

# 航空会社殿向け映像配信システムで の UML 適用事例インタビュー報告

2011 年 11 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

株式会社 OKI ソフトウェア

Copyright (C) 2011 Oki Software, Ltd. All rights reserved

# インタビューの内容

インタビュー日：2011年8月18日

インタビュアー：中原俊政、竹政昭利

インタビュイー：株式会社 OKI ソフトウェア 原田恭宏

## 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。

## プロジェクト概要

### プロジェクトについて

[中原]では、インタビューさせていただきます。

御社名は株式会社 OKI ソフトウェアで、所属の部署名は情報ソリューション事業部関西支社システム第一部、原田様ということでしょうか。対象プロジェクト名が航空会社向け映像配信システムですね。業務分野としては旅客業になりますか。対象システムの業務内容をご説明ください。

[原田]空港構内にカメラが何台も設置されています、それをカウンターのグランドホステスさんとか、バックヤードの方であるとか、空港整備をしている方々が見られるような映像をパソコンで見るシステムがメインです。ただ、それだけじゃなくて気象情報や、発着の情報などのデータを一緒に扱い、チャンネルを切りかえるとそういった情報が見られるシステムを開発しました。

空港は空港公団の方が使われるシステムと、民間の各航空会社さんが使われるシステムがあり、今回のシステムは、航空会社さんが独自に持たれているシステムになります。

[竹政]エンドユーザーはこれを使えますか。

[原田]エンドユーザーは使えません。エンドユーザーが航空会社の方になります。

[竹政]カメラは何台ぐらい設置されているのですか。

[原田]空港によりですが、羽田だったら 200 台位は設置されています。例えば「ちょっと飛行機が飛んだか見てきて」とならないように各ゲートに設置されています、通路に設置され

ているのを見てそこが混んでいるとか、飛行機が飛び立った後であるとか、待合室にどれぐらい人がいるかとかを見るシステムです。あとは空港施設には整備場もあるので、そういうところの状況を、見られるような形になっています。

データのほうは、発着の情報や、気象情報をホストからデータをもらって、それをビジュアルに GUI 化して表示しています。GUI 化したデータと映像を、チャンネルを切りかえて見られるようなシステムです。

サーバーは UNIX のアプリケーションサーバーで WebLogic を使用し、データは EJB で処理しています。WebLogic の JMS 配信機能をその当時一番最初にそれを使って、データコンテンツを配信実現しています。映像のほうは、OKI が持っている映像配信システムを使用しています。

[竹政]端末はパソコンになるのですか。

[原田]パソコンになります。

[竹政]映像もパソコンで。

[原田]映像はプレーヤーで見えるようになっています。

[竹政]パソコン上のチャンネルのようなもので、映像を切りかえるのですか？

[原田]そうです。パソコン上で切りかえると、ネットワーク上に流れているデータを捨てるようになっています。

[竹政]いつでもパソコンがあればというものです。何人ぐらいの方が同時に使っているシステムですか。

[原田]羽田や伊丹という大きなところにはこのシステムをそのまま入れています。小さい空港は、一括で管理しています。

全国 50 空港あるので、端末の台数は空港によって多い少ないがあります。例えば一番大きい羽田空港だったら端末台数にすると、約 700 台ぐらいはあると思います。同時に見ることに対しては、パフォーマンスが落ちるとかそういうことはないです。

[竹政]全国の空港で御社のシステムのみに入っているような独占状態なのですか。

[原田]いま言ったみたいにメーカーさん、航空会社さんに入れているので。

[竹政]そういうことですね。

[中原]構成はホストコンピュータがあって、サーバーがあって、PC ということですか。

[原田]ホストコンピュータがあって、ゲートウェイがあって、DG サーバーがあって、配信サーバーがあるという形になります。DG サーバーは、ただデータを組み立てているだけのサーバーです。データを組み立てたものを配信するサーバーと、端末という構成になります。

 **詳細資料 3 参照。**

[中原]このシステムは、すべて新規に開発されたのですか。

[原田]これは新規に開発しました。ただ、データの解析の部分についてはほかのシステムで開発していました。EJB で開発した経験はありませんでした。エミュレーションは開発していましたが、今回 Java (EJB 使用) で開発することは初めてになります。

