



Eclipse & TPTPを拡張した Rational品質管理ポートフォリオのご紹介

2006年1月31日
日本アイ・ビー・エム株式会社
ソフトウェア事業
ラショナルテクニカルセールス & サービス
富永 康之

本セッションの内容

- EclipseってJavaのエディターでは？
→そもそもEclipseやTPTPとは何か
- EclipseやTPTPを拡張したIBM Rational製品群とは？
→開発フェーズをまたがり、IBM Rational製品群は品質向上に
どう役立つのか

※扱わない内容

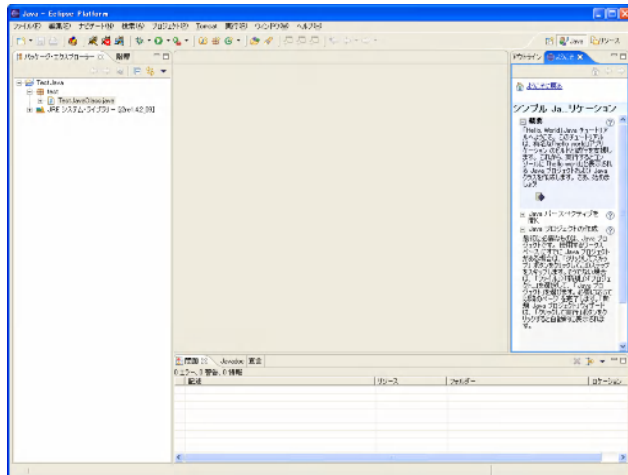
- ・Eclipse & TPTPの詳細な内部構造、使い方

アジェンダ

- Eclipse & TPTPの基礎
 - ～Eclipseとは
 - ～Eclipse TPTPとは
- Rational品質管理ポートフォリオ
- Rational品質管理製品を用いた開発 & テストイメージ
- (参考) Rational品質管理のアプローチ
- まとめ

Eclipseとは何か？

- 「ソフトウェア開発ツール」や「リッチクライアント」のためのオープンなプラットフォーム
- コアのプラットフォームにEclipse“プラグイン”を組み合わせて構成
- リリース毎に洗練され、機能を拡張



- 1999年OTIとIBMとの共同プロジェクトが開始
- 2001年11月、IBMはEclipseプロジェクトに対する4000万ドル相当のツールの供与を発表
- Eclipseプロジェクト発足

<http://www.eclipse.org/>

アプリケーション開発ツールにオープン(オープンソース:CPL)なプラットフォームを提供

- 2004年、IBMから独立し、Eclipse Foundation発足
- 2004年6月 Eclipse 3.0リリース

テストツールの悩み

- 多彩な役割と技術への対応が必要

- 様々な種類のテスト

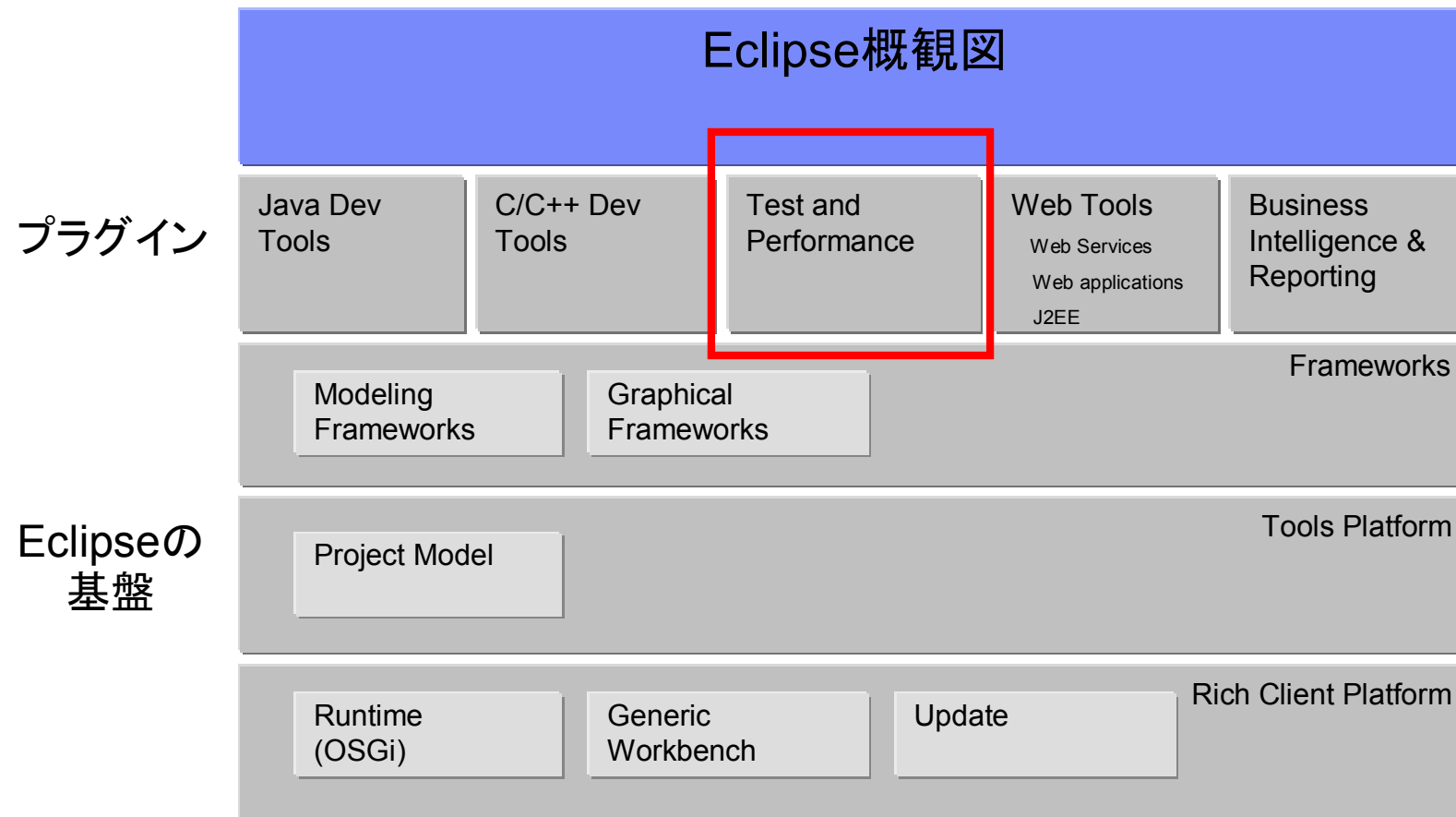
単体テスト、機能テスト、負荷テスト、メモリー分析、カバレッジ測定 etc...

