

販売基幹システム構築プロジェクトでの UML 適用事例インタビュー 報告

2011 年 3 月

特定非営利活動法人 UML モデリング推進協議会
オフショアソフトウェア開発部会

Copyright (C) 2011 特定非営利活動法人 UML モデリング推進協議会 All rights reserved

日本アイ・ビー・エム株式会社

Copyright (C) 2011 IBM Japan, Ltd. All rights reserved

インタビューの内容

インタビュー日：2010年12月13日

インタビュアー：サザーランド真理亜（一部 中原俊政、藤野博之、竹政昭利）

インタビューイ：日本アー・ビー・エム株式会社 畠山園子（一部 西川啓、吉田亮）

イントロダクション

[中原] 部会では、オフショア開発に UML を適用してプロジェクトを成功させるためにはどうすればいいかということの研究しており、その成果としてオフショア向け UML 適用ガイドライン（以降ガイドラインと略す）をまとめ、合わせていろいろな開発事例を集めています。ガイドラインの適用事例に限らず、UML を使ったオフショア開発の事例についてインタビューし、そこから成功要因、例えばこういった開発であればこのようなガイドラインの使い方をすると有効であるということを読み出すことが目的です。

インタビューの方法として、あらかじめアンケートにより基本情報として記入いただいた内容に従ってインタビューさせていただき、もう少し掘り下げたいところはより詳細に聞かせていただく形で進めます。そして最後に、どのようなことを実施したことで成功につながったという形でまとめたいと思っています。

基本情報1の確認

プロジェクト概要

プロジェクトについて

では、最初に基本情報の確認から始めさせていただきます。

プロジェクトは製造業の分野を対象にした業務システム。現行システムがある中での再構築ということですが、現行システムが最初に構築されてからどれぐらい経っての再構築なのでしょうか。

正確な年数はわからないのですが、かなり古いシステムということで、再構築といってもほぼ作り替えというようなイメージです。

対象プロジェクトの規模は数億円、開発期間は12カ月程度ということで、これは日本側とオフショア先の中国側で、それぞれどれぐらいの方が携わった開発なのでしょうか。

五月雨式で進めていたプロジェクトですので人数に増減の波があるのですが、最大時には IBM 側 70 名、中国側 50 名になります。

中国の方が日本にいられて、ブリッジ SE 的な形で体制に加わるというようなことは無かったと理解してよろしいでしょうか。

このプロジェクトの特徴でもあると思うのですが、中国から日本に来ていただいて、ある期間日本側と一緒に作業を進めるという形式はとっていません。

次に、プロジェクトの特徴をお聞かせください。まず「IBM 製品（Rational Tool）やアセット

を利用したモデル駆動による開発を実施し、開発工数の削減と品質の向上を実現」ということですが、**Rational Tool** は具体的にどのように利用したのでしょうか。

主に設計をする工程やコーディングをする工程、あるいはテストをする工程にこのツールを使うことができます。それをうまく活用しながら、日中間でよりコミュニケーションがスムーズにとれて、かつ、中国側の進捗が見やすいように工夫しながら使いました。オフショアの開発チームとの間で並行開発をより行いやすくするためのツールと位置づけられると思います。

具体的にはどのようによくなるのでしょうか。

いろいろあるのですが、例えば、中国側から日本側作成の設計書に対して質問がある場合、その質問を起票して、起票された質問に対して日本側が回答するというワークフローを定義しておく、関連のプロジェクト・メンバー全員がそのツール環境で同時に質問・回答を見る、即座に情報共有できるといった利点が挙げられます。

「アセットを利用したモデル駆動による開発」とは具体的にどのようなものなのでしょうか。

モデル駆動開発では、モデルを中心として、前後工程の成果物間での追跡可能性を確保しながら開発していきます。社内でいろいろなプロジェクトを経験したメンバーらがモデル駆動開発を効率よく進めるためのアセットを開発して、社内アセットとして登録しています。それを有効活用しながら、さらに拡張して再登録するといった活動もこのプロジェクトで行っていました。それをここで挙げさせていただきました。

Rational Tool やアセットを利用したモデル駆動による開発を取り入れることによって、開発工数の削減と品質向上が実現できたということでしょうか。

はい。

次に、「業務アプリにおける開発作業の約 97%をオフショア先に依頼するとともに、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」ということですが、この約 97%とは、今回の開発作業の 97%ということでしょうか。

その通りです。

社内標準のプロジェクト管理手法ということですが、**IBM** 様がオフショア開発をされる際、常に使っている標準の管理手法ということでしょうか。

はい。**IBM** グローバルの社内標準ですので、日本、中国、米国など皆が同じ手法を使っています。

【西川】この手法でプロジェクト管理について成果物定義も含めて全般的に網羅されていますが、そこから各プロジェクトの適性に応じてピックアップして利用する形になります。方法的な部分がこの管理手法に定義されており、それを実際にプロジェクト適用する際に、**Rational Tool** によって選択使用できるようにデザインされています。

次に、「各作業においては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いて成果の最適化を実施する」ということで、成功したプロジェクトとしてこれだけで終わらせるのではなく、類似の開発に成果を再利用できるよう、具体的にどのような取り組みをされたのでしょうか。

オフショア先と協業する上での考慮点や、**Rational Tool** で実施した成果や過程など使い方を、社内研修を通して展開しました。また、**Rational Tool** をうまく使うための活用法を経験に基づいてまとめた資料などは社内アセットとして登録することを行いました。

