

Rational Suite®

入門

バージョン: 2003.06.10

G126-5381-00

WINDOWS/UNIX

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レメント。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	ix
対象読者	ix
その他の参照先	ix
Rational Suite のマニュアル構成 - Windows 版	x
Rational Suite のマニュアル構成 - UNIX 版	xi
Rational カスタマ サポートの連絡先	xii
1 Rational Suite について	13
プロジェクトを成功に導くために	13
Rational の最善の実践原則	14
Rational Suite のツール	15
チーム統一用ツール	16
プロフェッショナル サービス	18
Rational Developer Network	18
Rational のその他のプロフェッショナル サービス	18
Rational Suite 入門	18
Rational Unified Process の概要	19
2 最適なシステムの定義: アナリスト	21
着手前の問題の理解	21
要求と反復型開発	21
Rational Suite AnalystStudio	22
要求管理	22
チーム メンバーへの情報の伝達	22
ユース ケースの使用	22
要求変更の管理	22
全チーム メンバーによる情報の共有	23
チームのコミュニケーションの視覚化	23
進捗状況の測定とプロジェクト レポートの提供	24
ユース ケースの使用	24
3 複雑さの管理: 開発者	25
分析と設計: 反復型開発の次のステップ	25
Rational Suite DevelopmentStudio	26
ビジュアル モデルによる複雑さの管理	26
使い慣れた環境でのコーディング	26
コード実装の迅速化	26

コードとモデルの整合性の維持	26
変更に伴うリスクの評価	27
要求変更の評価	27
システムに対する変更の検証	27
システムに対する変更の管理	27
統一変更管理 (UCM)	28
チームへの最新情報の提供	28
初期段階から頻繁に行うコードのテスト	29
テスト コードの生成	29
実行時エラーの検出	30
パス カバレッジの解析	30
パフォーマンスが不十分な箇所の特定	30
テスト結果の追跡	30
Rational Suite DevelopmentStudio: RealTime Edition	31
複雑なリアルタイム システムの構築	31
4 品質の評価: テスト担当者	33
品質の評価とリリースの決定	33
テスト: 反復型開発の最終ステップ	33
Rational Suite TestStudio	34
テスト計画の作成	34
アプリケーションのテスト	34
自動機能テスト	34
自動信頼性テスト	35
自動アプリケーション パフォーマンス テスト	35
自動システム パフォーマンス テスト	35
テスト範囲検証テスト	35
テスト結果の追跡	35
チームへの最新情報の提供	36
5 変更とリスクの管理: マネージャーとプロジェクト リーダー	37
プロジェクト計画の重要性	37
Rational Suite Enterprise: 反復型開発向けの完全なソリューション	37
最善のソフトウェア開発手法の実装	37
UCM を使用した変更の管理	38
変更依頼管理	38
構成管理	38
プロジェクト ステータスの管理	39

6 次のステップ	41
Rational Suite の導入	41
Rational Suite の全製品の導入	41
Rational Suite の段階的導入	42
Rational Suite のツールの使用法	42
その他の Rational ツール	43
Rational ClearCase	43
Rational XDE	43
用語集	45
索引	51

まえがき

本書では、Rational Suite[®] ツールの概要、Rational[™] による最善のソフトウェア開発手法について説明します。本書をお読みいただくことで、次のことが可能になります。

- ソフトウェアの品質向上
- 開発期間の短縮
- チーム コミュニケーションの向上

対象読者

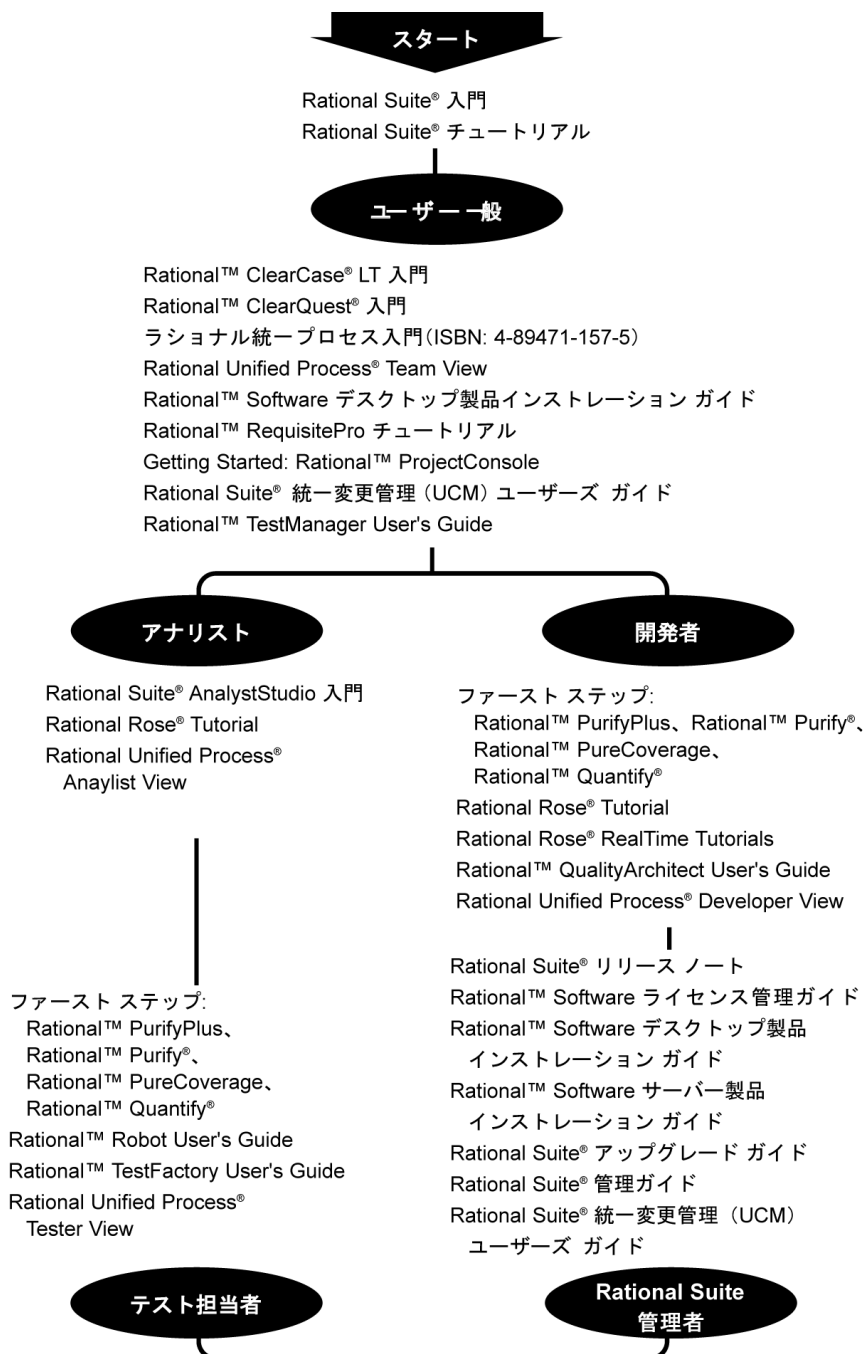
本書は、ソフトウェア開発に携わる次の方々を対象としています。

- Rational Suite を初めて使用する方
- Rational Suite を使用した統合ソフトウェア開発に興味のある方

その他の参照先

- 以下の Web サイトと CD で、技術ドキュメントを参照できます。
 - <http://www.rational.com/documentation> (ただし、英語のみのご利用となります)
 - Rational Solutions オンライン ドキュメント CD
- ラショナル ユニバーシティ : <http://www.rational.co.jp/services/ru/>
- Rational Developer NetworkSM: <http://www.rational.net>