[中原]規模としては数千万規模ですか。

[原田]はいそうです。

[中原]期間は 12 カ月間程度ということですね。

[原田]そうです。

### オフショア先について

[中原]次にオフショア先ですけども、これをインドに出されたということで、中国よりもインドのほうがおつき合いがあったということですね。

[原田]中国もインドもつき合いはあります。インドの会社と中国の会社と両方 2000 年からつき合わせていただいていたいて、中国の会社のほうは、その会社に担当の人がいなくなって、継続的に技術やうちのやり方を理解している者がいなくなったので 2005 年ぐらいにやめています。インドの会社は、会社名は変わっていますが人は一緒に仕事のやり方は理解しています。オフショアで発注をしないと海外発注は利益が上がらないので、オンサイトをずっと使うことはせずに、オフショアで開発をしてもらいました。

今回お話しさせていただいている内容は、2004 年に 1 年かけてやったシステムになります。

[中原]わかりました。オフショア先の企業規模は中小企業とありますけど。

[原田]どこまでが中小なのかということがありますけれども、150 人ぐらいです。インド人の人口からしたら少企業だと思う。

[中原]そうですね。ちょっとその辺はあんまり意味がないです。目安ですから。

オフショア先の UML のスキルの状況ですけど、インドということで UML のスキルを保有していたということでしょうか。

[原田]UML のスキルは持っていました。記述レベルを合わせるためにガイドラインをつくりました。プログラム仕様書ガイドラインを作成し、意識合わせをしました。ツールはラショナル社の Rose を使いました。Rose の使い方をマニュアルにして入力の方法がわかるようにしました。

[竹政]モデルの規約みたいなものですね。

[原田]そうです。ユースケースであれば、いくら決められた定義があるからといっても、書くときの目線が変わったりするので、きちっと整理をさせていただいています。ツールの使い方という書き方にしていますが、UML の認識合わせのためにやっています。

[竹政]こういうものをつくるというのは、ツールを使う上での割と共通的なところでもありますね。

[原田]そうですね。お互い同じ定規の上で UML というものを共有しようと思いました。本来そういうことは必要がないのが UML だと思いますが、やはり自社の人間も含めて読み方、書き方を合わせたほうがメリットがあると思います。

[竹政]UML 自体、ユースケースの書き方とかそのあたりは大体みんなご存じだったということですね。

[原田]そうです。今回のやり方で何を書くのかという工程だけを合わせるような認識ですね。

[竹政]ユースケースレベル。

[原田]そうです。勝手に Rose を使って書けるのですが、書くレベルが少ないとか多いとかがあまりないように気をつけたつもりです。

[竹政]UML の開発は、これ以外にも普通に割合とやられていたのですか。

[原田]オージスさんのゴールドとかシルバーの資格を部下に全員取らせて、勉強はさせていました。お客様によっては UML を書いて持っていったら、こんなもの読めないというお客さんもおられたりするので、お客様との意識を合わせるための要件定義の部分ではあまり使えていないです。

要件定義書から機能をどのように実現するかを納入仕様書にしてお客様に同意を求めます。要件定義を受けての確認のため UML を本来使わないといけないと思っていますが、そのころはお客様が同じ土台にのってきていただいていたので使っていなかったです。

機能をメソッドであったりタスクであったり、グラスであったりとか製造レベルに落とすところを、協力会社を含めて意識を合わせるために UML を使いました。インドとは UML でやりましょうということを使い出した形になります。

[竹政]インドはやはり進んでいますか。

[原田]インドは進んでいますね。インドの方は勉強しても使う機会が少ない部分が結構ありますね。

今回のシステムは WebSphere とか WebLogic が出てきた時代の話で、EJB を知っている日本の協力会社が私の周りにあまりありませんでした。インドの協力会社に聞いたら詳しく説明してくれました。ただ、技術的にもわかっているのですが、JMS と EJB の違いがぐちゃぐちゃだということもありました。だけど Java で書けているところだけでもすごいなと思いました。

[竹政]その辺はやっぱりアメリカから発注が来ているからやっていることなのですか。それとも一生懸命勉強しているのですか。

[原田]日本人が勉強しようとしたら和訳して勉強しないといけないのですが、インド人の技術者たちはその必要がないので、アメリカのサイトで勝手に自分でいつでも勉強できるわけです。やっぱりそれが日本人とインド人の違いかなと思いますね。

[竹政]英語でリアルタイムに情報を収集。

[原田]だから速さはすごく速いです。ラショナル社の Rose もうちがやる前から使っていました、ツールとか技術的な話は向こうのほうがやっぱり進んでいると思います。使ったところがバンガロールとかじゃなくて、チェンナイ（昔のマドラス）の協力会社だったのですが、それなりにインフラを含めて環境を持っているから、リアルタイムに勉強はしていました。

