杉田 一之- 日本アイ･ビー･エム株式会社 ソフトウェア開発研究所 ラショナル･サービス　ITアーキテクト

15 April 2011

プロジェクト現場から見た現状分析と課題解決のヒント

目次

1. プロジェクトにおけるCOBOL開発

今まで担当したCOBOL開発プロジェクトの特徴は以下のとおりです。

- **COBOL開発の対象領域**
  - COBOL開発の主な対象領域は、ビジネス・アプリケーション
- **COBOLアプリケーションの実行環境**
  - System z, z/OSメインフレーム上で稼動する、COBOLアプリケーションが多い
- **COBOLアプリケーションの対象**
  - 主に基幹系システムの構築実施
- **COBOLアプリケーションの開発規模**
  - 基幹系システムが多いことから、大規模な開発が多い
- **COBOLアプリケーションの開発拠点**
  - お客様先の国内開発サイトと、海外サイトを含む複数拠点に跨る開発実施

・z/OS
・言語：COBOL
・ミドルウェア：CICSやIMS
・DB：DB2やIMS DB

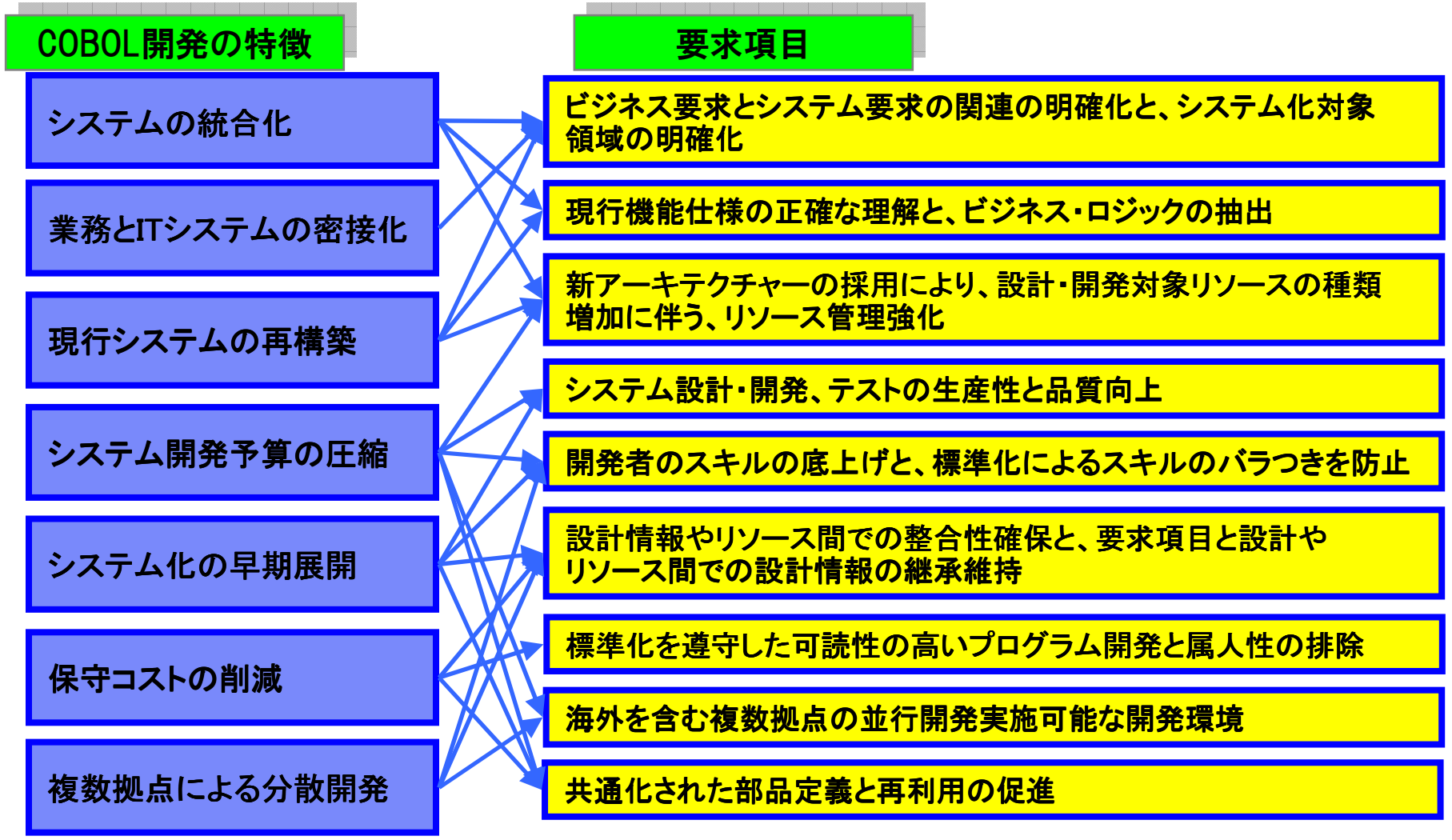
2. 最近のCOBOL開発プロジェクトの特徴

**COBOL開発プロジェクトの最近の傾向としては以下のような特徴があり、
開発実施上の複雑性が増し、様々な考慮が必要となってきていると考えています。**

- **システムの統合化**
  - 企業間の業務提携や、合併などによる基幹システムの統合化開発
- **業務とITシステムの密接化**
  - 業務とITシステムの関係がますます密接化することにより要件の複雑化
- **老朽化した現行システムの再構築**
  - 現行システム上での普遍なビジネス・ロジックの継承維持
- **システム開発予算の圧縮**
  - 開発コスト削減の中で、システム開発の生産性向上が必須
- **システム化の早期展開**
  - 業務側に対し出来るだけ早期にメリット享受ができるよう、短期間での開発
- **保守コストの削減**
  - 保守性を高めるための新しいアーキテクチャーの採用
- **複数拠点による分散開発**
  - 開発予算圧縮で、海外オフショア活用による分散開発環境の必要性
- **バッチシステムをオンライン化**
  - サービス向上のため、バッチからオンライン化による再構築

