

Rational Suite®

チュートリアル

バージョン: 2003.06.10

G126-5387-00

WINDOWS

Rational™

the software development company

support@japan.rational.com
<http://www.rational.co.jp>

法的通知

© 1998-2003, Rational Software Corporation. All rights reserved.

バージョン番号: 2003.06.10

本マニュアル(「本著作物」)は、アメリカ合衆国その他の国々の著作権法及び種々の条約により保護されています。**Rational Software Corporation**の文書による事前の同意を得ることなく本著作物を複製し又は頒布することは、禁じられています。

本著作物はライセンスに基づいて提供されるもので、ライセンス規定に従う場合にのみ、使用または複製できます。ライセンス契約で明示的に許可されている場合を除き、本著作物または本著作物の複製を第三者に提供することは禁じられています。本著作物の権利または所有権を譲渡することはできません。ライセンス条項の全文については、ライセンス契約書をお読みください。

Rational Software Corporation、Rational、Rational Suite、Rational Suite ContentStudio、Rational Apex、Rational Process Workbench、Rational Rose、Rational Summit、Rational Unified process、Rational Visual Test、AnalystStudio、ClearCase、ClearCase Attache、ClearCase MultiSite、ClearDDTS、ClearGuide、ClearQuest、PerformanceStudio、PureCoverage、Purify、Quantify、Requisite、RequisitePro、RUP、SiteCheck、SiteLoad、SoDa、TestFactory、TestFoundation、TestMate、TestStudio は、Rational Software Corporation の米国およびその他の国における登録商標です。Rational のロゴ、Connexis、ObjecTime、Rational Developer Network、RDN、ScriptAssure、XDE は、Rational Software Corporation の米国およびその他の国における商標です。その他すべての名前は、識別の目的でのみ使用されているものであり、それぞれの会社の商標または登録商標です。

米国特許番号 5,193,180、5,335,344、5,535,329、5,574,898、5,649,200、5,675,802、5,754,760、5,835,701、6,049,666、6,126,329、6,167,534、6,206,584 の請求の範囲内の部分。このほかにも米国特許及び国際特許申請中。

米国政府の権利

このソフトウェアおよび文書は、「商業的コンピュータソフトウェア」、「商業的ソフトウェア」または「使用が制限されたコンピュータソフトウェア」として提供され、規約は該当する DFARS 252.227、DFARS 252.211、FAR 2.101、FAR 52.227 (またそれ以前に定められた条項) に規定されています。本ソフトウェア製品およびドキュメントの使用、複製、または開示は、DFARS 227.7202、FAR 52.227-19 の下位条項 (c)、または FAR 52.227-14 (またはその改訂された規定) に定められるように、該当する Rational Software Corporation ライセンス契約書の条項の制約を受けます。

免責事項

本書および関連ソフトウェアは、ライセンス契約に基づいて使用することができます。そのような使用許諾契約書に別段の明示的な規定がある場合を除き、また、それぞれの国の法律により禁止または制限されている場合を除き、Rational Software Corporation は、本メディア、ソフトウェア製品、およびその関連文書について、明示的にも暗黙的にも、商品性に関する保証、非権利侵害性に関する保証、特定目的への適合性に関する保証、取り扱い、使用、または取引行為に伴う保証、およびライセンシーによる静穏無事な製品使用に対する妨害がないことの保証について一切の責任を負いません。

第三者の通知、コード、使用許諾および確認

Portions Copyright © 1992-1999, Summit Software Company. All rights reserved.

Microsoft、Microsoft のロゴ、Active Accessibility、Active Client、Active Desktop、Active Directory、ActiveMovie、Active Platform、ActiveStore、ActiveSync、ActiveX、Ask Maxwell、Authenticode、AutoSum、BackOffice、BackOffice のロゴ、bCentral、BizTalk、Bookshelf、ClearType、CodeView、DataTips、Developer Studio、Direct3D、DirectAnimation、DirectDraw、DirectInput、DirectX、DirectXJ、DoubleSpace、DriveSpace、FrontPage、Funstone、Genuine Microsoft Products のロゴ、IntelliEye、IntelliEye のロゴ、IntelliMirror、IntelliSense、J/Direct、JScript、LineShare、Liquid Motion、Mapbase、MapManager、MapPoint、MapVision、Microsoft Agent のロゴ、Microsoft eMbedded Visual Tools のロゴ、Microsoft Internet Explorer のロゴ、Microsoft Office Compatible のロゴ、Microsoft Press、Microsoft Press のロゴ、Microsoft QuickBasic、MS-DOS、MSDN、NetMeeting、NetShow、Office のロゴ、Outlook、PhotoDraw、PivotChart、PivotTable、PowerPoint、QuickAssembler、QuickShelf、RelayOne、Rushmore、SharePoint、SourceSafe、TipWizard、V-Chat、VideoFlash、Visual Basic、Visual Basic のロゴ、Visual C++、Visual C#、Visual FoxPro、Visual InterDev、Visual J++、Visual SourceSafe、Visual Studio、Visual Studio のロゴ、Vizact、WebBot、WebPIP、Win32、Win32s、Win64、Windows、Windows CE のロゴ、Windows のロゴ、Windows NT、Windows Start のロゴ、XENIX は、Microsoft Corporation の米国およびその他の国における商標または登録商標です。

Sun、Sun Microsystems、Sun のロゴ、Ultra、AnswerBook 2、medialib、OpenBoot、Solaris、Java、Java 3D、ShowMe TV、SunForum、SunVTS、SunFDDI、StarOffice、および SunPCi は、Sun Microsystems の米国および他の国における商標または登録商標です。

Purify は、Sun Microsystems, Inc. の米国特許番号 5,404,499 の下にライセンス供与されています。

Globetrotter ソフトウェア (FLEXIm ライブラリおよびユーティリティ) の本来の用途は、ソフトウェアライセンス管理であり、他の製品またはアプリケーションにこれらのソフトウェアを組み込むことは、ライセンスに含まれません。

BasicScript は、Summit Software Company の登録商標です。

デザイン パターン: Erich Gamma、Richard Helm、Ralph Johnson および John Vlissides による再使用可能なオブジェクト指向のソフトウェアのエLEMENT。Copyright © 1995 by Addison-Wesley Publishing Company, Inc. All rights reserved.

追加の法的通知は、お客様の Rational ソフトウェア インストールに含まれています。

目次

まえがきxi
対象読者	xi
その他の参照先	xi
Rational Suite のマニュアル構成	xii
Rational カスタマ サポートの連絡先	xiii
1 Rational Suite について	15
ソフトウェア開発の原則	15
Rational Suite の利点	18
Rational Suite とは	19
チーム統一用ツール	19
Rational Suite Team Unifying Platform	19
アナリスト用ツール	21
Rational Suite AnalystStudio	21
開発者用ツール	21
Rational Suite DevelopmentStudio	22
Rational Suite DevelopmentStudio – RealTime Edition	23
テスト担当者用ツール	23
Rational Suite TestStudio	23
Rational Suite Enterprise	24
Rational Suite: 完全なライフサイクル ソリューション	25
詳細	26
次の演習	26
2 このチュートリアルについて	27
前提事項	27
インストールされている Rational Suite ツールの確認	27
ClassicsCD.com: チュートリアル用サンプル アプリケーション	28
チュートリアルの背景の説明	28
チュートリアル用サンプル アプリケーションと関連ファイルのインストール	28
入門	30
プロジェクトの登録	30
変更依頼データベースとプロジェクトの関連付け	31
アプリケーションの使用	33
ビルド内容の決定	34
このチュートリアルの使い方	35

まとめ	36
この章の学習内容	36
次の演習	36
3 Rational Unified Process について	37
対象読者	37
概要	37
Rational Unified Process (RUP) とは	38
Rational Unified Process と Rational Suite	39
仕組みについて	39
プロセスの概要	40
主な概念	41
ワークフローについて	43
アクターとユースケースの使用	44
ツール メンター : Rational ツールによるプロセスのサポート	46
基本的なことについて	46
まとめ	47
詳細	47
クリーンアップ	47
この章の学習内容	47
次の演習	47
4 プロジェクト成果物の変更管理	49
対象読者	49
統一変更管理とは	49
UCM ツール	50
UCM におけるツールの使用 : ClearQuest と ClearCase LT	51
Web 開発のためのコードとコンテンツの統合	53
まとめ	54
詳細	54
この章の学習内容	54
次の演習	54
5 要求の作成	55
対象読者	55
概要	55
要求管理が必要な理由	57
要求の決定方法	57
要求管理	57
RequisitePro の使用	58

ユース ケースの使用	58
ユース ケースを使用する理由	58
RequisitePro での要求の処理方法	61
ユース ケースについての詳細	61
Rose を使用したユース ケース作業の継続	62
ユース ケース図を使用した作業	62
Rose モデルと RequisitePro プロジェクトの関連付け	64
新しい要求の作成	65
データベース内の要求の確認	66
別の要求へのリンク	67
追跡可能性リンクとサスペクト リンク	67
その他の要求タイプ	68
要求収集の終了時期	68
拡張ヘルプ	68
まとめ	69
詳細	69
クリーンアップ	69
この章の学習内容	69
次の演習	69
6 テスト計画	71
対象読者	71
概要	71
テスト計画とは	72
リスクの管理	72
計画の作成と進行の測定	72
テスト計画の作成	73
テスト計画の構成	73
テスト内容の決定	75
テスト ケースを使用した作業	76
Rational Rose からのテスト入力	77
Rational RequisitePro からのテスト入力	77
追跡可能性リンクとサスペクト リンク	78
テスト ケースの詳細決定	79
テスト計画による影響の理解	79
テスト計画作成の継続	80
リスクとリソース	80
実行するテストのタイプ	81

テストの段階	81
プロジェクトのスケジュールリング	81
テスト成果物についての詳細	82
まとめ	83
詳細	83
クリーンアップ	83
この章の学習内容	83
次の演習	83
7 機能拡張のモデリング	85
対象読者	85
概要	85
ビジュアル モデリングとは	86
Rational Rose の使用	86
ビジュアル モデリングとチュートリアル	86
シーケンス図を使用した作業	86
シーケンス図の表示	87
機能拡張用メッセージの追加	89
Web へのモデル要素の発行	90
シーケンス図での作業の継続	92
オブジェクトのリファイン	92
コードの実装	92
データのモデリング	93
利点	93
まとめ	94
詳細	94
クリーンアップ	94
この章の学習内容	94
次の演習	94
8 プロジェクト ステータスの伝達	95
対象読者	95
概要	95
プロジェクト ステータスの管理	96
SoDA とは	96
SoDA テンプレートの使用	96
ユース ケース レポートを生成する理由	97
ユース ケース レポートの作成	97
SoDA テンプレートの操作	99

ProjectConsole とは	99
プロジェクト Web サイトの使用	99
プロジェクト メトリクスを使用した作業	100
まとめ	102
詳細	102
クリーンアップ	103
この章の学習内容	103
次の演習	103
9 信頼性テスト	105
対象読者	105
信頼性テスト用のツール	105
Rational Suite のランタイム分析ツール	106
Rational Purify	106
Rational PureCoverage	107
Rational Quantify	108
Purify、PureCoverage、Quantify をほかの Rational ツールと共に使用する	108
まとめ	109
詳細	109
この章の学習内容	109
次の演習	109
10 機能テスト	111
対象読者	111
概要	111
機能テストとは	112
テスト スクリプトの操作	112
スクリプトとモジュール化	112
開始点の設定	113
テスト スクリプトの操作	113
スクリプトの記録	115
スクリプトの記録開始	115
検証ポイントの作成	116
記録セッションの終了	117
ルーチンへのテスト スクリプトの追加	118
テスト計画へのテスト スクリプトの組み込み	119
新しいビルドでのスクリプトの再生	120
結果の分析	121
失敗への対処	121

意図的な変更への対処	121
実質的なエラーへの対処	122
エラーの報告	122
まとめ	124
詳細	124
クリーンアップ	124
この章の学習内容	124
次の演習	124
11 次の反復の計画	125
対象読者	125
概要	125
プロジェクトの状態の評価	126
作業量の表示	126
拡張依頼の操作	128
その他の計画アクティビティ	130
次の反復での作業	130
まとめ	131
詳細	131
クリーンアップ	131
この章の学習内容	131
このチュートリアル of 学習内容	132
次の演習	132
用語集	133
索引	139

まえがき

Rational Suite® は包括的な統合ツールであり、ソフトウェア開発における最善の実践原則を具体化すると共に、ソフトウェア開発工程全体をカバーします。このチュートリアルでは、**Rational Suite** を使用してアプリケーションを計画、設計、実装、テストするための基本的な方法について説明します。チュートリアルには、ユーザーが理解を深められるようにさらに詳しい情報も記載されています。

対象読者

このチュートリアルは次のような読者を対象としています。

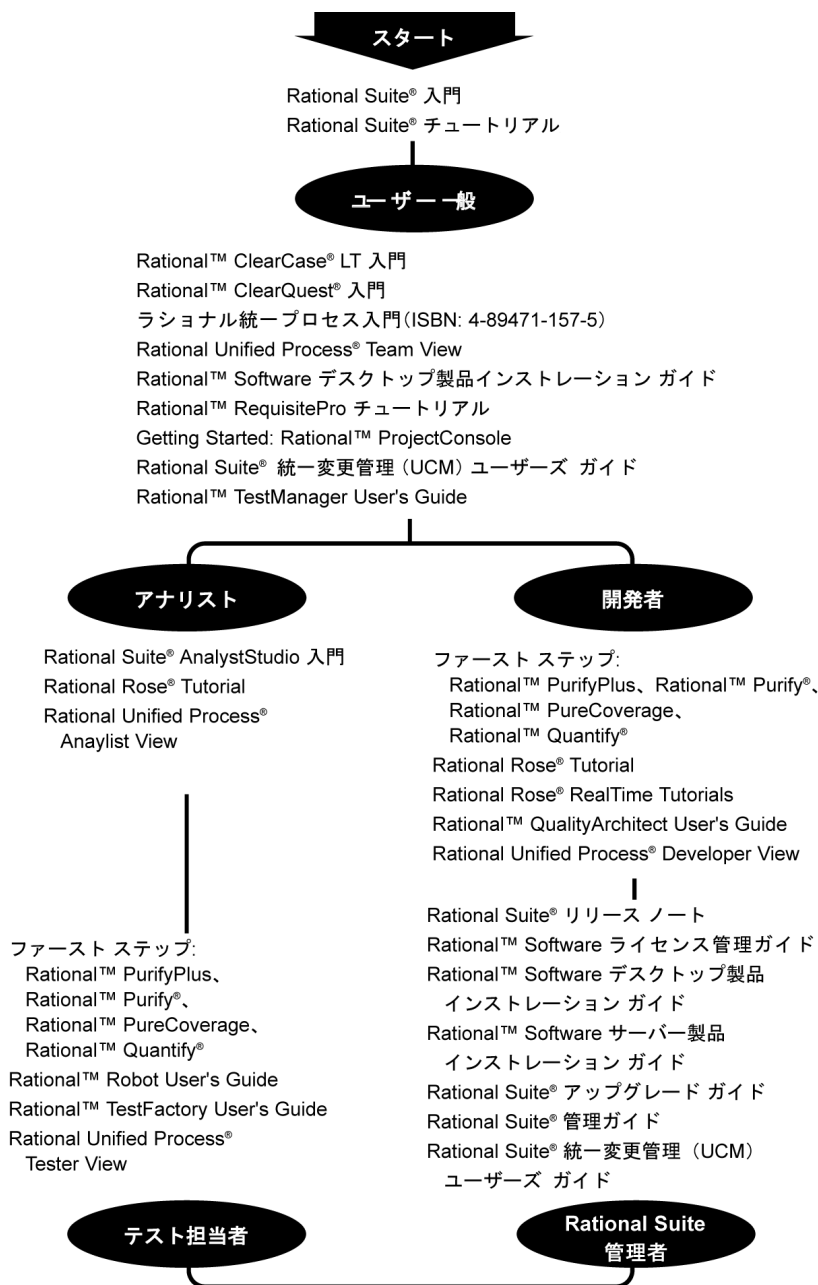
- アナリスト、開発者、テスト担当者、プロジェクトリーダー、マネージャーなどの開発チームメンバー
- Windows ベースのアプリケーション開発の経験者

このチュートリアルは、**Rational™** のツールを使用した経験がなくても使用できます。

その他の参照先

- マニュアルはすべてオンライン (HTML 形式または PDF 形式) で参照できます。オンラインマニュアルは、**Rational Solutions for Windows** のオンラインドキュメント CD-ROM に収録されています。
- **Rational** の技術資料については、<http://www.rational.com/documentation> (ただし、英語のみのご利用となります) を参照してください。
- トレーニングコースの詳細については、ラショナルユニバーシティの Web サイト <http://www.rational.co.jp/services/ru/> を参照してください。
- **Rational Suite** 製品を使用したソフトウェア開発に関する記事、ディスカッションフォーラム、またはウェブベースのトレーニングコースについては、次の手順に従って、**Rational Developer NetworkSM** に参加してください。Windows の [スタート] メニューから、[プログラム]、[Rational Software] の順にポイントし、[Logon to the Rational Developer Network] をクリックします。

Rational Suite のマニュアル構成



Rational カスタマ サポートの連絡先

本製品のインストール、使用、保守に関するご質問については、以下の Rational カスタマ サポートまでお問い合わせください。

地域	電話	Fax	メール
アジア太平洋 (日本を含む)	+61-2-9419-0111	+61-2-9419-0123	support@apac.rational.com (英語のみ対応) support@japan.rational.com (日本語対応可)

メモ: まず次の内容に答える準備をしてから、Rational カスタマ サポートにお問い合わせください。

- お名前、会社名、電話番号、電子メールアドレス
- オペレーティング システム、バージョン番号、適用されているサービス パック またはパッチ
- 製品名とリリース番号
- サービス リクエスト番号 (報告済みの問題についてお問い合わせいただく場合)

前回のソフトウェアプロジェクトを振り返ってみてください。製品は予定どおりリリースされましたか。開発費は予算内におさまりましたか。チームメンバー間のコミュニケーションはスムーズでしたか。要求の定義、設計の開発、コードの作成において、プロジェクトをとおして整合性を維持できましたか。ビルドプロセスは反復可能でしたか。ソフトウェアは要求を満たし、ユーザーの満足を得ることができましたか。実行に信頼性はありますか。

多くのプロジェクトチームが、これらの問題を経験しています。ソフトウェアプロジェクトでは、予定より遅れて完了したり（完了しない場合もあります）、プロジェクトの結果が要求に対応していなかったりすることがよくあります。多くのプロジェクトでは、プロセスの後半に設計上の重大な欠陥が見つかっています。開発中ではなく、ソフトウェアの出荷後に障害が見つかる場合もあります。

次のプロジェクトを成功させるにはどうしたらよいでしょうか。

ソフトウェア開発の原則

Rational は、ソフトウェアの品質を向上させると共に開発期間を短縮し、ソフトウェア開発における多くの問題を解決します。Rational のソリューションを使用すると、次の要素を組み合わせたソフトウェア開発が可能です。

- ソフトウェア開発における最善の実践原則
- 最善の実践原則を自動化する統合ツール
- 最善の実践原則とツールの導入と実装を促進するプロフェッショナル サービス (図 1 を参照してください。)

図 1 Rational の最善の実践原則、ツール、サービス



Rational は、以下に示すソフトウェア開発の最善の実践原則に焦点を当てて、生産性と効率の向上を支援します。

ソフトウェアの反復型開発

反復型開発とは、プロジェクトのライフサイクル全体にわたってシステムのインクリメンタルサブセットを分析、設計、実装していくことです。プロジェクト チームは、1 つの反復ごとに、必要なシステム機能のサブセットについて計画、開発、テストを行います。チームは、次のインクリメントを開発し、その結果を前の反復と統合します。各反復により内部または外部へのリリースが行われ、要求を満たす製品のリリース目標へと近づくことができます。

反復型の開発には次のような利点があります。

- プロジェクトの予測性を高めることができる。
- 早期にフィードバックを収集できる。
- プロジェクトの早い段階でリスクを特定して取り除くことができる。
- プロジェクトのライフサイクルを通じてテストを行うことができる。

要求管理

要求とは、プロジェクトを成功させるための 1 つの基準です。プロジェクトの要求は、「顧客の要望」や「次のバージョンに不可欠な新しい機能」などの質問に答える形で示されます。ほとんどのソフトウェア開発チームは、要求に対して作業を行います。小規模で非公式なプロジェクトでは、要求をテキストファイルや電子メール メッセージに保管する場合があります。その他のプロジェクトでは、要求を記録し管理するためにより形式的な方法がとられます。

要求管理を行うということは、要求変更がプロジェクトにどのような影響を及ぼすかを理解し、すべてのチーム メンバーと利害関係者に対してその要求を効率的に通知することができるということを意味します。効率的な要求管理は、組織内で製品を所定の目標に到達させるために役立ちます。

コンポーネント ベースのアーキテクチャの使用

ソフトウェア アーキテクチャは、ソフトウェア プロジェクトを構築するための基本的なフレームワークです。アーキテクチャを定義する場合は、システムの構造要素とその動作を設計し、これらの要素を段階的に増大するサブシステムにどのように適合させるかを決定します。

コンポーネントは、システムの重要で置き換え可能な独立した部品であり、明確な目的を実現するためのデータと機能で構成されています。コンポーネントはゼロから作成することもできますし、以前作成したコンポーネントを再利用することもできます。また、ほかの会社からコンポーネントを購入することもできます。

コンポーネント ベースでアーキテクチャを設計することによって、アプリケーションのサイズを小さくして複雑化を避けることができると共に、保守性と拡張性を向上させることができるため、より強力で柔軟なシステムを構築することが可能です。

視覚的なモデリング

ビジュアル モデリングによって、複雑なソフトウェア設計の管理が容易になります。ビジュアル モデリングとは、単純な言い方をすれば、システム アーキテクチャの青写真を視覚的に作成することです。ビジュアル モデルを使用すると、要求、設計、実装間の不整合を見つけやすくなります。また、システムのアーキテクチャを評価する助けとなるため、健全な設計を行うことができます。

ビジュアル モデルは、大量の情報を簡潔に表すことができるので、チーム全体のコミュニケーションを向上させることもできます。**Rational** ツールを使用すれば、ソフトウェア システムの視覚化とドキュメント化に業界標準言語の統一モデリング言語 (UML) を使用してビジュアル モデルを作成できます。

継続的な品質の検証

製品品質の検証は、コード品質を確認するためのテスト、ドキュメント作成、製品関連のトレーニングなどの作業を通じて行われます。テストには、システムが要求された機能と信頼性を備えていることや、負荷が加えられた状態で所定の性能を発揮できることなどの確認も含まれます。これは、効果的なユーザー サポート用の材料があることも意味しています。

反復型開発の重要な利点として、開発プロセスの初期からテストを開始できることが挙げられます。すべての反復をテストすることによって、早い段階で問題を発見し、要求、設計、実装の間にある不整合を明らかにすることができます。

変更管理

変更管理は、追跡、繰り返し、予測が可能な方法で行うことが重要です。変更管理の内容としては、並行開発の簡素化、拡張依頼と変更依頼の追跡と処理、反復可能な開発プロセスの定義、ソフトウェア ビルドの確実な再現などがあります。

プロジェクトの進行に伴って変更が増えてきた場合、明確に設定された繰り返し可能な変更プロセス用ガイドラインの存在が、進行についての確実なコミュニケーションを促進する上で大きな助けとなります。チーム メンバーに変更を認識させることで、変更が管理されていない場合に生じるリスクを管理しやすくなります。

Rational Suite の利点

以上のソフトウェア開発の原則を実現するために、Rational では、Rational Unified Process® によってサポートされている業界最先端のソフトウェア開発ツール群である Rational Suite を提供しています。これらのツールでは、プロジェクトのライフサイクル全体をとおしてユーザーを支援します。

Rational Suite には複数のツールとプロセスがパッケージされており、アナリスト、開発者、テスト担当者など、開発チームの特定のエンジニアに合わせてカスタマイズされたいくつかのエディションがあります。

これらは単独でも、ソフトウェア作成を成功させるための支援ツールとして各国で使用されています。これらのツールを Rational Suite に統合すると、次のことが可能になります。

- チームの統一: コミュニケーションを円滑にし、共通のツールを使用することによって、チームを統一します。
- 個人の生産性の向上: チーム内の主要な役割に合わせてカスタマイズされた Suite のエディションにパッケージされている業界最先端の開発ツールによって、個人の生産性を高めます。
- 導入の簡素化: インストール、ライセンス、ユーザー サポート計画を単純化した包括的な統合ツール セットを提供することによって、導入作業を簡素化します。

Rational Suite とは

Rational Suite のエディションは、すべてのチーム メンバーの役割に合わせてカスタマイズされたツールのセットです。Suite の各エディションには、Rational Suite Team Unifying Platform のツールが含まれています。Team Unifying Platform は、チームのパフォーマンス向上に焦点を当てた共通ツール セットです。また、Rational Suite のエディションには、開発チームの特定のエンジニア向けに選択されたツールも含まれています。以下の項では、Suite の各エディションとそれに含まれているツールについて説明します。

チーム統一用ツール

Rational Suite Team Unifying Platform

Rational Suite Team Unifying Platform では、最高の生産性と品質を得るために、ソフトウェア開発チームのすべてのメンバーを統一します。この製品は、要求からリリースまでのプロセスにおける変更の管理、高品質アプリケーションの作成、結果の伝達を行うための最善の実践原則と統合ツールを提供します。

Rational Suite Team Unifying Platform は、プロジェクト メンバーが共通のプロジェクト情報にアクセスする必要があっても Suite のほかのエディションに見られるような特定の役割用に最適化されたツールを必要としない場合に有効です。Suite のこのエディションのツールは、プロジェクト マネージャー、プログラム マネージャー、プロジェクト管理者、開発マネージャーなどが使用します。

Team Unifying Platform は、Rational Suite のすべてのエディションに含まれています。以下のツールが収められています。

Rational Unified Process (RUP):

Rational Unified Process は、チームのソフトウェア開発プロセスの最善の実践原則 (概要については 16 ～ 18 ページを参照) を集めたオンライン ソフトウェアです。RUP® とも呼ばれるこのオンライン ソフトウェアは、開発ライフサイクルの各フェーズにおけるガイドライン、テンプレート、ツール メンター (特定の Rational ツールにガイドラインを適用する手順) を提供します。また、RUP をカスタマイズして次のことを行うこともできます。

- プロジェクトの各段階で必要となるプロセス コンポーネントのみを使用する
- 独自のプロセス コンポーネントを作成する
- 同僚や上司と実践的手法を交換する

Rational RequisitePro

プロジェクト要求の変更の整理、優先順位決定、追跡、管理を行うためのツールです。RequisiteWeb インターフェイスと合わせて使用すれば、要求へのアクセス、作成、管理を Web ブラウザから行うことができます。

Rational ClearQuest

ソフトウェア開発に関連する変更アクティビティ (拡張依頼、障害レポート、ドキュメントの変更など) を管理するツールです。ClearQuest® Web インターフェイスを使用して、レコードの登録と検索、クエリーとレポートの作成または編集、ショートカットの作成など、ClearQuest の主要な操作を Web ブラウザから実行することができます。また、ClearQuest MultiSite を使用すると、異なる場所に分散するチーム内で変更依頼情報を共有できます。

Rational SoDA

ソース コードや Rational ツールで作成したファイルなど、プロジェクト開発中に作成したファイルから情報を抽出することによってプロジェクト ドキュメントを自動的に生成します。SoDA では、あらかじめ定義されているテンプレートやユーザーがカスタマイズしたテンプレートを使用して情報を形式化します。SoDA は Microsoft Word と統合されているので、簡単に使用したりカスタマイズしたりすることができます。

Rational ClearCase LT

ソフトウェア構成管理と、すべてのソフトウェア プロジェクト アセット (要求、ビジュアルモデル、コードなど) に対する変更を追跡するためのビルトインプロセスを提供します。また、Web インターフェイスを備えており、ClearCase® LT の主要な操作を行うことができます。ClearCase LT では、変更管理とワークフロー制御のための Rational の最善の実践原則プロセスである統一変更管理をサポートしています。

Rational TestManager

実環境機能テストとマルチユーザー テストを作成して、Web アプリケーション、複数階層アプリケーション、データベース アプリケーションの信頼性とパフォーマンスを判定するために使用します。TestManager では、計画されたテスト数、スクリプト化されたテスト数、実行されたテスト数、対象となった要求、テストの合否数を追跡します。TestManager を使用すると、プロジェクトのステータスを客観的に評価し、その結果をプロジェクトの利害関係者に伝達するレポートを作成してカスタマイズすることができます。

Rational ProjectConsole

ソフトウェア開発中に作成されたデータからグラフとゲージを自動的に生成して、プロジェクト進捗情報を追跡します。ProjectConsole は Microsoft Project と統合されているので、集中管理するプロジェクト計画を作成できます。また、中央の Web サイトにプロジェクトの成果物をまとめることができるので、すべてのチーム メンバーが成果物を参照することができます。

Rational Developer Network

専門的スキルを広げると共に磨きをかけて、技術の最先端に位置することができます。**Rational Developer NetworkSM** は、ソフトウェア専門家のためのオンライン コミュニティです。ここでは、有益な情報を得たり、ソフトウェアの専門家同士でアイデアを交換する機会を持ったりすることができますほか、ソフトウェア開発チームのための最善の実践原則を利用することができます。このネットワークでは、ホワイト ペーパー、ドキュメント、記事、トレーニングなどを提供しています。

アナリスト用ツール

アナリストには、次の役割があります。

- システムの機能の決定
- 開発組織へのユーザー ニーズの提示
- 要求の決定と管理

Rational Suite AnalystStudio

Rational Suite AnalystStudio は、チームのアナリストを支援します。AnalystStudio には、Team Unifying Platform と次のツールが含まれています。

Rational Rose (Professional Data Modeler Edition)

業界標準の統一モデリング言語 (UML) を使用して、データベース、アーキテクチャ、コンポーネントのビジュアル モデリングを行うことができます。UML は、ソフトウェア システムの仕様決定、視覚化、体系化、ドキュメント化を行うための言語です。Rose のこのエディションでは、モデリング環境とデータベース設計環境を統合し、オブジェクトとデータ モデルのマッピングや、ビジネス モデル、アプリケーション モデル、データ モデルにまたがる変更管理を行います。

開発者用ツール

開発者には、次の役割があります。

- システムの動作の決定
- アーキテクチャの定義
- コードの作成、変更、管理、テスト

Rational Suite には、チームの開発者とアーキテクトを支援するエディションが 2 つあります。Rational Suite DevelopmentStudio と Rational Suite DevelopmentStudio – RealTime Edition です。

Rational Suite DevelopmentStudio

Rational Suite DevelopmentStudio は、システム開発者、設計者、アーキテクトを支援します。Rational Suite DevelopmentStudio には、Team Unifying Platform と以下のツールが含まれています。

Rational Rose (Enterprise Edition)

UML を使用して、アーキテクチャ、コンポーネント、データのビジュアル モデリングを行うことができます。Rational Rose® では、コードのフレームワークを、Java、C++、Microsoft Visual Basic、その他の一般的な言語で再生成できます。Rose は Microsoft Visual Studio、IBM VisualAge for Java と緊密に統合化されています。また、Rose はラウンドトリップ エンジニアリング機能を備えており、モデルと実装間の整合性を維持するためのプロセスを自動化することができます。

Rational QualityArchitect

QualityArchitect は Rational Rose のアドインで、ビジュアル モデルからテスト コードを生成することで、テスト コード作成の機械的な側面を自動化します。この機能により、開発者はアプリケーションの完成前にスタブとドライバを作成してコンポーネント テストを自動的に生成することができます。早い段階でテストを実行するとプロジェクトのリスクを減らすことができるので、これは非常に重要な機能です。開発チームは、設計を進める前に、実装しようとしているシステム アーキテクチャが機能要求やパフォーマンス要求をどの程度満足しているかを判断することができます。Rational QualityArchitect では、Enterprise JavaBeans、COM、COM+、DCOM によるモデルをサポートしています。

Rational PureCoverage

Java プログラム、Visual C++ プログラム、Visual Basic プログラム、Visual Basic.NET プログラム、Visual C#.NET プログラムのどの部分が実行され、どの部分が実行されていないかを特定します。これによってテスト漏れの部分が明確になるので、テストされていないアプリケーション コードがユーザーに渡るのを防ぐことができます。

Rational Purify

Visual C++ アプリケーション コードのランタイム エラーやメモリ リークの箇所と、Java や .NET 管理アプリケーション コードのガーベッジ収集に関連するエラー箇所を特定します。

Rational Quantify

Java アプリケーション、Visual C++ アプリケーション、Visual Basic アプリケーション、Visual Basic.NET アプリケーション、Visual C#.NET アプリケーションのプロファイリングを行い、ソース コード内のパフォーマンスの問題を特定します。

Rational Suite DevelopmentStudio – RealTime Edition

Rational Suite DevelopmentStudio – RealTime Edition は、リアルタイムの組み込み型開発を中心に行う開発者向けに設計されています。このエディションには、Rational Suite DevelopmentStudio のすべてのツールが含まれています。ただし、Rational Rose Enterprise Edition ではなく Rational Rose RealTime Edition が使用されています。

Rational Rose (RealTime Edition)

リアルタイム開発におけるさまざまな問題に対処するために、表記法、プロセス、ツールの強力な組み合わせを提供します。Rose RealTime を使用すると、次のことができます。

- 実行可能モデルを作成してコンパイルし、ホストプラットフォームまたはターゲットプラットフォーム上で UML 設計のシミュレーションを監視します。結果として、早い段階で設計を修正したり、品質を継続的に検証したりすることができます。
- リアルタイム オペレーティング システム向けの UML 設計モデルから直接、完全かつ展開可能な実行可能ファイルを C または C++ で生成します。このようなアプリケーションを生成することによって手動による変換作業が不要になり、コスト上昇の原因となる設計解釈上のエラーを回避することができます。

