

既存環境でソフトウェア開発プロセスの集中化、自動化、高速化を実現

**Rational.** software

The IBM logo, consisting of the letters 'IBM' in a bold, sans-serif font, is positioned in the top right corner of the page. The letters are white and set against a black background.

## ソフトウェア開発のためのビルドおよびリリース・プロセス自動化ソリューション

## 目次

- 2 はじめに
- 3 今日開発が抱えるいくつかの最重要課題への取り組み
- 5 IBM Rational Build Forge 製品
- 6 システムの仕組み
- 11 IBM Rational Build Forge 製品の利点
- 12 ソフトウェア開発のためのソリューション
- 14 品質の向上
- 23 監査およびコンプライアンス管理の実現
- 26 効果の実現
- 27 まとめ

## はじめに

IBM Rational® Build Forge™ ソフトウェアは、ソフトウェア開発プロセスの集中化、自動化、高速化を可能にする、ビルドおよびリリース・プロセス自動化のためのシステムです。Build Forge を IBM Rational ソフトウェアの統合ソフトウェア開発スイートと併用することにより、企業は開発環境をエンドツーエンドで自動化し、ソフトウェアのデリバリーを合理化するとともに、持続可能なコンプライアンス管理を実施することができます。Rational Build Forge は、既存ツールを活用して、チームの「手渡し」の改善、自動化、追跡可能性を実現する柔軟な統合プラットフォームを、異種のソフトウェア開発アプリケーションを抱える企業に提供します。

Rational Build Forge は、分散したチームの足並みを揃え、作業効率向上を支援します。また、このソフトウェアによって複数のアプリケーションやハードウェア・プラットフォームへの一元的なアクセスが可能になるため、報告、トラッキング、さらにはハードウェア資源の最適利用などをグローバルに行うことができます。Rational Build Forge 製品を使用することにより、さらに賢く仕事ができ、作業に余裕が生まれます。チームは繰り返しの作業から解放されるため、ビルドおよびリリース・プロセスを長期的に計測し、改善を図ることに集中できます。

本書では、Rational Build Forge がビルドおよびリリース・プロセスを管理および自動化するアプローチを説明するとともに、ソリューションでチーム、プロセス、システムを統合することでソフトウェア開発サイクルをどう向上させ、その結果、開発チームの生産性、製品品質、コンプライアンス管理にいかにより多くの改善をもたらすことができるかについて論じます。

### 今日開発が抱えるいくつかの最重要課題への取り組み

プロシージャー、人、ツールを集めて、結束の固い、効率の高いソフトウェア・デリバリー・システムを構築する場合に、IBM Rational Build Forge は力を発揮します。このソフトウェアによって、アプリケーション・ライフ・サイクル管理 (ALM: Application Lifecycle Management)、ツールの標準化、コンプライアンスという 3 つの大きな開発上の課題に取り組むことができます。

### ALM の課題

最適なパフォーマンスを実現し、ソフトウェアを迅速に生み出すためには、製品がコーディングからスタートして実運用に至る過程において、人、プロセス、テクノロジーが複雑に絡み合ったネットワークを統合する必要があります。開発チームは通常、次のような共通する課題に直面します。

- 開発、構成管理、品質保証 (QA)、リリース、お客様サポートといった重要なフェーズを担当するチームが、組織の区分、ツール・セットの連携不備、地理的な距離によって分断されている。
- 各チームが独自のプロセスを持ち、しかもそれらの多くが手作業であるか、ほとんど文書化されていないため、ビルドおよびリリースの重要なノウハウが数名の担当者に属人化している。
- バグ・トラッキング・データベースやソース・コード管理システムなどの必須のツール間につながりがなく、重要情報がサイロ化しているが、それらを連携させるには手間や時間がかかる。その結果、組織は顧客への納入物の全容を、実運用中に壊れるまで把握できない。

質の高い製品をより短いサイクルで送り出すことがますます求められる中で、開発チームがこれに対応するには、再現性、信頼性、およびトラッキング性能を提供できる確固とした基盤が必要です。このような主幹業務のインフラストラクチャーを構築しようとする場合、チームには次の 2 つの選択肢があります。

- 1 つのプロバイダーからエンドツーエンドで ALM ソリューションを購入する
- 既存のツールやプロセスを補完したり、これらに接続できる、統合型のフレームワークを実装する

IBM Rational Build Forge は、お客様がどちらの方法を採用しても、チームの開発効率向上を支援します。

#### 標準化するのか、しないのか

設定不要で統合でき、しかも堅固なツールのスイートを希望するチームにとって、IBM Rational のソフトウェア開発プラットフォームは魅力ある選択肢です。IBM Rational ClearCase®、IBM Rational ClearQuest®、IBM Rational RequisitePro®、IBM Rational Build Forge、IBM Tivoli® Provisioning Manager などのソフトウェアを組み合わせると、要件から実運用までの包括的な自動化とコントロールが可能になります。信頼できるプロバイダー 1 社からソリューションをトータルで調達することで、統合基盤が安定し、顧客サポートもシンプルになります。

全面的な標準化が現実として難しい環境もあります。開発ライフ・サイクルのどのポイントでもコミュニケーションと統合が欠かせませんが、チーム間の整合を強制することで縄張り争いが生じ、実施できなくなる場合があります。チームが標準化に合意しても、他の企業を買収したプロジェクトの外注先が別のツールを使っているなど、整合が再び乱されるような状況が発生する可能性もあります。結局、一部の開発チームには、異種混合のツール・セットをサポートし、将来の開発要件に適宜対応できる、柔軟なアーキテクチャーが必要となります。IBM Rational Build Forge は、組織の開発環境に含まれている任意のツールを自動化、統合、報告を可能にする強力な選択肢です。

IBM Rational Build Forge は、現行のすべてのツールと連携するほか、新しいツールを随時使用環境に追加できる設計であるため、統合型の開発エコシステムの構築に伴うハードルも一気に乗り越えることができます。このソフトウェアには、さまざまなオペレーティング・システムやハードウェアを超えて機能にアクセスするための、一貫性のあるインターフェースが用意されています。これを使うことで、1 つのインターフェースから Microsoft® Windows® プログラムや UNIX® のコマンドを呼び出すことができます。あるいはプロセスを 1 つ起動して、そこから Apple Macintosh、Microsoft Windows、Linux® などの複数のプラットフォーム上で並行してタスクを実行することもできます。このように、開発チームは、各自の環境に最も適したソリューションを自由に選択できるのです。

#### 監査およびコンプライアンス管理の実現

自社のプロセスが法令に遵守していることを証明する必要がある企業の数はいくらも増えています。開発チームが監査をパスするには、保有する開発プロセスが再現可能であり、十分なアクセス制御を有していることを実証しなければなりません。また、システムの変更理由や変更者についても一貫して文書化しなければなりません。自動化された ALM システムがなければ、監査担当者に開発の全容を漏れなく報告するのにアプリケーションから逐一データを集約することになり、それには膨大な時間を要します。