3. 最近のCOBOL開発プロジェクトの特徴を満たすための要求項目

開発実施上の複雑性が増し、こうした状況の中でCOBOL開発を実施する上での要求項目を、下記のとおり整理してみました。

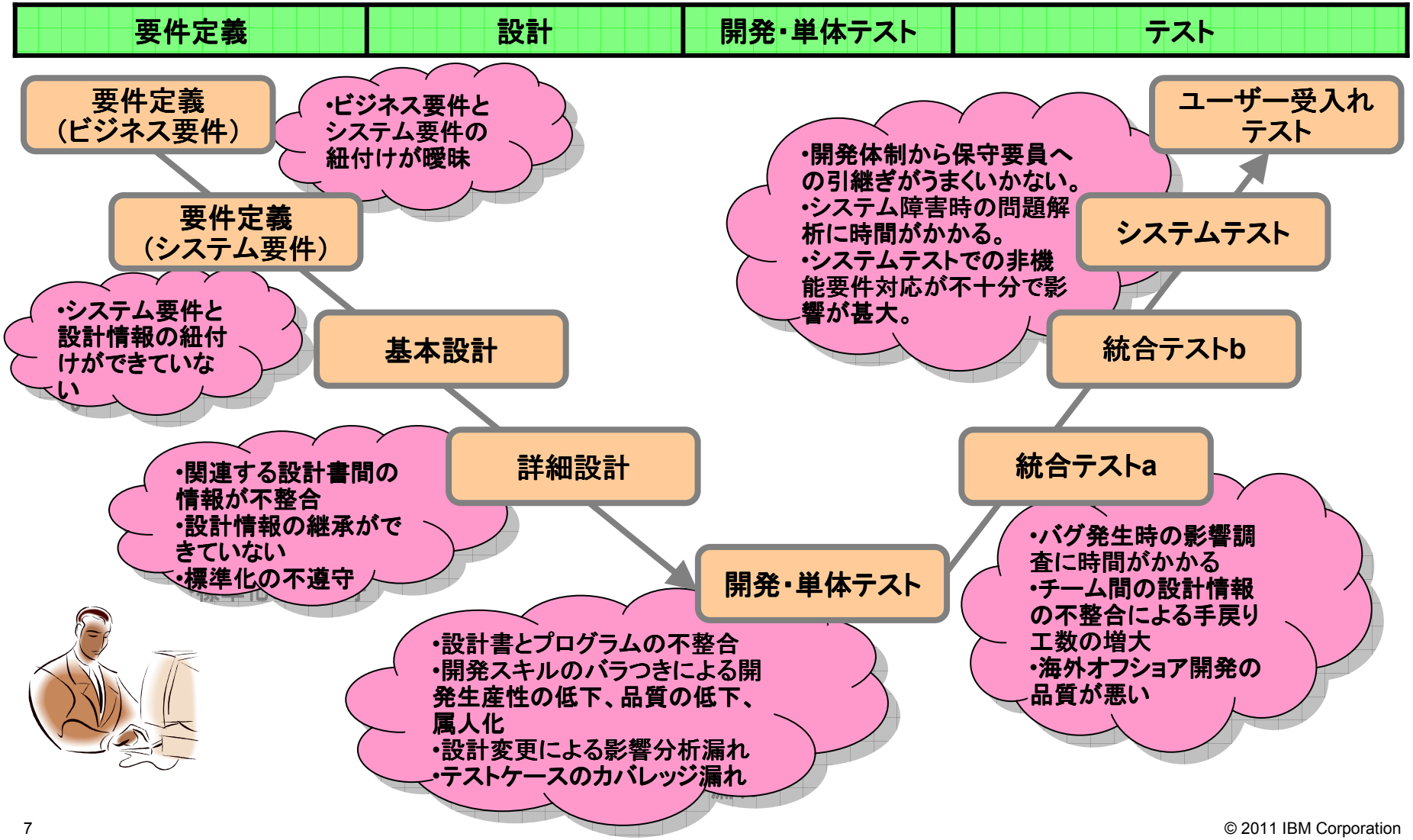


4. 要求項目を実現するための問題点

要求項目を満たす上で、現実的に下記のような幾つかの問題点が出てくると考えています。

5. 開発工程全般からみた問題点

要件定義からテスト局面までのように、開発工程全般から問題点を捉えた場合、
以下のように考えています。



6. COBOL保守現場での問題点

COBOL保守現場からは、以下のような問題点が掲げられています。

- 標準化ガイドはあるが徹底できていない。
- 仕様書の記述レベルが統一されていない。
- 仕様書が陳腐化してしまっているか、仕様書がない。
- データ項目の一元管理ができていない。
- データ項目の異音同義や同音異義語がある。
- 開発ツールが使いこなせない。
- 開発ツールが充実していない。
- 影響分析を実施するのに時間がかかる。
- 開発やテスト環境用のホストリソースが逼迫している。

- 度重なる機能追加の結果として、システムが肥大化しシステムの全体像が理解できない。
- ロジックの意図がわからない（設計思想がわからなくてっている）。
- システム全体のフローが把握できていない。
- システム毎にスキルが属人化していて、知識の共有ができていない。
- 機能が部品化されていない。
- 同じロジックが複数プログラムに散在し重複してしまっていて保守性が悪い。
- 不要ロジック（デッド・コード）が存在する。
- 文書ファイルで大量に設計書を作成したが、設計情報間の連携がないために保守が困難。

7. 問題の傾向分析と問題を解決するためのキーワード

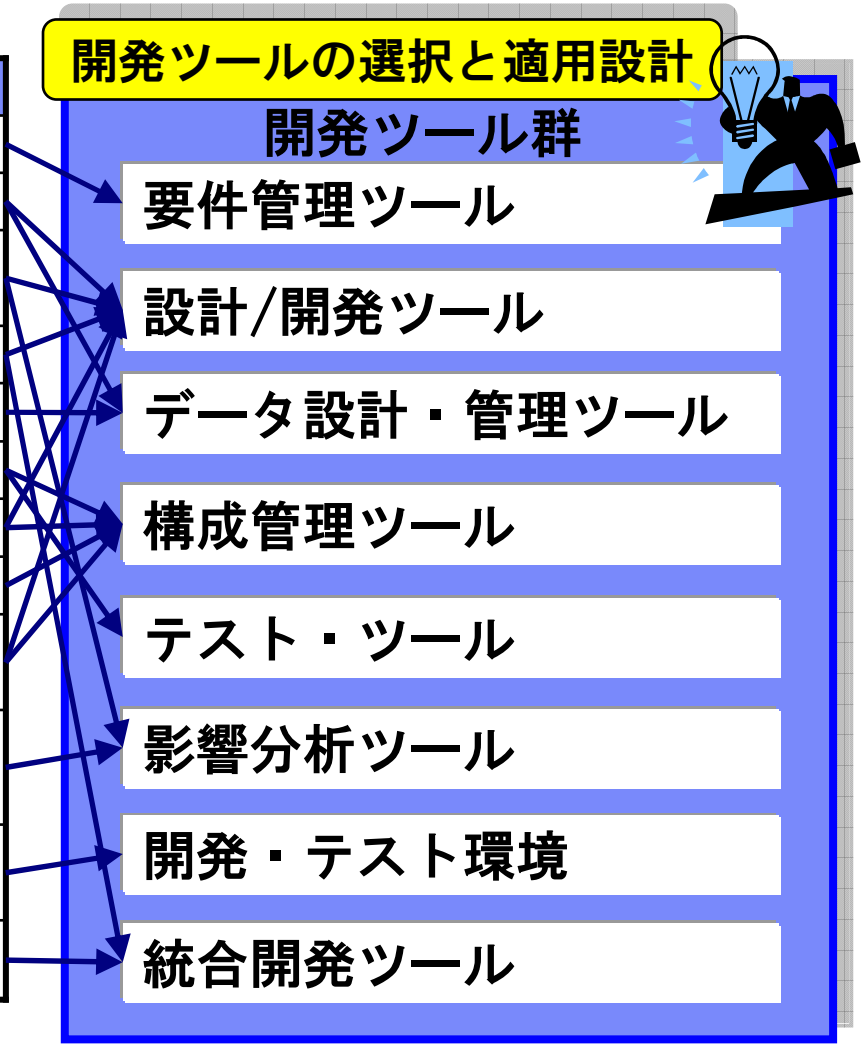
今まで列挙した問題の傾向をまとめ、問題解決のキーワードを以下のように定義しました。

