェクトのスクラム マスターとなって、作業を疑似体験する記述となっています。

表 3.5 例題プロジェクトでの役割の割り当て

| 役割 | 担当 | 説明 |
|------------|---------------------------------------|--|
| プロダクト オーナー | コミュニティ事務局 担当者 | コミュニティを運営するコミュニティ事務局の一員。コミュニティ メンバーからの要求を取りまとめ、開発チームに開発を依頼する |
| スクラム マスター | あなた | 例題プロジェクトを成功させる責務を負う |
| スクラム チーム | 開発チーム ・ 設計者兼プログラマ ・ デザイナー ほか | コミュニティ メンバーからの要求をコミュニティ サイトとして実現させる |
| ステークホルダー | 関係者 ・ 一般サイト閲覧者 ・ 製品ユーザー | コミュニティ サイトに何らかの形で関わる |

| 役割 | 担当 | 説明 |
|----|---------------------|----|
| | ・ コミュニティ メンバー ほか | |

スクラムではプロジェクトの作業項目を管理するため、表 3.6 の成果物が作られます。ここで、製品バックログは 2 章で説明したプロダクト バックログのことです。TFS ではプロセス テンプレート MSF for Agile Software Development v5.0 においてプロダクト バックログを提供する作業項目クエリを標準で提供しており、ここからはこの名称を使用します。なお、プロセス テンプレートは TFS におけるプロジェクトの雛形で、プロジェクト作成時に指定します。MSF for Agile Software Development v5.0 はアジャイル開発や中小規模開発に適したテンプレートです。それぞれの成果物は表 3.7 のように TFS の成果物と対応付けられ、管理責任を持つ役割が決められます。

なお、製品バックログとリリース バックログの関係は図 3.1 のようになり、製品バックログが複数のリリース バックログに分割されます。基本的に製品バックログの一項目は、リリース バックログの一項目に対応します。製品バックログ項目とリリース バックログ項目はストーリー形式で表されるため、単にストーリーとも呼ばれます。同様に、スプリント バックログ タスクも単にタスクと呼ばれます。

表 3.6 例題プロジェクトに登場する作業成果物

| 成果物名 | 説明 |
|-----------------|---|
| 製品バックログ | 要件一覧を元に、開発すべき製品機能としてまとめたもの。リリース バックログの集合として扱われる |
| 製品バックログ項目 | ストーリー形式で表される製品バックログの項目。製品機能の項目として扱われる |
| リリース バックログ | 製品バックログをリリース単位に分割したもの |
| リリース バックログ項目 | リリース単位に割り当てられた製品バックログ項目 |
| スプリント バックログ | ストーリーを実現するために必要なタスクを定義したもの |
| スプリント バックログ タスク | スプリント バックログのひとつの項目 |

表 3.7 例題プロジェクトに登場する作業成果物の形式と管理者

| 成果物名 | 形式 | 管理者 |
|-----------------|--------------------------|-------------------------------------|
| 製品バックログ | TFS クエリ | プロダクト オーナー |
| 製品バックログ項目 | TFS 作業項目 (ユーザー ストーリー) | プロダクト オーナー |
| リリース バックログ | TFS クエリ | プロダクト オーナー スクラム チーム スクラム マスター |
| リリース バックログ項目 | TFS 作業項目 (ユーザー ストーリー) | プロダクト オーナー スクラム チーム スクラム マスター |
| スプリント バックログ | TFS クエリ | スクラム チーム |
| スプリント バックログ タスク | TFS 作業項目 (タスク) | スクラム チーム |

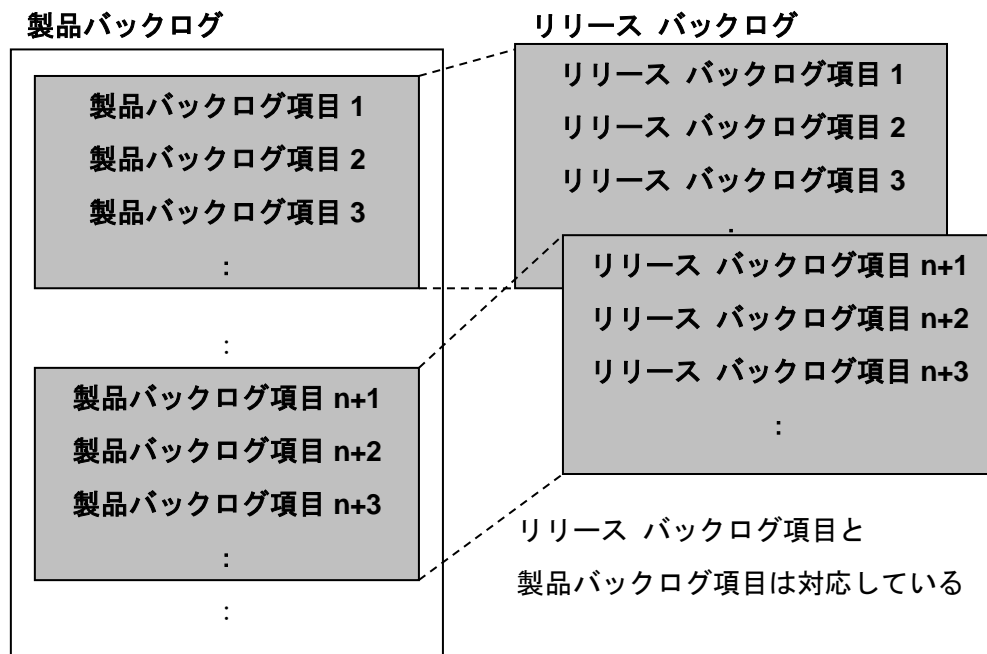


図 3.1 製品バックログとリリース バックログの関係

3.3 Team Foundation Server によるアジャイル開発の実践

それでは、例題のプロジェクトを元に開発プロセスを追ってみましょう。あなたは**スクラム マスター**となり、プロジェクトを進めます。同時に、スクラム マスターは TFS の管理者となります。

ストーリーは以下の流れで進みます。

- プロジェクトの開始 (3.3.1)
- 最初のスプリント計画会議 (3.3.2)
- スプリント 1 の開始 (3.3.3)
- バージョン管理とビルド管理 (3.3.4)
- スプリント 1 の終了とスプリント 2 の準備 (3.3.5)
- スプリント 3 からリリースへ (3.3.6)

3.3.1 プロジェクトの開始

■ 担当者の話を聞く

場面は顧客との新規案件の打ち合わせ。あるソフトウェア製品のコミュニティ サイト立ち上げに関して、コミュニティ事務局の担当者と会うことになり、プロジェクト管理者のあなたは、営業担当者とチームの技術リーダーとともにコミュニティ事務局の担当者の話を聞きました。

コミュニティ事務局の担当者の話は次のようなものです。

- 製品に関するコミュニティ サイトを立ち上げたい
- コミュニティ サイトでは、特に製品ユーザー同士のコミュニケーションを促進し、製品の次のバージョンに組み込むためのフィードバックを得たい
- まだ、製品を使用していないサイト閲覧者のために製品紹介も行いたい
- いくつかのコンテンツは、ログイン後にだけ利用することができる

話を聞いて、あなたはいくつかの機能をイメージしたかもしれませんが、幸いにもコミュニティ事務局の担当者はメンバーから集めた要件の一覧表を持っていました。要件一覧は次のようなものです。

表 3.8 要件一覧表

| No. | 要件 | 記載者 |
|-----|-------------------------------|-----------|
| ① | 製品の説明を記載したい | ○月×日△△社□□ |
| ② | 製品の英語資料を日本語訳にして公開したい | ○月×日△△社□□ |
| ③ | 製品の最新のベータ版をダウンロードして試したい | ○月×日△△社□□ |
| ④ | 製品の最新のリリース版をダウンロードできるようにしてほしい | ○月×日△△社□□ |
| ⑤ | 製品に関するニュースを掲載してほしい | ○月×日△△社□□ |
| ⑥ | 製品のイベント情報を公開してほしい | ○月×日△△社□□ |
| ⑦ | サイト上で、製品に関する使い方を質問したい | ○月×日△△社□□ |
| ⑧ | サイト上で、製品機能に関する議論をしたい | ○月×日△△社□□ |
| : | | |

あなたはこの担当者に**プロダクト オーナー**役をお願いしました。コミュニティ事務局の担当者は、コミュニティ メンバーの要件を取りまとめる立場にあり、過去にもいくつかのシステムを発注した経験があると聞いたからです。コミュニティ事務局の担当者は、システムの発注者が、開発チームと一緒にあって問題に取り組まなければ良いシステムができないことを知っていたので快く引き受けてくれました。

あなたはさっそくプロダクト オーナーである事務局の担当者に、先ほどの要件一覧に「優先度」と「規模感」という項目を追加し、表を**製品バックログ** (2 章ではプロダクト バックログ) という名前に変更して欲しいと依頼しました。

■ 製品バックログを作成

自社に戻ると、プロダクト オーナーから要件一覧に優先度と規模感を追加した一覧が送られてきました。最初の製品バックログは次のようなものです。

表 3.9 最初の製品バックログ

| No. | 要件 | 優先度 | 規模感 | 記載者 |
|-----|-------------------------|-----|-----|-----------|
| ① | 製品の説明を記載したい | 高 | 小 | ○月×日△△社□□ |
| ② | 製品の英語資料を日本語訳にして公開したい | 中 | 中 | ○月×日△△社□□ |
| ③ | 製品の最新のベータ版をダウンロードして試したい | 低 | 中 | ○月×日△△社□□ |

| No. | 要件 | 優先度 | 規模感 | 記載者 |
|-----|-------------------------------|-----|-----|-----------|
| ④ | 製品の最新のリリース版をダウンロードできるようにしてほしい | 中 | 中 | ○月×日△△社□□ |
| ⑤ | 製品に関するニュースを掲載してほしい | 低 | 中 | ○月×日△△社□□ |
| ⑥ | 製品のイベント情報を公開してほしい | 中 | 中 | ○月×日△△社□□ |
| ⑦ | サイト上で、製品に関する使い方を質問したい | 高 | 大 | ○月×日△△社□□ |
| ⑧ | サイト上で、製品機能に関する議論をしたい | 高 | 大 | ○月×日△△社□□ |
| : | | | | |

あなたは開発チームとともに製品バックログの内容を確認します。その結果、製品バックログにいくつか変更が必要になりました。

まず、製品バックログの各項目はユーザー ストーリーの形式で管理するので、規模感にしたがってストーリー ポイントを算出しました。ストーリー ポイントとは人月のような時間による単位でなく、いくつかの要件の規模感を基準に、相対的なポイントで見積りします。そして、各ストーリーの統合や分割を行い、コミュニティ サイトを運用する上で必要と考えられる機能を追加しました。開発チームによってストーリー化され、優先度順に並び替えると製品バックログは次のようになりました。

表 3.10 ストーリー化された製品バックログ

| No. | 要件 | 優先度 | ストーリー ポイント | 記載者 |
|-----|---|-----|---------------|-----------|
| ① | サイト閲覧者は、製品の説明を閲覧できる | 高 | 3pt | ○月×日△△社□□ |
| ⑦ | ログイン済閲覧者は、フォーラムで製品に関する情報交換ができる ・ 製品の使い方の質問 ・ 製品機能に関する議論など | 高 | 20pt | ○月×日△△社□□ |
| ② | ログイン済閲覧者は、製品の英語資料 | 高 | 5pt | ○月×日△△社□□ |

| No. | 要件 | 優先度 | ストーリー ポイント | 記載者 |
|-----|---|-----|---------------|-----------|
| | の日本語訳を閲覧できる | | | |
| ④ | ログイン済閲覧者は、製品のリリース版をダウンロードできる | 中 | 5pt | ○月×日△△社□□ |
| ⑥ | サイト閲覧者は、イベント情報を閲覧できる | 中 | 5pt | ○月×日△△社□□ |
| ③ | ログイン済閲覧者は、製品のベータ版をダウンロードできる | 中 | 3pt | ○月×日△△社□□ |
| ⑤ | サイト閲覧者は、ニュースを閲覧できる | 低 | 3pt | ○月×日△△社□□ |
| : | | | | |
| ⑨ | サイト閲覧者は、ドメイン名でサイトにアクセスできる (インフラ構築用ストーリー) | 低 | 8pt | |
| ⑩ | 未登録閲覧者は、自身のアカウントを登録することができる | 低 | 8pt | |
| ⑪ | 未ログイン閲覧者は、ID とパスワードを指定してログインできる | 低 | 5pt | |
| : | | | | |

■ 製品バックログを共有

あなたは技術リーダーとともにプロダクト オーナーであるコミュニティ事務局の担当者に会い、製品バックログの最新版について説明しました。ストーリーの形式とストーリーポイントはあまりなじみのないものですが、丁寧に説明することで理解してもらえました。

そして、最初のリリース時期を確認すると、あなたは営業担当や開発チームと連携しながら、リリース時期と製品バックログを元に開発チームの体制を整え、見積り金額を算出しました。後日、無事に月単位での稼働時間による契約が成立し、ほっと一安心です。プロダクト オーナーが成果物でなく、稼働時間による契約を受け入れたのは次のような理由かもしれません。

- 設計する前にすべての機能が分かりきったような、どこにでもあるサイトを作る気は

なかった

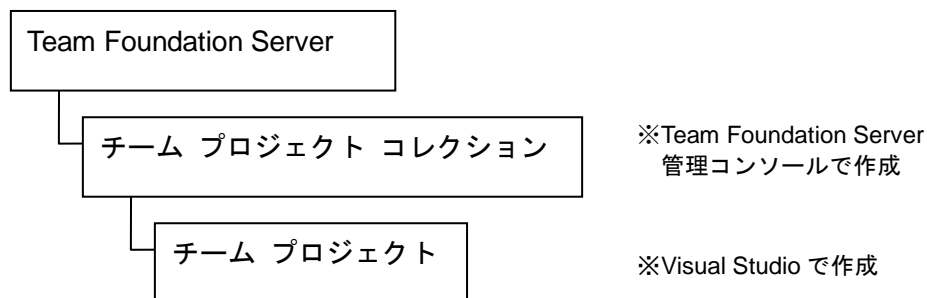
- 自分が未来のすべてを予測できる予言者ではないことを知っていた

あなたは、最初の**スプリント計画会議**の日程を調整し、プロジェクト開始の準備のため、TFS の設定を行います。ここでは TFS のインストールはすでに完了しているものとします。

■ [TFS 操作] チーム プロジェクトの作成

プロジェクトの立ち上げ時には、TFS にプロジェクトを作成し、メンバーの情報を登録します。

最初にチーム プロジェクトを作成します。チーム プロジェクトとは、チームの中心として計画の整理やソースコードの共有、ビルド、テストの計画や実行を行うための枠組みです。チーム プロジェクトは、以下のようにチーム プロジェクト コレクションでグループ化される階層構造となっていて、たとえば、組織の部門別にチーム プロジェクトをまとめることができます。



まずは、チーム プロジェクト コレクションを作成します。チーム プロジェクト コレクションは Team Foundation Server 管理コンソール（以下、TFS 管理コンソール）から作成します。

- ① スタートメニューの「すべてのプログラム」から「Microsoft Team Foundation Server 2010」をクリックし、「Team Foundation Server 管理コンソール」を実行します。
- ② 管理コンソールの「サーバー名」 - 「アプリケーション層」を展開し、「チーム プロジェクト コレクション」を選択し、「コレクションの作成」ボタンをクリックします（図 3.2）。
- ③ コレクション作成のウィザードが起動されるので、ウィザードにしたがってコレクション名の設定（図 3.3）、SQL Server のインスタンスの設定（図 3.4）、SharePoint サイトの設定（図 3.5）、SQL Server のレポートサービスの設定（図 3.6）を行い、最後に構

成を確認 (図 3.7) したら、コレクションの作成を実行して完了です。

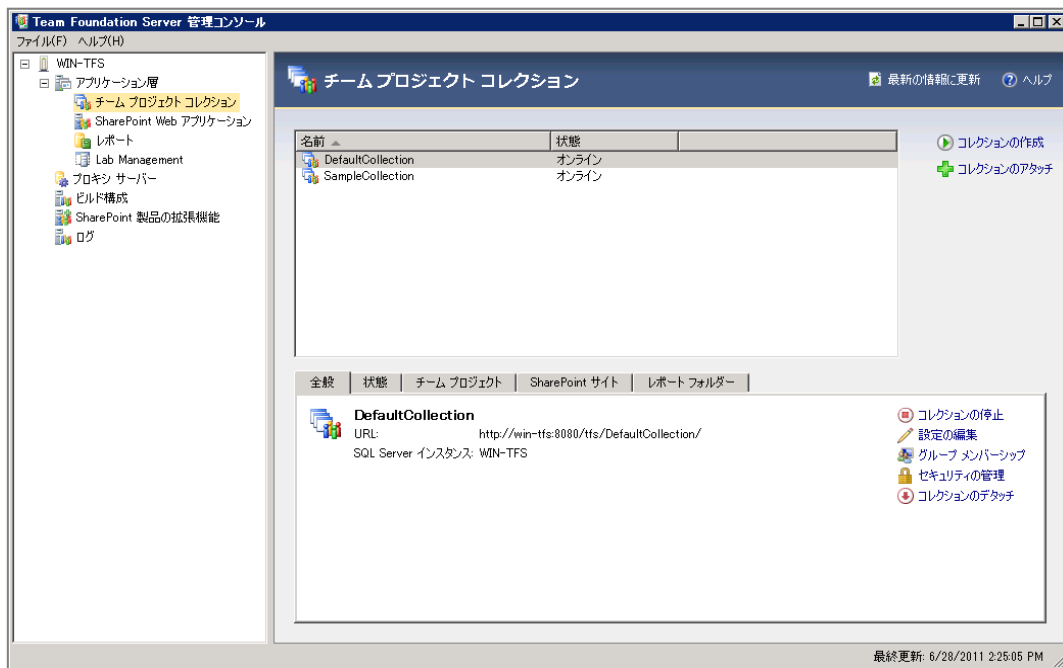


図 3.2 チーム コレクションの作成 (TFS 管理コンソール)

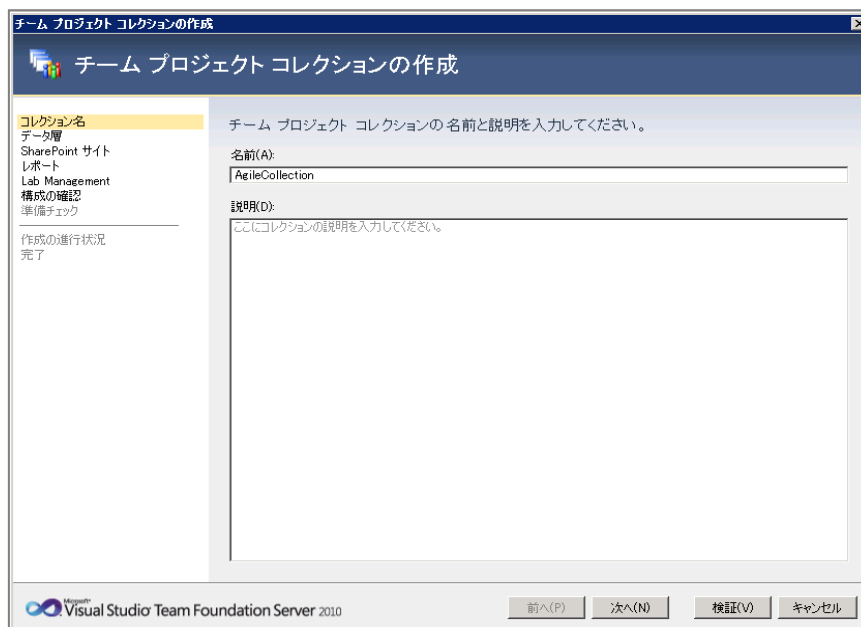


図 3.3 コレクション作成のウィザード (TFS 管理コンソール)

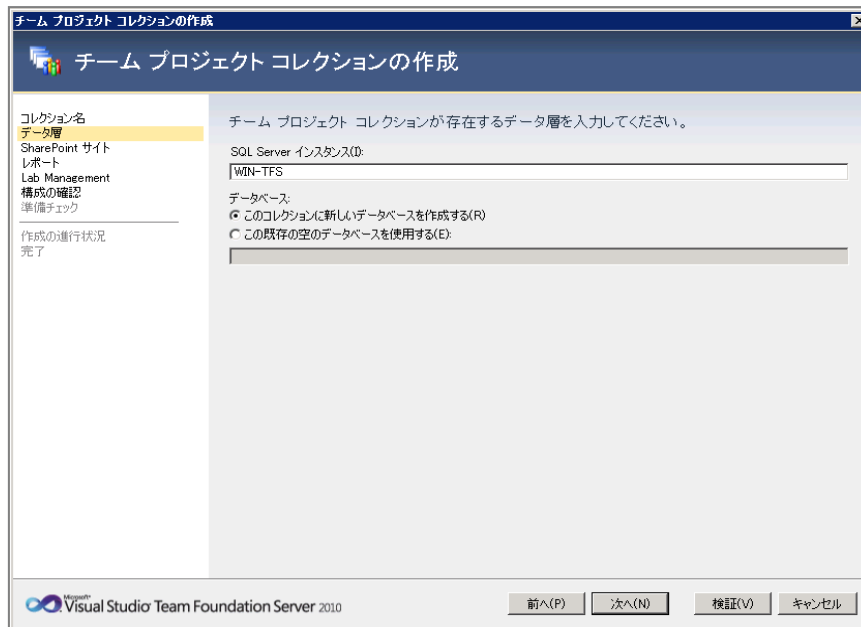


図 3.4 SQL インスタンスの設定 (TFS 管理コンソール)

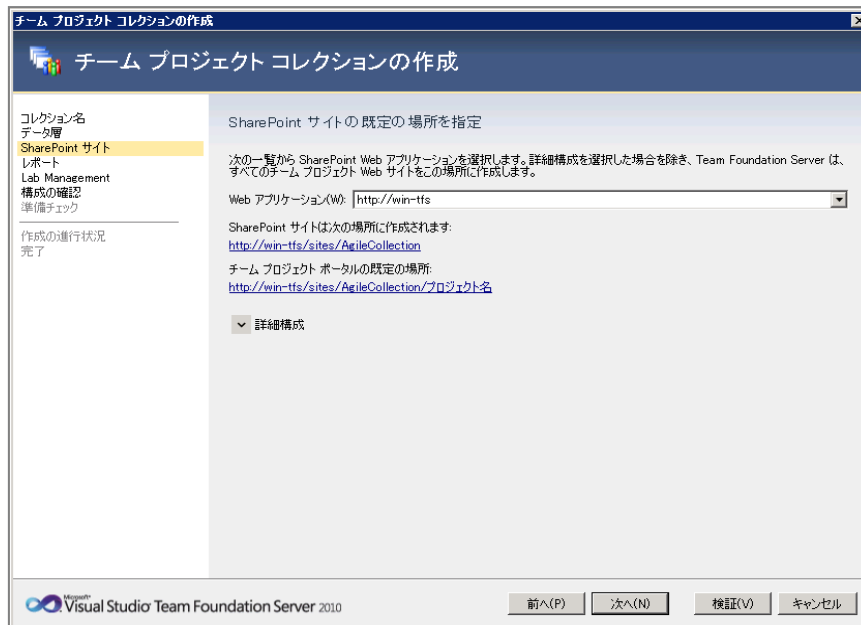


図 3.5 SharePoint サイトの設定 (TFS 管理コンソール)

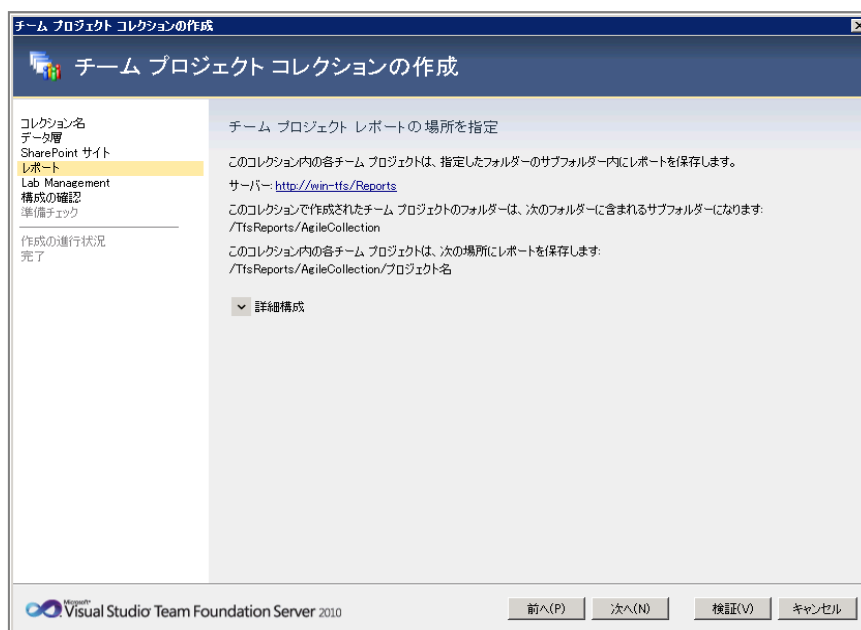


図 3.6 レポートサービスの設定 (TFS 管理コンソール)

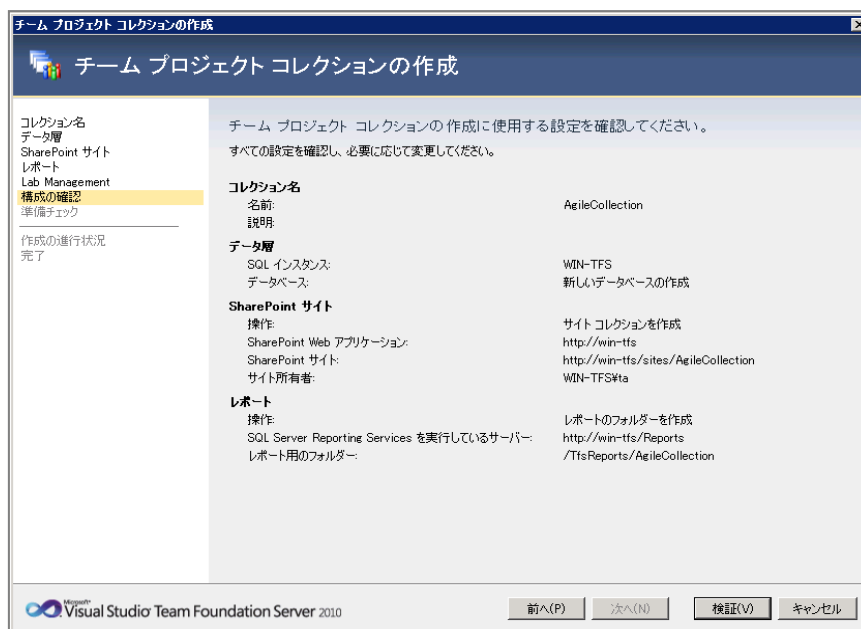


図 3.7 構成の確認 (TFS 管理コンソール)

つぎに、チーム プロジェクトを作成します。チーム プロジェクトを作成するには、まず、Visual Studio から TFS へ接続します。

- ① Visual Studio のファイルメニューから「チーム」 - 「Team Foundation Server へ接続」をクリックします (図 3.8)。