オフショア先について

では、オフショア先についてお伺いいたします。オフショア先は中華人民共和国。オフショア先企業規模は、相手国内のトップクラス規模の企業ということですが、具体的には。

【西川】IBM のソリューション子会社です。世界中にあるいろいろな開発拠点の一つといえますか、IBM グローバルで、オフショア開発する際の IBM 向けソリューション開発を担当する子会社という位置づけです。上海、大連、成都、深セン、武漢の5カ所にあります。

今回の開発で主に連携をとられたのは、そのうちのどちらでしょうか。

大連です。

そのオフショア先企業は、UML のスキルを保有していましたか。UML の教育をしましたか。UML スキルについては特に必要要件としておらず、教育など実施していないものの、今回参加メンバーの多数は UML スキルを保有していました。

このプロジェクトでは、UML を使った設計書を中国の方に書いてもらうことはしていませんので、教育はそれほど重視しませんでした。参加メンバーの皆は既に UML スキルを持っており、読んだりすることはできますので、特に要件として希望しませんでした。

もともと UML を使う予定ではあったわけですね。

はい、オフショアのために UML を採用したわけではありません。日本側で UML を用いて設計書まで書いています。そして、それを修正したり、新たに UML を使って設計書を作成したりするところまでは中国にはお願いしませんでした。読んで理解するということでは全く問題ありませんでした。

質疑応答 その1

質問者：中原俊政、藤野博之、竹政昭利、サザーランド真理亜

回答者：日本アー・ビー・エム株式会社 島山園子、西川啓、吉田亮

アセットについて

【中原】では、ここから質疑応答とさせていただきます。まずプロジェクトの特徴的な部分等について、アセットをうまく使ったというお話がありました。アセットにはいろいろな種類があると思いますが、その具体的な内容と、その運用を定着させるためにどのようにしたかということをお聞きしたいと思います。

プログラム部品だけでなく、オフショア先と進める中でのコミュニケーションや設計書の書き方、開発プロセスなど、ノウハウ的なものも含めて、何か工夫した点などありますでしょうか。

【島山】先に挙げた、利用した社内アセットというのは、ほとんどプログラム部品や技術に関わるものになります。オフショア先とのコミュニケーションをスムーズにするというアセットは特に利用しておらず、それはむしろこのプロジェクトで作ったといえます。

まず、私がこのプロジェクトに参画する前、長い間オフショア開発に携わっていたことがあり、上海に実際に赴任して、日本と上海とで協業する際に、設計書などのレベルをどこまで書くべきか、どういう設計書が必要かという標準化を行っていました。そういった経験と、今までの開発プロジェクトでの経験を活かし、このプロジェクトでオフショア開発チームと協業する前に、まず社内で見られる既に明らかに問題となっているところを整理してみました。それらの問題に対してどういう施策を打ち出せばよいかということを考えて上で、いろいろと対応していったわけです。

その代表的な一つの例に、例えば開発チームがいるのはリモート・サイトですので、どうしても中国側の進捗が日本からは見えにくいことがあります。中国側で何がどこまで進んでいるのかわかりにくいということは、他のプロジェクトの PM メンバーも言うところです。そこでうまく可視化するために、Rational 製品群の中の、変更管理をするために使う ClearQuest というツールを用い、それをカスタマイズして進捗管理ができるようにしました。

例えば 50 名という体制ではチーム分けをして何人かリーダーの方を設けるのですが、そのリーダーが自分のチームの進捗を報告するのではなく、コーディングなどを行っているメンバーそれぞれが自分の進捗をきちんと把握し、日本側に報告する仕組みとするため、ウェブ上で自ら進捗を入力して更新できるような進捗管理方法を作りました。このプロジェクトで作った代表的なアセットの例です。この手法は他のプロジェクトにも利用されています。中国だけでなく、インドと協業するときにも使っています。

【西川】補足しますと、社内にアセット登録ライブラリとも言えるものがあって、そこに作ったアセットを登録するわけです。プロジェクトが終わったときに、そのプロジェクトで作られた再利用性の高そうなアセットは、個々人がそれを整理して登録します。有用な内容であれば繰り返し再利用されますし、再利用件数の多かった上位幾つかのアセット登録者が表彰されるという社内の仕組みもあります。

【中原】例えば UML のメタ・モデルみたいなものはアセットにあるのでしょうか。やはり一番多いのはプログラムでしょうか。

【島山】お客様ごとに異なり、プロジェクトの特色が出るところですので、そういったモデルのアセットはあったとしても活用性は低いのではないのでしょうか。弊社ではフレームワークなど、プログラムというより部品の再利用が多いですね。

Rational Tool について

【藤野】 Rational Tool のお話が出てきましたが、これはプログラム開発用の環境としてでしょうか。 UML を設計段階から使っているというお話がありましたので、そのときに使ったツールが Rational Tool ということでしょうか。

【吉田】 ここでの Rational Tool には二つあります。 UML を使って設計、コーディングをするという開発系列のものと、もう一つは、リモートにいるチームと連携する、変更管理するというプロジェクト管理系列のものです。その二つあるうちの、これまでのお話では後者の方のお話であり、メンバー間で情報をやりとりする、共有するというところで活用した例として特に取り挙げています。開発にも Rational Tool は多く使われます。

【島山】 このプロジェクトでも、プログラム開発用に Rational Tool を使いました。 Rational Software Architect などと呼ばれるツール群です。

