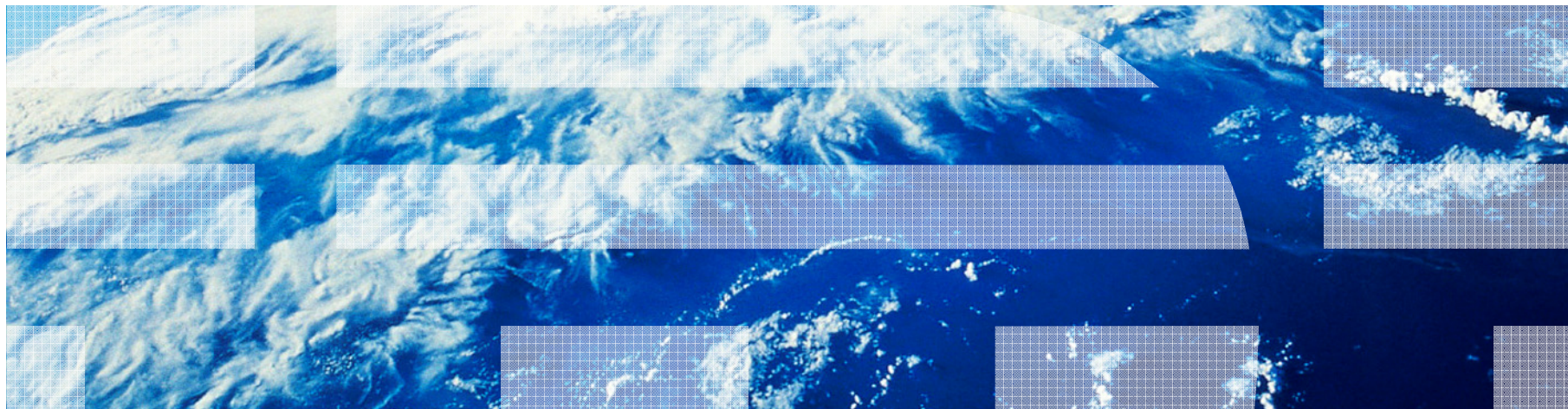


# オフショア利用による販売基幹システム開発での UML適用事例

日本アイ・ビー・エム株式会社  
グローバル・ビジネス・サービス事業 ITアーキテクト  
吉田 亮



## 本事例紹介の概要

- 社内フレームワークを利用したオフショア開発（Java によるカスタム・アプリケーション開発）を例に、どのように UML や UML 支援ツールを活用したかを具体的に紹介する。  
本事例における開発では UML を主に次のように活用している。

- ✓ UML が有効である場面においては、UML ツールや社内資産も活用し、オフショア先に仕様や範囲を伝えた
- ✓ UML でカバーできない部分においては事前定義した補足資料を作成し、それらをオフショア先と共有した
- ✓ オフショア先に開発依頼する際の進捗管理や Q&A 管理等の管理作業は、事前に洗い出し、ツールを用いて情報共有できるよう対策した

## 本事例紹介の構成

1. 事例として取りあげる開発プロジェクトの概要
2. プロジェクトにおける UML 利用方法
3. オフショア開発としての協業
4. IBM 製品 (Rational) や社内アセットによる開発
5. 併用した IBM 社内アセット
  
6. 具体的な UML 活用工程 — なぜ UML を用いたか
7. UML 活用 — 日中それぞれの技術者と作業分担
8. UML 活用 — 成功箇所、工夫箇所
9. UML 活用 — 利用ダイアグラムと補完作成資料
10. UML 活用 — 開発ツール
  
11. UML 利用の効果と課題 まとめ



IBM, Rationalは米国及びその他の国におけるInternational Business Machines Corp.の商標です。他の製品名等は、それぞれIBMの商標である場合があります。

Javaは Sun Microsystems, Inc.の米国及びその他の国における商標です。

UML, モデル駆動開発はObject Management Group Inc.の商標又は登録商標です。

## 事例として取りあげる開発プロジェクトの概要

- 製造業のお客様における販売系基幹システム再構築 — Java 言語による
- 開発期間： 2007年7月 ~ 2008年12月
- 日本側にて要件定義～詳細設計と結合テスト以降を担当。オフショアにて製造工程(単体テスト含む)を担当
- 実施にあたっての当プロジェクト方針：
  - Rational ツールや社内アセット(フレームワークや成果物)利用を前提としたモデル・ベース開発を実施。開発工数の削減と品質の向上を実現する
  - 製造工程の約97%をオフショアに依頼。Rational ツールを用いた社内標準のプロジェクト管理手法を適用して、管理やコミュニケーションを効率よく実施する
  - 各作業において当プロジェクトの成功だけでなく、横展開可能なナレッジの抽出と蓄積に重点を置き、成果物の最適化を実施する