- ② チーム プロジェクトへ接続ダイアログが開くので、サーバー、およびチーム プロジェクト コレクションを選択して「接続」をクリックします (図 3.9)。
- ③ 初めてサーバーに接続するときは、「サーバー」ボタンをクリックし、Team Foundation Server の追加および削除ダイアログでサーバーを追加します (図 3.10)。

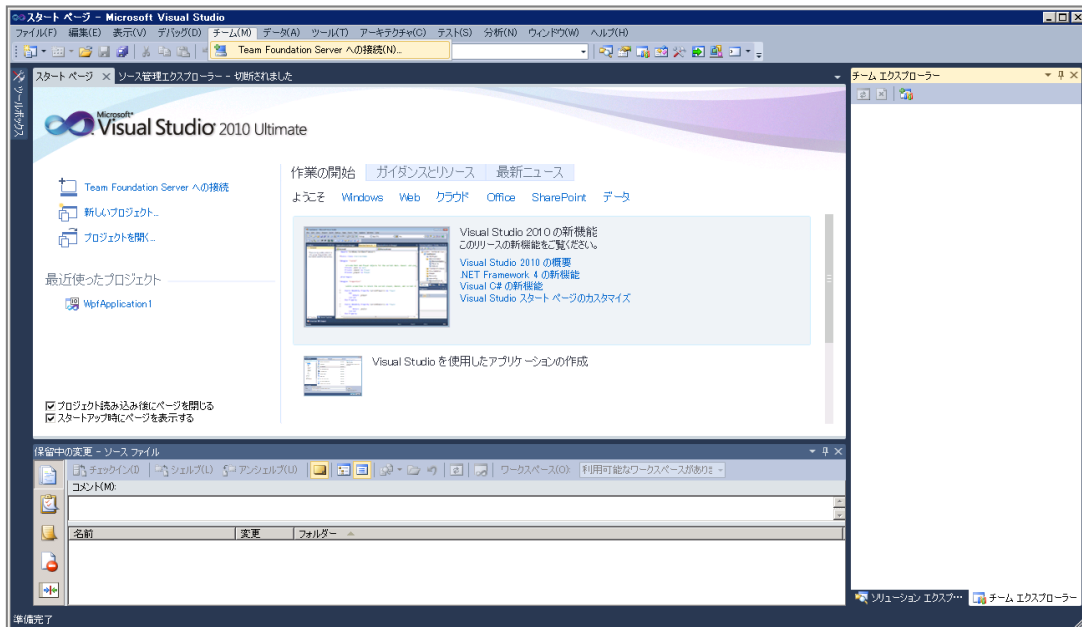


図 3.8 Team Foundation Server へ接続 (Visual Studio)

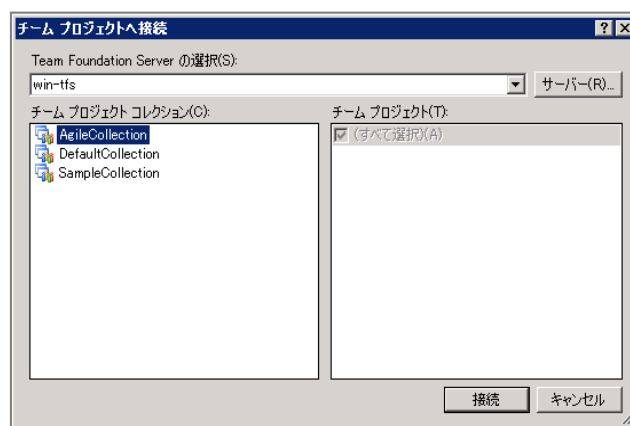


図 3.9 チーム プロジェクトへ接続 (Visual Studio)

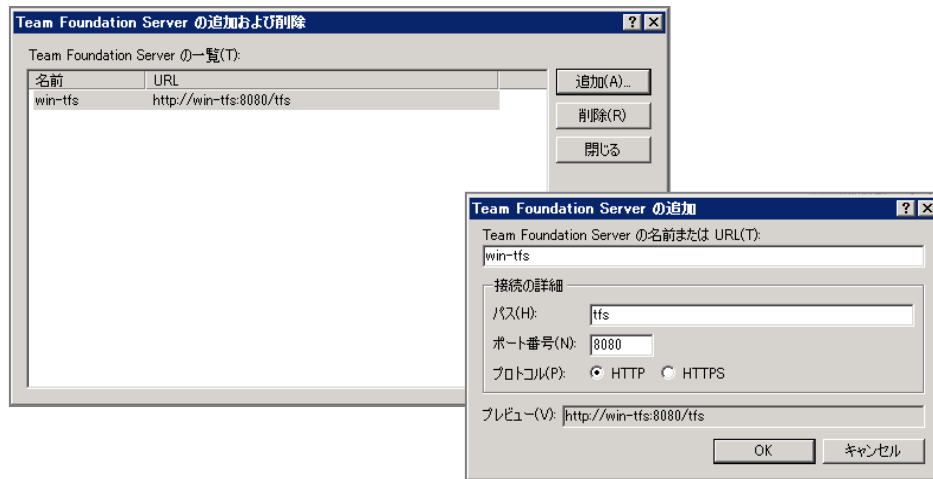


図 3.10 Team Foundation Server の追加 (Visual Studio)

Visual Studio を TFS へ接続したら、チーム エクスプローラー上でチーム プロジェクトを作成することができます。

- ① Visual Studio から TFS へ接続し、チーム エクスプローラー上のプロジェクト コレクションを右クリックして「新しいチーム プロジェクト」をクリックします (図 3.11)。
- ② 新しいチーム プロジェクトの作成ウィザードが起動するので、チーム プロジェクト名を設定し (図 3.12)、プロセス テンプレートの選択 (図 3.13)、チーム サイトの設定 (図 3.14)、ソース管理の設定を行い (図 3.15)、設定を確認 (図 3.16) してプロジェクトを作成したら完了です。なお、プロセス テンプレートは「MSF for Agile Software Development v5.0」を選択します。

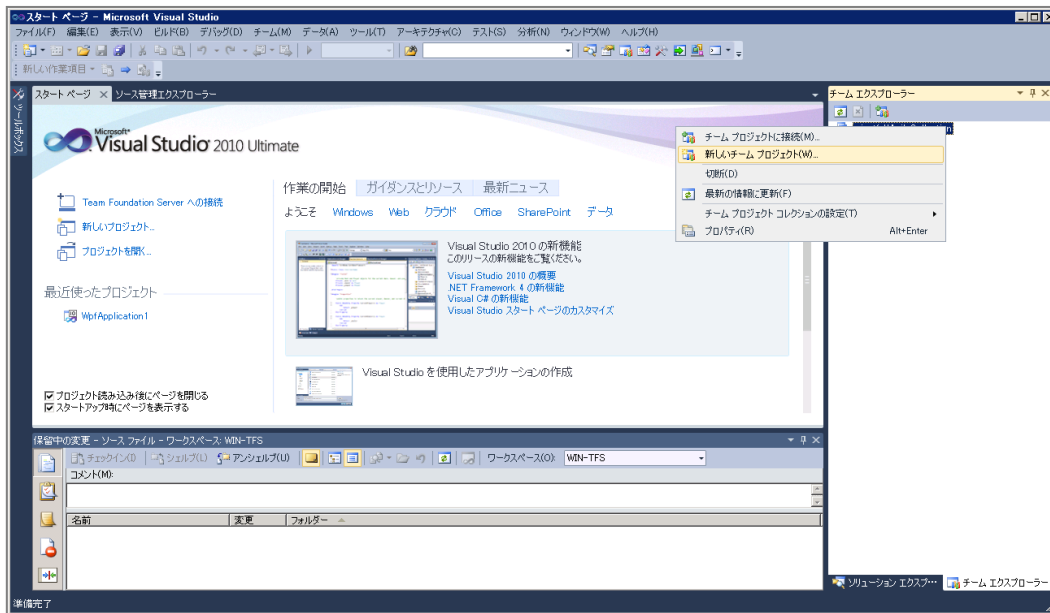


図 3.11 新しいチーム プロジェクトの作成 (Visual Studio)

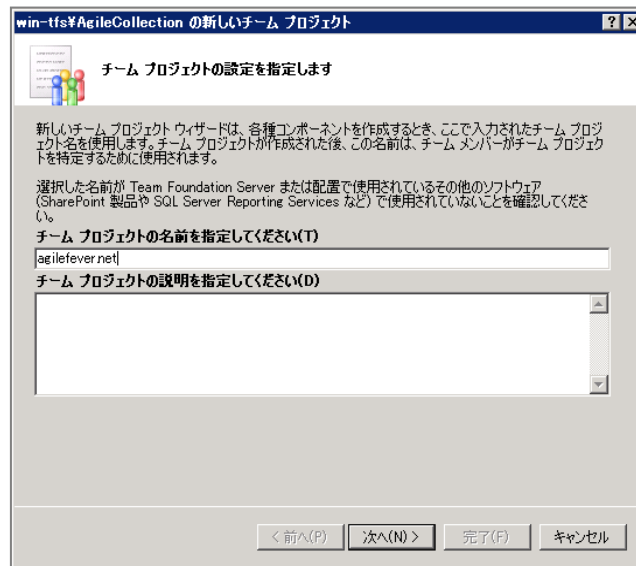


図 3.12 チーム プロジェクトの設定 (Visual Studio)

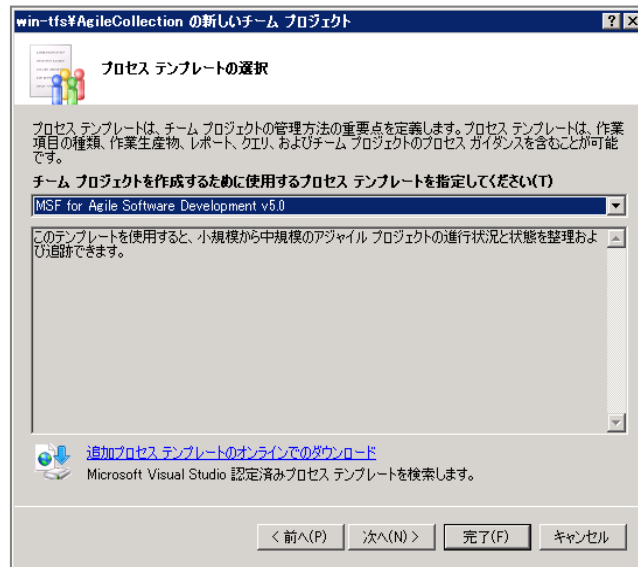


図 3.13 プロセス テンプレートの選択 (Visual Studio)

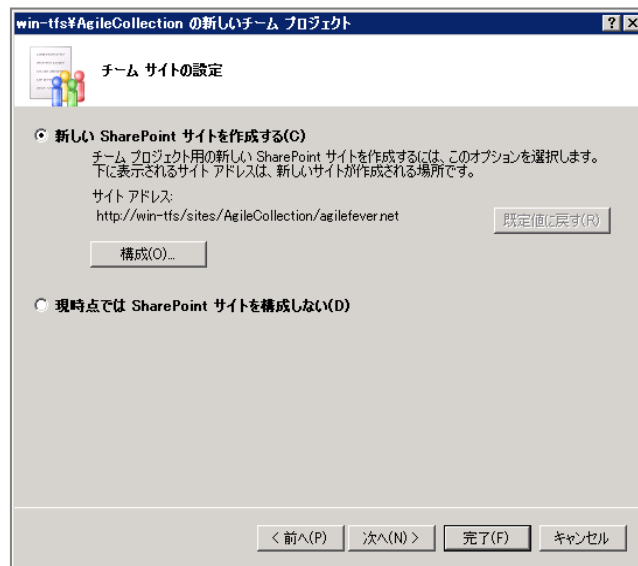


図 3.14 チーム サイトの設定 (Visual Studio)

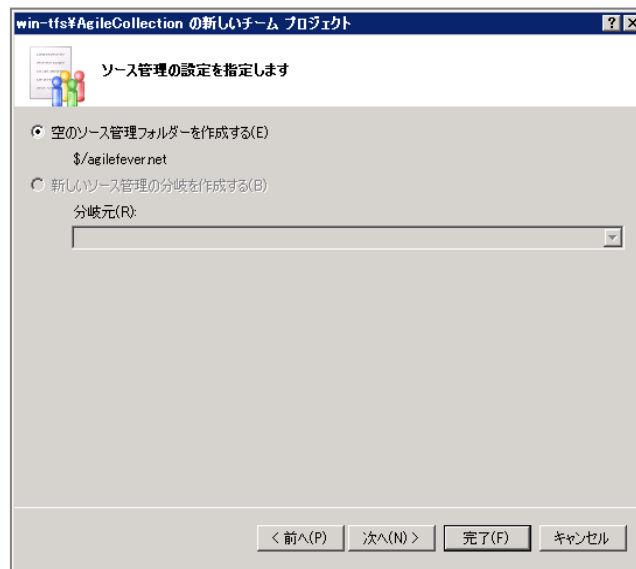


図 3.15 ソース管理の設定 (Visual Studio)

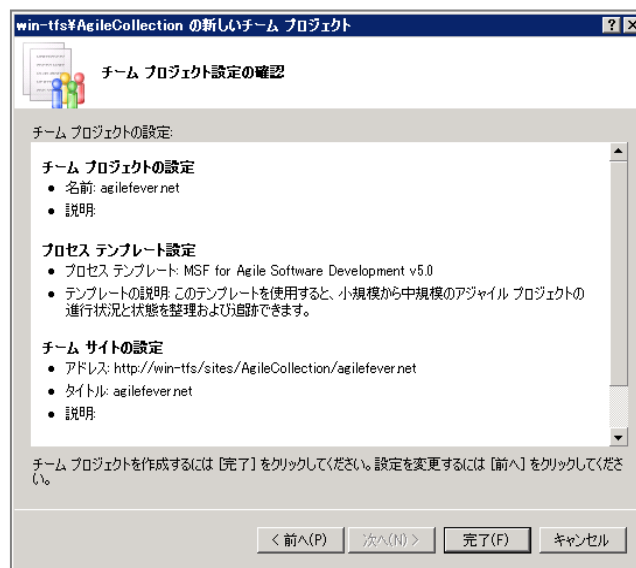


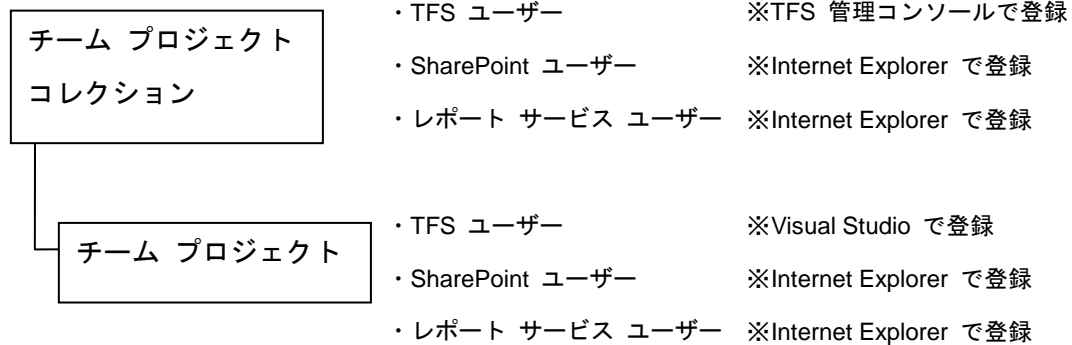
図 3.16 チーム プロジェクト設定の確認 (Visual Studio)

■ [TFS 操作] ユーザーの作成

次に、TFS を使用するユーザーを作成します。TFS のユーザーの対象はプロダクト オーナーであるコミュニティ事務局の担当者、**ステークホルダー**であるコミュニティ メンバー、そして開発チームのメンバーと、スクラム マスターであるあなた、つまり、プロジェクトの関係者全員となります。

TFS のユーザーは、Windows のアカウントを TFS や SharePoint、レポーティング サービスへ登録するため、あらかじめ Windows のアカウントを作成しておきます。ユーザー

ーはプロジェクト単位、およびチーム プロジェクト コレクション単位への登録ができ、TFS、SharePoint、レポート サービスへ、それぞれ登録します。



ここでは、チーム プロジェクトへのみ登録することにします。

まずは、TFS のグループへユーザーを追加します。TFS へのユーザーの登録は Visual Studio から実施します。

- ① Visual Studio を起動し、TFS サーバーへ接続します。
- ② ユーザーを追加するチーム プロジェクトを右クリックし、「**チーム プロジェクトの設定**」-「**グループ メンバーシップ**」を選択します (図 3.17)。
- ③ プロジェクト グループ ダイアログが開き (図 3.18)、ユーザーを追加します。TFS のプロジェクトにはあらかじめプロジェクト管理者、共同作成者、ビルダー、閲覧者というグループが用意されており、権限別にユーザーをグループに割り当てます。グループを選択して「**プロパティ**」ボタンをクリックすると、ユーザーを追加する画面が開きます (図 3.19)。

今回のプロジェクトでは、プロダクト オーナーとステークホルダーは閲覧者、開発メンバーは共同作成者、技術リーダー、およびスクラム マスターはプロジェクト管理者として割り当てることにします。

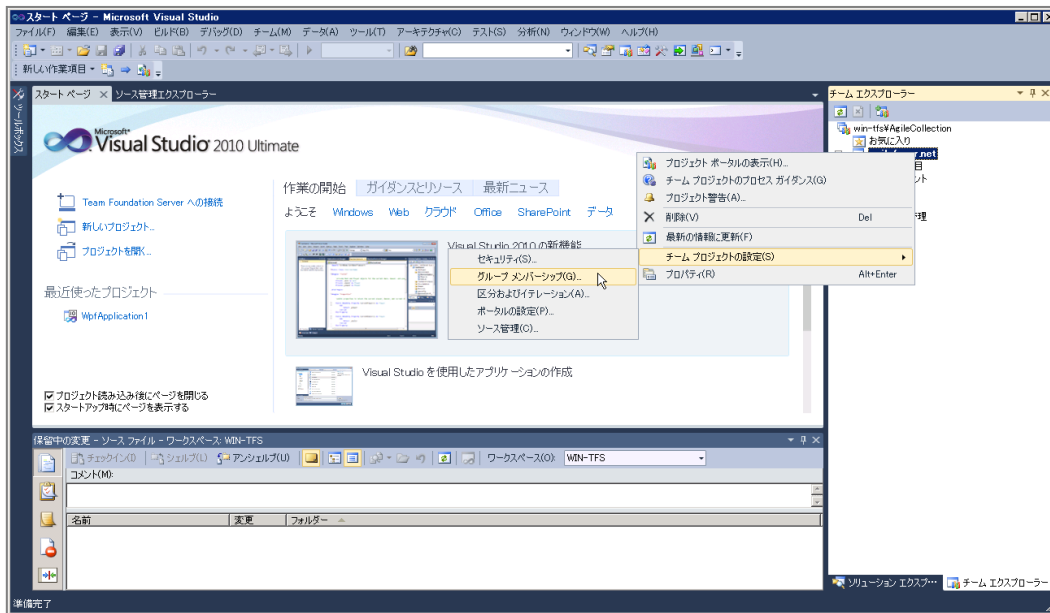


図 3.17 グループ メンバーシップ (Visual Studio)

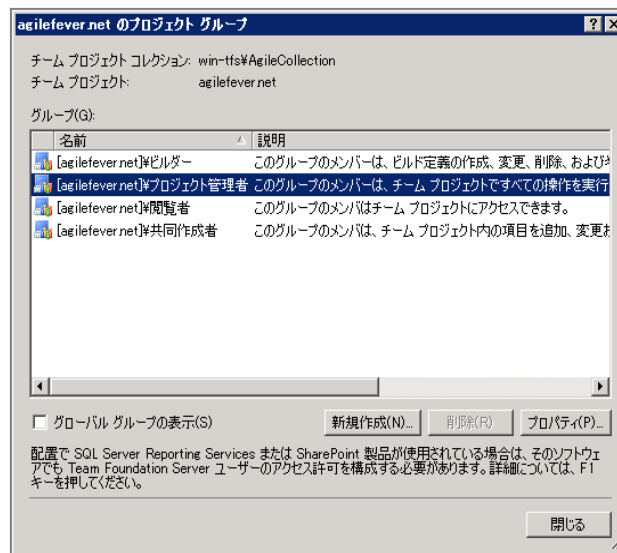


図 3.18 プロジェクト グループ (Visual Studio)

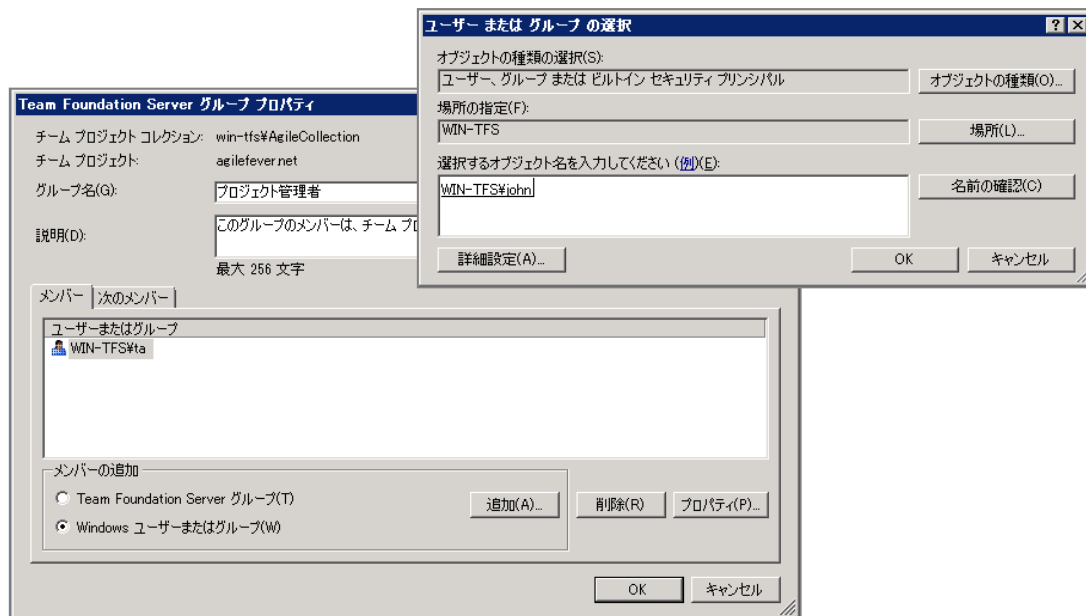


図 3.19 グループ プロパティの設定 (Visual Studio)

続いて、SharePoint へユーザーを登録します。SharePoint へのユーザーの登録はブラウザ (Internet Explorer) から実施します。

- ① Visual Studio を TFS サーバーへ接続し、チーム プロジェクトの右クリックメニューから「プロジェクト ポータルの表示」を選択します (図 3.20)。
- ② Internet Explorer が起動し、プロジェクトのポータルを開いたら「サイトの操作」 - 「サイトの設定」を選択し (図 3.21)、「ユーザーとグループ」をクリックします (図 3.22)。ユーザーとグループのページで「新規」 - 「ユーザーの追加」を選択し、ユーザーの追加画面でユーザーを登録します (図 3.23)。

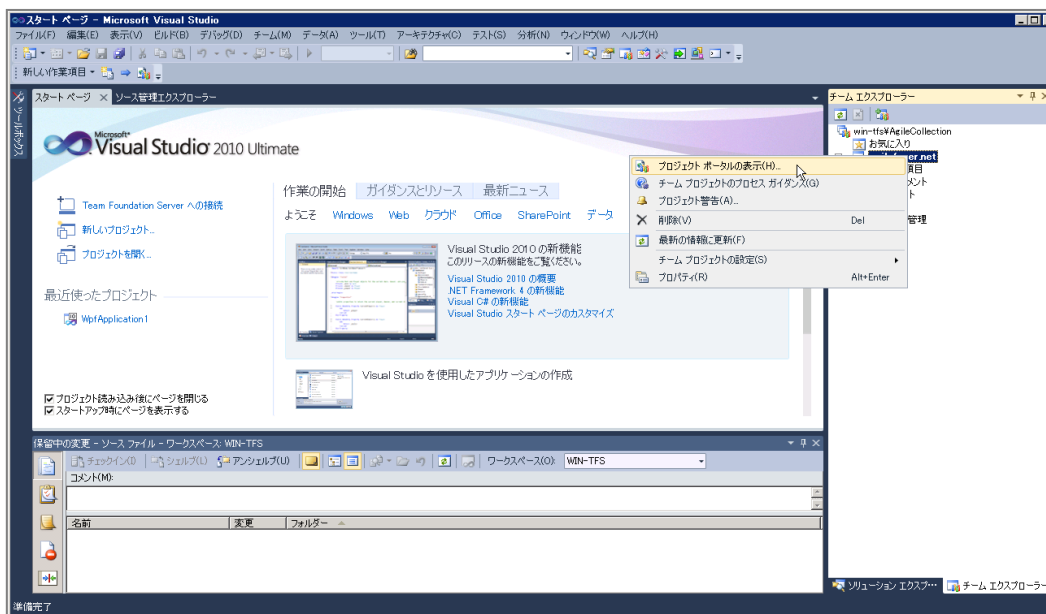


図 3.20 プロジェクト ポータルの表示 (Visual Studio)

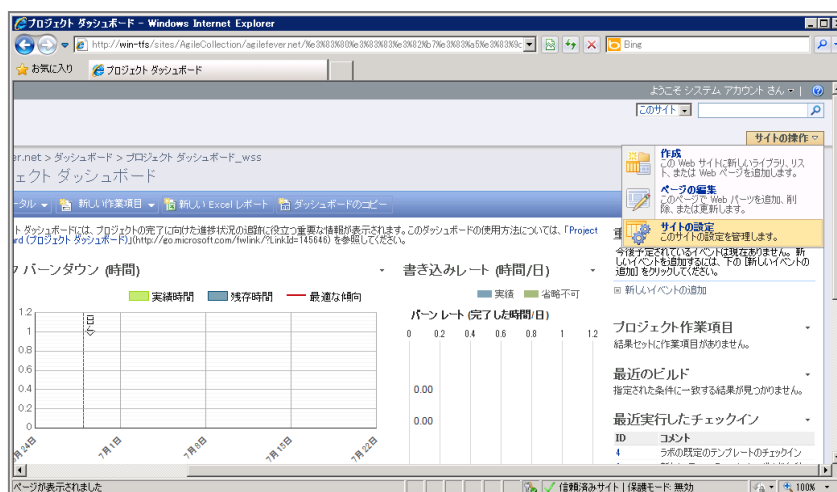


図 3.21 プロジェクト ポータル (Internet Explorer)



図 3.22 SharePoint サイトの設定 (Internet Explorer)