IBM Rational Build Forge を開発エコシステム内の他のツールと連結すると、開発プロセスに関する詳しい情報が自己文書化されて記録されるため、長い準備作業なしに監査を受けることができます。

このソフトウェアはリリース内容、変更者とその理由も含めた変更内容、実施したテストなどに関する重要なデータを逐一収集して詳しい部品表 (BOM: Bill of Materials) を作成するため、これをコンプライアンスや監査のチェック時の証拠として利用できます。

さらに、Rational Build Forge を IBM Rational スイートに含まれるすべての他の製品と共に使用すると、必要な承認、ワークフロー、監査証跡も含む、最初のコーディングから実運用に至る、開発におけるすべてのコンプライアンスの実践を管理、自動化、トラッキングすることができます。

#### IBM Rational Build Forge 製品

Rational Build Forge 製品ラインは次の 3 つのアプリケーションで構成されています。

- IBM Rational Build Forge Standard Edition: 包括的なビルドおよびリリース・プロセスの自動化、サーバーへの分散アクセス、プロセスのトラッキングなどを可能にします。
- IBM Rational Build Forge Enterprise Edition: プロセスのスレッド化、サーバー構成のディスカバリー、動的なサーバー・プール化といった高度な能力を備えています。
- IBM Rational Build Forge Adaptor Toolkit: サード・パーティーのソフトウェア開発アプリケーションと設定不要で統合でき、合わせて柔軟なアプリケーション・プログラミング・インターフェース (API: Application Programming Interface) を提供します。これらの機能によって、サード・パーティー製のソフトウェア構成管理 (SCM: Software Configuration Management) システム、独自ツール、ビルドおよびリリース環境とシームレスにリンクして、統合性や情報共有のほか、ソース・コード、障害、テスト結果などのトラッキング能力を向上させることができます。

### システムの仕組み

Rational Build Forge は本質的に自動化とプロセス管理のためのソリューションであり、その目的はビルドおよびリリース・プロセスの効率化です。このソフトウェアはいわばプロセスのための中央のリポジトリであり、そこから多様なプロセスを実行、トラッキング、スケジュールしたり、さまざまなコンピューターに分散したりできるのです。

このセクションでは Rational Build Forge 製品が行う自動化の範囲について述べ、システムのコンポーネントがどのように連携するのかを説明します。

### プロセスの定義と実行

IBM Rational Build Forge 内では、プロセスを「プロジェクト」と呼ばれる一式のタスクとして定義することができます。プロジェクトに含まれるタスクにはそれぞれコマンド・ラインや環境変数が含まれます。実行の際には、これらのコマンド・ラインや環境変数はネットワーク上のコンピューターに渡すことができます。プロジェクトが定義されると、システムはスケジュールに従って、あるいは許可されたユーザーがそれらを起動したときに、プロジェクトを実行できます。プロジェクトが開始された後は、システムはタスクを 1 台または複数台のサーバー上で実行し、その結果をデータベースに記録します。システムをプログラムして、一部のタスクに他のプロジェクトを起動させることもできます。

プロジェクトが終了すると、Build Forge は組織内の適切なメンバーにプロジェクトの成否を通知します。プロジェクトのアクティビティには、ソース・コードのコンパイルのような、従来からあるビルド処理を含めることができますが、Build Forge はそれらをはるかに超える、多彩なタスク自動化および管理能力を備えています。プロジェクトの中に、ソースのチェックアウトからビルド、テスト、デプロイといったフェーズまでをカバーした複雑な開発プロセスが絡んでいる場合があります。通常、プロジェクトは、次に示すタスクをすべて実行します。

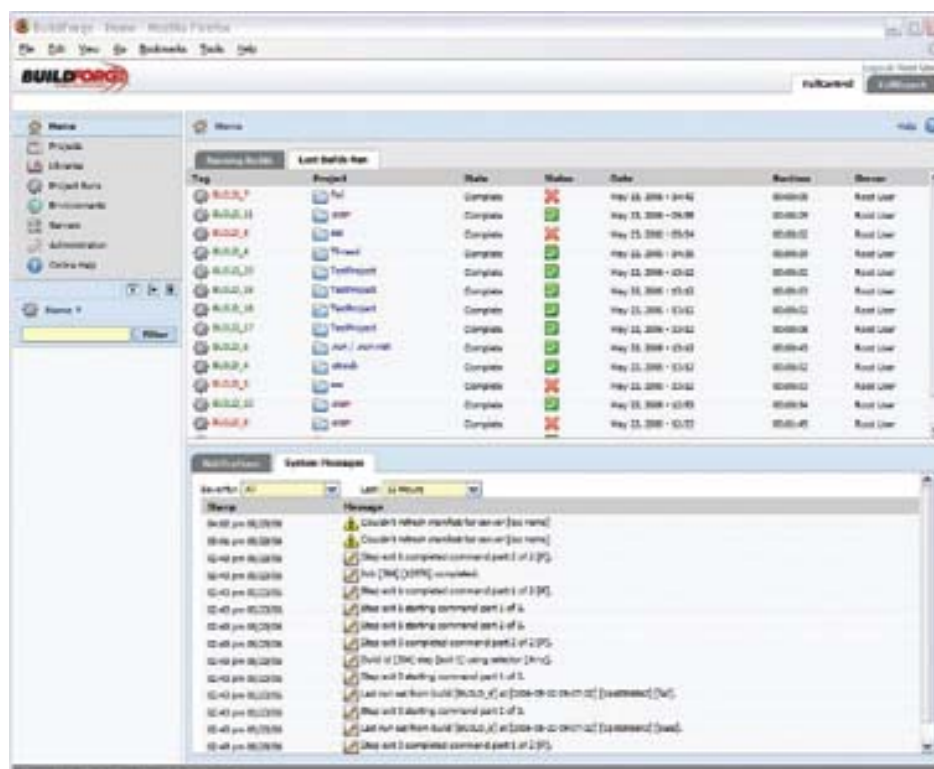
1. ソース・コード・ファイルをチェックアウトする
2. コードをコンパイルし、途中で進捗状況を報告する
3. コンパイルの成功後、自動で単体テストを実行する
4. インストーラーを作成する
5. ダウンロード用のサイトにインストーラーを公開し、チームにその旨を通知する
6. インストーラーを実行して、インストール済み実行可能ファイルを作成する
7. 実行可能ファイルに対して自動テストを実行する
8. テスト結果を報告する
9. 従属するプロジェクトを起動して標準ライブラリーを更新する
10. 実行可能ファイルなどのファイルを QA に送り、追加テストを行う
11. 完成したリリースを Web サーバーや CD 生産などの実運用環境にデプロイする