- 様々なツール、異なるプラットフォーム、異なる操作方法

- 新技術への対応

Eclipse TPTPとは?

- Eclipseプロジェクトの一つ、Eclipseをベースとする
- Eclipse Test & Performance Tools Platformの略(旧Hyades)
- テストおよびパフォーマンス用ツールためのプラットフォームと機能を提供



Eclipse TPTP

4つのサブプロジェクトから構成される

TPTP = Test & Performance Tools Platform

▶ TPTP Platform

Eclipse TPTPのコアとなる部分。他のサブプロジェクトの基礎となる共通インフラストラクチャー

▶ Monitoring Tools

モニタリングやロギングを行うツール作成のためのフレームワーク（およびアプリケーションサーバーやシステムをモニターする提供的なツール）

▶ Tracing & Profiling Tools

アプリケーション実行のトレースやプロファイリングを行うツール作成のためのフレームワーク（およびJVMPIモニタリングを使用した典型的なプロファイリングツール）

▶ Testing Tools

アプリケーションをテストするツール作成のためのフレームワーク（およびJUnitベースの典型的なツール）

Eclipse TPTP イン트로ダクション・デモ

TPTP = Test & Performance Tools Platform

▷ TPTP Platform

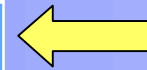
▷ Monitoring Tools

▷ Tracing & Profiling Tools

▼ ユーザーインターフェース

シーケンス図

統計およびパフォーマンス・ビューワー



▷ 実行環境

▷ Testing Tools

Eclipse & TPTP

ON DEMAND BUSINESS™

もう一步踏み込んで、Eclipse TPTPとは？

TPTP = Test & Performance Tools Platform

▶ TPTP Platform

- ▶ ユーザーインターフェース
- ▶ EMFモデル
- ▶ Analysis and rules
- ▶ データ収集とエージェント
- ▶ コミュニケーション
- ▶ 実行環境

▶ Testing Tools

▶ Trace & Profiling

▶ Monitoring

様々なツールが共通に使用する
基盤、およびデータモデルを
提供する！！

Eclipse TPTPのメリットとは？

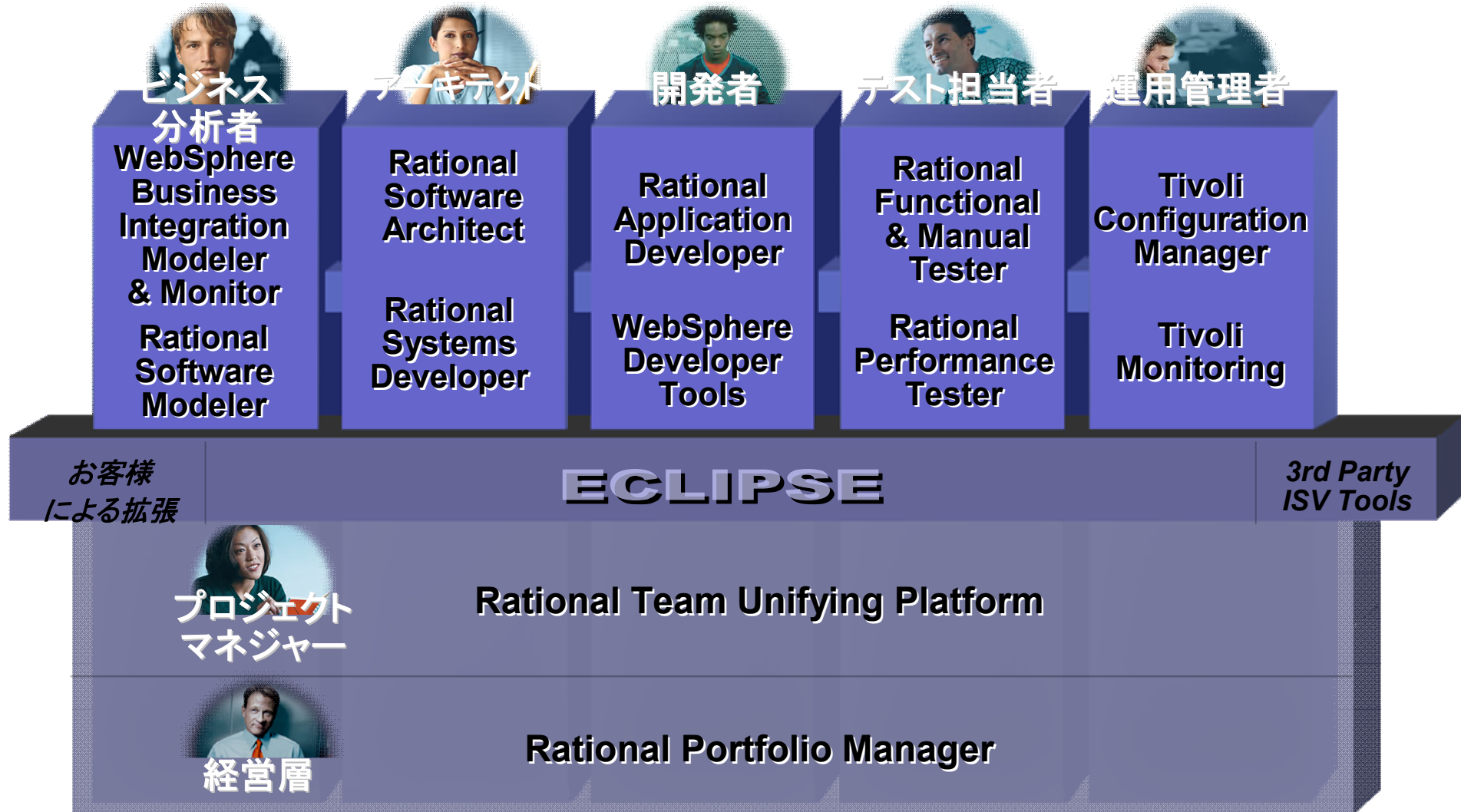
- TPTP自体がツールとしての機能も持つ
 - プロファイリング、JUnitベースのテストツールなど
- ツール結合の核となるプラットフォームである
 - 各種テスト&プロファイリングツールを共通の枠組み、共通のUIで利用可能
 - ※オープンソースと商用ツールの良いところを組み合わせた使い方が可能
 - データモデルをEMFベースに規定