このチュートリアルでは、Rose RealTime についての詳しい説明は行いません。詳細については、Rose RealTime の [Help] メニューからオンライン チュートリアルを参照してください。

テスト担当者用ツール

テスト担当者には、次の役割があります。

- ソフトウェアがすべての要求を満たしていることの確認
- テストの作成、管理、実行
- 結果の報告と修正の確認

Rational Suite TestStudio

Rational Suite TestStudio は、テスト担当者向けに設計されています。TestStudio には、Team Unifying Platform と以下のツールが含まれています。

Rational Robot

テスト スクリプトの記録と再生を自動化して、機能テストとパフォーマンス テストの両方を容易にします。このツールを使用すると、テストの記述、トラブルシューティング、実行が可能で、分析のためにテスト結果を取得することができます。

Rational TestFactory

自動テスト生成とソース コード カバレッジ分析を組み合わせ、テストを自動化します。また、すべての GUI 機能とソース コードのすべての行を含めた、アプリケーション全体のテストを行います。

Rational PureCoverage

Java プログラム、Visual C++ プログラム、Visual Basic プログラムのどの部分が実行され、どの部分が実行されていないかを特定します。テスト スクリプトを使用して、アプリケーションを実行します。これによってテスト漏れの部分が明確になるので、テストされていないアプリケーション コードがユーザーに渡るのを防ぐことができます。

Rational Purify

Visual C++ アプリケーション コードのランタイム エラーやメモリ リークの箇所と、Java アプリケーション コードのガーベッジ収集に関連するエラー箇所を特定します。テスト スクリプトを使用してアプリケーションを実行することによって、この処理を行います。

Rational Quantify

Java アプリケーション、Visual C++ アプリケーション、Visual Basic アプリケーションのプロファイリングを行い、コード内のパフォーマンスにおける問題を特定します。テスト スクリプトを使用してアプリケーションを実行することによって、この処理を行います。

Rational Suite Enterprise

プロジェクトによっては、チーム メンバーがさまざまな種類のタスクを行うことがあります。たとえば、小規模なプロジェクトでは、メンバーがそれぞれ複数の役割を果たすというのはよくあることです。大規模なプロジェクトでも、各メンバーが複数のタスクを担当することがあります。したがって、すべてのチーム メンバーがすべてのツールを使用できるようにするのは意味のあることです。

Rational Suite Enterprise には、Team Unifying Platform、AnalystStudio、DevelopmentStudio、TestStudio のすべてのツールが含まれているので、すべてのチーム メンバーのニーズを満たすことができます。

Rational Suite: 完全なライフサイクル ソリューション

Rational Suite は開発チーム用の統合ソリューションで、チームのニーズを的確に満たすさまざまな製品が用意されています。以下の表に、各 Rational Suite 製品に含まれているツールを示します。

		プロジェクト リーダー / マネージャー	アナリスト	開発者				テスト 担当者	すべての 役割
		Team Unifying Platform	Analyst Studio	DevelopmentStudio		DevelopmentStudio - RealTime Edition		TestStudio	Enterprise
		Windows	Windows	Windows	UNIX	Windows	UNIX	Windows	Windows
Team Unifying Platform	Rational Unified Process®	●	●	●	●	●	●	●	●
	Rational RequisitePro	●	●	●	● ^{WEB}	●	● ^{WEB}	●	●
	Rational ClearQuest®	●	●	●	●	●	●	●	●
	Rational SoDA	● ^{SW}	● ^{SW}	● ^{SW}	● ^{SF}	● ^{SW}	● ^{SF}	● ^{SW}	● ^{SW}
	Rational ClearCase® LT	●	●	●	●	●	●	●	●
	Rational TestManager	●	●	●	● ^{TA}	●	● ^{TA}	●	●
	Rational ProjectConsole	●	●	●		●		●	●
Rational Rose®			● ^M	● ^E	● ^U	● ^{RT}	● ^{RT/U}		● ^E
Rational PureCoverage				●	●	●	●	●	●
Rational Purify®				●	●	●	●	●	●
Rational Quantify®				●	●	●	●	●	●
Rational Robot								●	●
Rational TestFactory								●	●
		E = Enterprise Edition M = Professional Data Modeler Edition RT = RealTime Edition SF = Rational SoDA for FrameMaker SW = Rational SoDA for Word TA = Test Agent と共に使用 U = UNIX Edition WEB = Web インターフェイスからのみアクセス可能							

詳細

Rational Suite とソフトウェア開発の原則の詳細については、『Rational Suite 入門』を参照してください。

統一モデリング言語の詳細については、<http://www.rational.com/uml> の UML 技術情報を参照してください (ただし、英語のみのご利用となります)。この Web サイトには、UML 関連情報、UML を使用する際のヒント、参考文献などが記載されています。

次の演習

次の章では、このチュートリアルについてももう少し詳しく学習し、使用するファイルのインストールとセットアップを行います。

このチュートリアル について

2

このチュートリアルでは、**Rational Suite** を使用してアプリケーションを計画、設計、実装、テストするための基本的な方法について説明します。また、ユーザーが理解を深められるようにさらに詳しい情報も記載されています。

前提事項

先に進む前に、コンピュータに次のソフトウェアがインストールされていることと、各ソフトウェアに有効なライセンスがあることを確認してください。

- Rational Suite バージョン 2003.06.00 英語版
- Microsoft Internet Explorer 5.5 サービス パック 1 または 2 以降
- Microsoft Word 2000 以降

これらの条件が満たされていない場合でも、このチュートリアルを読むことで一定の知識を得ることができますが、実際に演習を行うことはできません。

インストールされている Rational Suite ツールの確認

表 1 に、Rational Suite に含まれているツールを示します。

表 1 Rational Suite ツール

<input type="checkbox"/> Rational ClearCase LT	<input type="checkbox"/> Rational RequisitePro
<input type="checkbox"/> Rational ClearQuest	<input type="checkbox"/> Rational Robot
<input type="checkbox"/> Rational ProjectConsole	<input type="checkbox"/> Rational Rose
<input type="checkbox"/> Rational PureCoverage	<input type="checkbox"/> Rational SoDA for Word
<input type="checkbox"/> Rational Purify	<input type="checkbox"/> Rational TestManager
<input type="checkbox"/> Rational Quantify	<input type="checkbox"/> Rational Unified Process

演習: コンピュータにインストールされているツールを確認します。

ツールがインストールされているかどうかを確認するには

- 1 [スタート] ボタンをクリックします。
- 2 [プログラム] をポイントし、[Rational Software] メニューをクリックして、メニューにツール名が表示されるかどうかを確認します。

今後は、手順 1 と 2 を合わせて、「[スタート] ボタンをクリックし、[プログラム] をポイントします。次に、[Rational Software] をクリックします」のように記載します。

これにより、表 1 に記載されているツールのうち、インストールされていないツールを確認することができます。たとえば、購入したエディションによっては該当するツールが含まれていないことがあります。また、インストール時に特定のツールを除外した場合があります。

- 3 27 ページの表 1 で、インストールされているツールにチェック マークを付けます。
- 4 後から表 1 を参照できるように、27 ページにしおりを付けます。

ClassicsCD.com: チュートリアル用サンプル アプリケーション

このチュートリアルでは、大規模な開発プロジェクトの一部を実装します。Rational のツールとプロセスを使用して、要求の開発、ビジュアル モデルの操作、アプリケーションのテストを行います。

チュートリアルの背景の説明

このチュートリアルでは、クラシック音楽の CD をオンラインで販売する成長企業の ClassicsCD 社という会社にチュートリアルの読者が勤務しているという設定で説明をしていきます。読者が所属するチームは、ClassicsCD.com という Web サイトのバージョン 2 を担当し、Rational Suite を使用してこのバージョンの Visual Basic アプリケーションの計画、設計、実装、テストを行います。このチュートリアルでは、ClassicsCD.com に新しい拡張を 1 つ追加します。

チュートリアル用サンプル アプリケーションと関連ファイルのインストール

チュートリアルの演習を行う前に、必要なファイルをインストールしてセットアップする必要があります。Rational Suite チュートリアルのインストールには、次のディスク容量が必要です。

- ・ アプリケーションのダウンロードとセットアップに 10 MB
- ・ チュートリアル ファイルのダウンロードに別に 65 MB

メモ: Windows NT を使用している場合は、セットアップを正しく行うためには管理者権限が必要です。

演習: チュートリアル用のアプリケーションとファイルをインストールします。

- 1 チュートリアルのインストールを開始します。[スタート] ボタンをクリックし、[プログラム] をポイントします。次に、[Rational Software] をポイントし、[Download Rational Suite Tutorial] をクリックします。

自動的に Rational の Web サイトの Documentation ページ

(<http://www.rational.com/documentation>) が表示されます (ただし、英語のみのご利用となります)。

- 2 [View Documentation Online (Login required)] で、ドロップダウン メニューから [Rational Suite Tutorial] を選択します。
- 3 ログオンするか、Rational のメンバー プロファイルを新たに作成します。

ログオン後に Rational のホーム ページに戻ってしまった場合は、

<http://www.rational.com/documentation> を表示して、手順 2 と同じ方法でチュートリアルにアクセスしてください。

- 4 Rational Suite Tutorials ページで、「Version 2003 Tutorials」まで画面をスクロールし、[Rational Suite Tutorial ReadMe File] をクリックします。

このページには、チュートリアルのインストールとセットアップに関する情報と手順が記載されています。チュートリアルのプログラム ファイルとドキュメントをインストールするときは、これらの内容に従ってください。

- 5 内容を確認したら、[戻る] をクリックして Rational Suite Tutorials ページに戻り、必要に応じて「Version 2003 Tutorial」までスクロールします。
- 6 [Rational Suite Tutorial Manual] をクリックします。
画面上よりも印刷した方が見やすいので、PDF ファイルを開いたら印刷することをお勧めします。このファイルはハード ディスクに保存することもできます。
- 7 Rational Suite Tutorials ページに戻り、必要に応じて「Version 2003 Tutorial」まで画面をスクロールします。
- 8 [Rational Suite Tutorial v2003.06.00 Kit] をクリックします。
表示される指示に従って、コンピュータにチュートリアル ファイルをインストールします。
- 9 インストールが完了したら、Web ブラウザを閉じます。

入門

以上で、ClassicsCD 社の作業を開始する準備が整いました。Rational Suite がユーザーの開発環境に適していることを理解するために、チュートリアルを最初から最後まで実行することをお勧めします。各演習では示された手順に確実に従い、各章のすべての演習を終えたことを確認してから、次の章の演習に進んでください。各章の演習は、それより前の章で学習した情報とスキルに基づいて構成されています。

プロジェクトの登録

通常の開発環境では、プロジェクト管理者が Rational ソフトウェア環境の設定を行い、Rational Administrator 内に Rational プロジェクトを作成します。プロジェクト管理者は、Rational Administrator を使用して、Rational Suite に関連付けられた複数の成果物をグループ化します (たとえば、RequisitePro データベースと Rose モデル)。チュートリアルの読者は、ネットワーク上のプロジェクトに接続して作業ができるように、自分が使用しているコンピュータにユーザーとしてプロジェクトを登録します。このチュートリアルでは、専用の Rational プロジェクトが既に作成されています。

演習: このチュートリアルで使用する Rational プロジェクトを登録します。

- 1 [スタート] ボタンをクリックし、[プログラム] をポイントします。次に、[Rational Software] をポイントし、[Rational Administrator] をクリックして、Rational Administrator を起動します。
- 2 Rational Administrator の左側のペインで [Projects] を右クリックします。ショートカットメニューの [Register Existing Project] をクリックします (図 2 を参照)。

図 2 Rational Administrator によるプロジェクトの登録



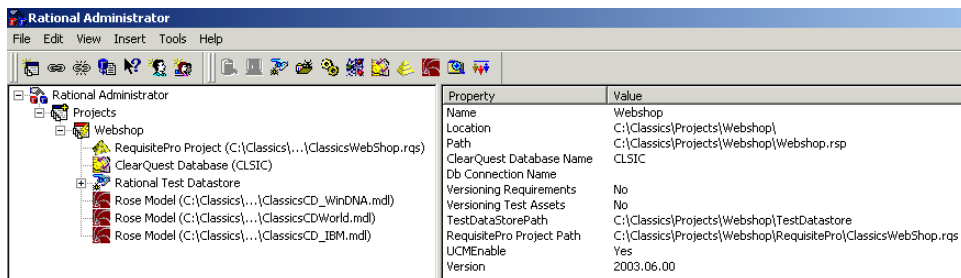
- 3 [Select Rational Project] ダイアログ ボックスで、C:\¥Classics¥Projects¥Webshop を指定します。
- 4 Webshop.rsp をクリックし、[開く] をクリックします。

ツリー ブラウザの [Projects] エントリの下に [Webshop] が追加されます。これがチュートリアルで使用するプロジェクトです。

- 5 Rational Administrator の左側のペインで [Webshop] を右クリックし、[Connect] をクリックします。

Webshop プロジェクトに関連付けられたすべてのアセットと開発情報が、[Webshop] エントリの下に表示されます (図 3 を参照)。これで、プロジェクトに正常に接続できました。

図 3 Rational Administrator によるプロジェクトへの接続



次のタスクのために Rational Administrator は開いたままにしておきます。

変更依頼データベースとプロジェクトの関連付け

Rational プロジェクトには、コンピュータにインストールされている Rational ツールによって収集されたソフトウェア開発情報を関連付けます。Rational プロジェクトは、データベースまたはデータストアに保存されている開発情報を参照します。データストアは、1 つまたは複数のデータベースで構成されます。Rational プロジェクトを作成したプロジェクト管理者は、そのプロジェクトをさまざまな Rational 製品のデータストア、データベース、プロジェクトに関連付けることができます。

プロジェクト管理者が開発アセットと Rational プロジェクトとを関連付けた後は、それぞれの製品の機能を使用して、1 つのデータベースまたはデータストアからほかのデータベースまたはデータストアにデータをリンクさせることができます。

このチュートリアルでは、次の製品を使用します。

- ClearQuest 変更依頼データベース: プロジェクトの変更依頼 (障害と拡張依頼) が保存されます。
- RequisitePro データベース: プロジェクトのビジネス要求とシステム要求が保存されます。
- Rational Test データベース: プロジェクトのテストに関する情報 (テストアセット、ログ、レポート) が保存されます。
- Rational Rose モデル: プロジェクトのビジュアルモデルで構成されます。

演習: 変更依頼データベースをプロジェクトに関連付けます。

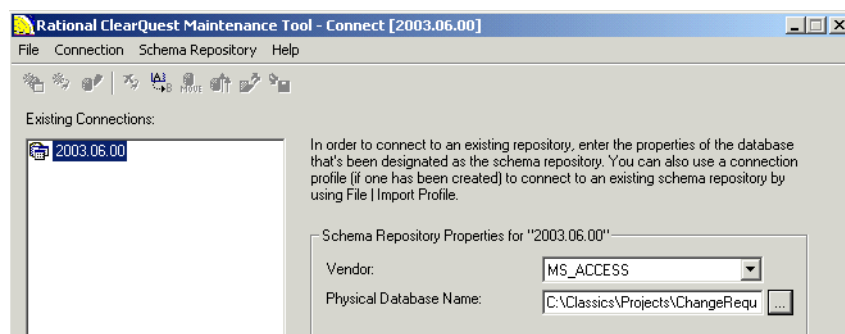
- 1 Rational ClearQuest がコンピュータにインストールされているかどうかを確認します。
27 ページの 表 1 を参照してください。インストールされている場合は、以下の手順で作業を進めます。インストールされていない場合は、このチュートリアルで ClearQuest を使用できません。
- 2 Rational Administrator で、[Tools] メニューの [Rational ClearQuest Maintenance Tool] をクリックします。
- 3 [ClearQuest Maintenance Tool] メニューバーで、[Connection] メニューの [New] をクリックします。

メモ: 以下の手順では、このチュートリアル用のスキーマ リポジトリへの接続方法を説明します。ClearQuest で別のスキーマ リポジトリを使用している場合は、チュートリアル終了後にそのリポジトリに接続し直す必要があります。不明な点がある場合は、プロジェクト管理者に問い合わせてください。

- 4 [Schema Repository Properties for "2003.06.00"] を見つけます。
 - a [Vendor] ボックスの値が [MS_ACCESS] になっていることを確認します。
 - b [Physical Database Name] ボックスで ... (参照ボタン) をクリックし、C:\Classics\Projects\ChangeRequests に移動します。
 - c CQMaster.mdb をクリックし、[開く] をクリックします。

以上で、このチュートリアルで使用する変更依頼データベースのプロパティを入力できました (図 4 を参照)。

図 4 変更依頼データベースの関連付け



- d [Finish] をクリックします。
- 5 [Status] メッセージで、変更依頼データベースとプロジェクトが接続されていることを確認し、[Done] をクリックします。

6 ClearQuest Maintenance Tool を閉じます。

7 Rational Administrator を終了します。



メモ: 通常は、これらのタスクはプロジェクト管理者が行います。ただし、このチュートリアルでは、ユーザーがこれらのタスクを実行してください。

アプリケーションの使用

実際のオンライン ストアでは、Web ページはデータベースに格納されている情報やユーザー入力への反応によって動的にロードされます。チュートリアルで使用するアプリケーション ClassicsCD.com は、静的な Web ページで構成されています。ページがユーザー入力に応答して変化することはありません。通常のプロジェクトでは、静的なページを使用してプロトタイプを作成し、その後で動的なページを使用して変更を加えます。

このチュートリアルでは、手順に確実に従ってください。定められた手順を守らないと、アプリケーションが正しく動作しないことがあります。ここで使用する Web ページは静的なもので、実際に入力に応答することはありません。

演習: ClassicsCD.com を起動して CD を 2 枚注文します。

- 1 アプリケーションを起動するには、Windows エクスプローラで
C:\¥Classics¥ClassicsCD_com_sites¥v1 に移動し、index.htm をダブルクリックします。
Web ブラウザに ClassicsCD.com の最初のページが表示されます。
- 2 ClassicsCD.com のホーム ページで、[Explore our storefront] をクリックします。
- 3 storefront のページで [Catalog] をクリックします。
- 4 アルファベット順に表示されている作曲家の中からベートーベンを探し、[Beethoven: Symphonie Nr. 5] をクリックして、アルバムの詳細を表示します。
- 5 アルバムのページで、 (ショッピング カート ボタン) をクリックして、このアルバムを注文します。
CD Catalog ページが再び表示されます。
- 6 ページの上部で、[Bach: Brandenburg Concertos 1 + 3] の横にある  をクリックします。

演習: 購入を確定します。

次に、購入を確定し、フィードバックを ClassicsCD 社に送信します。

- 1 ページの左側で、[Shopping Cart] をクリックします。
- 2 左側にある Shopping Cart ページで、[Cashier] をクリックします。
注文を確定するには、ログインする必要があります。

- 3 [CustomerID] ボックスと [Password] ボックスの両方に「jmoore」と入力し、[Submit] をクリックします。

パスワードを Windows に保存するかどうかを確認するメッセージが表示されたら、[いいえ] をクリックします。

Checkout ページに注文の一覧が表示されます。ただし、出荷日は表示されません。

- 4 ページをスクロールして、下部にある [Place Order] をクリックします。
- 5 ページの下部にある [Your Feedback] をクリックして、ClassicsCD 社へのフィードバックを準備します。
- 6 フィードバック フォームの [Dear ClassicsCD.com] ボックスで、次の操作を行います。
 - a 「When I place an order, I want to know when my order will ship.」 と入力します。
 - b [My e-mail] ボックスに、「jmoore@clicker.com」と入力します。
 - c [Send!] をクリックします。
- 7 ClassicsCD.com を終了し、必要に応じて Windows エクスプローラも終了します。

ビルド内容の決定

ClassicsCD 本社では、マーケティング部の社員がこのフィードバックを受信し、変更依頼を管理するツールの ClearQuest にその内容を入力します (図 5 を参照)。

図 5 ClearQuest による拡張依頼の表示

The screenshot displays the ClearQuest application window. The top menu bar includes 'History', 'Requirements', 'ClearCase', and 'Unified Change Management'. Below this, a sub-menu bar shows 'Main', 'Analysis', 'Customer', 'Notes', 'Resolution', and 'Attachments'. The main form area contains the following fields: 'ID:' with the value 'CLSLC00000035', 'State:' with the value 'Assigned', 'Headline:' with the value 'Need to display estimated ship date', 'Suite Project:' with a dropdown menu showing 'Webshop', and 'Customer Priority:' with a dropdown menu showing 'Medium'. The 'Description:' field is a large text area containing the text: 'It would be nice to see when an order will ship. Customers are unsure what to expect when order is completed. Customer Support has been inundated with calls from confused customers. This enhancement will help reduce the burden on our support staff.' On the right side of the window, there are buttons for 'OK', 'Cancel', 'Print Record', and an 'Actions' dropdown menu.

このチュートリアルでは、Rational Suite を使用してこの拡張依頼を実装する方法を学習します。新しい要求の開発、ビジュアル モデルの操作、アプリケーションのテストを行います。

「第 11 章 次の反復の計画」では、ClearQuest を使用します。

このチュートリアルの使い方

Rational Suite がユーザーの開発環境に適していることを理解するために、チュートリアルを最後まで実行するか、自分の役割に最も合っていると思われる章を選んで読んでください。チュートリアルを最初から最後まで実行する場合は、各章の演習はそれより前の章で学習した情報とスキルに基づいて構成されているので、必ず各章のすべての演習を終わらせてから次の章の演習に進むようにします。チュートリアルを読むだけの場合は、「第 1 章 Rational Suite について」とこの章から読み始めることをお勧めします。

表 2 に、チュートリアルを読むだけの場合にお勧めする章を示します。

表 2 このチュートリアルの使い方

役割	主なタスク	推奨の章
アナリスト	システムの機能の決定 ユーザー ニーズの提示 要求の決定と管理	3 Rational Unified Process について 4 プロジェクト成果物の変更管理 5 要求の作成 8 プロジェクト ステータスの伝達 11 次の反復の計画
開発者	システムの動作の決定 アーキテクチャの定義 コードの作成、変更、管理、 テスト	3 Rational Unified Process について 4 プロジェクト成果物の変更管理 5 要求の作成 7 機能拡張のモデリング 8 プロジェクト ステータスの伝達 9 信頼性テスト 11 次の反復の計画
テスト担当者	要求への適合の確認 テストの作成、管理、実行 結果の報告と修正の確認	3 Rational Unified Process について 4 プロジェクト成果物の変更管理 6 テスト計画 8 プロジェクト ステータスの伝達 9 信頼性テスト 10 機能テスト 11 次の反復の計画
マネージャーまたは プロジェクト リーダー	プロジェクトにおけるリスクの 識別と管理 チームの進行状況の監視 各反復の計画	3 Rational Unified Process について 4 プロジェクト成果物の変更管理 5 要求の作成 6 テスト計画 7 機能拡張のモデリング 8 プロジェクト ステータスの伝達 9 信頼性テスト 10 機能テスト 11 次の反復の計画

まとめ

この章の学習内容

この章では、次のことを学習しました。

- コンピュータにインストールされている Rational Suite ツールの特定。
- チュートリアルで使用するファイルのインストールとセットアップ。
- コンピュータにインストールされている Rational ツールによって収集されたソフトウェア開発情報を Rational プロジェクトに関連付ける方法。

次の演習

このチュートリアルでは、Classics CD Webshop プロジェクトを使用して拡張依存を実装します。次の章では、Rational Unified Process について説明します。このプロセスを使用して、後の章での作業について学習します。

それでは、次の章に進んでください。

Rational Unified Process について

3

この章では、Rational Unified Process (RUP) について説明します。この章では RUP の使用に慣れると共に、次の章で行う作業のためのガイドラインについて学習します。

対象読者

この章は、ソフトウェア開発チームのすべてのメンバーを対象としています。

概要

この章では、Rational Unified Process を使用します。使用しているコンピュータに RUP がインストールされているかどうかを確認するには、27 ページの「表 1 Rational Suite ツール」で記入したツール チャートを参照してください。

RUP がインストールされていない場合は、この章を読むことで多くのことを学習できますが、演習を行うことはできません。

RUP がインストールされている場合は、[スタート] ボタンをクリックし、[プログラム] をポイントします。次に、[Rational Software] をポイントし、[Rational Unified Process] をクリックします。

既にコンピュータに Java プラグインがインストールされていても、Java プラグインのインストールを要求するメッセージが表示される場合があります。これは、インストールされているプラグインのバージョンと RUP に必要なバージョンとが異なるためです。このバージョンのプラグインがインストールされていない場合でも、この項を読むことで多くのことを学習できますが、演習を行うことはできません。

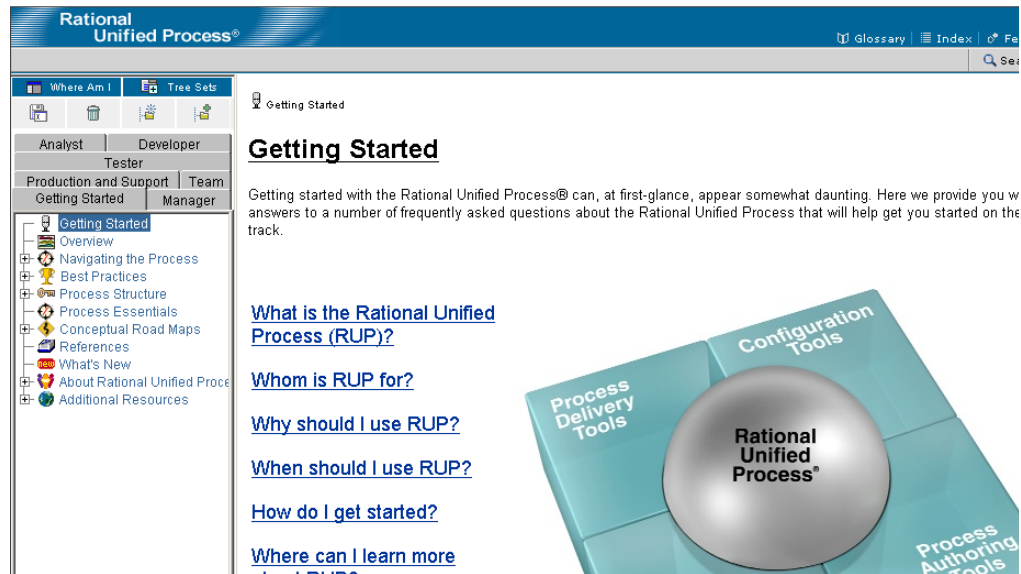
RUP 用の適切な Java プラグインがインストールされた後に、署名済みアプレットの信頼が求められることがあります。この警告メッセージが今後表示されないように、[常に] を選択することをお勧めします。

その後、Web ブラウザに **Getting Started** ページが表示されます。

左側のペインには、RUP ツリー ブラウザが表示されます。RUP の要素は、ツリー セットで表示され、ソフトウェア開発プロセスのトピック名が付けられています。選択すると、各ツリー セットがツリー ブラウザに表示されます。右側のペインには、選択した要素の内容が表示されます (図 6 を参照)。

選択した要素を見やすいようにウィンドウを調整できます。

図 6 Rational Unified Process 環境



Rational Unified Process (RUP) とは

RUP は、ソフトウェア開発用のプロセス フレームワークであり、次の目的で使用します。

- 開発チーム全体における開発上の責任範囲の調整
- 高品質ソフトウェアの作成
- ユーザーのニーズの反映
- 設定されたスケジュールと予算内での作業
- 新技術の利用

RUP は、世界中の数多くの企業で大小さまざまな規模の開発作業に使用されています。これは、チームに最善の実践原則を通じて、管理された反復型ソフトウェア開発をもたらすためです。

RUP は、Web ブラウザで参照できるオンライン ガイドと知識ベースとして実装され、開発ライフサイクルの各フェーズにおけるガイドライン、テンプレート、ツール メンター (特定の Rational ツールにガイドラインを適用する手順) を提供します。また、RUP をカスタマイズして次のことを行うこともできます。

- プロジェクトの各段階で必要となるプロセス コンポーネントのみを使用する
- 独自のプロセス コンポーネントを作成する
- 同僚や上司と実践的手法を交換する

Rational Unified Process と Rational Suite

Rational Unified Process を使用すると、個人やチームの作業効率を向上させることができます。RUP はカスタマイズ可能なフレームワークであり、ソフトウェア開発のための最善の実践原則についての共通の視点を開発チームに提供します。Rational Process Workbench を使用すると、プロジェクトやチームの特定のニーズに合わせて RUP をカスタマイズできます。

Rational Suite に RUP を導入していない場合でも、プロジェクトを成功させることは可能です。また、RUP を、Rational Suite やそのコンポーネント ツールを使用しないプロジェクトに使用することもできます。

RUP に沿って作業しない場合でも、RUP をソフトウェア開発に関する情報ソースとして使用できます。たとえば、ヘルプには、UML の概念の理解に役立つトピックが含まれています。

このチュートリアルでは、Rational Unified Process に沿って作業を進めます。

仕組みについて

Rational Unified Process は、次の各作業分野についてソフトウェア開発ライフサイクルのあらゆる側面でユーザーを支援します。

- ビジネス モデリング
- 要求
- 分析と設計
- 実装
- テスト
- 展開
- 構成、変更管理
- プロジェクト管理
- 環境管理

RUP は、ソフトウェア開発プロジェクトで適用されるこれらの各作業分野の情報とガイダンスを提供します。

演習：RUP 内をナビゲートする方法について学習します。

1 ツリー ブラウザで、[Getting Started] ツリー セットを選択します。

2 [Getting Started] ツリー セットの [Navigating the Process] をクリックします。

Web ブラウザに Navigating the Process ページが開き、RUP ブラウザ環境の要素が表示されます。

3 任意の場所をクリックすると、ナビゲート方法の概要が表示されます。

4 RUP のナビゲート方法を確認し終わったら、Navigating the Process ページに戻ります。

プロセスの概要

Rational Unified Process の概要は、RUP が理解しやすいようにカラフルな図で示されています。

演習 : Rational Unified Process の概要を学習します。

- 1 [Getting Started] ツリー セットの [Overview] をクリックします。

Rational Unified Process の概要図が表示されます (図 7 を参照)。

図 7 Rational Unified Process の概要図

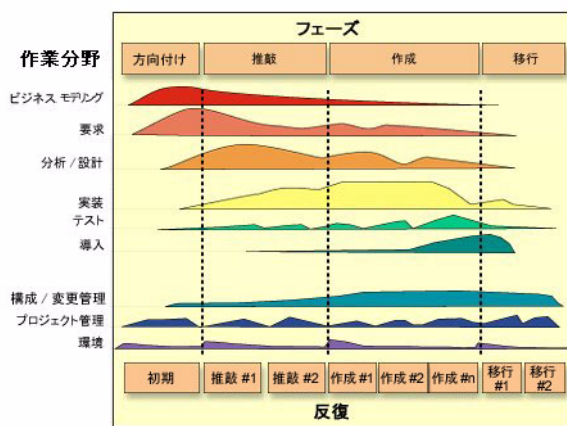


図 7 には、ソフトウェア開発がフェーズごとに最適に構成され、各フェーズが一連の反復で実行される様子が示されています。各フェーズを通じ、ソフトウェア開発においてそれぞれ異なる役割を持つプロジェクトチームメンバー (アナリスト、開発者、テスト担当者、マネージャー) が、1 つまたは複数の作業分野のアクティビティを実行します。図からは、各作業分野における重点が反復ごとに異なっていることが分かります。

たとえば、要求に関連する作業の多くは開発サイクルの初期段階に発生し、プロジェクトをとおして続きます。これに対し、テストはプロジェクトの初期段階でも開始できますが、通常は作成の最終段階で最も集中的に行われます。

RUP の概要図は、次の重要な概念を表しています。

- Rational Unified Process のソフトウェア ライフサイクルは時間の経過に応じて 4 つのフェーズに分けられ、それぞれの終了が主要マイルストーンとなる。つまり、それぞれのフェーズは 2 つの主要マイルストーンにはさまれた期間を表します。
- 開発反復は、すべての作業分野を網羅し、製品のリリースにつながる 1 本の完全な経路である。

- 2 必要に応じて、概要図内の **Phases** や **Iterations** という語をクリックすると、詳しい説明が表示されます。

概念についての説明を読み終えたら、Web ブラウザの [戻る] ボタンをクリックして **Rational Unified Process: Overview** ページに戻ります。

- 3 [Getting Started] ツリーセットのほかの要素をクリックし、RUP に関するほかの説明を読みます。
- 4 RUP を開いたままにしておきます。

主な概念

Rational Unified Process では、最も重要な概念の要約を表示することができます。

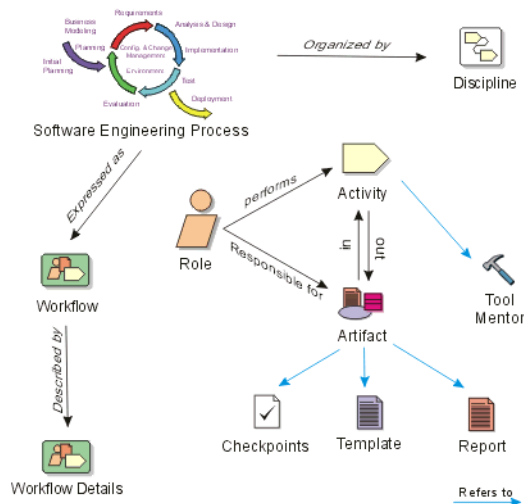
演習：主な概念について学習します。

- 1 [Getting Started] ツリーセットの [Process Structure] をクリックします。

新しい Web ページが開きます。ページの上部に、RUP の主な概念の関係を示す図が表示されます (図 8 を参照)。

図 8 Rational Unified Process における主な概念

The Basic Elements of RUP



- 2 概念の詳細を参照するには、画像内のシンボルをクリックします。次に、概念の説明を示します。
- 作業分野には、ソフトウェア アセットの特定の組み合わせを作成するすべてのアクティビティが表示されます。RUP は、すべての役割、ワークフロー、アクティビティ、成果物などの要約を含め、概要レベルで開発作業分野を記述します。
 - 役割では、プロジェクト チームにおける個人またはグループの活動と責任を定義します。1 人のユーザーがプロジェクトの流れの中でいくつかの役割を果たすことができます。これとは逆に、複数のユーザーがプロジェクトの中の 1 つの役割を果たすこともできます。成果物は、これらの役割によって作成されます。
 - ワークフローとは、共通の目標に向かって作業者が行う一連のアクティビティのことです。ワークフロー図は、関連アクティビティの組み合わせを表す高レベルなマップとして使用できます。アクティビティ間の矢印は、アクティビティ間の作業の一般的な流れを示します。
 - アクティビティとは、特定の役割によって実行される作業の組み合わせのことです。アクティビティは、レシピのように、成果を作成するための一連の手順がセットになったものです。
 - 成果物とは、ある役割によるアクティビティの結果として生成されるものを示します。RUP では、1 つのアクティビティによって生成された成果物は、多くの場合、別のアクティビティへの入力として使用されます。成果物には、大きいものや小さいもの、複雑なものや単純なもの、形式的なものや非形式的なものなど、さまざまな形態のものがあります。成果物の例としては、テスト計画、開発構想書、システム アーキテクチャのモデル、ビルドを自動化するスクリプト、アプリケーション コードなどがあります。
- 3 図から下にスクロールし、最初の **Software Engineering Process** セクションで **Rational Unified Process** の要約を参照します。

ワークフローについて

演習：要求作業分野について理解します。

RUP は、既存のシステムをどのように拡張すべきかのガイドラインを示します。このチュートリアルでは、RUP のガイドラインを使用して、オンラインストア アプリケーション ClassicsCD.com を改良するための作業を行います。

ここでは、要求に関するワークフローの詳細を学習します。

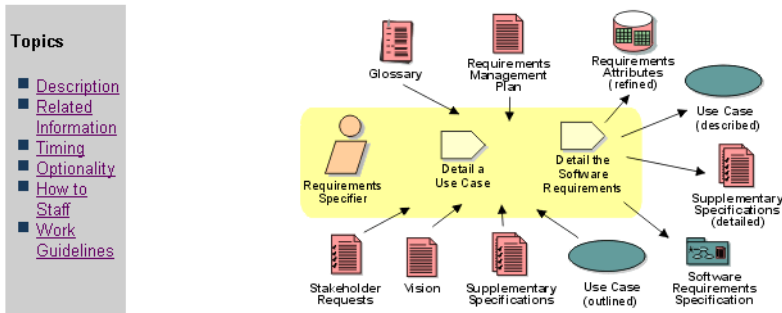
- 1 ツリー ブラウザで、[Analyst] ツリー セットを選択します。
- 2 [Analyst] ツリー セットで [Requirements]、[Workflow] の順にクリックし、Requirements: Workflow ページを表示します。
- 3 Requirements: Workflow ページで、[Refine the System Definition] をクリックして、ワークフローの詳細を表示します (図 9 を参照)。

ワークフローの詳細 (図 9 で示すような詳細) には、関係する役割、入力として使用された成果物、結果として得られる成果物、ワークフロー全体の中でその部分を構成しているアクティビティが示されます。これらの要素の詳細については、詳細ページでその要素をクリックしてください。

図 9 RUP における要求ワークフローの詳細

Workflow Detail: Refine the System Definition

The purpose of this workflow detail is to further refine the requirements in order to capture the consensus understanding of the system definition.