そのほかにも数々のプロジェクトやアクティビティーが考えられます。IBM Rational Build Forge を使うと、次を行うことができます。

- 文書のソース・ファイルから PDF (Portable Document Format) ファイルを作成し、Web サイトのダウンロード・エリアにコピーしたり、Web ページに掲載されているファイル一覧の更新や、Web サイトの検索インデックスの更新を行う
- 各開発者が個別のコンポーネントのビルドを起動できるようにするほか、1 日単位でマスター・ビルドを実行し、個別のビルドを改めて全部呼び出して、製品全体をコンパイルする
- 1 つのプロセスを 5 台または 20 台のコンピューターで順次または並行して実行する
- プロジェクト・チームに自分のビルドの実行は許す一方、Web サーバーの更新は許可されたチームに限定する

#### システム・コンポーネントの仕組み

IBM Rational Build Forge 製品ファミリーは、特定のツールの使用を強いることなく、信頼できるビルドおよびリリース・プロセス管理システムを開発チームに提供するというシンプルなコンセプトからスタートしました。この発想から生まれたのが、世界各地に分散した開発チームが使用できる、柔軟かつ堅固な自動化インフラストラクチャーです。Rational Build Forge ソリューションによって、ターゲット・マシンの場所やオペレーティング・システムに左右されることなく、Web ベースのインターフェースから、任意のコマンド・ラインを複数のサーバー上で実行することができます。

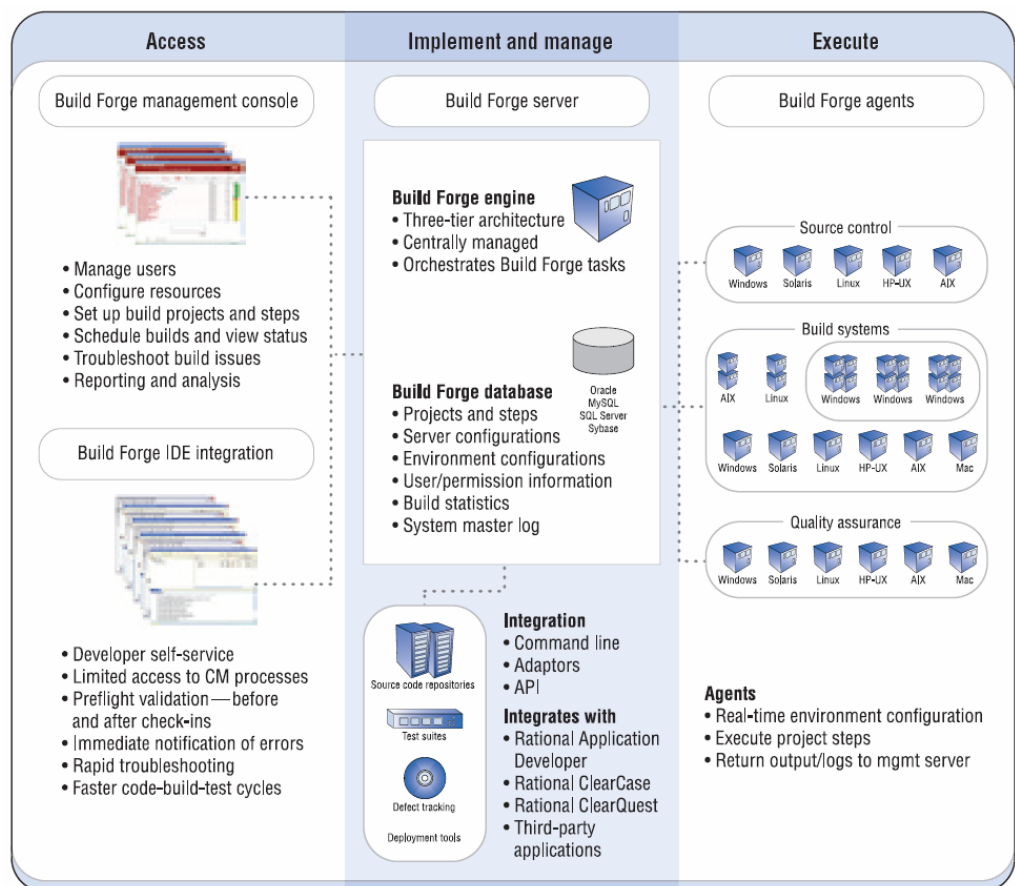


Web ベースの管理コンソールは、すべてのアクティビティをリアルタイムで把握できるため、地理的に分散した開発チームに最適です。

システムには次のコンポーネントが含まれています。

- 管理コンソールは、システムのユーザー・インターフェースです。Web ベースの PHP アプリケーションで、Apache HTTP サーバー上で稼働します。このコンソールにより、コマンドを整理してプロジェクトにまとめたり、これらのコマンドが必要とするサーバー資源や環境変数を管理することができます。
- プロセス・エンジンは、管理コンソールから入力され、データベースに格納された情報を使用してエージェントと通信したり、プロジェクトのタスクや通知を実行します。プロセス・エンジンは、管理コンソールからの命令に従って稼働します。
- データベースは包括的なプロジェクト情報を格納し、プロジェクトの各実行をトラッキングします。ユーザーやシステムのアクションはこのデータベースに格納され、そこから監査や報告用のデータを抽出したり、分析できます。

- コンピューターでは、それぞれが使用しているオペレーティング・システム用にコンパイルされたエージェントを実行して、システムからの命令に応答できるようにします。エージェントは、データベースから情報を取り出して個々のコマンドがそのニーズに完全にマッチした環境で実行されるように、プロジェクトに適した環境を動的に構成します。



### IBM Rational Build Forge システムのコンポーネント

#### 例

Build Forge システムは、これらのコンポーネントを装備して、開発資源を 1 つにまとめるフレームワークの役割を果たします。以下に示すのは、Rational Build Forge で自動化およびトラッキングできる開発シナリオの例です。

- 構成管理担当者 (CM: Configuration Manager) が管理コンソールを使用して、主力製品の最新開発バージョンを複数のオペレーティング・システム用にコンパイルするプロセスを毎日実行するスケジュールを組みます。システムはそのプロジェクトの完了に必要なプロセスを逐一把握しており、実行時には仕様に基づいて適切な環境を作ります。いくつかの共通のステップを実行した後、プロジェクトは Red Hat Linux オペレーティング・システム用と Microsoft Windows オペレーティング・システム用の 2 つのサブプロセスにフォークします。これらのサブプロセスはソース・コードをそれぞれチェックアウトし、並行して製品のコンパイルを開始します。
- 構成管理チームがプロセス全体を通じてプロジェクトの進捗をリアルタイムで把握しています。これらのプロセスの 1 つが失敗すると、サブプロセスが開発者にその作業の担当者を通知します。プロジェクトが正常に完了すると、完成した実行可能ファイルはテスト用として内部 Web サーバーに公開され、QA チームに自動的に通知されます。
- 開発者が担当している製品のテスト用ビルドを起動します。この開発者は、普段使用しているマシンが停止していることに気付いていません。システムはプロセスを別のサーバーに自動的にリダイレクトし、実際に使用されるサーバーを記録します。プロセスが完了すると、開発者には最新のビルドによる成果物の場所が記載された電子メールが届きます。開発者や構成管理担当者は、サーバーの停止に慌てて対応する必要はありません。
- 更新された Web ページのファイルを 1 日に何度かステージング・サーバーから収集し、それらを公開サーバー上の適切な場所にコピーするプロセスを Web 管理者が作成します。この Web 管理者は、自動化されたプロセス 1 つでファイルを更新し、Web サーバーを再起動して、古いバージョンのファイルを別のマシンにアーカイブすることができます。Web 管理者が不在の場合は、チームの他のメンバーに最小限の必要なトレーニングを提供して、このプロセスを実行させることができます。
- プロジェクト・チームは、製品のストレス・テストを多数のマシンで実行する必要があります。プロジェクト・チームは構成管理チームと協力して、プロセスが複数の部署にあるサーバーを利用して、夜間に実行されるようにスケジュールを組みます。プロジェクト・チームには、予定されているテスト用に、サーバーへの制限付きのアクセス権が与えられます。そのため各部署ではサーバー環境を損なわずに、資源を安全に共有することができます。テスト中にインストールまたは生成されたファイルは、システムが自動的にクリーンアップして、サーバーを元の状態に戻します。