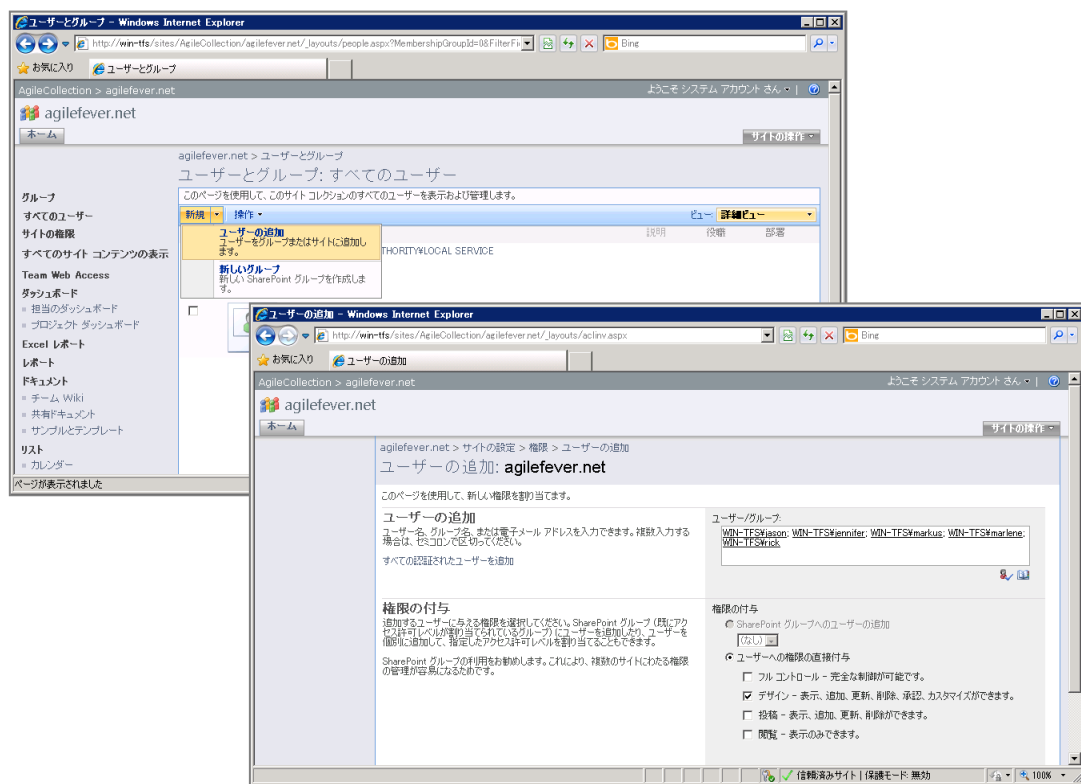


図 3.23 SharePoint ユーザーの追加 (Internet Explorer)

レポート サービスを利用する場合は SQL Server Reporting Services へのユーザー追加も必要となります。レポート サービスのユーザー登録も、SharePoint と同様、Internet Explorer から実施します。

- ① Visual Studio を TFS サーバーへ接続し、チーム プロジェクトのレポート ノードを右クリックし、「レポート サイトの表示」を選択します (図 3.24)。
- ② Internet Explorer が起動し、レポート マネージャーが表示されたら、「プロパティ」タブの「セキュリティ」をクリックし (図 3.25)、続いて「新しいロールの割り当て」をクリックして登録画面を開き (図 3.26)、ユーザーとロールを選択してユーザーを登録します。

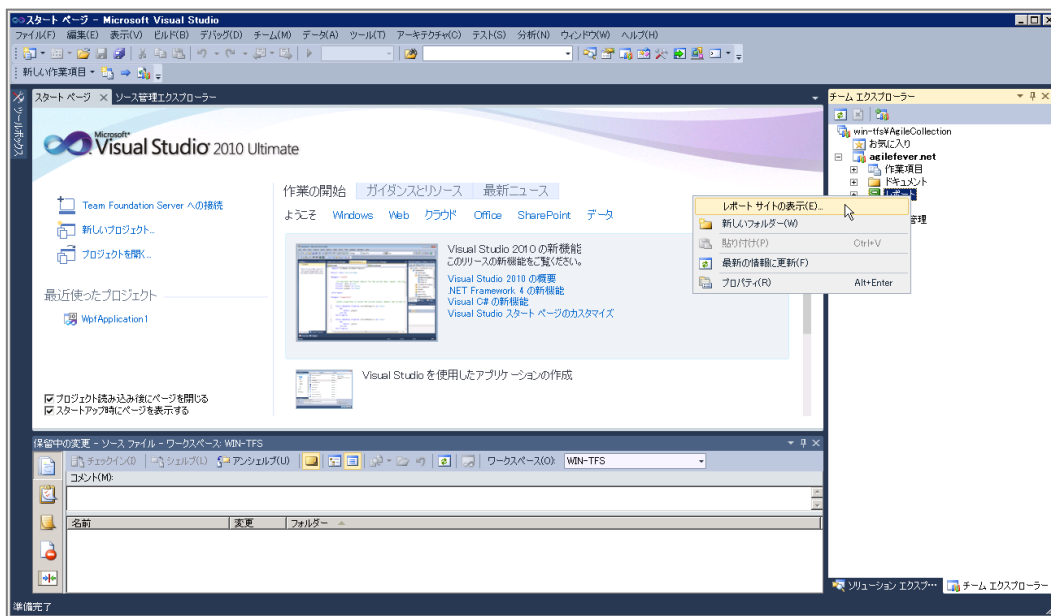


図 3.24 レポート サイトの表示 (Visual Studio)

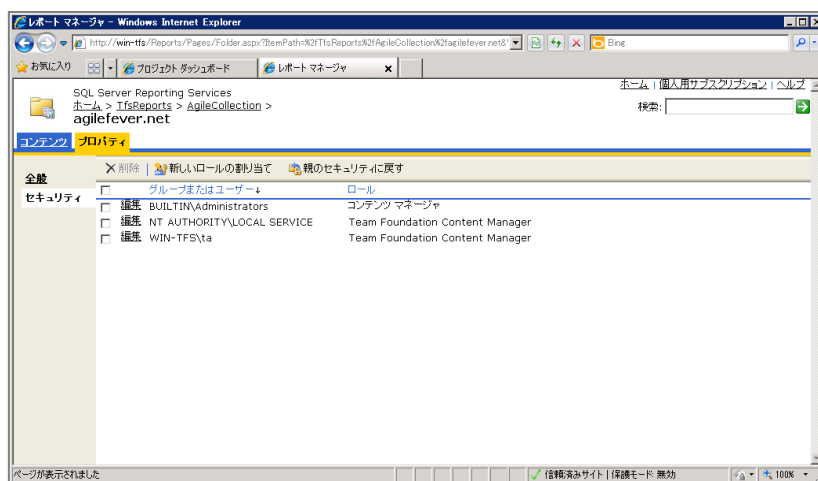


図 3.25 レポート マネージャー (Internet Explorer)

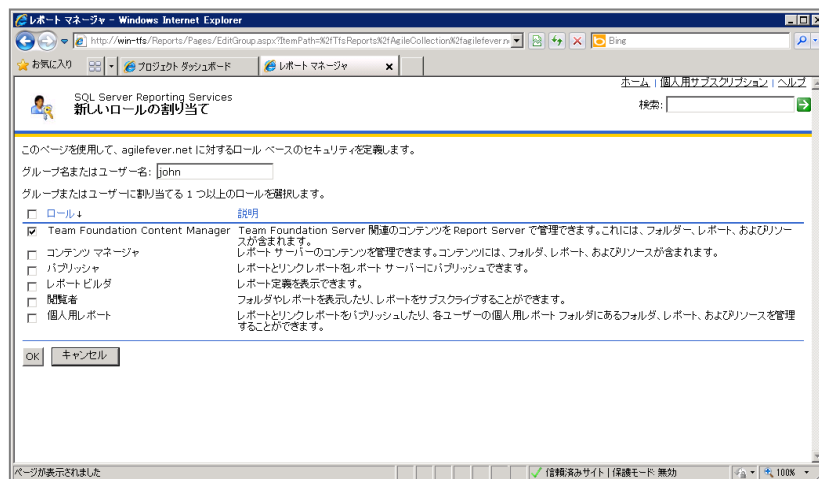


図 3.26 新しいロールの割り当て (Internet Explorer)

■ [TFS 操作] TFS サーバー接続の確認

TFS サーバーにプロジェクトを作成すると、Visual Studio 以外にも統合開発環境の Eclipse や Web ブラウザからも接続できるようになります。Web ブラウザによるアクセスには、先ほどのユーザー登録で開いたプロジェクト ポータル (図 3.27) や、ポータルの「Team Web Access」リンクからたどれる Team Web Access があります (図 3.28)。Team Web Access はウェブ版のチーム エクスプローラーで、作業項目の作成など一部の作業をウェブ上で実施することができます。

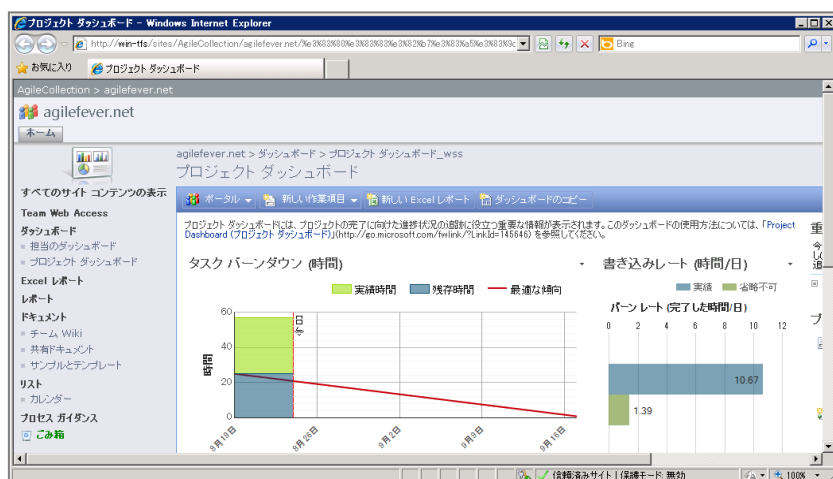


図 3.27 プロジェクト ポータル (Internet Explorer)

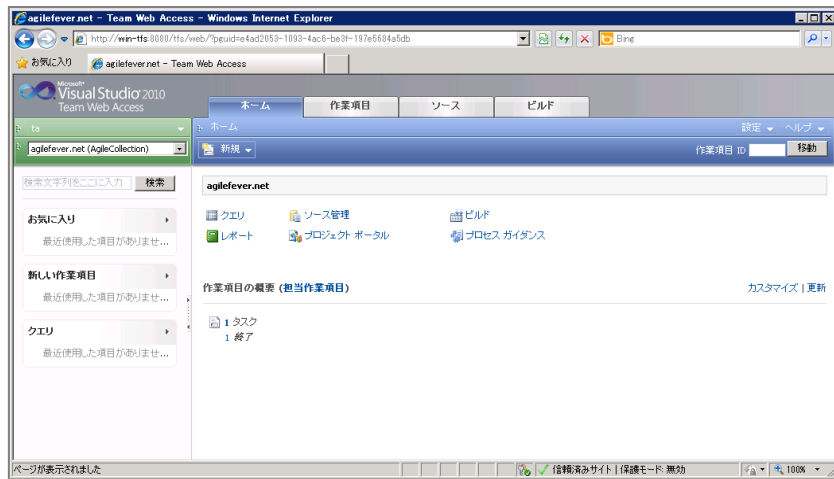


図 3.28 Team Web Access (Internet Explorer)

Eclipse からアクセスするには「Visual Studio Team Explorer Everywhere」という Eclipse プラグインを使います。プラグインをインストールして「Team Foundation」パースペクティブを開くとチーム エクスプローラーが起動します (図 3.29)。チーム エクスプローラーは Visual Studio のものと同じように操作することができます。

初めて使うときはチーム エクスプローラーを TFS サーバーのプロジェクトへ接続します。

- ① チーム エクスプローラーの「既存のチーム プロジェクトを追加」ボタンをクリックし、チーム プロジェクトの選択ダイアログを開きます (図 3.30)。
- ② すでに登録済みの TFS サーバーがあればコンボボックスから選択できます。新しいサーバーに接続するときは「サーバー」ボタンをクリックして、TFS サーバー一覧のダイアログを開いて TFS サーバーを登録します (図 3.31)。
- ③ TFS サーバーを選択し、接続したいプロジェクトをチェックして「完了」をクリックします。

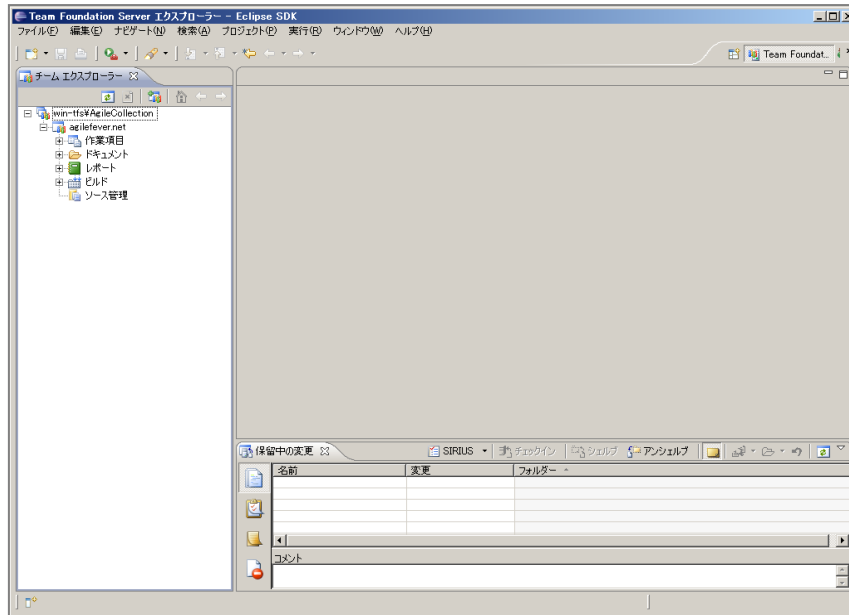


図 3.29 Team Explorer Everywhere (Eclipse)

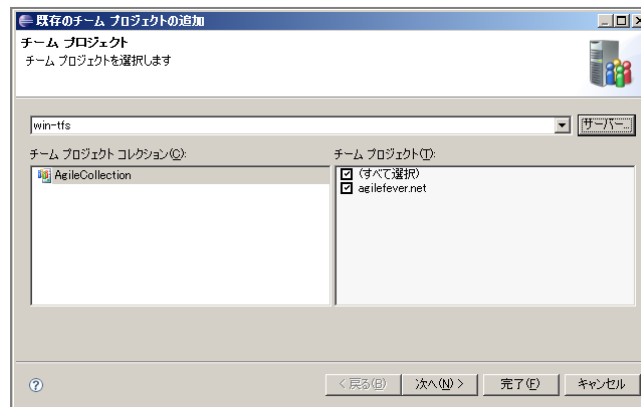


図 3.30 チーム プロジェクトの選択 (Eclipse)

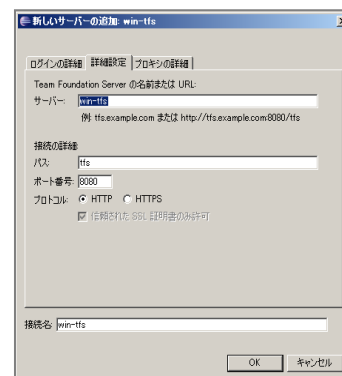
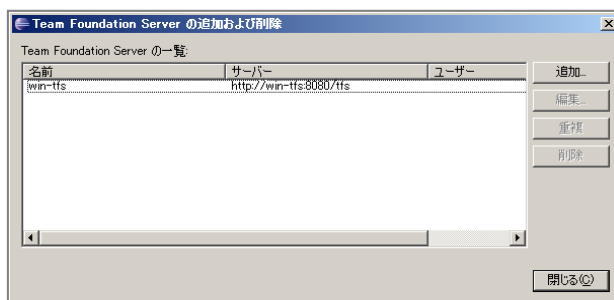


図 3.31 TFS サーバーの追加 (Eclipse)

3.3.2 最初のスプリント計画会議

スプリント計画会議にはプロダクト オーナーやステークホルダー、開発チームのメンバーを集めます。開発チーム全員が移動すると時間の浪費となるため、自社の会議室にコミュニティ事務局の担当者（プロダクト オーナー）とメンバーを招いて開催することにしました。

■ スプリント計画会議 (前半)

会議の前半の 4 時間は、プロダクト オーナーであるコミュニティ事務局の担当者が議長となり、製品バックログの内容について話し合いました。最初のスプリントでは、開発チームが 4 週間で実現できる範囲のストーリー（製品バックログ項目）を対象とします。また、3 ヶ月後のリリースに含めるストーリーを暫定的に決めました。会議の前半の話し合いで、製品バックログは次のように変更になりました。

表 3.11 スプリント計画会議前半後の製品バックログ

| No. | 要件 | 優先度 | ストーリー ポイント | 記載者 |
|-------------|---|-----|---------------|---------------|
| リリース 1.0 候補 | | | | |
| スプリント 1 候補 | | | | |
| ① | サイト閲覧者は、製品の説明を閲覧できる | 高 | 3pt | ○月×日 △△社□□ |
| ⑦ | ログイン済閲覧者は、フォーラムで製品に関する情報交換ができる ・ 製品の使い方の質問 ・ 製品機能に関する議論など | 高 | 20pt | ○月×日 △△社□□ |
| ② | ログイン済閲覧者は、製品の英語資料の日本語訳を閲覧できる | 高 | 5pt | ○月×日 △△社□□ |
| ⑨ | サイト閲覧者は、ドメイン名でサイトにアクセスできる (インフラ構築用ストーリー) | 低 | 8pt | |
| ⑩ | 未登録閲覧者は、自身のアカウントを登録することができる | 低 | 8pt | |

| No. | 要件 | 優先度 | ストーリー ポイント | 記載者 |
|-------------------|---------------------------------|-----|---------------|---------------|
| ⑪ | 未ログイン閲覧者は、ID とパスワードを指定してログインできる | 低 | 5pt | |
| スプリント 2 候補 | | | | |
| ④ | ログイン済閲覧者は、製品のリリース版をダウンロードできる | 中 | 5pt | ○月×日 △△社□□ |
| ⑥ | サイト閲覧者は、イベント情報を閲覧できる | 中 | 5pt | ○月×日 △△社□□ |
| ③ | ログイン済閲覧者は、製品のベータ版をダウンロードできる | 中 | 3pt | ○月×日 △△社□□ |
| ⑤ | サイト閲覧者は、ニュースを閲覧できる | 低 | 3pt | ○月×日 △△社□□ |
| : | | | | |

リリース計画は以下の図のように、スプリントを 3 回実施した後、最初のバージョンである 1.0 をリリースすることになりました。

今回の「スプリントの目標」は、「コミュニティ サイトは適切なプライバシー ポリシーの上で、フォーラムおよび、それを実行するための基盤を提供する」としました。

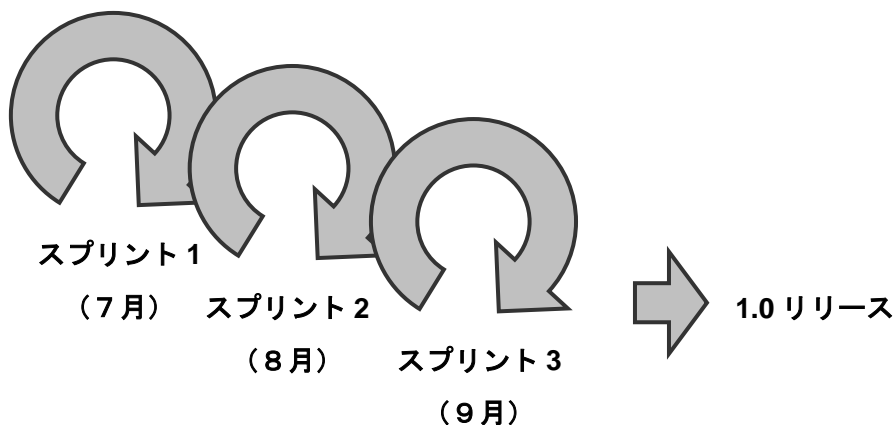


図 3.32 リリース計画

■ スプリント計画会議 (後半)

スプリント計画会議の前半の結果を元に、後半では開発チームによって製品バックログを **スプリント バックログ** のタスクに詳細化します。スプリント バックログは、スプリントの目標を達成するため、ストーリー（製品バックログ項目）毎にどのようなタスク（スプリント バックログ タスク）が必要かを、開発チームのメンバーたちによって導き出します。図 3.33 はスプリント計画会議の後半で使用されたホワイトボードのイメージです。

| ○月×日「Agile Fever (仮) コミュニティ サイト構築」スプリント計画会議 (後半) | |
|---|---|
| サイト閲覧者は、ドメイン名を指定してサイトにアクセスできる ・ インフラ設計 8h ・ ドメイン取得 4h ・ 開発サーバー調達 4h ・ 開発サーバーOS セットアップ 4h ・ 開発クライアント セットアップ 4h×7台分 : | サイト閲覧者は、製品の説明を閲覧できる ・ 原稿執筆→事務所 ・ 画面モックアップ作成 4h ・ 開発サーバーに原稿投入 1h : |
| ログイン済閲覧者は、フォーラムで、製品に関する情報交換ができる : | : |

図 3.33 スプリント計画会議 (後半) のホワイトボード

開発メンバーたちの意欲は高く、プロジェクトを成功させ、翌月以降もプロジェクトを継続したいと考えています。しかし、ここでは意欲的な反面、無理な見積りとならないか、過去の経験に基づいた現実的な見積りかに注意しましょう。なお、タスクはより具体的な作業項目なので、規模感を表すストーリー ポイントではなく、作業時間で見積もっています。

また、開発メンバーたちは自分たちの技術力をアピールし、ステークホルダーの信頼を得たいと考えているようです。したがって、今回のスプリントの候補になっている製品バックログの項目が早めに完了してしまうことに備え、次のスプリントの候補から優先度が高めのものも、ある程度意識合わせしておきましょう。

スプリント計画会議が終わったら、その結果を TFS へ反映します。この時点では、主に会議で決定したストーリーやタスクをバックログとして登録します。

リリース/スプリントの設定

TFS では工程に「区分」と「イテレーション」を設定できるので、区分をリリース、イテレーションをスプリントとして使います。新しいプロジェクト作成時にあらかじめいくつかのイテレーション定義が用意されるので、それをカスタマイズすればいいでしょう。

ストーリーの登録

TFS ではストーリーやタスクなど個別のテーブルではなく、作業項目という汎用的なテーブルに「ユーザー ストーリー」、「タスク」などの種別を持たせて作成します。

バックログのクエリを作成

TFS では、汎用的な作業項目としてストーリーやタスクを登録し、作業項目全体から該当項目を抽出するクエリを作成することができ、これを使って製品バックログ、スプリント バックログ、リリース バックログ用のクエリを用意します。たとえば、リリース バックログであれば、種類がユーザー ストーリーで、かつ、区分が「リリース 1.0」という条件に一致する作業項目を抽出するクエリを作成します。

タスクの登録

ストーリーと同様、タスクとしての作業項目を作成します。さらに、関連するストーリーの作業項目とのリンクを作成してストーリーとタスクの親子関係を設定します。

■ [TFS 操作] リリースとスプリントの設定

TFS の区分とイテレーションを使って、リリースとスプリントを設定します。なお、新しいプロジェクト作成時にイテレーションとそれぞれの標準クエリが3つ分用意されているので、これらをスプリントの1～3として使うことにし、ここでは、リリース 1.0 と 1.1 の区分を作成します。

- ① Visual Studio を TFS サーバーに接続し、チーム エクスプローラーのプロジェクトを右クリックして、「チーム プロジェクトの設定」・「区分およびイテレーション」を選択します (図 3.34)。
- ② 区分およびイテレーションのダイアログが開く (図 3.35) ので、区分タブの区分を選択して、「子ノードの追加」をクリックすると新しい区分が追加されるので、名前を「リリース 1.0」とします。同様にリリース 1.1 も作成します。なお、区分およびイテレー

ションはそれぞれ階層化することもできます。

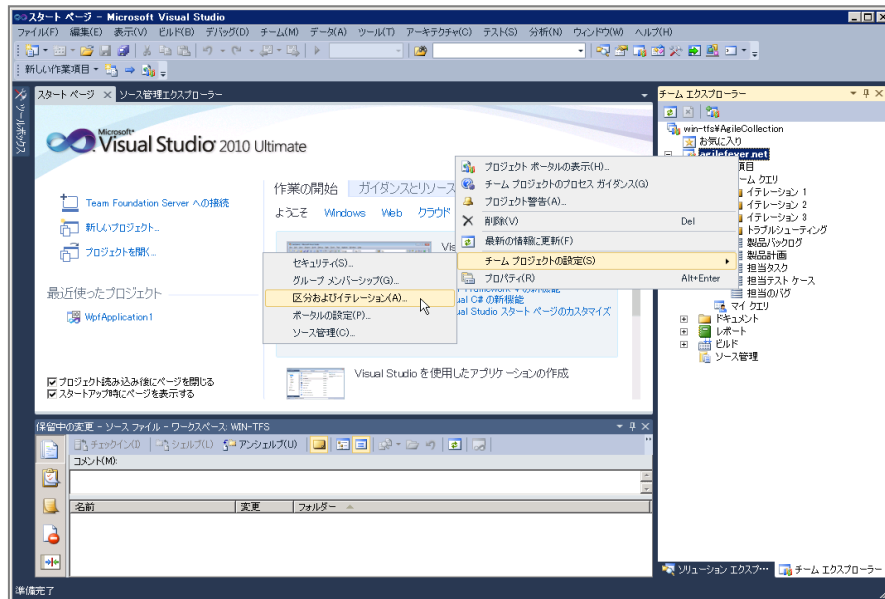


図 3.34 区別およびイテレーション (Visual Studio)

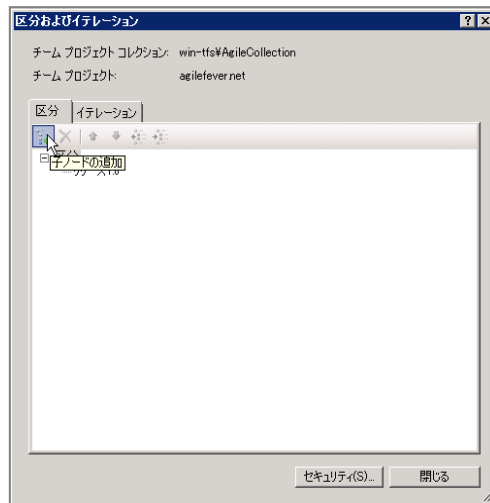


図 3.35 区別の追加 (Visual Studio)

■ [TFS 操作] ストーリーの作成

製品バックログのストーリーを作業項目として登録し、スプリント計画会議で決定したリリース、スプリントにそれぞれ振り分けていきます。

- ① Visual Studio を TFS サーバーに接続し、チーム エクスプローラーのプロジェクト -

作業項目を右クリックし、「新しい作業項目」-「ユーザー ストーリー」を選択します (図 3.36)。

- ② ユーザー ストーリーの編集画面が開く (図 3.37) ので、タイトルにストーリーを記述し、分類の区分とイテレーションにそれぞれリリースとスプリントを設定し、ストーリー ポイントなどを記入します。最後に「作業項目の保存」ボタンをクリックすると、ストーリーが作成されます。
- ③ チーム エクスプローラーの作業項目にある「製品バックログ」クエリをダブルクリックして開くと、ストーリーの一覧 (製品バックログ) を参照できます (図 3.38)。