Rational Suite のマニュアル構成 - Windows 版



Rational Suite のマニュアル構成 - UNIX 版



Rational カスタマ サポートの連絡先

本製品のインストール、使用法、保守に関するご質問については、Rational カスタマ サポートに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: お問い合わせいただく際は、次の情報をお知らせください。

- 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
- Rational 製品の名称とバージョン番号
- ケース ID 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。
- 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。実行に信頼性がありますか。
- 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。
- 開発プロセスは反復可能でしたか。

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。その結果、次のような状況に陥っています。

- プロジェクトの完了が予定より遅れて完了する (完了しない場合もあります)。
- プロジェクトの結果が、要求を満たしていない。
- 開発の末期段階で、設計上の重大な欠陥が見つかった。
- 開発中ではなく、ソフトウェアの出荷後に不具合が見つかった。

プロジェクトを成功に導くために

Rational は、お客様がこれらの問題を解決し、ソフトウェア開発プロジェクトを成功に導くことができるように、次のものを提供しています。

- 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採用と実装を促進するプロフェッショナル サービス

Rational の最善の実践原則

実践原則	プロジェクトの ライフサイクル タスク	利点
ソフトウェアの反復型開発	分析、設計、システムのサブセットのインクリメンタルな実装	<ul style="list-style-type: none"> ▪ 初期段階でのリスクの軽減 ▪ より高い品質の達成 ▪ プロジェクトアセットの再利用性の向上
要求管理	要求変更の検出、文書化、管理、追跡	<ul style="list-style-type: none"> ▪ 変更に伴うリスクの軽減 ▪ 製品に関する所定の目標の達成
コンポーネント ベースのアーキテクチャの使用	<p>システムの構造的要素の設計、それらの要素の動作設計、段階的に増大するシステムにそれらを適合させる方法の決定</p> <p>どのサブシステム コンポーネントを開発、再利用、購入するか判断</p>	<ul style="list-style-type: none"> ▪ 製品の信頼性、保守性、拡張性の向上
視覚的なモデリング	システム アーキテクチャ設計をグラフィカルに作成	<ul style="list-style-type: none"> ▪ 設計の複雑さの管理 ▪ 複雑なシステムの理解度の向上 ▪ 低コストで代替的な設計を探索、比較 ▪ 実装の基礎の形式化
継続的な品質の検証	各反復作業でのテスト	<ul style="list-style-type: none"> ▪ システムが必要な機能と信頼性を備えていることと、負荷がかかった状態で稼動する能力を備えていることの検証 ▪ 要求、設計、実装に関する問題の早期発見
変更管理	反復可能な開発プロセスの確立	<ul style="list-style-type: none"> ▪ チーム コミュニケーションの向上 ▪ リスク管理 ▪ リソース割り当ての改善 ▪ 並行開発の実現

Rational Suite のツール

Rational は、次のことを実現するツールにより、これらの最善の実践原則を提供しています。

- ・ チームを統合し、コミュニケーションを円滑にするツール
- ・ チーム メンバーの生産性を向上するツール
- ・ 共通のインストール、ライセンス、ユーザー サポート計画を簡素化するツール

Rational Suite エディションは、各チーム メンバーの役割に適合するようにカスタマイズされた一連のツールの集まりです。次の表内のハイパーリンクをクリックして、各エディションの詳細を参照してください。

		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							

チーム統一用ツール

Rational Suite Team Unifying Platform 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、品質の向上、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 Unified Process (RUP™): RUP を使用して開発プロセスの構築と実装を行うことで、チームの作業効率を向上させ、コミュニケーションを円滑化することができます。RUP は、次のように構成できます。

- 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

Suite RUP には、Windows 版と UNIX 版が用意されています。

Rational RequisitePro: データベースを使用して、より効率的に要求管理を実現するツールです。RequisitePro を使用して、次の作業を行うことができます。

- 要求の編成
- 要求の優先順位の決定
- 要求の追跡
- 要求の文書化

RequisitePro に備わっている RequisiteWeb インターフェイスを使用すれば、Web ブラウザから要求データのアクセス、作成、管理を行うことができます。RequisitePro は、Windows でのみ使用できます。RequisiteWeb は Windows と UNIX の両方で使用できます。

Rational ClearQuest: ソフトウェア開発における次のような変更アクティビティを管理するツールです。

- 拡張依頼
- 障害
- ドキュメントの修正

ClearQuest Web インターフェイスを使用すれば、Windows ユーザーと UNIX ユーザーの両方が、Web ブラウザから、レコードの登録と検索、クエリーとレポートの作成/編集、ショートカットの作成など、ClearQuest の基本的な操作を実行できます。ClearQuest MultiSite を使用すれば、チームが地理的に分散している場合でもメンバー間で情報を共有できます。ClearQuest は、Windows クライアント ワークステーションと UNIX クライアント ワークステーションで使用できます。どちらのプラットフォームで使用する場合も、Windows ワークステーションを ClearQuest リポジトリの管理用コンピュータとして設定する必要があります。

Rational ClearCase LT: ソフトウェア構成管理と、次に示すソフトウェア プロジェクト アセット 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。

- 要求
- ビジュアル モデル
- コード
- ドキュメント

ClearCase LT が備える **Web** インターフェイスでは、**ClearCase LT** の基本的な操作を実行できます。**Rational ClearCase LT** では、変更管理とワークフロー制御のための **Rational** プロセスである統一変更管理をサポートしています。**ClearCase LT** は **Windows** と **UNIX** の両方で使用できます。

Rational SoDA: ソース コードや **Rational** ツールが作成したファイルなどの開発成果物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生成するツールです。**SoDA** では、定義済みまたは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** には、**Windows** 版の **Microsoft Word** と **UNIX** 版の **Adobe FrameMaker** 用のテンプレートが用意されています。

Rational TestManager: 実環境の機能とマルチ ユーザーのテストを作成して、**Web** アプリケーション、多層アプリケーション、データベース アプリケーションのパフォーマンスの評価に使用します。**TestManager** では、次のデータを追跡できます。

- いくつかのテストを計画、開発、実行したか
- どの要求がテスト済みか
- いくつかのテストが合格または不合格だったか

TestManager ではこのデータを使用して、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通知するレポートを作成できます。**TestManager** は、**Windows** クライアント ワークステーションと **UNIX** エージェントで使用できます。ただし **UNIX** で使用する場合は、**Windows** ワークステーションを **TestManager** リポジトリの管理用コンピュータに設定する必要があります。

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** と統合されており、プロジェクト計画を一元的に作成、表示、レポートすることができます。また **ProjectConsole** では、単一の **Web** サイト上でプロジェクト成果物が集中管理されるため、すべてのチーム メンバーが成果物の情報を参照できます。**ProjectConsole** は、**Windows** でのみ使用できます。

プロフェッショナル サービス

Rational Developer Network

Rational Developer Network (RDN) は、Rational 製品のすべてのユーザーにご利用いただけるオンライン コミュニティです。RDN で提供するコンテンツは、次のとおりです。

- ほかのソフトウェア技術者とアイデアや実践的ノウハウを交換するためのフォーラム
- Web ベースのトレーニング
- 記事、ホワイト ペーパー、各種入門コースなどのコンテンツ