### IBM Rational Build Forge の利点

自動化に対する IBM Rational Build Forge の独自のアプローチは、他の方法と比較して明らかに優れています。このような利点は次から生じています。

- Build Forge は、ソフトウェア開発プロジェクトの真の成果物、すなわち実行可能ファイル上に構築されている
- Build Forge 製品には、開発中の状況を正確に報告できる能力が装備されている

開発の成果物—実行可能ファイルという基盤上に構築

コードから最終的な製品を生成するプロセスは、コードを書く労力に比べて陰が薄くなりがちです。ソフトウェアのビルドに関わるプロセスは、製品を出荷する寸前に仕方なく行う作業と見なされがちで、組織内の構成管理またはビルド・チームがわずかな人数で処理しているのが現状です。しかし、実行可能ファイルは、製品開発チームが現実に作成する唯一の納入物なのです。次の例を考えてみましょう。

- 開発者が数カ月かけてコードを記述したにも関わらず、ビルドの不備によって実行可能ファイルがお客様に納品された後に正しく機能しない。
- ソース管理システムによれば障害は修復されたはずだが、ブランチが最終リリースで障害を解決できない。コードのトラッキングに関しては SCM システムと同様に役立つはずのソース管理システムだが、特定の問題が解決したかどうかをトラッキングしようとしても、製品で実際に何が行われたのかを表示する機能を装備していない。
- バグ・トラッキング・データベースによれば問題は修復されたはずだが、データベースへの入力には人手に頼っている。チームが作成した実行可能ファイルは、まさに記録の産物です。正確を期すには、実行可能ファイルから開発管理データを取り出せるような方法が必要です。Rational Build Forge を使用して製品を作成すると、システムは実行可能ファイルに入り込んだバグ修復を報告し、バグ修復に使用されたコードを割り出すことができます。Rational Build Forge は開発プロセスを 1 つにまとめ、問題から原因となったコードへと遡ってトラッキングすることを可能にします。その結果、ビルドおよびリリース管理プロセスを、アプリケーション・デリバリー・プロセスの他のフェーズと密接に連携させて管理することができるようになり、開発上のトラブルを最小限に抑えることができます。

### 状況の報告

Rational Build Forge は、ソフトウェアのコンパイルであれ、自動テストや Web サイトの更新であれ、ある特定のプロセスの実行から得られる詳しい記録を基にしてレポートを作成します。情報源が手入力である他の文書ベースの開発システムとはこの点が異なります。

Rational Build Forge を使用した統合型開発環境には、次のような利点があります。

- ソース管理システムで分かるのはチェックインされたコードだが、Rational Build Forge では、リリースに実際に含まれているコードを知ることができる
- 障害トラッキング・システムで分かるのはリリースに組み込まれるべきフィックスだが、Rational Build Forge では、どのフィックスが実際に組み込まれたのかを知ることができるほか、自動テストを実行してフィックスを検証することができる
- 特定のソフトウェア・リリースの問題がお客様から報告された場合、Rational Build Forge を参照してそのリリースに含まれているコードを調べることができる
- Web サイトで問題が発生した場合は、Rational Build Forge で変更内容を正確に把握し、必要に応じて以前のバージョンにロールバックすることができる

### ソフトウェア開発のためのソリューション

Rational Build Forge は、開発プロセスのスピードアップ、製品品質の向上、分散したチームの機能向上、監査およびコンプライアンス要件の準拠のいずれに対しても効果を発揮します。

#### プロセスのスピードアップ

Rational Build Forge は、次のような一般的なボトルネックを解消する機能によって、開発プロセスのスピードアップに貢献します。

- 各担当グループ間のプロジェクトの連携が効率的でない
- 処理の速度が遅く、逐次的である
- システムのダウン時間
- ハードウェアの使用率が低い
- バッチ・プロセスが不透明

このセクションでは、Rational Build Forge システムの機能によってこれらの問題がどのように解決されるのかについて説明します。

#### 連携の迅速化と無駄な作業の削減

Rational Build Forge の自己文書化機能によって、メンバーが異なるビルにいる、あるいは異なる国にいるといった場合でも、緊密なグループ協調を図ることができます。チームは下流工程の次のチームに適切な情報を渡すことに気を配る必要がなくなります。情報の格納と受け渡しはシステムが行うため、チームは安心して担当部分の作業を正しく処理し、情報をシステム内にカプセル化することに集中できます。このプロセスは次のような機能によって支えられています。

- **BOM:**システムはファイルの変更を示すチェックポイントを含めた、プロセスの実行内容に関する記述をまとめ、パッケージに格納します。
- **ログ:**システムはアクティビティーやコマンド出力を記録し、特定のプロジェクトに対して指定したセキュリティ設定に従って、組織内にそのログを公開します。
- **メモ:**ユーザーはプロジェクトやプロジェクト内のタスクに関するメモを挿入できます。

これらの機能を組み合わせることにより、開発プロセス中の状況を明確かつ正しく描き出すことができます。システムはどのマシン上で誰が何を行ったか、どのような結果が得られたか、そしてそのプロセスにどの程度の時間を要したかをトラッキングします。このようなトラッキング機能によって、監査規則の効率的な遵守が可能になります。このトピックの詳細については、本論文の『監査およびコンプライアンス管理の実現』セクションを参照してください。

#### 処理速度の向上と動作可能な時間の延長

Rational Build Forge を使用すると、その並行処理とサーバー・プール機能によって、処理速度を向上させ、動作可能な時間を延長できるというメリットがあります。これらの機能によって次が可能になります。

- 複数のマシンをグループ化してサーバー・プールとし、複数のプロセスの実行を開始します。1 台のサーバーがフル稼働の状態になると、システムは同じプール内の他のサーバーに処理をリダイレクトします。
- 1 つのプロセスを使用して複数のマシン上で並行してタスクを起動することで、結果を得るまでの時間を短縮できます。
- 異なるマシン上でプロセスを起動して、例えば異なるオペレーティング・システム用に、異なるバージョンのアプリケーションを、それぞれに適したネイティブ環境でビルドできます。