[竹政]インドの人は勤勉ですか。そういう意味では勉強が。

[原田]好きみたいです。

[竹政]なるほど、そんな感じはしますね。

[原田]中国の人も勤勉だとは思いますが、知っているとおっしゃってふたをあけたら、「今から勉強する」みたいな話であったり、実力値にやっぱり差があったりとかありました。インド人もひとによりますが、大方のインド人が知っていると中国人が知っているとというのは、当時私は意味が違うと理解していました。

[竹政]その辺の考え方はちょっと違うのかもしれないですね。

[原田]インド人は日本人に近いですね。どうするのだということに対してのすまないという気持ちであるとか、何とかしようという気持ちがあるような気がします。勤勉、まじめだと思うのでずっとつき合っているというのもあります、そういう意味で今のご質問の内容からすると技術力はあると思います。向こうのほうが進んでいるとは思いますが。ただ当時は仕事が少なく試せる環境を持っていないので、やったことがない人が多かったと思います。

[竹政]勉強はしているけど、机上の知識。

[原田]机上の知識なり、自分でそれを商品にするという部分について、やはり IT をやっている人口の割合にその数が少ないと思います。実践している人が少ないと思います。なので新しいことをやろうとって一緒にやり出すと、すごく一生懸命喜んでやってくれます。

[竹政]すごいいい機会だと。

[原田]そうですね。

[竹政]インドのコミュニケーションは英語で行うのですか。

[原田]当然日本語をしゃべれない人もいるので英語でやっている部分と、私の部下より日本語がうまいインド人もいて、大阪弁でいろいろしゃべったりとかします。すごく丁寧な共通語で、片言じゃない流暢にしゃべられる方がおられたりするし、漢字書けないとよく言いますよね、部下より漢字を知ってるのではないかというように書ける人もいたりして、そういう意味では日本語を勉強する環境を持った会社でした。

もともと創立された人がインドの日本人クラブの創設者で、日本と親交がすごくある方で日本語を教える学校を持っていたりして、そういう意味では結構日本語の勉強はされていました。でも半分以上は英語ですね。メールはほとんど、オフショアとやる分は英語でやりません。

[竹政]英語がメインになるのですね。

[原田]そうですね。だからテレビ会議とかもしましたが、うちの部下や私が英語はそんなに言うほどできないので、すごく時間がかかりました。技術的な話なのである程度できましたが、やっぱりコミュニケーション向こうで通訳の人が必要でした。

[竹政]その部下の方がいる意味ブリッジ SE 的な役割も・・・

[原田]2007年までの発注形態は、日本から直接インドに発注したときに、向こうでやるやり方とか、うちにオフショアの PL を置いてやるとか、向こうにも同じような体制をとらせて置いてやるとか、この3パターンぐらいいろんなことをやっていました。でも、そういうやり方でうまくいかないところも実はありまして、2009年に自社の者を出向させて、こちらと向こうに1名ずつ置いてやるやり方をとっています。  [詳細資料4参照。](#)

ブリッジを置いてブリッジの人が優秀なのはいいのですが、中国人もインド人もそうですが、どうもあの方たちは自分らの同一国人同士の会話がすごく長いです。私らが会議をやったら1時間ぐらいでお話しできるようなことを3時間も4時間もお話しされていてというのが結構あって、本当にそれで伝わっているのかなという疑問もあり、そこをショートカットするやり方を2009年にとっています。

**[竹政]**これは改善されましたか。本社の人間を置くわけですね。

**[原田]**改善されました。要件定義からじゃなくて、プログラムの仕様書からつくる。機能とかそういうところはもう決まっていて、それをどう実現するかというところからつくってもらっています。

データを GUI 化するとか、EJB とかをつくってもらったので変換するとかで問題ないですけど、日本の旅行の仕組みとインドなり中国の旅行の仕組みが、申し込んで行くとかいうところが全然違ったりして、業務の内容が理解してもらえないのです。

文化が違うので、インドの技術者に飛行機に乗ったことがあるかと訪ねても、空港に行つてチェックインしたことがありますかと聞いても、ないと言う方がいっぱいおられる。その人らに物をつくってもらおうとすると、僕らは理解して省いているところが理解できなくて、何でそんなことしなきゃいけないんだとか、何でこんな画面を出さなきゃいけないんだとかいう話があって、いろいろそういうのでうまくいかなかったことがありました。

それを UML といってもそこは難しくて。UML のユースケースで本当は要件定義がきちっとできますというのは、同じ土台に立った人が共通の言葉を省いてやっているわけじゃないですか。このシステムといたら、どういうシステムにしなければいけないかという目的をちゃんと理解できていますね。それを文化の違いでわからない人にこうだからと言っても、それは何のためだとなるわけです。本当にそこを改善できるようなことがあればよかったですけど、当時は駄目でした。