※オフショアではあるが、同じ IBM グループ内での開発依頼となるため、統一された IBM 社内ツールや手法がそのまま利用可能であることに留意

### ■ 開発規模：

このタイミングで日本からオフショアに引継ぎ↓

単位：人/月（人数は最大時）

	要件定義	基本設計 (アーキテクチャ 説明)	レビュー Q&A対応	詳細設計	製造(単体テスト含む)	結合テスト	受入テスト
弊社お客様							
日本アイ・ビー・エム			70			70	
オフショア					30~50		

## プロジェクトにおけるUML利用方法

- 当プロジェクト実施にあたって UML を使用したが、「オフショア開発向け UML 適用ガイドライン」そのものは参照していない
- 当プロジェクトに UML 技術者や UML を前提とする開発ツール（Rational）経験者らが多く参画していたため、ガイドに頼らず UML を用いて開発可能な素地が揃っていた。オフショア開発のために UML を採用したわけではない
- 使用した図とタイミング(要件定義～結合テスト)

要件定義/基本設計				
	図名	目的	作成者	効果
日本アイ・ビー・エム	クラス図	・お客様の要求の定義 ・システム(アプリケーション)のアーキテクチャ、構造と振る舞いの定義	日本アイ・ビー・エム	UMLを前提に開発ツール(Rational)を使用したことで、定義・設計の生産性が向上した。
	ユースケース図			
	シーケンス図			
	(ロバストネス図)			
詳細設計・製造・結合テスト				
	図名	目的	作成者	効果
日本アイ・ビー・エム	クラス図	アプリケーションの設計仕様を正確に伝えるため。	日本アイ・ビー・エム	シーケンス図に加え、メソッド記述書も定義して作成することで、メソッドでの処理内容が正確に伝わった。
	シーケンス図			
オフショア	クラス図			
	シーケンス図			

## オフショア開発としての協業 — 中国

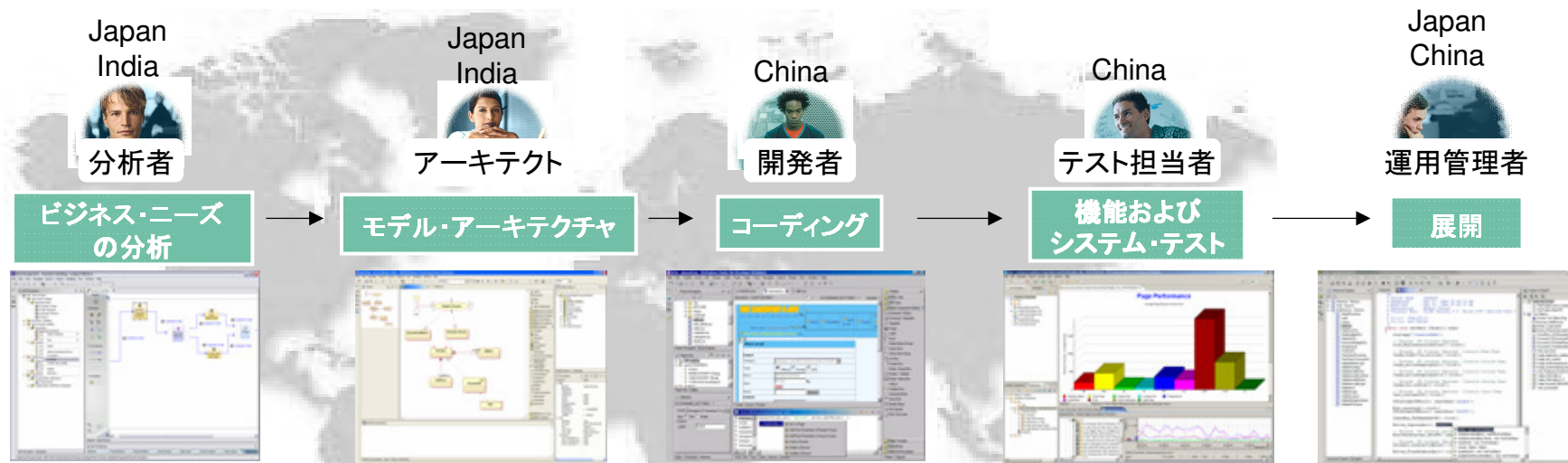
- IBM 中国グローバル・デリバリ・センターの拠点のひとつ、大連センターが製造工程を担当
- UML スキルについてはプロジェクト参画の必要要件とせず、教育などは実施していないものの、参画メンバーの多くは UML スキルを既に保有。当プロジェクト実施上、特に UML での仕様を読んで理解することに支障なし