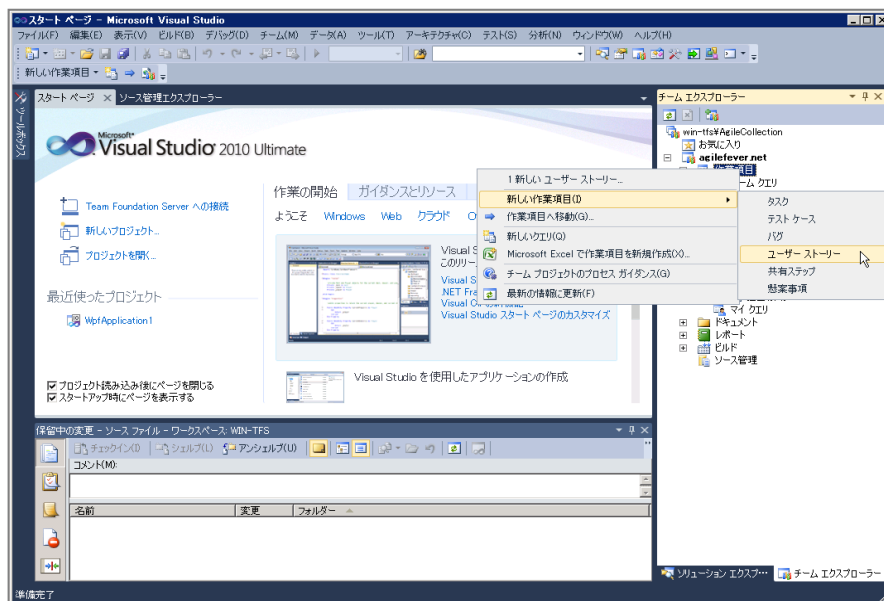


図 3.36 ユーザー ストーリーの新しい作業項目 (Visual Studio)

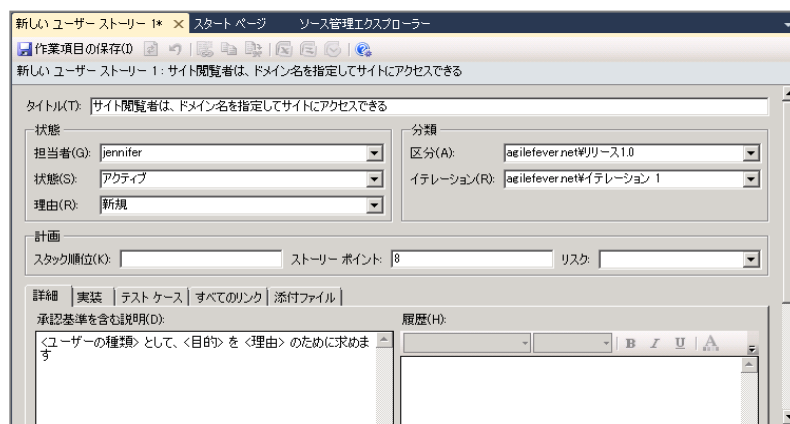


図 3.37 新しいユーザー ストーリー (Visual Studio)

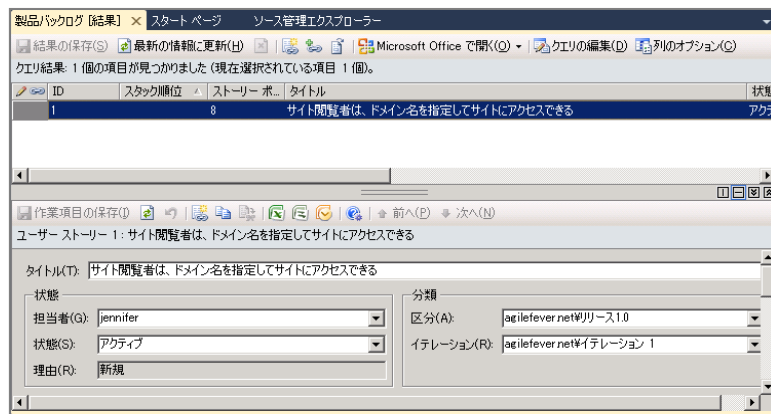


図 3.38 製品バックログの参照 (Visual Studio)

■ [TFS 操作] リリース バックログの作成

作業項目の一覧はクエリを作成して参照します。製品バックログやスプリント毎のクエリはあらかじめ用意されているものを使えばいいので、ここではリリース バックログ用に新しくクエリを作成します。

- ① Visual Studio を TFS サーバーに接続し、チーム エクスプローラーのプロジェクト – 作業項目 – チームクエリを右クリックし、「新しいクエリ」を選択します (図 3.39)。
- ② クエリの編集画面が開く (図 3.40) ので、作業項目の検索条件を設定します。リリース バックログの場合は作業項目の種類を「ユーザー ストーリー」、区分パスを追加して「プロジェクト/リリース 1.0」を設定します。
- ③ 「クエリの保存」ボタンをクリックするとクエリの保存名を聞かれるので、ここでは「リリース バックログ 1.0」とします (図 3.41)。

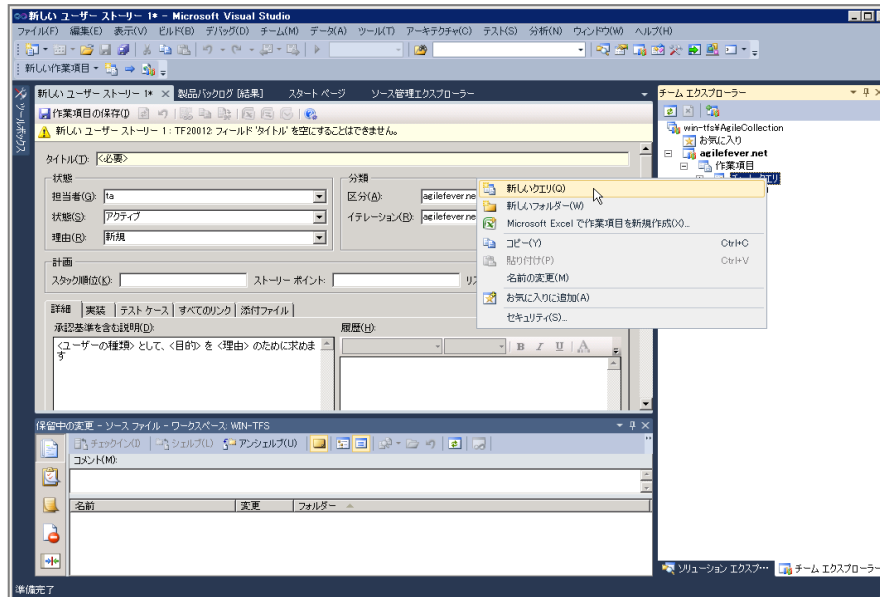


図 3.39 新しいクエリ (Visual Studio)

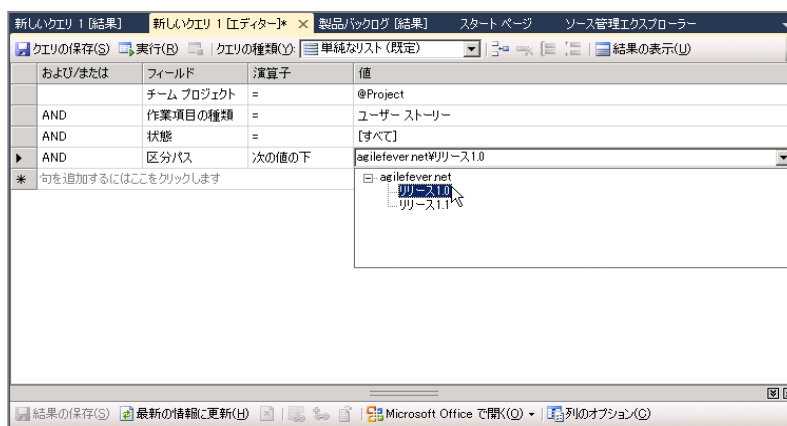


図 3.40 クエリの編集 (Visual Studio)

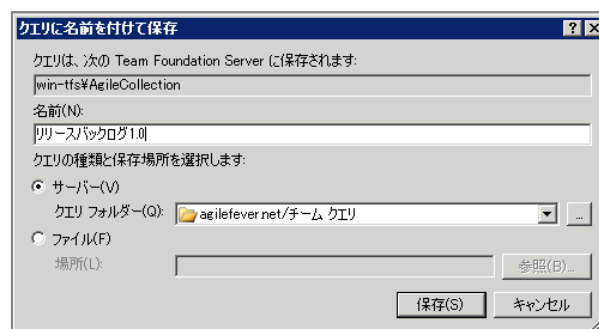


図 3.41 クエリの保存 (Visual Studio)

■ [TFS 操作] タスクの作成

タスクの作成は、ストーリーと同じように、スプリント バックログのタスクを作業項目として登録し、スプリント計画会議で決定したリリース、スプリントにそれぞれ振り分けることができます。

しかし、TFS による製品計画、スプリント計画では、計画ツールとしての Excel が強力なため、MSF for Agile Software Development v5.0 には、あらかじめ Excel ブックが用意され、製品計画やスプリント計画を Excel から作業できるようになっています。使い慣れた Excel から計画や進捗管理、適切なフィードバックとフォローを実施し、開発チームのメンバーは Excel を参照することなく、いつもの開発環境 (Visual Studio や Eclipse) の中で自分のタスクを把握しておけば、TFS がこれらを自然に連携してくれます。

ここでは、まず製品計画ブックによる製品計画について説明します。

- ① Visual Studio のチーム エクスプローラーまたは、プロジェクト ポータルからドキュメント・共有ドキュメントを展開し、「製品計画.xlsx」を開きます (図 3.42)。
- ② 製品計画ブックが開き、製品バックログの一覧が表示されます (図 3.43)。ここから区分、イテレーションの設定ができ、Excel のチーム タブから「区分およびイテレーションの編集」ボタンをクリックすると、設定ダイアログが開きます (図 3.44)。
- ③ 区分、イテレーションを設定したら、「イテレーション」シートを開き、スプリント期間を設定します。「中断」シートで休日を設定すれば、イテレーションの日数から休日分が除かれます (図 3.45)。
- ④ 「製品バックログ」シートに戻り、ストーリー ポイントを見ながらスプリント間の負荷が偏らないよう、ストーリーの割り振りを調整します。

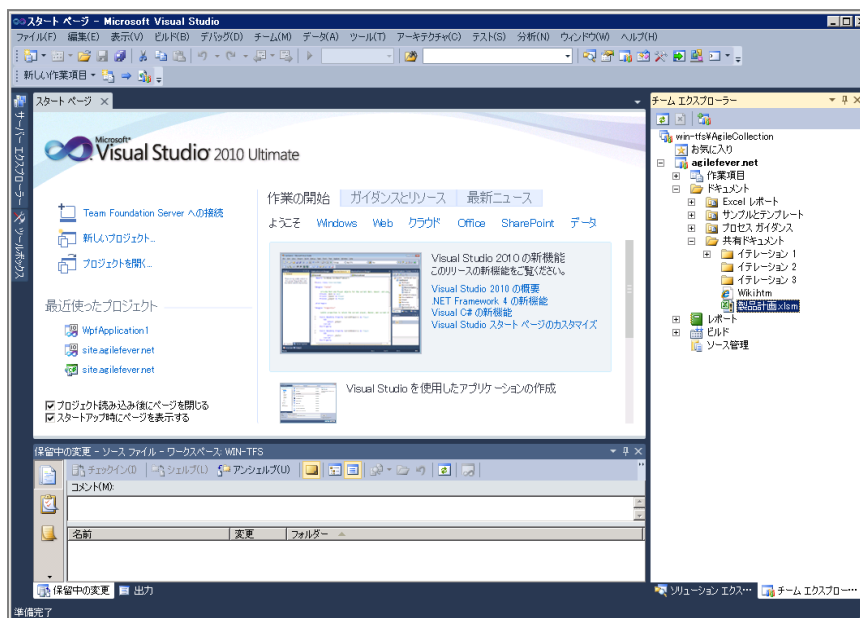


図 3.42 製品計画ブックを開く (Visual Studio)

製品計画.xlsx - Microsoft Excel

チーム プロジェクトの区分およびイテレーションを編集します。

チーム: agilefever.net サーバー: win-tfsVAgileCollection クエリ: 製品計画 リストの種類: フラット

| ID | ステータス | ストーリー | タイトル | 状態 | イテレーション | 優先度 | 区分 |
|----|-------|-------|--------------------------------|-------|-----------|-----|------|
| 3 | 1 | 8 | サイト閲覧者は、ドメイン名を指定してサイトにアクセスできる | アクティブ | イテレーション 1 | 高 | リリース |
| 4 | 1 | 3 | サイト閲覧者は、製品の説明を閲覧できる | アクティブ | イテレーション 1 | 高 | リリース |
| 5 | 4 | 20 | ログイン済閲覧者は、フォーラムで製品に関する情報交換ができる | アクティブ | イテレーション 1 | 高 | リリース |
| 6 | 5 | 5 | ログイン済閲覧者は、製品の英語資料の日本語訳を閲覧できる | アクティブ | イテレーション 1 | 高 | リリース |
| 7 | 6 | 8 | 未登録閲覧者は、自身のアカウントを登録することができる | アクティブ | イテレーション 1 | 高 | リリース |
| 8 | 7 | 5 | 未ログイン閲覧者は、IDとパスワードを指定してログインできる | アクティブ | イテレーション 1 | 高 | リリース |
| 9 | 8 | 5 | ログイン済閲覧者は、製品のリリース版をダウンロードできる | アクティブ | イテレーション 2 | 高 | リリース |
| 10 | 9 | 5 | サイト閲覧者は、イベント情報を閲覧できる | アクティブ | イテレーション 2 | 高 | リリース |
| 11 | 10 | 3 | ログイン済閲覧者は、製品のページ版をダウンロードできる | アクティブ | イテレーション 2 | 高 | リリース |
| 12 | 11 | 3 | サイト閲覧者は、ニュースを閲覧できる | アクティブ | イテレーション 2 | 高 | リリース |
| 13 | 18 | 3 | サイト閲覧者は、適切な承認メールを受け取ることができる | アクティブ | イテレーション 2 | 高 | リリース |

図 3.43 製品バックログ (Excel)

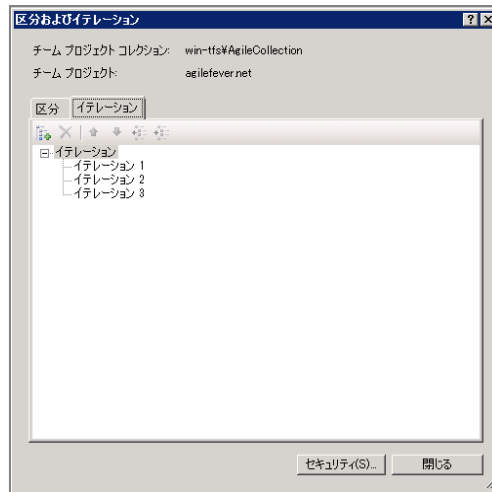


図 3.44 区別、イテレーション設定ダイアログ (Excel)

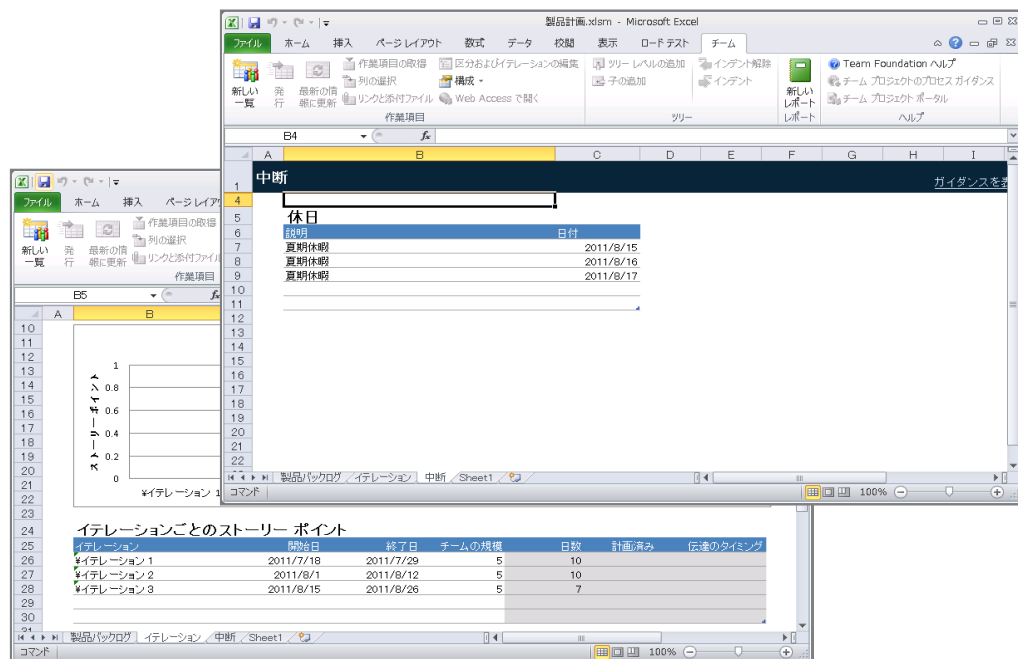


図 3.45 スプリント期間、休日の設定 (Excel)

続いて、イテレーション バックログ ブックによるスプリント計画について説明します。

- ⑤ Visual Studio のチーム エクスプローラーまたは、プロジェクト ポータルからドキュメント・共有ドキュメント – イテレーション 1 を展開し、「イテレーション バックログ.xlsm」を開きます (図 3.46)。
- ⑥ イテレーション バックログ ブックが開き、スプリント バックログ タスクの一覧が表

示されます (図 3.47)。

- ⑦ 製品計画のときと同じように、「設定」シートでスプリント期間、「中断」シートで休日や、メンバーの個々の予定を設定することができます (図 3.48)。
- ⑧ 「キャパシティ」シートでは、個々のメンバーの負荷状況を確認でき、特定メンバーへの負荷の集中を防ぐことができます。「バーンダウン」シートでは、バーンダウンチャートを表示し、タスクの進捗を確認することができます (図 3.49)。

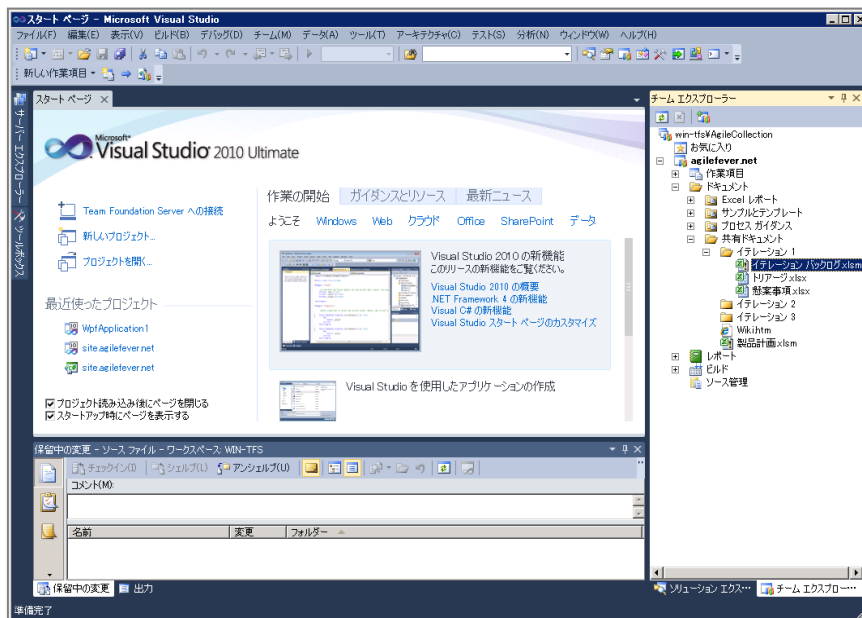


図 3.46 イテレーション バックログ ブックを開く (Visual Studio)

ファイルホーム挿入ページレイアウト表形式データ校閲表示ロードデストチームデザイン

新しい一覧

最新の情
報に更新

作業項目の取得
の選択

区分およびイテレーションの編集
構成

リンクと添付ファイル
Web Access で開く

作業項目

発行

イテレーションの追加

子の追加

インデント解除

インデント

ヘルプ

Team Foundation ヘルプ

チーム コマンドのプロセス ガイダンス

チーム コマンド ポールカ

イテレーション「バックログ」.xlsx - Microsoft Excel

新しいレポート
レポート
ヘルプ

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ」.xlsx - Microsoft Excel

イテレーション「バックログ

図 3.47 スプリント バックログ (Excel)

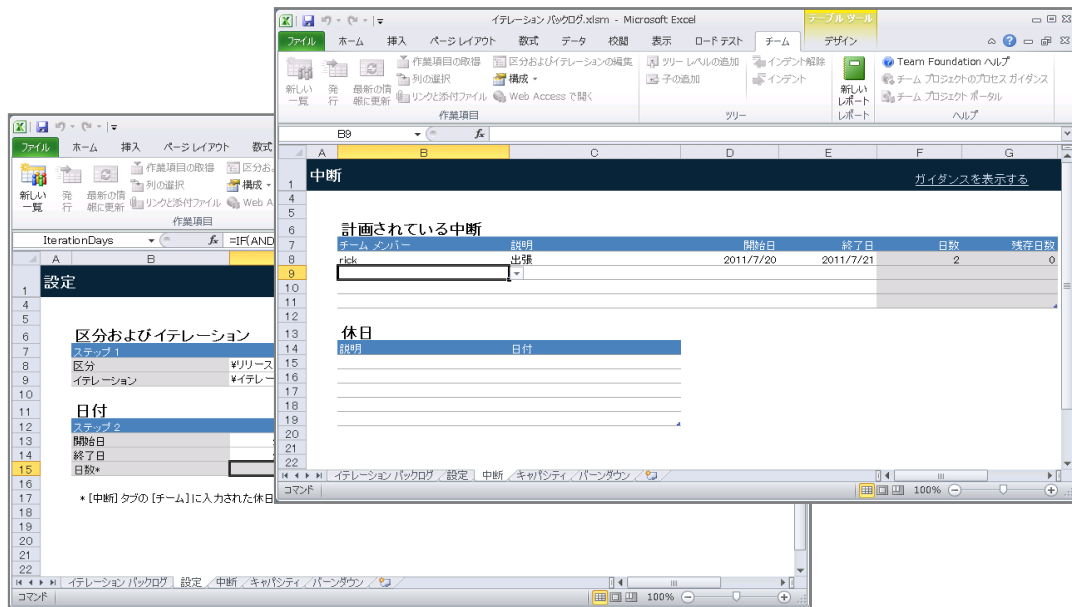


図 3.48 スプリント期間、休日の設定 (Excel)

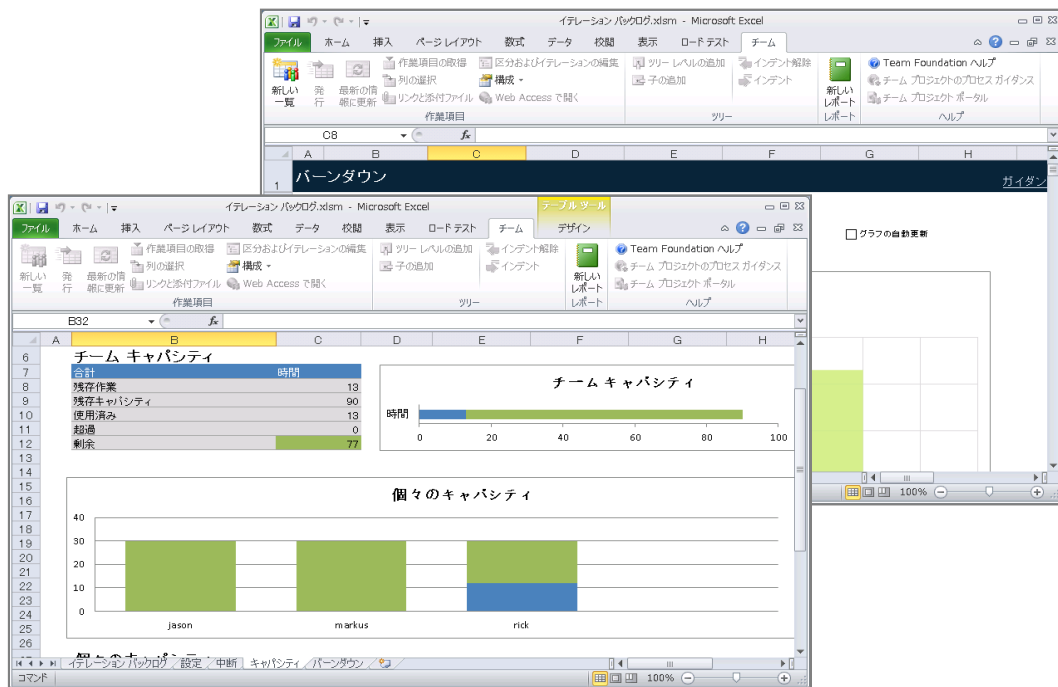


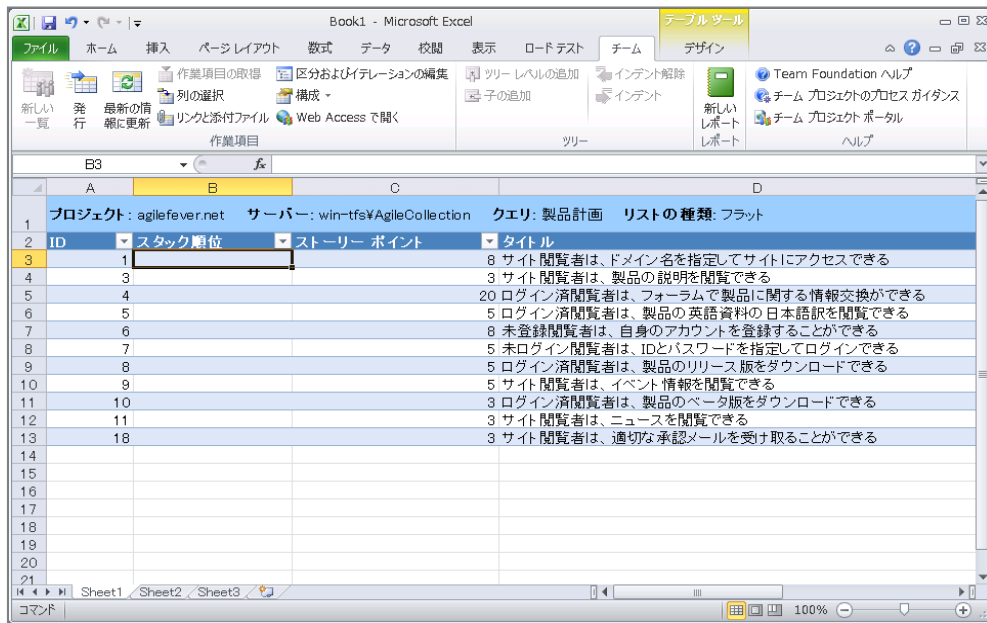
図 3.49 負荷状況と進捗状況の確認 (Excel)

■ [TFS 操作] Excel によるバックログの参照

TFS の情報は、ブックと同じように Microsoft Excel で、クエリの検索結果をベースに参照、編集することもできます。表形式でストーリー ポイントなど複数項目をまとめて編