オフショア側と日本側の分担について

【藤野】 お話の中で作業の 97% をオフショア先で実施ということでしたが、残りの 3% はどこで行われた何の分なのでしょう。

【島山】 97% 分を開発するための共通コンポーネントとして核になる、日本側の標準アーキテクト・チームが構築した分と捉えていただければよいと思います。

この 97% とは、コーディング作業の 97% の意味です。コーディング作業全体を 100% としたときの 97% の作業をオフショア先にお願ひし、残りの 3% は、日本側で共通コンポーネントの開発を行ったということになります。 オフショア先で行ったのはコーディング、単体テストです。

アセットの運用形態

【中原】 アセットを使って、その利用者が拡張もできるというお話がありましたが、アセットが勝手に拡張されて登録されると混乱するので、アセットの管理部署がチェックするような仕掛けが御社にあるのでしょうか。

【西川】 方法論的なアセットについては、基本的に IBM 内でグローバルに統一されていますので、各国でばらばらになったりすることはありません。また、アセット製造部署が作るプログラム部品やフレームワークといったアセットに関しては、その製造部署がオーナーシップを持って開発や拡張管理を行いますので、アセットそのものが勝手に改変されることはありません。アセットが登録されている状態ではそのように変わるものではないのですが、それをプロジェクト側で拡張や変更して使うことは可能です。先程のお話はこのことを指しています。

基本情報2の確認

インタビュアー：サザーランド真理亜（一部 中原俊政、藤野博之、竹政昭利）

インタビュイー：日本アー・ビー・エム株式会社 畠山園子（一部 西川啓、吉田亮）

UML 活用状況

UML を適用した工程

今回、オフショアでのプロジェクト実施の際、UML は、要件定義、基本設計、詳細設計の各工程で採用され、UML 採用の理由は「オブジェクト指向による開発実施のため」、成果（メリット）は「統一された手法・モデリングによる情報共有」ということですね。

はい。

UML 導入によるデメリットとして、「お客様が UML を理解できるためのスキルが必要。そのため別途業務フローなどの資料も作成」と挙げられています。それから、詳細設計工程では「UML の成果物だけでは仕様を伝えることが困難なため詳細なメソッド記述書（メソッド毎の処理機能記述書）を作成」とののですが、UML で使われた図は具体的にどのようなものだったのでしょうか。

ユースケース図、ユースケース記述とシーケンス図が中心です。

今回、UML 導入によるデメリットについてお客様に対するデメリットとしてお答えいただいているのですが、中国側ではどういうメリットとデメリットがあったとお感じでしょうか。

メリットとして、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く、それをお渡しすれば中国側でわかっていただけということはあるものの、シーケンス図だけでは設計仕様を詳細に伝えることが困難でした。また、今回のプロジェクトでは、中国の方を日本に呼んで口頭で説明するという事は全く実施していません。ただし、リモート側にきちんと詳細に仕様を伝えることは UML だけでは難しかったので、メソッド記述書という資料を別途作ることにしました。

そのメソッド記述書は日本語で記述されたのでしょうか。

はい、日本語です。メソッド記述書に限らず、仕様書は日本語になります。

中国側の日本語能力はどれぐらいのレベルなのでしょうか。

【西川】プロジェクトによって違います。日本語能力を求めるプロジェクトにはなるべく日本語がよくできる方を体制に入れます。日本側と接する役目のオフショア側担当者には、相対的に日本語を流暢に話せるレベルの方があたるようお願いしています。

仕様書を日本語で作成してお渡ししても、問題なく理解していただけるということでしょうか。

【西川】問題は完全にはありませんが、何度かやりとりするうちに収斂されてくるレベルですね。

Rational Tool を使って情報共有したり、質疑応答されたのでしょうか。

質疑応答では Rational Tool を使いました。しかし、仕様書の共有などでは特にツールを使うことはなく、それ用にファイル・サーバを立てて、中国からアクセスして参照したりするイメージ

です。

UML を使用したきっかけ

次に、UML を使用したきっかけです。「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を主体としており、それを実現でき、Java での開発とも親和性の高い UML を採用した。また業界標準なのでオフショアでも理解している技術者が多く、またインターネットや一般書籍などでも情報量が多く調べやすいという点と、IBM のツールを利用しての作成が可能であり、それらの推進と実績を社内で事例として展開するという点でも貢献できると感じたため UML を採用した」ということでよろしいでしょうか。

はい。

実際に今回のプロジェクトに携わられたのは多いときで 70 名ということですが、UML を日ごろから、国外に限らず、国内でも頻繁に使っておられるのでしょうか。

そうなります。

UML はオフショア開発の中では標準的に使われているのでしょうか。

【西川】必ずしも標準とは言えないと思います。プロジェクトの性格、性質に応じて、使う・使わないを自分たちで決めるというところがあります。一昔前、日本 IBM 成りの標準ガイドがあり、開発工程上流から下流までほぼ完結していて、その中で UML という話はほとんどなかったのですが、日本だけでなく IBM 内でグローバルに統一された標準にしようと。そのために方法論自体を置きかえる動きがあって、その過程で UML が使われるようになったとは思いますが、標準として必須かという点、そうではないと思います。