## IBM製品 (Rational) や社内アセットによる開発 — プログラム開発面

- IBM グループ共通のツール & アセットによる効率的な協業:  
IBM オフショア開発では、世界各国の拠点において、IBM グループ共通の開発ツール & アセットを活用することにより、効率的な協業を実現し、高品質な優れた製造工程サービスを確保



### IBMのオフショア開発

世界中に展開されている IBM のグローバル・デリバリー・センターは、IBM グループとして、共通のツール・アセット・方法論・用語・インフラを整備。このため、各国のグローバル・デリバリー・センターのメンバーがどのようにチームを組んでも、開発手法、ツール、用語などが統一されているため、円滑に効率的な協業が可能。  
これにより、グローバル・レベルで最適化されたひとつのチームとして、IBM の優れたサービス提供がオフショア開発でも可能になっている。

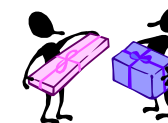
## 併用したIBM社内アセット

- オフショア開発チームと協業する前に、社内経験から既に明らかに問題となっているところを整理。それらに対してどのような施策をとればよいか検討した上で、オフショア協業開始前に対応を行った

- プログラム開発面

UML を使って設計、コーディングをする上では共通開発ツールとして Rational Software Architect を利用

- 既存 IP アセットは日本側で開発されたプログラム部品などが主
- UML によるメタ・モデルなどはプロジェクト(お客様要件)により異なり、再利用度は低い



- プロジェクト管理面

日中間あるいは開発チーム・メンバー間で情報をやりとりする、共有する上でのひとつの活用例

- オフショア・チームとの変更管理、進捗状況管理



Rational ClearQuest (カスタマイズにより進捗状況管理機能を付与)

50名という体制下、各開発チーム・リーダーが自チームの進捗を報告するのではなく、コーディングを担当するメンバーそれぞれが自身の進捗を入力し、日本側に報告する仕組み(ウェブ上で自ら進捗を入力して更新できる仕組み)を進捗管理方法として確立



社内アセット化し、アセット登録ライブラリに登録

これを用いた同様の手法が他の後続プロジェクトでも利用可能に。中国だけでなく、インドと協業する際にも適用可能



## 具体的なUML活用工程 — なぜUMLを用いたか



### ■ UML の利用工程

- 要件定義、基本設計、詳細設計の各工程
- ユースケース図、ユースケース記述、シーケンス図を中心に利用。 UML に限らず日本語で記載

### ■ UML 利用の成果(メリット)

- 「統一された手法・モデリングによる情報共有」  
標準化されているため、ユースケース図やシーケンス図の読み方から説明する必要がない

### ■ UML 導入によるデメリット

- お客様と
  - お客様側に UML による説明を理解できるスキルが求められてしまう。当プロジェクトでは、そのため、別途業務フロー図などの資料を作成して補完した
- オフショア・チームと
  - 詳細設計工程において、シーケンス図だけで設計仕様を詳細・正確にすべてを伝えることは困難。また、当プロジェクトではオフショア・メンバーを日本に呼んで口頭で説明する機会を全く持たなかった。そこで、オフショア・メンバーに詳細・正確に仕様を伝えるべく、UML 以外に、メソッド記述書(メソッド毎の処理機能記述書)を別途作成して対応した

### ■ UML 採用の理由

- 当プロジェクトの特徴としてオブジェクト指向／モデル駆動による開発が主体
  - それらを実現でき、Java での開発とも親和性の高い UML を採用
- 標準としてオフショア・チームにもスキルを持つ技術者が多く、インターネットや一般書籍などでも情報が得られて調べやすい
- IBM Rational ツールを利用しての UML 作成が可能であり、実績から事例として展開する面でも貢献可能

## UML活用 — 日中それぞれの技術者と作業分担

### ■ 日本側体制（最大時 70 名の作業分担）:

#### ➢ アーキテクト・チーム

UML 利用経験や実開発経験でスキル・レベルの高い十数名が、アーキテクチャ策定と、作成する成果物の粒度や作成方法を決定。それに基づき、標準化チームがアプリ・チームに対してガイドを作成。

当プロジェクトでは、Rational ツールの使い方(プロジェクト独自のツールの使い方があることから)に関して、アーキテクト・チームから一度だけオフショアの現地にて勉強会を開催