アジェンダ

- Eclipse & TPTPの基礎
 - ～Eclipseとは
 - ～Eclipse TPTPとは
- Rational品質管理ポートフォリオ
- Rational品質管理製品を用いた開発 & テストイメージ
- (参考) Rational品質管理のアプローチ
- まとめ

The IBM Software Development Platform

開発の各フェーズをカバーする製品群



IBM Rational Application Developer (RAD)

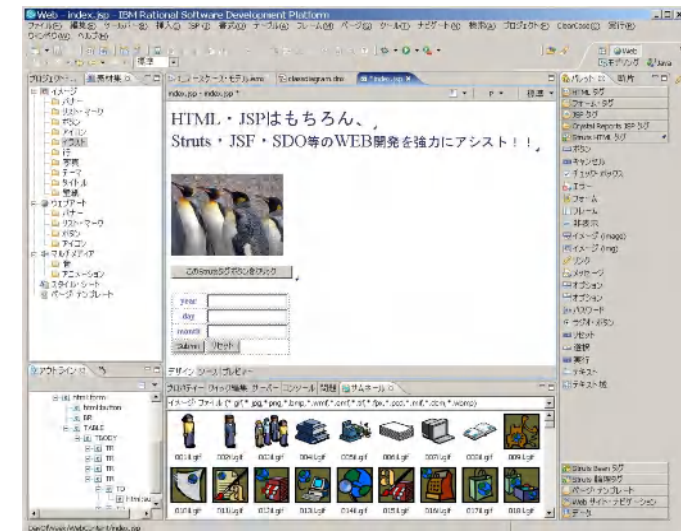
J2EE開発のための統合開発環境

J2EE統合開発環境

- 定評あるホームページビルダーベースのJ2EE向けGUI開発
- サーバーテスト環境
- JSF、UMLビジュアライズ、ポータル開発等、さまざまなテクノロジーをサポート

品質管理の観点からは下記機能を搭載

- 静的解析(コードレビュー)
- 動的解析(プロファイリング)
- コンポーネントテスト



ON DEMAND BUSINESS™

IBM Rational Functional Tester (RFT)

初心者、上級者ともに使える自動化テスト



- Web, J2EE, .Net アプリケーションに対する回帰テストの自動化
- 主な特徴
 - スクリプト言語にJavaを採用
 - 保守作業を軽減するためのテクノロジー (ScriptAssure™、正規表現を用いた期待値設定、共用テストオブジェクトマップなど)
 - データ駆動テストをサポート
 - 初心者にも優しいウィザードによるテスト支援
 - 専門家向けの強力なAPIとIDE



IBM Rational Performance Tester (RPT)