- Rational Build Forge があるプロセスを実行しようとしたときに、そのプロセスの既定の実行マシンがダウンしている場合、そのプロセスをサーバー・プール内の他のメンバーに引き継いで、プロセスを中断なく実行させることができます。
- 同一のタスクをプール内のすべてのサーバーで一度に実行します。例えば、Linux サーバーのプールに含まれるすべてのサーバー上にコードのアップデートを配置する場合、1 回のタスクでこれを行えば、並行して更新処理を行うことができます。

#### 少ないハードウェアで処理をスピードアップ

IBM Rational Build Forge を実装すると、部署間でハードウェア資源の共有ができます。このような集中化が行われていない部署では、ハードウェアが 1 つのプロジェクト専用で使用されるケースが多く、長期的にはハードウェアのアイドル状態を招いて、使用率が低下します。Rational Build Forge を導入すると、次のシナリオが可能になります。

- 時々 Macintosh システム上でプロセスを実行するニーズのある部署に、その Macintosh システムが存在する場所に出向いたり、そのシステムに変更を加えたりすることなく、システムへの限定的なアクセスを提供できます。アクセスを特定のタスクに制限し、クリーンアップを自動化できます。使用頻度が上がった場合は、アクセスを取り消すこともできます。
- 1 つの部署のサーバーを別の部署のバックアップ用に設定して、ある部署でメインのサーバーがダウンした場合には、優先マシンがオンラインに戻るまで、作業を自動的にサブサーバーに切り替えられるようにできます。
- マシンのグループを利用可能サーバーのプールに指定して、関係する部署やプロジェクトに関わらず、どのプロセスにも利用できるようにできます。
- 他の部署から処理能力の一部のみを利用できるように、マシンをセットアップできます。

これらのシナリオは、組織での処理速度の向上とコストの削減に役立ちます。

#### 持続的改善のためのプロセスの明確さと分析

Rational Build Forge は、コーディングから実運用までの間に実行したすべてのアクティビティをトラッキングします。そのため、保存されている豊富な情報を、チームの効率の向上に利用することができます。また、Rational Build Forge が生成する構築済みのレポート集を使用して、プロセスを分析し、継続的な改善につなげることができます。膨大なバッチ・ファイルやスクリプトを使用してもプロセスに対する可視性がほとんど得られないのとは対照的に、Rational Build Forge では次のようなことができます。

- 膨大なログ・データを検索せずに、エラーを即刻正確に特定する
- 個別のプロセスを見直して一般的な処理時間を調べ、異常や傾向を探す
- 複数の部署間、または部署内で、問題の発生源をトラッキングする
- サーバー使用率を長期的にトラッキングする
- コード・ベースで変更頻度が最も高いエリアを特定し、エラーのホットスポットを正確に探し当ててプロジェクト計画を改善する
- 文書化されたシステムのスキーマを使用して、Rational Build Forge データベースから独自のレポートを生成する

### 品質の向上

注目すべきはスピードだけではありません。組織内に Rational Build Forge を導入すると、一貫性のあるビルドおよびリリース自動化システム、標準化された開発プロセス、チームのコミュニケーション改善などの点から、さまざまな形で品質向上を実現できます。

### 再現可能なプロセス

IBM Rational Build Forge はビルドおよびリリース・プロセスに一貫性を提供するため、プロセスは毎回同じ方法で実行されます。Rational Build Forge からアクティビティーを実行すると、以前のビルドで使用された値をシステムに提供してプロジェクトの実行を再現したり、表示される値を実行時にカスタマイズすることができます。Build Forge ソリューションは、繰り返し型開発や、テストや診断の目的で過去のプロセスを再現する必要がある場合に理想的なソリューションです。プロセスを共有ライブラリー化して、プロジェクト間で使いまわすこともできます。これによって、標準化による有益な品質管理が実現され、新規プロジェクトの立ち上げに要する時間を短縮できます。

### 知識の保持

退職や人事異動は現実としてどのような組織でも避けられません。しかし、Rational Build Forge があれば、その難局も切り抜けることができます。システムのナレッジ・ベースが自動的にプロセス情報を企業インフラストラクチャー内に集中的に格納して、情報への投資を保護すると同時に、格納された個々のプロセスの学習に必要な時間を大幅に短縮します。Rational Build Forge のナレッジ・ベースによって、チームのメンバーは次のようなことを行うことができます。

- プロセスの現在の定義を見る
- プロセスの以前の実行結果を見る
- プロセス定義の変更履歴を見る
- プロセスの変更理由メモを閲覧する

システムはプロセスのナレッジに関するリポジトリを作成し、それを自動的にアーカイブします。そのため、標準化された、一貫性のある再現可能な形でプロセスを実行できます。

#### 自動化による人為ミスの削減

Rational Build Forge は、包括的なビルドおよびリリース自動化システムの実装に必要なツールを提供します。プロセスを設計した後、それを簡単に自動化して、組織内の他の部門と共有することができます。

システムでは次のような自動化がサポートされています。

- プロセスが定期的に行われるようにスケジュールする
- 実行、または一部実行の結果の成否に基づいてアクティビティを実行する
- ソース・コードに対する変更に基づいてアクティビティを実行する
- プロセスの成否に基づいて自動的にレポートを作成する
- 実行のたびにプロセスのコンポーネント、実行理由、変更作成者を記録した BOM を自動生成する

これらの機能により、どのチームの生産性も向上します。

- 開発チームはより短いサイクルでコードをビルドして、コーディング結果の成否を即座にフィードバックできるため、作業を正しい方向に進めることができます。
- QA チームは各リリースの入手可能性や内容に関する詳しい情報を入手できるほか、自動テストの結果からリリースの問題の検証を始めることができます。QA チームは BOM を通じて変更箇所と実行すべきテストを把握できます。自動テストを行うことで、多数の一般的な問題を検討対象から除外できます。
- 構成管理チームは日常業務の管理に割く時間を減らして、開発プロセスの最適化に集中できます。
- IT チームは IBM Rational Build Forge のレポートを使用して、現在のシステムのニーズを査定したり、将来の需要を予測できます。

#### 集中型のコントロールと分散型のアクセス

チーム間のコミュニケーションがタイムリーなリリースの障害になることは少なくありません。効率的なビルドおよびリリース自動化システムがなければ、構成管理チームが新しいビルド、新しいテスト、新しいプロジェクトに対応できず、開発プロセスに遅れが出る可能性があります。さらに、ビルド・プロセスに接続できない開発者は結果を待たねばならず、その間は生産性が低下します。同様に、プロジェクトが再現不能で、文書化もされていないと、製品の各バージョンに関する情報を特定するのが難しくなり、お客様の問題を修復する際に最もこずるのが問題の再現だということにもなりかねません。