アクターとユースケースの使用

システムを設計したり拡張したりする場合、Rational Unified Process では、チームのメンバーがそのシステムの高レベルな動作について共通の認識を持つところから開始することを推奨しています。これは、アクターとユースケースを特定することによって行います。

- ・ アクターとは、そのシステムと連携して動作するエンティティのことを言います。多くの場合、人がアクターになります (たとえば、販売員や管理者)。また、外部のハードウェアシステムやソフトウェアシステムがアクターになることもあります (たとえば、レジや金融機関が提供するクレジット カード確認システム)。
- ・ ユースケースは、アクターがそのシステムをどのように使用したり、やり取りしたりするのかを表します。形式的に言えば、ユースケースとは、特定のアクターに対してシステムがどのようなサービスや機能を提供するのかを記述するものです。ユースケースは、アクターに対し目に見える形で価値を生み出す一連のアクションを完全に記述することによって定義します。

ユースケースは、RUP と UML の重要な概念です。ユースケースは各開発チーム間のコミュニケーションを向上させるので、問題を的確に把握して解決すると共に、ユーザーが必要とするシステムを定義することができます。

演習: ユースケースについて学習します。

- 1 要求ワークフローの詳細内で、[Detail a Use Case] をクリックします。

ユースケースの記述方法を示すページが表示されます。これには、アクティビティを開始するために必要な成果物と、アクティビティを実行した結果得られる成果物に関する詳細が含まれます。さらに、アクティビティ自体についての詳細も表示されます。

- 2 Resulting Artifacts セクションの [Use Case] をクリックして、記述方法も含めたユースケースの詳細について参照します。

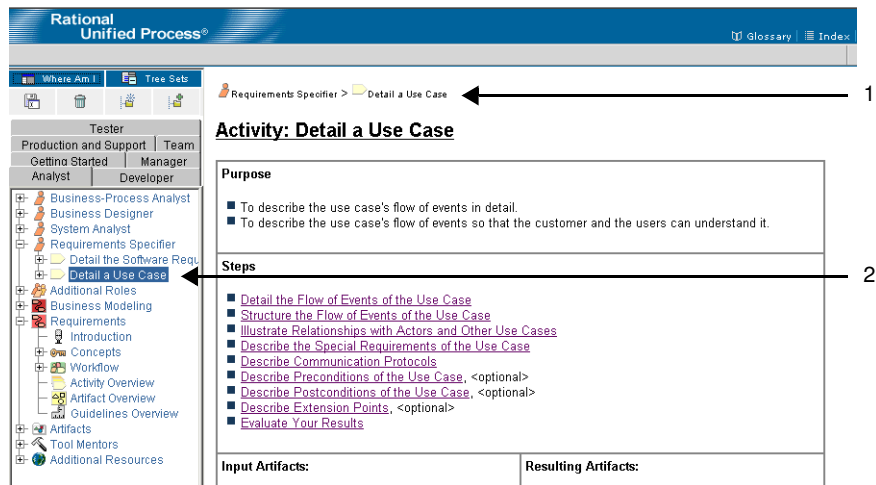
この新しいページには、ユースケースの使用方法、代表的なユースケースのアウトライン、責任者など、ユースケースの概要が成果物として分かりやすく表示されます。

このページには、ユースケース テンプレートへのリンクがあります。ユースケースを作成する際は、ユースケース開発における整合性と完全性を確保するために、このテンプレートを使用するか、所属するグループが作成したテンプレートを使用することをお勧めします。テンプレートを使用することによって、すべての利害関係者が重要なプロジェクト情報を参照し、理解することができます。

テンプレートを確認してください。ユースケース ドキュメントは、このチュートリアルの後ろの方で使います。

- 3 使用しているブラウザの [戻る] ボタンをクリックして、Activity: Detail a Use Case ページに戻ります。
- 4 この時点で RUP 階層内での現在の位置を確認するには、次のいずれかの操作を実行します。
 - 表示されているページの上部を確認します。ここには、現在位置を示す RUP ビジュアルが表示されます (図 10 の 1 を参照)。表示された RUP ページの位置を確認する場合は、この機能を使用します。
 - または、ツリー ブラウザで、[Where am I] をクリックします。ツリー ブラウザが更新され、現在位置が強調表示されます。下の例では、現在位置は、[Analyst] ツリーセットの [Requirements Specifier] の [Detail a Use Case] です (図 10 の 2 を参照)。

図 10 RUP 内での現在位置



ユース ケースの詳細については、「第 5 章 要求の作成」を参照してください。

ツール メンター : Rational ツールによるプロセスのサポート

Rational Unified Process は、ソフトウェア開発のすべてのフェーズについてのガイドラインを示します。RUP では、ツール メンターを使用して Rational ツールの使用方法を説明します。ツール メンターを使用すると、1 つまたは複数のツールを使用して、プロセスの手順を実行する方法、または特定の成果物やレポートを作成する方法について詳細を参照することができます。

次の演習では、最初に行う作業であるユース ケースの定義の手順をツール メンターで表示します。

演習 : ツール メンターを操作します。

ツール メンターを読んで、RUP がどのように Rational ツールに統合されているかを確認します。

- 1 **Activity: Detail a Use Case** ページで画面を下にスクロールし、ツール メンターのリストを表示します。

- 2 **[Detailing a Use Case Using Rational RequisitePro]** をクリックします。

ツール メンターが表示され、ツールの目的、概要、使用手順が表示されます。

- 3 **[Tool Steps]** の画面をスクロールして、**[Step 3]** の **[Create requirements in the detailed Use-Case Specification]** をクリックします。

表示された手順をよく読みます。この手順の一部は、このチュートリアルで実行します。

基本的なことについて

Rational Unified Process は、このチュートリアルで ClassicsCD.com について行う作業を含め、一定の作業を行うためのガイドラインを示します。

演習 : Process Essentials について学習します。

- 1 **[Getting Started]** ツリー セットの **[Process Essentials]** をクリックします。
- 2 各項目を参照して、RUP の重要な要素について学習します。

演習 : Web 用アプリケーション開発について学習します。

- 1 **[Getting Started]** ツリー セットで、**[Conceptual Road Maps]**、**[Developing e-business Solutions]** の順に展開します。
- 2 各項目を参照して、Web アプリケーション開発について学習します。

まとめ

詳細

Rational Unified Process の詳細については、ツリー ブラウザで対象の項目を選択し、その内容を参照してください。

クリーンアップ

RUP を終了します。または、RUP のウィンドウを最小化して、このチュートリアルで後述する項目を詳しく学習する際の補足手段として使用することもできます。

この章の学習内容

この章では、次のことを学習しました。

- Rational Unified Process は、ソフトウェア開発の最善の実践原則を実現するためのカスタマイズ可能なガイドラインを示します。RUP は Rational Suite に含めることを推奨しますが、任意です。
- ワークフローは、一連の関連アクティビティを定めたもので、目標達成に重点を置いています。各アクティビティにおいては、ある役割が、前のアクティビティで作成された成果物から別の成果物を作成します。
- アクター (システムと連動するユーザーと外部システム) とユース ケース (システムがアクターに提供するサービス) は、要求段階の初期で定義します。
- ツール メンターを使用すると、適切な Rational ツールで RUP アクティビティを実行する際の明確な手順を参照できます。

次の演習

次の章では、ソフトウェア開発ライフサイクルを通じて変更を効果的に管理する上で、Rational ClearCase LT がどのように役立つのかを学習します。

プロジェクト成果物の 変更管理

4

この章では、Rational ClearCase LT と統一変更管理 (UCM) を使用して行うソフトウェア開発時の変更管理方法について学習します。ClearCase LT と UCM を組み合わせて使用することにより、要求からリリースまで複数の反復を通じてプロジェクト成果物の変更を適切に管理することができます。

対象読者

この章は、チームのすべてのメンバーを対象としています。

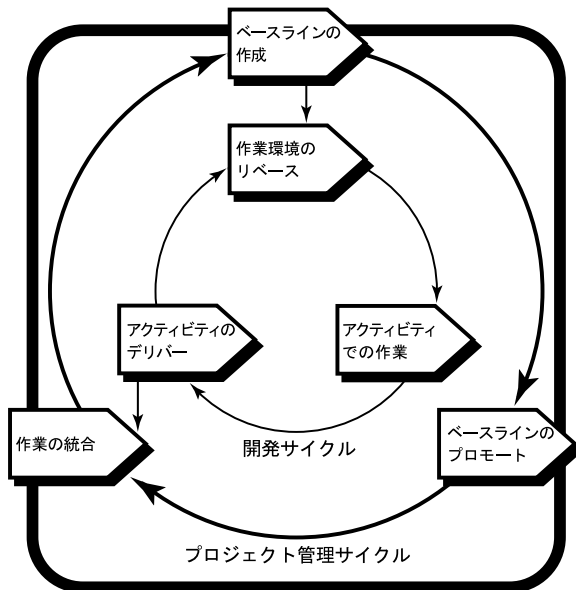
統一変更管理とは

Rational では、要求からリリースまでのソフトウェア システム開発における変更を管理する最適な手段として統一変更管理 (UCM) を提供しています。UCM は、主に次の概念に基づいています。

- UCM アクティビティは、開発タスクの完了に必要な作業を表します。UCM のアクティビティは、障害や拡張依頼を含むさまざまなソースから派生します。
- 成果物とは、ソフトウェア開発ライフサイクルにおいて、アクティビティを実行した結果として作成、変更、使用されるものです。Rational Unified Process (RUP) においては、1 つのアクティビティによって作成された成果物が、ほかのアクティビティへの入力として使用されることがよくあります。概念上の成果物としては、要求、ビジュアル モデル、テスト ケース、ソース コード、ドキュメント、プロジェクト計画などがあります。成果物はプロジェクトの成功には不可欠なもので、構成管理またはバージョン管理の対象にする必要があります。通常、成果物はファイルまたはファイル セットの形をとります。

UCM の重要な特長は、ソフトウェア開発進行を計画、追跡するために使用するアクティビティと、ソフトウェア アプリケーションを作成、設計、ビルドするために使用する成果物を統合することができるという点です。図 11 に、UCM を使用した変更管理の代表的な例を示します。

図 11 一般的な UCM のワークフロー



UCM ツール

UCM モデルの重要な側面のひとつは、プロジェクト進行の計画と追跡に使用するアクティビティと、変更対象の成果物を統合することができるという点です。UCM モデルは、プロセスとツールの両方によって作成されます。**Rational ClearQuest** と **Rational ClearCase LT** は、UCM をサポートするツールを提供します。たとえば、**ClearQuest** ではプロジェクトのタスク、障害、拡張依頼（これらをまとめてアクティビティと呼びます）を管理し、**ClearCase LT** ではソフトウェアプロジェクトによって作成された成果物を管理します。両者を組み合わせて使用することにより、ソフトウェアチームは、ソフトウェア開発ライフサイクルを通じて変化する要求や複雑になる開発を効率よく管理することができます。

Rational ClearQuest: ソフトウェア開発に関連する変更アクティビティ（拡張依頼、障害レポート、ドキュメントの変更など）を管理するツールです。また、プロジェクトの進行を追跡してすべての利害関係者に知らせるためのグラフやレポートを作成する機能も備えています。

Rational ClearCase LT: ビルトイン UCM 開発プロセスを使用して、要求、ビジュアルモデル、ソースコードを含むすべてのソフトウェアプロジェクトファイルに対する変更の追跡と管理を行います。

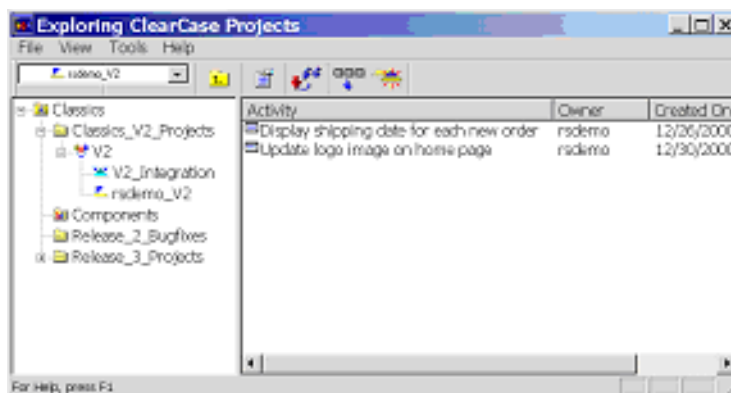
UCM におけるツールの使用 : ClearQuest と ClearCase LT

RUP でも説明されているように、通常、UCM は ClearQuest、ClearCase LT と組み合わせて次のように使用します。

- 1 (1 回のみ) プロジェクト マネージャーまたは管理者が、Rational ソフトウェアをインストールします。インストールを行ったユーザーが ClearQuest と ClearCase LT の環境設定を行い、Rational プロジェクトを作成します (Rational プロジェクトは Rational ツールで作成、保存されたデータを関連付け、プロジェクトとデータの統合を可能にします)。また、プロジェクト マネージャーは、UCM プロジェクトを作成して、Rational プロジェクトとの関連付けも行います。
- 2 (1 回のみ) プロジェクトに参加し、プロジェクトに対して自分の識別情報を指定します。これにより、個人用ワークスペース (開発ストリームと開発ビューで構成される) が作成されます。チーム全体が使用できるワークスペースにもアクセスできるようになります。この共用ワークスペースは、インテグレーション ストリームで構成され、付随するインテグレーション ビューを作成して独自に使用できます。

ユーザーは、ビューを使用して、ワークスペースの各エレメントの 1 つのバージョンを選択します。UCM では、ストリームによってこれらの設定手順がビューに表示され、アクティビティを追跡できます。プロジェクトに参加すると、UCM によってストリームが自動的に設定され、ワークスペースに最適なバージョンが表示されます (図 12 を参照)。

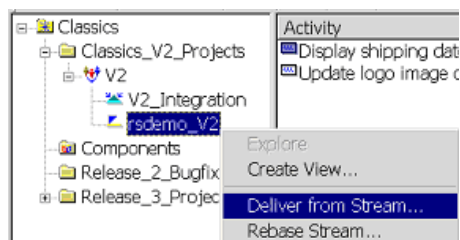
図 12 UCM による ClearCase LT プロジェクトの表示



- 3 プロジェクト マネージャーは、ClearQuest を使用して、ユーザーにアクティビティを割り当てます。

- 4 ユーザーは **ClearQuest** クエリーを実行して、自分に割り当てられたアクティビティを検索できます。検索結果は、リスト形式で表示されます。このリストで、操作するアクティビティを決定します。
- 5 通常どおり成果物を処理します。個人用ワークスペースで **ClearCase LT** を使用して、次の操作を行います。
 - 成果物をチェックアウトします。**ClearCase LT** で成果物をチェックアウトする場合は、操作するアクティビティを指定します。指定したアクティビティと関連付けられた変更セット (変更された成果物のリスト) が、バックグラウンドで追跡されます。
 - 変更内容を編集してから確認します。
 - 成果物をチェックインします。成果物をチェックインしても、この成果物は個人用ワークスペースの内部にあります。この成果物をデリバリーするまでは、変更はほかのメンバーには公開されません (手順 6 を参照)。
- 6 アクティビティでの操作が完了したら、アクティビティ全体に変更内容をデリバリーします (図 13 を参照)。変更セットは追跡されているため、デリバリーする成果物のリストを指定する必要はありません。変更内容をデリバリーすると、変更内容はインテグレーション ストリームを介してほかのメンバーに公開されます。プロジェクト マネージャーが設定したポリシーによっては、変更内容をデリバリーすると、アクティビティが終了する場合があります。

図 13 ClearCase LT による UCM プロジェクトへの変更内容のデリバリー



- 7 開発者が一連のアクティビティをデリバリーしたら、プロジェクト マネージャー はベースラインを作成します。このベースラインには新しいアクティビティや変更された成果物が含まれ、すべての開発者にとって共通の新しい開始点になります。プロジェクトによっては、新しいベースラインが定期的に作成されます (毎日作成される場合もあります)。
- 8 テストまたはほかのレビュー プロセスによってベースラインの変更が承認されると、プロジェクト マネージャーはそのベースラインを推奨ベースラインとしてプロモートします。

- 9 次のアクティビティを操作する場合に最新の推奨ベースラインから開始できるように、開発ストリーム (ワークスペース) をリベースします。手順 4 から再度実行して、次に操作するアクティビティを選択します。

開発を行う組織の構造によっては、UCM 用の Unified Process の定義に従って、チームでこれらの役割を別の名前と呼んだり、さまざまな役割の責任を複数のユーザーではなく 1 人のユーザーに割り当てたり、1 つの役割の責任を複数のチーム メンバーで分担したりすることも考えられます。チームの構造に関係なく、UCM は有効に使用することができます。UCM のモデルは、構成、変更管理のための基本的プロセスに従っており、ClearCase LT によって成果物とアクティビティ監視の大部分を自動化することができます。

Web 開発のためのコードとコンテンツの統合

ソフトウェア アプリケーションと Web サイトのサイズが大きくなり、複雑さと戦略価値が増すにつれて、それに対する変更を管理する必要性も大きくなります。Web ベースのソフトウェア開発アプリケーションと従来型のソフトウェア開発アプリケーションに共通するこれらの特性とは別に、両者の間には大きな相違があります。たとえば、Web アプリケーションには次のような特徴があります。

- 変化のペースが極めて速い。
- 技術的役割と非技術的役割の両方を含め、Web サイトには多くの利害関係者が関わっている。
- 変更の頻度が高く、利害関係者の範囲や種類も多いため、エラーの可能性も大きくなる。

このチュートリアルでは主に従来型のソフトウェア プロジェクトにおける開発概念について述べていますが、説明されている考え方、ツール、プロセスの多くは Web アプリケーションにも適用できます。

まとめ

詳細

ClearCase LT の使用方法については、ClearCase LT のチュートリアルを参照してください。
[スタート] ボタンをクリックし、[プログラム] をポイントします。次に、[Rational Software] をポイントし、[Rational ClearCase]、[Tutorial] の順にクリックします。[トピックの検索] ウィンドウで、[Tutorial] をダブルクリックし、[Rational ClearCase Tutorial] をダブルクリックします。

Rational ClearCase LT と UCM を Rational Suite と共に使用方法については、『Rational Suite 統一変更管理 (UCM) ユーザーズ ガイド』を参照してください。

Rational ClearCase LT の一般的な情報については、UCM と組み合わせるかどうかにかかわらず、『Rational ClearCase 入門』を参照してください。

この章の学習内容

この章では、次のことを学習しました。

- UCM は、要求からリリースまでのソフトウェア開発で、ソフトウェア チームによる変更管理を支援します。
- ClearCase LT と ClearQuest は UCM の基礎です。ClearCase LT は、ソフトウェア開発プロジェクトに関連する成果物を管理します。ClearQuest は、プロジェクト アクティビティを管理します。また、プロジェクトの進行を追跡してすべての利害関係者に知らせるためのグラフやレポートを作成する機能も備えています。
- UCM では、チーム メンバーは ClearCase LT を使用してソース管理下にある成果物を管理し、個人用開発ワークスペースにおいてアクティビティに関する作業を行います。アクティビティ完了後に、変更した成果物をインテグレーション ストリームにデリバーします。

次の演習

この章では、開発ライフサイクルを通じて変更を管理する上で、ClearCase LT と UCM がどのように役立つのかを学習しました。次の章では、ClassicsCD.com の拡張依頼のために Arrange Shipment ユースケースを使用する作業を行います。

この章では、Rational RequisitePro と Rational Rose を使用して、実装する拡張用のユース ケースを作成します。

対象読者

この章は、主にアナリストを対象としていますが、ほかのすべてのチーム メンバーに関する内容も含まれています。

概要

この章では、RequisitePro と Rose を使用します。使用しているコンピュータにこれらのツールがインストールされているかどうかを確認するには、27 ページの「表 1 Rational Suite ツール」で記入したツール チャートを参照してください。

これらのツールがインストールされていない場合は、この章を読むことで多くのことを学習できますが、一部の演習は行うことができません。

インストールされている場合は、次の手順で RequisitePro を起動します (この章の後半では Rose も起動します)。

[スタート] ボタンをクリックし、[プログラム] をポイントします。次に、[Rational Software] をポイントし、[Rational RequisitePro] をクリックします。RequisitePro が起動し、[Open Project] ウィンドウが表示されます。[Let's Go RequisitePro] ウィンドウも表示された場合は、[Close] をクリックします。

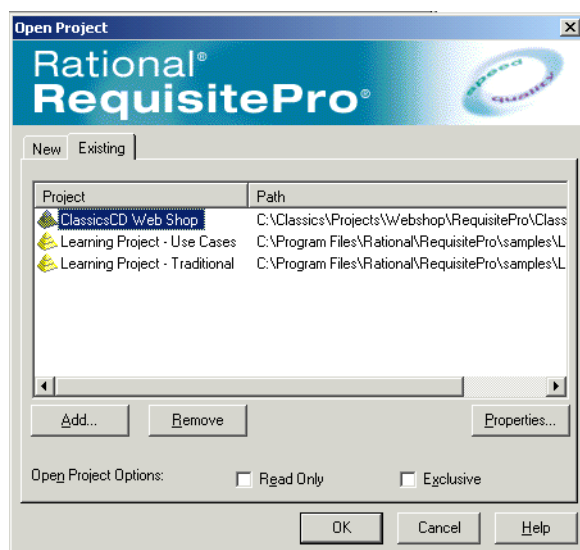
メモ: RequisitePro を初めて起動した場合は、Exploring Rational RequisitePro ページが表示されます。このページを読み、RequisitePro のユーザー インターフェイスについて学習します。このページを印刷してクイック リファレンスとして使用することもできます。読み終えたらウィンドウを閉じます。作業する RequisitePro プロジェクトを選択するウィンドウが表示されます。

演習 : ClassicsCD.com Webshop プロジェクトを開きます。

- 1 [Open Project] ウィンドウで、[Add] をクリックします。
- 2 [Add Project] ウィンドウで、C:\Classics\Projects\Webshop\RequisitePro に移動します。
- 3 ClassicsWebShop.rqs をクリックし、[開く] をクリックします。

[Existing] のプロジェクト リストに、ClassicCD Webshop に関連付けられた RequisitePro データベースが表示されます (図 14 を参照)。

図 14 RequisitePro を使用して ClassicsCD Webshop プロジェクトを開く



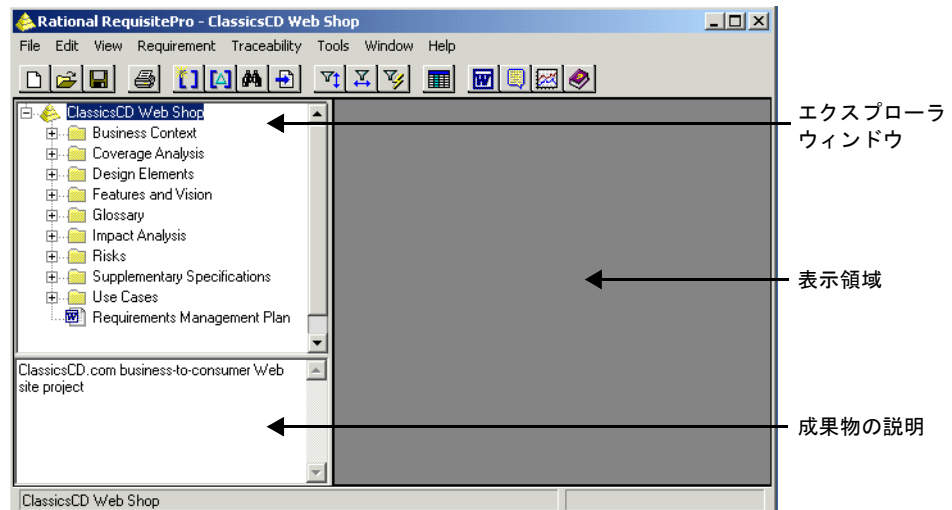
- 4 [OK] をクリックします。[Project Logon] ダイアログ ボックスが表示されます。
- 5 [User name] ボックスと [Password] ボックスの両方に「pat」と入力し、[OK] をクリックします。

メモ: このチュートリアルでは、Rational へログオンするためのユーザー名とパスワードに pat を使用します。

プロジェクトが開き、左側のペイン (エクスプローラ ウィンドウ) にある階層ツリー ブラウザにプロジェクトの成果物が表示されます。エクスプローラ ウィンドウで選択した成果物の内容が、エクスプローラ ウィンドウの下部に表示されます。右側のペインには、選択した成果物のビューが表示されます (図 15 を参照)。

選択した要素を見やすいようにウィンドウを調整できます。

図 15 RequisitePro の操作



要求管理が必要な理由

プロジェクト成功の定義の 1 つとして、完成したプロジェクトがその要求を満たしていることが挙げられます。要求とは、システムを満たす必要のある条件または機能のことです。分かりやすく言うと、要求とはシステムが実現しなければならない機能や動作を表したものです。

要求の決定方法

アナリストの最初の仕事は、利害関係者、つまりそのプロジェクトに関心を持つすべての人々のニーズを集めることから始まります。これらのニーズを特定するには、ユーザーとその他の利害関係者から聞き取り調査を行って拡張依頼を分析し、プロジェクトチームのメンバーと作業を進めます。このような作業を経て、どのニーズをプロジェクト要求にするかを決定します。

要求管理

要求管理とは、次の目的を達成するための体系的なアプローチです。

- 要求の決定、ドキュメント化、構成、追跡。
- システムの要求に関する顧客とプロジェクト チームの間における合意の確立と管理。

要求は、プロジェクトを通じて常に変化するため、管理が容易ではありません。たとえば、不可欠な機能についてのユーザーの意見は変わることがありますし、最初からユーザーの要求が明確であるとは限りません。競合他社が同種のソフトウェアの新しいバージョンをリリースする可能性もあります。この場合は、途中でプロジェクト計画を変更してこれに対応しなければなりません。また、法律の変更がソフトウェアに影響を及ぼすこともあります。変更管理を行わないと、予定外の機能変化によってプロジェクトの進行が遅れ、プロジェクトが複雑化してしまう恐れがあります。

RequisitePro の使用

RequisitePro を使用すると、要求の設定と管理が容易になります。RequisitePro は、Microsoft Word と統合されており、Word のテンプレートがパッケージ化されているので、簡単に使用することができます。RequisitePro は、チームのあらゆるメンバーが使用できるように設計されています。

- アナリストは、要求をドキュメント化して管理することができます。
- 開発者は、要求を使用してアーキテクチャを設計し、より詳細な仕様を決定することができます。
- テスト担当者は、要求を使用してテストを設計し、テスト範囲を確認することができます。
- プロジェクト リーダーとマネージャーは、RequisitePro を使用して、使用可能なリソース（時間、予算、人員など）に基づきプロジェクト作業を計画することができます。

ユース ケースの使用

「第 2 章 このチュートリアルについて」では、フィードバックに応答する形で入力された拡張依頼がありました。チーム メンバーの 1 人が、この拡張依頼に対応するユース ケースでの作業を開始したものとします。

ユース ケースを使用する理由

ユース ケースは、チームの全員が理解可能な共通言語でシステムの動作を記述します。ユース ケースを使用する作業は、Rational Unified Process (RUP) における重要な統一メカニズムです。

ユース ケースは、プロジェクトに関わるすべてのメンバーにとって重要なものです。

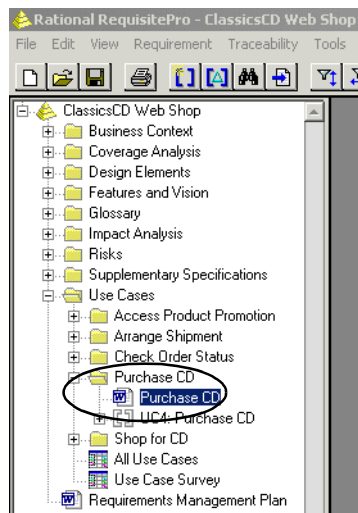
- アナリストは、ユース ケースを使用して、システムのとるべき動作を表し、予定された変更を利害関係者と共に確認します。
- 開発者と設計者は、人間の話す普通の言語とグラフィックによるユース ケースを使用して、作業を開始できます。この場合は、まずアーキテクチャの仕様を作成し、次にクラスを作成します。
- テスト担当者は、ユース ケースに基づいてテストを設計します。

- システムのテスト担当者は、設計フェーズのできるだけ早い段階でユース ケースを使用して、システム動作を検証します。
- プロジェクト リーダーとマネージャーは、ユース ケースを使用して、要求の結果がシステムに対する顧客の考えに一致しているかどうかを正式に確認します。

演習：ユース ケース ドキュメントを開きます。

- 1 エクスプローラ ウィンドウで、[ClassicsCD Web Shop]、[Use Cases]、[Purchase CD]、[Purchase CD] の順にツリーを展開し、このエントリをダブルクリックします (図 16 を参照)。

図 16 RequisitePro でユース ケース ドキュメントを開く



RequisitePro で Word ドキュメントが開き、Purchase CD ユース ケースが表示されます。このドキュメントは、RUP のテンプレートを基に作成されています。

第 3 章で、このテンプレートのアウトラインへのリンクが設定されています。RUP では、このアウトラインは [Analyst] ツリー セットにあり、[Artifacts]、[Requirements Artifact Set]、[Use-Case Model]、[Use Case]、[Use Case Specification (Informal)] の順に展開します。

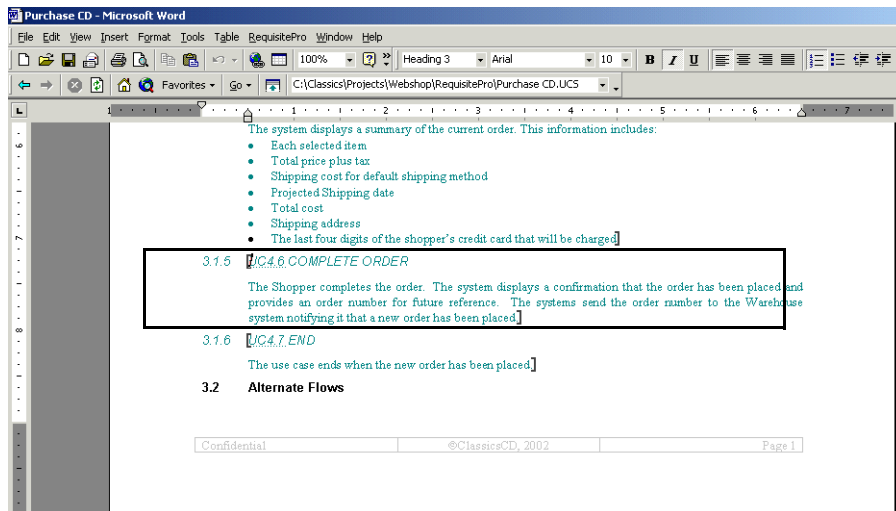
- 2 Table of Contents で [Section 3, Flow of Events] に進みます。[Section 3.1.5 Complete Order] を見つけます。対応するページ番号をクリックします。