時代が変わってきて、今だと大丈夫だと思います。でも当時は文化的な違いがあったりとか、理解してもらえなかったことが結構いっぱいあったのでいろいろ失敗しました。言ったとおりにやれとか、書いてあるとおりに、UML どおりにつくれと言うのですが、やっぱり人間なのでこうしたほうがいいのか、余計なことを思ったりするわけです。それでうまくいかないことが結構ずっと続いていました。2009年ほううまくできるようにはなったのですけど。

ただ、この自社の者を出向させていることで、向こうに財政のしわ寄せが行って、2009年、2010年の2年足らずで今年ももうやっていません。出向から戻しています。それは仕事量がこの景気の状態でもそこまでというのもあるって、雇えないという話とか。これはこれですごくいい話で、うまくいくモデルだとは思っています。

## オフショア開発時の UML の活用状況

**[中原]**UML をどの工程で使用したか。具体的に、どの工程で、どの UML の図を使用したか

教えてください。

[原田]詳細設計のところですよ。

ここに3枚ほどお持ちしたもので何を使ったか書いています。使い方が合ってる、合っていないはありますが、私たちが理解できるようにつくりです。見てもらえればわかるように、ユースケースで概要であるとか、シナリオであるとか、イベントのフローで全体と個別の処理のケースを書かせて、クラス図、シーケンス図、コラボレーション図を使っています。個々のモジュールの仕様書はステートチャートか、もしくはアクティビティ図を使って書きなさいという定義をしています。  **詳細資料5参照。**

次のページにプログラム仕様書、モジュール仕様書の手引きをつくって、そこからシーケンス図とかコラボレーション図をここでかけとか、パッケージ図、クラス図で代用しなさいとか、またアクティビティ図とかステートチャート図を使いなさいという定義をして、いま言ったようなほぼ使っている感じになります。

[中原]ユースケースを入れて7種類の図を使用しましたか。

[原田]そうです。

[中原]一通り使われたのですか。

[原田]一通りあると思います。UMLを何に使いなさいということまで一応意識合わせをしました。例えばここにUMLのクラス図があれば、そのクラスで何を表現するかの意識は合っているの、見やすいといえれば見やすかったです。フローチャートの記号が決まっているのと同じような感じですよ。本当はガイドラインを利用してという話であればよかったですと思いますが、Rose使いほぼ一般的な使い方だと思っています。

3ページ目に書いたように、UMLでの作成時の注意事項で、OKIにもUML推進チームがあつて、色々相談してRose使用方法もレクチャーは受けました。1からつくっているわけではなくて、ある程度ガイドライン的なものがあつてそれをもとにつくっている形になっています。

世の中一般的な話とあまり変わらないと思いますけど、それにうちの仕様書の構成を合わせているだけです。

[中原]普通、モジュール仕様書とか書くのはないですか。概要、入力、出力、処理の内容などを記載できる表を作成するのではなくて、アクティビティ図とかステートチャート図を作成するのですか。それとも両方作成しますか。

[原田]そこは両方使っていました。

[中原]モジュール仕様書をつくり、入力とか出力はモジュール仕様書に書いて、処理のところを例えばアクティビティ図で書かせたのですか。

[原田]結局メソッドの処理が複雑で、文章でただ書かれるのがすごく嫌なので、そのかわりに、アクティビティ図とかステートチャート図を使って表現させました。文章で書くと英語であろうが日本語であろうがただ書かして、結局何が起きているのかわからない話になると思いますが。文章で書くのをやめるために、その二つの図を使って表現させました。

[中原]わかりました。

[竹政]クラス図などは日本語ですか、英語ですか。

[原田]クラス図は英語です。英語というか、そのままパラメーターとして使える名前です。Java なので、ローマ字を日本語と言われると日本語ですが（笑）。

[竹政]アルファベットではあるのですか。

[原田]アルファベットではあるので。なるべく英語を使わせていたのですが、単語のわからない部下がいて、なぜかわからないけどローマ字になっているところもありました。そういうのがだめだと思うのですが。

[中原]シーケンス図は、どのように作成しましたか。すべてを書くのが大変だと思います。

[原田]書くのは大変ですけど、少なくともある程度メソッドに分ける前の段階で、プログラムメソッドを、入出力なりデータの流れは書かせました。本当は DFD とか使えばいいかと思うのですが、それをいま言うシーケンスでも書くと決めて、データの流れについてはシーケンスを使っています。