RDN に参加するには、<http://www.rational.net> にアクセスしてください (英語のみのご利用となります)。

Rational のその他のプロフェッショナル サービス

Rational では、次のサービスも提供しています。

- コンサルティングと指導
- Quickstarts、導入サービス、ワークショップなどのパッケージ サービス
- 個別にカスタマイズしたサービスとプロジェクト

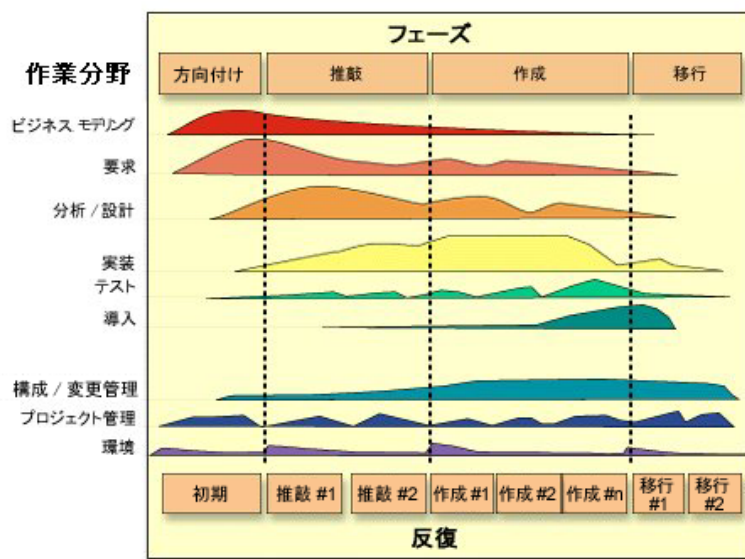
詳細については、http://www.rational.co.jp/services/pro_sv/index.html を参照してください。

Rational Suite 入門

高品質なソフトウェアを継続して配布するには、密接なチームワークと開発作業に関する共通の理解が必要です。このため、RUP のような予測可能で、繰り返し可能なプロセスの実装が非常に重要となります。

RUP の機能	RUP の利点
プロセスとツールのライブラリ。	個々のプロジェクトに最適なプロセスを選択、導入できます。
プロセス構成設計用のカスタマイズ可能な Web インターフェイス。	組織内のプロセス標準を確立できます。
Rational ツールの使用法を参照できるツール メンター。すべての Rational ツールから RUP 内の関連トピックを参照可能な拡張ヘルプ (Windows のみ)。拡張ヘルプには、独自のコンテンツも追加できます。	個々のプロセスで Rational ツールをどのように使用すればよいかを参照できます。

Rational Unified Process の概要



この図は、Rational が収集した実際のデータに基づいて作成したもので、ソフトウェア開発工程は次の 4 つの「フェーズ」に分類できることを示しています。

- **方向づけ:** プロジェクトの範囲を定義します。
- **推敲:** プロジェクトを計画し、機能とアーキテクチャのベースラインを指定します。
- **作成:** 製品をビルドし、テストします。
- **移行:** 製品をユーザーに提供します。

各フェーズは、一連の「反復」内で実現されます。各反復は、各「作業分野」の作業を一通り完全に遂行することを表します。開発中、各作業分野を担当するチーム メンバーは、1 つまたは複数の「ワークフロー」内でアクティビティを実行します。

図は、個々のワークフローにおける重点が反復ごとに変化することを示しています。たとえば、初期段階の反復では要求に多くの時間が割かれており、末期段階では実装に多くの時間が割かれています。

1 回の反復プロセスにおける作業手順は、アナリスト、開発者、テスト担当者、プロジェクトリーダーなど、チームの個々の役割に関連して決められています。各役割の担当者は、各反復で、できるだけ早くこれらのアクティビティを実行します。たとえば、要求分析は反復の開始時に集中的に行われますが、アナリストは各反復とプロジェクトのライフサイクルをとおして要求の追跡を継続的に行います。

最適なシステムの定義： アナリスト

2

ユーザーの要求に対処し、ユーザーの抱える問題を解決していますか。

最終段階での機能要求が原因で、リリースが遅れたことはありませんか。

予算もスケジュールも予定どおりにいきましたか。

ユーザーが製品に不満をもったり失望したりしていませんか。

着手前の問題の理解

最適なシステムを明確に定義するために、アナリストは利害関係者と共に次の作業を行います。

- 問題の分析
- 利害関係者の要求の理解

アナリストは、収集した情報を分析し、要求として明確化した上で、次の作業を行います。

- システムの定義
- システム範囲の管理

要求と反復型開発

要求は、開発中に変わっていきます。たとえば次のような場合です。

- 競合製品の新しいバージョンがリリースされた場合、それに対抗するために、プロジェクトへの新機能の追加が必要になります。
- 最初の時点でユーザーの要求が部分的に不明確だったり未決定だったりした場合、要求の追加、削除、明確化が必要になります。
- 開発を開始した後に関連技術に変化が生じた場合、製品の品質や開発スケジュールに影響を及ぼさずに、その技術を組み込むかどうかを判断する必要があります。
- 要求の実現に費用がかかりすぎたり、スケジュール内で要求を実現できない場合、要求の削除やスケジュールの延長が必要になります。

要求の管理では、要求の変更がほかのチームメンバーの作業に及ぼす影響を把握することが重要です。アナリストは、要求の変更、要求の優先順位、要求の変更による開発スケジュールへの影響について、チームメンバー全員に伝達する必要があります。

Rational Suite AnalystStudio

Rational Suite AnalystStudio は、要求の管理を最も効果的に行うために必要な、すべてのツールを備えています。

要求管理

アナリストは Rational RequisitePro を使用して、すべての利害関係者がアクセスできるフォームで要求の収集と管理を行い、明確にすることによって、チーム内のコミュニケーションを円滑にすることができます。RequisitePro では、広く使用されている Microsoft Word とリレーショナルデータベースの機能を組み合わせています。

- アナリストは、Microsoft Word を使用してプロジェクト要求を文書化し、製品の開発構想、特徴、機能、属性などの情報をメンバー間でやり取りすることができます。
- またアナリストは、RequisitePro データベースにこの情報を格納し、情報を随時確認します。このデータベースでは、組織、優先順位決定、変更履歴の要求に属性を割り当てることができます。これらの属性は、関連する要求まで追跡できるため、開発を進めるときに要求の変更による影響を迅速に評価できます。

チーム メンバーへの情報の伝達

すべてのチーム メンバーが共通のプロジェクトの目標と目的を把握し共有する必要があるため、アナリストはチーム全体に最新の要求の動きを常に知らせる必要があります。Rational RequisiteWeb を使用すると、UNIX ユーザーなどの RequisitePro にアクセスできないチーム メンバーも、要求を作成、確認、更新できます。

ユース ケースの使用

RequisitePro を使用して、アナリストは、要求と共に、システムの目的の機能と環境を表した関連する「ユース ケース」を管理できます。ユース ケースはアナリストによって作成され、次の作業に使用されます。

- 開発者によるシステム仕様の記述とコーディング
- テスト担当者によるテスト計画の立案とテスト スクリプトの作成

要求変更の管理

Rational ClearCase LT は、小規模なプロジェクト チーム向けの構成管理ツールです。アナリストは、このツールを使用して要求の変更を管理します。ClearCase LT では、要求を次のものと関連付けることができます。

- 個々のリリース
- コード

- ビジュアル モデル
- テスト スクリプト

ClearCase LT ではすべてのプロジェクト ファイルの変更を追跡できるため、チーム メンバーが並行して作業を行い、変更内容をプロジェクトのベースラインに統合できます。

Rational ClearQuest は、障害、新機能、製品の機能拡張などの変更依頼の追跡に使用する、変更管理ツールです。以下のような特長があります。

- 変更依頼の評価
- 変更依頼によるシステムへの影響
- 変更の検証 (該当する場合)