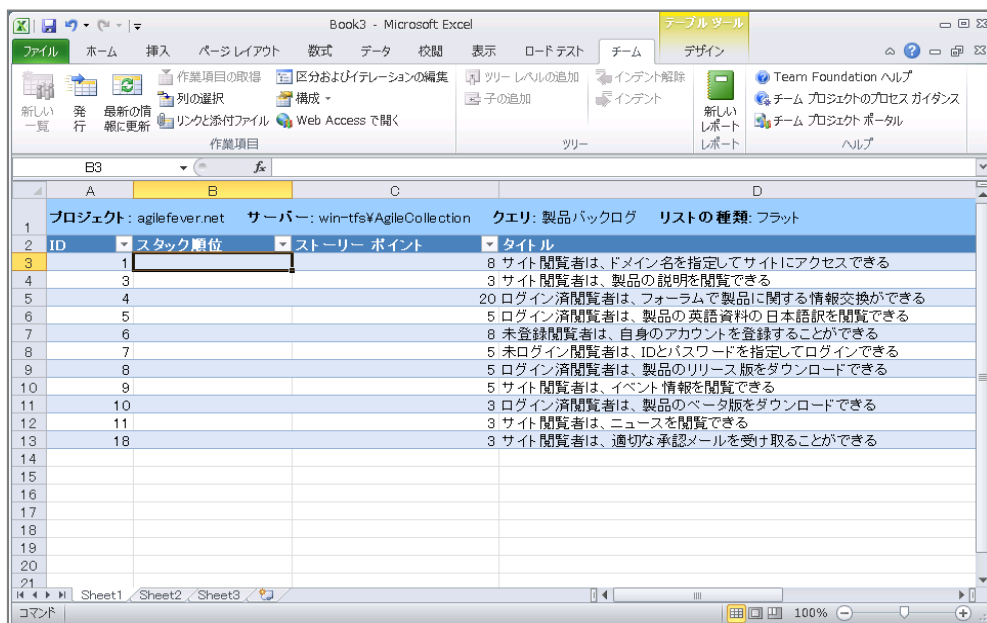
集できるので便利です。また、Visual Studio のチーム エクスプローラーからクエリを開き、「Microsoft Office から開く」をクリックして Excel を起動することもできます。

以下は、製品計画、製品バックログ、イテレーション バックログの各クエリを Excel で開いた場合の画面です (図 3.50、図 3.51、図 3.52)。また、一覧表示だけでなく、Excel から新しく作業項目を追加することもできます (図 3.53)。



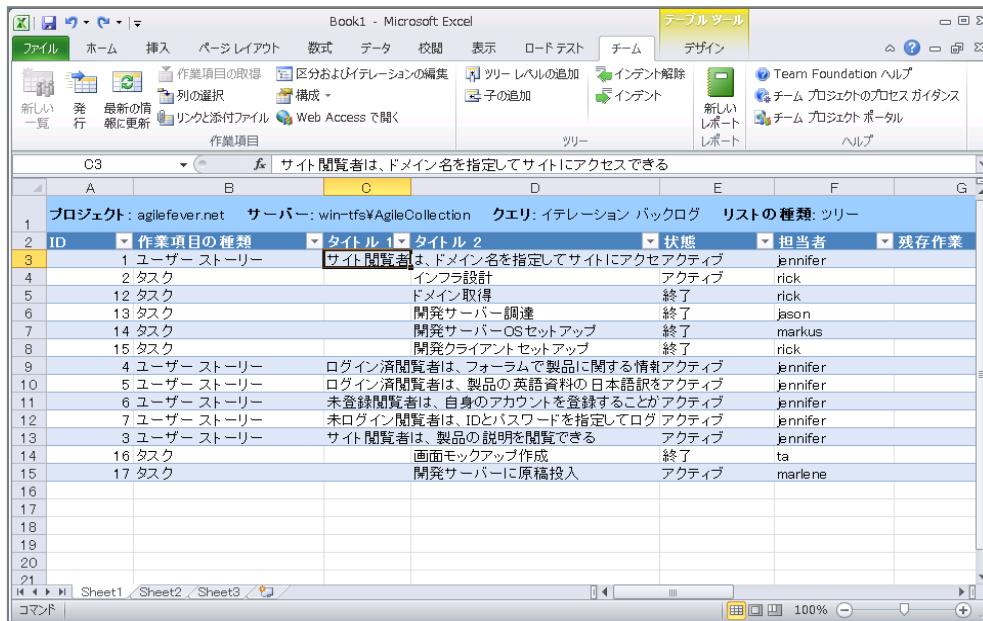
| ID | スタック順位 | ストーリー ポイント | タイトル |
|----|--------|------------|-----------------------------------|
| 1 | | | 8 サイト閲覧者は、ドメイン名を指定してサイトにアクセスできる |
| 2 | | | 3 サイト閲覧者は、製品の説明を閲覧できる |
| 3 | | | 20 ログイン済閲覧者は、フォーラムで製品に関する情報交換ができる |
| 4 | | | 5 ログイン済閲覧者は、製品の英語資料の日本語訳を閲覧できる |
| 5 | | | 8 未登録閲覧者は、自身のアカウントを登録することができる |
| 6 | | | 5 未ログイン閲覧者は、IDとパスワードを指定してログインできる |
| 7 | | | 5 ログイン済閲覧者は、製品のリリース版をダウンロードできる |
| 8 | | | 5 サイト閲覧者は、イベント情報を閲覧できる |
| 9 | | | 3 ログイン済閲覧者は、製品のベータ版をダウンロードできる |
| 10 | | | 3 サイト閲覧者は、ニュースを閲覧できる |
| 11 | | | 3 サイト閲覧者は、適切な承認メールを受け取ることができる |
| 12 | | | |
| 13 | | | |
| 14 | | | |
| 15 | | | |
| 16 | | | |
| 17 | | | |
| 18 | | | |
| 19 | | | |
| 20 | | | |
| 21 | | | |

図 3.50 製品計画シート (Excel)



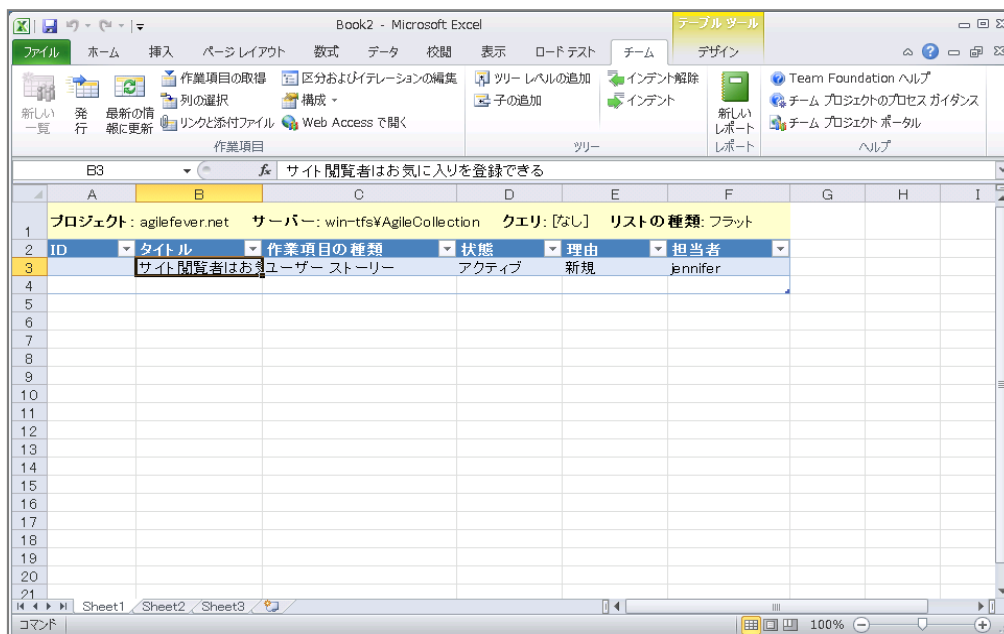
| ID | スタック順位 | ストーリー ポイント | タイトル |
|----|--------|------------|-----------------------------------|
| 1 | | | 8 サイト閲覧者は、ドメイン名を指定してサイトにアクセスできる |
| 2 | | | 3 サイト閲覧者は、製品の説明を閲覧できる |
| 3 | | | 20 ログイン済閲覧者は、フォーラムで製品に関する情報交換ができる |
| 4 | | | 5 ログイン済閲覧者は、製品の英語資料の日本語訳を閲覧できる |
| 5 | | | 8 未登録閲覧者は、自身のアカウントを登録することができる |
| 6 | | | 5 未ログイン閲覧者は、IDとパスワードを指定してログインできる |
| 7 | | | 5 ログイン済閲覧者は、製品のリリース版をダウンロードできる |
| 8 | | | 5 サイト閲覧者は、イベント情報を閲覧できる |
| 9 | | | 3 ログイン済閲覧者は、製品のベータ版をダウンロードできる |
| 10 | | | 3 サイト閲覧者は、ニュースを閲覧できる |
| 11 | | | 3 サイト閲覧者は、適切な承認メールを受け取ることができる |
| 12 | | | |
| 13 | | | |
| 14 | | | |
| 15 | | | |
| 16 | | | |
| 17 | | | |
| 18 | | | |
| 19 | | | |
| 20 | | | |
| 21 | | | |

図 3.51 製品バックログシート (Excel)



| ID | 作業項目の種類 | タイトル 1 | タイトル 2 | 状態 | 担当者 | 残存作業 |
|----|------------|-------------------------------|--------|-------|----------|------|
| 1 | ユーザー ストーリー | サイト閲覧者は、ドメイン名を指定してサイトにアクセスできる | | アクティブ | jennifer | |
| 2 | タスク | インフラ設計 | | アクティブ | rick | |
| 3 | タスク | ドメイン取得 | | 終了 | rick | |
| 4 | タスク | 開発サーバー調達 | | 終了 | jason | |
| 5 | タスク | 開発サーバーOSセットアップ | | 終了 | markus | |
| 6 | タスク | 開発クライアントセットアップ | | 終了 | rick | |
| 7 | ユーザー ストーリー | ログイン済閲覧者は、フォーラムで製品に関する情報 | | アクティブ | jennifer | |
| 8 | ユーザー ストーリー | ログイン済閲覧者は、製品の英語資料の日本語訳を | | アクティブ | jennifer | |
| 9 | ユーザー ストーリー | 未登録閲覧者は、自身のアカウントを登録することが | | アクティブ | jennifer | |
| 10 | ユーザー ストーリー | 未ログイン閲覧者は、IDとパスワードを指定してログ | | アクティブ | jennifer | |
| 11 | ユーザー ストーリー | サイト閲覧者は、製品の説明を閲覧できる | | アクティブ | jennifer | |
| 12 | タスク | 画面モックアップ作成 | | 終了 | ta | |
| 13 | タスク | 開発サーバーに原稿投入 | | アクティブ | marlene | |

図 3.52 イテレーション バックログ (スプリント バックログ) シート (Excel)



| ID | タイトル | 作業項目の種類 | 状態 | 理由 | 担当者 |
|----|-------------------|------------|-------|----|----------|
| 3 | サイト閲覧者はお気に入り登録できる | ユーザー ストーリー | アクティブ | 新規 | jennifer |

図 3.53 新しい作業項目の作成 (Excel)

3.3.3 スプリント 1 の開始

それでは、スプリントの開始です。最初のスプリントでは、ステークホルダーが注目する重要な機能を 1 つ実現することを目標にしました。

初日は、主に開発環境のセットアップと、開発手順の確認に費やされるでしょう。いくつ

かのタスクは、スプリント計画会議の後半ですでに意識合わせができていますので、開発チームのメンバーは、個々に確認を取りながら作業を始めました。

■ 日次スクラム ミーティング

スプリントを開始するにあたっては、毎日 13:00 からの 15 分間、会議室で**日次スクラム ミーティング**を実施することにしました。この時間は、経験的に開発チーム全員が集まりやすい時間だからです。日次スクラム ミーティングでは、開発チームのメンバー全員が以下の内容を順番に発表して確認します。

- 前回の日次スクラムから行ったタスク
- 次回の日次スクラムまでに行うタスク
- タスクを行う上での問題点

会議室には印刷可能なホワイトボードを置き、さらに、プロジェクターを配置して、ノート PC から TFS の登録状況を見ながら、日次スクラムの発表内容を確認できるようにしました。

メンバーのうち誰かが、次回の日次スクラムまでに行うタスクがない場合は、その場で次のタスク候補を探します。開発チームは、自分たちでスプリントの目標を達成させるためのタスクを導き出すことができるので、タスクの内容があいまいで作業できないということはありません。

日次スクラムの結果は、スクラム マスターとチームの各メンバーがすぐに TFS へ反映します。

■ [TFS 操作] タスクの着手

開発チームのメンバーがタスクに着手するには、イテレーションと担当者を設定し、作業時間を適宜設定します。

担当者に割り当てられたタスクはクエリを使って一覧できるので、日次スクラムで決まった内容はいつでも TFS で確認することができます。また、割り込みでタスクを割り当てられることがあるかもしれないので、随時、担当タスクを確認するようにしましょう。

- ① チーム エクスプローラーから「**イテレーション バックログ**」(スプリント バックログ)のクエリを開き、タスクの変更画面を開きます (図 3.54、図 3.55)。
- ② 担当者、作業時間を変更し、タスクが完了したら状態を「**終了**」に変更します。
- ③ Excel を使って複数タスクを一括で変更することもできます (図 3.56)。
- ④ 担当者に割り当てられたタスクを確認するには、「**担当タスク**」のクエリを開きます (図

3.57)。

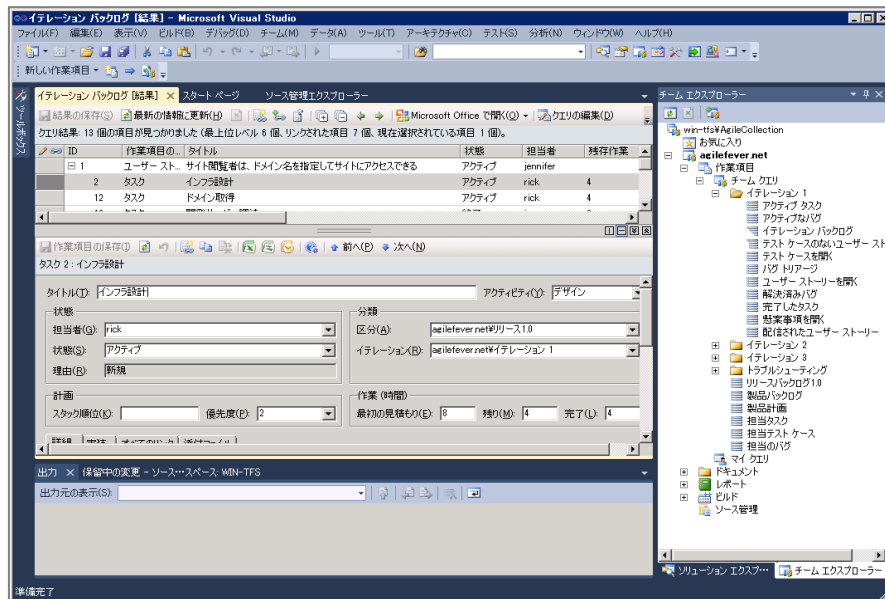


図 3.54 スプリント バックログ (Visual Studio)

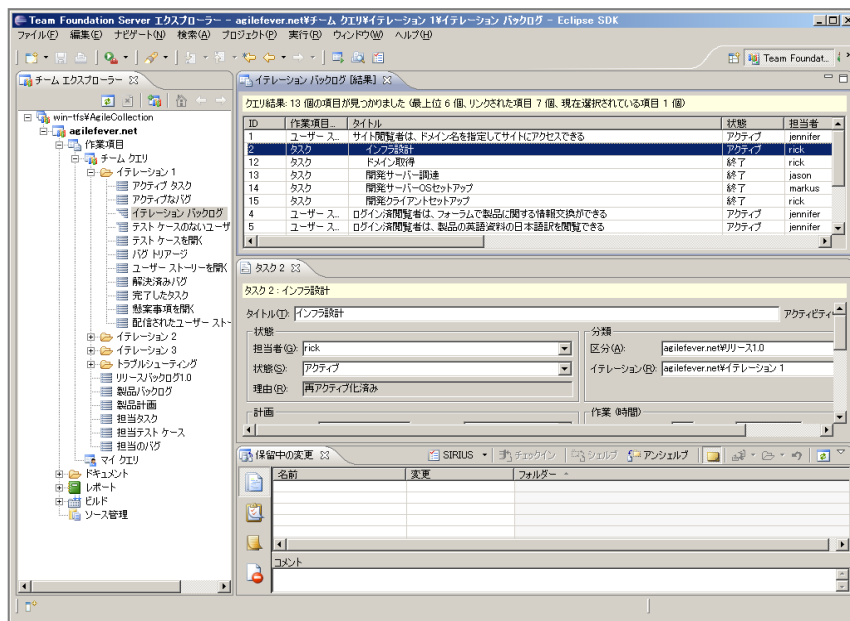


図 3.55 スプリント バックログ (Eclipse)

| ID | 作業項目の種類 | タイトル 1 | タイトル 2 | 状態 | 担当者 | 残存作業 |
|----|------------|----------------------------|--------|-------|----------|------|
| 1 | ユーザー ストーリー | サイト閲覧者は、ドメイン名を指定してサイトにアクセス | | アクティブ | jennifer | |
| 2 | タスク | インフラ設計 | | アクティブ | rick | |
| 12 | タスク | ドメイン取得 | | 終了 | rick | |
| 13 | タスク | 開発サーバー調達 | | 終了 | jason | |
| 14 | タスク | 開発サーバーOSセットアップ | | 終了 | markus | |
| 15 | タスク | 開発クライアントセットアップ | | 終了 | rick | |
| 4 | ユーザー ストーリー | ログイン済閲覧者は、フォーラムで製品に関する情報 | | アクティブ | jennifer | |
| 5 | ユーザー ストーリー | ログイン済閲覧者は、製品の英語資料の日本語訳を | | アクティブ | jennifer | |
| 6 | ユーザー ストーリー | 未登録閲覧者は、自身のアカウントを登録すること | | アクティブ | jennifer | |
| 7 | ユーザー ストーリー | 未ログイン閲覧者は、IDとパスワードを指定してログ | | アクティブ | jennifer | |
| 3 | ユーザー ストーリー | サイト閲覧者は、製品の説明を閲覧できる | | アクティブ | jennifer | |
| 16 | タスク | 画面モックアップ作成 | | 終了 | ta | |
| 17 | タスク | 開発サーバーに原稿投入 | | アクティブ | marlene | |

図 3.56 スプリント バックログ (Excel)

| ID | スタック順位 | 優先度 | 状態 | 担当作業 | 実務作業 | タイトル |
|----|--------|-----|-------|------|------|--------|
| 2 | 2 | 2 | アクティブ | 80 | 00 | インフラ設計 |

図 3.57 担当者タスク (Eclipse)

■ [TFS 操作] 進捗状況の確認

作業の進捗状況を確認するには、TFS のレポートを使うと便利です。レポートを用いて次のような項目を確認し、作業配分の見直しなどを実施します。

- 各ストーリーの残存作業量は予定どおりか
- 優先度の高いストーリーが先に着手されているか
- 各ストーリーにいくつテストが定義されているか
- テストはいくつ合格しているか

● テスト ケースが定義されていないストーリーはどれか

- ① チーム エクスプローラーでプロジェクトの「レポート」を展開し、「プロジェクト管理」 - 「ストーリーの概要」を開きます (図 3.58)。

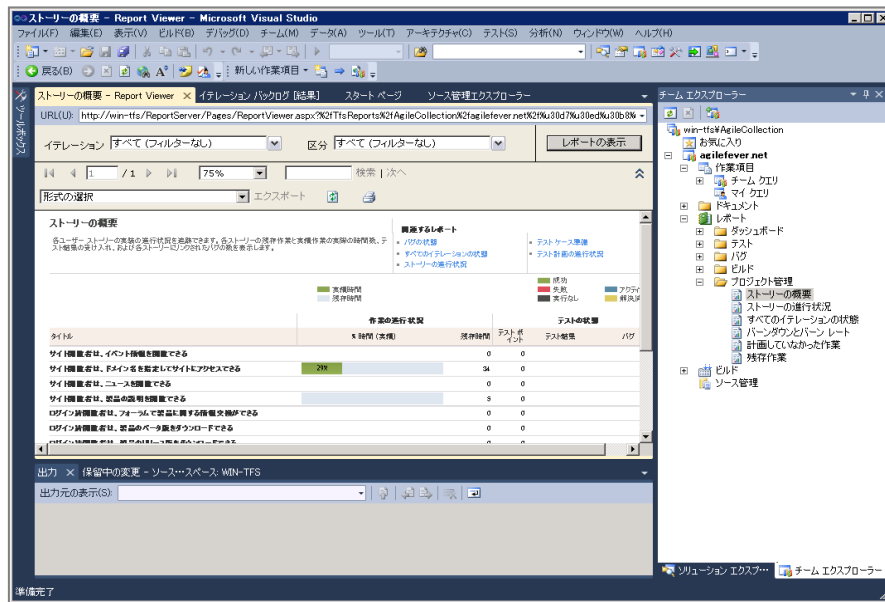


図 3.58 レポート : ストーリーの概要 (Visual Studio)

■ [TFS 操作] スプリント バーンダウン チャートの確認

日次スクラムでは、メンバーの担当タスクを確認するとともに、チームのスプリント バーンダウン チャートも確認することにしましょう。

バーンダウン チャートは、スプリント バックログ内の未完了タスクの残作業時間を日ごとに記録し、折れ線グラフにしたもので、これによって、作業が順調に消化されているか、スプリント終了日までに予定の作業が完了するかの見通しを立てることができます。バーンダウン チャートを確認して、作業消化が予定より遅く、スプリント終了日までに予定のタスクが完了できない場合は、早急に何らかの対策をとる必要があります。

このように、バーンダウン チャートはプロジェクトの進捗度合いを一目で把握し、対策が必要な場合にすばやく行動に移せるようにするためのツールです。

- ① チーム エクスプローラーのプロジェクトを右クリックし、プロジェクト ポータルを表示すれば、Web ブラウザでバーンダウン チャートを確認できます。「タスク バーンダウン」のリンクを開けば、バーンダウンの詳細なレポートが参照できます (図 3.59)。

- ② チーム エクスプローラーからは、「レポート」-「ダッシュボード」を展開し、「バーンダウン」をクリックすれば、バーンダウンのレポート画面が開きます（図 3.60）。

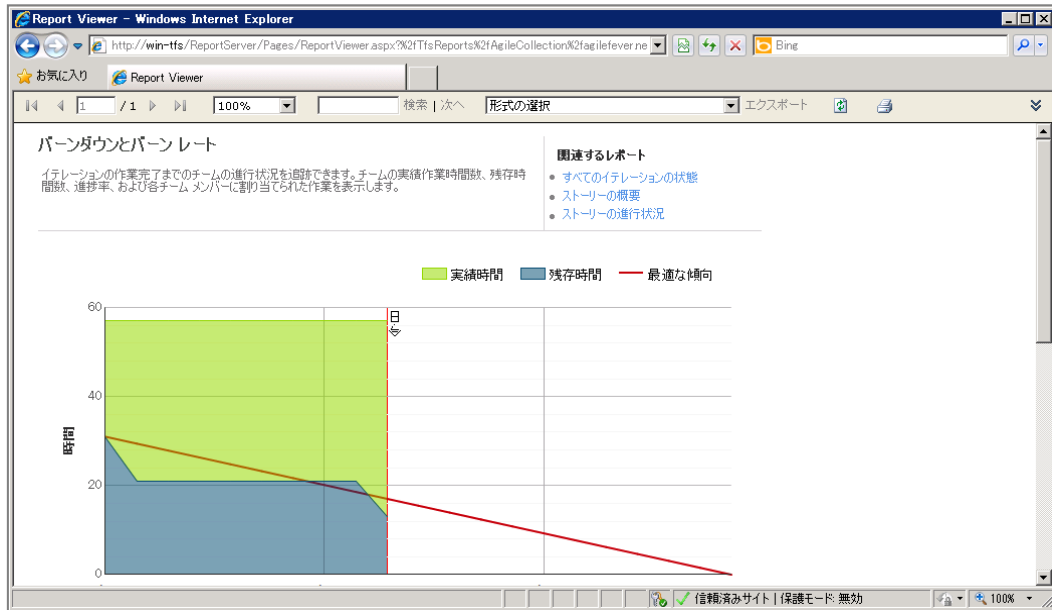


図 3.59 バーンダウン チャートの表示 (Internet Explorer)

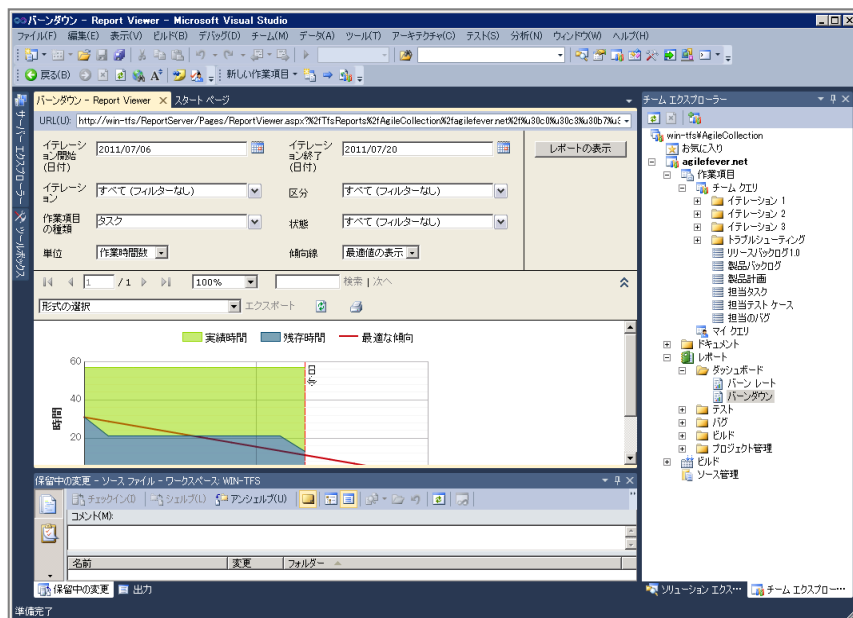


図 3.60 バーンダウン チャートの表示 (Visual Studio)

3.3.4 バージョン管理とビルド管理

スプリントの計画にしたがって開発を進めていきますが、実作業ではソースコードの管理

も重要になります。アジャイル開発では**ソースコードの共同所有**として、開発チームのメンバーがいつでもソースコードを参照、編集できるようにします。

■ バージョン管理

ソースコードは毎日繰り返し修正され、また、チームでの開発では複数の修正が同時並行的に行われます。したがって、最新のソースコードや、変更内容がメンバー間で共有されなければいけません。

バージョン管理では、リポジトリと呼ばれるデータベースでソースコードを管理します。単なるファイルの変更内容だけでなく、変更に関する一連の情報をひとまとめ（変更セット）にして変更履歴を記録することで、変更に関する情報や、これまでの経緯、過去のある時点でのソースコード一式などを、メンバー全員が参照できます。なお、ソースコードをリポジトリから取得することをチェックアウト、変更をリポジトリへ記録することをチェックイン（コミット）といいます。