·問題のまとめ	•問題解決のキーワード	·カテゴリー
要件が曖昧で、不確実性が残る場合があり、テスト時における手戻りが発生。	ビジネス要件とシステム要件の紐付け	要件管理
標準化ガイドはあっても徹底できていないことによる、作成物の粒度のバラつきが発生し、品質確保が困難。	標準化を遵守するためのワークフローの確立	標準化
仕様書の陳腐化や仕様書が存在しないなどの影響による、現行機能の理解に費やす時間が増大。	現行リソースの分析と現行リソースからの仕様書作成	現行リソース解析
開発スキルのバラつきが生じ、プロジェクトの遅延や品質が低下。	開発プロセスの定義と扱いやすいツールの適用	スキル平準化
データ項目の異音同義、同音異義語が発生することにより、システムの保守性が低下。	データ項目の一元管理とディクショナリー化	データ項目管理
テストケースのカバレッジ漏れ等によるテスト品質の低下。	テスト結果の可視化とテストデータの管理	テスト管理
同じロジックが複数プログラムに散在し、冗長化することで、開発、テスト工数の増加と、保守性の低下。	共通ロジックの部品化推進と管理	部品化/共通化
新アーキテクチャーの採用による開発対象のリソースの種類が増加し、リソース管理が困難。	仕様書やリソースの一元管理	リソース一元管理
仕様書とリソースの一元管理ができていないことにより、整合性維持が困難となり、品質が低下。	仕様書やリソースの一元管理、トレーサビリティの確保	• 整合性維持 · トレーサビリティ
バグ発生時や設計変更による影響分析漏れが発生し、テスト時に手戻りが発生。	影響分析の機械化	影響分析
バグ発生時や設計変更による影響調査に時間がかかり、タイムリーな対応が不可。	影響分析の結果に対するリポジトリー化	
開発/テスト環境のホストリソースが逼迫し、開発/テスト時の運用統制による開発生産性の低下。	開発/テスト環境の増強かメインフレーム以外の開発/テスト環境構築	開発/テスト環境
ツール適用の部分的利用により、設計・開発作業の連続性が失われ、生産性が低下する。	操作の統一性とリソース連携された開発ツール	統合開発ツール

8. 問題解決のための開発ツール群

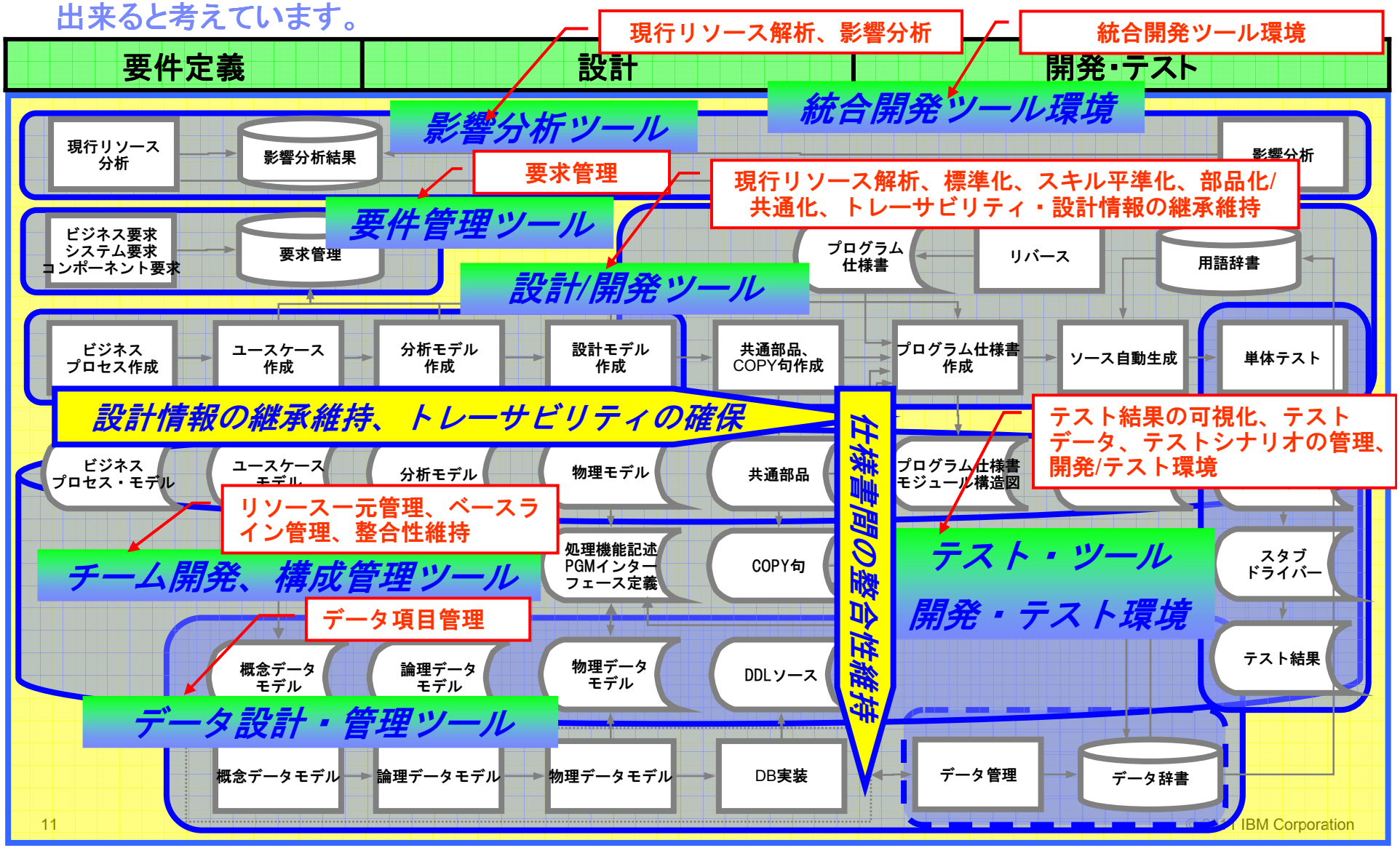
開発実施上での問題解決には、ツール適用は必須と考えます。ただし、ツール導入は問題解決のきっかけとなりますが、十分条件ではありません。特に大規模開発の場合、ツール使用の開発手順や適用設計が重要となります。また、予算に応じ必要なツール選択をするのが現実的です。