変更依頼を、機能と主な要求の構造に組み込む方法を確立するために、Rational RequisitePro 上の既存または新規プロジェクト要求と依頼を関連付けることができます。

全チーム メンバーによる情報の共有

Rational RequisiteWeb を使用すると、UNIX ユーザーなどの RequisitePro にアクセスできないチーム メンバーも、要求を作成、確認、更新できます。

Rational ClearQuest Web インターフェイスを使用すると、UNIX ユーザーなどの ClearQuest にアクセスできないチーム メンバーも、ClearQuest での作業に参加できます。

ClearQuest MultiSite を使用すると、地理的に分散したチームが各リモート サイトで中央のデータベースのレプリカを作成し、サイト間で変更を同期することができます。

チームのコミュニケーションの視覚化

Rational Rose (Professional Data Modeler Edition) を使用して、ビジュアル モデルを作成できます。ビジュアル モデルは次の作業に使用します。

- システム アーキテクチャの構造と動作の文書化
- 複雑なシステムの管理と把握
- 変更の影響の伝達

アナリストは、システムを高レベルで記述したユース ケース モデルを作成します。ユース ケース モデルは次の作業に使用します。

- 開発者によるシステム設計作業
- テスト担当者によるテスト作業計画の立案

Rose モデルは業界標準のソフトウェア設計用言語である UML で記述されているため、チーム メンバー全員が簡単に共有し修正できます。

進捗状況の測定とプロジェクト レポートの提供

Rational ProjectConsole は、Rational 製またはサードパーティ製の開発ツールからデータを収集し保管します。収集した情報は開発チームの共通 Web サイト上にレポートとメトリクスとして表示されるため、すべてのチーム メンバーが最新の情報にアクセスできます。ProjectConsole は、チーム メンバーが各自の担当作業の遂行に必要な情報だけを素早く入手できるように設定することができます。

Rational SoDA は、RequisitePro や ClearQuest などの Rational 製ツールから抽出したデータに基づいてプロジェクト レポートを作成します。SoDA を使用することで、より効率的で一貫性のあるチーム コミュニケーションを実現できます。SoDA には、Windows 版の Microsoft Word と UNIX 版の Adobe FrameMaker 用のテンプレートが用意されています。これらのテンプレートをカスタマイズして使用したり、ProjectConsole Template Builder を使用して新しいテンプレートを作成したりすることができます。

アナリストは、ProjectConsole と SoDA を使用して次の作業を行うことができます。

- プロジェクトのステータスと進捗の、監視と伝達
- 要求の実装の進捗確認
- 要求のテストの進捗確認

ユース ケースの使用

アナリストが作成したユース ケースに基づいて、テスト担当者は Rational TestManager を使用し、テスト計画、テスト ケース、手動テストと自動テスト、テスト結果などからテスト アセットを構築します。TestManager ではまた、テスト計画、設計、開発、実行、分析を管理し、テスト結果をチーム メンバーと共有することができます。TestManager は、すべてのチーム メンバーがテスト アセットを利用できるように、テスト アセットを一元的に保管します。

修正するには手遅れの、設計上の重大な欠陥はありませんか。

プロジェクト モジュールは適切に統合されていますか。

システム アーキテクチャの一貫性は維持されていますか。

システムのデザインは簡単に拡張できますか。

プロジェクト コンポーネントは再利用できますか。

チームは何を作成しているかを理解していますか。

チームは、どのように変更内容を相互にやり取りしていますか。

分析と設計：反復型開発の次のステップ

ソフトウェア開発者は要求に基づいて「システムがどのように動作すべきか」を定義し、その概念をシステム設計として具体化します。分析と設計では、開発者は次の作業を行います。

- アーキテクチャの改良
- アーキテクチャ候補の定義
- アーキテクチャ統合の実施
- 動作の分析
- コンポーネントの設計
- データベースの設計

開発者はこれらの作業を行い、要求の変更に応じた修正が可能な、フレキシブルでスケーラブルなシステムを設計します。この適応性という概念に基づいて、開発者はシステムのコードを作成、修正、管理します。

Rational Suite DevelopmentStudio

Rational Suite DevelopmentStudio (Windows 版 と UNIX 版) は、最適な製品の設計と構築を行い、システム アーキテクチャ設計後のパフォーマンス テストを行うために必要な、すべてのツールを備えています。

ビジュアル モデルによる複雑さの管理

Rational Rose (Windows の場合は Enterprise Edition、UNIX/Linux の場合は UNIX Edition) を使用してビジュアル モデルを作成できます。ビジュアル モデルは次の作業に使用します。

- システム アーキテクチャの構造と動作の文書化
- 複雑なシステムの管理と把握
- 変更の影響の伝達

Rose とビジュアル モデルを使用して、開発者は次の作業を行うことができます。

- 使い慣れた環境でのコーディング
- コード実装の迅速化
- コードとモデルの整合性の維持
- 変更に伴うリスクの評価

使い慣れた環境でのコーディング

Rose は、Visual Basic、Visual C++、ANSI C++、Java など、数多くの言語をサポートしています。また、IBM Visual Age for Java、Visual Studio、HP Workbench、Sun Workshop など、数多くの IDE をサポートしています。

コード実装の迅速化

開発者は、Rose を使用して、既存のモデルからコード フレームワークを自動生成し、アプリケーションを「フォワード エンジニアリング」することができます。

コードとモデルの整合性の維持

開発者は Rose を使用して、プロジェクトをとおしてアプリケーションの整合性が保たれるように、変更したコードを自動的にモデルと同期させ、アプリケーションを「リバース エンジニアリング」することができます。

変更に伴うリスクの評価

Rose によるアプリケーションのリバース エンジニアリングは、変更がシステム アーキテクチャに及ぼす影響を把握するためにも役立ちます。変更がプロジェクト標準や要求に反していたり、アーキテクチャ上の決定事項に影響したりしないかどうかを評価できます。

要求変更の評価

すべてのチーム メンバーが共通のプロジェクトの目標と目的を把握し共有する必要があるため、アナリストはチーム全体に最新の要求の動きを常に知らせる必要があります。Rational RequisitePro では、すべての利害関係者がアクセスできるフォームで最新の要求データが提供されます。Rational RequisiteWeb を使用すると、UNIX ユーザーなどの RequisitePro にアクセスできないチーム メンバーも、要求を作成、確認、更新できます。

開発者は、RequisitePro と RequisiteWeb を使用して次の作業を行うことができます。

- 要求変更の確認
- 要求変更によるシステムへの影響の把握
- 要求変更の Rose モデルへの組み込み

システムに対する変更の検証

Rational ClearQuest は、障害、新機能、機能拡張などの変更依頼の追跡に使用する、変更管理ツールです。Rational ClearQuest の Web インターフェイスを使用すると、UNIX ユーザーなどの ClearQuest にアクセスできないチーム メンバーも、変更依頼データの作成、確認、フィードバックを行うことができます。

開発者は、ClearQuest と Rose を使用して、変更依頼によるシステム アーキテクチャへの影響を、視覚化し、評価することができます。

メモ: ClearQuest は、Windows ワークステーションが ClearQuest リポジトリの管理用コンピュータとして設定されている場合のみ、UNIX クライアント ワークステーションで使用できません。

システムに対する変更の管理

Rational ClearCase LT を使用して、開発者は次の対象物への変更を管理できます。

- Rose モデル ファイル
- ソース コード
- バイナリ
- 実行可能ファイル
- ドキュメント

- ライブラリ
- ユーザー定義オブジェクト

また ClearCase LT では、並行開発をサポートする開発体系と開発プロセスにより、ビルドサイクルとリリース サイクルを迅速化できます。

統一変更管理 (UCM)