「IBM のツールを利用しての作成」ということですが、御社にある UML の作成ツールを活用するわけですね。

そうです。それが先程お話しした Rational Software Architect という開発ツールで、その環境で UML を作成することができます。

そういうモデル図例みたいなものがあり、それを開発者自身が思っているようなものに変更していくとか、拡張していくとか、そういうイメージでしょうか。

例えばユースケースでは、Actor や Use Case という部品がツール上に事前定義されていますので、それを画面上でドラッグ・アンド・ドロップしながら作っていくというイメージです。

御社のグローバルの子会社でも同じようなツールが使われるのでしょうか。

そうですね。弊社のソフトウェアですので、オフショア先で開発するときも同じツールを使う例が多いと思います。

UML を使って上手くいった点

次に、UML を使って上手くいった点については、「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、開發生産性も向上することができる」。これは実際にそのようにできた、UML の効果があったということではよろしいでしょうか。

はい。

UML を使う上で工夫／改善した点

UML を使う上で工夫／改善した点として、「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するようにして、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ためにより仕様を明確にするようにした」ということで、設計担当者が多くいて、一緒にひとつのものを設計していくというイメージでよろしいでしょうか。設計者が複数いて、皆それぞれ自分の担当のシーケンス図などを作っていくのですが、その際、シーケンス図の作成の仕方や粒度などがばらばらになってしまっても困りますので、まずその標準化を行ってから設計を始めました。

ばらばらになってしまわないように、担当者が一度集まり、このように定義していこうという会議などを持って、合意の上で進めていくということでしょうか。

そういう標準化を行うチームが設計者チームと別にありますので、私なども支援して、その人たちが標準としての定義を作り出し、それを設計者らに展開するというイメージです。

設計者の誰が作成しても同じレベルの仕様書が得られるということですね。

そうですね。

UML の活用についての今後の基本方針

オフショア開発での UML 活用について今後の基本方針はどうでしょうか。UML を積極的に使用したいということで、その理由としては、「業界標準なのでオフショアのメンバーでもスキルを保有している人が多い。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可能な環境にある。また、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 での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、開發生産性を向上できる」ということでよろしいでしょうか。

はい。

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ということですが、御社で資料作成される際に使われる言語としては日本語と英語が主で、中国語などはどうでしょうか。

提供する仕様書を中国語で作成することはありません。中国のオフショア先に依頼する際の基本は日本語ですが、インドやフィリピンなど、他のオフショア先に依頼するときには英語になります。

中国だけは日本語ができるメンバーがいるので、日本語でよいということでしょうか。

そうですね。

UML を導入する際、必要と感じた物

最後に、オフショア開発に UML を導入する際、必要を感じたものとして、「UML によるモデリングだけでは仕様の完全な伝達、理解は不十分になると思われる。そのため、各クラス仕様書などの処理機能を記述した文書を別途作成するなどの必要があると感じて、そのように実施した」ということで、これは先程言われたように、UML 図だけでは足りないので、処理機能を記述した文書を別途作成されたということですね。

はい。

質疑応答 その2

質問者：中原俊政、藤野博之、竹政昭利、サザーランド真理亜

回答者：日本アー・ビー・エム株式会社 畠山園子、西川啓、吉田亮

作成した UML の図と UML 以外の成果物

【中原】では、質疑応答に移らせていただきます。UML で使ったのはユースケース図、シーケンス図ということで、要件定義としては、お客様が UML だけでは理解できないため、別途、業務フローなどの資料を作成したとのことでした。業務フローとしてはエクセル・ベースやアクティビティ図などがあるのですが、具体的にはどのようなものを作られたのでしょうか。

【畠山】エクセル・ベースでの業務フローです。

【中原】アクティビティ図ではやはり部署や物などがわかりづらいということでしょうか。

【畠山】そうですね。ちょっとわかりにくいと感じました。お客様の視点を重視していますので、エクセルを使ってより詳細な業務フローを作りました。

【中原】作ったものとしてユースケース図とシーケンス図と言われたのですが、後の工程ではクラス図も作成と出てきていますので、クラス図も作られたのでしょうか。

【畠山】はい、クラス図と、それから、ロバストネス図も作成しています。

【吉田】専用ツールですので、どの図を作ったかというお答えは難しく、この図を作ると、そこから結びつけて考えられる他の図も自動的にツールで生成されているということが出てきます。実際に開発チーム内やお客様との仕様確認などに使った図が先ほど挙げられたものだと捉えていただくのがよいと思います。

【中原】Rational Tool でクラス図はできるのででしょうか。

【吉田】例えばシーケンス図で A と B のコンポーネント間でやりとりをしましようというときには、A と B というコンポーネントがあるはずですねということで、ツールで自動的にこの 2 つのクラスが生成されます。ですから、お答えとしては難しいですね。

【畠山】そういったクラス図をお客様に見せて仕様を確認するという事は行いませんでした。

【中原】なるほど、では要件定義でお客様のレビューに使った主な資料としては、ユースケース図とユースケース記述、エクセル・ベースの業務フローという理解でよろしいでしょうか。

【畠山】はい。

【中原】基本設計、詳細設計では補完資料が必要ということで、詳細設計ではメソッド記述書を書かれたということですが、基本設計ではクラス図、シーケンス図、ロバストネス図以外に何か資料を作られたのでしょうか。

【畠山】作りました。ただそれは UML ではなく、先程の業務フロー資料のように UML とは異質のものです。図で書き表す類ではなくて、エクセル上で記述したものです。