専門家だけでなく開発者もターゲットとした負荷テストツール

- 親しみやすくなった最新負荷テストツール

- 主な特徴

- 容易なテスト作成

- コーディングレスのテスト作成

- 入力フィールドなどを容易にパラメータ化

- 自動的にサーバ応答の相関関係を設定

- 強力なスケジューリング

- コーディングレスのスケジュール作成

- 現実の負荷シナリオをモデル化する柔軟性

- ボトルネック分析

- 遅い実行ページを判断する

- リソースのボトルネックを判断する

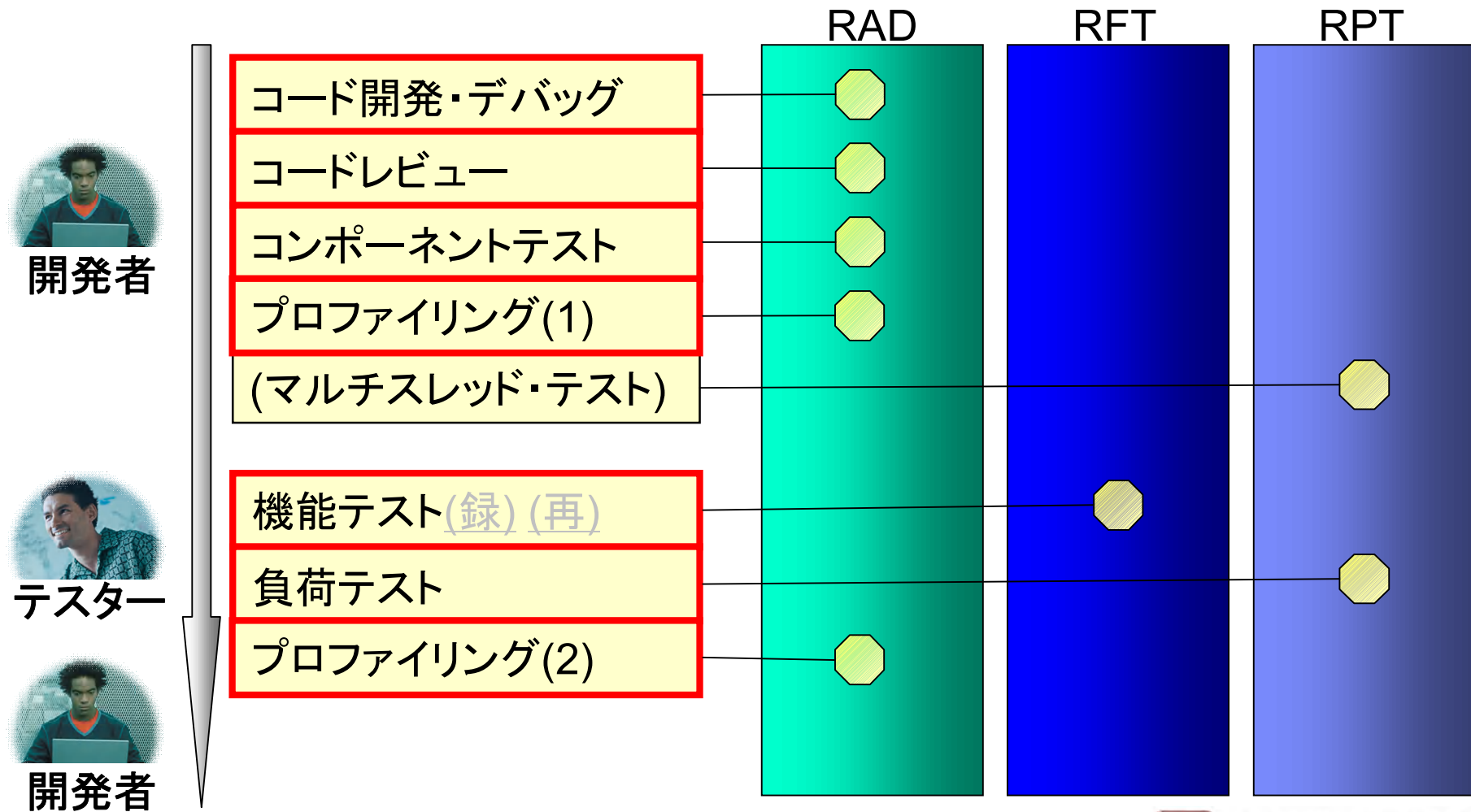


ON DEMAND BUSINESS™

アジェンダ

- Eclipse & TPTPの基礎
 - ～Eclipseとは
 - ～Eclipse TPTPとは
- Rational品質管理ポートフォリオ
- Rational品質管理製品を用いた開発 & テストイメージ
- (参考) Rational品質管理のアプローチ
- まとめ

Rational品質管理製品を用いた開発&テストイメージ

**ON DEMAND BUSINESS™**

(参考)コードレビュー <RAD>

ソースコードを解析→潜在的な問題の検出

- J2SE や J2EE のベスト・プラクティスをコードに適用
- 200種類以上のコーディング・ルールを提供:
 - J2EE ベスト・プラクティス
 - J2SE ベスト・プラクティス
 - グローバリゼーション
 - パフォーマンス
 - プライベート API
 - 命名規則
 - 設計方針
- テンプレートに従った新規ルールを追加可能

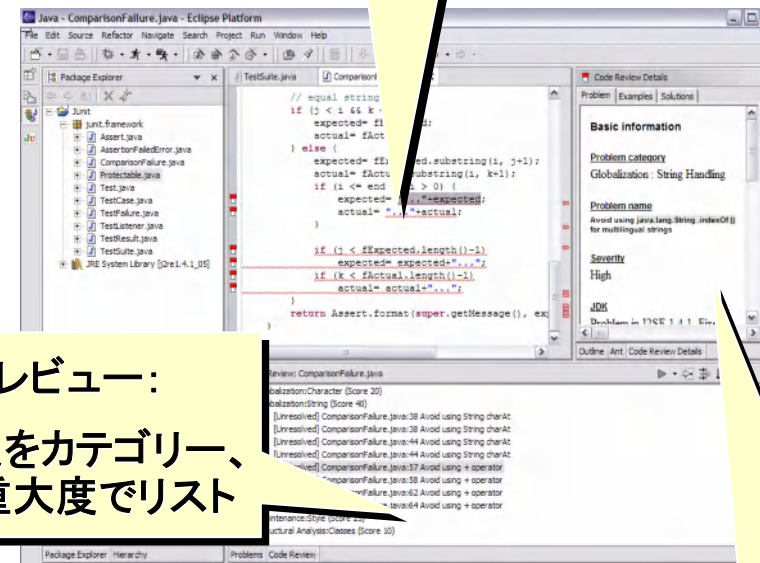
エディター:

ルール違反は赤下線

コード・レビュー:

ルール違反をカテゴリー、
ファイル、重大度でリスト

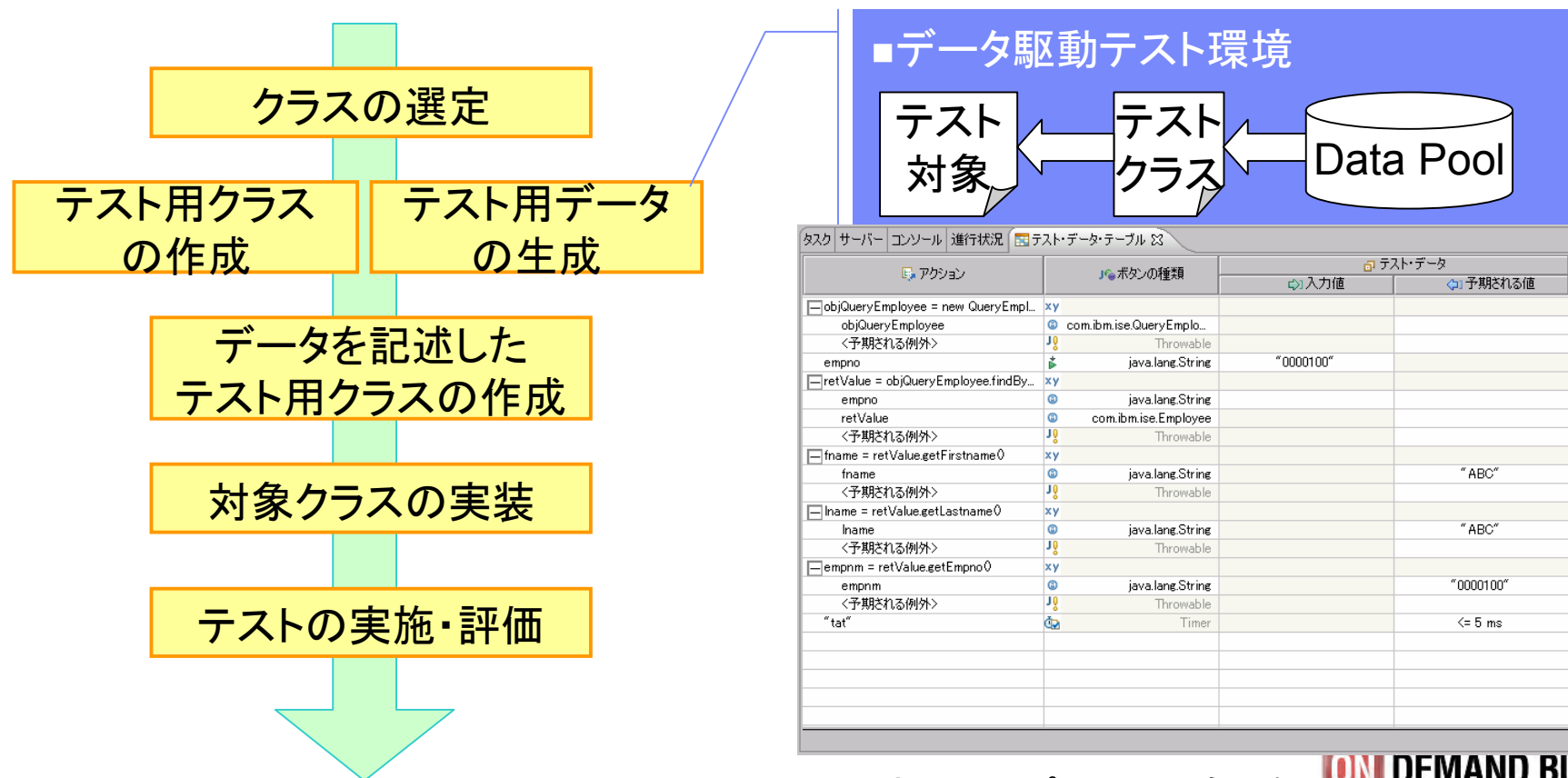
Why / Example /
How to fix



ON DEMAND BUSINESS™

(参考)コンポーネントテスト <RAD>

JUnitフレームワーク&TPTP(Hyades) ベースのデータ駆動テスト
 テスト対象：Javaクラス、EJB(Session bean, Entity bean)、Web サービス



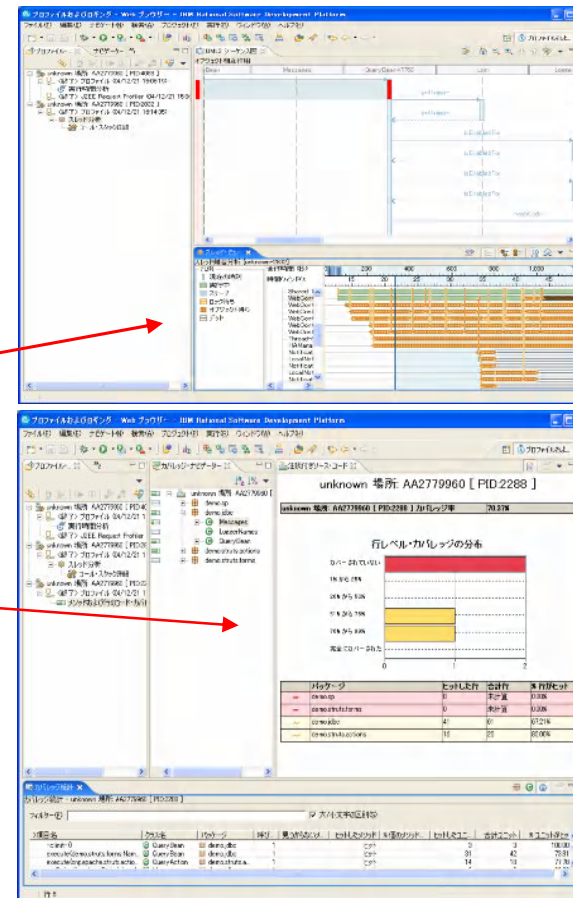
※HTML形式でのレポート出力も可能

ON DEMAND BUSINESS™

(参考)プロファイリング <RAD>

実際に動作させてアプリケーションの状況・ボトルネックを分析
 ※TPTP(Hyades)をベースにPurifyテクノロジーで拡張

- メモリー・リーク分析
 ヒープ・ダンプを取得しリーク候補を分析
 オブジェクト・リファレンス・グラフ
- パフォーマンス・ボトルネック分析
 パフォーマンスを検査してボトルネックを発見
- スレッド・ボトルネック検出
 競合、デッドロックの発見
- コード・カバレッジ
 メソッド・レベルのコード・カバレッジ
 行レベルのコード・カバレッジ
- Probekit
 トランザクションの実行時間ログ
 オペレーターへのレポート通知

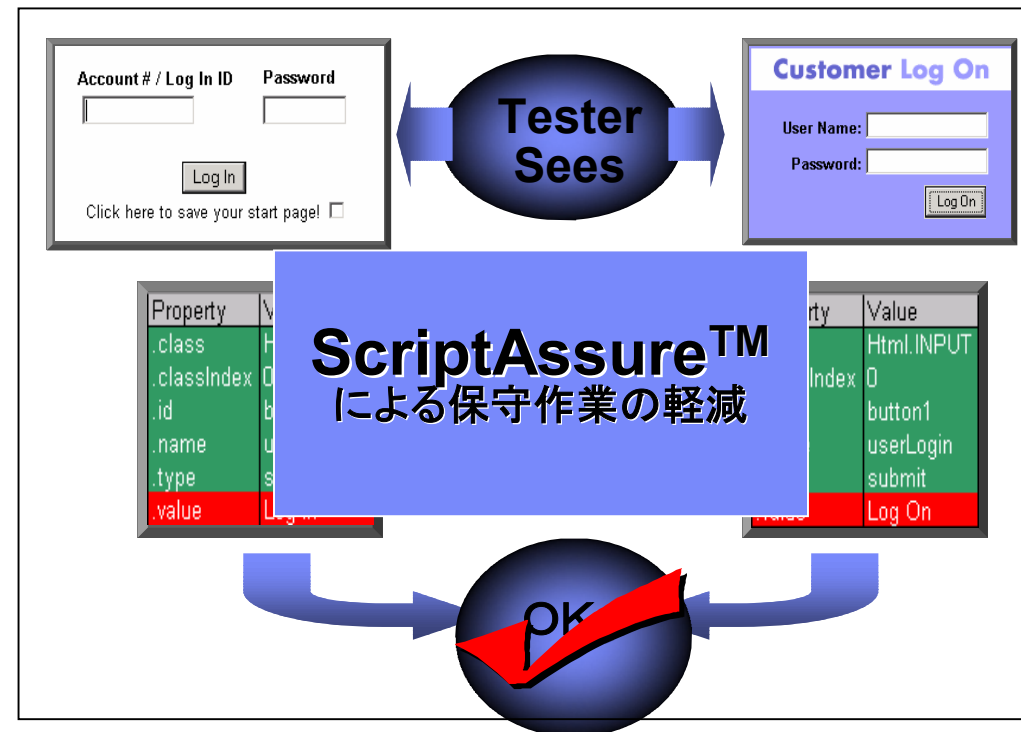


ON DEMAND BUSINESS™

(参考)機能テスト <RFT>

操作をキャプチャーし回帰テストを自動化

- スクリプト記録、(必要に応じて)編集、再生の3ステップ
- 保守作業軽減のためのテクノロジー
 - ScriptAssure™
 - 正規表現による期待値設定も可能
 - データ駆動テスト