統一変更管理 (UCM) は、ClearCase LT と ClearQuest における次の作業を支援する、ビルトインの Rational プロセスです。

- 要求、設計モデル、ドキュメント、コンポーネント、ソース コードに対する変更の管理方法の定義
- ソフトウェア開発の計画や追跡に関するアクティビティと、開発中に変更される成果物とのリンク

UCM の例

- 1 プロジェクト リーダーが、ClearQuest を使用してアクティビティ (変更依頼) を開発者に割り当てます。
- 2 ClearCase LT と UCM は、そのアクティビティに対する変更セット (変更された成果物のリスト) を管理します。
- 3 開発者はそのアクティビティを使用して作業し、アクティビティに関連付けられた成果物の作成や変更を行います。
- 4 次に、作業結果をインテグレーション ストリーム (プロジェクト規模の共有ワークスペース) にデリバーします。
- 5 このストリーム内で、チーム メンバーは最新バージョンのプロジェクト共有要素の作成とテストを行うことができます。
- 6 この後、プロジェクト リーダーが新しいベースラインを作成する場合があります。ベースラインは、プロジェクトのコンポーネントの安定した構成を表し、将来の作業の基準となります。

メモ: UCM は、Windows 版の ClearCase LT でのみ使用可能です。

チームへの最新情報の提供

Rational ProjectConsole は、Rational 製またはサードパーティ製の開発ツールからデータを収集し保管します。収集した情報は開発チームの共通 Web サイト上にレポートとメトリクスとして表示されるため、すべてのチーム メンバーが最新の情報にアクセスできます。ProjectConsole は、チーム メンバーが各自の担当作業の遂行に必要な情報だけを素早く入手できるように設定することができます。

Rational SoDA は、Rational 製ツールから抽出したデータに基づいてプロジェクト レポートを作成します。SoDA を使用することで、より効率的で整合性のあるチーム コミュニケーションを実現できます。SoDA には、Windows 版の Microsoft Word と UNIX 版の Adobe FrameMaker 用のカスタマイズ可能なテンプレートがあらかじめ用意されています。

ProjectConsole と SoDA を使用して、開発者は次の作業を行うことができます。

- プロジェクトのステータスと進捗の、監視と伝達
- コンポーネントの開発の進捗確認
- コンポーネントのテストの進捗確認
- コンポーネントの品質評価

初期段階から頻繁に行うコードのテスト

開発者は、コードの実装後、直ちにテストを実施する必要があります。Rational Suite では、アーキテクチャの決定後や実行可能プログラムの作成後に直ちに使用できるテスト ツールが提供されています。このテスト ツールを使用して次の作業を行うことで、アプリケーションの信頼性とパフォーマンスを向上させることができます。

- テスト コードの生成
- 実行時エラーの検出
- コード内のパス カバレッジの解析
- パフォーマンスが不十分な箇所の特定制

テスト コードの生成

Rational QualityArchitect は、Rational Rose (Enterprise Edition) の 1 つの機能です。開発者は、Rational QualityArchitect を使用してビジュアル モデルからテスト コードを生成し、次の作業を行うことができます。

- コンポーネントのテストを自動生成し、アプリケーションの完成前にスタブやドライバを構築することができます。
- 設計を進める前に、潜在的なシステム アーキテクチャが機能要求や性能要求を満たしているかどうかを判断できます。

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。

メモ: Rational QualityArchitect は、Windows 版の Rational Suite DevelopmentStudio でのみ使用可能です。

実行時エラーの検出

Rational Purify は、実行時エラーやメモリアクセス エラーをチェックします。これらのエラーはしばしば、偶発的に表面化するまで未検出のまま残されます。Rational Purify は、C/C++ コードのチェック (Windows 版と UNIX 版)、Java コードと .NET 管理コードにおけるガーベッジ収集関連のメモリ管理上の問題のチェック (Windows 版のみ) を行います。

パス カバレッジの解析

Rational PureCoverage が生成するレポートを使用して、開発者は、コード内のどの行がテスト済みかを確認できます。

パフォーマンスが不十分な箇所の特定

Rational Quantify を使用して次の作業を行うことで、開発者は、より高品質なソフトウェアを構築できます。

- コードの実行パフォーマンスが不十分な箇所の特定
- 十分なパフォーマンスが得られない原因の解明

テスト結果の追跡

Rational TestManager では、テスト計画、設計、開発、実行、分析を管理し、テスト結果をチームメンバーと共有することができます。開発者は、TestManager を使用して、Rose モデルのどの要素がテスト済みかと、どのテストに合格または不合格だったかを確認できます。

TestManager には、Windows 版と UNIX 版が用意されています。TestManager は、Windows クライアントと Windows エージェント上で動作します。UNIX 環境で使用する場合、TestManager のホストとなる Windows サーバーと、テストやデータ収集などのアクティビティを実行するテスト エージェントが動作する UNIX コンピュータが必要になります。詳細については、『Rational TestManager User's Guide』を参照してください。

Rational ClearQuest は Rational テスト ツールと統合されており、障害情報を簡単なプロセスで入力できるようになっています。

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のツール群です。RealTime Edition には、DevelopmentStudio のすべてのツールが含まれています。ただし、Rational Rose の代わりに Rational Rose RealTime が含まれています。

複雑なリアルタイム システムの構築

Rational Rose RealTime は、複雑なリアルタイム システムの構築用に最適化されています。

- リアルタイム システム専用の UML プロファイルを使用して、リアルタイムの組み込みシステムのすべてについて詳細な構造と動作を表現できます。
- 並行性を部分的に、または完全に管理することができます。
- ホストまたはターゲット プラットフォーム上でモデルを監視、実行、デバッグすることができます。
- UML モデルから完全な C、C++、または Java アプリケーションを生成できます。
- 定義済みの複数のリアルタイム オペレーティング システムをサポートします。

前回のバージョンの製品をリリースした後に、重大なバグはいくつ発見されましたか。

システムでは、増加するユーザー負荷への対応が考慮されていますか。

ソフトウェアの品質をリリース前に正確に把握していますか。

包括的で完全なテストを実施していますか。

品質の評価とリリースの決定

テスト担当者は、テストを計画、設計、作成、実施し、ソフトウェアがリリース可能かどうかを判断するためのデータを作成します。全般的に、テスト担当者は次の作業を行います。

- ビルドの安定性の検証
- テスト アプローチの確認
- テスト アセットの改善
- 評価作業の定義
- テストと評価
- 適切な担当作業の遂行

テスト：反復型開発の最終ステップ

通常、テスト担当者は、インクリメンタルにテストを実施します。つまり、各反復作業ごとに実行可能リリースの統合とテストを行います。この方法で、品質の確認作業が全開発工程にわたって行われます。

テスト担当者は、次のように反復的にテストを実施します。

- ユース ケースと要求を追跡し、その情報に基づいてテスト計画を作成します。
- テスト計画に従って、自動テストと手動テストを作成します。
- テストを実施し、信頼性、機能、パフォーマンスに関する問題を特定します。
- テスト結果を分析し、品質を確認します。

Rational Suite TestStudio

Rational Suite TestStudio は、ソフトウェアの信頼性、機能、パフォーマンスを評価するために必要な、すべてのツールを備えています。

テスト計画の作成

テスト担当者は、Rational TestManager を使用して、ほかの Rational ツールから提供されるプロジェクト アセットに基づいて、テストを計画、設計することができます。たとえば、TestManager を使用して次の作業を行うことができます。