あなたは、さっそくプロジェクト用のリポジトリを作成し、メンバーのそれぞれの開発環境（Visual Studio、Eclipse）にチェックアウトさせ、開発をはじめる準備をさせました。メンバーにはその日の変更を毎日チェックインするようにし、かつ、すべての単体テストにパスする状態を維持するようルールを決めました。

■ [TFS 操作] リポジトリへチェックイン

リポジトリにソースコードをチェックインします。

- ① Visual Studio のソリューション エクスプローラーで、対象とするプロジェクトやファイルを右クリックし、「**チェックイン**」をクリックすると、チェックイン ダイアログが開きます（図 3.61）。
- ② チェックイン ダイアログで変更に対するコメントや、対応付ける作業項目を選択して「**チェックイン**」ボタンをクリックすればチェックイン完了です。作業項目に関連付けて「**解決**」とした場合は、作業項目のクローズも同時に行われます（図 3.62）。
- ③ ダイアログ以外に「**保留中の変更**」ビューからチェックインすることもできます（図 3.63）。

Eclipse の場合は、初めてリポジトリにアクセスするときに、プロジェクトの共用設定を行います。

- ① Eclipse のパッケージ エクスプローラーで、対象のプロジェクトを右クリックし、「チーム」を選択します。初めての場合は「プロジェクトの共有」を選択し、「リポジトリの共有」ダイアログが開いたら、リポジトリの種別を「Team Foundation Server」に設定します (図 3.64)。
- ② 2 回目以降は、「チーム」にリポジトリ操作のメニューが表示され、「保留中の変更をチェックイン」を選択するとチェックイン ダイアログが開くので、Visual Studio と同じように項目を入力してチェックインします。
- ③ Visual Studio と同様、「保留中の変更」ビューからチェックインすることもできます (図 3.65)。

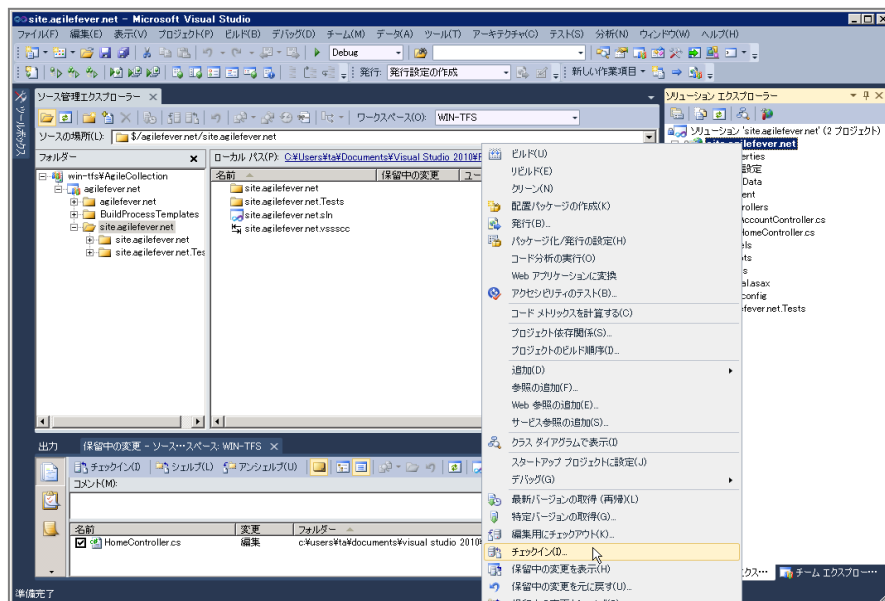


図 3.61 ソリューション エクスプローラーからチェックイン (Visual Studio)

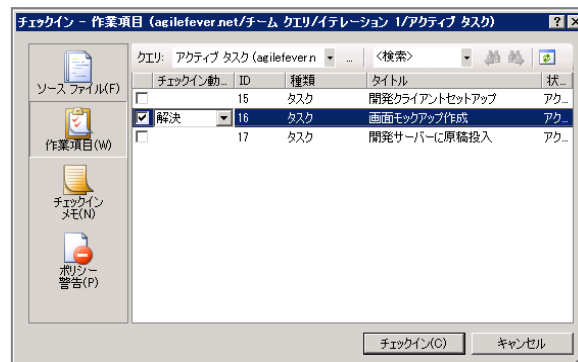


図 3.62 チェックイン ダイアログ (Visual Studio)

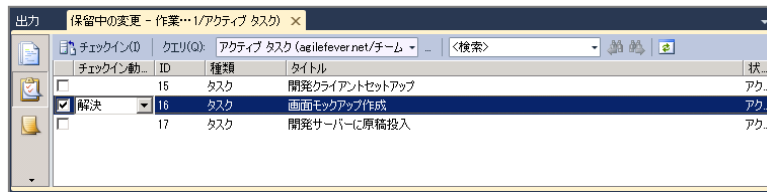


図 3.63 保留中の変更をチェックイン (Visual Studio)

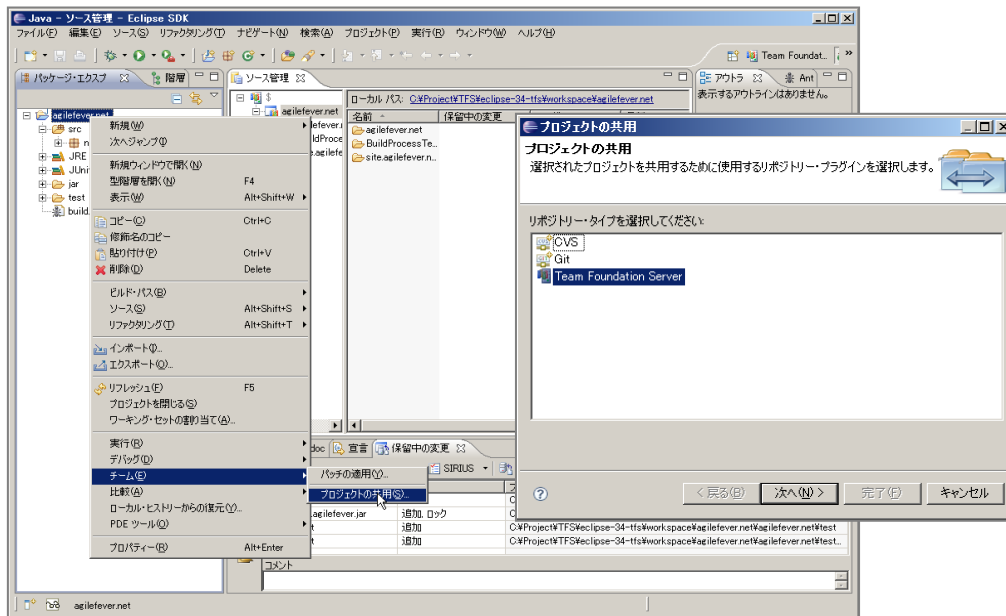


図 3.64 プロジェクトの共用 (Eclipse)

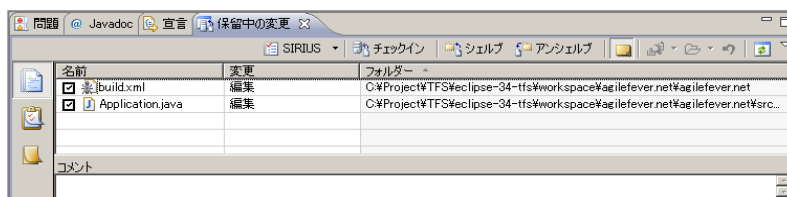


図 3.65 保留中の変更をチェックイン (Eclipse)

■ [TFS 操作] リポジトリからチェックアウト

リポジトリからチェックアウトします。

- ① Visual Studio のソリューション エクスプローラーで、対象となるプロジェクトやファイルを右クリックし、「編集用にチェックアウト」を選択すると、チェックアウト ダイアログが開きます (図 3.66)。
- ② チェックアウトの対象やロック種別を選択して「チェックアウト」ボタンをクリックす

ると完了です (図 3.67)。

- ③ Eclipse の場合は、パッケージ エクスプローラーの対象となるプロジェクトを右クリックし、「編集用にチェックアウト」を選択すると (図 3.68)、Visual Studio と同様チェックアウト ダイアログが開きます (図 3.69)。

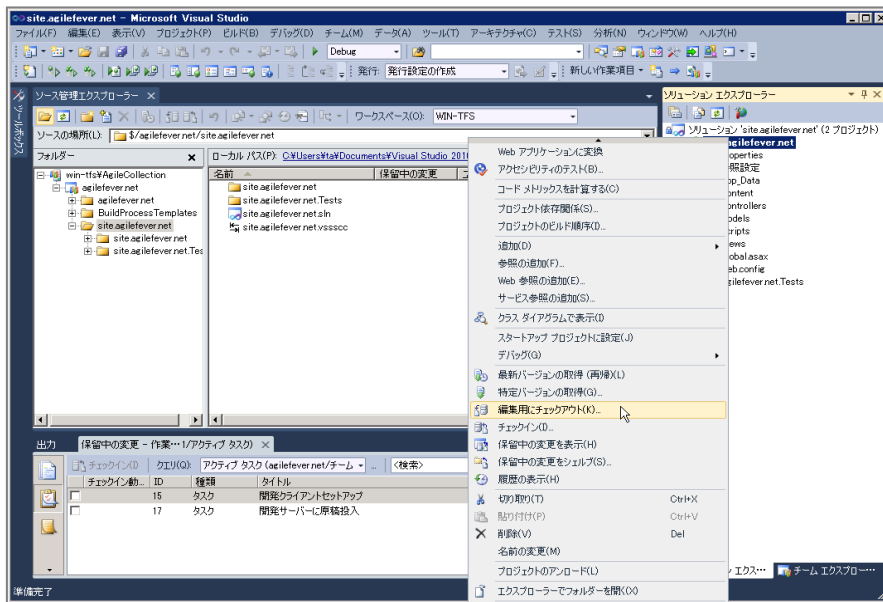


図 3.66 編集用にチェックアウト (Visual Studio)

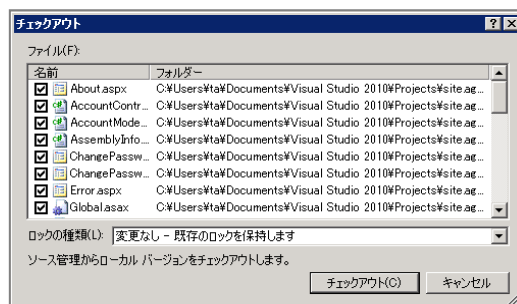


図 3.67 チェックアウト ダイアログ (Visual Studio)

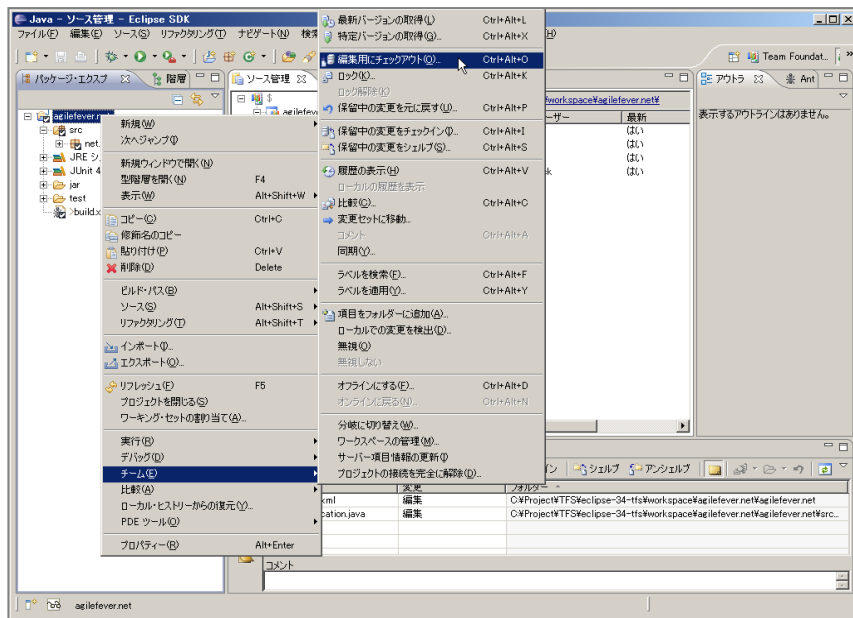


図 3.68 編集用にチェックアウト (Eclipse)

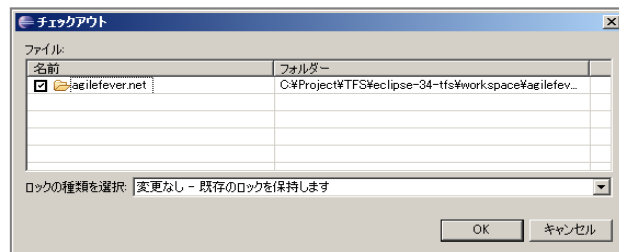


図 3.69 チェックアウト ダイアログ (Eclipse)

■ ビルド管理

開発の早い段階から絶えずビルドを実施し、結合テストを実施することで、リリース段階での想定外の障害を防ぐことができます。アジャイル開発では、これを**継続的インテグレーション**と呼んでいます。

今回のプロジェクトでは、プロジェクトのリポジトリから最新のソースコードを取得し、毎日、自動でビルドを実施することにしました。機能が追加されたらビルドとともにテストも実施し、継続的インテグレーションへ発展させる予定です。

■ [TFS 操作] ビルド サービスの構築

自動ビルドを行うには、TFS ビルド サービスを構成します。ビルド サービスは TFS 管理コンソールから設定します。

- ① TFS 管理コンソールを起動して「ビルド構成」を選択し、「インストール済みのフィッチャーの構成」をクリックすると、構成センターの画面が開きます（図 3.70）。
- ② 構成センターで「ウィザードの開始」ボタンをクリックすると「ビルド サービス構成ウィザード」が開始されるので、画面にしたがって設定を進めます（図 3.71）。
- ③ プロジェクトコレクションの選択では「AgileCollection」を選択します（図 3.72）サービスの構成、コンピューターの構成を設定し（図 3.73）、準備チェックを経て構成を完了します（図 3.74）。



図 3.70 構成センター (TFS 管理コンソール)

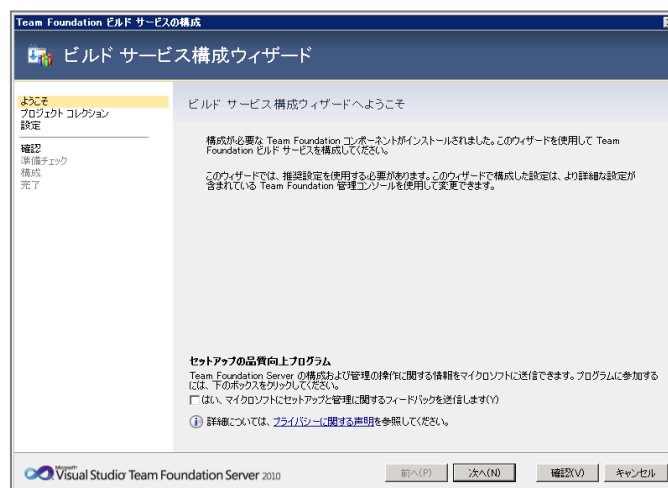


図 3.71 ビルド サービス構成ウィザード (TFS 管理コンソール)

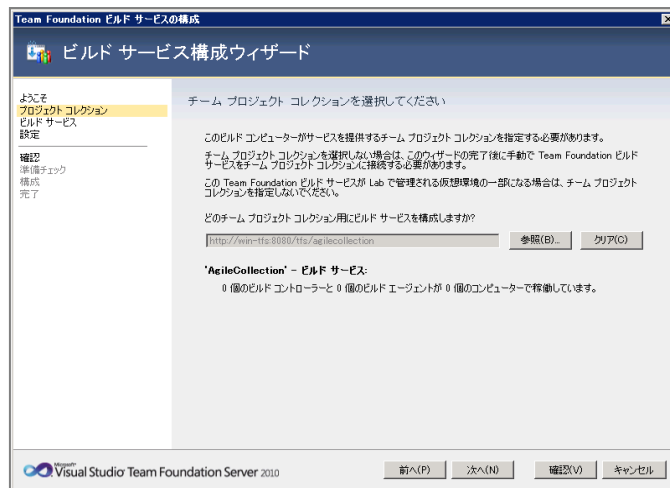


図 3.72 チーム プロジェクト コレクションの選択 (TFS 管理コンソール)

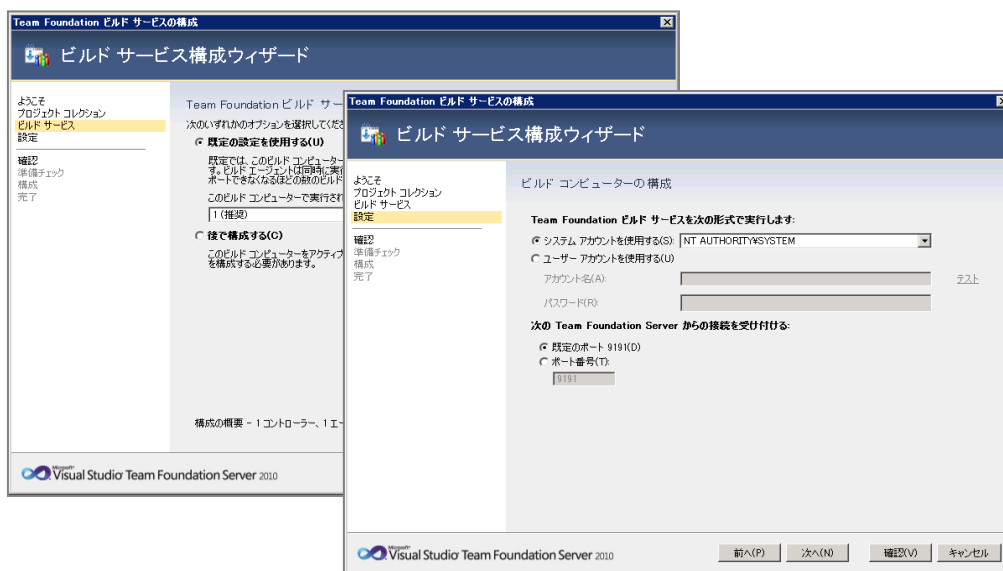


図 3.73 ビルドの構成、およびコンピューターの構成 (TFS 管理コンソール)

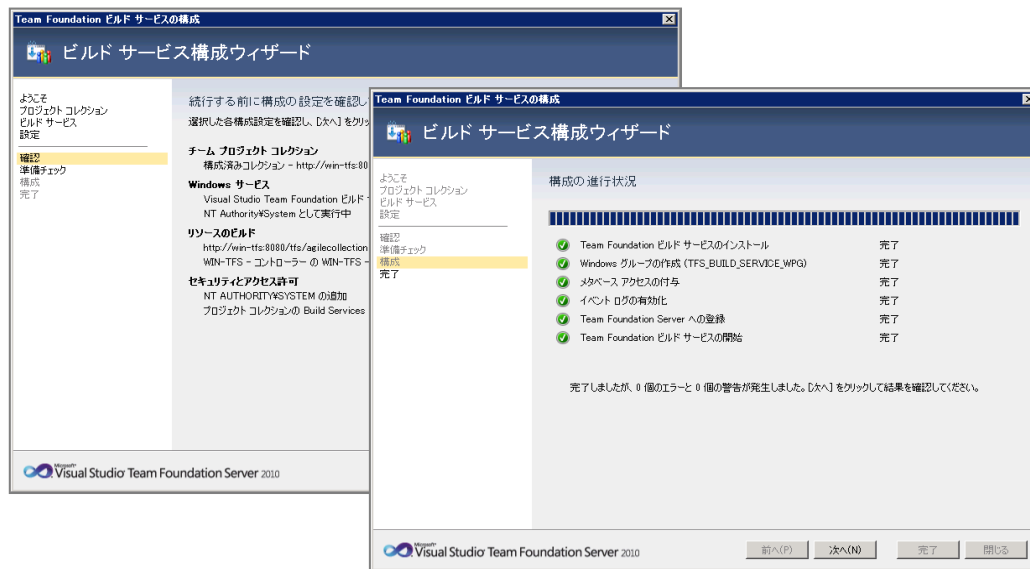


図 3.74 構成の確認、およびビルド サービスの構成 (TFS 管理コンソール)

■ [TFS 操作] ビルド定義を作成

ビルド サービスにビルド方法やスケジュールを指定したビルド定義を登録することで、自動ビルドを実施します。日次ビルドする前に、まず、ビルド方法を定義して手動でビルドを実行し、問題ないようならスケジュールを設定して日次ビルド化するといいいでしょう。

- ① チーム エクスプローラーのプロジェクトを展開し、「ビルド」を右クリックして「ビルド定義の新規作成」を選択すると、ビルド定義の設定画面が開きます (図 3.75)。
- ② ビルド定義の名前を設定し (図 3.76)、トリガー設定でひとまず「手動」を選択しておきます (図 3.77) ビルドの規定値設定画面では、ビルドの出力先となる共有フォルダパスを指定し、プロセス設定画面でビルドの実施内容を定義します。「Ctrl-S」で内容を保存します (図 3.78)。
- ③ チーム エクスプローラーに追加された新しいビルド定義を右クリックし、「新しいビルドをキューに配置」を選択すると、ビルド配置のダイアログが開きます (図 3.79)。
- ④ 「キューに登録」ボタンをクリックすると、ビルド エクスプローラーの「キューに挿入済み」タブにビルド定義が追加され、自動ビルドが実施されます (図 3.80)。
- ⑤ 自動ビルドが完了すると、「完了」タブに表示され、テストの成否などを確認できます (図 3.81)。

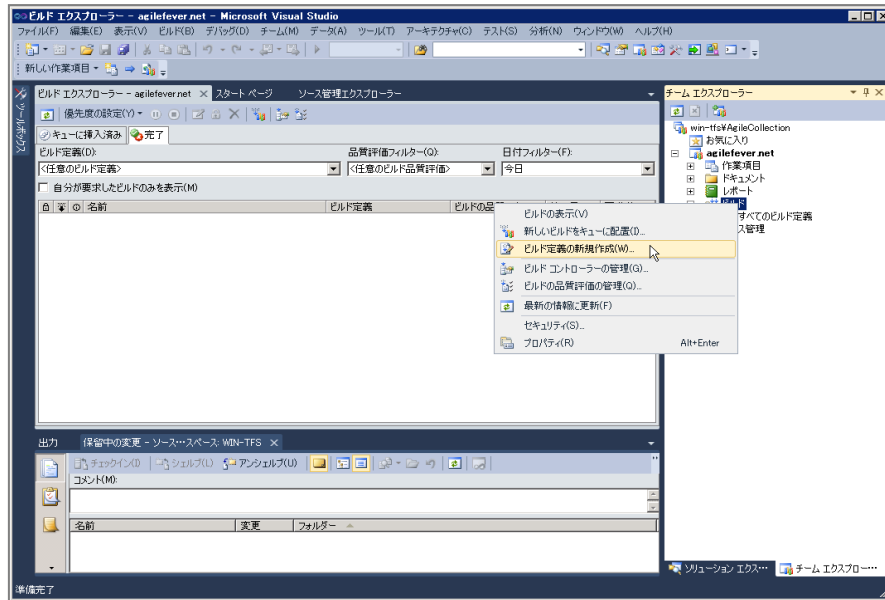


図 3.75 ビルド定義の新規作成 (Visual Studio)

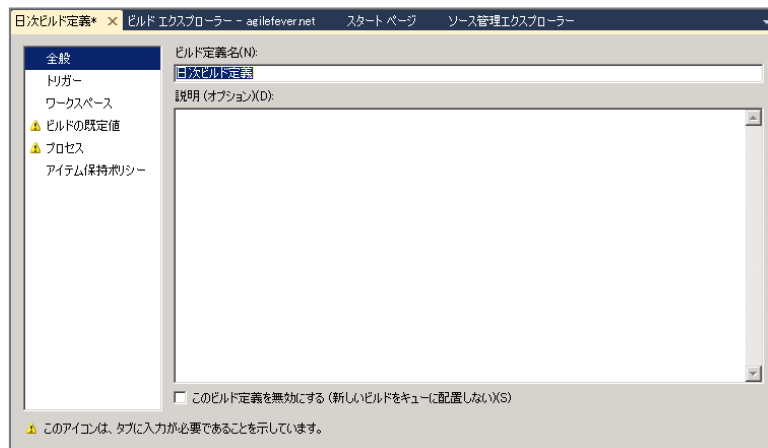


図 3.76 ビルド定義：全般 (Visual Studio)

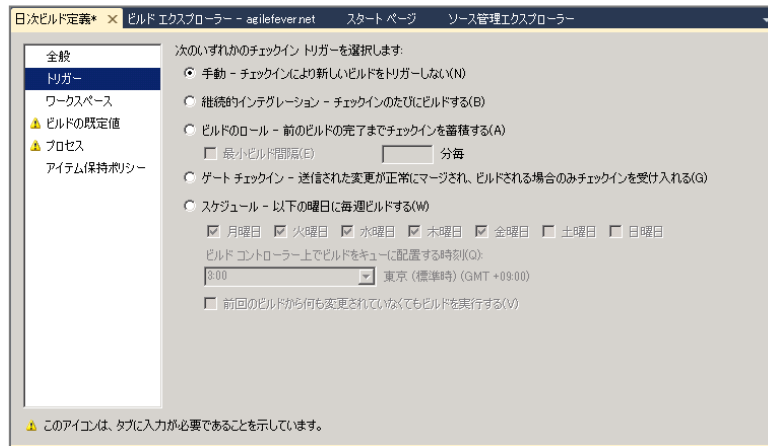


図 3.77 ビルド定義：トリガー (Visual Studio)

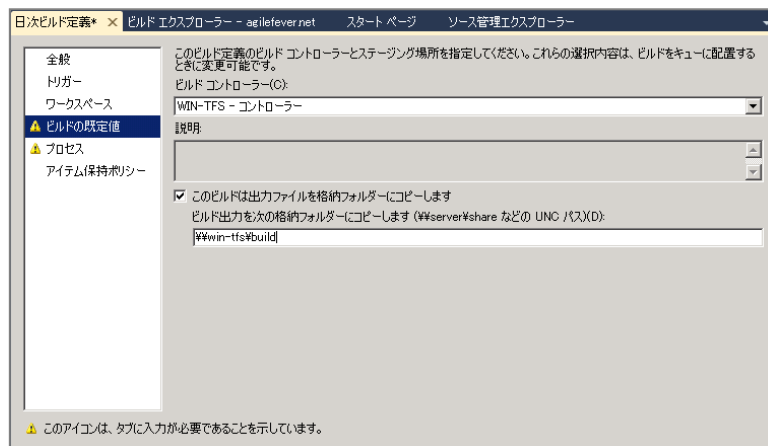


図 3.78 ビルド定義：既定値 (Visual Studio)