[中原]データの流れを。

[原田]はい。

[中原]そうするとシーケンスのそれぞれの送り先、送り元は割合と大きな単位ですか。モジュールとか。

[原田]そうですね、モジュールとかですね。モジュールというよりは、個別処理のところを使っていて、ここでどう流れていくかを見て、モジュールのほうは当然パラメーターの渡しであるとか、どこに書いたとかそういう内容になる。なので、そこではシーケンスは使わずに、その前の段階で流れがわかるような使い方をしています。

[中原]シーケンス図、コラボレーション図を適宜必要に応じて使い分けているのですね。

[原田]そうですね。いま言うシーケンス図として見やすいほうは見やすいように。それで書けといっても書けないという場合もあるので、そこはどっちかで書けるはずなので、それで書いてという話をしています。

## UML のメリット／デメリット

[中原]メリットとかデメリットとかがあれば教えてください。

[原田]さっきも言ったみたいに、仕様の定義が合っているという話が、私は一番のメリットだと思っています。

お互いの仕様書なり仕様の理解のそごをなくすために、どこを見れば何が書いてあるという意識合わせとして、なるべく同じ UML という言葉の世界の中で合わせることはできるのはメリットだと思います。

デメリットは UML にも行間があるので、その部分で補足するのが文章になるとやっぱりそれは齟齬になるので、それを完璧にするのはすごく難しい話だと思っています。なので UML を使うメリットは、同じ土台で同じ言葉である程度意味がわかることだと思います。

[中原]それは国際標準であるから、オフショア先でも使っていますか。

[原田]そうですね。先ほど英語ですかと聞かれましたけど、すべて英語で同じように話せて何かしたら同じことができるのかということ、僕はそうでもないと思っています。よりコンピュータの業界で技術的な話として使うには、UMLは向いていると思います。もともと設計するに当たって必要なものが大体そろっていますから。

[竹政]国内でやった場合とオフショアに出した場合の違いで、何かその差はありますか。

[原田]国内の協力会社は行間も読むし、わかっていることを説明する必要はありません。一方でその結果が思い込みや勘違いであったりすると思うわけです。UMLを使って本当に国内もきちっとやれば、それがどんどん減っていったって品質が上がると思うのですが、いかんせん日本の協力会社さんはあまりそういうものを使いたがらない。それは人口の高年齢層が影響しているのかもしれない。

国内の協力会社もUMLを使えば思い込みとか知ったかぶりがなくなって僕はいいと思いますが、あまり使いたがらないし、できないと平気で言う人もいるし、既存の協力会社さんだとこれを使うから勉強しなさいと言ったらそれなりに勉強してやろうとはしますが、時間がかかりますね。したがって、最初からそれを勉強されて使っている協力会社さんであれば、思い込みや齟齬はなくなっていくと思います。

海外の話にすると、思い込みやそこ以前の問題なので、どちらかという思い込みとかそこよりも、もう少しお互いの意思疎通なり仕様を理解してもらうというのでいいと思います。そこに至っていないので。

[竹政]そこに至る前のところ、初めからという意味で。

[原田]そうですね。

## UMLの適用範囲

[中原]オフショア担当分は、詳細設計工程ですから、そこは全部オフショア側でUMLを使ってつくらせたのですね。それ以前の要件定義と基本設計は御社のほうでつくって提供したのですね。

[原田]要件定義はお客さんとの話ですね。基本設計と言っている部分と詳細設計と、実は物のつくり方としてあまり切り離せないです。これはプログラム仕様書ですけど、この上にシステム仕様書とか納入仕様書があります。さっき言ったみたいにどのようにして要件を機能として実現しますかという話を要件定義書というのであれば、それはうちでいう納入仕様書になります。

要求仕様を実現するためにメソッドや、いま言うプログラムのつくりを設計するなら、基本設計と詳細設計と合わせた形になるので。完全に基本設計と言っているところを、基本設計だとおっしゃるなら、それは使っていません。言っている意味、わかりますか。

[中原]ここで言っているのは基本的な枠組み（基本的な構造）のようなところですか。

[原田]だったら、基本的な構造のところは使っていません。

[中原]使わずに開発できますか。

[原田]使わないというよりは、うちでやっています。

[中原]御社のほうでやってそれも見せているわけですよね。

[原田]見せています。

[中原]それは UML を使う。

[原田]UML を使っている部分もありますが、お客さんに出している部分もあるので、そこはそのまま普通の仕様書で書いているところもあります。特に隠す必要はないので、協力会社には全部出しています。あなたはここしかつからないからとやると、さっきの話じゃないですけど変な質問が来るので、とりあえず全部渡して英訳して、それを意識した上で詳細設計を一緒にしましょうという形です。