問題解決のキーワード	カテゴリー
ビジネス要件とシステム要件の紐付け	要件管理
標準化を遵守するためのワークフローの確立	標準化
現行リソースの分析と現行リソースからの仕様書作成	現行リソース解析
開発プロセスの定義と扱いやすいツールの適用	スキル平準化
データ項目の一元管理とディクショナリー化	データ項目管理
テスト結果の可視化とテストデータの管理	テスト実施/管理
共通ロジックの部品化推進と管理	部品化/共通化
仕様書やリソースの一元管理	リソース一元管理
仕様書やリソースの一元管理、トレーサビリティの確保	•整合性維持 ·トレーサビリティ
影響分析の機械化	影響分析
影響分析の結果に対するリポジトリー化	
開発/テスト環境の増強かメインフレーム以外の開発/テスト環境構築	開発/テスト環境
操作の統一性とリソース連携された開発ツール	統合開発ツール



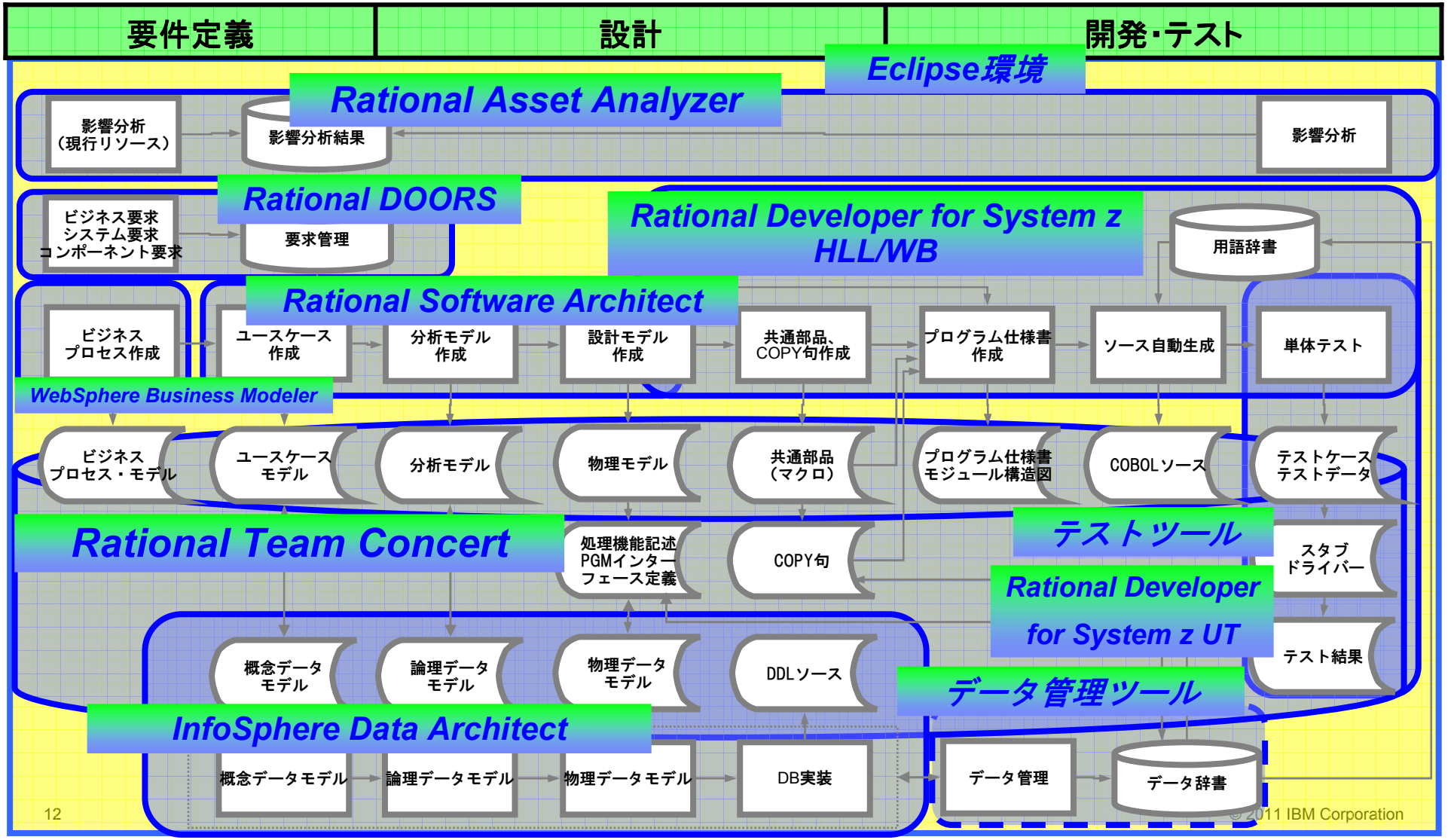
9. COBOL開発のための統合開発ツール環境ソリューション

下記のような開発環境ソリューションを構築することで、COBOL開発の問題を軽減することが出来ると考えています。



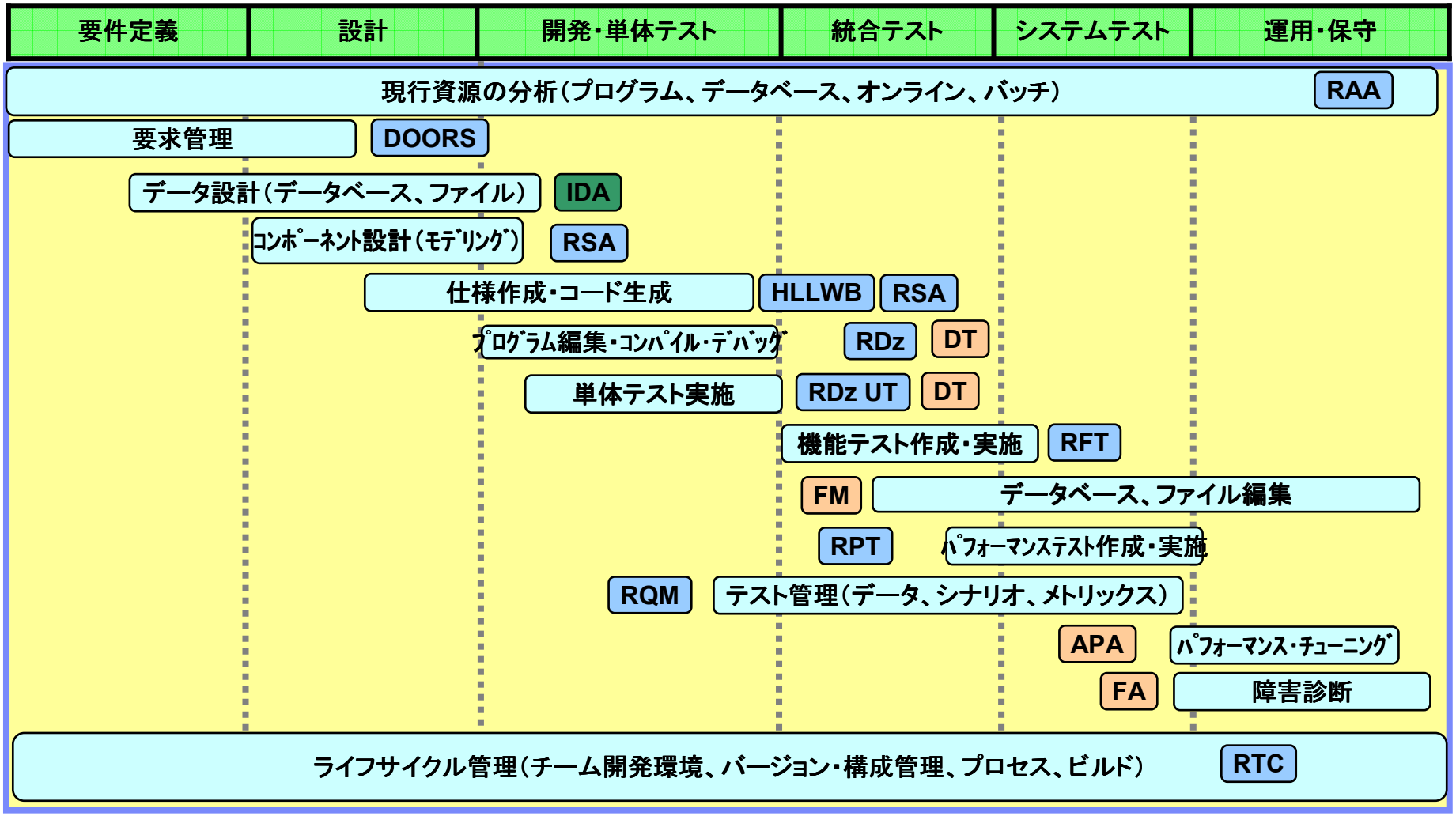
10. COBOL開発を実現するIBMの開発ツール

IBMの開発ツールマッピングすると下記のようになり、開発ツール間の連携を行ったツールの適用設計と導入・運用を行うことで、COBOL開発の生産性と品質向上に寄与すると考えています。



11. 開発工程と開発ツール

開発工程と各局面毎での開発ツールの適用範囲は以下のようになります。すべての局面を
フルライフ・サイクルで支援する管理系ツールを土台にして、それぞれの開発ツールがあります。