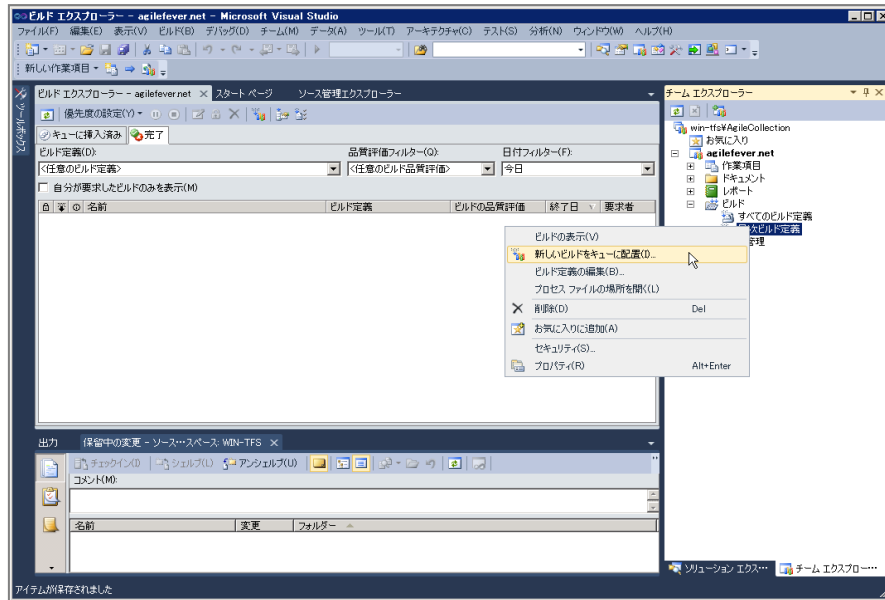


図 3.79 新しいビルドをキューに配置 (Visual Studio)

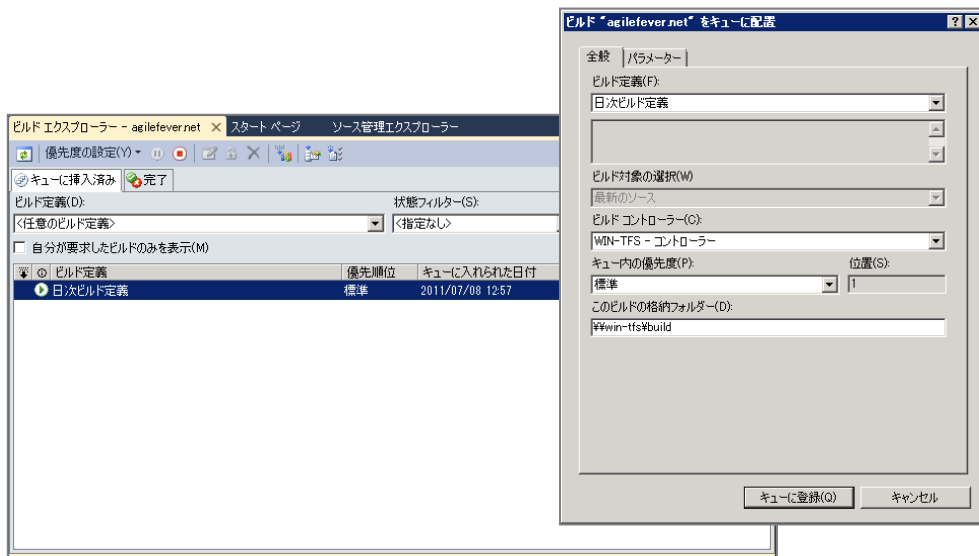


図 3.80 ビルド定義をキューへ配置 (Visual Studio)

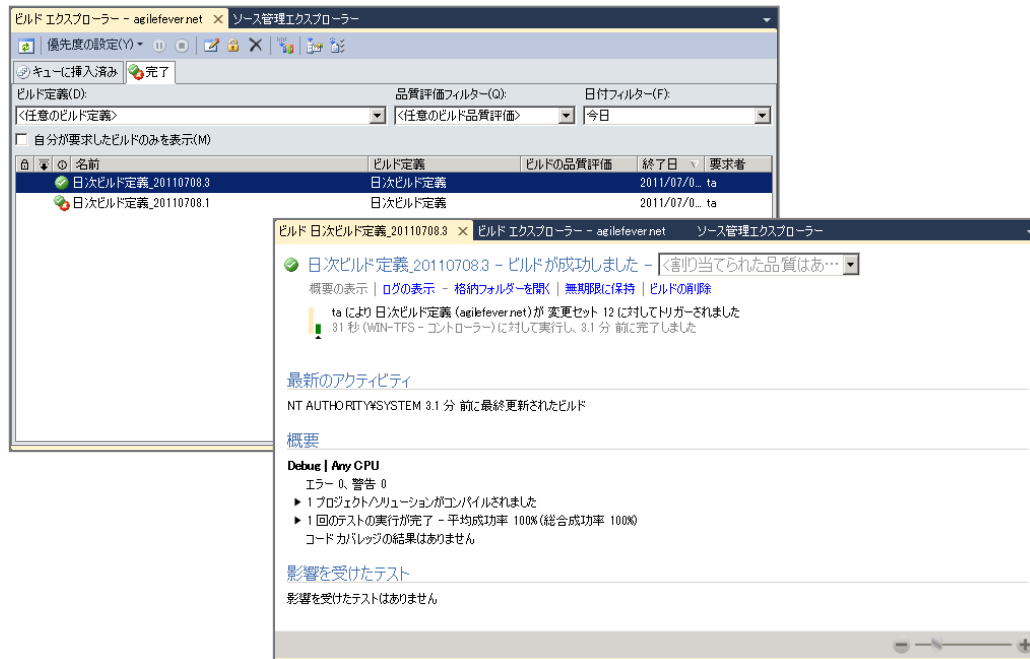


図 3.81 完了済みのビルド (Visual Studio)

■ [TFS 操作] 日次ビルドのスケジューリング

ビルド定義を手動で実施し、問題なければ自動ビルドのスケジュールを設定し、日次ビルド化します。

ここでは、日次ビルドを取り上げますが、ビルドのトリガーの選択によってさまざまなスケジューリングを実施することが可能です。「**継続的インテグレーション**」を選択すれば、チェックイン時に自動で、あらかじめ設定したコード分析やテストをビルドサーバー上で自動実行し、テスト結果やコードのカバレッジ、テストの影響範囲の記録などを行います。また、「**ゲート チェックイン**」を選択すれば、チェックイン操作時に即時にチェックインせず、継続的インテグレーションと同様、ビルドやコード分析、テストを実行し、問題がない場合に自動でチェックインし、問題がある場合はただちに開発者へ通知し、修正を促すことができます。これにより、リポジトリへの汚れたコードの混在を未然に防ぐことが可能になり、コードの共同所有をよりの確に実施することができるようになります。

- ① チーム エクスプローラーの対象となるビルド定義を右クリックし、「**ビルド定義の編集**」を選択すると、ビルド定義の設定画面が開きます (図 3.82)。
- ② トリガー設定画面で「**スケジュール**」にチェックを入れ、日次ビルドの曜日と時間を指定します (図 3.83)。
- ③ ビルド定義をキューに登録すると、指定の日時に自動ビルドが実施されます。

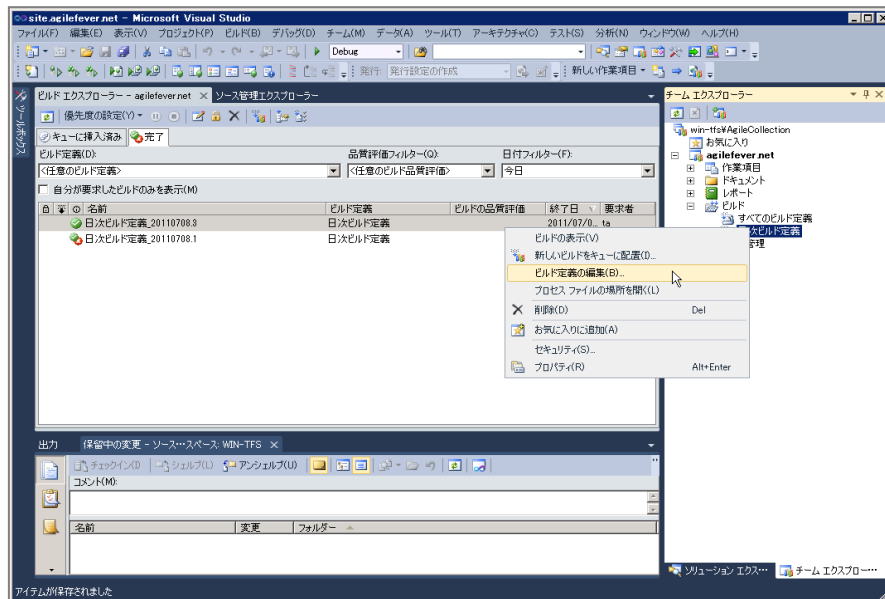


図 3.82 ビルド定義の編集 (Visual Studio)

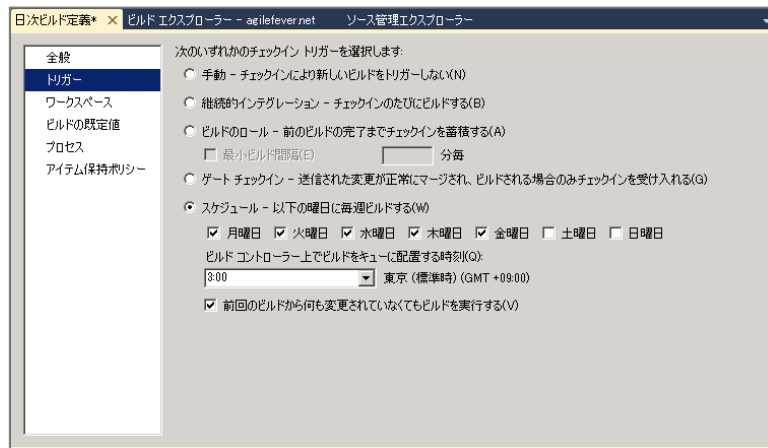


図 3.83 自動ビルドのスケジュール設定 (Visual Studio)

3.3.5 スプリント1の終了とスプリント2の準備

最初のスプリントは大変ですが、開発チームはなんとかスプリント1を乗り切ることができました。メンバーたちは少しでも早く**スプリント レビュー会議**の場で、ステークホルダーのフィードバックを得たいと思っていました。作り上げた機能を評価して欲しいという気持ちと、機能の不明点を解消したいという気持ちであふれていたからです。普段はプロダクト オーナーとは密接にコミュニケーションをとっていましたが、ステークホルダーの何人かからは初めてフィードバックを得る機会となります。

■ スプリント レビュー会議

あなたは、スプリント計画会議と同じように、自社の会議室でスプリント レビュー会議を始めます。スプリント レビュー会議では、製品バックログを参照しながら、個々のストーリーのデモを実施します。必要であれば、インフラ構築に関する設計書などのドキュメントも用意します。

今回のスプリントでは、これほど早く動くソフトウェアを見ることができると思っていなかったコミュニティ メンバーを驚かせ、開発チームのモチベーションの高さと、技術力に対する信頼を得ることができました。一部の機能で、コミュニティ メンバーの想像と異なるものがありましたが、それも早い段階で意識合わせできたとあって、逆に喜ばれるという結果となりました。

■ 反省会ミーティング（ふりかえり）

スプリント レビュー会議が終わると、次のスプリント計画会議までの間に少し日が空いていたので、反省会を実施しました。あなたは開発メンバーを会議室に集め、スプリント 1 について、次の 3 つの内容を自由に発言してもらいました。

- ① **Keep** : 次のスプリントでも続けた方がいいと思うこと
- ② **Problem** : 問題だと思ったこと
- ③ **Try** : 次のスプリントでやってみること

反省会ミーティングの結果、ホワイトボードは次のようになりました。

| ○月×日「Agile Fever (仮) コミュニティ サイト構築」反省会ミーティング | |
|--|---|
| Keep <ul style="list-style-type: none"> ・ テスト駆動開発 ・ 継続的な統合 ・ コーディング規約の厳守 | Try <ul style="list-style-type: none"> ・ 日次スクラムの開始を 13:00 から 13:30 に変更する ・ 遅刻したら罰金 100 円にする ・ 罰金をプールし、ペアプログラミング用のお菓子にあてる ・ ドキュメントのルールを作る |
| Problem <ul style="list-style-type: none"> ・ 日次スクラムに遅刻してくる人がいる ・ ドキュメントの様式が統一されていない ・ 日次スクラムで問題を言わない人がいる | |

問題と思うことは日次スクラムでも随時あげられていたので、ここではスプリント後半で発生した問題や、すぐには解決できなかった問題があげられました。

反省会ミーティングが終わったら、その結果を **TFS** へ反映して次のスプリントに向けて準備します。

反省会ミーティングの Wiki

反省会ミーティングの結果を Wiki に登録します。

ふりかえり項目の追加

反省会ミーティングであがった項目のうち、タスクとして扱えるものをスプリント バックログへ登録します。

障害の登録

発見されたバグを障害としてスプリント バックログに登録します。

■ [TFS 操作] 反省会ミーティングの Wiki

反省会ミーティングの結果は、TFS のチーム Wiki に記録しておくとい良いでしょう。

- ① Web ブラウザでプロジェクト ポータルを開き、「**チーム Wiki**」をクリックします。
- ② 「**新規**」-「**新しい Wiki ページ**」をクリックすると新しいページの編集画面が開きます (図 3.84)。
- ③ 「**OK**」をクリックすると Wiki ページが保存されます (図 3.85)。

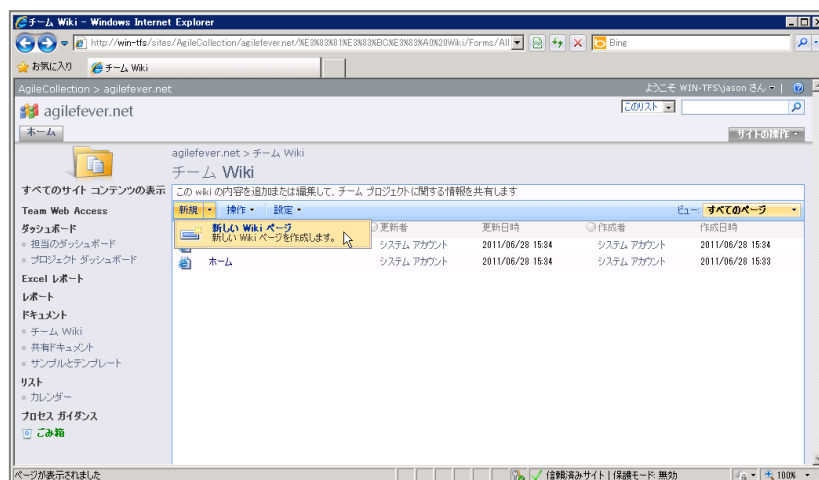


図 3.84 チーム Wiki (Internet Explorer)

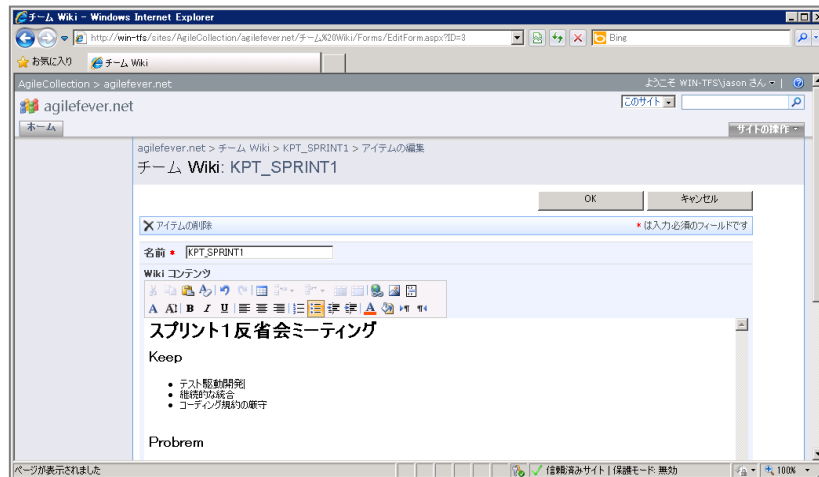


図 3.85 新しい Wiki ページの編集 (Internet Explorer)

■ [TFS 操作] ふりかえり項目の追加

反省会ミーティングであがった項目のうち、スプリント バックログのタスクとして扱えるものを TFS へ登録します。とくに「Try」の項目はできるだけタスク形式に落とし込んでおくと、具体的に作業化されるでしょう。

ふりかえり項目は次のスプリントの「スプリント2」のタスクとして登録します。あらかじめ、TFS に「スプリント2」のイテレーションを用意しておきましょう。

■ スプリント2の準備

「スプリント1」の「スプリント レビュー会議」の結果を受けて、「スプリント2」の「スプリント計画会議」を実施します。製品バックログには、「スプリント1」開始時から存在したもの、「スプリント1」の最中に追加されたもの、「スプリント計画会議」で新たに追加されるものがあります。ストーリーやタスクの見積りは、「スプリント」を重ねるにつれて、より精度の高いものになるでしょう。

開発チームのメンバーは、新たな目標に向かって全力で進んでいきます。

■ [TFS 操作] 障害の登録

「スプリント2」以降の「スプリント」では、以前の「スプリント」で開発した機能に発見されたバグも作業項目として登録しておきます。作業項目は元のストーリーに属するタスクとして作成するか、新たに障害をストーリー化してリリース バックログに含めてもいいでしょう。

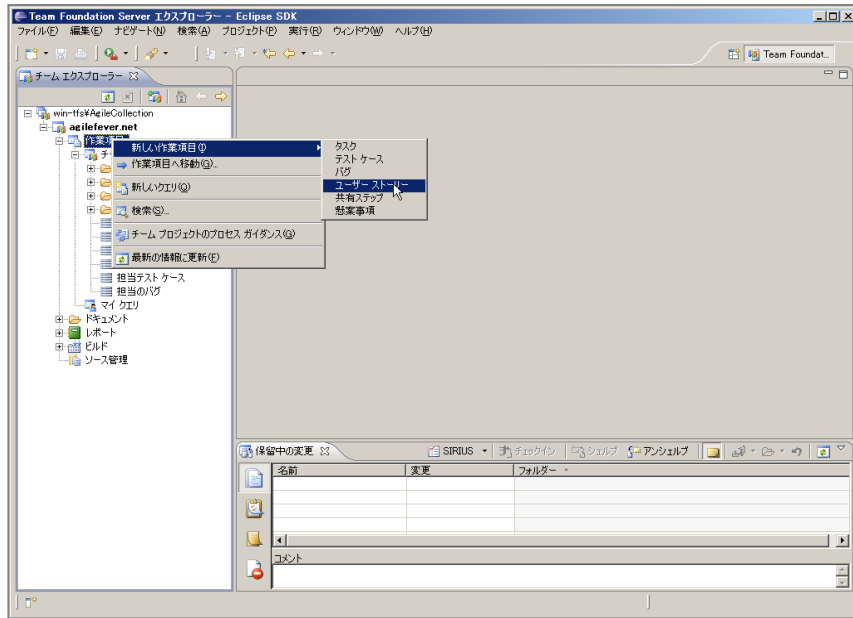


図 3.86 障害のストーリー作成 (Eclipse)

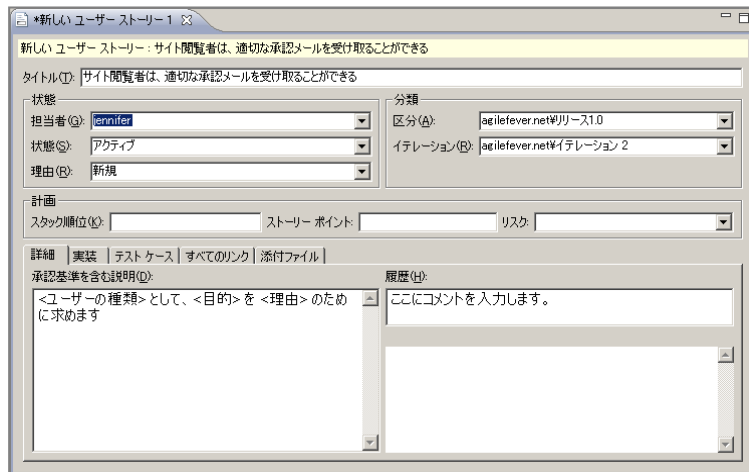


図 3.87 障害のユーザー ストーリー (Eclipse)

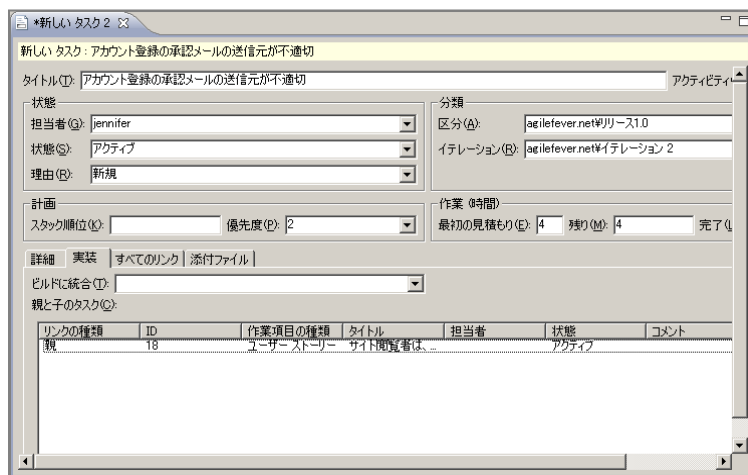


図 3.88 障害のタスク (Eclipse)

3.3.6 スプリント 3 からリリースへ

スプリント 3 はリリース直前のスプリントとなるため、新しい機能を追加するのではなく、結合テストや障害対応、また、製品の品質をアップさせるためのリファクタリングなどに作業をあてます。スプリント計画会議の段階で、リリース直前まで機能追加をすることがないように、あらかじめペースを考えておくといいいでしょう。

■ 受け入れテスト

リリースでは、リリース バックログにまとめられたユーザー ストーリーが実現できていることが重要になります。アジャイル開発では、ユーザー ストーリーが正しく実現されているかを、顧客（プロダクト オーナー）の立場でテストします。これは**受け入れテスト**と呼ばれ、リリースやスプリント（イテレーション）毎に実施します。

開発チームのメンバーたちは、ユーザー ストーリー毎に、ユーザー視点の**テスト ケース**を作成し、これにしたがってテストを実施することにしました。

■ [TFS 操作] テストの計画

Test Manager を使えば、テストの計画から実施、バグの管理まで、より本格的にテスト工程を管理することができます。Test Manager ではテスト スイートによってテスト ケースをグループ化、階層化することができ、テスト ケースに紐付けてテストを実施することができます。

また、要件（ユーザーストーリー）をそのままテスト スイートとして取り込むことができます。要件に対するテストが明確になり、**受け入れテスト駆動開発**の流れをスムーズに進めることができます。

- ① スタートメニューから「Microsoft Test Manager 2010」を起動します。初めて起動する場合は、TFS サーバーへの接続設定を行います (図 3.89、図 3.90)。
- ② 最初にテスト計画を作成します。テスト計画を選択してテスト センターを起動し、テスト スイートやテスト ケースなどを設定します (図 3.91、図 3.92)。
- ③ テスト計画のプロパティには、テスト対象のビルドや、テスト方法 (手動テストと自動テスト)、テスト環境、テスト構成、テスト期間を設定できます (図 3.93)。
- ④ テスト計画にテスト スイートを追加します。テスト スイートはフォルダのように、配下にテスト ケースやテスト スイートを作成することができます (図 3.94)。
- ⑤ 要件を取り込むには、テスト計画、もしくはテスト スイートを選択して「要件の追加」ボタンをクリックすると、要件の選択画面が開きます (図 3.95)。
- ⑥ 要件を選択して「要件を計画に追加」ボタンをクリックすると、要件がテスト スイートとして取り込まれます (図 3.96)。



図 3.89 TFS サーバーの選択 (Test Manager)



図 3.90 プロジェクトの選択 (Test Manager)

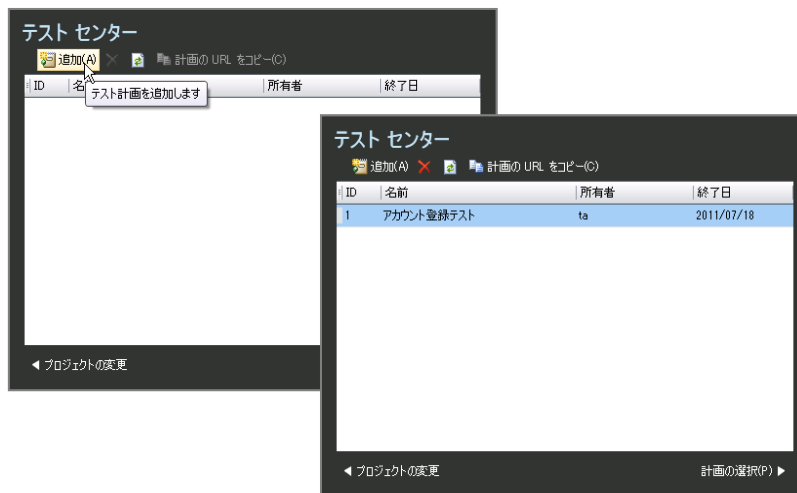


図 3.91 テスト計画の作成 (Test Manager)

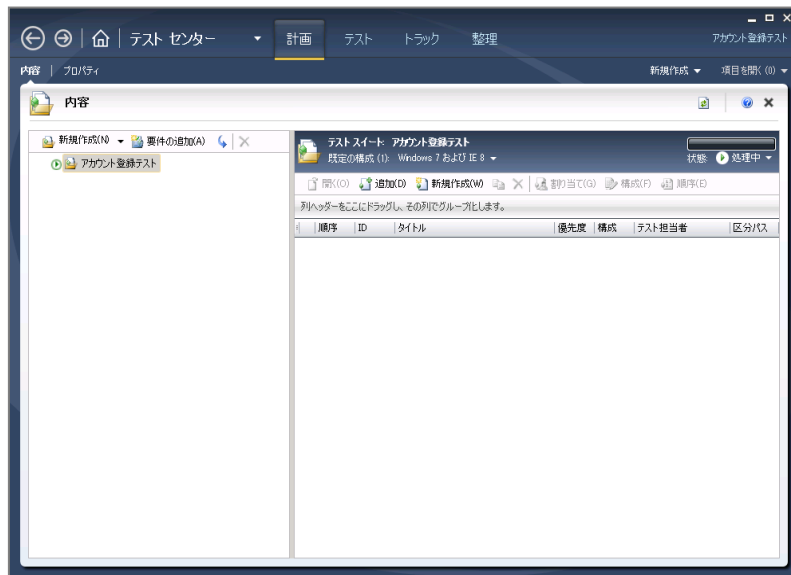


図 3.92 テスト計画の設定 (Test Manager)