[竹政]それはもともと日本語の仕様書になるのですね。

[原田]日本語の仕様書になります。

[竹政]それを何か訳したりされるのですか。

[原田]向こうで訳す部署があって英語に訳しています。

[竹政]英語化して展開する。

[原田]それをちょっと見せてもらおうと、だめなところを直したりとか。やはり専門用語が出てくるじゃないですか。専門用語がわからないのです。専門用語で書かないとお客様は納得してもらえないですよね。例えば航空会社の予約情報は PNR というのですが、PNR という言葉で、そういうデータでお客さんとの間で意味が通じますし。「予約」と言ったら「おまえ、PNR も知らないのか」と言われるような形で、PNR と書いて送ってもこれは何の略だと。そんな感じですね。

[中原]基本設計でお使いになっている UML は、今この中の一部ですね。クラス図、ユースケースぐらいですか。

[原田]ユースケースとクラス図です。若干シーケンス図とかもありますけど。

[中原]要件定義では、UML をお使いにならなかったのですか。

[原田]要件定義はお客様が読めないとおっしゃいました。今は読めるみたいです。そのころは読めないという方がおられてだめでしたけど。

[中原]お客様も進んできたのですね。

[原田]進んできたと思いますね。

## UML を使用して上手くいった点

[中原] UML を使ってうまくいった点は。

[原田]文化の違いの仕様解釈とか認識の共通の土台合わせは、すごく有効だと思います。

[中原]UML は途中から使われたわけですね。それを使う前と使った後で何か具体的な効果はありましたか。

[原田]日本語のものを英訳して向こうで考えて、プログラムとしてとてつもないものが出て

きましたから、何で仕様書に書いてあることがわからないのかというさっきのお客様の話ですね。お客様はわからないのと一緒で、当然わからないと思います。そういうところは UML を使うと格段に違います。UML でうちの間人が書いたものを向こうのインド人が読んで、間違ふことは当然少なくなってくるので。それが日本語だと間違ふことがいっぱいあるので、そういう意味では格段にその品質は上がったと思います。

## 今後のオフショア開発での UML の活用方針

[中原]わかりました。今後オフショア開発で UML を使っていくか。これは積極的に使っていきますか。

[原田]使わないとだめだと思います。プログラム仕様書を書いて渡してもだめだと思います。やはり余計なことが出てきたり、進捗がおくれた理由がこうしたほうがいいとか、改善が期待とか、言ったとおりにつくれと言ってもうまくいかないの、それを UML でおりにつくれと言えばまだ通じるような気がします。

[竹政]じゃあオフショアはデフォルト UML でやりたいという感じですね。

[原田]そうですね、やりたいですね。

[竹政]日本で出す場合は、なかなか UML はまだ出しにくいですか。

[原田]出しにくいというよりは、うちが書いて読めという形になる部分がまだ多いです。なぜかよくわかりませんが、世の中の的にはどうなのですか。例えば今のこの業界で何割ぐらいが普通に使えることになってるのか、すみませんちょっと僕よくわかってないんですけど。

[竹政]欧米と比較すると日本はやっぱり比率が低いと思います。

[原田]きっと理由は、やはり仕様書を書くときの行間を読めるなり、ノウハウを持っているとか、経験者であるとかいう使い回しを日本のソフトウェア業界はやっていると思います。例えばコンペをして仕事を受注して、それを協力会社に出すときにコンペしているかという、ほとんどしてないですね。既存のところを使うなり、それはやっぱり人がわかっているという信頼関係があるのと、無理を言っても聞くとかいろんなことの附帯事項が気に入って使っていると僕は思います。

そこをオープンにして何か持ってないとだめという話になると、それはそれなりにもっと広がるのではないかと思うのですが、今はそういう環境じゃないので、あまりご存じのところはないです。ただ、なるべくうちの部下が書くときに書いているので、それが読めなければできないのもあるので、使ってる部分は使っています。先ほど言ったみたいに、要件定義だけじゃなくて全部出せと言うお客もいるのです。全部出せの中には、例えばソースまで見るといってお客さんがいるのです。

[竹政]すごいですね。

[原田]ソースまで見るといったときに、仕様書で UML といったときに、これは仕様書じゃないというのがあるので、一部使えないところもあります。

[竹政]お客さんが、UML だと仕様書じゃないと言う人もいますか。

[原田]います。

[竹政]読めないから。

[原田]読めないから、ソースを見るのにわかりやすいと思うのですけど。

[竹政]そうですね、逆にソースまで見られるようなお客さんだったら、見てもおかしくないような気はしますけどね。

[原田]例えば日本語の記述が間違っているとか指摘で返ってきたり、仕様書どおりじゃないと言ってくるのですが、それが UML だとすると、これは読めないから書き直せというお客さんも中にはいます。この世界は法律がないじゃないですか。情報処理の業界の中で使わないと認めないみたいな法律が何も無い。よそにはいっぱいありますよね、建築法だとかいろんな法律があるのにソフトの世界だけ法律がないのです。なのでそこを法律で規定すれば、もっと使えるようになると思うのですが。