【中原】画面などでしょうか。

【畠山】はい、画面定義資料も作っていますし、DB系(ER図)資料などもそうです。

【吉田】設計書諸々、従来の開発でもよく使われてきた成果物群と考えてください。

【中原】詳細設計は、シーケンス図と詳細クラス図、あとはメソッドの中身を書いたメソッド記述書ということですね、このメソッド記述書の中身は具体的にどのようなものでしょうか。定型フォーマットがあって、例えば箇条書きにするなど、きちんとわかりやすく書くように工夫されているのでしょうか。

【畠山】プロジェクト独自で定型フォーマットを作りました。基本的には、よく使われる IPO (Input/Process/Output) が明確になるようにしています。また、例えば Rational Tool 上にあるモデルのどのクラスに対する処理の内容なのかということが正確にわかり、きちんと連携できるようにするという工夫をしています。

【中原】書くときにはあまり長くないように箇条書きにするといいますか、文を短くするというようなことは自由なままでしょうか。

【畠山】文は短く、主語や目的語は明確に示すようにガイドしています。そして、ここはかなり詳細に標準化しており、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います。

【藤野】アセットの、特にプログラム部品などを使うときの指定は、メソッド記述書の中で行うのでしょうか。

【畠山】はい、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いて、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してきちんと書くようにガイドしています。

【竹政】メソッド記述は Rational Tool の中には入っていないのでしょうか。

【畠山】メソッド記述書は入っていません。メソッド自体の定義はツールに入っています。業務ロジックをコーディングするときの参照資料として、別途エクセルを用いて Rational Tool の外で書いています。

【竹政】中国側の方はそれを見てコーディングしていくという感じでしょうか。

【畠山】そうです。中国側はその資料を見ながら、Rational Tool 上に定義されているメソッドに対して、Rational Tool の環境上でコーディングしていくというイメージです。

【藤野】そして、そこから自動的にソースが生成されると。

【畠山】メソッドなどの外枠まで、ツールによりすべて自動的に生成されます。ですから、後は外枠に対してその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していくことになります。

【竹政】中国側は主にメソッド仕様書とシーケンス図ぐらいで十分コーディングできるという感じでしょうか。あるいはメソッド仕様書だけでもコーディングできてしまうとか。

【畠山】シーケンス図は必要です。シーケンス図とメソッド記述書が中心になります。クラス図なども要らなくはないのですが、Rational Tool 上にすべて揃いますので、必要になればクラス図と、それからユースケース図まですべて見ることができます。

【藤野】オフショア開発での UML の活用の割合はどれぐらいでしょうか。半分ぐらいなのか、それとも 3 割程度なのか、UML を使って作るのが標準的なのか、それとも特定のプロジェクトでなければ使われないのか、差し支えなければ教えていただけますでしょうか。

【吉田】お客様側が理解できるかということもありますし、開発言語がオブジェクト指向ではなく、例えば従来型の COBOL であったりと、プロジェクト特性がいろいろありますので、なかなかこうとは一概にはお答えしにくいですね。Java などのオブジェクト指向言語利用の場合で

あれば、UMLの使用が候補に挙がるということになります。

【藤野】あとはお客様側の要件ということでしょうか。

【吉田】はい。プロジェクトで開発ツールがそれだけ揃えられるかとか、作成したファイルを出力してお客様にお渡しして表示して見ることができるかとか、いろいろな条件が絡んできます。

【藤野】別の見方をすると、御社の中では UML を使うことに対してお客様からリクエストがあるといえますか、ニーズが一致した場合には、何も困らずに、そのまま製造まで対応できるような体制をとることができる状況になっているということでしょうか。

【吉田】はい。ツール、教育、経験が揃っています。

【竹政】お客様の事情としては、日本特有というところがあるのでしょうか。海外のお客様の場合はそれほど違和感がないとか。

【吉田】それはあると思います。例えば米国とインドによるオフショア開発で、米国のお客様に行くのであれば、そのまま Rational Tool を使ってもいいと思います。ここで日本語のバージョンでやりましょうといっても、ツール側のこの機能では日本語が表示できないとも言いかねませんので、日本の事情ということはあると思います。

【西川】逆にお聞きしますが、他の開発者社様やお客様などでは積極的に UML は採用いただいているものなのでしょうか。

【竹政】日本では進んでいないところがあります。それは主にお客様の事情で、UML はよくわからないので、日本語文章での記述仕様書を好まれるというところがあります。オフショア先になると、日本からの開発依頼は UML ではないが、ヨーロッパやアメリカなどからの開発依頼は UML で対応しているというところが多いようです。

メソッド仕様書の記述レベル

【中原】メソッド記述書でメソッドの処理を書くとのことですが、どのレベルまで書くのでしょうか。フローチャートのような図を使って書くのではなく、日本語文章で書くとしても、どのレベルまで書くのでしょうか。あまり詳しく書くとよくないといえますか。

【畠山】記述レベルはかなり詳細です。コードに近いレベルだと捉えていただいてもいいと思います。

【中原】それはすべてのロジックを詳細に押さえなければ書けないと思いますが、そこまで細かく書くのであれば、自分で実装したほうが早いような気がします。オフショアに出すときはそのあたりの記述レベルが難しいと思っていたのですが、どうなのでしょう。詳細レベルまで書いても、オフショア側が実装したものが思ったように動作しないといったやりとりが結構出るのではないかと思うのですが、そういうことはなかったのでしょうか。