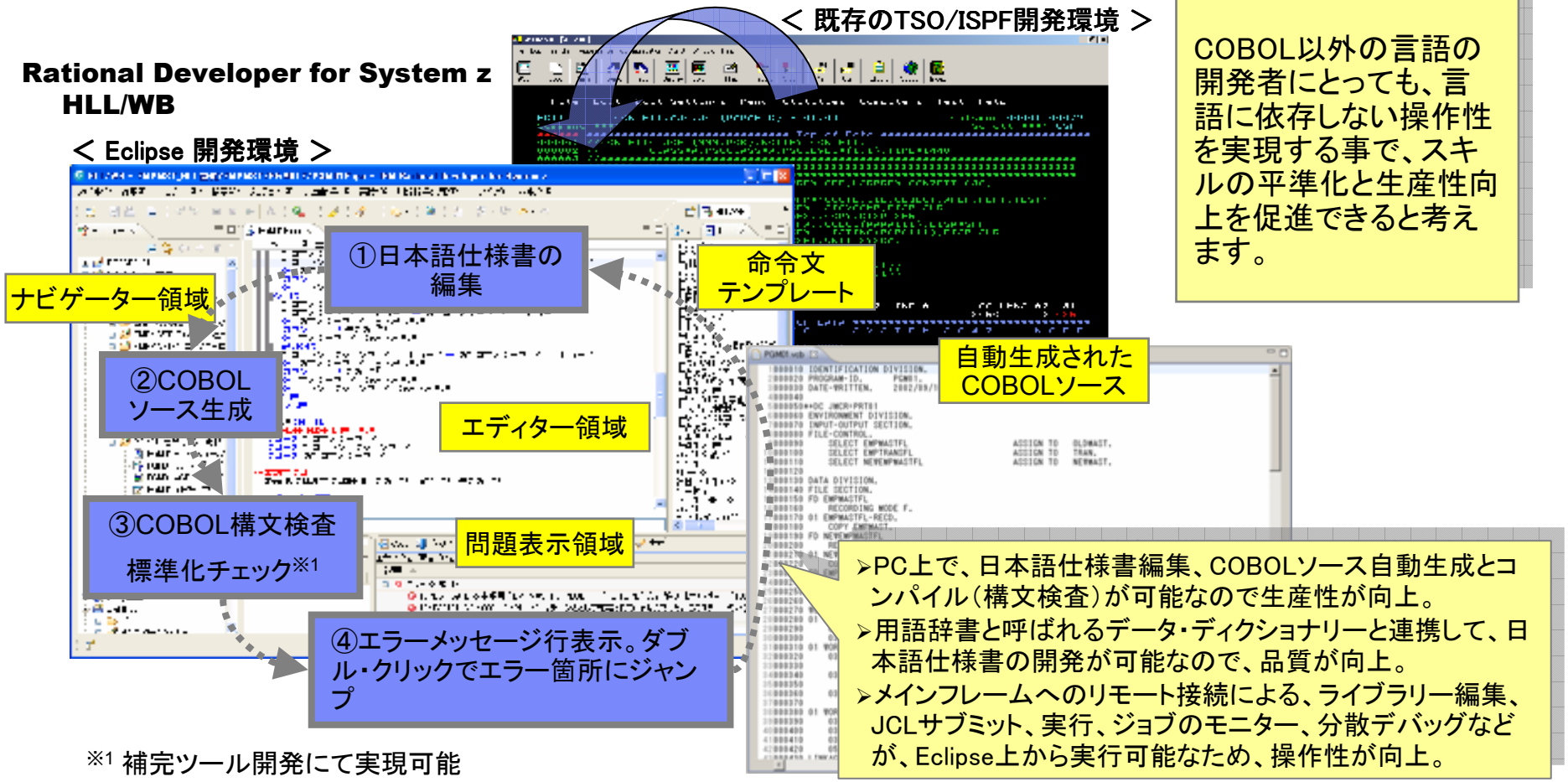
12. 開発ツール一覧（参考資料）

開発ツール製品の名称と機能は下記のとおりです。

カテゴリー	機能	製品名
影響分析	既存アプリケーション資産の棚卸・可視化、及び変更に対する影響分析（Eclipse対応）	Rational Asset Analyzer（RAA）
要件管理	要求内容を、設計項目および他の要求とリンクさせ管理と追跡が可能。また、RQM（Rational Quality Manager)と連携することで、テスト計画、テスト・ケースとの紐づけが可能	Rational DOORS
設計（モデリング）	UMLによるモデリングから、JAVA, J2EE, Web Service, SOA, C/C++アプリケーションのコード生成が可能（Eclipse対応）	Rational Software Architect（RSA）
設計／開発	日本語マクロ開発、用語辞書機能により、日本語仕様書からCOBOLやPL/Iのソースを生成、或いはリバースが可能。また、コードカバレッジの機能もあり（Eclipse対応）	Rational Developer for System z HLL/WB Edition（HLL/WB）
データ設計・管理	エンタープライズ・データモデリングから、データベース設計、及びデータ管理機能を提供（Eclipse対応）	InfoSphere Data Architect (IDA) RDz HLL/WB Edition（用語辞書）
チーム開発・構成管理	進捗管理、変更管理、構成管理など開発支援機能を一つのツールで一元化でき、チーム開発のための環境を提供する開発基盤（Jazz対応）	Rational Team Concert（RTC）
テスト	ファンクション・リグレッションテスト（RFT）、ストレス・パフォーマンステスト（RPT）を自動化（Eclipse対応）、またテスト・品質管理（RQM）の機能を提供（Jazz対応）	Rational Functional Tester（RFT）/ Rational Performance Tester（RPT）/ Rational Quality Manager（RQM）