#### ➢ アプリ・チーム

ガイドに基づき実際に設計するメンバーであり、オフショア・チームとの Q&A に関しては、実際に設計書を書いているアプリ・チームが対応

### ■ オフショア

#### ➢ 製造工程は、仕様として日本側から伝えられた通りにコーディング

#### ➢ 単体テストまではオフショア側で、コーディング不良の点検と修正し、日本側に渡す前に不良箇所は解決していることを前提

#### ➢ ツールでテスト実施カバレッジを測定するようしており、単体テストはオフショアでテスト項目を消化していれば、再度の検収を日本側で行う必要はなかった

### ■ 作業分担

	要件定義	基本設計	詳細設計	管理 (スケジュール ・品質)	製造(単体テスト含む)	結合テスト	受入テスト
お客様							
日本アイ・ビー・エム	●	●	●	●		●	●
オフショア				●	●		



## UML活用 — 成功箇所、工夫箇所

### ■ 成功箇所

- ▶ 業界で標準化された表記法のため、各設計書の標準化作業を軽減できた。オブジェクト指向言語とツールのサポートにより、モデルからコードを生成して実装の基盤とすることができ、開発生産性の向上につなげることができた。

### ■ 工夫箇所

- ▶ クラス図やシーケンス図の作成方法や作成基準などは事前に定義し、各設計担当者間で記述上の差異、設計上の差異が生じないようにした。また、その他の資料(メソッド毎の詳細な処理機能記述書など)も適宜補完して、オフショアに実装を依頼する上で、より仕様を明確にするようにした。

### ■ 今後の基本方針

- ▶ 引き続き UML を積極的に使用したい。
  - 業界標準であることからオフショア・チームにもスキルを保有している要員が多い。また、スキルが不足している場合でも情報源は多く、スキル向上が容易に可能な環境にある。
  - 統一されたモデリング表記のため、日本語・英語といった言語差により仕様理解が大きく左右されることが少ない(言語差による影響を抑えることができる)。
  - IBM 開発ツールのサポートにより、モデルからコードを生成して実装の基盤とすることができ、開発生産性を向上できる。

## UML活用 — 利用ダイアグラムと補完作成資料

### ■ 要件定義工程

- UML としてはユースケース図とシーケンス図を中心に、クラス図とロバストネス図も利用
- お客様によるレビューに用いた主要資料としては  
ユースケース図とユースケース記述、マイクロソフト社エクセルによる業務フロー図

要件定義/基本設計					
	図名	目的	作成者	効果	工夫した点
日本アイ・ビー・エム	クラス図	<ul style="list-style-type: none"> <li>・お客様の要求の定義</li> <li>・システム(アプリケーション)のアーキテクチャ、構造と振る舞いの定義</li> </ul>	日本アイ・ビー・エム	UMLを前提に開発ツール(Rational)を使用したことで、定義・設計の生産性が向上した。	<ul style="list-style-type: none"> <li>①業務フローは、お客様がUMLに慣れていないこともあり、EXCELを用いてUMLと異なる記法で作成した。</li> <li>②クラス図・シーケンス図の作成方法や作成基準は事前に定義し、各設計書間で担当者に依存する記述レベル差異が生じないようにした。</li> </ul>
	ユースケース図				
	シーケンス図				
	(ロバストネス図)				
詳細設計・製造・結合テスト					
	図名	目的	作成者	効果	工夫した点
日本アイ・ビー・エム	クラス図	<ul style="list-style-type: none"> <li>・アプリケーションの設計仕様を正確に伝えるため。</li> </ul>	日本アイ・ビー・エム	シーケンス図に加え、メソッド記述書も定義して作成することで、メソッドでの処理内容が正確に伝わった。	<ul style="list-style-type: none"> <li>①シーケンス図だけでは仕様伝達が難しかったため、メソッド記述書(メソッド毎の処理機能記述書)を別途定義して作成した。メソッド記述書は、I/P/O記述形式の定型フォーマットを作成し、各メソッドの入力と出力、処理内容を明確にした。</li> </ul>
	シーケンス図				
オフショア	クラス図				
	シーケンス図				