【畠山】それほどありませんでした。詳細に書いていましたので、その点ではここがわからないといった Q&A は比較的少なかったと言えます。

【中原】仕様としてそのように依頼したはずなのに正しく動作しないということはないのでしょうか。

【畠山】それはなかったです。五月雨式で進めていたこともあり、最初の段階ということであえて仕様を詳細に書きました。次回の五月雨式のときにはもう少し記述を少なくするといった工夫をしようという意図もありましたので、初回はオフショア側のメンバーの理解を深めるためにも、あえて詳細に書きました。フローチャートのような図は基本的に使わないです。メソッド単位ですので、日本語でロジックを箇条書きのように記述しました。

【竹政】 オフショアにお任せするとき以外にも、普通にその詳細レベルで書くのでしょうか。

【島山】 その必要はないと思っています。今回は日本側のプロジェクト開発チーム・メンバーが全くオフショア開発の経験がないという状況で私が出たという事情もありましたので、成功させるために、一旦そのように細かいレベルで書くようにしました。ですから、必ずしもすべてのプロジェクトにおいて同じように適用しなければならないということではないと思っています。

【竹政】 そのレベルまで書くとなると、日本側の作業量が多くなりますよね。

【島山】 それはありました。ただ、オフショア開発を実施するときにはどうしても設計書の品質を高めておくことが大事ですので、そこを開発チームに理解してもらうため、日本側の設計書のレビューも厳密に行いました。そういう意識を徹底させるという意味で。

【竹政】 日本側でコーディング・レベルまでかなり把握していないと、そこまでは仕様書を書けないのではないかという感じはしますね。

【島山】 そうですね。ここでは Rational Tool 上でのモデル駆動開発を行うことで、仕様はメソッド単位ですべて分割されて定義されますので、チームの皆が同じイメージを共有できていたということはあると思います。

【竹政】 全体的なシーケンスの流れはそこでできて、プラス・アルファとしてメソッド仕様書でメソッドの中身を書くということでしょうか。

【島山】 そうです。シーケンス図をきちんと理解してメソッド仕様書を書こうとしているチーム・メンバーであれば、中身を書くのはそれほど難しくありません。その理解のギャップは生じないと思っていますので、そこでコーディング・レベルまで想像できないということはないと思います。

【竹政】 シーケンス図や一部のクラス図を書く人が、そのままメソッドの仕様書まで書くということでしょうか。

【島山】 そうです。設計しているメンバーがメソッド記述書も書くというイメージです。

【中原】 メソッド仕様書の記述レベルの推奨はどうでしょうか。オフショアでうまくやるためには、やはり細かく書いたほうがいいのか、それとも、ある程度ラフでも、入出力さえ明確にしておけばいいのか。私は後者のほうではないかと思っていますのですが、こうすればうまくいくということがあれば教えてくださいませんか。

【島山】 思いとしては、日本側の設計書がきちんとした品質を確保できていれば、メソッド記述書を今回ほど詳細に書く必要は全くないということを感じます。

【中原】 細かく書くと、日本側でオーバー・ヘッドがかかるように思うのですが。

【西川】 そこは非常に議論の分かれるところで、失敗が許されないプロジェクトにおいては過剰に詳細化する傾向はあります。標準化ルールを決めているのですが、それでも日本側の設計書の品質が低かったりすると、オフショア先から届くものへのやり直し作業が大変になってしまいます。現状では、一度当該システムの経験を積んだ後であれば比較的緩くなるのですが、初めての、経験のないケースでは詳細まで指定してがんじがらめにする傾向があります。

【島山】 このプロジェクトも初めてでしたので、かなりがんじがらめにした面があります。

UML 技術者

【竹政】日本側の開発メンバーは UML については何ら問題なく、すぐに対応できたわけですね。御社社員であれば UML 知識や経験は必須という感じなのでしょうか。

【西川】それもまたケース・バイ・ケースです。既存システムで全く UML を使わずにいる人や、そういうシステムのメンテナンスにずっと携わっている人からすると、UML を使う動機づけがありませんので、UML 実使用経験がない人もいるとは思いますが。一方、新しく Java による開発といったプロジェクトに参加すれば普通に UML を適用しています。

【島山】そうですね。社員全体としてどれぐらい浸透しているかはわからないのですが、このプロジェクトに関しては、参加メンバーにそこで UML 教育をするというようなことはありませんでした。

【藤野】UML を使える方でも、上級者、中級者、単にシーケンス図が書けるだけの人など、いろいろなレベルがあると思いますが、日本の 70 名の中でその割合はどうだったのでしょうか。例えばリーダー的な方がいて、その方が皆さんに教育してレベルアップを図られたのでしょうか。70 人のレベルがほとんど同じということも考えられないです。

【島山】70 人全員が同じ高いスキル・レベルを必要とされる作業をしているわけではありませんので、設計に携わるのは上級者が多いと言えます。今回数十人ぐらいでしょうか。全体的にプロジェクト・メンバーの UML の知識レベルは高かったと思います。そういうメンバーを積極的に集めてチーム構成したというところでしょうか。

【竹政】全体を考えるアーキテクトみたいな方がいて、もう少し詳細な設計者がいて、メソッドレベルの作成者がいるというイメージでしょうか。

【島山】そうですね。体制の上で、アプリ・チームとアーキテクト・チームがあり、アーキテクト・チームが作成する成果物の粒度や作成方法を決めていきました。実際に設計するのはアプリ・チームのメンバーです。標準化のアーキテクト・チームがそこに対してガイドを作っています。