唯一法律のない業種で仕事をしているので、そういうことも考えたりしました。法律がないからこんなに使わないのかとか、いいものだから使えとかどうだと言ったときに、100 人いたらいくら便利だといっても半分ぐらいは要らないと言いますよね。自分が読めないからだと思いますけど。

[竹政]そうですね。やってきたことを変えなきゃいけないとか、否定するようなことになると抵抗がありますからね。

[原田]そこは日本人はすごく保守的ですよ。

[竹政]やっぱり UML も全部できないので、こうやっていかに効果的に使うか考えるじゃないですか。全部 UML なんてできるものじゃないですよ。だからどこをどう使うかは、やっぱり大変なのですね。

[原田]大変なのですが、ここを見れば UML でこういうことが書いてあると定義するだけで大分違います。数式と一緒にあるので、このものはこういうふうに見ればという認識合わせを定義しておいてあげれば、見やすいのは見やすいです。UML の七つなら七つをどこでどう使おうが、使い方が間違っていようが、UML を使ったという人がいて使ったらそれを読むのはすごく大変だと思うのです。

少し違っていてもきちっとそういうことを定義してあげてから使えば、もう少しよくなると思います。ガイドラインとしました。でもガイドラインもそういう意味では一般的で、それを受けて自分のところである程度かみ砕いてももう少し詳細にしないとだめだと思います。ですから、そうやって使いました。

[中原]UMTP のオフショア開発向け UML 適用ガイドラインは、各社でそういう規則をつくる程度しか言ってないのです。開発時の細かい規則などは、説明していません。

[原田]なので会社によって、もしくはリーダーによって、少しずつ考えが違うかもしれないですね。

[中原]そうですね、そこは多分変わる可能性が高くて。

[原田]これ使わないとだめという人もいれば、これとこれでどちらでもいいという人もいたりすると思います。だけどその意識を統一しないと、やっぱり使っても結構つらいものがあります。記述してあるのはわかるけど記述してあるだけで、使ってるだけで何を目的で使ってるのかわからないので、そこはちゃんと決めてあげないとだめだと思います。

## オフショア開発に UML を導入する際、必要と感じたもの

[中原]わかりました。次の質問で、オフショアに UML を導入する際に必要を感じたものがありましたか？

[原田]やはり仕様の定義をきちんとしてやらないとだめだということに尽きると思います。

[中原]UML を導入する上においてどういうふうに使えばいいか、効果的なことをうまく説明した書籍というか、そういうものがなかったということですか。

[原田]さっき具体的な話をしたような、例えばこういうシステムだとこういうふうにしたという前例があれば、ひょっとしたらいいとこ取りしてどうだとできると思います。今日のインタビューで私が本当に言いたいことは、使うには自分で考えてそのまま使いなさい、UML だったらいいですよと言うだけじゃなくて、中の使い方をきちっと決めてやらないとうまくいかないと言いたいのです。

例えばモジュール仕様書のこういう部分はこれのほうが本当はよくわかりますとか、一応どんな本でもいろんなことは書いてあるのですが、一般的過ぎて、やはりこういう例をいっぱい集めていただいてそれを何例かまとめてもらって、ケースバイケースなのでそこに法律的なものをつくれるわけじゃないと思っているのですが、その自由度があるからいいという話もあるとは思うのです。だけどやはりこういう工夫しましたというのをもっともっと見ていくと、自分をもっとよく使えるようになるのじゃないかなと思います。ただあまりないので、雑誌とかでは時々あったりとかするので見たりするのですが、そんなのだけを集めてというのは、すみません、私はあんまり見たことなかったのです。

[竹政]この辺の使い方とかは、RUP のあたりで若干あるかな。

オフショアガイドラインもそこまでは言ってないです。あとは事例を集めて、ケースバイケースで整理したいとは思っているのです。

[原田]そういう意味では、当時参考にできるものがあまりなくて。今なら若干はあるかなと思うのですが、UML の本と Rose の仕様書とかからこうだよと決めてやった結果だけなので、これ以降の進化はどれぐらいしているかといっても、あまり進化していないですね。もう少し情報を集めやすいといいかなと思います。おっしゃるみたいにそういうのをどんどん開示していただいて、何か参考にできればいいかなと思います。