開発・テスト環境	PC上のLinux環境でz/OSをエミュレートする機能を提供して、単体テストまでを実施することが可能	Rational Developer for System z Unit Test Feature（RDz UT）
統合開発環境	Eclipseの統合開発環境（IDE）を製品として提供	Rational Developer for System z（RDz）
問題判別	ホスト環境の開発・運用・保守局面で、生産性を向上させるためのデバッギング、簡易なファイル操作・テストデータ生成、障害解析、パフォーマンス・チューニングの機能などを提供する製品群	Problem Determination Tools（PD Tools） - Debug Tool (DT) / File Manager（FM） / Fault Analyzer (FA) / Application Performance Analyzer (APA)

13. 開発ツールを利用した感想

従来のコマンド実行を主体としたTSO/ISPFと比較して、グラフィカルなユーザー・インターフェイスを提供するEclipseの共通基盤上で言語に依存しない開発環境を実現でき、日本語仕様書からCOBOLソース自動生成機能等を利用することにより、COBOL言語に精通していない開発要員でも、生産性や品質を維持しながら開発が可能という事を実感しています。



COBOL以外の言語の開発者にとっても、言語に依存しない操作性を実現する事で、スキルの平準化と生産性向上を促進できると考えます。

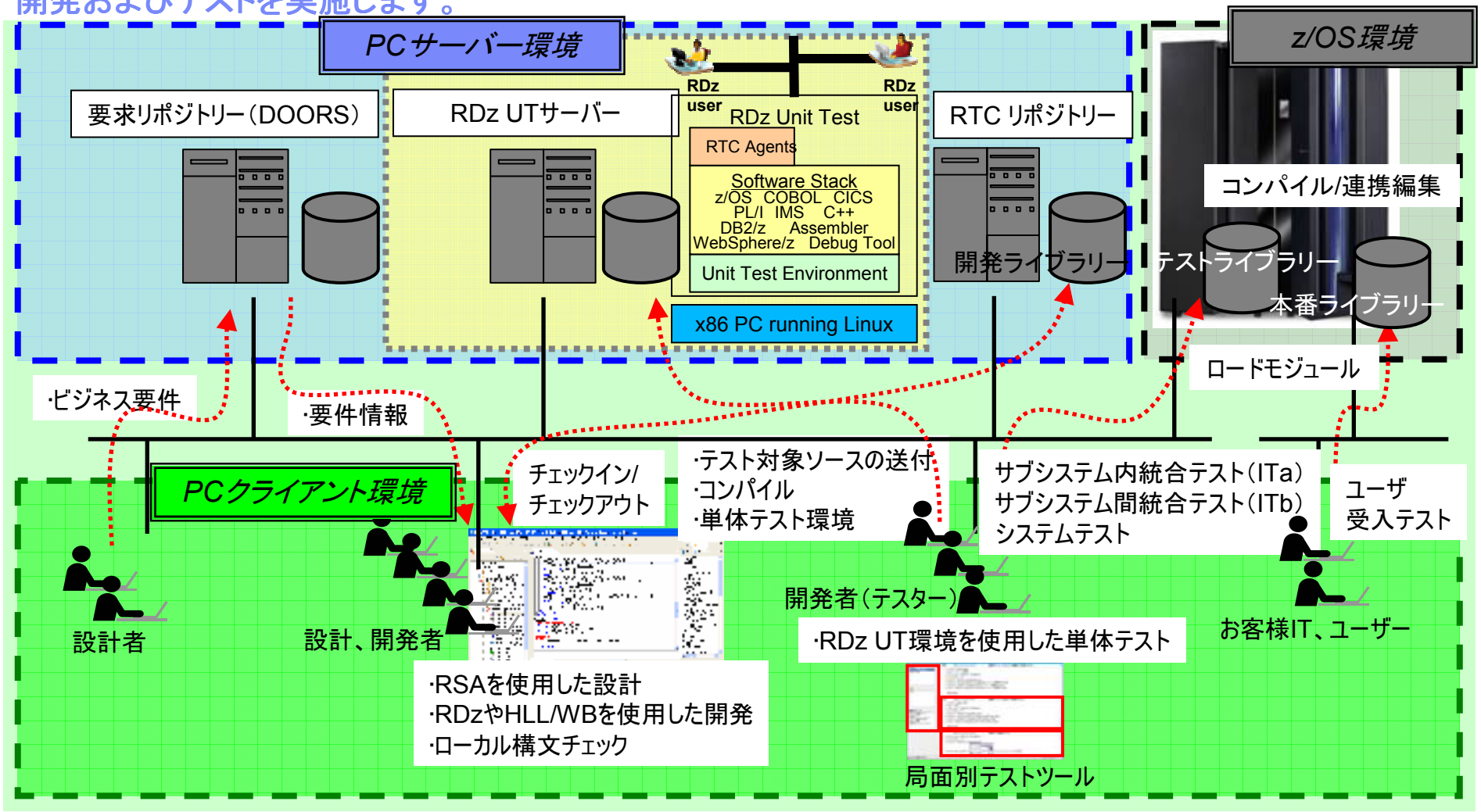
- PC上で、日本語仕様書編集、COBOLソース自動生成とコンパイル（構文検査）が可能なので生産性が向上。
- 用語辞書と呼ばれるデータ・ディクショナリーと連携して、日本語仕様書の開発が可能なので、品質が向上。
- メインフレームへのリモート接続による、ライブラリー編集、JCLサブミット、実行、ジョブのモニター、分散デバッグなどが、Eclipse上から実行可能なため、操作性が向上。