【竹政】アーキテクト・チームがアーキテクチャを大体決めて、それに沿ったような形で設計するというイメージでしょうか。

【島山】そうです。

【竹政】アーキテクト・チームには何人ぐらいおられたのでしょうか。

【島山】10 数名ですね。UML 経験や実開発経験でのレベルの高い者が IT アーキテクトのリーダーとなり、いろいろと新しいことも取り入れていくという進め方でした。

【竹政】アーキテクト・チームの方がオフショア先と直接コミュニケーションをとることはあまりないわけですか。それはアプリケーション・チームの方がされるのでしょうか。

【島山】基本的にアーキテクト・チームではないです。Q&A などに関しては、実際に設計書を書いているアプリ・チームのメンバーが対応します。ただ、Rational Tool の使い方などに関して、アーキテクト・チームから最初に 1 回だけオフショア先の現地に行って勉強会を開きました。プロジェクト独自のツールの使い方共有というところからです。日中間でのプロジェクト管理や Q&A の行い方などはすべて Rational Tool を使っており、そこでもプロジェクト独自の使い方が生じますので、その説明も合わせて現地で行いました。

オフショア先との作業分担

【中原】単体テストを含めた製造工程をオフショア先に依頼しているということですが、結合テストはどちらでしょうか。

【吉田】結合テスト以降は日本側です。単体テストまでは、中国側で不良の点検と修正をして、日本側に渡す前に不良は直していただいている形になります。

【中原】例えばシーケンス図に絡むような不良が発生したときは、どういうやりとりをするのでしょうか。オフショア側は、あくまでも UML を読んで実装するだけでしょうか。

【吉田】製造工程では、仕様として日本側から渡された通りにコーディングします。シーケンス図に基づく不良など、結合テストを実施したときに何か起きたら、どちらがどう直すのかという質問でしょうか。

【中原】メソッド記述書が完璧にできていればコーディング・ミスとしての修正でしょうけれども、そもそも設計の不備などがあると結構やりとりが生じるのではないかと思います。そういうことは特になかったでしょうか。

【島山】今回のプロジェクトでは特に大きな不備はなかったですね。そこはよかったと思っています。

【中原】五月雨式ということですが、依頼した部分の受け入れテストと、次に依頼する部分の設計を、並行して作業するわけですね。そこは日本側でも受け入れテストを担当する人とメソッド記述書を書く人を分けて作業したということでしょうか。

【島山】単体テスト・レベルでの担当者による動作確認は、基本的にすべてツールに任せていましたので、そこで工数を割くことはなかったのですが、結合テストではチームを分けました。結合テストを担当するチームと、他の設計を担当するチームというように分けて、並行で作業を進めました。

五月雨式といっても、依頼単位ごとに分かれてテストできるし、動作もそれなりに完結したシステムだったということで、そういう意味では、システム内のコンポーネント分割がきちんと行っていたと思っています。

【中原】単体テストは、中国側できちんとテスト項目を消化できていれば、ツールを使っただけのテストなので、特に検収しなくても問題なかったということでしょうか。

【島山】そうですね。ツールでテスト実施のカバレッジも測定するようにしていましたので、その結果を見ることで、再度検収を日本側で行う必要はなかったということです。

【藤野】五月雨式とは、サブシステムごとに区切って依頼しているというイメージでしょうか。

【島山】今回はサブシステムというよりもう少し細かい粒度でした。

開発方法、開発ツール

【藤野】テストツールとしては何を使われたのでしょうか。

【島山】同じように Rational Tool の中から Functional Tester や、社内アセットとして挙げられる点検ツールなど使っていました。

【藤野】Functional Tester の中で、C1 カバレッジをどれぐらいカバーしているか、分岐関係などすべてのチェックがされているかどうかということも出てくるのでしょうか。

【島山】そこは Functional Tester の機能で出るのではなく、別のツールを組み合わせ使っていました。

【中原】Rational のツールにはいろいろな製品があるので、総合テストまですべて網羅されるということでしょうか。

【島山】そうですね。ツールを組み合わせ使うことで網羅可能になります。

【藤野】このプロジェクトがうまくいった要因、こういうところを押さえたのでうまくいったというポイントはどこでしょうか。

【畠山】二つあると思います。一つ目は技術面で、日本側で事前にアーキテクチャを設計し、各種成果物の標準化を徹底して、品質を高めたことだと思います。また、Rational のツールを使う、社内のアセットを使うといった資産の有効活用を積極的に実施したということも大きいと思います。

もう一つは、最初にもお話ししたのですが、オフショア先と協業するときにはどうしても管理面でいろいろな障害が生じます。進捗が見えない、あるいは Q&A の内容がうまく管理できず、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、日本側の過剰な工数負荷が問題になるということもあります。そういう懸念点を最初に洗い出して対策を立て、準備期間を経て、適切なツールを使ったりして改善した、それらが成功の要因だと思います。

【藤野】進捗が具体的に見えるようにするという目的でツールとして導入して、オフショア先との作業の仕方を最初から決めたということでしょうか。

【畠山】そうです。

【藤野】それをまたアセットとして登録して、他のプロジェクトでも活用できるようにしたということでしょうか。

【畠山】そうです。他に、Q&A 管理には Rational Portfolio Manager という製品を使い、その利用方法についても詳細なガイドを作成して登録しました。そのようなツールを使いながら、リモートでもうまく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事前に実施していったことが成功要因の一つだと思っています。