Rational Build Forge は、管理を集中化すると同時に、役割に応じた、セキュアなアクセスを提供することで、分散した開発チームをサポートします。多数の部署のプロセスを中央の 1 つの報告システムにまとめ、それらのプロセス（およびそれらが生成する情報）に対するアクセス権を、必要に応じて幅広く組織内に配布することができます。集権化された構成管理グループがプロセスを作成およびテストし、そのプロセスを使用する機能を適切な担当者のグループに付与します。この担当者たちが同一の部署に属している必要はありません。また、構成管理チームの正式なメンバー以外でも、独自のプロセスを作成し、企業内でそれらを共有することができます。同時に、どのユーザーに自動プロセスの実行、レポートの閲覧、新しいプロセスの定義などを許可するかについても管理することができます。

Rational Build Forge を使用すると、開発者は自分の組織からアクセス権が認められた範囲で、ビルド・プロジェクトを起動したり、即座にフィードバックを入手したり、新しい機能やフィックスをテストすることができます。このようなタイプのセルフ・サービスによって、開発者は仕事の生産性を高め、CM がさまざまな要求に対応する時間を短縮することができます。この発想を、自動化するあらゆるプロセスに応用することができます。

#### テスト・サイクル短縮によるアジリティーの実現

自動テストが QA チームの洞察に取って代わるようなことは決してありませんが、Rational Build Forge を使うことによって、効率的な自動テストをビルドおよびリリース・プロセス全体に組み込んで、テスト結果を組織全体で共有することができます。さらに、Rational Build Forge システムの自己文書化機能によって、ビルド内で何が行われたのか、その実行可能ファイルには何が含まれているのかといった情報が各ビルドに含まれるようになるため、それぞれのテスト・サイクルの生産性が向上します。アジャイル開発などの手法を始めとする反復型の開発プロセスでは、絶えず「コード-ビルド-テスト」というアクティビティーによって、開発やテストのチームに頻繁にフィードバックすることが必要です。Rational Build Forge は、強力な自動化、スケジューリング、ソース管理の統合機能を備えて、チームが継続して統合を進められるようにします。

IBM Rational Build Forge により、テスト担当者は、次のような詳細が決定された後に、製品の評価を開始できます。

- 自動テストで検出可能な問題は、そのリリースが QA チームの目に触れる前に開発チームにフィードバックされます。そのため手動テストは自動テストにパスしてからでなければ開始されません。
- 過去に発生したエラーを手早く再テストできるため、回帰プロセスのスピードと品質が向上します。そのため新たな問題を招く恐れが減り、開発者やテスト担当者は問題の解決に集中することができます。
- ストレス・テストおよびロードを自動的に実行できます。
- Rational Build Forge は、テストをすべてのターゲット・マシン上で開始し、その結果を収集するためのインフラストラクチャーとなります。
- Rational Build Forge のカスタム・フィルターがテスト結果を前処理し、警告や失敗の詳細なリストを作成して高度なレポートを提供します。

- BOM レポートにはビルドに含まれるバグ修正が詳しく記録されています。そのため、それらを個別にテストしたり、自動テストの結果と付き合わせたりすることができます。
- Rational Build Forge は使用中のソース管理システムやバグ・トラッキング・データベースと統合することができます。

IBM Rational Build Forge は、例えば、テスト・グループ向けに次のように利用することができます。

- スモーク・テストを実行する。各ビルド後に基本的なテストの一部を実行し、その成否を報告して、そのビルドが変更後も正しく作動するかを判断します。
- 完成したビルドを複数のターゲット・プラットフォームにインストールする作業を自動化する。Rational Build Forge を使ってテスト環境を構成し、テスト担当者にマシンがテスト・スクリプトの実行に利用できることを通知できます。
- 実行するテストを決定する。Rational Build Forge システム内のプロジェクトには、それぞれ異なるクラスのタグが付けられます。このプロジェクトのクラスを使って、起動するテストを決めることができます。
- 開発者に即席のスクラッチ用ビルドが必要な場合、テストを省略する。
- スケジュールされたビルドでスモーク・テストを実行し、主要コンポーネントが破損していないことを確認する。
- 実運用用ビルドとして、システム資源のかなりの部分を必要とするテストのすべてのスイートを、できればスケーリングや負荷テストも含めて実行する。

これらの機能によって、プロセス中にテストをより多く、より効率的に組み込めるようになり、製品リリースの品質がさらに向上します。

#### 統合に適した設計

IBM Rational Build Forge は本来、アプリケーションを互いに結び付け、より大きな開発タスクを実行できるようにするためのソリューションです。Rational Build Forge によって、アプリケーション、オペレーティング・システムのコマンド、バッチ・ファイルやシェル・スクリプトなどを、プロジェクト内に簡単に混在させることができます。

Rational Build Forge にはコマンド・ラインをカプセル化できる独自の機能があり、それによって、コマンドを実際に送出する前に、そのコマンドが必要とする環境がターゲット・マシン上に作成されます。また、コマンド用の環境を定義して、使用したい他のソフトウェアと統合することができます。

アプリケーションを結び付けるという基礎的な機能に加えて、Rational Build Forge は、以下のように他のソリューションと統合する能力も備えています。

- Lightweight Directory Access Protocol (LDAP) 標準を介して既存のユーザー・データを統合することができます。そのため、新しい情報システム用のユーザー・ログインを作成する必要がありません。
- IBM Rational Build Forge には、設定不要で使用できる、主なソース・コード管理、障害トラッキング、テスト自動化システム用のアダプターが用意されており、それによって開発アクティビティからプロセスを起動し、情報を更新することができます。
- API によって、使用する情報システム内の他のアプリケーションから IBM Rational Build Forge 機能を実行し、プロセスをきめ細かくコントロールすることができます。

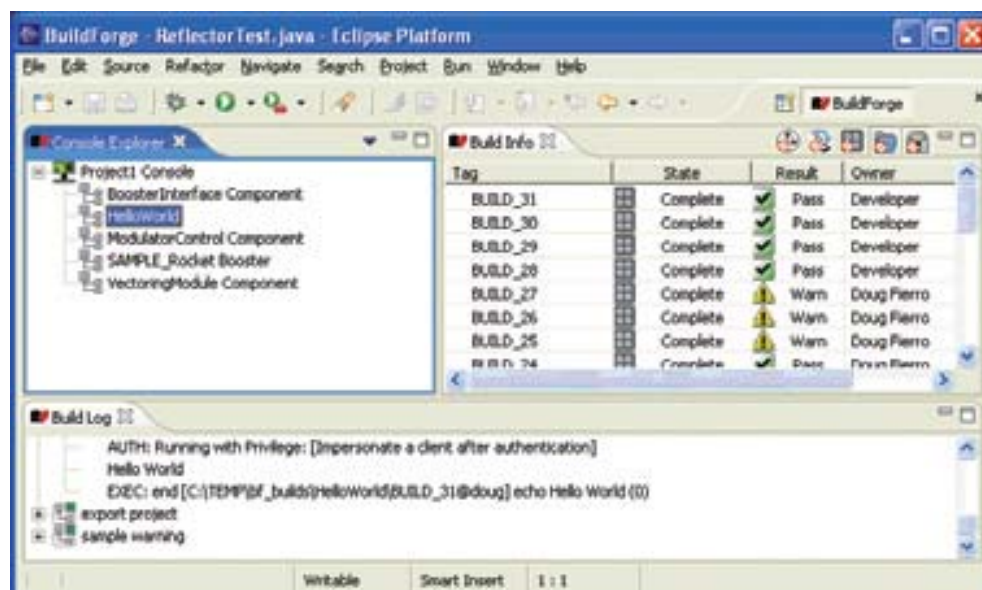
これらの機能により、Rational Build Forge は開発環境と完全に一体化し、企業の情報システム内でも周囲に悪影響を及ぼしません。