※1 補完ツール開発にて実現可能

※Rational Developer for System z HLL/WBについて：http://www.ibm.com/software/jp/rational/solutions/em/hllwb/

14. 開発/テスト環境例

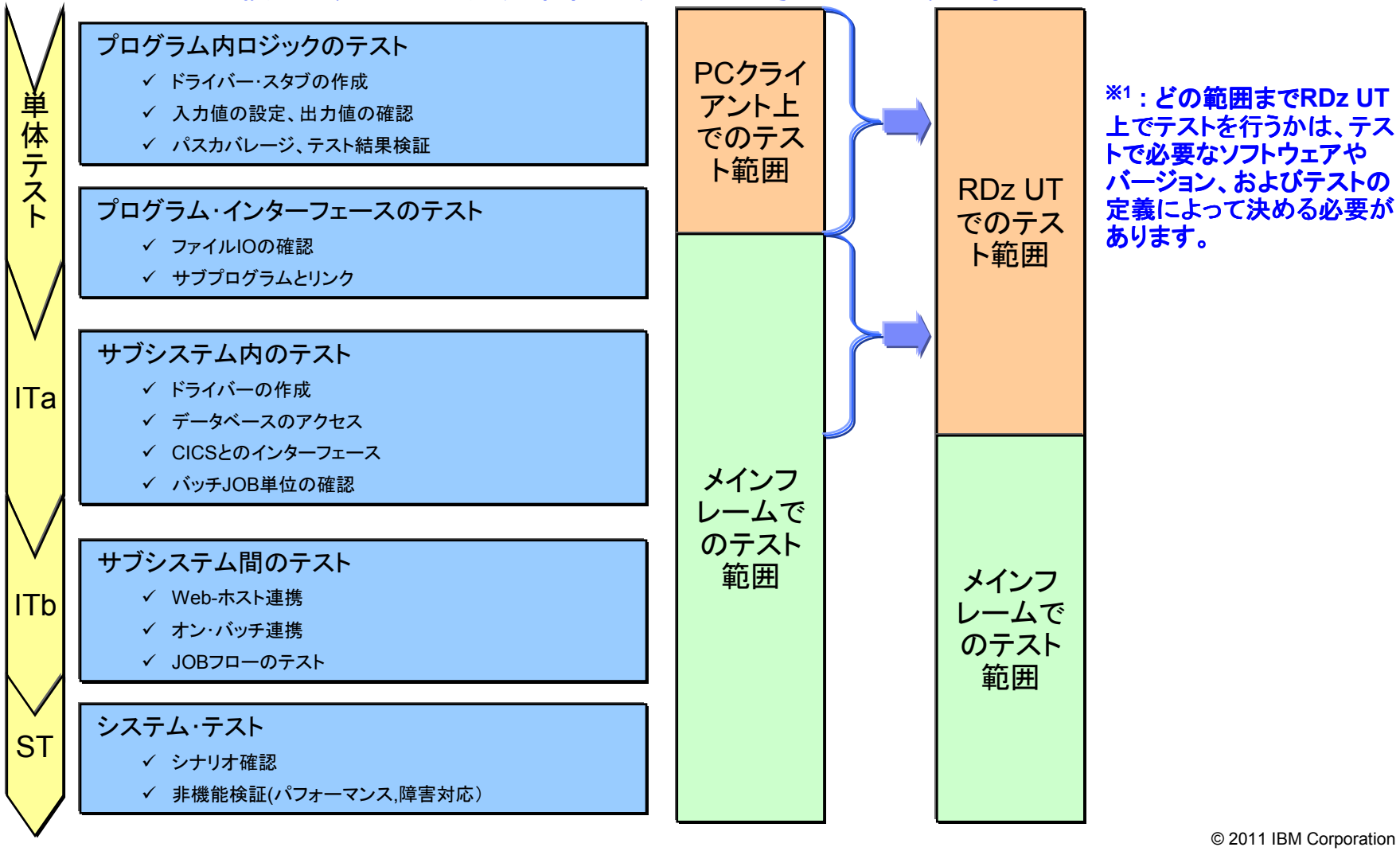
要件定義、設計から開発、テストまで実施した場合の開発/テスト環境の構築例です。基本的に、メインフレームのCOBOL開発でも、PCクライアント、PCサーバーおよびz/OS環境を組み合わせて、設計、開発およびテストを実施します。



※RTC (Rational Team Concert)について: http://www.ibm.com/software/jp/rational/products/scm/rtc/