ON DEMAND BUSINESS™

(参考) 負荷テスト・マルチスレッドテスト <RPT>

複数ユーザーの同時アクセスにより本番稼働をシミュレート

※TPTP(Hyades)をベースに構築されたツール

- スクリプト記録
～コーディング作業なしにスクリプトを生成
- スケジュール
～何ユーザーが、どのタイミングで、どんなシナリオを実行するかをGUIで設定
- 実行と分析
～高負荷時、アプリケーションがどう振る舞うかをテスト結果レポートから分析

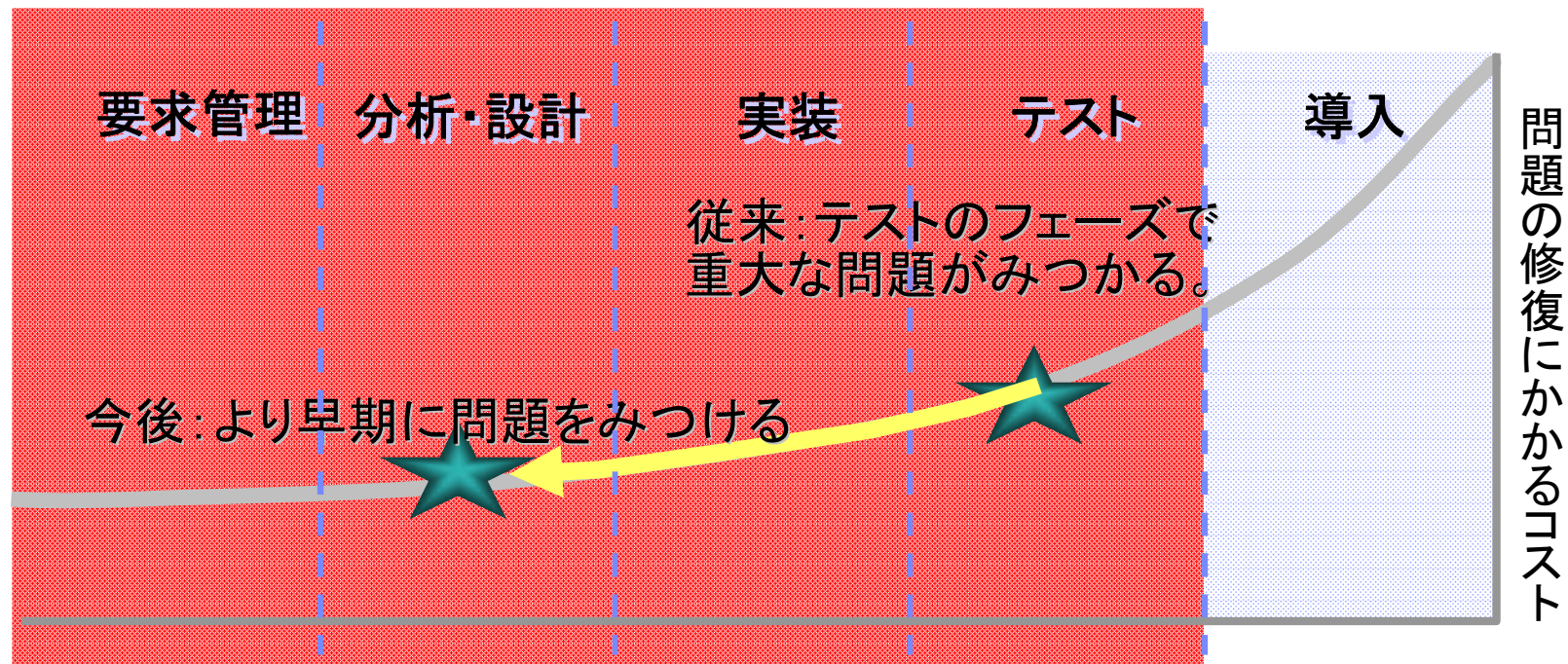
Rational品質管理製品群は Eclipse & TPTP をどのように活かしているのか

- 開発環境を統合するベースとしてEclipse & TPTPを活用
 - 各製品を統合するベースとなっている
 - ツールをまたがり共通の枠組み・共通のGUIを持つ
 - 開発における様々な作業を、統一された環境で実行できる
 - オープンソースや他プラグインとも組み合わせ可能である

アジェンダ

- Eclipse & TPTPの基礎
 - ～Eclipseとは
 - ～Eclipse TPTPとは
- Rational品質管理ポートフォリオ
- Rational品質管理製品を用いた開発 & テストイメージ
- (参考) Rational品質管理のアプローチ
- まとめ