メモ: コンピュータにインストールされている Microsoft Word のバージョンによっては、ページへのハイパーリンクを有効にするのに [CTRL] を押すことが必要な場合があります。

対応するユース ケース仕様 UC4.6 Complete Order に自動的にジャンプします (図 17 を参照)。

青緑色の斜体の大文字で書かれたテキストが、ユース ケース要求を示します。これらのユース ケース要求 (プレフィックス UC で識別) は、システムの動作を示す高レベルの要求です。

図 17 RequisitePro を使用したユース ケース ドキュメントの操作



- 3 Brief Description と Flow of Events の内容を確認します。購入者が注文した時点では、出荷予定日の表示が元のユース ケースに含まれていない点に注意してください。この後の演習で、この機能を要求として扱います。

これは、要求に関する作業を開始する際の代表的な方法です。要求のドキュメント化には、使い慣れた Word を使用します。一方、プロジェクト要求の確認と詳しい検討には RequisitePro を使用します。また、RequisitePro を使用して、関連する要求を示します。さらに、RequisitePro はシステムへの変更が要求にどのような影響を与えるかと、要求への変更がシステムにどのような影響を与えるかを追跡します。

- 4 使用しているコンピュータに Rose がインストールされていない場合は、Word と RequisitePro のウィンドウを最小化します。Rose がインストールされている場合は、Word を終了します。変更を保存するかどうかを確認するメッセージが表示されたら、[いいえ] をクリックします。RequisitePro も終了します。プロジェクトを閉じるかどうかを確認するメッセージが表示されたら、[はい] をクリックします。

RequisitePro での要求の処理方法

RequisitePro はドキュメントとデータベースの両方を中心として管理を行います、次のような利点を有しています。

- ドキュメント機能によって、使い慣れた環境 (Word) で説明を作成したり作業内容をプロジェクトの利害関係者に通知したりすることができます。要求ドキュメントを開始するには、既存の Word ファイルを RequisitePro にインポートするか、RequisitePro の Word ドキュメントを使用します。
- データベース機能によって、要求の整理、作業の優先順位の決定、要求の変更の追跡、ほかの Rational ツールとの情報の共有を行うことができます。データベース機能を使用するには、RequisitePro ビューを使用します。

ユース ケース ドキュメントでは、ドキュメント内に要求テキスト (デフォルトでは二重下線が引かれた文) が含まれています。データベースには、要求の追跡時に使用する属性 (優先度や割り当て先など) と共に要求テキストも保存されています。この章の後半では、RequisitePro のデータベース機能を使用して作業を行います。

ユース ケースについての詳細

要求に関する作業の多くは、ユース ケースを作成することから開始します。ユース ケースを使用する場合は、Rose を使用してユース ケースをビジュアル モデルに組み込み、次に RequisitePro を使用して説明、属性、リンクを追加します。

RUP では、ユース ケースの記述方法を参照することができます。これには、アクティビティを開始するために必要な成果物と、アクティビティを実行した結果得られる成果物に関する詳細が含まれます。さらに、アクティビティ自体についての詳細も含まれており、ユース ケースを作成するためのテンプレートも用意されています。

このテンプレートは、ユース ケースを構成するためのガイドラインの役割を果たします。このテンプレートを基に、ユース ケース要求を定義することができます。ユース ケースを作成する際は、整合性と完全性を確保するために、このテンプレートを使用するか、所属するグループが作成したテンプレートを使用することをお勧めします。テンプレートを使用することによって、すべての利害関係者が重要なプロジェクト情報を参照し、理解することができます。

Rose を使用したユース ケース作業の継続

アナリストは、Rational Rose を使用して、システムの動作をユース ケース図として視覚化することができます。これらの図からシステムの全体像をとらえることができるので、複雑なプロジェクトの管理が容易になります。ユース ケース図には次の項目が示されます。

- システムの動作。ユース ケースはシステムが何を行うのかを記述します。
- システムの範囲。アクターによって、システムと連携して動作する外部エンティティを表します。
- ユース ケースとアクターの関係。

Rose を使用してユース ケース図を作成すると、システムのユース ケースを集中化した形で視覚的に表現することができます。これによって、プロジェクトの目標や期待される製品についてすべての利害関係者が共通の認識を持つことができます。

また、Rose を使用することは、開発ライフサイクルにおける変更の影響を継続的に伝達するための有効な方法でもあります。ユース ケース図は、簡単に理解できるソフトウェア設計用の業界標準言語である UML で記述されているので、すべてのチーム メンバーが容易に共有、変更することができます。たとえば、アナリストは Rose を使用して高レベルなシステムを記述するユース ケース図を作成することができます。この章の後半では、アーキテクトが Rose を使用することによってどのようにこの作業を継続し、より詳細なシステム設計を行うのかを示します。結果として、システム図、アーキテクチャ、データは、Rational Rose という 1 つのツールと UML という 1 つの言語によって管理されることになります。

ユース ケース図を使用した作業

ここでは、機能拡張を実装する第 1 段階として、引き続き ClassicCD.com の Purchase CD ユース ケースに関する作業を行います。

演習: Rose を起動します。

- 1 [スタート] ボタンをクリックし、[プログラム] をポイントします。次に、[Rational Software] をポイントし、[Rational Rose] をクリックします。

Rose が起動し、[Create New Model] ダイアログ ボックスが表示されます。[Don't show this dialog in the future] チェック ボックスがオフになっていることを確認してください。このチュートリアルで今後 Rose モデルを開くのが簡単になります。

- 2 [Cancel] をクリックして、[Create New Model] ダイアログ ボックスを閉じます。

- 3 Rational Rose を使用して、ワークスペース間でモデルを移動したりコピーしたりすることができます。このためには、次のように Rose を設定する必要があります。
 - a Rose のメニュー バーで [File] メニューの [Edit Path Map] をクリックします。
[Virtual Path Map] ウィンドウが表示されます。
 - b [Virtual Symbol to Actual Path Mapping] ボックスで、\$CURDIR を探します。ボックスにこのシンボルがある場合は、[Virtual Path Map] ウィンドウを閉じ、手順 4 に進みます。
 - c [Symbol] ボックスに「CURDIR」と入力します。
 - d [Actual Path] ボックスに「&」と入力します。
 - e [Add] をクリックします。
 - f [Close] をクリックします。
- 4 Rose のメニュー バーで [File] メニューの [Open] をクリックし、C:\¥Classics¥Projects¥Webshop¥Rose に移動します。
- 5 ClassicsCD_WinDNA.mdl をクリックし、[開く] をクリックします。
- 6 サブユニットをロードするかどうかを確認するメッセージが表示されます。[はい] をクリックします。

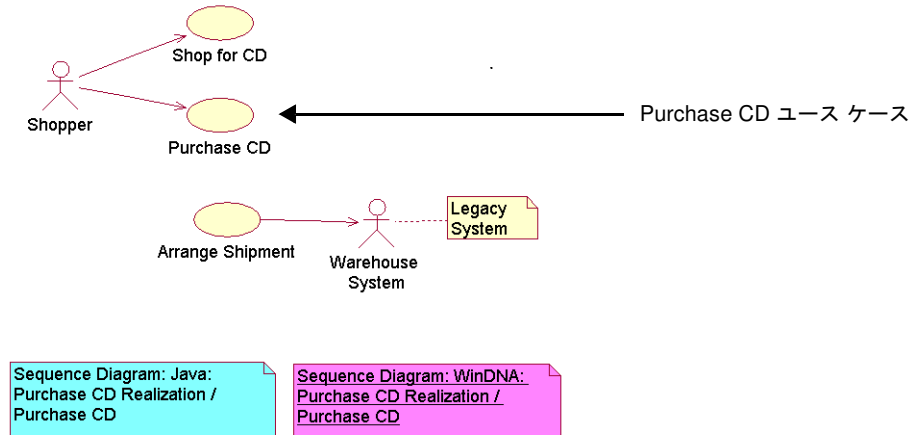
左上のペインに階層ツリー ブラウザ (Rose ブラウザ) が表示されます。右側のペイン (図ウィンドウ) にはアーキテクチャの論理ビューが表示され、システム内の最上位パッケージが互いにどのように関わっているのかを見ることができます。必要に応じて図ウィンドウを最大化します。

演習: ユース ケース図を表示します。

- 1 Rose ブラウザで、[Use Case View]、[System Models]、[Main] の順にツリーを展開し、[Main] をダブルクリックしてユース ケース図を表示します (図 18 を参照)。必要に応じて Rose 内で図ウィンドウを最大化します。

図 18 Rose を使用した ClassicsCD.com ユース ケース図の表示

ClassicsCD.com



Rose モデルと RequisitePro プロジェクトの関連付け

この章の前半では、RequisitePro で ClassicsCD.com Webshop プロジェクトを開きました。ここでは、ClassicsCD.com の Rose モデルと RequisitePro のプロジェクトとの関連付けを表示する方法を学習します。また、要求を作成して、ユース ケースを要求にリンクさせます。

演習：Rose モデルと RequisitePro プロジェクトの関連付けを表示します。

- 1 ユース ケース図上で、Purchase CD ユース ケース オブジェクトを右クリックします。ショートカット メニューの [View RequisitePro Association] をクリックします。

ログイン ボックスが表示されたら、[User name] ボックスと [Password] ボックスの両方に「pat」と入力し、[OK] をクリックします。

[View RequisitePro Association] ダイアログ ボックスが開き、この章の前半で示した Purchase CD ユース ケース ドキュメントと Purchase CD 要求に関する情報が表示されます。

- 2 ユース ケースの RequisitePro の関連付けを確認し、[OK] をクリックします。

Purchase CD ユース ケースは、テキストとビジュアル モデルの両方で表現されます。Rose と RequisitePro が統合されているので、ユース ケース自体は単一の要素として扱われます。

新しい要求の作成

Purchase CD は確立されたユース ケースですが、顧客からの注文に対してその出荷予定を表示するという拡張依頼に対応する要求を確認する必要があります。

演習：要求を作成します。

- 1 使用しているコンピュータに Rose がインストールされていない場合は、Word のウィンドウを最大化し、手順 2 に進みます。Rose がインストールされている場合は、Rose モデルに対する作業を続けます。Purchase CD ユース ケースを右クリックし、ショートカットメニューの [Use-Case Document] をポイントして、[Open] をクリックします。

RequisitePro の Word ドキュメントが開き、ユース ケースを操作することができます。

- 2 ドキュメントをスクロールし、ユース ケース仕様 UC4.6 Complete Order を表示します。2 番目の文で、文末のピリオドの前に、「as well as an expected ship date for each CD」という文を挿入します。

これにより、2 番目の文は次のようになります。「The shopper completes the order. The system displays a confirmation that the order has been placed and provides an order number for future reference as well as an expected ship date for each CD.」

- 3 Word のメニュー バーで [RequisitePro] メニューの [Document] をポイントし、[Save] をクリックします。

UC4.6 の変更に関する記述の入力を求めるメッセージが表示されます。「Modified to correspond with the enhancement request to display an estimated ship date for customer orders.」と入力し、[OK] をクリックします。

ドキュメントが保存され、要求が更新されます。

演習：要求のプロパティについて学習します。

- 1 Word のメニュー バーで [RequisitePro] メニューの [Requirement] をポイントし、[Properties] をクリックします。[Requirement Properties] ダイアログ ボックスが表示されます。
- 2 [General] タブで、この要求に関する情報を確認します。
- 3 [Attribute] タブをクリックします。要求の属性を確認して、優先度やステータスなど、Complete Order 要求の内容を詳しく調べます。
- 4 [Hierarchy] タブをクリックします。このタブの内容から、チュートリアルでの拡張依頼を反映するように変更を加えた Complete Order 要求が、Purchase CD ユース ケースの一部であることが分かります。
- 5 要求のプロパティを確認したら、[OK] をクリックします。

データベース内の要求の確認

ここでは、RequisitePro データベースを使用して、作業対象となっている拡張に関係するほかの要求を参照します。データベース上で作業をする場合は、特定の形式でデータをフィルタするビューを使用します。RequisitePro の動作に影響はありません。

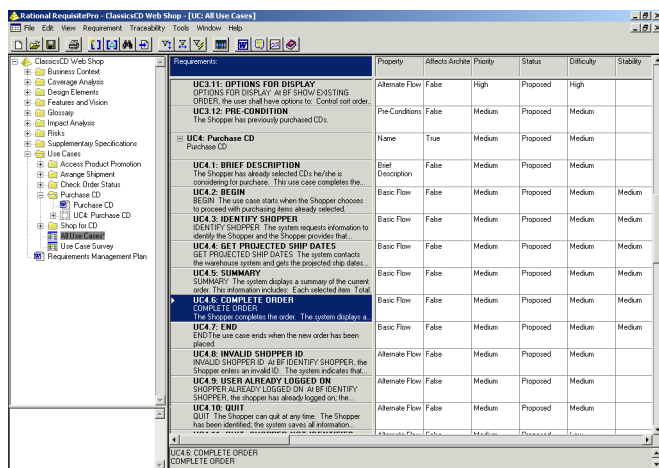
演習：RequisitePro データベースを使用して要求を表示します。

- 1 Word ドキュメントで、[RequisitePro] メニューの [Show RequisitePro Explorer] をクリックします。RequisitePro が起動します。
- 2 エクスプローラ ウィンドウで、[ClassicsCD Web Shop]、[Use Cases] の順にツリーを展開し、[All Use Cases] をダブルクリックします。

このユース ケースは階層構造になっており、それぞれの階層が ClassicsCD.com の機能領域を表しています。子要求のユース ケースは、それぞれの親の下に一覧表示されます。親要求は全般的のものであるのに対し、子要求は特定の領域に関する内容になっています。ユース ケースの子を参照するには、ユース ケース名の横にある [+] をクリックして、親要求を展開します。

- 3 UC4.6: Complete Order まで画面をスクロールします。このビューの要求のプロパティを確認します (図 19 を参照)。

図 19 RequisitePro を使用した要求の表示



Requirements	Property	Affects Archite	Priority	Status	Difficulty	Stability
UC4.11: OPTIONS FOR DISPLAY OPTIONS FOR DISPLAY. A/B IF SHOW EXISTING ORDER, the user shall have options to: Control sort order.	Alternate Flow	False	High	Proposed	High	
UC4.12: PRE-CONDITION The Shopper has previously purchased CDs.	Pre-Conditions	False	Medium	Proposed	Medium	
UC4: Purchase CD Purchase CD	Name	True	Medium	Proposed	Medium	
UC4.1: BRIEF DESCRIPTION The Shopper has already selected CDs he/she is considering for purchase. This use case completes the...	Brief Description	False	Medium	Proposed	Medium	
UC4.2: BEGIN BEGIN. The use case starts when the Shopper chooses to proceed with purchasing items already selected.	Basic Flow	False	Medium	Proposed	Medium	Medium
UC4.3: IDENTIFY SHOPPER IDENTIFY SHOPPER. The system requests information to identify the Shopper, and the Shopper provides that.	Basic Flow	False	Medium	Proposed	Medium	Medium
UC4.4: GET PROJECTED SHIP DATES GET PROJECTED SHIP DATES. The system contacts the warehouse system and gets the projected ship dates.	Basic Flow	False	Medium	Proposed	Medium	Medium
UC4.5: SUMMARY SUMMARY. The system displays a summary of the current order. The information includes: Each selected item, Total	Basic Flow	False	Medium	Proposed	Medium	Medium
UC4.6: COMPLETE ORDER COMPLETE ORDER. The Shopper completes the order. The system displays a...	Basic Flow	False	Medium	Proposed	Medium	Medium
UC4.7: END END. The use case ends when the new order has been placed.	Basic Flow	False	Medium	Proposed	Medium	Medium
UC4.8: INVALID SHOPPER ID INVALID SHOPPER ID. A/B IF IDENTIFY SHOPPER, the Shopper enters an invalid ID. The system indicates that...	Alternate Flow	False	Medium	Proposed	Medium	
UC4.9: USER ALREADY LOGGED ON USER ALREADY LOGGED ON. A/B IF IDENTIFY SHOPPER, the Shopper has already logged on, the...	Alternate Flow	False	Medium	Proposed	Medium	
UC4.10: QUIT QUIT. The Shopper can quit at any time. The Shopper has been identified, the system saves all information...	Alternate Flow	False	Medium	Proposed	Medium	

- 4 要求の属性とプロパティをさらに表示するには、画面を右にスクロールします。

別の要求へのリンク

ここまでは次の作業を行ってきました。

- 新しい拡張の実装の決定
- Rose でのユース ケース図の確認
- モデルと RequisitePro プロジェクトのリンク
- 新しいユース ケース要求の作成
- 要求の属性への値の追加

ここではさらに、ユース ケース要求が、別のタイプの要求である機能要求にリンクされていることを確認します。システムの機能要求は非常に高いレベルで作成されるもので、システム全体の基礎となります。

演習: ユース ケース要求と機能要求の関係を表示します。

- 1 RequisitePro のエクスプローラ ウィンドウで、[ClassicsCD Web Shop]、[Coverage Analysis] の順にツリーを展開し、[Use Cases to Features relationships] をダブルクリックします。

このビューが開き、要求の階層全体が表示されます。詳細を表示するには、マトリックスの縦横のデバイダを移動します。


- 2 UC4 まで画面をスクロールし、必要に応じて展開して子要求を表示します。

セル内に矢印のシンボルが表示され、Purchase CD ユース ケース要求 (そのすべての子要求を含む。子要求は変更を加えたものも含む) と機能要求 Feature1: Secure Payment Method の関係を示します。追跡可能性については、次の項で詳しく説明します。

- 3 スクロール バーを使用してこの要求のビューを参照し、ユース ケースと機能の関係を詳細に確認します。

追跡可能性リンクとサスペクト リンク

RequisitePro ビューのマトリックスには、要求間のリンクの一部が表示されます。これらのリンクは、要求同士の依存関係を表します。

斜線が引かれた矢印のシンボル  が表示されている場合は、そのリンクがサスペクトリンクであることを示しています。リンク関係にある要求が変更されると、そのリンクはサスペクトリンクになります。サスペクトリンクになると、要求が変更されたことがプロジェクトチームに警告されます。アナリストは、この情報を使用して次のことを行います。

- 変更によるプロジェクトへの影響を評価する。
- 要求をより効率的に管理する。
- マネージャー、プロジェクト リーダーと共同して、より効率的にプロジェクトの範囲を管理する。

その他の要求タイプ

ここまでは、高レベルの機能要求と詳細なユース ケース要求について説明しました。要求によってはユース ケースを使用できない場合もあるため、**RequisitePro** では別のタイプの要求もサポートしています。たとえば、パフォーマンス目標やプラットフォーム サポートに関して補足的な要求を定義することができます。その他の要求タイプには、設計要求、ビジネス ニーズ、用語集に関する要求などがあります。

また、新しい要求タイプを定義することもできます。**RequisitePro** では、プロジェクトに必要なあらゆるタイプの要求を管理できます。

要求収集の終了時期

要求は、アナリストとプロジェクトの利害関係者 (アプリケーションのユーザーやマーケティング チームのメンバーなど) との一連のやり取りの中から発生します。要求を収集したら、適切な利害関係者と作業の確認を行います。利害関係者と作業チームの合意が得られれば、要求収集における最初の仕事は完了です。

当然、プロジェクトの進行に合わせて要求管理を継続し、何かを追加して別のものを削除したり、変更に対応したりといった作業が必要になります。

拡張ヘルプ

拡張ヘルプは、より高いレベルのタスクに関する情報を提供する **Rational Suite** の強力な機能です。拡張ヘルプでは、**Rational Suite** ツールから RUP に直接アクセスすることができます。さらに、拡張ヘルプには組織のガイドラインや標準を追加することができます。

演習: 拡張ヘルプを表示します。

- 1 **RequisitePro** のメニュー バーで **[Help]** メニューの **[Extended Help]** をクリックします。

[Extended Help: Overview] ウィンドウが RUP 内に表示されます。また、このツール用の拡張ヘルプ ツリー セットも表示されます。

- 2 **[Extended Help - ReqPro]** ツリー セットの左側ペインのリスト下部で、**[More Content]**、**[Tool Mentor: Managing Use Cases Using Rational Rose and Rational RequisitePro]** の順にクリックします。

このツール メンターを読み、この章で行った作業を確認します。

拡張ヘルプの詳細は、どの **Rational** ツールからでも参照できます。使用しているツールのメニュー バーで **[Help]** メニューの **[Extended Help]** をクリックします。使用している **Suite** ツールに関連する拡張ヘルプ トピックが表示されます。

まとめ

詳細

RequisitePro の詳細については、RequisitePro のチュートリアルを参照してください。
RequisitePro のメニュー バーで、[Help] メニューの [Tutorial] をクリックします。

Rose の詳細については、85 ページの「機能拡張のモデリング」を参照してください。

クリーンアップ

拡張ヘルプを終了します。

RequisitePro のメニュー バーで [File] メニューの [Exit] をクリックして、RequisitePro を終了します。プロジェクトの終了を確認するメッセージが表示されたら、[はい] をクリックします。ビューへの変更を保存するかどうかを確認するメッセージが表示されたら、[いいえ] をクリックします。

Rose を使用している場合は、Rose のメニュー バーで [File] メニューの [Exit] をクリックして、Rose を終了します。変更を保存するかどうかを確認するメッセージが表示されたら、[いいえ] をクリックします。

この章の学習内容

この章では、次のことを学習しました。

- 要求とは、プロジェクトが準拠すべき条件または機能です。
- 要求管理とは、システム機能と属性の決定、ドキュメント化、体系化、追跡を行うための体系的なアプローチです。
- RequisitePro によって、要求と複数の要求タイプを管理します。
- RequisitePro はドキュメントとデータベースの両方を中心として管理を行うため、開発チームはその両方の利点を作業に活かすことができます。
- ユース ケースを使用する場合は、Rose を使用してユース ケースをビジュアル モデルに組み込み、次に RequisitePro を使用して説明、属性、リンクを追加します。
- 拡張ヘルプを使用すると、RUP と Rational Suite ツールの使用法に関する情報にすぐにアクセスすることができます。拡張ヘルプにはユーザー独自の情報を追加できます。

次の演習

次の章では、この章で確認した要求を使用してテスト計画を作成します。

これまでの章では、ClassicCD.com の機能拡張のための要求を定義しました。モデル作成やコード実装はまだ行っていませんが、Rational TestManager によるテスト計画の作成は開始することができます。

対象読者

この章は、テスト担当者、品質保証マネージャー、システム テストに関係するその他のチームメンバーを対象としています。

概要

この章では、Rational TestManager を使用します。使用しているコンピュータに TestManager がインストールされているかどうかを確認するには、27 ページの「表 1 Rational Suite ツール」で記入したツール チャートを参照してください。

TestManger がインストールされていない場合は、この章を読むことで多くのことを学習できますが、演習を行うことはできません。

TestManager がインストールされている場合は、[スタート] ボタンをクリックし、[プログラム] をポイントします。次に、[Rational Software] をポイントし、[Rational TestManager] をクリックします。[Rational Test Login] ダイアログ ボックスが表示されます。

演習: ClassicsCD.com Webshop プロジェクトを開きます。

[Rational Test Login] ダイアログ ボックスで、次の操作を行います。

- 1 [User Name] ボックスと [Password] ボックスに「pat」と入力します。
- 2 [Project] ボックスに Webshop が表示されていることを確認します。
- 3 [Location] ボックスに C:\¥Classics¥Projects¥Webshop¥Webshop.rsp が表示されていることを確認します。
- 4 [OK] をクリックします。

Webshop プロジェクトが開きます。左側のペイン (テスト アセット ワークスペース) の下部にある [Planning] タブが選択されていることを確認します。これで、プロジェクトの作業を行うことができます。

テスト計画とは

テスト計画を作成することにより、開発チームはそのプロジェクトにおけるテスト作業に対し、効果的な評価と管理を行うことができます。テスト計画の作成過程においては、実行するテストのタイプ、テストの実装と実行の方針、テストに必要なリソースを決定する必要があります。

テスト計画の作成は、開発サイクルの初期の段階において最初の要求セットを確認した時点から開始します。ユース ケース要求、プロジェクト スケジュール、ビジュアル モデルなどの成果物は、テスト入力として使用し、テスト対象を決定するのに役立てることができます。テスト ケースは、これらの入力に含まれる情報を使用して作成します。これらのテスト ケースはテスト計画作成プロセス全体を通じチェックリストとして使用し、これによってテストの受け入れ基準を決定します。テスト入力は、テスト構成、つまりコンピュータのハードウェアとソフトウェアに関する属性を決定するために使用することもできます。これらの属性は、作成したテスト ケースに適用できます。

開発と同様に、テスト計画の作成も反復を伴うプロセスです。テスト計画の作成は、アナリストが要求の変更や詳細決定を行った場合や、開発者がコードを設計して実装した場合など、プロジェクトのライフサイクル全体を通じて続けられます。

リスクの管理

テスト計画において推奨される方針は、まずそのプロジェクトにおいて最もリスクの高い部分に重点を置くことです。たとえば、次のような事項を検討することによってリスクを特定できます。

- アプリケーションの特定部分をテストしなかった場合に生じる結果。
- アプリケーションの特定部分が正常に機能しなかった場合に生じる結果。
- 製品出荷後にエラーが発見される可能性。
- プロジェクト メンバーではなくユーザーがアプリケーションのエラーを発見した場合に発生が予想される問題。

計画の作成と進行の測定

Rational TestManager を使用してテストを計画、設計、実行する際は、あらゆるタイプのプロジェクト成果物を使用することができます。テスト計画の作成には、要求、ビジュアル モデル、ソース コードを使用できるので、製品の機能、システム アーキテクチャ、コードを含め、システムのすべての側面をテストすることが可能です。

各種 Rational ツールとの統合により、プロジェクト アセットを共有することができるため、開発ライフサイクルの初期に品質確認のためのテストを開始することができます。

Rational TestManager では、次のようなことも可能です。

- テストに関連するすべての情報と成果物にアクセスできるので、開発チームはプロジェクトのステータスを簡単に知ることができます。
- チーム メンバーがテストの進行に関する情報を共有することができます。
- 計画、スクリプト作成、実行されたテストの数を追跡することができます。
- テストの対象となった要求を示したり、テストの可否の数を表示したりすることができます。

TestManager は Suite の各エディションに付属しており、チーム メンバーはこれを使用して、自分たちがプロジェクトの要求をどの程度満たしているのかの評価、プロジェクト全体のステータスの監視、ほかのプロジェクト利害関係者とのテスト作業に関する情報のより効果的な共有と調整などを実現することができます。

テスト計画の作成

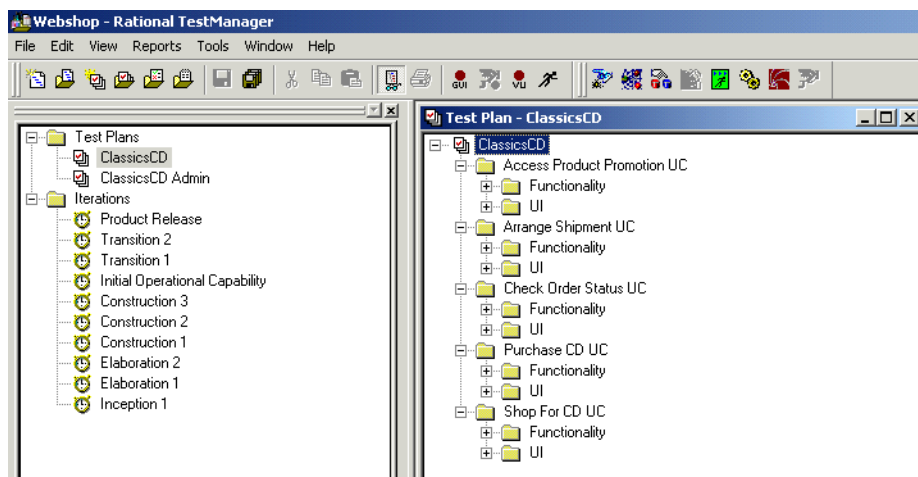
TestManager を使用したテスト計画には、Rational プロジェクトでのテストの目的と目標に関する情報と、テストの実装と実行の方針に関する情報を含めることができます。テスト計画は、1 つのプロジェクトに対して複数作成できます。テスト計画には、テスト計画名、テスト計画に関連付けられた構成、テスト計画を終了すべき時間枠など、複数のプロパティを含めることができます。

また、テスト計画のプロパティに基づいてレポートを作成することもできます。たとえば、テスト計画の一部であるテスト ケースを特定するレポートを作成できます。このようなレポートには、プロジェクトのテストの状態に関する重要な情報が示されます。

テスト計画の構成

テスト計画にはテスト ケース フォルダが含まれ、テスト ケース フォルダにはテスト ケースが含まれます。テスト ケースは、テストと検証が可能な、システム内の動作です。テスト計画とテスト ケース フォルダは、その開発組織に適した方法で構成することができます。たとえば、テスト タイプごと、ユース ケースごと、テストの反復またはフェーズごとにテスト計画を作成できます。また、このチュートリアルに示すように、ユース ケースを表すテスト ケース フォルダでマスター テスト計画を作成することもできます (図 20 を参照)。

図 20 TestManager におけるテスト計画構成の例



テスト ケース フォルダには、テスト ケース フォルダ名などのプロパティがあり、構成と反復が関連付けられています。

演習 : ClassicsCD.com テスト計画の構造を理解します。

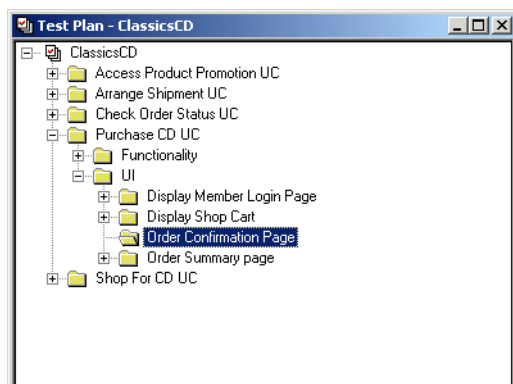
- 1 テスト アセット ワークスペースの [Planning] タブで、[Test Plans] を展開し、[ClassicsCD] をダブルクリックします。

右側のペインに ClassicsCD のテスト ケース フォルダが表示されます。ClassicsCD.com Webshop プロジェクトのテスト ケース フォルダは、前に示したようにユース ケースで構成されています (図 20 を参照)。

- 2 Purchase CD ユース ケースのテスト ケース フォルダを展開します。[Functionality] と [UI] の 2 つのフォルダが表示されます。このチュートリアルでは、システム動作を表すテスト スクリプトが [Functionality] テスト ケース フォルダに置かれ、この機能を Web サイト上で表現するユーザー インターフェイスが [UI] テスト ケース フォルダに置かれています。
- 3 [UI] テスト ケース フォルダを展開し、[Order Confirmation Page] フォルダを開きます。

このフォルダには、テスト ケースは関連付けられていません。また、前に行った Complete Order ユース ケース要求への変更に対するテスト スクリプトもまだ存在していません (図 21 を参照)。注文を受けた CD の出荷日を表示するという機能拡張に対するテスト スクリプトを後で作成できるようにするには、テスト ケースを作成する必要があります。

図 21 TestManager のテスト ケース フォルダ



テスト内容の決定

開発しているアプリケーション用のテスト ケースを確認することにより、テスト計画の作成を続けます。各テスト ケースは、テスト対象になっているアプリケーションの特定の領域を表します。各領域には、テストが必要な幅広いクラスの状況を含めることができます。たとえば、現金販売取引のテストにおいては、次の項目のテストを行うことになります。

- 有効な入力 (顧客が価格どおりの金額を支払う。顧客が価格よりも多く支払って釣銭が必要になる。)
- 無効な入力 (顧客が価格よりも少なく支払う。販売員が無効な製品番号を入力する。)

では、テストする項目はどのように決めればよいでしょうか。テスト計画作成のこの段階 (テストの分析と設計) においては、多くの場合、既存のプロジェクト アセットを参考にしながら、自分自身の知識と経験に頼って作業を進めることになります。

テストを設計する場合、その最初のステップはシステムの予想される動作を理解することです。この分析を通じて、次のことを確認するためにどのような状況でテストを行わなければならないかを把握します。

- そのアプリケーションは意図した動作をする。
- そのアプリケーションは意図しない動作をしない。

テスト ケースを使用した作業

テスト ケースは、テストと検証が可能な、システム内の動作を記述します。テスト ケースでは、アプリケーション内のテスト範囲を記述することもできます。要求などの既存のプロジェクト成果物はアプリケーションに関する情報を提供するものであり、テスト ケースに対するテスト入力として使用できます。**TestManager** には組み込みのテスト入力タイプが用意されていますが、ほとんどすべての成果物をテスト入力として使用することができます。

例として、どの成果物をどのようなテスト入力として使用できるかを次に示します。

- 要求は、システムが満たすべき条件または機能を記述します。
- ビジュアル モデルは、システムの構造と相互関係を視覚的に表現します。

また、ソース コード、ソフトウェア ビルド、機能仕様などのカスタム テスト入力タイプを定義することもできます。

テスト入力を決定したら、テスト ケースを作成してテスト入力に関連付けることができます。関連付けると、テスト入力に変更があった場合、それに対処することができます。このような変更があった場合は、テスト ケースやその実装の変更が必要になることがあります。

演習 : Complete Order 要求の機能拡張に対するテスト ケースを作成します。

- 1 **TestManager** の右側ペインで、[Order Confirmation Page] フォルダを右クリックし、ショートカット メニューの [Insert Test Case] をクリックします。

[New Test Case] ダイアログ ボックスが表示されます。このボックスに入力する情報は、テスト ケースの定義に使用できます。定義したテスト ケースは、テスト ケース フォルダの下に挿入されます。

- 2 [General] タブで、次の操作を行います。

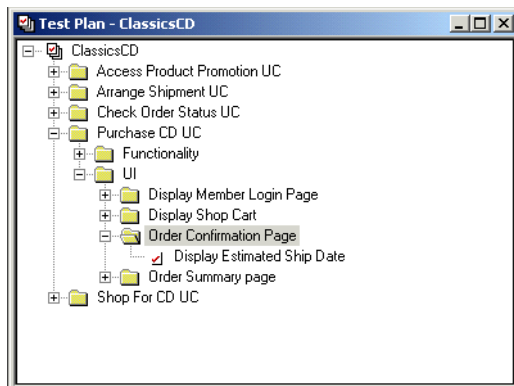
- a [Name] ボックスに「Display Estimated Ship Date」と入力します。
- b [Description] ボックスに、次のように入力します。
Warehouse system gets order and responds with estimated ship date.