設定不要の価値ある、IBM Rational ソフトウェア開発スイート

IBM Rational Build Forge Standard Edition と IBM Rational Build Forge Enterprise Edition ソフトウェアには、IBM Rational ソフトウェア開発プラットフォーム・ソリューションと設定不要で緊密に統合できる機能が装備されており、お客様は即座にその恩恵を受けることができます。これらの製品を共に使用することにより、企業は要件から実運用に至る開発ライフ・サイクル全体を管理、自動化、およびトラッキングすることができます。統合が可能な製品は次のとおりです。

- IBM Rational ClearCase :Build Forge は統一変更管理 (UCM: Unified Change Management) 環境およびベースの Rational ClearCase 環境で Rational ClearCase Versioned Object Bases (VOB) と直接リンクし、シームレスな統合と、ビルドやリリースごとの詳しいソース・コードのトラッキングを可能にします。双方向に情報を共有できるため、Rational Build Forge のビルド・プロジェクトとビルド成果物は、Rational ClearCase ベースラインの一部として格納することができます。

- IBM Rational ClearQuest : Rational Build Forge は、使用中のビューに基づいて、特定のビルド・セッションに対する Rational ClearQuest のアクティビティを自動的に検出します。正常なビルド終了ステータスが検出されると、Rational BuildForge は関連付けられている障害の状態を自動的に解決済みとし、Rational ClearQuest アプリケーションのビルド・レコードを参照します。Rational Build Forge はまた、IBM Tivoli Provisioning Manager などのデプロイ・エンジンと連携できるデプロイユニットも作成します。
- IBM Rational Application Developer : Rational Build Forge は、開発者が実運用ビルド・プロセスに Rational Application Developer 統合開発環境 (IDE) から制限付きでアクセスできるようにすることで、開発と構成管理の間の壁を取り払います。これによって開発者はグループ・ビルドの前にプリフライト・ビルドを実行してその結果をすぐ入手できるようになり、リリースの品質向上とスピードアップを図ることができます。



システムは IDE 内から IBM Rational Build Forge コンソールと直接通信します。

#### 他社製アプリケーションとのリンク

IBM Rational Build Forge Adaptor Toolkit 製品によって、ソース・コード管理ツール、障害トラッキング・ツール、テスト管理ツールなどの他社製のソフトウェアのユーザーは、ソフトウェア構成管理およびビルド環境の間にシームレスなリンクを作って効率を向上させるとともに、ソース・コード、障害、テストの変更をトラッキングすることができます。CVS、Perforce SCM、Borland StarTeam、Microsoft Visual SourceSafe、Subversion、Bugzilla などの環境用には、サード・パーティー製のアダプターが用意されています。自社製のソフトウェア開発ツールやサード・パーティー製のソフトウェア開発ツールには、これらのアダプターを加工したり、独自の統合機能を作成したりすることができます。

Rational Build Forge Adaptor Toolkit ソリューションには次のような統合オプションが多数用意されており、開発プロセスの自動化および最適化に役立てることができます。

- ソース・コードの変更の監視: IBM Rational Build Forge アダプターは、サード・パーティーのソース・リポジトリを常時監視し、変更が発生すると自動的にビルドを実行します。システムはソースの変更を直接監視し、測定基準値を収集して、ビルドごとにリポジトリの状態を詳しくトラッキングします。Rational Build Forge の管理コンソールを使用すると、変更の発生箇所を、提出コメントから実際のファイルの違いに至るまで、ビルド単位で把握することができます。
- 障害トラッキングおよびテストの自動化: Rational Build Forge の障害トラッキング・アダプターは障害のトラッキングを支援し、特定の障害について「検証済み」や「終了済み」などの状態を BOM に報告します。障害は自動的に報告されるため、特定のビルド内でどの障害が修正されたかを把握できます。
- 提供されたテスト・アダプターを使用した、テスト・ケースの実行と BOM へのテスト結果収集の自動化

アダプターを使ってソース・コードの変更、障害、およびテストと特定のビルドを関連付けることにより、ビルドのコンポーネントを詳しく理解することができます。IBM Rational Build Forge は、加えられた変更、ビルド日時といったビルドの主な統計情報を収集し、その情報を中央の拠点に保存して、迅速にアクセスできるようにします。

### 監査およびコンプライアンス管理の実現

先に述べたように、コンプライアンスは大部分の開発チームが抱える共通の課題です。チームが会社の財務システムを直接担当していなくても、そのグループが自社の収益ストリームを左右する製品やサービスを提供していれば、監査に備える必要があります。

サーベンス・オクスリー法の余波を受け、開発プロセスを標準化および文書化し、再現可能にすることは、業務上欠かせないベスト・プラクティスであると多くの企業が考えるようになりました。この鍵になるのが、この監査データをいかに少ない手間を集められるかという点です。IBM Rational Build Forge は重要なデータを収集し、最初のコーディングから実運用に至るまでのトラッキングを可能にして、コンプライアンスや監査の証拠として使用できるようにします。

Rational Build Forge を使用してビルドおよびリリース・プロセスを実行すると、システムがプロセスを最初から最後まで自動的にトラッキングします。Rational Build Forge システムでビルドするプロセスが増えるほど、開発環境の全容がより明確に把握できるようになります。システムはすべてのプロセスのバージョン情報を保存します。そのため、成果物のバージョンごとに、変更の内容や変更者、変更理由などを知ることができます。さらに、システムはプロセスの実行が完了するごとに、BOM を作成します。

### プロセスの自動トラッキング

Rational Build Forge はあらゆるプロジェクト実行に関するデータを格納するため、このシステムを通じて実行すれば、ごく単純なスクリプトでも便利なコンプライアンス用ツールになります。ソースの場所から Web サーバーにファイルをコピーするバッチ・ファイルやシェル・スクリプトを Rational Build Forge から実行すると、これらのバッチ・ファイルやシェル・スクリプトは即座に、単なるスクリプトを超えた機能を発揮します。

次の例を考えてみましょう。

- プロセスの成否を、電子メールだけでなく Web ベースのダッシュボード上にも通知する
- 実行したコマンド、およびその結果の出力またはエラー・メッセージをログに記録する
- システムがスクリプトを反復実行するようにスケジュールして、標準プロセスが毎回同じ方法で、必要な回数実行されるようにする