- テスト計画の作成、テストの進捗状況の確認。
- テスト入力の作成。テスト入力、RequisitePro の要求、Rose モデル、ソース コードを使用して作成します。
- テスト ケースの作成。テスト ケースはテスト入力を使用して作成します。テスト ケースは、テストと検証が可能なシステム動作、アプリケーションのテスト範囲を表します。
- テストの更新。テスト入力に変更されたときは、アプリケーションが完全にテストされるようにテストを更新する必要があります。
- Rational Manual Tests、Perl スクリプト、バッチ ファイル、Rational Robot スクリプトなど、さまざまなタイプのテストの管理と実行。

TestManager には、Windows 版と UNIX 版が用意されています。TestManager は、Windows クライアント上と Windows エージェント上で動作します。UNIX 環境で使用する場合、TestManager のホストとなる Windows サーバーと、テストやデータ収集などのアクティビティを実行するテスト エージェントが動作する UNIX コンピュータが必要になります。詳細については、『Rational TestManager User's Guide』を参照してください。

アプリケーションのテスト

自動機能テスト

Rational Robot により、ユーザーのシナリオに対する応答をテストして、システムが要求に合致しているかどうかを判断します。Robot を使用して、次の作業を行うことができます。

- テストの記録
- 想定動作を監視する検証ポイントの挿入
- テストの再生。テストは、必要に応じて何回でも再生できます
- テストの実行
- テスト結果の参照。実行したテスト、発生したエラーのタイプ、エラーの発生箇所、エラーの発生した検証ポイントなど、テスト時のエラーの詳細を参照できます

自動信頼性テスト

Rational Purify は、実行時エラーやメモリアクセス エラーをチェックします。これらのエラーはしばしば、偶発的に表面化するまで未検出のまま残されます。Rational Purify は、C/C++ コードのチェック (Windows 版と UNIX 版)、Java コードと .NET 管理コードにおけるガーベッジ収集関連のメモリ管理上の問題のチェック (Windows 版のみ) を行います。

自動アプリケーション パフォーマンス テスト

テスト担当者は、Rational Quantify を使用して次の作業を行い、ソフトウェアの品質を評価します。

- コードの実行パフォーマンスが不十分な箇所の特定
- パフォーマンス不足の原因の解明。原因を解明することで、開発者がシステム パフォーマンスを改善できるようにします

自動システム パフォーマンス テスト

Rational TestManager を使用すれば、数千ユーザーからのアクセスがあるようなシステムの状況もシミュレートできます。この機能により、Web アプリケーション、多層アプリケーション、データベース アプリケーションのシステム パフォーマンスの測定と予測を行うことができます。

テスト範囲検証テスト

Rational Robot と Rational PureCoverage を使用して、システム内の未テストの部分を確認できます。これにより、次のことを実現できます。

- より包括的なテストの実施
- 品質の向上
- リスクの軽減

テスト結果の追跡

Rational ClearQuest は Rational テスト ツールと統合されています。このため、テスト担当者が簡単に障害情報を入力し、それをテストの説明と結果に関連付けることができます。

Rational TestManager では、テスト結果をチーム メンバーと共有できます。

- アナリストは、どの要求がテスト済みか、どのテストに合格または不合格だったかを確認できます。
- 開発者は、Rose モデルのどの要素がテスト済みか、どのテストに合格または不合格だったかを確認できます。
- マネージャーとチーム メンバーは、テストの進捗を確認し、テスト結果を分析して品質を評価できます。

チームへの最新情報の提供

Rational ProjectConsole は、Rational 製またはサードパーティ製の開発ツールからデータを収集し保管します。収集した情報は開発チームの共通 Web サイト上にレポートとメトリクスとして表示されるため、すべてのチーム メンバーが最新の情報にアクセスできます。ProjectConsole は、チーム メンバーが各自の担当作業の遂行に必要な情報だけを素早く入手できるように設定することができます。

Rational SoDA は、Rational RequisitePro や ClearQuest などの Rational 製ツールから抽出したデータに基づいてプロジェクト レポートを作成します。SoDA を使用することで、より効率的で整合性のあるチーム コミュニケーションを実現できます。SoDA には、Windows 版の Microsoft Word と UNIX 版の Adobe FrameMaker 用のテンプレートが用意されています。これらのテンプレートをカスタマイズして使用したり、ProjectConsole Template Builder を使用して新しいテンプレートを作成したりすることができます。

テスト担当者は、ProjectConsole と SoDA を使用して次の作業を行うことができます。

- 要求の確認
- システム設計とコンポーネントの把握
- プロジェクト ステータスの監視
- テストの進捗状況の伝達
- テスト結果の評価

変更とリスクの管理： マネージャーと プロジェクト リーダー

5

ソフトウェアの変更を管理していますか。変更は偶発的に発生していませんか。

ユーザーの要求をどの程度満たしていますか。

プロジェクトはスケジュールどおりに進行していますか。また、予算を超えていませんか。

プロジェクト計画の重要性

マネージャーとプロジェクト リーダーは、プロジェクト開始時に、プロジェクト計画を作成し、スケジュール、リソース、作業分担、予算を取り決めます。

また、マネージャーとプロジェクト リーダーは、開発工程を効率的に管理するために、次の作業を行います。

- プロジェクトの各フェーズの終了
- 反復作業の管理と計画
- 新しいプロジェクトの立案
- プロジェクト開発範囲とリスクの評価

Rational Suite Enterprise: 反復型開発向けの完全なソリューション

Rational Suite Enterprise は、チームの各役割とプロジェクトの各フェーズをサポートするために必要なすべてのプロジェクト管理ツールとソフトウェア開発ツールを備えています。

最善のソフトウェア開発手法の実装

プロジェクト リーダーとマネージャーは、Rational Suite Team Unifying Platform の各種ツールを使用して、開発チームが最善のソフトウェア開発手法を利用できるようにします。たとえば、次のツールがあります。

- RUP: ソフトウェアの反復型開発を支援するプロセスを構築、実装します。
- RequisitePro: 要求を管理します。
- Rose: コンポーネントベースのアーキテクチャとモデルをビジュアルに構築します。
- TestManager: 継続的に品質を検証します。
- ClearCase LT、ClearQuest、UCM: 変更を管理します。

UCM を使用した変更の管理

プロジェクト リーダーは、UCM を使用して、関連するプロジェクト アセットの一貫性を維持しながら、最初の変更依頼から最終的なソフトウェア リリースまでのアクティビティを簡単に追跡できます。UCM には次の機能があります。

- 変更依頼管理
- 構成管理

変更依頼管理

Rational ClearQuest は、障害、新機能、機能拡張、変更依頼を追跡します。プロジェクト リーダーは、ClearQuest を使用してチーム メンバーと情報をやり取りし、変更依頼による影響を把握することができます。また、ClearQuest が生成するグラフとレポートを使用して次の作業を行うことができます。

- 変更依頼の傾向分析
- チーム メンバーの作業負荷の評価
- 次の反復作業の計画作成

構成管理

Rational ClearCase LT と UCM を使用して、プロジェクト リーダーは、開発サイクルの迅速化、リリース精度の確保、効果的な分散開発プロセスの構築を行うことができます。ClearCase LT では、バージョン、リリース、複数製品の並行開発を管理できます。たとえば次のような場合です。

- ソフトウェア ファイルとディレクトリはバージョン化されたコンポーネントで構成されます。
- プロジェクト リーダーはプロジェクトを作成し、そのコンポーネントの作業にプロジェクト チームを割り当てます。
- チーム メンバーは割り当てられたアクティビティ (タスク、障害、変更依頼) に基づいて変更を行います。
- 新しいファイルとディレクトリ バージョンは、開発中に収集され、アクティビティと関連付けられます。
- 関連付けが完了すると、アクティビティとその変更された成果物は配布され、共有のプロジェクト統合領域に統合されます。
- 新しいコンポーネント ベースラインの作成、テスト、昇格を行います。
- 複数のコンポーネントを組み合わせ、新しいアプリケーション反復を作成します。