## UML活用 — 利用ダイアグラムと補完作成資料

### ■ 基本設計、詳細設計工程

- ▶ UML としてはシーケンス図、クラス図とロバストネス図
- ▶ UML 以外に作成した主要資料として  
画面定義資料、DB 定義( ER 図)資料、メソッド記述書など
- ▶ メソッド記述書は当プロジェクト独自で、I/P/O ( Input/Process/Output ) が明確になるよう定型フォーマットを指定。
  - ▶ Rational ツール上にあるモデルの、どのクラスに対する処理の内容かが正確に特定可能
  - ▶ 文は短く、主語や目的語は明確に示すようガイド
  - ▶ 詳細に標準化し、例えば違うメソッドを呼び出す場合の書き方などのレベルまで定義してガイド  
(例えば、“ここで共通コンポーネントを使う”という場合は、共通コンポーネント名を書き、処理の順番がわかるように、呼び出す箇所についても明確に書くことをガイド)

### ■ オフショア・チーム作業工程

- ▶ シーケンス図とメソッド記述書を中心に読み、仕様を理解
- ▶ Rational ツール上、メソッドの外枠までツールにより自動的に生成され定義される。これらメソッドに対して、Rational ツール環境上でコーディングを実施。  
(外枠に対して、その処理内容となる中身の業務ロジックを、メソッド記述書を見ながらコーディングするというイメージの作業)



## UML活用 — 開発ツール

### ■ 開発

#### IBM グループ共通の開発環境

- Rational Software Architect

### ■ テスト

担当者によるプログラム動作確認は基本的にツールで実施する方針（ツールで可能な範囲内）

- Rational Functional Tester や社内アセットとしての点検ツール

### ■ 管理

進捗が具体的に見えるようにする、コミュニケーションをとり、お互いの状況を把握できるようにする

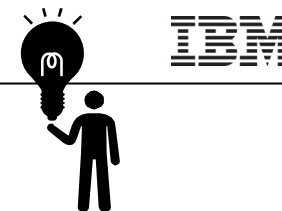
- 変更管理、進捗状況管理には Rational ClearQuest
- Q&A 管理には Rational Portfolio Manager

### ■ ツール適用にあたって

- 他社メンバーも参画してマルチ・ベンダーで開発するプロジェクトとなると、ツールの使い方など教育の工数が生じる
- 当プロジェクトより小規模なプロジェクトであれば、ツールの使い方を変えることも検討する（少人数プロジェクトでは、ツールを多く使うことで、ライセンス費用や教育工数の相対的割合増加）

### 現在の Rational ツール群

- エンタープライズ・アーキテクチャの管理
  - Rational System Architect
- 要求管理及び要求定義
  - Rational Requirements Composer, RequisitePro, DOORS
- アーキテクチャの管理
  - Rational Software Architect, Application Developer for WebSphere Software, Rhapsody, PurifyPlus, Business Developer
- 変更とリリースの管理
  - Rational Team Concert, Build Forge, Change, Synergy, ClearCase, ClearQuest, Software Analyzer
- プロセスとポートフォリオの管理
  - Rational Focal Point, Insight, Method Composer
- 品質の管理
  - Rational Quality Manager, Functional Tester, Performance Tester
- ウェブ・サイト・セキュリティとコンプライアンスの確保
  - Rational AppScan, AppScan Source Edition, Policy Tester



## UML利用の効果と課題 まとめ

- ✓ UML が有効である場面においては UML ツールや社内資産も活用し、オフショア先に仕様や範囲を伝えた
- ✓ UML でカバーできない部分においては事前定義した補足資料を作成し、それらをオフショア先と共有した
- ✓ オフショアに開発依頼する際の進捗管理や Q&A 管理等の管理作業は事前に洗い出し、ツールを用いて情報共有できるよう対策した
  
- ✓ オフショア開発に UML を導入する際、必要を感じるもの：  
当プロジェクト事例から、UML モデリングだけでは仕様伝達に不十分といえる。開発工程において UML だけでは不十分な箇所を補足するため、いつどのような資料をどのレベルで作っていけばよいか(当プロジェクトの例では、メソッドに関するメソッド仕様書を作成など)ガイドできると、より広く安心して利用しやすくなる

### UML 以外での成功のポイント:

- ✓ 技術面では、日本側で事前にアーキテクチャを設計し、また、各種成果物の標準化を徹底して、オフショアへの引渡し前に品質を高めたこと。また、Rational ツールを使う、社内アセットを使うといった社内資産の活用を積極的に実施したこと
  
- ✓ オフショアでの協業には管理面でも様々な障害が生じる。(例えば、進捗状況が見えない、Q&A の内容が共有されず、いろいろな開発担当者から同じような質問を繰り返し投げられて、日本側に過剰な工数負荷が生じる、など。) それら懸念点を事前に洗い出して対策を立て、適切な準備期間を経て、適切なツールを使って対応したこと