[中原]今、私もそう思ってやっています。あとはガイドラインについては特にはないですね。

[原田]今、お話しさせていただいたようなことになりますかね。

## オフショア開発時の体制

[竹政]ちょっと戻りますけど、開発体制は日本側が何人で、インド側が何人ぐらいですか。

[原田]さっき1年という話をちらっとさせてもらいましたが、人数ベースの話ですか。

[竹政]そうですね。また構図ですか。

[原田]売価にすると3分の1ぐらい出しています。3分の1と言いながらも、向こうの金額からすると半分ぐらいです。

[竹政]月何人ぐらい。大ざっぱでいいですけど。

[原田]向こうは10人ぐらい動いていたと思います。1年間常時いたわけではないですけどね。オンサイトはそういう意味では1名いました。オフショアとしては増減があって、ほかの仕事もあったので変わっていると思いますけど、MAX10人ぐらいはいたと思います。それが減ったりふえたりとか、減ったりもとに戻ったりとかそういう形だったと思いますけど。

[中原]これは単体テストまでですよ。

[原田]詳細設計から製造・単体テストまで、平均5人です。オンサイトで1人。基本設計は1人ぐらいです。

[中原]御社のほうは何人ぐらい。

[原田]うちは協力会社を入れて7人ぐらいだったと思います。

## UML の課題

[中原]UML の課題という意味ではどういうことがありますか。

[原田]UML の課題ですか。やっぱり自由度が高い分、すべてをUMLで終わらせられないというのは課題だと思います。言葉って一応完結しているじゃないですか。言葉を使うことで日本語なら日本語が完結していて一応すべてをあらわせると思いますけど、UMLは行間とかに日本語を書かないと補完できない部分もあるので、UMLだけですべてが完結しないのが私は課題だと思います。本当はすべてが完結する言語にしないとだめだと思います。

[竹政]そうですね、そのとおりなんです。これ以上のものができるかどうかわかりませんが、確かにそのとおり。

[原田]これ以上のものは難しいと思います。それは日本語であり、英語でありそういうものを含めてすべての人がという意味で共通にできるなら、それこそ共通の言葉でもう終わっていますよね。

[竹政]それこそ世界共通語になっちゃいますね。

[原田]表記法なので、本来は全部定義しないとだめだと思うのです。その表記法を使えばすべてが書けてないといけないのですが、行間を書かなければいけなかったり、補足しないといけないのは、補足のための表記法も何かつくっておけばいいと思うのですが、それは触れないじゃないですか。

[竹政]OCL ぐらいじゃ全然足りませんものね。

[原田]足りないですね。

[竹政]あれは数値的なものぐらいですからね。

[原田]どうしようもないと思うのですが、そのための自由度が高いのと今の部分は結局反比例する内容だと思うので。自由度があるから使いやすいと UML は言っているのですが、そこはどうしようもないことだと思ってますが、そういう部分で何か問題が残ったりすることもやっぱりあるので、そこは弱点という気がします。

## プロジェクトの成功のポイント

[中原]このプロジェクトを成功されたと思うのですが、その成功のポイントは UML を定義に使うということでしょうね。

[原田]成功した要因はいろいろあります。リーダーがしっかりしていたこともありますし、WebLogic の JMS という機能もバグとかいっぱいあったのですが、まあまあうまく使えたところもあります。ソフト的に言わせていただくと、やはり EJB でちゃんをつくったことだと思います。海外を使ったら、よく JMS とまざったりとか、いくら書いてあっても内容を見たら、これは HTML かみたいなコーディングであったりすることがあるのですが、それが UML を使ってそういうことが書けなくしている部分があるので、うまくいったのだと思います。

[中原]リーダーがしっかりしていたのは、技術面もすぐれているし、プロジェクト管理も両方含めてちゃんとできたんですか。

[原田]技術的にすぐれている人間も中にいて、その人が結構頑張ったというのもありますし、リーダーがきちっとプロジェクト管理ができたというのもあります。そういう意味では人材的にいい人材を使ってできたと思いますけど。

[中原]UML という意味ではリーダーがしっかりしていたのと、WebLogic が使いこなせたのと、EJB を使いこなせたということですか。

[原田]UML という意味では、やっぱり UML を使ったことだと思います。UML を使ってプログラム仕様に齟齬がないようにできたことだと思います。

[中原]インタビューは以上です。今日は、どうもありがとうございました。

### 成功へのポイント

- プログラム仕様書に UML を取り入れ、作成する際のガイドラインを作り徹底させた。
- プロジェクトのリーダーが、技術面、管理面の両方できた。
- WebLogic、EJB を有効に活用した。