- c [Owner] ボックスで pat が選択されていることを確認します。

- 3 [OK] をクリックします。

テスト計画の階層に [Display Estimated Ship Date] が表示されます (図 22 を参照)。

図 22 TestManager での ClassicsCD.com のテスト計画と新しいテスト ケース



Rational Rose からのテスト入力

コンピュータに Rational Rose がインストールされライセンスされている場合は、TestManager で Rose モデルを登録して、Rose モデル要素をテスト入力として使用できます。各モデル要素は [TestManager Test Input] ウィンドウに表示することができ、モデル要素とテスト ケースの間に関連付けを定義することができます。

Rational RequisitePro からのテスト入力

RequisitePro の要求をテスト入力として使用できます。チーム メンバーまたは管理者は、Rational Administrator を使用して、RequisitePro プロジェクトと Rational プロジェクトを関連付けることができます。関連付けてからそのプロジェクトにログオンすると、[TestManager Test Input] ウィンドウに要求が表示されます。その後、要求とテスト ケースを関連付けることができます。要求自体の作成と管理は RequisitePro で行いますが、要求のプロパティは TestManager でも変更できます。

演習：テスト入力とテスト ケースを関連付けます。

- 1 TestManager の右側ペインで、[Display Estimated Ship Date] テスト ケースを右クリックし、ショートカット メニューの [Associate Test Input] をクリックします。

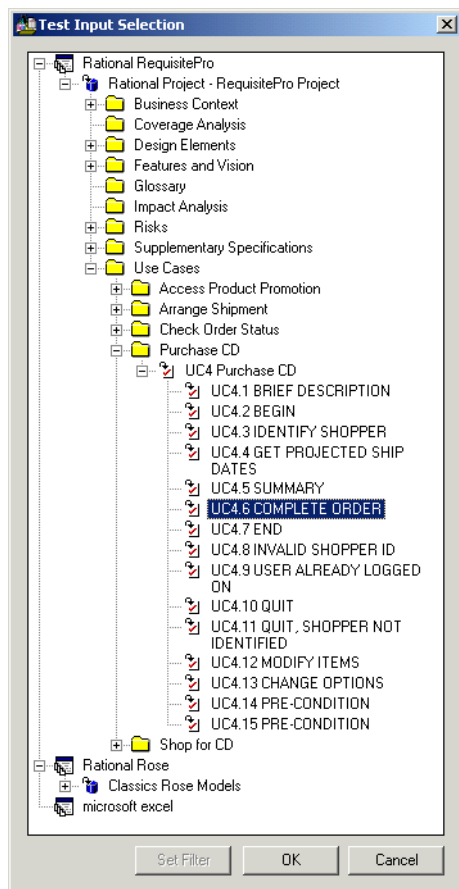
Webshop プロジェクトに関連付けられた成果物へのアクセスには時間がかかります。そのまましばらく待ってください。間もなく、[Test Input Selection] ダイアログ ボックスが表示されます。この機能を使用して、プロジェクト成果物をテスト ケースに関連付けます。

- 2 [Rational RequisitePro] の下の [Rational Project - RequisitePro Project] を展開します。

RequisitePro から情報を取得する間、プログレス インジケータが表示されます。少し間を置いて、関連付けられた RequisitePro アセットが [Test Input Selection] ウィンドウ内の階層に表示されます。

- 3 [Rational Project - RequisitePro Project]、[Use Cases]、[Purchase CD]、[UC4 Purchase CD]、[UC4.6 COMPLETE ORDER] の順にツリー ノードを展開します (図 23 を参照)。

図 23 TestManager での ClassicsCD.com のテスト計画の選択



- 4 [UC4.6 COMPLETE ORDER] を選択し、[OK] をクリックします。

これで、このチュートリアルで既に変更を加えたユース ケース要求がこのテスト ケースに関連付けられました。

- 5 TestManager のメニューバーで、[File] メニューの [Save All] をクリックします。

追跡可能性リンクとサスペクト リンク

テスト入力をテスト ケースに関連付けると、2 つの成果物間の依存関係が作成されます。この例の場合、要求 UC4.6 COMPLETE ORDER が変更されると、テスト ケース Display Estimated Ship Date がサスペクトになります。

「第 5 章 要求の作成」で学習した追跡可能性リンクとサスペクト リンクを思い出してください。**TestManager** と **RequisitePro** 間の統合はこれに基づいているため、要求の属性が変更されると、関連付けられているすべてのテスト ケースがサスペクトになります。**TestManager** は、要求が変更されたことと、テスト ケースが影響を受ける可能性があることをプロジェクト チームに警告します。テスト担当者は、この情報を使用して次の作業を行います。

- テスト ケースとテスト計画に対する変更の影響を評価する。
- テスト計画をより効率的に管理する。
- ほかのチーム メンバーと共同で、より効率的にプロジェクトのアクティビティと成果物を管理する。

テスト ケースの詳細決定

テスト計画を作成する際は、その一環としてテストを設計する必要があります。テストの設計では、テスト ケースを詳細に記述します。これにより、テスト ケースをどのように実装するのかを理解するために必要な詳細情報が提供されます。設計作業は、そのプロジェクトのニーズに応じて、テスト ケースの計画と同時に行うこともできますし、テスト ケースの計画終了後に行うこともできます。

テストの設計には、**Design Editor** を使用します。このステップの間に、テスト ケースにおける最も基本的かつ可能性の高いフローを把握し、検証基準または検証ポイントを追加します。

テスト計画による影響の理解

ここまでは、テスト計画の構造を決定してこれを構成する方法について学習しました。「第 7 章 機能拡張のモデリング」では、システム構造に **Checkout ユーザー インターフェイス用の Arrange Shipment** 拡張を追加します。実装が完了したら、テストを開始できます。「第 10 章 機能テスト」では、この章で作成したテスト ケースを使用して拡張に対する機能テストを行います。

テスト計画を作成することにより、早い段階でテストの方針を確認し、テスト作業の意図をすべての利害関係者に伝えることができます。**TestManager** でのテスト計画は、チームのすべてのメンバーが使用できるように設計されています。

- アナリストは、テスト計画を使用して、テスト入力の設定、プロジェクトの反復の定義、テスト カバレッジ レポートを実行できます。
- 開発者は、テスト計画を使用して、ユニットテストの実行と、テスト ケースが実装計画と開発計画に一致しているかどうかの確認ができます。
- テスト担当者は、テスト計画を使用して、テスト ケース (テスト入力から作成したもの) の構成、テストの作成と実行、テスト結果の分析ができます。
- プロジェクト リーダーとマネージャーは、テスト計画を使用して、プロジェクトの反復の定義、カスタム レポートの作成、テスト カバレッジ レポートを実行できます。

テスト計画作成の継続

テスト計画の作成は、プロジェクト初期に開始され、繰り返し行われます。テスト計画は、アナリストが要求の変更やユース ケースの詳細決定を行った場合、開発者がコードを設計して作成した場合、テスト担当者が要求やテスト ケースを再検討してテスト対象の部分や条件を新たに発見した場合など、そのたびごとに作成されます。テスト計画の作成は、テストを含め、ほかの開発作業と並行して行われます。

実際にテスト計画に関する作業を行う場合は、少なくとも以下に挙げる項目について考慮することをお勧めします。これらの項目については、以降の項で説明します。

- リスクとリソース
- 実行するテストのタイプ
- テストの段階
- スケジューリング

リスクとリソース

リスクの特定は、テスト計画作成の重要な部分を占めます。テストに使用できるリソースを確認したら、避けることのできないリソース上の制約と、プロジェクト、テスト上のリスクとのバランスを取る必要があります。これによって、テスト方針をより確実なものにすることができます。

テストには、次のような優先度を付けることをお勧めします。

- **実行が不可欠 (高):** 重大なリスクを回避したり、開発サイクルの早い段階でプロジェクトの弱点を確認したりするために、絶対に実行する必要があるテストです。このテストを完了するまでは、プロジェクトのテストを終了することはできません。
- **実行が望ましい (中):** スケジュールに入れるべきテストですが、リソースに問題がある場合は省略も考慮できます。
- **省略可能 (低):** 実行すると有益なテストですが、プロジェクトにとって不可欠ではありません。このテストは、より重要なほかのテストにおいて進展が得られない場合に実行します。
- **不要 (低):** このテスト計画に含めないテストです。この優先度に分類されるテストは、テスト計画の境界を定義するので、テスト内容を絞り込む助けになります。

実行するテストのタイプ

テスト計画を作成する際に考慮すべきテストには、次に示すものを始めとして多くのタイプがあります。

- **信頼性テスト**: アプリケーションがエラーなしで機能するかどうかをテストします。信頼性テストには、Purify、Quantify、PureCoverage を使用します。
- **機能テスト**: アプリケーションが機能要求を満たしているかどうかをテストします。機能テストには、Rational Robot を使用します。
- **パフォーマンス テスト**: 変化する作業量の下でのシステム パフォーマンスが許容できるものかどうかをテストします。パフォーマンス テストの記録には、Robot を使用します。異なる作業量の下でのこれらのテスト スクリプトの実行とその結果の分析には、TestManager を使用します。

テストの段階

テスト計画を作成するにあたっては、考慮すべきいくつかのテスト段階があります。これらの段階は小さなコンポーネントのテストからシステム全体のテストへと進むもので、通常はシステム開発サイクルのさまざまな段階に適用されます。

- **ユニット テスト**: 個々のコンポーネント、つまりテスト可能なソフトウェア最小要素の確認を行います。
- **統合テスト**: ユース ケースのテストを実行するために実装モデルのコンポーネントを組み合わせた場合に、正しく動作するかどうかを確認します。
- **システム テスト**: ソフトウェアが全体的に機能するかどうかを確認します。
- **受け入れテスト**: ソフトウェアが出荷可能で、要求を満たしていることを確認します。

ユニット テストは通常、ソフトウェア開発者によって実行されます。テスト担当者は、主に統合テスト、システム テスト、受け入れテストに重点を置きます。

プロジェクトのスケジューリング

テスト計画の作成には、スケジュールの設定も含まれます。テスト計画を作成するときは、プロジェクトのほかの部分を担当しているチーム リーダーたちと調整を行い、その担当部分がいつテスト可能になるかを把握する必要があります。把握したら、テストのスケジュールを最も効果的なものとするために、元のスケジュールと確認済みのリスクやリソースとのバランスを取る必要があります。テストの各反復では、1 つまたは複数のテスト ケースが検証されます。反復に基づいてテスト スケジュールを設定すると、テスト ケースをフィルタできるので、ソフトウェア開発の各段階におけるテストの設計、実装、実行をより効果的に行うことができます。

80 ページの「リスクとリソース」に従って優先度を付ける場合は、「不可欠」なテスト (優先度高) と「望ましい」テスト (優先度中) をスケジュールに含める必要があります。プロジェクトの進行に伴ってリソース上の制約が生じた場合は、「不可欠」なテストが求める絶対的な品質目標を保ち、優先度の低いテストを省略することができます。

TestManager には RUP で定義された反復が組み込まれていますが、ユーザー独自の反復を作成することもできます。反復とテスト ケースは関連付けることができ、反復に基づいてこれらのテスト ケースを実行することができます。

RequisitePro は Microsoft Project と統合されているので、要求とタスクをプロジェクト スケジュールにリンクさせることができます。詳細については、次の手順で RequisitePro を起動してください。

- 1 [スタート] ボタンをクリックし、[プログラム] をポイントします。次に、[Rational Software] をポイントし、[Rational RequisitePro] をクリックします。RequisitePro が起動し、[Open Project] ダイアログ ボックスが表示されます。

RequisitePro の起動後に [Let's Go RequisitePro] ウィンドウが表示された場合は、[Close] をクリックします。

- 2 [Open Project] ウィンドウで、[Cancel] をクリックします。
- 3 RequisitePro メニュー バーで、[Help] メニューの [Contents and Index] をクリックします。
- 4 RequisitePro のヘルプ ブラウザの [目次] タブで、[Wizards, Integrations and Components]、[RequisitePro Wizards]、[MS Project Integration Wizard] の順にダブルクリックします。

ヘルプ トピックが表示され、RequisitePro と Microsoft Project を使用した作業について参照することができます。

テスト成果物についての詳細

このチュートリアルではテスト計画のすべての側面について説明できるわけではありませんが、この項では関係する成果物について詳しく知りたい場合の指針を示します。

演習：テスト成果物の詳細について学習します。

- 1 Rational Unified Process (RUP) が起動している場合は、RUP のウィンドウをアクティブにします。RUP が起動していない場合は、[スタート] ボタンをクリックし、[プログラム] をポイントします。次に、[Rational Software] をポイントし、[Rational Unified Process] をクリックします。
- 2 Web ブラウザで [Tester] ツリー セットを選択し、[Test]、[Artifact Overview] の順に展開して、Test: Artifact Overview ページを表示します。

この概要ページで、テストに関係する役割と成果物を確認することができます。これらの要素の詳細については、その要素をクリックしてください。

- 3 RUPでの作業が終わったら、RUPを終了します。また、このチュートリアルをとおしてRUPを使用し、学習対象となるトピックの詳細を知りたい場合は、RUPを最小化しておけばRUPをいつでも使用することができます。

まとめ

詳細

テスト計画の詳細については、次のマニュアルを参照してください。

- テスト計画の詳細については、『Rational TestManager User's Guide』を参照してください。
- テスト計画のより詳しい取り扱いについては、『Testing Computer Software』、Cem Kaner 他共著、Vnr Computer Library、ISBN: 1850328471 (邦訳:『基本から学ぶソフトウェアテスト』、テスト技術者交流会訳、日経BP社、ISBN: 4822281132)を参照してください。

クリーンアップ

TestManagerのメニューバーで[File]メニューの[Exit]をクリックして、TestManagerを終了します。

必要に応じて、[Rational RequisitePro Online Help]ウィンドウを閉じ、RequisiteProを終了します。終了を確認するメッセージが表示されたら、[はい]をクリックします。

この章の学習内容

この章では、次のことを学習しました。

- テスト計画の作成は、プロジェクトの早い段階、初期の要求を確認した後で開始できます。
- テスト計画の作成は反復を伴うプロセスです。プロジェクトとテストに関するリスクを網羅し、製品に対する要求、使用可能なリソース、プロジェクトのスケジュールなどが関係します。
- テスト計画の作成には、テストケースの作成や、テストケースとテスト入力の関連付けも含まれます。
- 分析と設計は、効果的なテストを作成する上で重要な部分を占めます。
- テストの優先度を付けることは、テストすべきアプリケーションにおいて最もリスクが高く、かつ最も重要な部分にテスト作業の重点を置く助けとなります。

次の演習

次の章では、ClassicsCD.comのビジュアルモデルにArrange Shipment拡張依頼を組み込みます。

これまでの章では、ClassicCD.com の機能拡張のためのユース ケース要求を定義しました。テスト担当チームはテスト計画の作成を開始しています。この章では、Rational Rose を使用して、ユース ケース要求の設計を ClassicsCD.com ビジュアル モデルに組み込む作業を続けます。

対象読者

この章は、ソフトウェア設計者と開発者を対象としています。

概要

この章では、Rational Rose を使用します。使用しているコンピュータに Rose がインストールされているかどうかを確認するには、27 ページの「表 1 Rational Suite ツール」で記入したツールチャートを参照してください。

Rose がインストールされていない場合は、この章を読むことで多くのことを学習できますが、演習を行うことはできません。

Rose がインストールされている場合は、ここで起動します。

- 1 [スタート] ボタンをクリックし、[プログラム] をポイントします。次に、[Rational Software] をポイントし、[Rational Rose] をクリックします。
- 2 55 ページの「要求の作成」の演習を行わなかった場合は、62 ページの手順で Rose を起動します。その後、この演習の次の手順に進みます。
- 3 [Create New Model] ダイアログ ボックスで、[Recent] タブをクリックします。
- 4 Rose モデル ClassicsCD_WinDNA をクリックして、[開く] をクリックします。サブユニットをロードするかどうかを確認するメッセージが表示された場合は、[はい] をクリックします。

左上のペインに階層ツリー ブラウザ (Rose ブラウザ) が表示されます。右側のペイン (図ウィンドウ) にはアーキテクチャの論理ビューが表示され、システム内の最上位パッケージが互いにどのように関わっているのかを見ることができます。

ビジュアル モデリングとは

ビジュアル モデリングとは、開発するシステムの構造と相互関係を表現した図を作成することです。これにより、システム アーキテクチャの青写真を描くことができます。ビジュアル モデルには、次のような特徴があります。

- テキスト形式ではなく図で表現されるので、システム全体を一目で簡単に把握できます。
- 設計コンポーネント間の関係を理解することができるので、より明確な設計を考案して、管理しやすいコードを記述することができます。
- ビジュアル モデルはプロジェクト要求に基づいているので、顧客のニーズに応じることができます。
- 標準のグラフィカル言語である統一モデリング言語 (UML) を使用してシステム アーキテクチャを伝えるので、チーム内のコミュニケーションが円滑になります。

Rational Rose の使用

業界標準の UML を Rational Rose で使用して、アーキテクチャ、コンポーネント、データのビジュアル モデルを作成することができます。ビジュアル モデルは、要求とアーキテクチャをコードに変換する前に視覚化し、理解し、改良するのに役立ちます。Rose では、開発ライフサイクル全体を通じてビジュアル モデルを開発することにより、目的に合ったシステムを確実に構築することができます。アーキテクチャ モデルは、ビジネス プロセス モデルとシステム要求の両方にまでさかのぼって追跡することが可能です。

ビジュアル モデリングとチュートリアル

「第 5 章 要求の作成」では、Rose と RequisitePro を使用して、ユース ケース要求を作成しました。この章では、Rational Rose を使用して、要求の設計に関する作業を続けます。

シーケンス図を使用した作業

シーケンス図は、ユース ケースの 1 つのパスでの手順を視覚的に表現したものです。プロジェクト メンバーと利害関係者は、シーケンス図 (グラフィック表現) かユース ケース要求 (テキスト形式の説明)、またはその両方を使用して、プロジェクトの方向性を評価し、プロジェクトの基準にすることができます。

多くの場合、ユース ケースには複数のパスがあります。常に存在するのは、ユース ケースで使用する最も一般的なパスについて説明した基本フローです。エラー条件などのほかのパスについて説明する代用フローが含まれていることもあります。

シーケンス図は、アクターがシステムとやり取りする方法とその順序を示します。シーケンス図での作業が初めての場合は、人間の話す普通の言語をラベルに使用することが多くなりがちですが、システム設計の改良を進めるに従い、図を変更して次の要素を特定できるようにします。

- クラス: 共通構造と共通動作を共有するオブジェクトのセットです。
- メッセージ: オブジェクト間の相互作用です。

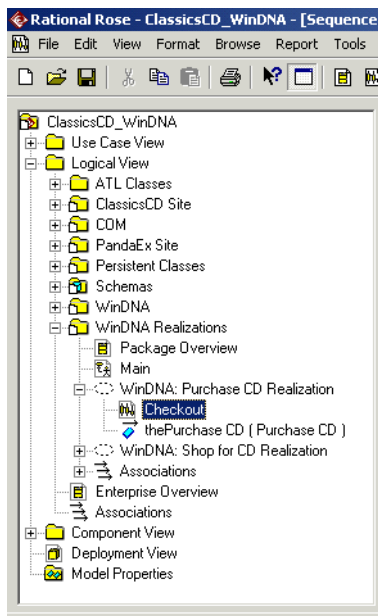
シーケンス図の表示

既存のシーケンス図を開くことから開始します。

演習: シーケンス図を開きます。

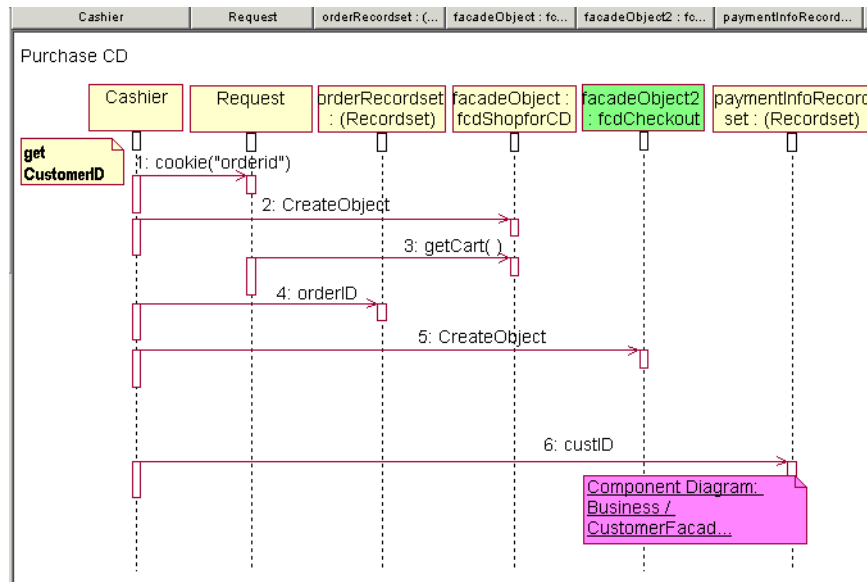
- 1 Rose ブラウザで、[ClassicsCD_WinDNA]、[Logical View]、[WinDNA Realizations]、[WinDNA: Purchase CD Realization]、[Checkout] の順にツリーを展開します。シーケンス図 [Checkout] をダブルクリックして開きます (図 24 を参照)。

図 24 Rose によるシーケンス図の表示



Rose に、Checkout シーケンス図が表示されます (図 25 を参照)。

図 25 Rose の Checkout シーケンス図



- 2 図全体が表示されない場合は、[View] メニューの [Fit in Window] をクリックします。

このシーケンス図には、アプリケーション内でアクターとほかのオブジェクトがどのようにやり取りするかが示されています。横方向の実線の矢印はメッセージシンボルです。縦方向の点線で示されたライフラインの間にあるメッセージシンボルは、シーケンス図に含まれるオブジェクト同士のやり取りを表します。シーケンス図の項目は、時系列で配置されています。

このチュートリアルでは、オブジェクトに次の命名規則を適用します。

- 名前には小文字で始めます。
- 名前にはスペースを使用しません。
- 名前内では、各単語の最初の文字を大文字にします。

シーケンス図には、顧客の発注後に ClassicsCD.com サーバーが Checkout プロセスを処理する方法の詳細も示されます。この処理では、Cashier Active Server Page によって Checkout Facade が呼び出され、メンバーのログインの確認、支払い情報の取得、注文の確定が実行されます。先頭にあるいくつかのメッセージには、具体的には次のような意味があります。


- 1 Cashier Active Server は、メンバーの ID を取得するために Cookie を要求します。
- 2 Cashier Active Server は、Facade オブジェクトを作成します。
- 3 Cashier Active Server は、Facade オブジェクトにメッセージを送り、メンバーのショッピングカートの中身に関する情報を取得します。

機能拡張用メッセージの追加

ここでは、メッセージをシーケンス図に追加します。メッセージでは、作業中の機能拡張を実装するために、システムのオブジェクトがどのようにやり取りするかを示します。

演習：シーケンス図に追加します。

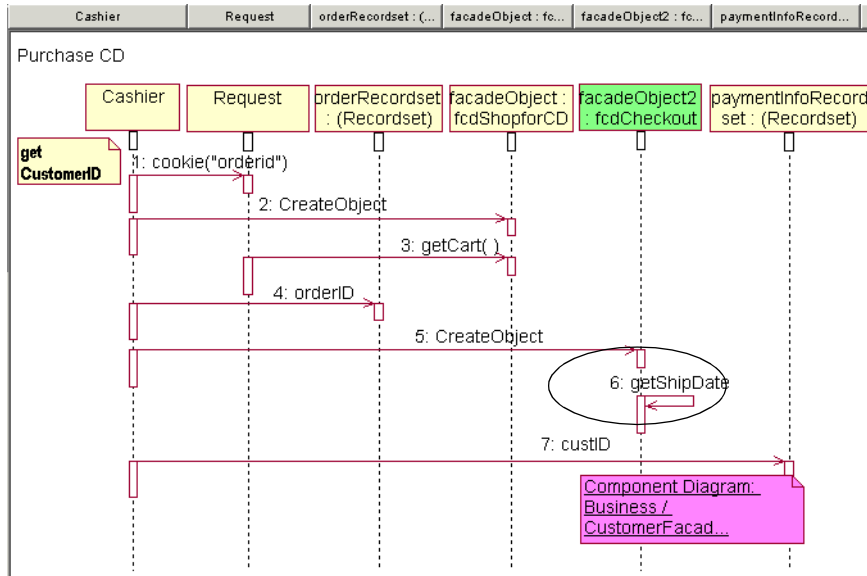
シーケンス図を開きます。

- 1 図のツールバー (Rose ブラウザと 図ウィンドウの間) で、 (再帰メッセージ ボタン) をクリックします。
- 2 メッセージを挿入するには、図で、メッセージ 5 の CreateObject のすぐ下にある facadeObject2:fdcCheckout ライフライン (緑色の Rose オブジェクトと関連付けられた縦線) をクリックします。

後続のメッセージの番号が変更されます。

- 3 新しく挿入した Message to Self オブジェクトが選択された状態のまま、「getShipDate」と入力し、図の背景部分をクリックします (図 26 を参照)。
- 4 [File] メニューの [Save As] をクリックします。モデルのサブユニットを保存するかどうかを確認するメッセージが表示された場合は、[はい] をクリックします。モデルのサブユニットを新しいフォーマットで保存するかどうかを確認するメッセージが表示された場合は、[はい] をクリックします。

図 26 Rose によるメッセージ追加後のシーケンス図



Web へのモデル要素の発行

モデルのこの部分での作業は終了しました。次に、モデルの Web バージョンを作成し、Rose をインストールしていないチーム メンバーもモデルを検討してフィードバックができるようにすることをお勧めします。

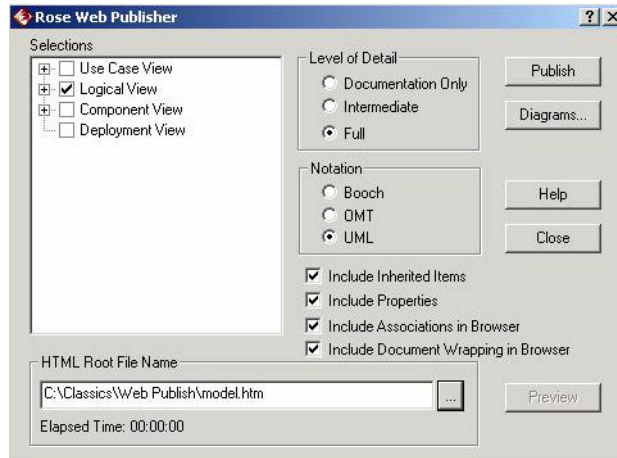
Rose Web Publisher により、Rose モデルの Web ベース (HTML) バージョンを作成して、ほかのメンバーが標準の Web ブラウザでモデルを参照できるようにすることができます。Rose Web Publisher では、図、クラス、パッケージ、関係、属性、操作などの Rose モデル要素を再作成します。Web バージョンの発行後は、ハイパーリンクを使用して、Rose を使用する場合と同じようにモデル内を移動することができます。

演習: モデルを Web に発行します。

- 1 Rose のメニュー バーで [Tools] メニューの [Web Publisher] をクリックして、[Rose Web Publisher] ダイアログ ボックスを表示します。
- 2 [Selections] ボックスで、次の操作を行います。
 - a [Use Case View] チェック ボックスをダブルクリックしてオフにします。
 - b [Component View] チェック ボックスをダブルクリックしてオフにします。
 - c [Deployment View] チェック ボックスをクリックしてオフにします。
 - d [Logical View] チェック ボックスをオンにします。

つまり、オンにするのは [Logical View] のみです (図 27 を参照)。

図 27 Rose による Web へのモデルの発行



- 3 [HTML Root File Name] ボックスで、... をクリックし、C:\Classics\Web Publish に移動します。[ファイル名] ボックスに「model」と入力し、[保存] をクリックします。

[HTML Root File Name] ボックスに、このパスが表示されます (図 27 を参照)。

- 4 [Rose Web Publisher] ダイアログ ボックスで、[Publish] をクリックします。

モデルにアクセスして HTML に変換する間、プログレス インジケータが表示されます。この処理には、数分かかる場合があります。Web ファイルの準備が整ったら、プログレス インジケータが消えます。

- 5 [Preview] をクリックします。

Web ブラウザの左上のペインに階層ツリー ブラウザ (Rose ブラウザ) が表示されます。Rose Web Publisher に関する情報が右側のペイン (図ウィンドウ) に表示されます。Rose モデルを選択すると、それが図ウィンドウに表示されます。

- 6 Rose Web Publisher ブラウザで、[Logical View]、[WinDNA Realizations]、[<<use case realization>> WinDNA: Purchase CD Realizations]、[Checkout] の順にツリーを展開します。[Checkout] をダブルクリックして、この章で作業したシーケンス図を参照します。

- 7 シーケンス図を参照したら、次のいずれかの操作を行います。

- この演習を開始する前に Rational Unified Process (RUP) を開いていた場合は、RUP に戻るまで [戻る] をクリックします。Rose の [Rose Web Publisher] ダイアログ ボックスで、[Close] をクリックします。
- この演習の前に RUP を開いていなかった場合は、Web ブラウザを閉じます。その後、[Rose Web Publisher] ダイアログ ボックスで、[Close] をクリックします。

シーケンス図での作業の継続

モデルのこの部分での作業は終了しました。次に、追加のタスクをいくつか実行します。このチュートリアルでは、これらのタスクについて説明するだけで、演習は行いません。

オブジェクトのリファイン

シーケンス図で、ユース ケースに関連のあるオブジェクトを特定します。次に、オブジェクトが属するクラスを特定します。Rose クラス図を使用して、関連クラスをグループ化し、その詳細を決定します。

たとえば、クラス図を表示するには、Rose ブラウザで、[Logical View]、[ClassicsCD Site]、[Client Composition Diagrams] の順にツリーを展開します。[cashier_Client Diagrams] をダブルクリックします。それぞれのクラス図には、クラスの属性と操作が示されます (クラス図をダブルクリックすると、クラスの詳細が表示されます)。

クラスを特定したら、シーケンス図を修正し、最初に割り当てた人間言語による名前の代わりにクラス名と操作名を使用します。

コードの実装

コードを実装する準備が整いました。作成した図を基に、Rose Enterprise Edition を使用して、開発したモデルと整合性のあるコード スケルトンを作成することができます。これは、フォワードエンジニアリングと呼ばれます。この処理では、開発者は生成されたコードから始めて、アルゴリズムの詳細を決定します。

新しいコードを生成するか、既存のコードを更新するには、Rose の [Tools] メニューからコマンドを選択します。たとえば、作業してきた機能拡張に Java コードを実装するには、[Tools] メニューの [Java / J2EE] をポイントし、[Generate EJB-JAR File] または [Generate WAR File] をクリックします。

コードの変更を開始するときに、モデルが古くなっている場合があります。コードの変更時に毎回モデルを手動で更新するのは面倒な作業です。また、すぐに古くなるモデルを作成するのも望ましくありません。Rose では、リバース エンジニアリングと呼ばれるプロセスによって、コードとモデルの整合性を維持することができ、コードの変更時にモデルが自動更新されます。ソース コードの更新後にリバース エンジニアリングを実行するには、[Tools] メニューの [Java / J2EE] をポイントし、[Reverse Engineer] をクリックします。

[Tools] メニューから分かるように、Rose では Java 以外の言語もいくつかサポートされています。サポートされている言語には、ANSI C++ や Visual Basic などがあります。

メモ: Rose Enterprise Edition と Rose Professional Data Modeler Edition では、コードの生成、コードの更新、モデルの更新を行うことができます。データベース スキーマについては、どちらのエディションでも、Rose ビジュアル モデルからコードを生成し、ラウンドトリップ エンジニアリングと呼ばれるプロセスによってソース コードから Rose モデルを更新することができます。

ます。ただし、Rose Professional Data Modeler Edition では、DDL スクリプトとデータベーススキーマの間のラウンドトリップ エンジニアリングのみがサポートされており、Java や Visual C++ などのほかの言語のラウンドトリップ エンジニアリングはサポートされていません。

データのモデリング

Rose を使用して、リレーショナル データベースをモデリングすることができます。Rose Professional Data Modeler Edition は、UML を使用してデータベースをモデリング、設計するツールです。次のような特長があります。