(参考) Rationalの品質管理アプローチ ～”テストだけ”では足りない、“テストから”では遅い



- より上流工程から品質を意識したプロセスを実践する
→「上流工程で獲得された品質を保証する活動」がテスト
- より早期のリスク発見

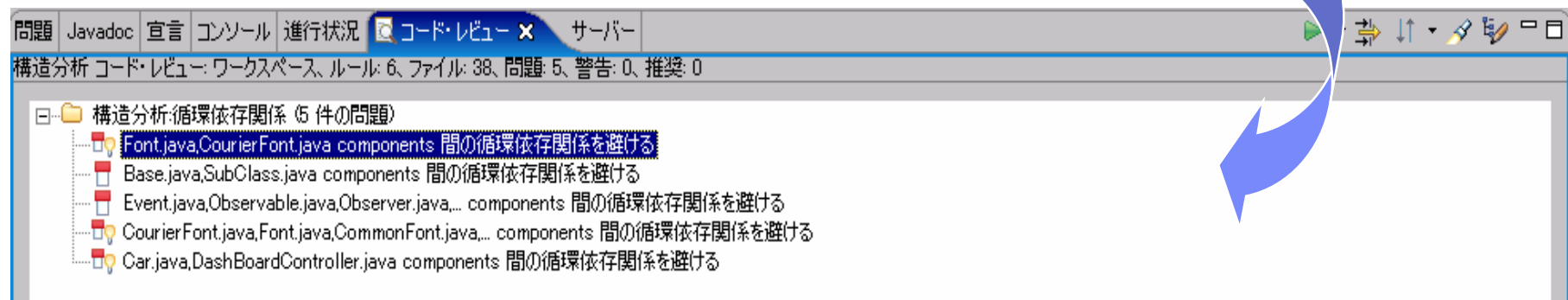
(参考) 構造分析 <RSA>



Rational Software Architect

- IBM Rational Software Architect (RADの上位製品、以下RSA) のみの分析機能
ハブ、ブレイカブル、循環依存関係の6ルール

- ☐ ☒ 構造分析 (6 件のルール、6 件が使用可能)
 - ☐ ☒ ハブ (2 件のルール、2 件が使用可能)
 - ☒ components が 10 以上のほかの components に依存しないようにし、10 以上の components を依存として持たないようにする
 - ☒ packages が 10 以上のほかの packages に依存しないようにし、10 以上の packages を依存として持たないようにする
 - ☐ ☒ ブレイカブル (2 件のルール、2 件が使用可能)
 - ☒ components が 10 以上のほかの components に依存しないようにする
 - ☒ packages が 10 以上のほかの packages に依存しないようにする
 - ☐ ☒ 循環依存関係 (2 件のルール、2 件が使用可能)
 - ☒ components 間の循環依存関係を避ける
 - ☒ packages 間の循環依存関係を避ける

**ON DEMAND BUSINESS™**

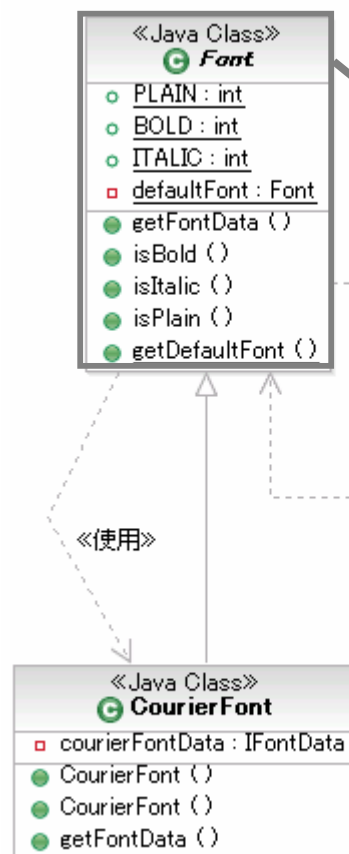
(参考) 循環依存関係 <RSA>



Rational Software Architect

構造分析: 循環依存関係 (5 件の問題)

Font.java, CourierFont.java components 間の循環依存関係を避ける



```

public abstract class Font {

    public static final int PLAIN      = 0;
    public static final int BOLD      = 1;
    public static final int ITALIC    = 2;

    //...

    public abstract IFontData getFontData();

    //...

    public boolean isBold() {
        return ( ( getFontData().getStyle() & BOLD ) != 0 );
    }

    public boolean isItalic() {
        return ( ( getFontData().getStyle() & ITALIC ) != 0 );
    }

    public boolean isPlain() {
        return ( getFontData().getStyle() == 0 );
    }

    //...

    public static Font getDefaultFont() {
        return defaultFont;
    }

    private static final Font defaultFont = new CourierFont();
  
```

アウトライン コード・レビューの詳細 ようこそ

説明 ループの詳細 例 解決策

カテゴリ
構造分析: 循環依存関係

名前
コンポーネント間での循環依存関係を避ける

ロケーション
[/MyFont/Font.java:1](#)
[/MyFont/CourierFont.java:1](#)

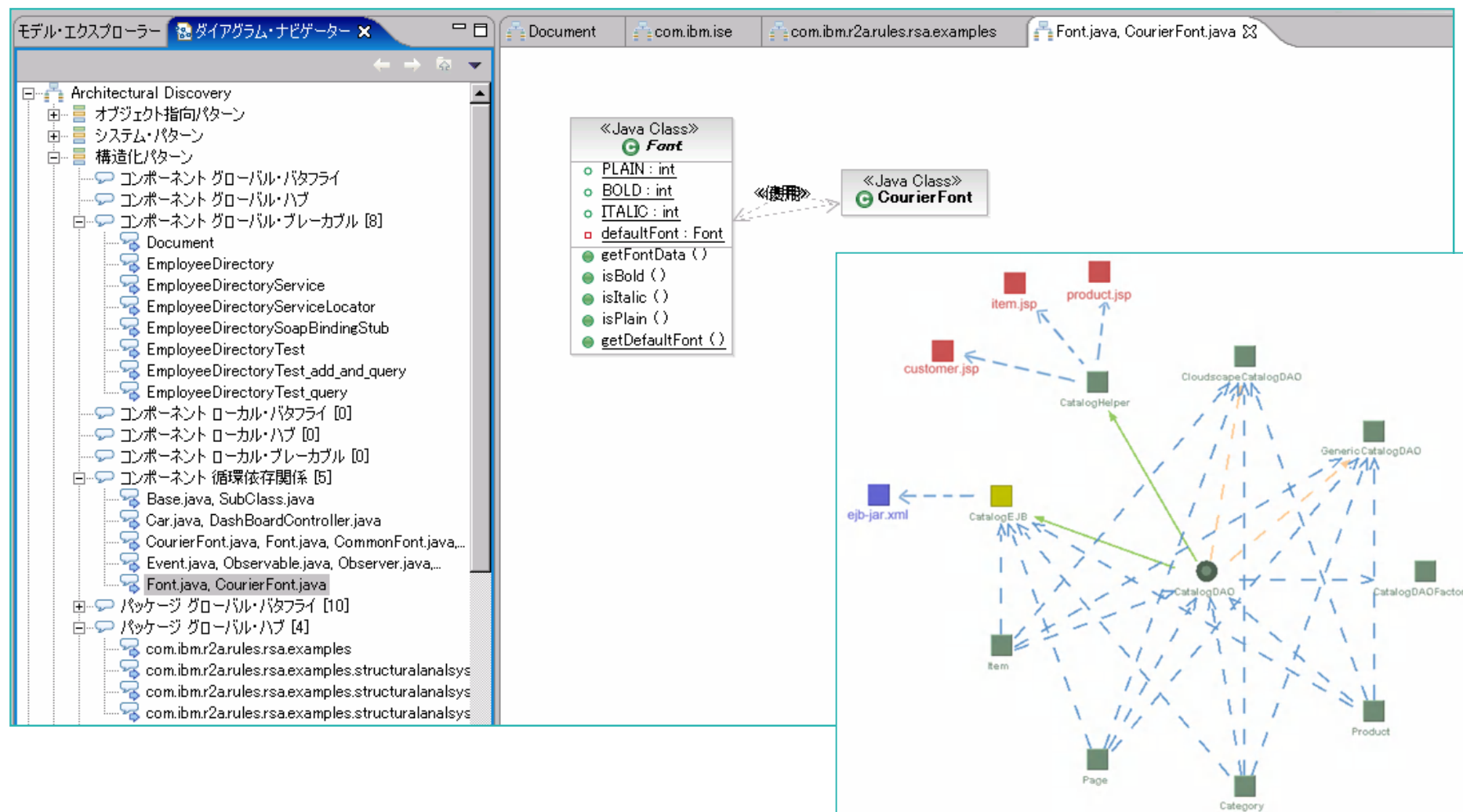
説明
コンポーネント間での循環依存関係は、重大な構造欠陥です。設計の観点から見た場合、複雑なシステムがどれもそうであるように、ソフトウェアには複数の責務からなる 1 つの階層が必要です。循環依存関係はこの階層を破壊し、システムのライフ・サイクルのすべての部分でさまざまな問題を発生させます。このような依存関係は、コードをわかりにくくすると同時に、コードの保守も困難にします。さらに重要な点としては、循環依存関係はテスト可能性、並行開発、および再利用の障害となります。多数の循環依存関係を持つ大規模ソフトウェアは、脆弱で不安定です。