プロジェクトステータスの管理

統合ソフトウェア プロジェクト管理 (USPM) は、要求からリリースまでソフトウェア プロジェクトの全工程を管理する Rational のアプローチです。USPM では、プロジェクトに関する次の情報の収集に焦点を当てています。

- ステータス
- 傾向
- 品質

Rational ProjectConsole は、Rational 製またはサードパーティ製の開発ツールからデータを収集し保管します。収集した情報は開発チームの共通 Web サイト上にレポートとメトリクスとして表示されるため、すべてのチーム メンバーが最新の情報にアクセスできます。

ProjectConsole を使用して、プロジェクト リーダーとマネージャーは次の作業を行うことができます。

- 目標とするメトリクスの集約。これによって、より正確にプロジェクトを評価できるようになります。
- 問題やスケジュールの遅れを事前に予測し、対処方法を決定、通知します。
- 特別な注意が必要な作業や、スケジュールの遅れを避けるためにリソースの割り当てが必要な作業を、事前に予測します。

Rational SoDA は、Rational RequisitePro や ClearQuest などの Rational 製ツールから抽出したデータに基づいてプロジェクト レポートを作成します。SoDA を使用することで、より効率的で整合性のあるチーム コミュニケーションを実現できます。

SoDA が生成するレポートを使用して、プロジェクト リーダーとマネージャーは次の作業を行うことができます。

- プロジェクト要求をどの程度満たしているかを評価します。
- 次の反復作業で実装する要求と拡張を選択します。
- 次の反復作業で修正する障害を決定します。

Rational Suite の導入

Rational Suite を開発チームに導入する際は、すべてのツールを最初から使い始めることも、一部のツールから段階的に使い始めることもできます。Suite を効果的に導入するためには、次の点を考慮する必要があります。

- ハードウェアとソフトウェアの環境
- Rational Suite のインストール要件
- トレーニングを含む効果的な導入計画と導入から一定期間を経過した後の評価

Rational Suite の全製品の導入

次期プロジェクトで Rational Suite のすべてのツールを使用することを予定している場合は、RUP と Rational Developer Network をご活用ください。

RUP を使用すれば、最善のソフトウェア開発手法をチームに導入できます。導入できる開発手法には、次のものがあります。

- ツール メンターを使用して、Rational ツールの使用法を参照できます。
- 拡張ヘルプ (Windows のみ) では、各種ツールと RUP のガイドラインがリンクされています。拡張ヘルプには、独自の内容を追加することもできます。

RUP の 1 コンポーネントである Rational Process Workbench では、組織内への RUP の導入に関する詳細な説明と、RUP の内容を追加/修正するツールが提供されています。Rational Process Workbench は、Rational Developer Network から入手できます。

また、Rational Developer Network を活用して、プロジェクトの作業効率を向上したり、各種の専門的スキルを身に付けたりすることができます。Rational Developer Network では、次のサービスを提供しています。

- 関連コンテンツ、実践的ノウハウ、業界内のエキスパートによる解説
- Web ベースのトレーニング、サードパーティによるさまざまなトレーニング コースの一覧
- 技術エキスパートが主催するディスカッション

Rational Suite の導入に関して Rational のプロフェッショナル サービスをご利用いただく場合は、18 ページを参照してください。

Rational Suite の段階的導入

Rational Suite のツールを段階的に導入する場合でも、チームの作業効率を向上させることができます。たとえば、要求の管理に焦点を置く場合があります。アナリストとプロジェクトリーダーは RequisitePro と Rose の使用法を参照し、要求分析に関する Rational Unified Process のガイドラインを開発プロセスに導入します。

次期プロジェクトの開始前に、次の作業を行ってください。

- 1 開発プロセスを再評価します。
- 2 問題の優先順位を決定します。
- 3 最初に対処する問題を決定します。
- 4 問題に対処するために最も適切な Rational ツールを選択します。
- 5 RUP、Rational Developer Network、Rational プロフェッショナル サービスを活用して、Suite のツールの使用法を習得します。

Rational Suite のツールの使用法

『Rational Suite チュートリアル』(Windows 版のみ)では、Rational Suite を使用してソフトウェアアプリケーションを計画、設計、実装、テストする際の基本的な使用法を説明しています。

『Rational Suite チュートリアル』は、次の方法で参照できます。

- 1 Rational Suite が既にインストールされている場合は、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、[Rational Suite チュートリアル] をクリックします。表示される説明に従って、チュートリアル アプリケーションのファイルと関連ドキュメントをインストールします。
- 2 Rational Suite がまだインストールされていない場合は、<http://www.rational.com/documentation> (ただし、英語版のみのご利用となります) にアクセスします。[Rational Suite Tutorial] のリンクをクリックします。必要に応じて Rational Member Profile のユーザー登録を行い、ログオンします。その後、表示される説明に従って、ドキュメントをダウンロードします。

メモ: チュートリアルのエクササイズを実行するには、Rational Suite をインストールする必要があります。

その他の Rational ツール

Rational ClearCase

Rational ClearCase は、大規模なプロジェクト チーム向けの構成管理ソリューションです。ClearCase は、ClearCase LT の機能に加え、次の機能を備えています。

- より進んだビルド管理
- 分散サーバーのサポート
- 自動データ複製

詳細については、<http://www.rational.co.jp/products/clearcase> を参照してください。

Rational XDE

Rational XDE™ は、ビジュアルな設計/開発機能を提供する拡張された開発環境です。XDE は Microsoft Visual Studio .NET や IBM WebSphere と統合されているため、開発者は使い慣れた環境で作業を行うことができます。また XDE は、Rational がサポートする Eclipse プラットフォーム ベースの Java 統合開発環境 (IDE) と介して単独で使用することもできます。

詳細については、<http://www.rational.co.jp/products/xde> を参照してください。

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、『**Rational Suite 管理ガイド**』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: 自由にカスタマイズできる変更依頼管理ツール。このツールを使用すると、ソフトウェア開発工程全体の障害と修正、拡張依頼、ドキュメントの変更などの変更アクティビティを追跡できます。ClearQuest の Web インターフェイスを使用すると、ClearQuest の主要な操作を行うことができます。

Rational ClearQuest MultiSite: 自由にカスタマイズできる Windows と Web ベースの変更依頼管理ツール。このツールを使用すると、地理的に分散したユーザーが、障害と修正、拡張依頼、ドキュメントの変更などの変更アクティビティを追跡できます。リモート サイトごとに中央のデータベースのレプリカを簡単に作成し、そのサイトで行われた変更と他サイトで行われた変更との同期をとることができます。

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 Developer Network は、Rational 製品のすべてのユーザーが利用できます。

Rational ProjectConsole: 自由にカスタマイズできるプロジェクト管理ツール。このツールを使用して、最適な実装方法の選択、反復型プロジェクトの計画と管理、進捗や品質の測定を行うことができます。

Rational ProjectConsole Template Builder: ProjectConsole のプラグインの 1 つ。このプラグインを使用して、ProjectConsole Web サイトに表示されるプロジェクト成果物の収集を自動化するテンプレートを作成できます。

Rational PureCoverage: 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: Visual Basic アプリケーション、Visual C++ アプリケーション、Java アプリケーションのパフォーマンス ボトルネック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 のインターフェイスを使用すると、RequisitePro の主要な操作を行うことができます。