- テーブル、列、インデックス、リレーションシップ、キー (基本キーと外部キー)、ストアド プロシージャ、ビューの作成など、ほとんどのデータベース固有機能をサポートします。
- 列制約を作成し、DRI (宣言参照整合性) トリガと RI (参照整合性) トリガの両方を作成します。
- カスタム トリガと、カスタム トリガから生成されるトリガ コードを作成します。

利点

Rose Professional Data Modeler を使用することには、次の利点があります。

- すべてのビジネス モデル、アプリケーション モデル、データ モデルが同じ業界標準言語の UML で記述されるので、整合性と追跡可能性が向上し、やり取りが容易になります。
- リレーショナル データベースのフォワード エンジニアリングとリバース エンジニアリングの両方がサポートされているので、データベースの実装とモデルの整合性を維持するプロセスが容易になります。

まとめ

詳細

Rational Rose の詳細については、Rational Solutions for Windows のオンライン ドキュメント CD-ROM に収録されている『Rational Rose Tutorial』を参照してください。

Rational Rose RealTime の詳細については、Rose RealTime のヘルプからオンライン チュートリアルを参照してください。このチュートリアルは、Rose RealTime ユーザーのあらゆるレベルのニーズに合わせて作成されています。

オブジェクト指向の分析と設計の詳細については、[Help] メニューの [Extended Help] をクリックして拡張ヘルプを参照してください。[Extended Help - Rose] ツリーセットで、[Tool Mentor: Managing Subsystems Using Rational Rose] をクリックします。

クリーンアップ

必要に応じて Rose の拡張ヘルプを終了します。

Rose を終了します。変更を保存するかどうかを確認するメッセージが表示されたら、[はい] をクリックします。サブユニットを保存するかどうかを確認するメッセージが表示されたら、[はい] をクリックします。

この章の学習内容

この章では、次のことを学習しました。

- ビジュアル モデリングとは、開発するシステムの構造と相互関係を表現した図を作成することです。
- Rose では、シーケンス図を使用して、ユース ケースでのパスを詳細に決定します。
- Rational Rose によって、コードとデータのビジュアル モデルを作成し、ビジュアル モデルからコードを生成し、モデルと変更されたコードを常に同期をとることができます。
- Rose を使用して、モデルと図の読み取り専用コピーを Web に発行できます。この機能により、共有可能な高品質のアーキテクチャ モデルを作成することでチームをまとめ、すべてのチーム メンバーがプロジェクトに対して共通の理解を持つことができます。
- Rose では、ANSI C++、Visual Basic、Visual C++、Java など、多くの言語がサポートされています。

次の演習

これで、機能拡張のビジュアル モデルは完成しました。次の章では、機能拡張のためのユース ケースに関するレポートを作成します。

プロジェクト ステータスの 伝達

8

Arrange Shipment ユース ケースを新しい要求によって詳細決定したので、要求に加えられた変更をすべてのチーム メンバーと利害関係者に伝達します。変更を伝達するには、そのユース ケースに関するすべての情報を統合したレポートを作成します。そのようなレポートには、**Rational Rose** のビジュアル モデルからのシーケンス図と、**Rational RequisitePro** のユース ケースからの対応する基本フローを含めることができます。

また、プロジェクトのステータスについて伝達する場合もあります。**Rational Suite** ツールからプロジェクトのメトリクスを収集し、**ClassicsCD.com** プロジェクトの反復作業における進行状況の判断が必要になる場合があります。グラフは傾向を判断する助けとなり、ゲージはデータと定義済みのしきい値を比較するのに役立ちます。

この章では、**Rational SoDA** を使用してユース ケース レポートを作成し、**Rational ProjectConsole** を使用してプロジェクト成果物の表示、メトリクスの分析、プロジェクト ステータスの決定を行います。

対象読者

この章は、主にプロジェクト リーダーとマネージャーを対象としていますが、ソフトウェア開発チームのすべてのメンバーに関係する内容も含まれています。

概要

この章では、**SoDA** と **ProjectConsole** を使用します。**Rose** と **RequisitePro** もコンピュータにインストールされている必要があります。使用しているコンピュータにこれらのツールがインストールされているかどうかを確認するには、27 ページの「表 1 Rational Suite ツール」で記入したツール チャートを参照してください。

インストールされていないツールがある場合は、この章を読むことで多くのことを学習できますが、演習を行うことはできません。

これらのツールがインストールされている 場合は、ここで **SoDA** を起動します。[スタート] ボタンをクリックし、[プログラム] をポイントします。次に、[Rational Software] をポイントし、[Rational SoDA for Word] をクリックします。

Microsoft Word が起動し、[SoDA] メニューが追加された状態で空のドキュメントが自動的に開きます。以下の操作を行います。

- マクロを有効にするかどうかを確認する警告メッセージが表示された場合は、必ず [マクロを有効にする] をオンにします。このように設定にしないと、SoDA を使用できません。
- 警告メッセージが表示されない場合は、[ツール] メニューの [マクロ] をポイントし、[セキュリティ] をクリックします。[セキュリティ] ダイアログ ボックスが表示されます。[セキュリティ レベル] タブで、セキュリティ レベルが [中] または [低] に設定されていることを確認します。このように設定にしないと、SoDA を使用できません。この章の演習を終えたら、マクロのセキュリティ レベルを高いレベルに設定し直してかまいません。

プロジェクト ステータスの管理

Rational ProjectConsole と Rational SoDA は、プロジェクト ステータスの管理をサポートするツールを提供します。

SoDA とは

SoDA は、ステータス レポートのようなソフトウェア ドキュメントの作成を自動化します。多くの Rational ツールと緊密に統合されているので、情報を容易に抽出して、要求、設計、テスト、障害ステータスに関するレポートを作成できます。たとえば、SoDA を使用して以下の事項を報告できます。

- ClearCase LT によるバージョン管理情報
- ClearQuest によって報告された障害
- RequisitePro による要求
- Rose によるビジュアル モデルと設計
- TestManager によるテスト スクリプト
- Windows 環境で作成されたほかのドキュメントから抽出された情報
- 複数の Rational ツールとほかの情報ソースから抽出され、1 つの統合ドキュメントにまとめられた情報

SoDA テンプレートの使用

SoDA では、レポートやほかのドキュメントを生成する際に Word の定義済みテンプレートを利用します。テンプレートを使用して情報を収集し、レポート形式にまとめます。

SoDA レポートやドキュメントに手動で情報を追加する場合は、Microsoft Word または FrameMaker のインターフェイスを使用して、レポートまたはドキュメントを生成する前にテキストを追加します。このテキストは、後に続くレポートまたはドキュメントの生成サイクルの間、SoDA に保存されます。

SoDA に付属する多数のテンプレートのの中からテンプレートを選択することも、使いやすい SoDA のテンプレート作成ツールを使用して独自のテンプレートを作成することもできます。

ユース ケース レポートを生成する理由

ユース ケース レポートにより、期待されるシステム動作 (ユース ケース要求に記述) に関するテキスト形式での説明と、システムとアクターのやり取りを示す図の両方が、1 つのドキュメントに集約されます。ユース ケース レポートは、開発チームのすべてのメンバーにとって有用です。

- アナリストは、顧客とその他の利害関係者にレポートを提示し、プロジェクトが正しい軌道に乗っているかどうかを共に確認することができます。プロジェクトの早い段階でこうした話し合いを持つことができるので、アナリストは、プロジェクトが動き出してからではなく、動き出す前に問題点やギャップに対処することができます。
- 開発者は、期待されるシステム動作に関するレポートの説明を使用して、エンジニアリング仕様の作成とシステム アーキテクチャの設計を開始します。
- テスト担当者は、レポートを使用して、ユース ケース用のテストを設計します。レポートに基づいて、テスト手順を考え、テスト条件を決定することができます。
- テクニカル ライターは、ユーザーとシステムのやり取りに関するレポートの説明に基づいて、ドキュメントの計画を開始します。
- ユーザビリティ エンジニアは、レポートを使用して、ユーザビリティ テストを設計します。多くの場合、紙面上でのプロトタイプから始めます。

ユース ケース レポートの作成

SoDA では、ユース ケース レポートを作成する際に、定義済みのテンプレートを使用して情報を収集し、1 つのレポート形式にまとめます。この章で作成するレポートには、RequisitePro と Rose からの情報を含めます。「第 5 章 要求の作成」と「第 7 章 機能拡張のモデリング」の演習を実行していない場合は、ユース ケース レポートを作成できません。

演習: テンプレートから開始してユース ケース レポートを作成します。

- 1 Word で、C:\¥Classics¥Projects¥Webshop¥SoDA¥RUP Use Case Report.doc を開きます。

この章の初めてでマクロのセキュリティ レベルをリセットした場合、マクロを有効にするかどうかを確認する警告メッセージが表示されます。必ず [マクロを有効にする] をオンにします。このように設定にしないと、SoDA を使用できません。

Word に、テキスト、マクロ コマンド、コメントが配置された SoDA テンプレートが表示されます。

2 Word のツールバーで以下のいずれかの操作を行って、SoDA コマンドを含むテンプレート全体を表示します。

- Word 2000 以降を使用している場合
 - [ツール] メニューの [オプション] をクリックします。[表示] タブで、[編集記号の表示] にある [すべて] チェック ボックスをオンにします。[OK] をクリックします。
- Word 2002 を使用している場合
 - [SoDA] メニューの [Show/Hide Commands] をクリックし、次に [表示] メニューの [Markup] をクリックして、コメント マーカーを表示します。

SoDA によって使用されるピンクと黄色の Word コメントがドキュメントに表示されます。ここには、SoDA コマンドが格納されます (図 28 に示す Word 2000 の例を参照)。SoDA for Word で作業する場合は、この隠し文字が表示されるようにする必要があります。

図 28 SoDA によるユース ケース レポート テンプレートの表示 (抜粋)

```
[OPEN1]
[OPEN2][OPEN3]...[The title and table of contents will come from the use-case specification]
[DISPLAY4]<UseCaseSpecificationFormattedText>[ENDDISP5]

▪ 1. → Relationships

▪ 1.1 → Communicate--Associations

[MASTER6][REPEAT7][LIMIT8]

▪ 1.1.1 → Communicate with actor [DISPLAY9]<RoleA_ToClass.Name>[ENDDISP10]
Documentation: [DISPLAY11]<Communicate.Documentation>[ENDDISP12][ENDLIM13]

▪ 1.1.2 → [LIMIT14]Communicate with actor [DISPLAY15]<RoleA_FromClass.Name>[ENDDISP16]
Documentation: [DISPLAY17]<Communicate.Documentation>[ENDDISP18]
[ENDLIM19][ENDREP20][ENDMAST21][LIMIT22]No <<communicate>> associations have been
defined. [ENDLIM23]
```

メモ: 隠し文字は、使用する Word のバージョンによって表示がやや異なる場合があります。若干の差があっても、SoDA を使用して作業する場合は、編集記号の表示機能をオンにして、すべての隠し文字を表示してください。SoDA 以外で Word を使用する際に編集記号が表示されないようにするには、チュートリアルを終了後にこのオプションをオフにします。

3 [SoDA] メニューの [Generate Report] をクリックします。

Word が閉じ、以前に作業したユース ケース ドキュメントと Rose モデルから情報を取得する間、SoDA にプログレス インジケータが表示されます。この処理には、数分かかる場合があります。SoDA for Word にレポートが表示されます。

4 レポート内を移動します。

- a 4 ページ目を開き、「第 5 章 要求の作成」で作業した要求を確認します。
- b レポートの 6 ページ目に移動し、「第 7 章 機能拡張のモデリング」で作業したビジュアル モデルを確認します。

SoDA テンプレートの操作

SoDA テンプレートの操作はこのチュートリアルの対象範囲から外れますが、この項では基本的な事柄について説明します。

オプション演習: この SoDA テンプレートのテンプレート ツールを表示します。

- 1 Word で、[ウィンドウ] メニューの [RUP Use Case Report] をクリックして、ユース ケース テンプレートの RUP Use Case Report.doc を表示します。
- 2 [SoDA] メニューの [Template View] をクリックして、SoDA テンプレート ビュー ツールを表示します。

このテンプレート ビューを使用して SoDA テンプレートを作成または拡張します。
- 3 [Help] メニューの [Help on SoDA] をクリックして、このテンプレート ビューの各要素についての説明を読みます。

この情報により、各行がどのように RUP ユース ケース レポート テンプレートの SoDA コマンドに変換されるかがわかります。
- 4 SoDA のヘルプとテンプレート ビューを閉じます。
- 5 [File] メニューの [Exit] をクリックして SoDA for Word を終了します。変更を保存するかどうかを確認するメッセージが表示されたら、[いいえ] をクリックします。

ProjectConsole とは

ProjectConsole を使用すると、プロジェクト情報に簡単にアクセスすることができるので、進行状況と品質の測定に役立ちます。すべての Rational Suite ツールからプロジェクト成果物を自動的に収集し、それらの成果物からプロジェクト Web サイトを構成して、Web 対応のすべてのチーム メンバーが使用できるようにします。また、ProjectConsole では、要求、設計、テスト、障害ステータスについてのメトリクスを収集して、プロジェクトの真のステータスを容易に把握できるようにします。

プロジェクト Web サイトの使用

ProjectConsole では、ユース ケース レポート、障害、メトリクス レポートのようなプロジェクト成果物をホストする 1 つのプロジェクト Web サイトを公開し、チーム内のコミュニケーションを促進します。これらの成果物は、Rational Suite ツールまたは選択されたサードパーティ ツールから自動的に抽出されます。ProjectConsole により、成果物がユーザー定義の構造に従ってプロジェクト Web サイトに表示されます (図 29 を参照)。

こうした柔軟性があるので、プロジェクト リーダーは Web サイトの情報構造をチームまたは組織のニーズに最も合うように調整できます。プロジェクト リーダーは、すべてのチーム メンバーがプロジェクト成果物を簡単に見つけてアクセスできるように、表示する情報の種類と構成方法を決定します。プロジェクト Web サイトは、すべてのチーム メンバーが常に最新の成果物を見られるように、必要に応じてまたはスケジュールに合わせて更新できます。

図 29 ProjectConsole Web サイトの例



プロジェクト メトリクスを使用した作業

ProjectConsole では、ソフトウェア開発プロジェクト全体をとおして生成されたデータから情報を抽出し、グラフやゲージなどの定義済みまたはユーザー設定のグラフィックを自動的に生成します。これらのメトリクスにより、プロジェクトのステータスに関する情報を自動的に収集して、チーム メンバーと共有することができます。ProjectConsole を使用すると、複数の Rational ツールから収集されたデータを単一の統合されたビューで分析することも可能です。

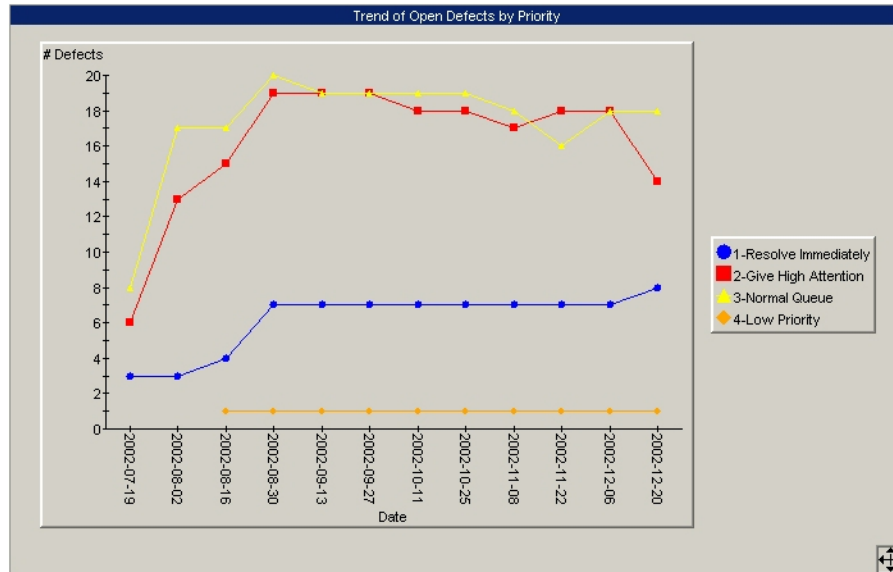
表 3 に、ProjectConsole を使用して進行状況と品質を測定する方法を示します。

表 3 プロジェクト ステータスを把握するための ProjectConsole メトリクスの使用

用途	収集データ	使用法
アプリケーションまたは製品が安定しているかどうかの判断	追加、変更、削除されるコード行	ClearCase LT
	追加、変更、削除されるビジュアル モデル要素	Rose
	報告された障害	ClearQuest
この反復で実行する必要がある追加タスクの数の表示	Open 状態の障害	ClearQuest
	Open 状態の機能、ユース ケース、要求	RequisitePro
	Open 状態のテスト ケース	TestManager
アプリケーションまたは製品の品質評価	重大度別の Open 状態の障害	ClearQuest
	テスト結果の傾向	TestManager

図 30 は、ProjectConsole を使用して、一定期間における Open 状態の障害の傾向を評価する方法を示しています。

図 30 ProjectConsole を使用した傾向グラフの操作



この傾向グラフでは、時間の経過と共に優先度の高い障害が減少しています。これにより、反復の終わりまでに開発者がアプリケーションの安定化を達成していると判断できます。

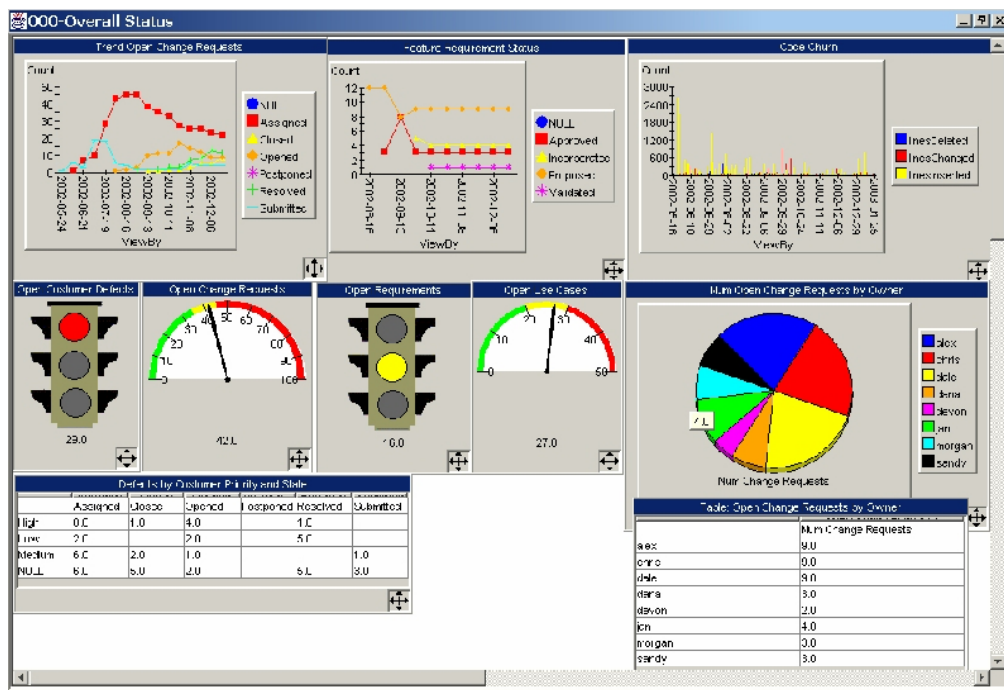
メトリクスの分析

ProjectConsole には動的なメトリクス分析ツールである Dashboard も備わっており、深く掘り下げて原因分析を行うことができます。

図 31 は、ProjectConsole Dashboard のインジケータとグラフを使用して全体的なプロジェクトステータスを評価する方法を示しています。また、図 31 は、Dashboard で次のことが可能であることも示しています。

- 各作業分野 (要求、設計、テスト) のメトリクスを表示する。
- データが現在のプロジェクト フェーズにどのように対応しているかを理解する。
- プロジェクト ステータスを分析する。

図 31 全体的なプロジェクトステータスの評価 (各作業分野)



まとめ

詳細

SoDA の詳細については、SoDA for Word で [Help] メニューの [Help on SoDA] をクリックしてください。SoDA についてのヘルプ トピックのリストを表示したウィンドウが開きます。関心のあるトピックを選択します。

ProjectConsole の使用法の詳細については、ProjectConsole のチュートリアルを参照してください。[スタート] ボタンをクリックし、[プログラム] をポイントします。次に、[Rational Software] をポイントし、[Rational ProjectConsole]、[Rational ProjectConsole Tutorial] をクリックします。

ProjectConsole Dashboard の使用法の詳細については、『Getting Started: Rational ProjectConsole』を参照してください。

ProjectConsole によるテンプレート作成の詳細については、Rational ProjectConsole Template Builder をインストールしてください。インストールしたら、[スタート] ボタンをクリックし、[プログラム] をポイントします。次に、[Rational Software] をポイントし、[Rational ProjectConsole Template Builder] をクリックして、ProjectConsole を起動します。Microsoft Word が起動し、[ProjectConsole] メニューが追加された状態で空のドキュメントが自動的に開きます。Word のメニュー バーで、[ヘルプ] メニューの [Help on Template Builder] をクリックします。トピックのリストを表示したウィンドウが開きます (必要に応じて、ウィンドウを最大化します)。関心のあるトピックを選択します。

クリーンアップ

必要に応じて Word を終了します。変更を保存するかどうかを確認するメッセージが表示されたら、[いいえ] をクリックします。

この章の学習内容

この章では、以下のことを学習しました。

- SoDA では、テンプレートに基づいてレポートを作成することにより、ソフトウェア ドキュメントの作成を自動化します。SoDA には使いやすいツールが備わっており、テンプレートの作成を支援します。
- ユース ケース レポートは、プロジェクトのすべてのメンバーにとって有用です。
- ProjectConsole では、Rational Suite ツールからプロジェクト成果物を自動的に収集し、それらの成果物からプロジェクト Web サイトを構成して、Web 対応のすべてのチーム メンバーが使用できるようにします。
- ProjectConsole では、ソフトウェア開発の期間中に作成されたデータから情報を抽出することにより、プロジェクト メトリクスを自動的に生成します。これにより、プロジェクト ステータスを正確かつ客観的に評価することができます。

次の演習

次の章では、Rational ツールを使用して信頼性テストを実行する方法を学習します。

これまでの章では、ClassicsCD.com の機能拡張を実装し、テストを計画しました。機能拡張をテストする準備が整っています。この章では、信頼性のテストについて説明します。

対象読者

この章は、テスト担当者、開発者、信頼性テストに関係するその他のチーム メンバーを対象としています。

信頼性テスト用のツール

この章では、以下の自動テスト ツールについて説明します。

- **Rational Purify:** Visual C++ アプリケーション コードのランタイム エラーやメモリ リークの箇所と、Java や .NET 管理アプリケーション コードのガーベッジ収集に関連するエラー箇所を特定します。
- **Rational PureCoverage:** Java プログラム、Visual C++ プログラム、Visual Basic プログラム、Visual Basic.NET プログラム、Visual C#.NET プログラムのどの部分が実行され、どの部分が実行されていないかを特定します。これによってテスト漏れの部分が明確になるので、テストされていないアプリケーション コードがユーザーに渡るのを防ぐことができます。
- **Rational Quantify:** Java アプリケーション、Visual C++ アプリケーション、Visual Basic アプリケーション、Visual Basic.NET アプリケーション、Visual C#.NET アプリケーションのプロファイリングを行い、コード内のパフォーマンスの問題を特定します。

Rational Suite のランタイム分析ツール

Rational Purify

ランタイム メモリ破壊エラーとメモリ リークは、見つけるのが困難で、修正の重要性が高いエラーとして挙げられます。これらのエラーはランダムなイベントによって発生するまで検出されないことが多く、プログラムが正常に動作しているように見えても実は障害があることもあります。

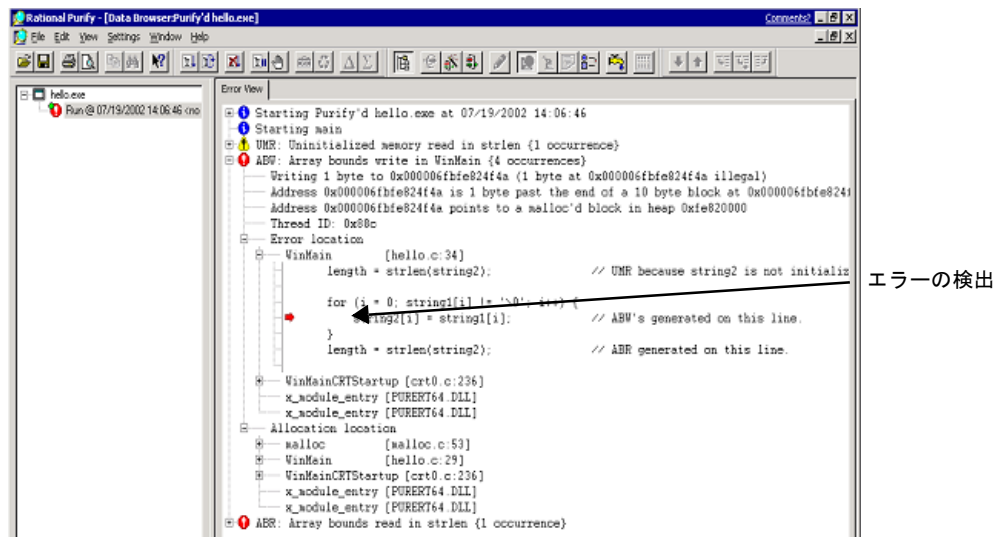
Rational Purify は、以下のものをチェックするための包括的なランタイム ツールです。

- C/C++ コード アプリケーション内のメモリ破壊とメモリ リーク
- Java や .NET 管理コード アプリケーション内のメモリ プロファイリングとメモリ リーク

ソース コードが分からない場合でも、Purify により、プログラムのすべてのコンポーネントでのメモリ エラーを検出することができます。ソース コードが分かっているアプリケーション領域でエラーが検出されると、無効なメモリ参照を引き起こしたコマンドが特定されて表示されます。Java や .NET 管理アプリケーションの場合、メモリ使用状況が分析されて報告されます。

また、Purify では、ユーザーがエラーの有無を調べるためにコードをチェックするときに、カバレッジ データが収集され、テストしていないプログラム部分が指摘されます。Purify のカバレッジ データは、すべてのコードにエラーがないことを確認するために使用することができます (図 33 を参照)。

図 32 Rational Purify からのサンプル エラー検出結果



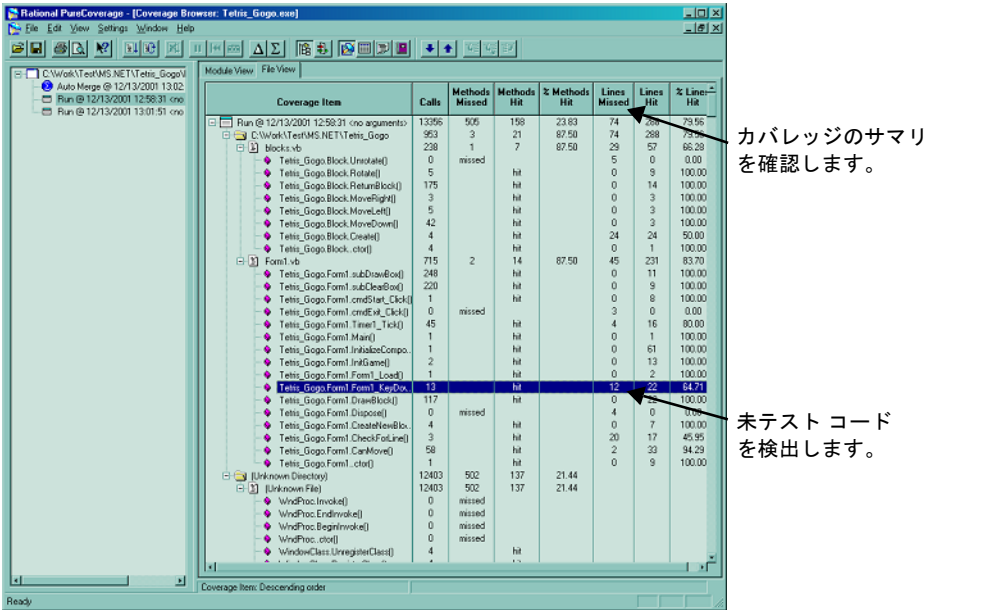
Rational PureCoverage

アプリケーションを効率よくテストするには、テスト時にテストが実行された部分と実行されなかった部分を把握する必要があります。この情報がなければ、致命的な問題領域を実際にテストしないまま、ソフトウェアの編集、コンパイル、デバッグのための貴重な時間を浪費する可能性があります。

Rational PureCoverage では、C/C++、Visual Basic 6、Visual Basic.NET、Visual C#.NET、Java コードのテスト漏れをすばやく簡単に特定することができます。

PureCoverage は、Rational Purify と Rational Robot の連携ツールとして特に有効です。たとえば、Purify によってすべてのメモリ エラーを検出するためと Robot によってすべてのアプリケーション機能をテストするためにコードを十分に検証したかどうかについて、PureCoverage からユーザーに通知することができます (図 33 を参照)。これは、自動テスト環境では欠かすことができません。

図 33 Rational PureCoverage からのサンプル テスト カバレッジ結果

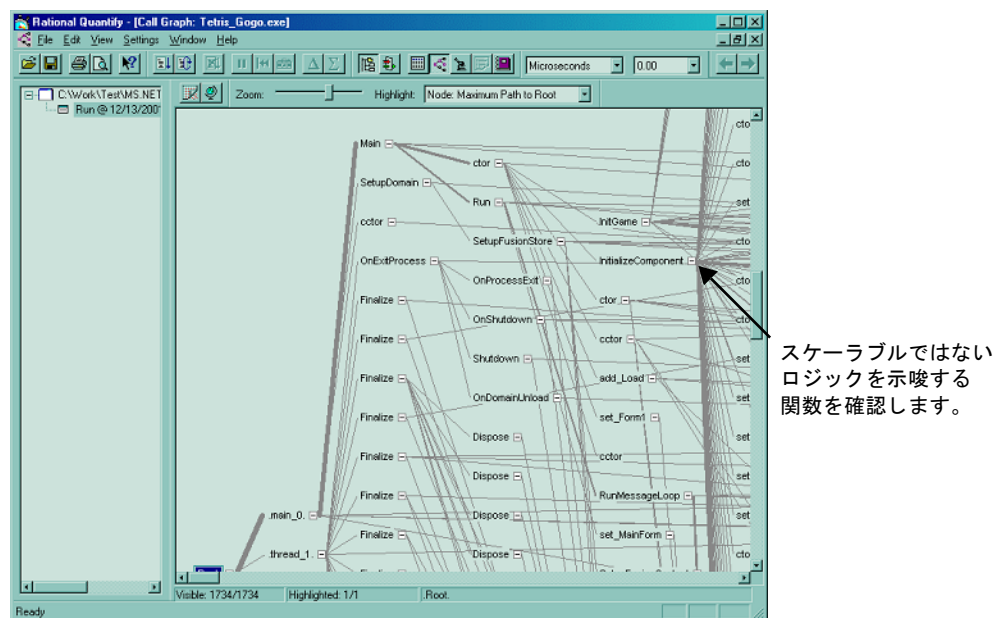


Rational Quantify

Rational Quantify では、C/C++、Visual Basic 6、Visual Basic.NET、Visual C#.NET、Java コードのパフォーマンスの問題をすばやく検出します。ソース コードが分からない場合でも、プログラムのすべてのコンポーネントに対して正確で反復可能なタイミング データを生成することにより、パフォーマンスの調整作業から困難さと当て推量を取り除きます。

Quantify を使用すると、効率的なコードを記述するために必要な情報を得ることができ、どんなプログラムでもより早く完成させることができます (図 34 を参照)。これにより、チーム全員をパフォーマンス エンジニアにすることも可能です。

図 34 Rational Quantify からのサンプル パフォーマンス分析



Purify、PureCoverage、Quantify をほかの Rational ツールと共に使用する

Rational Purify、PureCoverage、Quantify は、以下のツールと統合できます。

- **ClearQuest:** ランタイム分析ツールから直接障害をレポートしたり、信頼性に関する結果をほかのチーム メンバーとやり取りすることができます。
- **Robot:** テスト スクリプトの実行とアプリケーションのプロファイルを同時に行え、アプリケーションの信頼性と機能を同時にテストすることができます。これにより、チームの作業効率を向上させることができます。

まとめ

詳細

詳細については、以下を参照してください。

Purify、PureCoverage、Quantify: 『ファースト ステップ Rational PurifyPlus』、『ファースト ステップ Rational Purify』、『ファースト ステップ Rational PureCoverage』、『ファースト ステップ Rational Quantify』

Rational のマニュアルは、Rational Solutions for Windows のオンライン ドキュメント CD-ROM に収録されています。また、<http://www.rational.com/documentation> からオンラインで入手することもできます (ただし、英語のみのご利用となります)。

Purify、PureCoverage、Quantify の詳細については、各製品のヘルプを参照してください。

この章の学習内容

この章では、以下のことを学習しました。

- 開発者とテスト担当者は、以下のランタイム分析ツールを使用します。
 - Purify: C++ コードのランタイム メモリ破壊エラーを検出し、Java や .NET 管理コードのメモリ使用状況をプロファイルします。
 - PureCoverage: コード カバレッジを分析します。
 - Quantify: パフォーマンスの問題をピンポイント検出します。

次の演習

次の章では、ClassicsCD.com 機能拡張に機能テストを実行します。Robot を使用して、スクリプトの作成、既存のテスト ルーチンへのスクリプトの組み込み、テスト ルーチンの実行、テストで発見されたエラーの処理を行います。

開発プロセスのこの時点までに、開発者は **ClassicsCD.com** の機能拡張を実装し、テスト担当者は最初の信頼性テスト (このチュートリアル範囲外の作業) を実行しています。この章では、**Rational Robot** を使用して、拡張要求の機能テストを行うために「第 6 章 テスト計画」で作成したテスト ケースを実装します。また、**Rational TestManager** を使用して、実装結果を分析します。