15. RDz UTを使用した場合のテスト適用範囲例

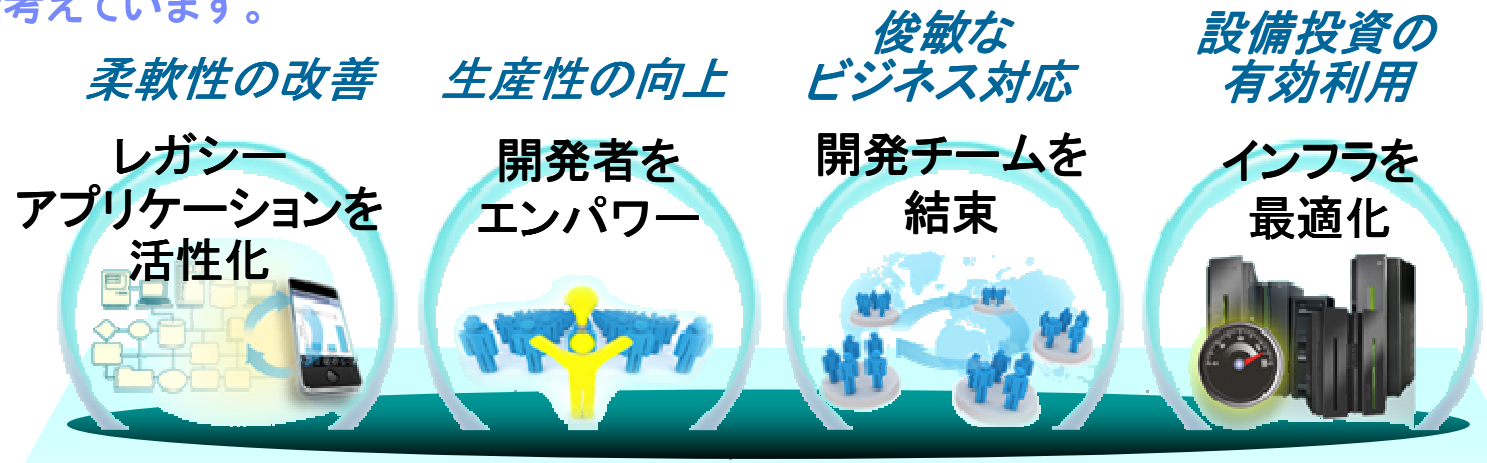
RDz UT (Rational Developer for System z Unit Test Feature)では、z/OSがPCサーバー上でエミュレーション可能となり、その上でDB2やCICS等のミドルウェアが稼動できるので、従来のPC環境によるテストと比較して、テストの適用範囲が広がると考えています※1。



※1：どの範囲までRDz UT上でテストを行うかは、テストで必要なソフトウェアやバージョン、およびテストの定義によって決める必要があります。

16. IBM Rationalのエンタープライズ・モダナイゼーション

開発現場がかかえる様々な問題を解決するヒントが、エンタープライズ・モダナイゼーションにあると考えています。



- アプリケーション・ポートフォリオを再活性し、*柔軟性を改善*
- 最新のスキルによりイノベーションを加速し、*生産性を向上*
- 組織のサイロを橋渡しして、*俊敏なビジネス対応*
- アプリケーションのインフラを最適化し、*設備投資を有効利用*

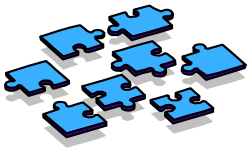
Rational Software Delivery Platform *powered by* Jazz

17. まとめ

本日はCOBOL開発プロジェクト現場が抱える様々な問題と、それらの問題を解決するためのヒントを述べてきましたが、この発表が皆様の今後のCOBOL開発を実施する上で一助となれば幸いです。

基本的に、プロジェクト活動の主体は人であり、人が考えたものがシステム仕様となりシステム稼動とつながります。ツールは、人が開発を行うための道具であり手段なので、設計・開発を行う前にツールをどのように適用するか、どのように運用していくのかのアセスを十分に行い、標準化することが重要と思います。下記にキーポイントをまとめます。

- 大規模開発ではツールを活用する
  - 開発効率よく、品質よく開発を実施するためには、ツール導入は必須
- ツール活用には以下の適用設計、標準化がポイント
  - プロジェクト体制、ロール・役割に応じた開発手順
  - 開発手順に適合したツールの適用設計
  - ツールによる作成物と、ツール以外の作成物との整合性確保
  - 開発環境における障害発生時の運用設計（シンプルな開発環境構築）
- 過去事例の調査とツール有識者へのヒヤリング



ワークショップ、セッション、および資料は、IBMまたはセッション発表者によって準備され、それぞれ独自の見解を反映したものです。それらは情報提供の目的のみで提供されており、いかなる参加者に対しても法律的またはその他の指導や助言を意図したものではなく、またそのような結果を生むものでもありません。本プレゼンテーションに含まれている情報については、完全性と正確性を帰するよう努力しましたが、「現状のまま」提供され、明示または暗示にかかわらずいかなる保証も伴わないものとします。本プレゼンテーションまたはその他の資料の使用によって、あるいはその他の関連によって、いかなる損害が生じた場合も、IBMは責任を負わないものとします。 本プレゼンテーションに含まれている内容は、IBMまたはそのサプライヤーやライセンス交付者からいかなる保証または表明を引きだすことを意図したものでも、IBMソフトウェアの使用を規定する適用ライセンス契約の条項を変更することを意図したものでもなく、またそのような結果を生むものでもありません。

本プレゼンテーションでIBM製品、プログラム、またはサービスに言及していても、IBMが営業活動を行っているすべての国でそれらが使用可能であることを暗示するものではありません。本プレゼンテーションで言及している製品リリース日付や製品機能は、市場機会またはその他の要因に基づいてIBM独自の決定権をもっていつでも変更できるものとし、いかなる方法においても将来の製品または機能が使用可能になると確約することを意図したものではありません。本資料に含まれている内容は、参加者が開始する活動によって特定の販売、売上高の向上、またはその他の結果が生じると述べる、または暗示することを意図したものでも、またそのような結果を生むものでもありません。
パフォーマンスは、管理された環境において標準的なIBMベンチマークを使用した測定と予測に基づいています。ユーザーが経験する実際のスループットやパフォーマンスは、ユーザーのジョブ・ストリームにおけるマルチプログラミングの量、入出力構成、ストレージ構成、および処理されるワークロードなどの考慮事項を含む、数多くの要因に応じて変化します。したがって、個々のユーザーがここで述べられているものと同様の結果を得られると確約するものではありません。

記述されているすべてのお客様事例は、それらのお客様がどのようにIBM製品を使用したか、またそれらのお客様が達成した結果の実例として示されたものです。実際の環境コストおよびパフォーマンス特性は、お客様ごとに異なる場合があります。

IBM、IBM ロゴ、ibm.com、RationalおよびDOORS.Infosphere,Jazz, Rational Team Concert, System z, z/OS は世界の多くの国で登録されたInternational Business Machines Corporationの商標です。

Javaは、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標。
Microsoft, Windows, Windows NT 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。
Linuxは、Linus Torvaldsの米国およびその他の国における商標。

他の製品名およびサービス名等は、それぞれIBMまたは各社の商標である場合があります。
現時点での IBM の商標リストについては、www.ibm.com/legal/copytrade.shtmlをご覧ください。
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。