Rational Robot: TestManager で実行する機能/パフォーマンス テストの自動テスト スクリプトの作成に使用します。

Rational Rose: 全世界でシェア No.1 のビジュアル コンポーネント モデリングおよび開発ツール。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ル化できます。

Rational Rose RealTime: リアルタイムの組み込みソフトウェア開発用の、UML ベースのモデリング/開発環境。

Rational SoDA: ソフトウェア ドキュメント自動作成ツール。各種 Rational ツールのデータからレポートを作成します。組み込みのテンプレートをそのまま使用することも、カスタマイズして使用することもできます。

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 Windows 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: Windows または UNIX で利用可能な、ソフトウェア開発に最適化された Rational Suite のエディション。Team Unifying Platform のツール、Rational Rose (Enterprise Edition)、Rational Purify、Rational Quantify、Rational PureCoverage が含まれています。

Rational Suite DevelopmentStudio RealTime: リアルタイム システムや組み込みシステムの開発者と設計者向けに最適化された Rational Suite のエディション。Windows または UNIX で利用できます。Team Unifying Platform、Rational Rose RealTime、Rational Purify、Rational Quantify、Rational PureCoverage が含まれています。

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 が含まれています。その他すべての Suite のエディションには、Team Unifying Platform のツールが含まれています。

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、Rational PureCoverage、Rational Purify、Rational Quantify、Rational Robot、Rational TestFactory が含まれています。

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 (RUP): 柔軟性の高いソフトウェア開発プロセス プラットフォーム。チームの生産性を向上させ、ソフトウェア開発のさまざまなアクティビティに対するガイドライン、テンプレート、ツール メンターを使用して実践的なソフトウェア開発手順を提供します。

Rational XDE: 拡張された開発環境 (eXtended Development Environment)。.NET と Java の両方のプラットフォーム用の Rational の設計/開発ツールを単一の環境に統合しています。

あ

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。

エージェント コンピュータ: Rational Agent ソフトウェアがインストールされている、テスト スクリプトを再生するコンピュータ。TestManager は、特定のエージェント コンピュータでテスト スクリプトを再生するように設定することも、テスト スクリプトを実行可能なその次のエージェント コンピュータで再生するように設定することもできます。エージェント コンピュータは、主に、Rational Suite TestStudio、TeamTest、Robot でのパフォーマンス テストに使用します。

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。

か

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、管理する担当者。

拡張ヘルプ: Rational Unified Process とユーザーが追加するカスタム情報へのリンクを提供する Rational Suite for Windows の強力な機能。

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

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、時間を節約し、信頼性が高く、予測可能で正確なプロセスを実現できます。

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。要求、モデル、モデル要素、ドキュメントなどのさまざまな種類の成果物があります。

た

ツール メンター: Rational Unified Process に記述されているアクティビティを実行する際の、個々の Rational ツールの使用手順。

テスト ケース: テスト入力の設定。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: システムを開発し、テストを行うために使用する成果物。

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、文書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。

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。

は

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、プロジェクトの成功という目標に近づきます。

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。

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。

ビルド: 最終的な製品に提供される機能のサブセットを実行する、動作可能なシステムまたはシステムの一部分。

フィーチャ 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。

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。

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

メトリクス: プロジェクト アクティビティの測定基準。

や

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、文書化し、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: 一連のバージョンから成り、バージョン ツリーの形に構成されているオブジェクト。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 ソフトウェア: 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。

利害関係者：システムの結果によって実質的な影響を受ける個人。

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特定、予測し、これに対応してリスク軽減計画を作成します。

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するように、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役割によって実行される一連のアクティビティ。

Rational Suite Team Unifying Platform
「Team Unifying Platform」を参照
Rational Suite TestStudio
「TestStudio」を参照
Rational Suite の導入 41
Rational TestManager
「TestManager」を参照
Rational Unified Process
「RUP」を参照
ソフトウェア開発の図 19
Rational XDE
「XDE」を参照
RequisitePro 16
Web インターフェイス 16, 22, 23, 27
アナリスト 22
開発者 27
テスト担当者 34
マネージャー/プロジェクト リーダー 37
要求管理 22
Robot 34
Rose 23
QualityArchitect 29
RealTime 31
アナリスト 23
開発者 26, 29
テスト担当者 34
マネージャー/プロジェクト リーダー 37
RUP 16
Rational Process Workbench 41
マネージャー/プロジェクト リーダー 37
利点 18

S

SoDA 17
アナリスト 24
開発者 29
テスト担当者 36
マネージャー/プロジェクト リーダー 39

T

Team Unifying Platform 16, 37
TestManager 17
UNIX 30, 34
アナリスト 24
開発者 30

テスト担当者 34, 35
マネージャー/プロジェクト リーダー 37
TestStudio 34

U

UCM 17, 28, 38
マネージャー/プロジェクト リーダー 37
例 28
UML 23
UNIX Suite 30, 34

X

XDE 43

あ

アナリスト 21
AnalystStudio 22
ビジュアル モデルの使用 23
ユース ケースの使用 22

か

開発者 25
共通の問題 25
組み込みシステム 31
コードのテスト 29
フォワード エンジニアリング 26
リアルタイム ソフトウェアの開発 31
リバース エンジニアリング 26
拡張ヘルプ 41
カスタマ サポート xii

き

機能テスト 34

く

組み込みシステム 31
グラフ 39

こ

コードのテスト 30
コンポーネント テスト 30

さ

最善のソフトウェア開発手法 37

し

自動テスト 34
進捗状況の確認 37
進捗状況の測定 28

そ

ソフトウェアの開発
開発工程の図 19
共通の問題 25
組み込みシステム 31
初期段階でのコードのテスト 29
テスト担当者 33, 37
フォワード エンジニアリング 26
リアルタイム システム 31
リバース エンジニアリング 26
ソフトウェアのテスト 33

つ

ツール メンター 41

て

テスト 33
開発者 29
コード 34
コンポーネント 30
初期段階でのテスト 30
テストの作成 34
パフォーマンス 30, 35
メモリ リークの検出 30, 35
要求 34
テスト ケース 24

テスト担当者 33
TestStudio 34
共通の問題 33

と

統一変更管理
「UCM」を参照
統一モデリング言語 23
統合ソフトウェア プロジェクト管理 39
ドキュメント ix

は

パフォーマンス テスト 30, 35
反復型開発
チーム内の役割 19
フェーズ 19

ひ

ビジュアル モデリング
コード生成 26
ソフトウェア設計 26
リアルタイム システム 31
品質確認 33
品質評価 33

ふ

フォワード エンジニアリング 26
プロジェクト管理 37, 39
プロジェクト ステータス 28, 39
プロジェクト ステータスの確認 24
プロジェクトの管理 37, 39
プロジェクトの計画 37
プロジェクト メトリクスの収集 39
プロジェクト リーダー 37
共通の問題 37
プロジェクト レポート 24, 39
プロフェッショナル サービス 41
分析と設計 25

へ

- ヘルプ ix
- Rational Suite チュートリアル 42
- 変更管理
 - アナリスト 22
 - コード内の変更の管理 27
 - プロジェクト リーダー 37

ま

- マネージャー 37

め

- メトリクス 39
 - コード 28
- メモリ リーク 30, 35

ゆ

- ユース ケース 22, 24
 - テスト 24

よ

- 要求 21
 - テスト 24
- 要求管理 21
 - AnalystStudio 22
 - 反復型開発 21
 - ユース ケースの使用 22
- 要求の定義 22
- 用語集 45

ら

- ラショナル ユニバーシティ ix

り

- リアルタイム ソフトウェア 31
- リスク 37
 - リスクの管理 37
- リソース ix
- リバース エンジニアリング 26
- リリースの決定 33