対象読者

この章は、テスト担当者と機能テストに関係するその他のチーム メンバーを対象としています。

概要

この章では、最初に **Rational TestManager** の使用法について説明します。使用しているコンピュータに **TestManager** がインストールされているかどうかを確認するには、27 ページの「表 1 Rational Suite ツール」で記入したツール チャートを参照してください。

TestManger がインストールされていない場合は、この章を読むことで多くのことを学習できますが、演習を行うことはできません。

Robot がインストールされていない場合は、この章を読むことで多くのことを学習できますが、演習を行うことはできません。

TestManager がインストールされている場合は、ここで起動します。

- 1 [スタート] ボタンをクリックし、[プログラム] をポイントします。次に、[Rational Software] をポイントし、[Rational TestManager] をクリックします。
- 2 [Rational Test Login] ダイアログ ボックスが表示されたら、以下の操作を行います。
 - a [User Name] ボックスと [Password] ボックスに「pat」と入力します。
 - b [Project] ボックスに **Webshop** が表示されていることを確認します。
 - c [Location] ボックスに **C:¥Classics¥Projects¥Webshop¥Webshop.rsp** が表示されていることを確認します。
 - d [OK] をクリックします。

Webshop プロジェクトが開き、左側のペインにテスト アセット ワークスペースが表示されます。テスト アセット ワークスペースの下部にあるタブが正しく表示されているかどうかを確認します。各タブのシンボルにポインタをあててタブ名を確認し、タブ名がはっきり見えるまで垂直デバイダを移動してワークスペースを拡大します。これで、プロジェクトの作業を始める準備が整いました。

機能テストとは

機能テストにより、システムが意図したとおりに動作するかどうかを調べることができます。システム動作のテストとして最も一般的な方法は、アプリケーションの GUI を使用して、ユーザー入力に対するシステムの反応が適切かどうかを検証することです。テストでは、GUI オブジェクトの動作と外観の両方を対象にすることができます。TestManager には、Robot で作成した機能テストを実装、実行するサポート機能が組み込まれています。

テスト スクリプトの操作

テスト計画の作成時には、「第 6 章 テスト計画」の説明に従ってテスト ケースを作成します。テスト ケースには、アプリケーションのテスト対象範囲を記述します。テストを実行するための前提条件、テスト中の入力データ、検査する変数、各テストの期待される結果をリストにまとめることができます。

テストを実装するには、テスト ケースから開始して、テスト スクリプトを作成します。次に、テスト ケースとテスト スクリプトを関連付けます。テスト スクリプトには、以下のコンポーネントがあります。

- スクリプト名やスクリプトの目的などのプロパティ セット。
- スクリプト言語コマンドを含むファイル。スクリプト ファイルは、Robot またはほかのスクリプト言語とツールを使用してアクティビティを記録するときに生成します。

スクリプトとモジュール化

テスト スクリプトによって、アプリケーションを起動し、いくつかの手順を実行して、一定の目的を達成することができます。多くのテスト スクリプトで特定のアクティビティを実行する必要がある場合は、その共通アクティビティのみを実行するスクリプトを作成するのが効果的です。

また、同じ手順から開始してアプリケーションの異なる部分をテストして終了する一連のテスト スクリプトを作成することもできます。Robot では、短いモジュラー テスト スクリプトを作成し、TestManager のテスト ルーチンに組み込んで順番に並べることができます。この方法により、同じスクリプトをさまざまなテストで再使用したり、製品の連続したビルドに対してこれらのルーチンを繰り返して実行したりすることができます。または、既に記録されたテスト スクリプトを再使用して、アプリケーションを適切な開始点に設定し、そこから続くスクリプトを記録することができます。

たとえば、以下の各動作のためのテスト スクリプトを作成できます。

- システムへのログイン
- 購入品目の選択
- 購入の完了
- システムからのログオフ

その後、各テスト スクリプトを個別に実行するか、すべてのスクリプトをルーチンとして組み合わせ一度に連続して実行することができます。

開始点の設定

「第 6 章 テスト計画」では、ユーザーの拡張依頼に合わせて Display Estimated Ship Date テスト ケースを作成しました。この章では、記録されたテスト スクリプトを再使用して、ClassicsCD.com アプリケーションを適切な開始点に設定し、このテスト ケース用に新しいテスト スクリプトを記録します。

演習：Robot を使用する準備をします。

- 1 TestManager で、[Tools] メニューの [Rational Test] をポイントし、[Rational Robot] をクリックして、Robot を起動します。
- 2 Robot で、[Tools] メニューの [GUI Playback Options] をクリックします。
- 3 [GUI Playback Options] ダイアログ ボックスの [Log] タブで、以下のオプションがオンになっていることを確認します。
 - Output playback results to log
 - View log after playback
 - Specify log information at playback
- 4 [OK] をクリックします。
- 5 [File] メニューの [Exit] をクリックして、Robot を終了します。

テスト スクリプトの操作

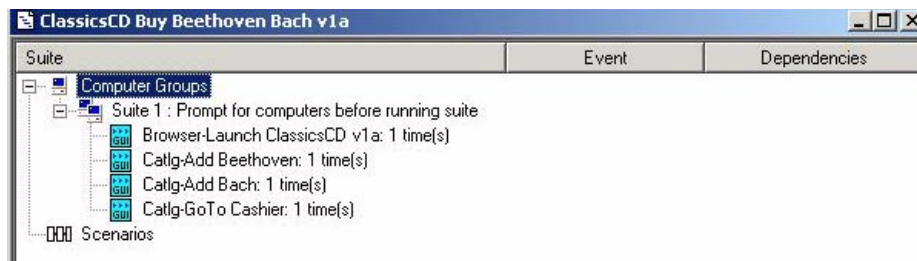
このチュートリアルでのテスト スクリプトは、コンピュータに Microsoft Internet Explorer がインストールされている場合にのみ記録、再生できます (このブラウザがデフォルト ブラウザである必要はありません)。Microsoft Internet Explorer がコンピュータにインストールされていない場合は、この章を読み進み、120 ページの「新しいビルドでのスクリプトの再生」から演習を開始してください。

演習：販売を完了するスクリプトを再生します。

- 1 TestManager のテスト アセット ワークスペース (左側のペインのツリー ブラウザ) にある [Execution] タブで [Suites] を展開し、[ClassicsCD Buy Beethoven Bach v1a] をダブルクリックします。

この ClassicsCD.com Webshop プロジェクト用テスト ルーチンのテスト スクリプトが、右側のウィンドウに表示されます (図 35 を参照)。

図 35 TestManager によるテスト スクリプトの表示



- 2 テスト アセット ワークスペースで、[ClassicsCD Buy Beethoven Bach v1a] を右クリックします。ショートカット メニューの [Run] をクリックします。

TestManager の [Run Suite] ダイアログ ボックスが表示されます。

[Build] ボックスで、[Build 1] が選択されていることを確認します。ほかの値についてはデフォルト設定を使用します。[Build 1] が選択されていない場合は、[Change] をクリックし、[Build] ボックスで [Build 1] をクリックします。

- 3 TestManager の [Run Suite] ダイアログ ボックスで、[OK] をクリックします。テスト ログを上書きするかどうかを確認するメッセージが表示された場合は、[いいえ] をクリックし、開いているテスト ログを保存または削除します。その後、手順 2 から演習を再開します。

TestManager がユーザーの依頼を処理してスクリプトを実行し、Robot がルーチンのスクリプトを再生している間は、ClassicsCD.com アプリケーションを操作しないでください。「今後、このページを開いたときにパスワードを入力する必要があるように、このパスワードを Windows に記録しますか？」という Windows のメッセージが表示されても、そのまま待機します。Robot の動作は続行されます。

スクリプトにより、2 品目の販売と支払いの取引が処理されます。特定の時点で、アプリケーションの値とベースラインの値が比較されます。スクリプトが終了すると、TestManager の [Test Log] ウィンドウにテスト結果が表示されます (必要に応じて、テスト結果が表示されるように ClassicsCD.com を最小化します)。すべてまたはほとんどの比較 (検証ポイント) が合格しています。

[Test Log] ウィンドウでは、イベント タイプの階層にあるスクリプトを展開することにより、結果の詳細を確認できます。場合によっては、行に「Unexpected Active Window」という警告メッセージが表示されることもあります。この警告メッセージは、再生中に別のウィンドウ (たとえば、「今後、このページを開いたときにパスワードを入力する必要があるように、このパスワードを Windows に記録しますか?」のメッセージボックス) が開かれたことを示します。この場合、Robot が開かれたウィンドウの存在を認識していても、そのウィンドウはテスト結果に影響していません。Robot に警告メッセージが表示された場合は、警告行をダブルクリックします。[Image Comparator] ウィンドウに、予期しないアクティブウィンドウのスクリーンショットが表示されます。スクリーンショットを参照したら、[Image Comparator] ウィンドウを閉じます。

4 [Test Log] ウィンドウを閉じます。

スクリプトの記録

アプリケーションの GUI の一部を実行中に、Robot を使用して、スクリプトを記録します。記録中は、ユーザーが実行するアクティビティがスクリプト言語コマンドに変換されます。Robot では、スクリプト言語として SQABasic が使用されます。SQABasic は、Microsoft Visual Basic に類似した言語で、自動テスト用のコマンドが追加されています。記録したスクリプトは、回帰テストやルーチンなどで再使用できます。

スクリプトの記録開始

ここまでで、記録済みのスクリプトを実行して、アプリケーションを一貫した既知の開始点に設定しました。「第 6 章 テスト計画」で作成した Display Estimated Ship Date テスト ケース用の新しいテスト スクリプトを作成して記録する準備が整いました。


演習: スクリプトを記録する準備をします。

ClassicsCD Buy Beethoven Bach v1a ルーチンが終了した時点から、スクリプトの記録を開始します。

- 1 ClassicsCD.com Web サイトを開きます。C:¥Classics¥ClassicsCD_com_sites¥v1a に移動して、Cashier.htm を開きます。

Web ブラウザに、ClassicsCD.com v1a の Cashier ページが表示されます。
ClassicsCD.com v1a は、ClassicsCD.com Web サイトの更新バージョンです。

- 2 Checkout ページをスクロールして、下部にある [Place Order] を表示します。出荷予定日が表示されます。

- 3 TestManager で、 (GUI スクリプト記録ボタン) をクリックします。

Rational Robot が起動し、[Record GUI] ダイアログ ボックスが表示されます。

- 4 [Name] ボックスに「Display Estimated Ship Date」と入力し、[OK] をクリックします。
[GUI Record] ツールバーが表示されます (図 36 を参照)。

図 36 [GUI Record] ツールバー



- 5 ClassicCD.com の Checkout ページが表示されているウィンドウをアクティブにします。

検証ポイントの作成

このテスト スクリプトでは、検証ポイントを作成して、オブジェクトのプロパティのベースライン値、またはアプリケーションの特定部分のデータを確立する必要があります。テスト スクリプトを再生すると、検出された値とユーザーが確立したベースラインの値が Robot によって比較されます。テスト スクリプトには、任意の数の検証ポイントを指定できます。

演習: スクリプトを記録し、検証ポイントを作成します。



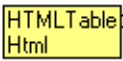
- 1 [GUI Record] ツールバーで、 をクリックします。
- 2 [GUI Insert] ツールバーで、 (オブジェクト データ ボタン) をクリックします。
[Verification Point Name] ダイアログ ボックスが表示されます。
- 3 [Name] ボックスに「Verify Estimated Ship Date」と入力し、[OK] をクリックします。
[Select Object] ダイアログ ボックスが表示されます。
- 4 チェック ボックス コマンドをオンにして、検証ポイントの選択後にこのダイアログ ボックスが自動的に閉じられるようにします。
- 5 手のポインタを、Checkout ページの「We estimate that your order...」で始まる行にドラッグします (図 37 を参照)。

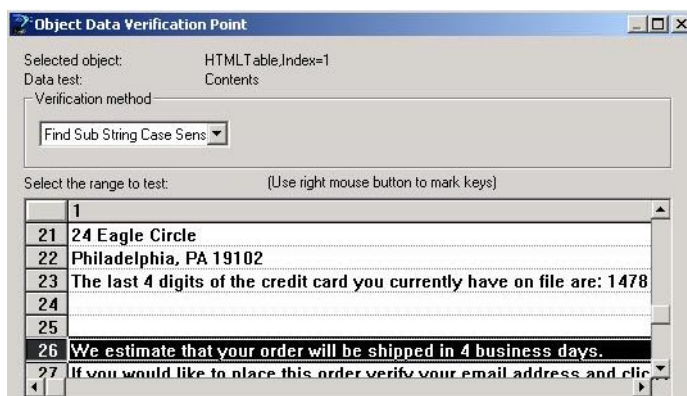
図 37 TestManager と Robot によるテスト検証ポイントの作成

We estimate that your order will be shipped in 4 business days.

If you would  this order verify your email address and click here:

- 6 [Object Data Tests] ダイアログ ボックスが表示され、HTML テーブルの内容を取得したことが示されます。[OK] をクリックして、取得したテキストを表示します。
- 7 [Object Data Verification Point] ダイアログ ボックスで、以下の操作を行います。
 - a [Verification method] ボックスで、[Find Sub String Case Sensitive] をクリックします。
 - b [Select the range to test] ボックスをスクロールして、26 行目 (「We estimate that...」) を表示し、クリックしてその行のみが選択されるようにします (図 38 を参照)。
 - c [OK] をクリックし、ダイアログ ボックスを閉じます。

図 38 TestManager によるテスト検証ポイントの操作



これで、検証ポイントの作成が終了しました。テスト スクリプトの記録もあと少しで終了します。次回このスクリプトが実行されると、取得したテキストがまだ表示されているかどうかを確認されます。スクリプトには、任意の数の検証ポイントを指定できます。このスクリプトの検証ポイントは 1 つだけです。

記録セッションの終了

記録セッションを終了します。

演習: スクリプトでの最後の手順を実行します。

- 1 Checkout ページで [Place Order] をクリックします。
- 2 [GUI Record] ツールバーで  をクリックして、記録を停止します。

Robot に、記録したばかりの Display Estimated Ship Date スクリプトが表示されます。
- 3 スクリプトを読み、スクリプトの記録中に実行した操作にコマンドがどのように対応しているかを確認します。
- 4 Robot を終了し、ClassicsCD.com を閉じます。

ルーチンへのテスト スクリプトの追加

Display Estimated Ship Date スクリプトを記録する前の演習では、ClassicsCD Buy Beethoven Bach v1a テスト ルーチンを実行してアプリケーションを設定しました。今後のテストでは新しい Display Ship Date スクリプトを繰り返し再生する必要がありますが、毎回アプリケーションを手動で設定するのは避けたい作業です。

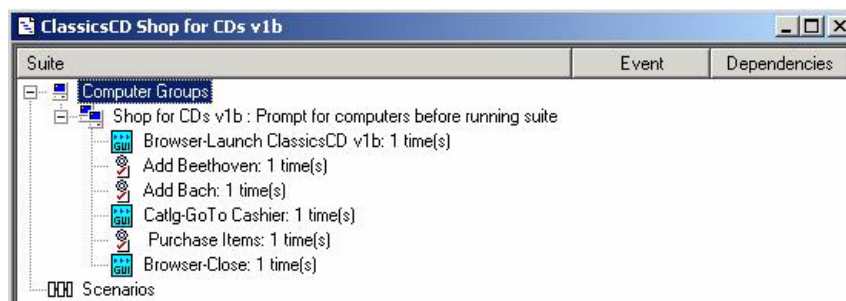
そこで、Display Ship Date スクリプトを既存のテスト ルーチンに追加し、テスト ルーチンからほかのスクリプトを呼び出すことで、アプリケーションを設定、シャットダウンすることができます。

演習: 新規スクリプトをテスト ルーチンに追加します。

- 1 TestManager のテストアセットワークスペースにある [Execution] タブで [Suites] を展開し、[ClassicsCD Shop for CDs v1b] をダブルクリックします。

このプロジェクト用テスト ルーチンのテスト ケースとテスト スクリプトが、右側のペインに表示されます (図 39 を参照)。

図 39 TestManager によるテスト ルーチン階層の表示



- 2 このビューで、[Purchase Items: 1 time(s)] テスト ケースを右クリックします。ショートカットメニューの [Insert] をポイントし、[Test Script] をクリックします。

[Run Properties of Test Script] ダイアログ ボックスが表示されます。

- 3 [Test script source] ボックスで、[GUI-(Rational Test Datastore)] をクリックします。
- 4 ダイアログ ボックスの [Select] セクションでスクロールし、[Display Estimated Ship Date] をクリックします。
- 5 [OK] をクリックします。

[Display Estimated Ship Date] テスト スクリプトが、階層の中に表示されます。

- 6 [File] メニューの [Save] をクリックします。

これで、テスト スクリプトがテスト ルーチンに関連付けられました。

開発したスクリプトとルーチンは、ソフトウェア ビルドごとに実行する回帰テスト セットを形成します。特定のテストの結果は、その後の反復の間に既存の障害が修正されたり、新しい障害やほかの変更が発生したりすると、変化する可能性があります。

テスト計画へのテスト スクリプトの組み込み

テスト スクリプトを作成して記録し、1 つまたは複数のルーチンに関連付けたら、テスト計画に組み込む必要があります。ClassicsCD テスト計画では、テスト ケース フォルダはユース ケースごとに整理されていました。「第 6 章 テスト計画」では、Display Estimated Ship Date テスト ケースを作成しました。次の演習では、新しいテスト スクリプトをこのテスト ケースに関連付けます。

演習: 新しいスクリプトをテスト ケースに追加します。

- 1 TestManager のテスト アセット ワークスペースにある [Planning] タブで、[Test Plans]、[ClassicsCD] の順に展開し、[ClassicsCD] をダブルクリックします。
ClassicsCD テスト計画の階層が表示されます。
- 2 このウィンドウで、[ClassicsCD]、[Purchase CD UC]、[UI]、[Order Confirmation Page]、[Display Estimated Ship Date] の順に展開します。
- 3 [Display Estimated Ship Date] を右クリックして、ショートカット メニューの [Properties] をクリックします。
テスト ケースの [Properties] ダイアログ ボックスが表示されます。
- 4 [Implementation] タブの [Automated implementation] で [Select] をクリックし、ドロップダウン メニューから [GUI - (Rational Test Datastore)] を選択します。
[Select Test Script] ダイアログ ボックスが表示されます。
- 5 スクロールして、[Display Estimated Ship Date] をクリックし、[OK] をクリックします。
- 6 [Test Case Properties] ウィンドウで [OK] をクリックします。
- 7 [File] メニューの [Save All] をクリックします。

これで、テスト スクリプトがテスト ケースに関連付けられました。

新しいビルドでのスクリプトの再生

テスト担当者と開発者は並行して作業する可能性があります。テスト担当者がテスト スクリプトとテスト ルーチンを作成する間に、多くの場合、開発者がアプリケーションの新しいビルドを作成しています。ここまでビルド 1 に対するスクリプトを記録しましたが、開発者によって UI に変更を加えたビルド 2 が開発されました。テスト ルーチンは、最新のビルドで実行することが重要です。

テスト担当者またはほかのチーム メンバーが、プロジェクトの初期にテスト ケースを実行することを望む場合もあります。チーム メンバーは、テスト計画にある任意のテスト ケースを右クリックして、ショートカットメニューからテスト ケースを実行できます。

この演習の目的から、また、現在は ClassicsCD.com 開発プロジェクトのバージョン 2 のテスト フェーズであることから、テスト スクリプトをテスト ルーチン自体から実行します。

演習: テスト ルーチンを最新ビルドで実行します。

- 1 TestManager のテスト アセット ワークスペースにある [Execution] タブで、[ClassicsCD Shop for CDs v1b] を右クリックし、ショートカット メニューの [Run] をクリックします。

TestManager の [Run Suite] ダイアログ ボックスが表示されます。[Log Information] 領域で、ビルド 1 が選択されていることを確認します。

- 2 [Change] をクリックします。[Build] ボックスで、[Build 2] をクリックします。ほかの値についてはデフォルト設定を使用します。[OK] をクリックします。
- 3 [Run Suite] ダイアログ ボックスで、[OK] をクリックします。

メモ: TestManager がユーザーの依頼を処理してスクリプトを実行し、Robot がテスト ルーチンのスクリプトを再生している間は、ClassicsCD.com アプリケーションを操作しないでください。「今後、このページを開いたときにパスワードを入力する必要がないように、このパスワードを Windows に記録しますか?」という Windows のメッセージが表示されても、そのまま待機します。Robot の動作は続行されます。

Robot はアプリケーションを起動してやり取りし、検証ポイントのプロパティとデータを取得して、アプリケーションを終了します。スクリプトの実行が終了すると、新しいテスト ログの [Details] タブにテスト結果が表示されます。

- 4 [Test Log] ウィンドウの [Details] タブをクリックして、テスト結果を表示します。

結果の分析

TestManager には、合格した検証ポイントと失敗した検証ポイントが表示されます。検証ポイントが失敗すると、スクリプトも失敗になります。TestManager のテスト ログを調査して、障害への対処方法を決定できます。

記録した Display Estimated Ship Date スクリプトは、ビルド 2 で UI の変更があっても合格します。ただし、ほかのスクリプトはこの新しいビルドで失敗していることに注意してください。

失敗への対処

特定のテストの結果は、その後の反復の間に既存の障害が修正されたり、新しい障害やほかの変更が発生したりすると、変化する可能性があります。スクリプトの失敗には、以下の 2 つのタイプがあります。

- 意図的な変更によるもの。スクリプトはアプリケーションに対して計画された変更によって失敗します。この場合は、検証ポイントのベースラインを変更する必要があります。
- 実質的なエラー。正しいベースラインを設定していてもスクリプトが失敗する場合です。実質的なエラーを報告するには、Rational TestManager と統合されている Rational ClearQuest を使用して、障害レコードを登録します。

意図的な変更への対処

演習：最初の失敗を検査します。

- [Test Log] ウィンドウの [Details] タブで、[Computer Start (Shop for CDs v1b [1])]、[TestCase Start (Purchase Items)]、[Script Start (Catlg-Purchase Items)] の順に展開します (図 40 を参照)。

図 40 TestManager による最初の失敗の選択

Event Type	Result
Suite Start (ClassicsCD Shop for CDs v1b)	Fail
Computer Start (Shop for CDs v1b [1])	Fail
Script Start (Browser-Launch ClassicsCD v1b)	Pass
TestCase Start (Add Beethoven)	Pass
TestCase Start (Add Bach)	Pass
Script Start (Catlg-GoTo Cashier)	Pass
TestCase Start (Purchase Items)	Fail
Script Start (Catlg-Purchase Items)	Fail
Verification Point (Page Header - Object ...)	Fail
Verification Point (Order Summary - Objec...	Fail
Script End (Catlg-Purchase Items)	Fail
TestCase End (Purchase Items)	Fail
Script Start (Display Estimated Ship Date)	Pass
Script Start (Browser-Close)	Pass
Computer End	Fail
Suite End (ClassicsCD Shop for CDs v1b)	Fail

- 2 失敗である [Verification Point (Page Header - Object Data)] をダブルクリックします。

ページ ヘッダーに対応する Grid Comparator が表示され、ページ ヘッダーが [Checkout] から [Cashier] に変更されたことが示されます。これは、変更が計画されたものであり、失敗は UI の変更によるものであることを意味します。

この場合は、ベースラインを変更して、次回スクリプトが再生されたときにページ ヘッダーと新しい値が比較されるようにします (ベースラインを変更するのは、アプリケーションへの意図的な変更によってテストが失敗した場合です)。

- 3 [Grid Comparator] ウィンドウで、[File] メニューの [Replace Baseline with Actual] をクリックします。

置換するかどうかを確認するメッセージが表示されます。

- 4 [Yes] をクリックします。

ベースラインとレポートが更新され、両者の間に違いがなくなります。

- 5 Grid Comparator を閉じます。

実質的なエラーへの対処

演習：別の失敗を検査します。

- 1 [Test Log] ウィンドウで、次の失敗である [Verification Point (Order Summary - Object Data)] をダブルクリックします。

Grid Comparator が表示されます。

ベースラインの 2 行目は、ユーザーはベートーベンの交響曲を購入するはずだったが、実際にはショッピング カートにモーツアルトの交響曲が入っていることを示しています。これは実質的なエラーです。これらの CD の価格は異なるので、このエラーによって購入価格が違ってきます。そして、Grid Comparator の対応する行に、違いがあることを示すフラグが付けられます。

- 2 Grid Comparator を閉じます。

エラーの報告

実質的なエラーを報告するには、TestManager と統合されている ClearQuest を使用します。

演習：エラーを報告します。

- 1 [Verification Point (Order Summary - Object Path)] を右クリックし、ショートカット メニューの [Submit Defect] をクリックします。

このメニュー コマンドが使用できない場合は、コンピュータに ClearQuest がインストールされていないので、この演習を終了できません。

コンピュータに ClearQuest がインストールされている場合は、[ClearQuest Login] ダイアログ ボックスが表示されます。

メモ: Webshop プロジェクトに変更依頼データベースを関連付けた場合は、次に進む前に、32 ページの演習を終了してください。

2 [ClearQuest Login] ダイアログ ボックスで以下の操作を行います。

a [User Name] ボックスと [Password] ボックスの両方に「pat」と入力します。

b [Database] ボックスで、[CLSIC] が選択されていることを確認します。選択されていない場合は、リストから [CLSIC: Rational Demo] を選択します。

CLSIC は、ClassicsCD.com への変更依頼 (障害と拡張依頼) を格納するデータベースの名前です。

c [OK] をクリックします。

[Submit Defect] ダイアログ ボックスが開き、障害に自動的に番号が付き、入力が必須なボックスの項目は赤で示されます。すべての必須ボックスに有効な値を入力しないと、障害を登録できません。

3 [Headline] ボックスに、「Wrong item in shopping cart」と入力します。

4 [Severity] ボックスから [1- Critical] を選択します (図 41 を参照)。

図 41 ClearQuest による障害の登録

Submit Defect CLSIC00000130

Iterations & Notes | Unified Change Management | Requirements | Main | Test Data | Environment | Attachments

ID: CLSIC00000130 State: Submitted

Headline: Wrong item in shipping cart

Suite Project: Webshop Keywords:

UCM Project: Keywords:

Owner: Symptoms:

Priority: Symptoms:

Severity: 1-Critical

Customer Priority:

Description: Auto-Generated by TestManager

OK Cancel

5 [Test Data] タブに移動します。TestManager により、この障害のテスト スクリプトに関連するボックスに既に値が入力されています。

6 [OK] をクリックして、[Submit Defect] ダイアログ ボックスを閉じます。

- 7 [Test Log] ウィンドウで右にスクロールして、[Defects] 列を表示します。入力した障害が、検証ポイントに自動的に関連付けられたことを確認してください。

これで、ClassicsCD.com のこの反復のテストは終了しました。

まとめ

詳細

テスト方針の詳細については、いずれかの Rational Test ツールで [Help] メニューの [Extended Help] をクリックしてください。拡張ヘルプブラウザで、目的のトピックを読みます。

Rational Test ツールの詳細については、『Rational TestManager User's Guide』と『Rational Robot User's Guide』を参照してください。両方とも、Rational Solutions for Windows のオンライン ドキュメント CD-ROM に収録されています。

クリーンアップ

TestManager を終了します。テスト結果を保存するかどうかを確認するメッセージが表示されたら、[Yes] をクリックします。

必要に応じて ClassicsCD.com を終了します。

この章の学習内容

この章では、以下のことを学習しました。

- 機能テストにより、システムが意図したとおりに動作するかどうかを調べることができます。
- Rational TestManager では、機能テストを計画、開発、実行、分析できます。
- テスト スクリプトは、Rational Robot を使用してアプリケーションとやり取りし、スクリプトに検証ポイントを設定することによって開発します。
- モジュール化されたスクリプトを開発し、ルーチンを使用してそれらのスクリプトを呼び出すことができます。開発者が新しいソフトウェア ビルドを開発するたびに、スクリプトを再使用します。
- Robot により、テスト中に発見された問題への対処が容易になります。
- Rational ClearQuest と TestManager の統合により、エラー報告プロセスの大部分が自動化されます。

次の演習

開発の反復でのアプリケーションをテストする方法について学習したので、チュートリアルをほとんど終了したことになります。次の章では、ClassicsCD.com の次の反復を計画します。

ClassicsCD.com の機能拡張はこれで完了です。この反復での学習は終了しました。この章では、次の反復の計画を開始する際の手順について説明します。

対象読者

この章は、ソフトウェア開発チームのすべてのメンバーを対象としています。

概要

この章では、Rational ClearQuest を使用します。使用しているコンピュータに ClearQuest がインストールされているかどうかを確認するには、27 ページの「表 1: Rational Suite ツール」で記入したツール チャートを参照してください。

ClearQuest がインストールされていない場合は、この章を読むことで多くのことを学習できますが、演習を行うことはできません。

ClearQuest がインストールされており、Webshop プロジェクトを変更依頼データベースに関連付けた場合は、32 ページの演習を完了していることを確認してください。次に、ClearQuest を起動します。

1 [スタート] ボタンをクリックし、[プログラム] をポイントします。次に、[Rational Software] をポイントし、[Rational ClearQuest] をクリックします。

2 [ClearQuest Login] ダイアログ ボックスで以下の操作を行います。

a [User Name] ボックスと [Password] ボックスの両方に「pat」と入力します。

b [Database] ボックスで、[CLSIC] が選択されていることを確認します。選択されていない場合は、リストから [CLSIC: Rational Demo] を選択します。

CLSIC は、ClassicsCD.com への変更依頼 (障害と拡張依頼) を格納するデータベースの名前です。

c [OK] をクリックします。

ClearQuest に 2 つのペインが表示されます。左側のペインには、ユーザーが表示できるグラフとレポートの階層が表示されます。右側のペインは空になっています。

プロジェクトの状態の評価

「第 10 章 機能テスト」では、ClearQuest を使用してソフトウェアの障害を報告しました。この章では、ClearQuest を使用してプロジェクトの状態を評価する方法について学習します。

Rational ClearQuest は変更依頼を管理するためのツールです。ClearQuest では、Rational のプロジェクトに関連付けられたすべてのアクティビティ（障害や拡張依頼など）を追跡し、管理できます。

情報をユーザー データベースに格納する ClearQuest には、すぐに使用できるスキーマが備わっています。スキーマには、ユーザー データベースのフィールドを記述します。ClearQuest は変更が容易なので、管理者は、クエリー、レコード、フィールド、アクティビティ、状態を開発プロセスに合わせてカスタマイズ、定義することができます。

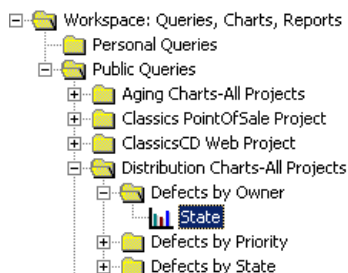
作業量の表示

反復の終わりには各プロジェクト メンバーの作業量を見直し、次の反復で最も効率良く作業を分担できるようにします。ClearQuest を使用すると、作業量をグラフで表示できます。作業量グラフでは、特定のチーム メンバーの作業量に関する情報を詳しく見ることができます。この機能は、各メンバーにアサインされた障害と関連の詳細情報を調べる場合に役立ちます。

演習：作業量を示すグラフを表示します。

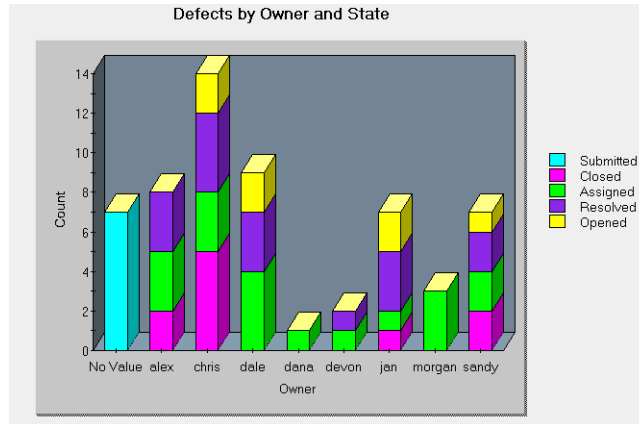
- 1 ClearQuest の左側のペインで、[Public Queries]、[Distribution Charts - All Projects]、[Defects by Owner] の順にツリーを展開します。
- 2 [State] エントリをダブルクリックします（図 42 を参照）。

図 42 ClearQuest でのクエリー ビューの選択



ClearQuest に作業量グラフが表示されます。必要に応じて、グラフを最大化して細部が明確に見えるようにします。左側にある青緑色のバーは、未アサインの障害を表します (図 43 を参照)。

図 43 ClearQuest による障害と作業量の表示



- 3 Jan のバーのさまざまな場所をクリックして、状態と障害数の詳細を確認します。
- 4 青緑色のバー (左端のバー) をダブルクリックして、このバーにまとめられている障害のリストを表示します。これらの障害は、まだチーム メンバーにアサインされていません。
クエリーを作成するかどうかを確認するメッセージが表示されます。
- 5 [OK] をクリックします。

[Result Set] タブに、障害のリストと選択した障害の詳細が表示されます。特定の障害についてすばやく掘り下げて調べ、必要に応じて修正することができます。ClearQuest のこの機能により、次の反復が計画しやすくなります。たとえば、この ClearQuest クエリーを使用して、以下の作業を行うことができます。

- 未アサインの障害の表示
- 障害のアサイン ([Main] タブを使用)
- 要求への障害のリンク ([Requirement] タブを使用)