- システムがスクリプトを実行する際に、1 台のマシンの可用性に依存するのではなく、利用可能なサーバー資源を探すことができる
- 誰がスクリプトを起動したかをシステムで記録する
- スクリプトの変更など、その他のユーザーのアクションを記録する

さらに、Rational Build Forge のユーザーのメモは、加えたすべての変更の理由を文書化する際に役立ちます。

Rational Build Forge を既存のツールと統合すると、情報のサイロをすぐに利用できる有用なコンプライアンス・データに変換できます。

#### プロセスのバージョン管理

Rational Build Forge にプロセスを追加すると、プロジェクトに含まれるステップ、プロジェクトに必要な環境変数、そのプロジェクトを実行するサーバー（およびサーバー・プール）など、そのプロジェクトに関する大量の情報が収集されます。このようなプロジェクト・レコードはプロセスを変更すると随時更新されます。システムはプロセスの最新バージョンを、そのプロセスのデフォルトの命令セットとして保存します。

プロジェクトを実行するたびに、システムは実行時の情報をコピーして保存します。プロセス定義が変更されていても、このプロジェクト実行の記録を見直して、特定のタイミングでどのステップが実際に実行されたのかを調べることができます。さらに、現在に至るまでの途中の変更は無視して過去の実行命令に従って、過去の実行を再現することができます。

このように、Rational Build Forge に追加されたプロセスは、即座に再現とトラッキングが可能になります。プロセスの変更が原因で品質に問題が発生した場合は、正常に終了した最後の状態に戻ることができます。システムはプロセスのレコードを自動的に自分のデータベースに格納しますが、これらのプロセス命令を、使用しているソース管理システム内にアーカイブすることもできます。これによって、プロセスの命令と、製品の生成に使用したファイルを共に格納しておくことができます。このアーカイブのプロセスも、Rational Build Forge を使って自動化することができます。

#### 部品表 (BOM) を介した自己文書化システム

IBM Rational Build Forge には、構成可能な BOM が含まれています。Build Forge システムは実行されるすべてのプロセスについて自動的に BOM を生成します。BOM はプロセス実行に関する情報を簡潔にまとめたパッケージで、これを使用して、プロセスの結果を見直すことができます。BOM は、製品のさまざまな利用者に合わせてプロセスを文書化するのにも役立ちます。

システムは BOM 内に、各ビルドで注目すべき特定の項目を自動的に組み込みます。

- 与えられたコマンドを使ってプロセスにタスクを追加し、BOM に追加情報を書き込むことができます。例えば、ファイルに関する情報をプロセスの作業用ディレクトリーに格納し、プロセスがそれらのファイルをどのように変更したかをさまざまなチェックポイントで調べることができます。
- BOM はプロセスの中でグループ間で受け渡しされるチケットの役割を果たします。BOM がなければビルドのパッケージ内容は知ることはできませんが、BOM によってチームは新しいビルドがどのような意味を持つのかを直ちに評価することができます。
- BOM は、技術文書の担当チームがリリースの文書化を始める際に、作業用の概要として使用することもできます。

BOM は、プロセスとそのエンドツーエンドの結果を、監査やコンプライアンスの検証に使いやすい 1 つのパッケージにカプセル化する役目を果たします。

### 効果の実現

柔軟なアーキテクチャー、既存スクリプトが再利用できる、設定不要で統合可能といった利点を持つ IBM Rational Build Forge は、使用環境に素早く実装し、すぐにその能力を確認できます。開発チームがとりあえずの効果を得るには、まず 1 つのプロジェクトの基本的なビルドおよびリリースを自動化することから開始し、最適化を繰り返しながら微調整できます。最初の成果が上がったら、他のプロジェクトやチームを巻き込んで、実装を拡張していきます。

### システムの実装: 段階的な配置

Rational Build Forge システムを実装する場合、プロセスを一挙に変更する必要はありません。システムを一度に導入することは、必要でもなく、望ましいことでもありません。むしろ逆に、システムにプロセスを 1 つ追加することから導入を開始するのが最適です。その結果に満足したら、次に追加する主要なプロセスを選び、残りのプロセスにも優先順位を付けます。

組織規模が大きい場合は、1 つの部署でまず採用し、効果を実証された後で他の部署に拡張します。トップから使用を強制されて仕方なく採用するのとは異なり、ユーザーがその価値を認めていると、システムは組織的に成長できます。

システムに追加したプロセスは、その 1 つ 1 つが組織に関する知識の管理されたりポジトリーであり、これを必要とする人にはアクセスを許可し、必要としない人にはアクセスを制限ができます。

### まとめ

IBM Rational Build Forge は、開発チームの生産性を向上させ、質の高い製品をより短期間で市場に投入できるようにする、総合的なビルドおよびリリース・プロセス管理および自動化システムです。Rational Build Forge システムを使用するお客様はまず、主要プロセスを既存のシステムに追加するところから開始します。投資収益率を分析すれば、プロセスをさらに追加したいという意欲が高まるはずです。このシステムはレガシー・ツールと共存できるため、すべての環境を再編成する場合と比較すれば、導入が及ぼす影響ははるかに少なく済みます。このシステムを使用すると、チームはプロセスやマシンに対するコントロール能力の獲得を実感できます。他方、システムに組み込まれている安全機能のおかげで、情報やツールをこれまで以上に幅広く配布できるようになります。システムによって、さまざまな由来を持つチームやツールの統合が可能になると同時に、ユーザーが自分の情報を中央のリポジトリに格納することで、複雑な環境プロセスの分散したコンポーネントにも、細かく目が行き届くようになります。



お問い合わせは、IBM ビジネスパートナー、製品販売店、弊社営業担当または、ダイヤル IBM (0120-04-1992) へ。受付時間：月～金 9:00～18:00 (祝日 12/30～1/3 を除く) 携帯電話でおかけのお客様は下記の電話番号でご利用ください。ダイヤル IBM 03-6220-8002 (この場合通話料金はお客様のご負担となります。)

Copyright IBM Japan, Ltd. 2007  
日本アイ・ビー・エム株式会社  
〒106-8711 東京都港区六本木 3-2-12  
Produced in Japan  
Nov 2007

All Rights Reserved

このカタログの情報は2007年11月現在のものです。内容は事前の予告なしに変更する場合があります。表示画面および印刷帳票の出力例のうち、特に断わり書きのない出力例のデータ部分は全て架空のもので、画面ははめ込み合成で実際の表示とは異なります。製品、サービス等詳細については、弊社もしくは IBM ビジネスパートナーの営業担当員にご相談ください。

Build Forge、ClearCase、ClearQuest、IBM、IBM ロゴ、Rational、RequisitePro、Tivoli は、International Business Machines Corporation の米国およびその他の国における商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

Microsoft および Windows は、Microsoft Corporation の米国およびその他の国における商標です。

UNIX は、The Open Group の米国およびその他の国における登録商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。