(参考)アーキテクチャー・ディスカバリー <RSA>

■ 高レベルなソフトウェア可視化

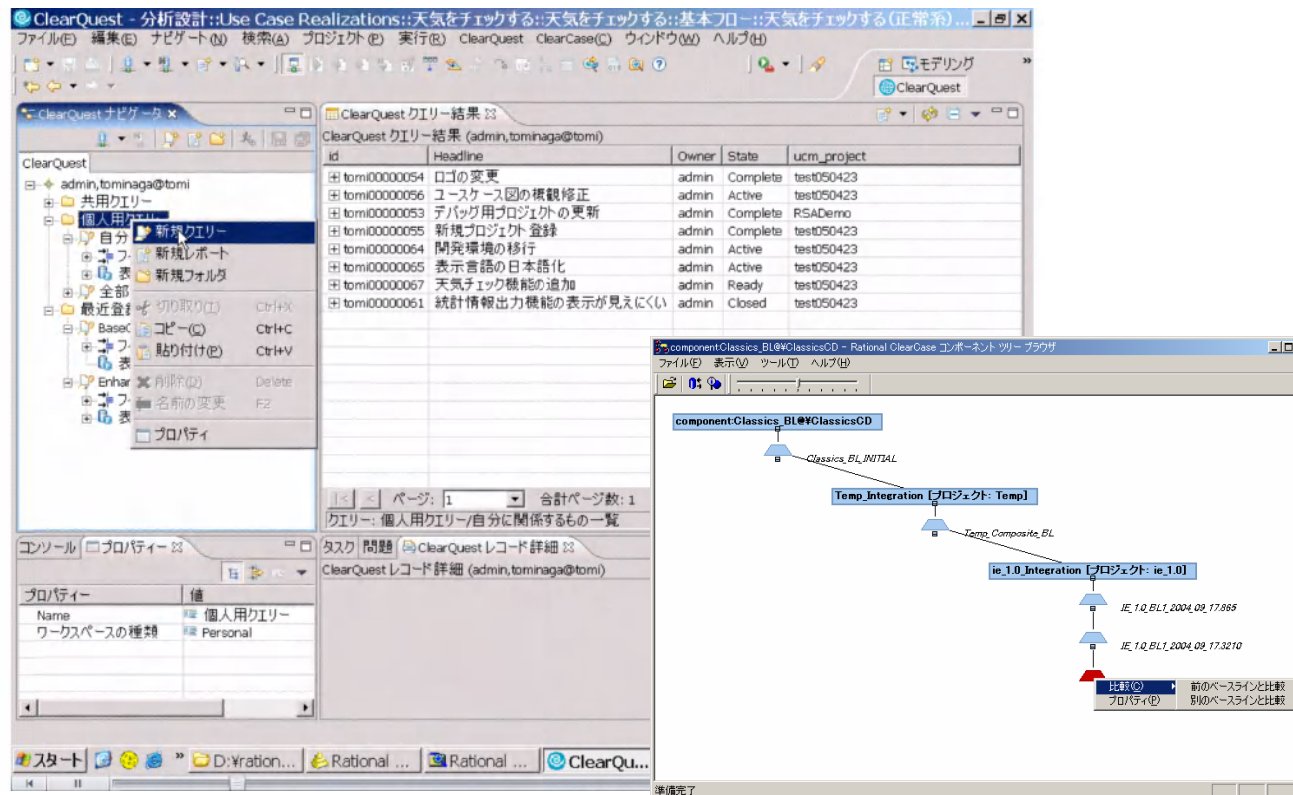


Rational Software Architect



(参考) 構成管理製品との統合

- 変更依頼(障害)管理やソフトウェア構成管理も広義の品質管理
- 変更依頼管理(ClearQuest)、構成管理(ClearCase)ともEclipse上で連携



アジェンダ

- Eclipse & TPTPの基礎
 - ～Eclipseとは
 - ～Eclipse TPTPとは
- Rational品質管理ポートフォリオ
- Rational品質管理製品を用いた開発 & テストイメージ
- (参考) Rational品質管理のアプローチ
- まとめ

まとめ

- Eclipse & TPTPはツール統合の核となるプラットフォームである
- Rational品質管理製品はEclipse & TPTPをプラットフォームとして構成されている
- Rational品質管理製品により、様々な作業を統一された環境で実施できる
- 品質管理は開発全行程に関わる
～Rationalの品質管理はテストツールだけではない



当資料は2006/1/31時点の情報に基づいて作成されていますが、事前の予告なく変更される場合があります。

- IBM、IBMロゴ、ON（ロゴ）DEMAND BUSINESSおよびRationalはIBM Corporationの商標
- "Microsoft" "Windows" および "Windows"ロゴは Microsoft Corporationの米国およびその他の国における商標。
- "Java" およびすべてのJava関連の商標およびロゴは Sun Microsystems, Inc.の米国およびその他の国における商標。
- 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。