- 6 「第 10 章 機能テスト」で登録した障害 (ID CLSIC00000130) がこのリストに表示されています (図 44 を参照)。

図 44 ClearQuest による障害の表示

id	Owner	State
CLSIC00000073		Submitted
CLSIC00000074		Submitted
CLSIC00000084		Submitted
CLSIC00000088		Submitted
CLSIC00000089		Submitted
CLSIC00000091		Submitted
CLSIC00000092		Submitted
▶ CLSIC00000130		Submitted

Result set / Query editor / Display editor /

PQC / Test Data / Environment

Unified Change Management / ClearCase / Requirements

Main / Analysis / Notes / Resolution / Attachments / History

ID: CLSIC00000130 State: Submitted

Headline: Wrong item in shipping cart

Suite Project: Webshop Keywords:

U/CM Project: Symptoms:

Owner: Priority: 1-Critical

Severity: Customer Priority: Description: ***** Auto-Generated by TestManager *****

Apply

Revert

Print Record

Actions ▼

ID: 00000130

拡張依頼の操作

チュートリアルでは、拡張依頼を開くことから始めて、その後その拡張を実装しました。次の演習では、次の回の反復で実装する拡張依頼についてさらに学習します。

演習：拡張依頼を検討します。

- 1 ClearQuest の左側のペインで、[Public Queries]、[ClassicsCD Web Project] の順に展開し、[All Project Enhancement Requests] をダブルクリックします。
- 2 右側のペインで、[CLSIC00000036, Need to notify customer via email when order ships] をクリックして、拡張依頼の詳細を表示します。拡張依頼は Submitted 状態になっていますが、Assigned 状態にはなっていません。[History] タブをクリックして、拡張依頼とその履歴を確認します。

- 3 今度は、[CLSIC00000031, Need to notify customer via e-mail when order ships] をクリックして、拡張依頼の詳細を表示します。拡張依頼が、Assigned 状態になっています。この依頼を割り当てられたメンバーを確認するには、[Analysis] タブをクリックします。

どちらの ClearQuest エントリも、このオンライン ストアと同じ拡張を依頼しています。通常は、同一ではなく非常に類似した拡張依頼または障害を持つ複数のエントリは、変更依頼データベースに格納されます。これにより、プロジェクト チームはそれぞれの ClearQuest エントリを評価、つまり選別することができます。チーム メンバーは、依頼や障害を誰に割り当てるかを判断できるだけでなく、既存または新規の依頼の重複になっているエントリを特定して変更し、重複について詳しく記述し、この解決方法によってエントリを閉じることができます。

演習：拡張依頼を変更します。

- 1 [CLSIC00000036, Need to notify customer via email when order ships] で開いた拡張依頼に戻ります。類似の依頼がすでに割り当てられていることが分かっているので、この依頼を重複として閉じます。
- 2 [Main] タブで、[Actions] をクリックし、[Duplicate] をクリックします。
- 3 [Mark as Duplicate] ダイアログ ボックスで、「31」と入力し、[Find] をクリックします。
[Entre ID] フィールドの値が、拡張依頼の完全な ID である CLSIC00000031 に変わります。
- 4 [Show more details] をクリックして、このレコードが、CLSIC00000036 を重複元とする正しいレコードであることを確認します。
- 5 データを確認したら、[Cancel] をクリックして [Mark as Duplicate] ダイアログ ボックスに戻ります。
- 6 [OK] をクリックして、CLSIC00000036 を CLSIC00000031 の重複としてマークします。
タブのボックスで入力が必須なものには、赤い四角が表示されます。
- 7 [Analysis] タブを開きます。
 - a [Owner] ボックスに赤いマークが付いているので、値の入力が必須です。このチュートリアルでは、pat としてログインし、このエントリの解決方法を判断しようとしていることになっています。したがって、[Owner] ボックスで [pat] をクリックします。
- 8 [Resolution] タブを開きます。TestManager により、このレコードの重複に関連するボックスに既に値が入力されています。
- 9 [Apply] をクリックします。場合によっては、ツールバーの [Run Query] をクリックして、クエリー結果を更新する必要があります。

拡張依頼 CLSIC00000036 の状態が Duplicate になります。

その他の計画アクティビティ

反復では、通常は障害の修正と機能拡張の実装の両方の作業を行います。計画作成の一環として、ClearQuest または RequisitePro を使用して、次の反復で行う作業を決定することもできます。

反復の計画時には、Rational SoDA レポートを作成して、次の反復に計画された障害と機能拡張を示すことができます。また、ProjectConsole により、RequisitePro や ClearQuest のようなツールから収集したメトリクスを使用して、グラフやゲージを自動生成することも可能です。このレポートを使用すると、次のプロジェクトのステータスを分析してチーム メンバーと共有することができます。

次の反復での作業

次の反復では、今回の反復と同じように、多くの作業を行います。計画の作成を終了すると、以下のアクティビティが明らかになります。

- すべてのチーム メンバーは、Rational Unified Process (RUP) を、ソフトウェア開発における最善の実践原則のガイドとして、またソフトウェア開発の情報ソースとして、プロジェクト全体を通じて使用します。
- プロジェクト リーダーとマネージャーは、RUP Builder を使用して、次の反復のニーズにより適うように RUP 構成を改良します。
- すべてのチーム メンバーは、統一変更管理 (UCM) をプロジェクト全体を通じて使用して、システム開発における変更を管理します。
- すべてのチーム メンバーは、Rational Developer Network を使用して、ターゲットとする開発内容、スキル上達のためのリソース、Rational Suite ユーザーのオンライン コミュニティにアクセスします。
- アナリストは、計画された機能拡張について利害関係者と話し合います。RequisitePro と Rose を使用して、アナリストは 1 つまたは複数のユース ケースを作成し、基本フローや代用フローなどの詳細を段階ごとに示します。
- テスト担当者は、Rational TestManager を使用して、この反復のテストを計画します。エンジニアは、テスト計画を作成し、テスト要求を開発して、テストを設計します。
- 開発者は、Rose に備わっているビジュアル モデリング技術を使用して、計画された機能拡張がどのようにシステム アーキテクチャに適合するかを説明します。
- すべてのチーム メンバーは、統合ソフトウェア プロジェクト管理 (USPM) を使用して情報をコンパイルし、ステータス、傾向、品質、その他プロジェクト管理とレポート作成にとって重要な要素を評価できるようにします。

- すべてのチーム メンバーは、**ProjectConsole** と **SoDA** を使用して、プロジェクト メトリクスを収集し、プロジェクト レポートを作成します。グラフ、ゲージ、レポートの形式で表される情報は、利害関係者との話し合いや、設計セッションにおいて役立ちます。
- 開発者は、**Rose** を使用して、機能拡張の実装を開始します。
- 開発者とテスト担当者は、**Purify**、**Quantify**、**PureCoverage** を使用して、反復の信頼性を検証します。
- テスト担当者は、**TestManager** と **Robot** を使用して、機能拡張が要求を満たしていること、障害が修正されていること、回帰障害が発生しないことを検証します。
- すべてのチーム メンバーは、**ClearCase LT** を使用して、プロジェクト成果物に変更を加えます。プロジェクト メンバーは、それぞれプライベートな開発ワークスペースで作業します。各チーム メンバーは、自分の作業が終了すると、チームの共用統合ワークスペースに成果物を提出します。
- プロジェクト リーダーとマネージャーは、**ProjectConsole**、**ClearQuest**、**SoDA**、**RequisitePro** を使用して、要求からリリースまでのプロジェクトの状態を評価します。その後、これらのツールを使用して、後に続く反復を計画します。

まとめ

詳細

このチュートリアルで説明した特定のトピックについてさらに詳しく学習するには、ラショナル ユニバーシティのコースを受講することをお勧めします。これらのコースでは、特定の **Rational** ツールを実際に操作したり、ソフトウェア開発の原則について詳しく学習したりすることができます。コースの詳細については、<http://www.rational.co.jp/services/ru/> を参照してください。

クリーンアップ

準備が整ったら、**ClearQuest** を終了します。

Rational Unified Process が開いている場合は、ここで閉じます。または、開いたままにして、**Rational Suite** について自分で詳しく学習する際の補足手段として使用することもできます。

この章の学習内容

この章では、以下のことを学習しました。

- **ClearQuest** は、プロジェクトにおける変更依頼の管理と監視に使用できる強力なツールです。
- 次の反復はどのように進行するのかと、プロジェクトの計画後に発生するアクティビティについて。

このチュートリアルの学習内容

- Rational Suite は、チーム内のコミュニケーションを促進してチームを統一します。
- Rational Suite は、ソフトウェア開発チームのすべてのメンバーに業界最先端の開発ツールを提供することにより、各チーム メンバーの生産性を高めます。
- Rational Suite は、単純なインストール、ライセンス、ユーザー サポート機能を備える包括的な統合ツール セットを提供することにより、導入作業を簡素化します。
- Rational Suite は、開発ライフサイクル全体と、アナリスト、開発者、テスト担当者、マネージャーなどの開発チームにおける主要な役割をサポートします。

次の演習

これで、Rational Suite のチュートリアルは終了です。

Rational Suite を使用してアプリケーションの計画、設計、実装、テストを行うことにより、ユーザーの開発チームは高品質のソフトウェアを短期間で開発するという課題を見事に達成できるでしょう。

次にやらなければならないことは、今後のプロジェクトで使用するツールについて学習することです。今から始めてください。

オプションのアクティビティ : Rational Developer Network に参加します (ただし、英語のみのご利用となります)。

Rational Suite を使用したソフトウェア開発に関する記事、ディスカッション フォーラム、Web ベースのトレーニング コースにアクセスするために、Rational Developer Network に参加することをお勧めします。

- 1 コンピュータに現在のバージョンの Rational Suite がインストールされている場合は、[スタート] ボタンをクリックし、[Rational Software] をポイントします。次に、[Logon to the Rational Developer Network] をクリックし、表示された手順に従って登録します。
- 2 コンピュータに現在のバージョンの Rational Suite がインストールされていない場合は、<http://www.rational.net> を開き、画面の指示に従って登録します。

Rational Developer Network に登録してログインしたら、この Web サイトを検索して、使用可能な目的のリソース、コミュニケーション、コラボレーション リソースについて調べます。Rational Developer Network は、Rational Suite を使用するプロジェクトを始める際の助けとなります。

用語集

英数字

Rational Administrator: Rational プロジェクトを管理したり、リポジトリを関連付けて Rational プロジェクトを定義したりするためのツール。詳細については、『Using the Rational Administrator』を参照してください。

Rational ClearCase LT: バージョン管理、ワークスペース管理、プロセス管理を含む包括的な構成管理ソリューションを提供します。

Rational ClearQuest: 高度なカスタマイズが可能な変更依頼管理ツール。このツールにより、ソフトウェア開発ライフサイクルを通じてあらゆるタイプの変更アクティビティ (障害と修正、拡張依頼、ドキュメントの変更など) を追跡できます。ClearQuest Web インターフェイスでは、ClearQuest の主要な操作を行うことができます。

Rational ClearQuest MultiSite: 高度なカスタマイズが可能な Windows または Web ベースの変更依頼管理ツール。集中管理されたデータベースのレプリカを各リモートサイトに作成し、各サイトで加えられた変更とほかのサイトで加えられた変更を同期させることにより、異なる場所に分散するユーザーがソフトウェア開発ライフサイクルを通じてあらゆるタイプの変更アクティビティ (障害と修正、拡張依頼、ドキュメントの変更など) を追跡できます。

Rational Developer Network: Web 上で動作する検索可能な知識ベース。最善の実践原則、再使用可能な成果物とアセット、Web ベース トレーニングが集められており、ソフトウェア専門家の専門技術の向上に役立ちます。Rational ユーザーは、Rational Developer Network を Rational Suite のコンポーネントとして利用できます。

Rational Process Workbench: 高度なカスタマイズが可能な、Web 上で動作する検索可能な知識ベース。チームの生産性を向上させ、重要なソフトウェア開発アクティビティに対し、ガイドライン、テンプレート、ツール メンターによって企業固有の最善の実践原則を提供します。

Rational ProjectConsole: 高度なカスタマイズが可能なプロジェクト管理ツール。最善の実践原則の選択と導入、反復プロジェクトの計画と実行、ソフトウェア開発ライフサイクルにおける進行状況と品質の測定を実現します。

Rational PureCoverage: ソフトウェア アプリケーションのコード内のテストされていない箇所を自動的にピンポイント検出します。

Rational Purify: ソフトウェア アプリケーションの検出しにくい実行時メモリエラーを、自動的にピンポイント検出します。

Rational Quantify: ソフトウェア アプリケーションのパフォーマンスにおける問題を、自動的にピンポイント検出します。

Rational RequisitePro: 開発チームが、システムまたはアプリケーションへの変更依頼を簡単かつ包括的に整理し、優先順位を決定し、追跡、管理できるようにするツール。Rational RequisitePro は、Microsoft Word とセキュリティの高いマルチユーザー データベースとの緊密な統合により、これを実現します。RequisiteWeb インターフェイスでは、RequisitePro の主要な操作を行うことができます。

Rational Robot: テスト スクリプトの記録と再生を自動化して、機能テストの実行を支援するツール。テスト ルーチンを構成、作成、実行し、テスト結果を収集、分析することができます。

Rational Rose: 先進のビジュアル コンポーネント モデリング、開発ツール。現在のビジネス ニーズを満たすソフトウェア アプリケーションをモデリングすることができます。

Rational SoDA (for Word): ソフトウェア ドキュメント自動生成ツール。異なる開発ツールのデータを統合するという難問を解決します。複数のソースからデータを抽出して、ソフトウェア、システム、プロジェクトの包括的なドキュメントを自動的に作成します。

Rational Suite: ソフトウェア開発チームを統合し、アナリスト、開発者、テスト担当者、プロジェクト マネージャーの生産性を最大限に高める、導入とサポートが簡単なソリューション。

Rational Suite AnalystStudio: システム定義用に最適化された、Rational Suite のエディション。Team Unifying Platform と Rational Rose (Professional Data Modeler Edition) で構成されます。

Rational Suite DevelopmentStudio: ソフトウェア開発用に最適化された、Rational Suite のエディション。Team Unifying Platform と、Rational Rose (Enterprise Edition)、Rational Purify、Rational Quantify、Rational PureCoverage で構成されます。

Rational Suite DevelopmentStudio - RealTime Edition: リアルタイム システムまたは組み込みシステムのシステム開発者や設計者用に最適化された、Rational Suite のエディション。Team Unifying Platform と、Rational Rose RealTime、Rational Purify、Rational Quantify、Rational PureCoverage で構成されます。

Rational Suite Enterprise: すべての Rational Suite ツールが組み込まれている、Rational Suite のエディション。

Rational Suite Team Unifying Platform: 生産性と品質を最大限に高めるために、ソフトウェア開発チームのすべてのメンバー用に最適化された、Rational Suite のエディション。Rational Unified Process、RequisitePro、ClearCase LT、ClearQuest、SoDA、TestManager、ProjectConsole で構成されています。

Rational Suite TestStudio: テスト担当者用に最適化された、Rational Suite のエディション。Team Unifying Platform と、Rational PureCoverage、Rational Purify、Rational Quantify、Rational Robot、Rational TestFactory で構成されます。

Rational TestManager: 1 つの集中管理ポイントから、すべてのテスト アクティビティを管理制御することができます。従来の独自テスト アセットを制御、表示できる機能も備わっています。テスト結果と、目標に向けての進行状況を、すべてのチーム メンバーに即座に伝達することにより、チームの生産性を向上させます。

Rational Unified Process: Web 上で動作する検索可能な知識ベース。チームの生産性を向上させ、重要なソフトウェア開発アクティビティに対し、ガイドライン、テンプレート、ツール メンターによってソフトウェアの最善の実践原則を提供します。

あ

アクター: システムまたはビジネスの外側にいて、システムまたはビジネスとやり取りする人または物。

アクティビティ: チーム メンバーが実行する作業の単位。

アナリスト: システムの用途を決定し、要求を設定、管理し、開発組織にユーザーの要求を示す担当者。

エディション: ソフトウェア開発チームの機能分野ごとにカスタマイズされた Rational Suite のツール セット。

オブジェクト: オブジェクト指向の分析と設計において、データの演算を行うためのデータとメソッド (プロシージャ) の集合を持つソフトウェア コンポーネント。

か

開発構想書: 開発するシステムについてのユーザーまたは顧客の理解に対する高レベルの見解をまとめたドキュメント。

開発者: システムの動作を決定し、アーキテクチャを定義し、コードを作成、変更、テスト、管理する担当者。

拡張ヘルプ: 各 Rational Suite 製品と、Rational Unified Process、ユーザー カスタム情報をリンクする、Rational Suite の強力な機能。

クラス: オブジェクト指向の分析と設計において、同じ責任、関係、操作、属性、セマンティクスを共有するオブジェクトセット。

構成管理: ソフトウェアを作成、変更、構築、リリースするときに、チームが毎日のソフトウェア開発作業を管理できるようにする管理機能。包括的なソフトウェア構成管理には、バージョン管理、ワークスペース管理、ビルド管理、プロセス管理が含まれ、プロジェクト管理と予測可能性を向上させます。

コンポーネント: システムの重要でほぼ独立した置換可能な部分。十分に定義されたアーキテクチャのコンテキストにおいて明確な機能を実現します。

コンポーネント ベースのアーキテクチャ: ソフトウェア システムを個々のコンポーネントに分解する設計方法。

さ

作業分野: 特定の成果物セットを生成するために実行するすべてのアクティビティの要約。

自動テスト: 繰り返しが多くエラーが発生しやすい手作業を、ソフトウェア ツールを使用して自動化するテスト技術。自動テストを使用すると、時間を節約できます。また、信頼性が高く、予測、反復可能で正確なプロセスを実現できます。

実装: 開発されたコンポーネントをユニットごとにテストし、実行可能なシステムにその結果を統合するプロセス。

ストリーム: ストリームは、UCM において設定手順をビュー (「ビュー」を参照) に表示し、アクティビティとベースライン (「ベースライン」を参照) を追跡します。

成果物: プロセスによって作成、変更、使用される情報。責任範囲を定義します。バージョン管理が必要です。成果物には、要求、モデル、モデル要素、ドキュメント、ソース コードなど、さまざまなタイプがあります。

た

追跡可能性: あるプロジェクト要素から、関連のあるほかのプロジェクト要素までを追跡できること。

ツール メンター: Rational Unified Process に記述されたアクティビティを実行する場合の特定の Rational ツールの使用方法を手順ごとに示した説明。

テスト ケース: テスト入力セット。テストと検証が可能なシステム動作、アプリケーション内のテスト範囲、各テストの結果を記述します。

テスト計画: プロジェクトにおけるテストの目的と目標に関する情報と、テストの実装方針に関する情報を含みます。

テスト構成: システムの潜在的な組織構造の一連の属性。テスト ケースに適用します。

テスト担当者: テストを作成、管理、実行する担当者。ソフトウェアがすべての要求を満たしていることを確認し、結果を通知して、修正を検証します。

テスト入力: システムを開発するために使用する成果物。テストに変化を与えるために使用されることもあります。

統一変更管理 (UCM): 要求からリリースまでのソフトウェア開発をとおして、変更を管理するための **Rational** のアプローチ。UCM では、開発ライフサイクルがカバーされ、要求、設計モデル、ドキュメント、コンポーネント、テストケース、ソースコードに対する変更の管理方法が定義されています。

統合ソフトウェア プロジェクト管理 (USPM): 要求からリリースまでのソフトウェア開発をとおして、ソフトウェアプロジェクトを管理するための **Rational** のアプローチ。USPM では、開発ライフサイクルがカバーされ、ステータス、傾向、品質、その他、プロジェクト管理と進行状況の明確化に欠かせない要素を評価するための情報収集に焦点が当てられています。

統一モデリング言語 (UML): ソフトウェア システムの仕様決定、視覚化、体系化、ドキュメント化を行うための業界標準言語。ソフトウェアの設計と設計に関するコミュニケーションを簡素化します。

は

バージョン管理: ファイルとディレクトリの改訂履歴を追跡するプロセス。

反復型開発: プロジェクトの過程において、計画と評価基準に従って異なる実行可能ファイル群をリリースするプロセス。各実行可能ファイルは、前の実行可能ファイルより強力であるか機能が増えています。反復処理のたびに、プロジェクト成功という目標に近づきます。

ビジュアル モデル: システムの構造と相互関係の視覚的な表現。

ビュー: ClearCase LT オブジェクト。1 人または複数のユーザーにソース バージョンを変更するためのワークスペースを提供します。

フィチャ クリープ: ソフトウェア開発チームによって使用される用語。開発プロセスをとおして (特に後半で)、製品の機能に計画していない変更を行うことを意味します。

フェーズ: 2 つの主要なプロジェクトマイルストーン間の期間。この期間中に、明確な目標セットが満たされ、成果物が完了し、次のフェーズに移行するかどうかについての決定が行われます。

フォワード エンジニアリング: Rational Rose のビジュアル モデルからコードを生成するプロセス。「ビジュアル モデル」を参照してください。

プロジェクト: 独特の製品またはサービスを生み出すために実施される一時的な取り組み。一時的であるのは、すべてのプロジェクトには明確な始点と終点があるからです。独特とは、製品またはサービスがほかのすべての類似製品またはサービスとは何らかの点ではっきり異なっているということです。プロジェクトは、多くの場合、組織のビジネス戦略を遂行するための重要な要素となります。プロジェクトは、人々によって実行され、有限のリソースによる制約を受け、計画、実行、管理されます。

プロジェクト リーダー: リソースの割り当て、優先順位の決定、顧客とユーザーのやり取りの調整を行い、プロジェクト チームが所定の目標を達成するよう全般的に管理する担当者。プロジェクト リーダーは、プロジェクト アクティビティやプロジェクト成果物の一貫性と品質を保証するための実践原則を決定します。

ベースライン: 整合性のある成果物バージョン セット。プロジェクトのコンポーネントの安定した構成を表します。

ま

メソッド: オブジェクト指向の分析と設計における、操作またはプロシージャの実装。

メトリクス: プロジェクト アクティビティの測定。

や

役割: ソフトウェア開発組織の枠内において、個人、またはチームとして作業する個人の活動と責任。ソフトウェア開発チームの従来の役割には、アナリスト、マネージャー、テスト担当者、管理者、プロジェクト リーダーなどがあります。

ユース ケース: システムで実行される一連のアクション。特定のアクターにとって価値のある注目すべき結果を提供します。ユース ケース仕様には、メインフロー、代用フロー、例外フローがあります。

要求: ユーザーのニーズから直接導き出されるか、契約書、規格書、仕様書、その他の公式文書に記述されている、システムの条件または機能。

要求管理: 変化するシステム要求を明確化、編成、ドキュメント化し、顧客とプロジェクト チーム間の合意を確立して維持するための体系的なアプローチ。

要素: ソフトウェア プロジェクトの成果物バージョンセットを含むオブジェクト。要素は、ファイルまたはディレクトリのいずれかです。

ら

ラウンドトリップ エンジニアリング: Rose ビジュアル モデルからコードを生成し (「フォワード エンジニアリング」を参照)、ソース コードから Rose モデル ファイルを更新する (「リバース エンジニアリング」を参照) 機能。

リアルタイム アプリケーション: 待ち時間、スループット、信頼性、可用性の要求が厳しいアプリケーションまたはシステム。一般には、人間の時間感覚と同じ速度で行われる動作に相当するものと理解されています。

リスク: プロジェクトへの悪影響 (たとえば、スケジュール、予算、技術上の問題など) の可能性。

リスク管理: プロジェクトの成功を確保する手段として、プロジェクトのリスクを自発的に特定、予測し、これに対応してリスクを軽減するための計画を作成します。

リバース エンジニアリング: ビジュアル モデルとコードが同期された状態を保つように、既存のコードから Rose のビジュアル モデルを作成または更新するプロセス。「ビジュアル モデル」を参照してください。

わ

ワークフロー: 識別可能な値を獲得するために、作業分野内の役割によって実行される一連のアクティビティ。

索引

A

AnalystStudio 21, 134

C

ClassicsCD.com

インストール 28

概要 28

実行 33

ClearCase LT 20, 49, 50, 96, 100, 133

ClearQuest 51

UCM 51

Web インターフェイス 20

ClearQuest 20, 34, 50, 96, 100, 125, 133

ClearCase LT 51

Rational プロジェクトへのデータベースの
関連付け 31

Robot 121, 122

Web インターフェイス 20

起動 125

スキーマ 126

プロジェクト ステータスの評価 126

ClearQuest MultiSite 20, 133

D

DevelopmentStudio 21, 134

RealTime Edition 134

E

Enterprise Edition

Rational Suite 134

Rose 22

G

[GUI Record] ツールバー 116

J

Java

PureCoverage 107

Quantify 108

Rose 92

M

Microsoft Project (RequisitePro) 82

Microsoft Visual Basic

PureCoverage 107

Quantify 108

Rose 92

Microsoft Visual C++

PureCoverage 107

Purify 106

Quantify 108

Rose 92

Microsoft Word

RequisitePro 59

P

Professional Data Modeler Edition

Rose 21

ProjectConsole 99, 133

PureCoverage 22, 81, 105, 107, 133

Java 107

Microsoft Visual Basic 107

Microsoft Visual C++ 107

Purify 22, 81, 105, 106, 133

Microsoft Visual C++ 106

Q

QualityArchitect 22

Quantify 22, 81, 105, 108, 133

Java 108

Microsoft Visual Basic 108

Microsoft Visual C++ 108

R

Rational Administrator 133
 Rational プロジェクトの登録 30
 Rational プロジェクトへの変更依頼データベースの
 関連付け 31
Rational ClearCase LT
 「ClearCase LT」を参照
Rational ClearQuest
 「ClearQuest」を参照
Rational Developer Network 21, 132, 133
Rational Process Workbench 25, 133
Rational PureCoverage
 「PureCoverage」を参照
Rational Purify
 「Purify」を参照
Rational Quantify
 「Quantify」を参照
Rational Robot
 「Robot」を参照
Rational Rose 64
 「Rose」を参照
Rational SoDA
 「SoDA」を参照
Rational Suite 134
 AnalystStudio 21, 134
 ContentStudio 134
 DevelopmentStudio 21, 134
 DevelopmentStudio - RealTime Edition 134
 Enterprise Edition 134
 Team Unifying Platform 19, 134
 TestStudio 23, 134
 概要表 25
 ツール 19, 27
 マニュアル構成 xii
 利点 18, 19
Rational Suite のツール 27
Rational TestFactory
 「TestFactory」を参照
Rational TestManager 134
 「TestManager」を参照
Rational Unified Process 18, 19, 37, 134
 概要 38
 拡張ヘルプ 68
 起動 37
 ツール メンター 46
 フェーズと反復 40

Rational プロジェクト 30, 51
 データストア 32
 登録 30
Rational プロジェクトの登録 30
Rational、ミッション 15
RequisitePro 55, 77, 96, 100, 133
 Microsoft Project との統合 82
 Rose 64
 Web インターフェイス 20
 Word ドキュメント 59, 61
 エクスプローラ 56
 起動 55, 82
 ツール バレット 55
 テスト計画 71
 データベース機能 61
 ドキュメント機能 61
 ビュー 61
Robot 23, 81, 107, 111, 133
 ClearQuest 121, 122
 [GUI Record] ツールバー 116
 起動 113
 スクリプトの再生、Robot 120
 テスト結果の表示 121
Rose 64, 77, 85, 92, 96, 100, 133
 Enterprise Edition 22
 Java 92
 Microsoft Visual Basic 92
 Microsoft Visual C++ 92
 Professional Data Modeler Edition 21
 RealTime 23
 RequisitePro 64
 Web 発行 90
 起動 62, 85
 クラス図 92
 図ウィンドウ 63, 85, 91
 ブラウザ 63, 85, 91
RUP
 「Rational Unified Process」を参照

S

SoDA 20, 95, 96, 134
 起動 95
 テンプレート 99
SQABasic 115

T

Team Unifying Platform 19, 134
TestFactory 24
TestManager 20, 72, 76, 81, 96, 100, 111, 112, 134
 Robot 121
 起動 111
TestStudio 23, 134

U

UCM (統一変更管理) 20, 136
UML
 「統一モデリング言語」を参照
Unified Process
 「Rational Unified Process」を参照

W

Web 開発 53
Word ドキュメント (RequisitePro) 61

あ

アーキテクチャ
 コンポーネント ベース 17
 ビジュアル モデリング 86
アクター 44, 134
アクティビティ 42, 49, 134
 ClearCase LT 52
アナリスト 21, 134
 ツール 64

い

インテグレーション ストリーム (ClearCase LT) 51
インテグレーション ビュー (ClearCase LT) 51

え

エディション 134
エラー レポート 121, 122

お

オブジェクト 134
 Rose での識別 92
親要求 66

か

開発構想書 135
 要件 58
開発者 21, 135
 ツール 64
開発ストリーム (ClearCase LT) 51
開発、ソフトウェア
 「ソフトウェア開発」を参照
開発ビュー (ClearCase LT) 51
拡張ヘルプ 68, 135
カスタマ サポート xiii
管理 96
管理、リスク 72

き

機能テスト 111, 112

く

クラス 135
 クラス図 92
 識別 (Rose) 92

け

検証ポイント 116

こ

構成
 Rational Suite のマニュアル xii
構成管理 20, 49, 135
コード、実装 92
コードの実装 92
子要求 66
コンポーネント 17, 135

コンポーネント ベースのアーキテクチャ 135
設計 (Rose) 17
コンポーネント ベースのアーキテクチャの設計 17

さ

作業分野 39, 42, 135
サスペクト リンク (RequisitePro) 67

し

シエル スクリプト 118
シーケンス図 86
オブジェクト 88
クラス 87
メッセージ 87, 88
ユース ケース 86
システム テスト
システム パフォーマンス
実装、テスト 76, 135
自動テスト 105, 135
障害レポート 121, 122

す

図ウィンドウ (Rose) 63, 85, 91
スキーマ 126
スクリプト
計画 (TestManager) 115
再生、Robot 120
シエル 118
スクリプトの計画、TestManager 115
スクリプトの再生 120
スケジュール
テスト作業 81
予測可能性 38
ストリーム、ClearCase LT 135
ストリームのリベース (UCM) 53

せ

成果物 30, 42, 49, 61, 72, 135
ClearCase LT 52
変更管理 49
設計
テスト 79

そ

測定
プロジェクト ステータス 96
ソフトウェア 18
ソフトウェア エンジニア、役割 21
ソフトウェア開発
共通の問題 15
コンポーネント ベースのアーキテクチャ 17
反復型開発 16
品質の検証 18
変更管理 18
要求管理
ソフトウェアの変更管理 18
ソフトウェア品質の検証
「テスト」を参照

ち

チュートリアル
ClassicsCD.com 28
サンプル アプリケーション 28
設定 28
前提条件 27
ツールのチェックリスト 27
マニュアル構成 35
リセット 30
チュートリアル の前提条件 27
チュートリアル のリセット 30
チュートリアル用サンプル アプリケーションの
インストール 28

つ

追跡可能性 135
リンク 67
ツール (Rational Suite) 19
ツール パレット (RequisitePro) 55
ツール メンター 46, 135

て

テスト
カバレッジ 107
機能 111, 112
検証ポイント 116
コードのパフォーマンス 22, 108

システム パフォーマンス 23, 81
信頼性 105
タイプ 81
品質の検証 18
メモリ リーク 106
ランタイム エラー 106
ルーチン 112
テスト計画 71, 73, 135
スクリプトの作成 115
スケジュール 81
テスト タイプ 81
リスクとリソースの識別 80
テスト ケース 72, 73, 75, 76, 112, 136
設計 79
フォルダ 73, 75
テスト構成 72, 135
テスト スクリプト 112
テスト担当者 23, 135
テスト入力 72, 136
データベース (RequisitePro) 61

と

統一変更管理 (UCM) 49
統一モデリング言語 17, 21, 26, 62, 86, 136
統合ソフトウェア プロジェクト管理 96
ドキュメント (RequisitePro) 61

は

バージョン管理 136
パフォーマンス テスト
システム 23, 81
パフォーマンス テスト、コード 22, 108
反復 40
反復型開発 16, 136
Rational Unified Process 40

ひ

ビジュアル モデリング 85, 86
コードとの整合性を維持 92
コードの実装 92
ビジュアル モデル 136
Web バージョン (Rose) 90
ビュー 136

ビュー (RequisitePro) 61, 66
ビルド 18
品質管理エンジニア、役割 23
品質、検証
「テスト」を参照

ふ

フィーチャ クリープ 136
フェーズ 40, 136
フォワード エンジニアリング 136
プロジェクト 136
プロジェクト ステータス 96
プロジェクト ステータス、評価 126
プロジェクトに参加 51
プロジェクト メトリクス (ProjectConsole) 99
プロジェクト リーダー 136
プロセス
「Rational Unified Process」を参照

へ

ベースライン 52, 136
プロモート 52
ヘルプ
Rational カスタマ サポート xiii
参照先 xi
ヘルプ、拡張 68, 135
変更管理 18
変更セット 52
変更の管理 18, 49

ま

マニュアル
Rational Suite のマニュアル構成 xii
参照先 xi
マニュアル構成、チュートリアル 35

め

メソッド 137
メトリクス 137
メモリ リーク 106

や

役割 40, 42, 137

ゆ

ユース ケース 44, 137
シーケンス図 86
チームでの利点 58
ビジュアル モデリング 58
レポート 95, 97
ユース ケース図 62
操作 63
ユース ケース要求 61
ユース ケース レポート 97

よ

要求 57, 137
親 66
管理 17, 55, 57, 137
子 66
タイプ 68
変更 58
要求管理
「要求」の「管理」を参照
要件
開発構想書 58
要素 136
予算と予測可能性 38

ら

ラウンドトリップ エンジニアリング 137
ランタイム エラー 106

り

リアルタイム アプリケーション 137
リスク管理 72, 137
リバース エンジニアリング 137
リンク
サスペクト、RequisitePro 67
追跡可能性、RequisitePro 67
要求と障害 127

れ

レポート、作成 (SoDa) 96

わ

ワークフロー 42, 137