【藤野】先程のところで、アーキテクチャを日本側でしっかりと設計したというお話がありましたが、そこでもツールをうまく活用されたのでしょうか。

【畠山】はい。ツールと、あとは社内アセットです。

【藤野】では、アプリケーション・チームは、アーキテクト・チームがきちんと作ったアーキテクチャに従って、業務のための設計に集中すればよかったということでしょうか。

【畠山】そうです。

【藤野】お伺いしていると、UML 使用は前提であって、それよりも、事前のアーキテクチャ設計をきちんと行う、実施期間中の進捗管理をきちんと行うということが成功のポイントなことなのではないでしょうか。UML 使用はもう普通という感じでしょうか。

【畠山】UML 使用は前提とも言えますが、UML を使う上で、図だけでは足りない部分の記述書を追加したり、ユースケース記述書のレベルも細かく指定したりしています。そのあたりが工夫したポイントになります。

【竹政】ツールやアセットなどがもうでき上がっているということですね。IBM の方であればそれがベースになるわけですが、IBM の方以外と開発されるときには、ツールの使い方など、その前提を確保するのが大変ではないでしょうか。IBM グループ以外と開発されることはあまりないのでしょうか。

【畠山】他社の方が入って、マルチ・ベンダーで開発するプロジェクトもあるとは思いますが。そのときにはおっしゃる通り、ツールの使い方など教育の工数が出てきますので、それは一つの考慮点になるのだと思います。その際に、社内アセットとして登録したガイド類などはうまく活用できるのでと思います。

【藤野】中規模、あるいは大規模でも超ビッグでないプロジェクトの場合には、実際に現地に行き、フェース・ツー・フェースのコミュニケーションをとることは多いのでしょうか。

【西川】大規模の場合は日本からオフショア先の現地にも向かいます。オフショア先から日本に呼ぶよりも、こちらからオフショア先に行って陣頭指揮をする体制です。中規模の場合は、最初の段階で行って説明した後、リモートでコミュニケーションをとりますので、オフショア先に常駐したり、オフショア先の方を日本に呼んだりということはほとんどしていません。

【藤野】なぜそういう質問をしたかと言いますと、先ほど畠山さんのご説明にあったように、たくさんツールを使うということは逆に言うと、人数が多いから標準ツールを使わなければ收拾がつかなくなるのではないかと思いますし、人数が少ないプロジェクトでもツールをたくさん使うとなると、今度はその費用がかかり過ぎて、費用的にも大変ではないかという気がして、それでどうなのかなとお聞きしたかったのですが。

【畠山】それはあると思います。このプロジェクトよりも小規模なプロジェクトについては、やり方を変えて、ツールを使わずにうまくやるということもしています。必ずしもツールですべてできるわけではないと思っています。

【藤野】UMLを使うときには基本的に Rational を使われるわけですか。

【畠山】そうですね。ユースケースやシーケンス図などを例えばエクセルで書いていくのは厳しいと思いますので、Rational Tool を使うことは必要であると思います。

【藤野】世の中には Rational 以外のツールもありますけれど、他社製ツールを使うわけにもいかないでしょうね。

【畠山】そういう意味では、使用ツールが標準化されていると言えますね。

【藤野】ツール活用以外に留意されたポイントはありますかでしょうか。

【畠山】ツールと、あとは管理基準ではないでしょうか。例えば進捗管理にしても、単純にこのコードの開発が終わりましたというだけでなく、例えば 100 ステップのうちの何%までが終わったのかなど、どこまで管理するかを視点を入れることは必要だと思います。今回のプロジェクトもそうですし、他のプロジェクト経験を通してそう感じましたので、規模に関係なくどのプロジェクトに対しても管理基準を設定して詳細に管理するようにしています。

【中原】小規模ではツールを使わない場合があると言われましたが、その場合の管理基準はどのように設定しておられますか。

【畠山】小規模の場合も基本的な考え方は同じです。完了基準などの管理基準は細かくするようにしていますが、それはツールを使わなければならないものではありません。

UML の課題

【中原】では最後に、UML を使ってみて、その機能や使い方に関して課題と感じられたことはありますかでしょうか。

【畠山】質問にあった「オフショアに UML を導入する際、必要を感じたもの」にも通じると思いますが、モデリングだけでは不十分なところが少なからずあると思います。今回のプロジェクトであれば、クラスについてはクラス仕様書を作成し、メソッドについてはメソッド仕様書を作成し、というように。そのように UML で不足している箇所を補足するためにどういう資料をどのレベルで作っていけばいいかということがガイドラインとして出せれば、より広く利用されるようになるのではないかと思います。

【中原】その辺をガイドラインに載せればいいのかということでしょうか。

【畠山】そうですね。シーケンス図だけでは表現できない仕様の詳細な部分について、例えばどういう資料を作ればいいのかというサンプルがあれば、よりわかりやすいのではないかと思います。

【中原】インタビューは以上になります。本日はありがとうございました。

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。

成功へのポイント

- Rational のツール、社内のアセットを有効活用し、製造工程での実装範囲が明確になるように、日本側できちんとアーキテクチャを設計した。
- オフショア先に開発を依頼する際の進捗、Q&A 管理等の管理作業の懸案事項をプロジェクト開始前に洗い出し、ツールを使用して事前に対策した。
- UML を使う上で、図だけでは足りない部分の記述書を追加したり、ユースケース記述書のレベルも細かく指定した。