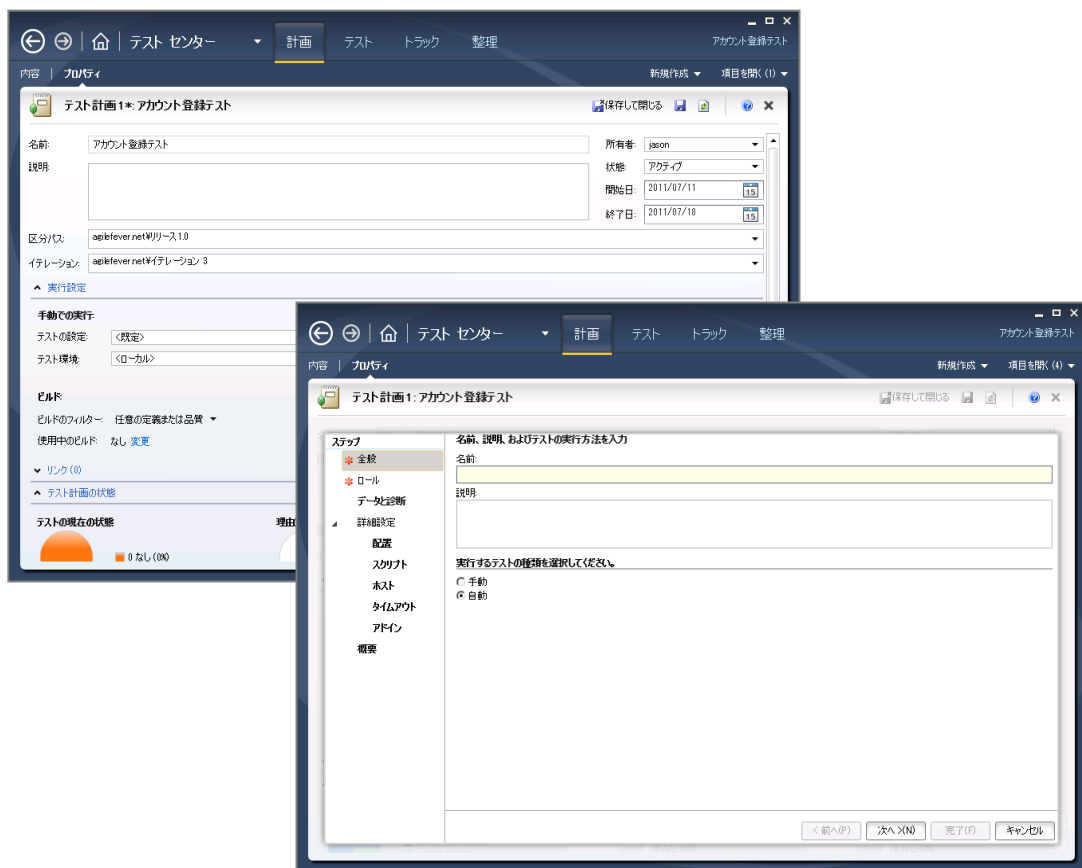


図 3.93 テスト計画のプロパティ (Test Manager)

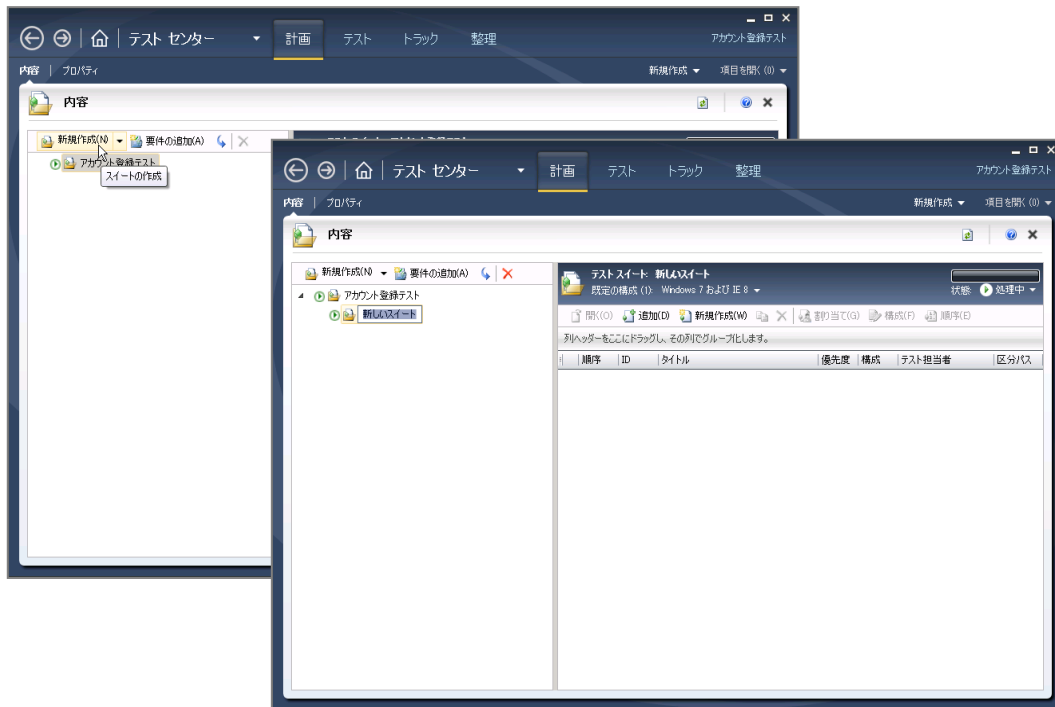


図 3.94 テスト スイートの追加 (Test Manager)

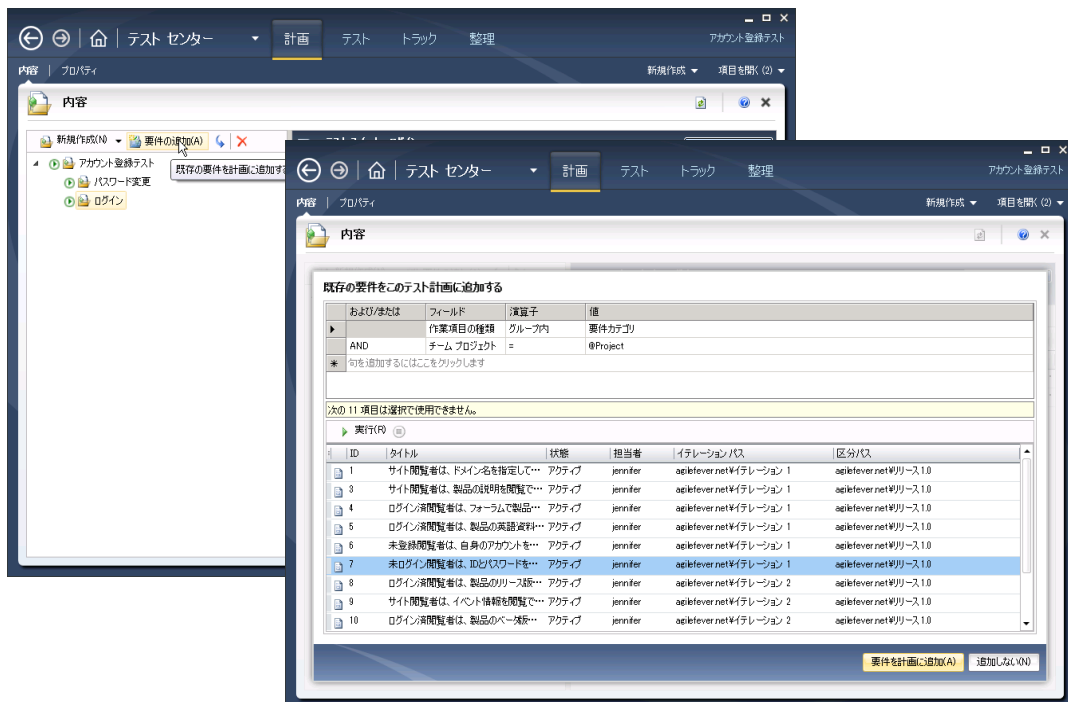


図 3.95 要件の選択 (Test Manager)

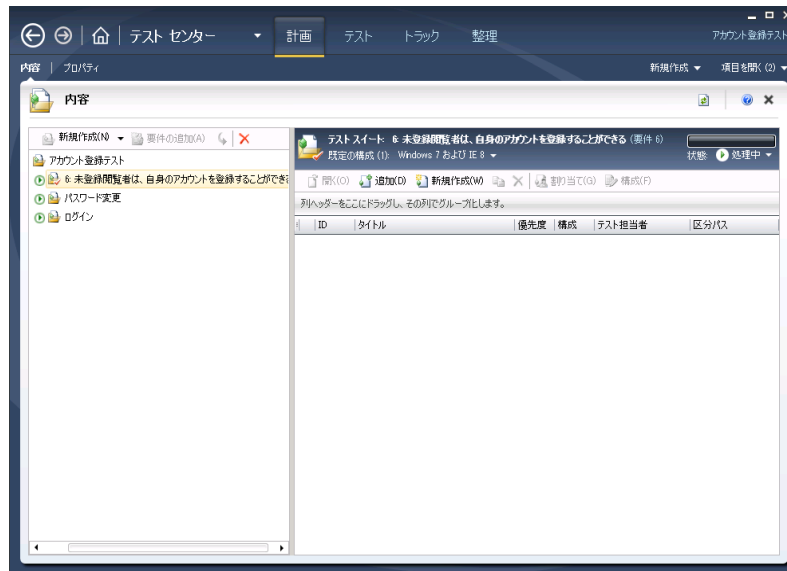
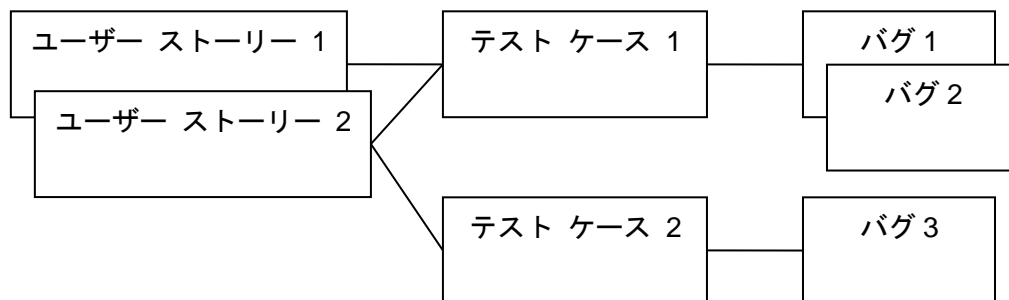


図 3.96 要件をテスト スイートとして追加 (Test Manager)

■ [TFS 操作] テスト ケースの作成

テストの計画で作成したテスト スイートにテスト ケースを追加します。テスト ケースはテスト項目に該当し、TFS の作業項目のひとつとして作成します。関連する作業項目はお互いにリンクすることができるので、テスト ケースはテスト対象のユーザー ストーリーやバグ (後述) などへリンクさせることで、障害のトレースが容易になります。また、テスト ケースを一覧するクエリを作成すれば、これがテスト項目一覧となります。



なお、Visual Studio や Eclipse のチーム エクスプローラーから、通常の作業項目の追加と同様に、テスト ケースを作成できますが、この場合、テストのステップを追加できないなどの制限があります。これらは主に開発中に作成する単体テストをテスト ケースとして誰でも自動実行できるようにする際に用いられます。

① Test Manager にて、テスト スイートにテスト ケースを追加します。テスト ケース

の追加先のテスト スイートを右クリックし、「新しいテスト ケース」を選択すると、テスト ケースの設定画面が開きます。設定画面では、タイトルや担当者、区分、イテレーションなどを入力します。さらに、「ステップ」タブでテストの手順を設定することができます (図 3.97)。

- ② 要件として登録したテスト スイートにテスト ケースを追加すると、テスト ケースの保存時に、自動で関連するユーザー ストーリーへのリンクが設定されます。個別にリンクを設定する場合は、「テストされたユーザー ストーリー」タブを開き、「リンク先」ボタンをクリックすると、リンクの追加ダイアログが開く (図 3.98) ので、テスト対象のストーリーを選ぶと、ストーリーへのリンクが設定されます (図 3.99)。
- ③ 「保存して閉じる」ボタンをクリックすると、テスト ケースが作成されます。

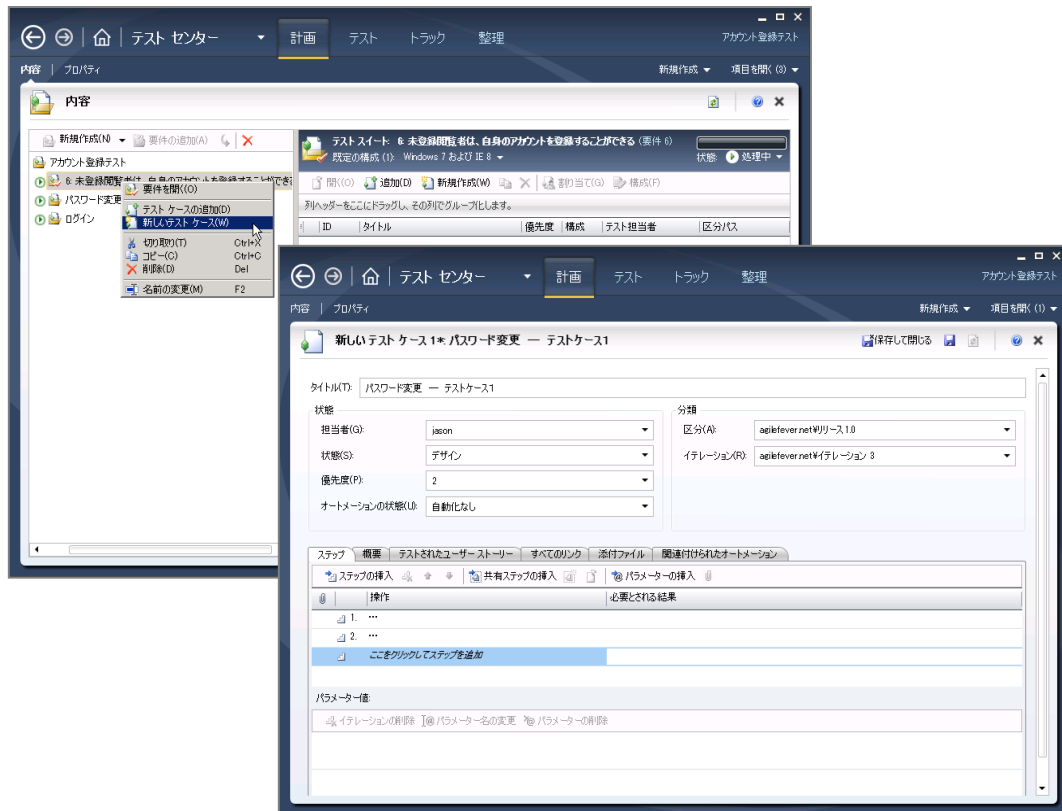


図 3.97 テスト ケースの追加 (Test Manager)

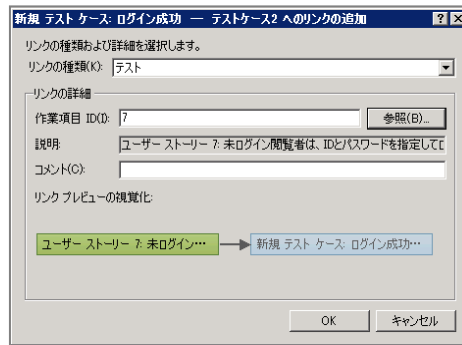


図 3.98 ユーザー ストーリーへのリンクの追加 (Test Manager)



図 3.99 ユーザー ストーリーへのリンク (Test Manager)

■ [TFS 操作] テストの実行

テストの計画で作成したテスト ケースを、テスト ランナーで実行することができます。テスト ランナーで実行した結果は、テスト ケースと関連づけて記録されます。また、テスト結果が NG の場合は、テスト結果に関連づけてバグを起票することができ、テストの計画、実行からバグ追跡という流れがスムーズに進められます。

- ① Test Manager にて、テスト センターの画面上部の「テスト」をクリックし、テスト ケースを選択して「実行」ボタンをクリックすると、テスト ランナーが起動してテストを実施します (図 3.100)。
- ② テストの実施後にテスト結果が記録されます。テストの実施手順を記録しておいて、あとでテストを自動化することもできます (図 3.101)。

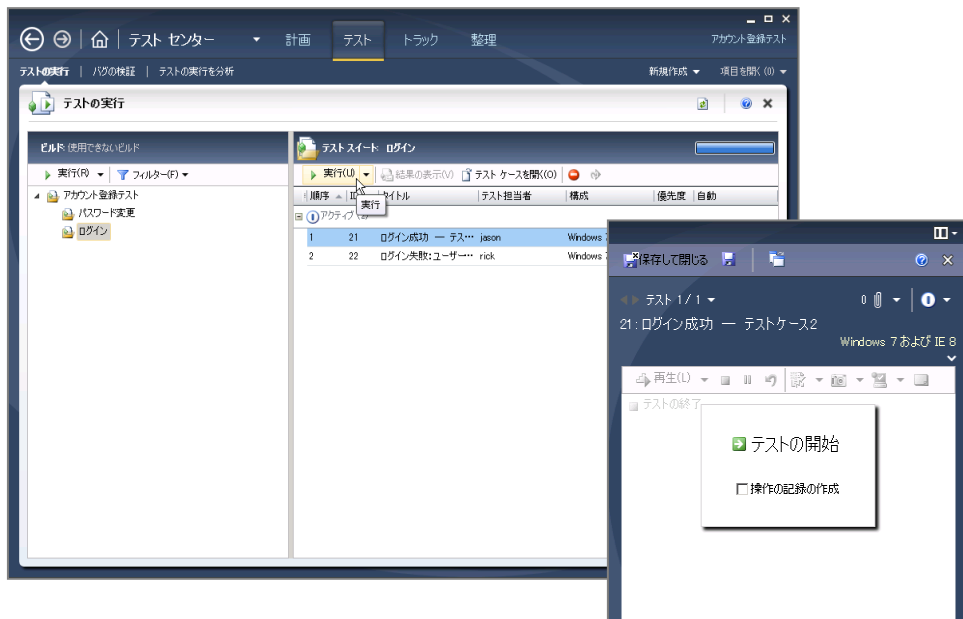


図 3.100 テスト ランナーによるテストの実施 (Test Manager)

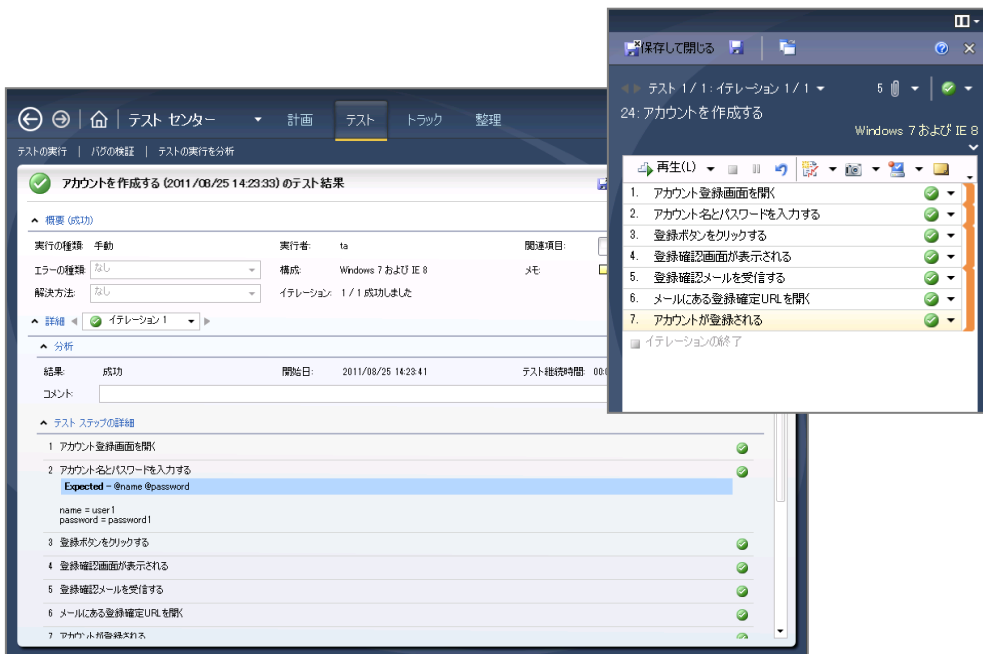


図 3.101 テスト結果 (Test Manager)

■ [TFS 操作] バグの管理

テストの実行結果が NG の場合、バグに関連する情報を作業項目のひとつとして登録することができます。Test Manager を使えば、ユーザー ストーリーからテスト ケース、ビ

ルド バージョン、テスト結果など、バグに関連する情報をすべて関連づけて記録できるので、開発者によるバグの追跡が容易になります。

なお、テスト ケースと同じく、Visual Studio や Eclipse のチーム エクスプローラーから、作業項目としてバグを登録することもできますが、こちらは主に開発中に発見したバグを記録するためのもので、受け入れテストなどのテスト工程では Test Manager を使うのが望ましいといえます。

- ① Test Manager にて、テスト結果画面の「バグの作成」ボタンをクリックすると (図 3.102)、新しいバグの作成画面が開きます (図 3.103)。
- ② テスト結果からバグを登録する場合は、区分やイテレーション、リンクするテスト ケースなどがあらかじめ設定されています (図 3.104) テスト計画のプロパティに、対象となるビルド バージョンを設定しておけば、自動でシステム情報に反映されます。
- ③ 個別にテスト対象を設定する場合は、「システム情報」タブを開き、発見されたビルドのコンボボックスから、バグが発生したビルド バージョンを選択します (図 3.105)。
- ④ 「保存して閉じる」ボタンをクリックすると、バグが作成されます。



図 3.102 テスト結果からのバグの起票 (Test Manager)

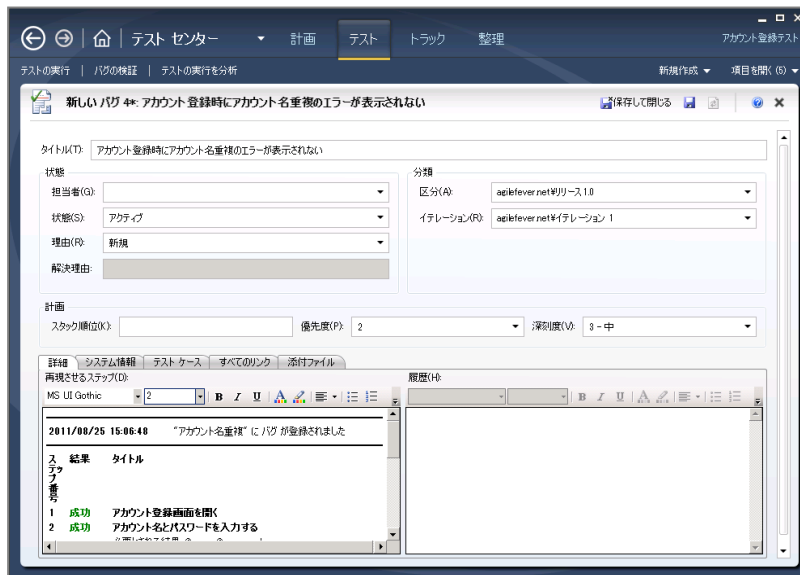


図 3.103 新しいバグ (Test Manager)

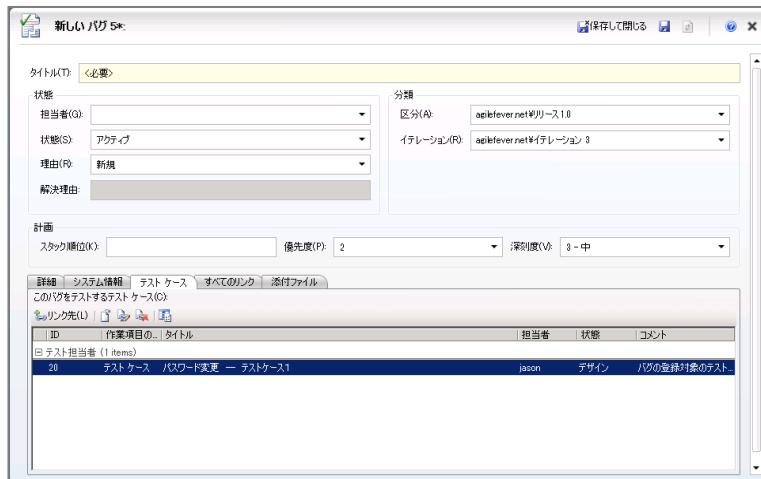


図 3.104 テスト ケースへのリンク (Test Manager)

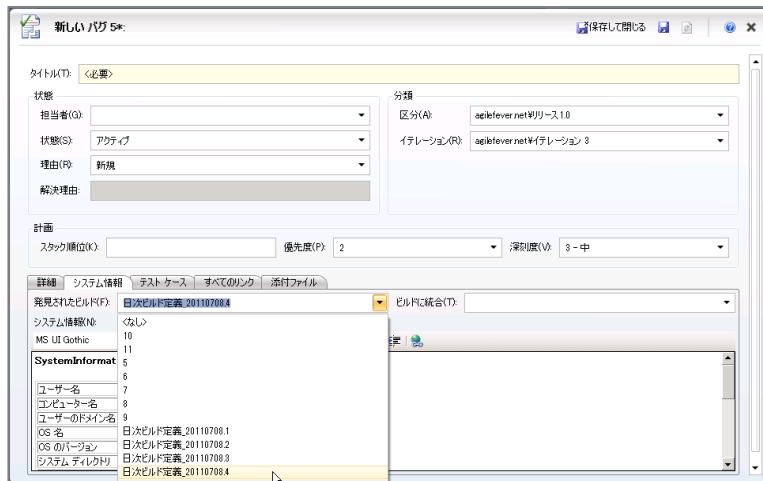


図 3.105 バグ発生時のビルド バージョンの設定 (Test Manager)

■ リリース

受け入れテストが完了し、製品の品質も十分に安定しました。操作手順など必要なドキュメントも用意し、いよいよ最初のリリースです。

本プロジェクトでは、スクラムを進める上で TFS を活用することにより、コミュニティ事務局の担当者の協力や、開発チームのチームワークを最大限に生かし、全員が納得できるコミュニティ サイトをリリースすることができました。TFS の豊富でパワフルな機能から必要な機能を適切に選択し、プロジェクトに合わせて活用することによってアジャイル開発は進化し、チームはさらなる開発の効率化を図ることができるでしょう。

4. まとめ

TFS を使ったアジャイル開発についてみてきました。

これまでのアジャイル開発では、作業トラッキング、ソース管理、テスト管理などの複数のツールを連携させて使われていましたが、TFS ではこれらの機能を、1つのツールですべてカバーできるようになったのがポイントの1つでしょう。TFS が開発に関するあらゆる情報を一元管理するデータベースとして機能することにより、アジャイルの原則である「**個人の相互作用**」が一段高いレベルで実現することができます。

ただし、TFS は多くの機能に、逆に圧倒されてしまうかもしれません。アジャイル開発では、タスク管理やソース管理など開発系のプラクティスから始め、要求管理や受け入れテストなど、顧客を巻き込んだプラクティスまで、段階的に適用することになるでしょう。

さらに、TFS は Visual Studio だけでなく、Eclipse や Web からの参加できるようになり、従来の Visual Studio ユーザー以外の開発者や、開発以外のメンバーへの門戸が開かれ、より情報共有が進むようになりました。組織においては1つのプロジェクトだけでなく、部門から会社全体など、広く展開することにより、より多くのノウハウを蓄積できるようになります。したがって、TFS は繰り返し使い続けることによって、TFS というデータベースにナレッジが蓄積され、組織のポテンシャルを引き出すことが期待できます。