

# ベトナムでのオフショア開発に おける 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

グローバル・ナレッジ・ネットワーク

Copyright (C) 2011 グローバル・ナレッジ・ネットワーク All rights reserved

# インタビューの内容

インタビュー日：2011年8月12日

インタビュアー：細谷 竜一

インタビューイ：グローバル・ナレッジ・ネットワーク 渡邊 弘平

## ベトナムの会社との協業の経緯

**【細谷】** 御社の UML を使用したオフショア開発の事例についてインタビューさせていただきます。渡邊さんは独立の立場でやられているのですか？

**【渡邊】** メーカー勤務でしたが、現在は独立して自営で活動しています。ベトナムとは、メーカー勤務時代に、たまたま一つの案件がありまして、そのときベトナムと関わりのある知人を介してからのお付き合いで、もう 10 年以上になります。

**【細谷】** ベトナムのほうなのですね。

**【渡邊】** もともとメーカーで組み込み機器開発を担当しておりました。市場としては海外と国内の両方をターゲットとしていましたが、当時は海外事業の比率が非常に高い事業でしたので入社して直ぐに OEM 関係の製品開発の担当となり、ヨーロッパとかアメリカとか、そういう地域を対象にした開発の仕事をしました。合わせて自社ブランドでの海外製品展開も進められていたので、海外との付き合いが結構長くつづきました。

**【細谷】** 海外へは頻繁に行かれていて。

**【渡邊】** 出張ベースで対応してきました。駐在経験はありません。

所属していた事業部では世界のいろいろな地域で開発をやっていましたが、ばらばらにやったので非効率な点が多くトータルにどう効率を上げるのかということいろいろ取り組んできました。その経緯の中で日本にソフトウェアアウトソーシング専門部隊を作って統括することになりました。その組織でインド、中国他と対応して来たのですが、リスク分散の観点から、ベトナムというのもおもしろそうだなということで調査する事になりました。英語が通じる国であるというのと、これからの国だということで、98年に初めて訪問した後、いろいろお付き合いが始まりました。

**【細谷】** ベトナムはどちらですか、ホーチミンですか。

**【渡邊】** ホーチミンですね、はい。

**【細谷】** 98年当時というと、どうやってそういう会社を見つけられたのですか。

**【渡邊】** アウトソーシング専門部隊という事で、日本人以外に各国の関係会社等から人材を受け入れて展開しました。彼らからいろいろ紹介や、情報提供を受ける形で情報を収集してみつけました。

**【細谷】** 最終的にどこか、既に現地にある会社とパートナーになるというか。

【渡邊】 ホーチミン市国家大学（Ho Chi Minh City National University）の教授が設立された会社がありまして、そちらとおつき合いする所から始めました。

【細谷】 それは FPT という会社とはまた違う。

【渡邊】 FPT は国営企業ですね。

当時 FPT はまだ数千人の会社と聞いていましたが、今は2万人を超えていると聞いています。FPT とお付き合いした事は有りません。比較的小規模ないろいろなところとつき合っています。

【細谷】 98 年当時という、ベトナムにはソフトウェアの技術者もそれほどいなかったのではないのでしょうか。

【渡邊】 そうですね。中国、インドに比べると絶対数は少ないのですが、ソフトウェア技術者はいましたよ。欧米との関係、特に主宗国だったフランスの他、米国またオーストラリア等との関係が深く、これらの国からソフトウェアアウトソーシングを受注して事業が行われていました。ただ当時は、日本からの注目度は低かった様におもいます。既にいわゆる大手からの下請をされているソフトハウスさんは出ていらっしやいましたが大手の話は聞きませんでしたね。

【細谷】 そうですか。

【渡邊】 もともとベトナムは、中国、インドに比べるとソフトハウスの規模は小さいので、あの当時でも FPT が最大手で数千人という時代でした、まあ 100 名いれば中堅という感じでしたね。その下という 20 名程度の会社がほとんどでしたから、まあソフトウェア技術者がいないと思われても仕方なかったかも知れませんね。

【細谷】 お仕事されるときは、じゃあ基本的には英語で全部やるという感じですか。

【渡邊】 ええ。海外事業が主体でしたので、基本は英語という形でやっていました。

【細谷】 それはメーカーさんのほうで、日本で開発されたものをいろいろな現地のローカライズするような開発をされるのか、それとも新しいものを、例えばベトナムだったらベトナムで組み込みのソフトとか。

【渡邊】 国内向けの製品に対する周辺ソフト開発が多かったですね。ソフトウェア開発費のコストダウンが主たる目的でした。

【細谷】 社員の皆さんは、英語でそうやってやりとりすることに関して、あまりストレスは感じないですか。

【渡邊】 そういうこともあったので、その中間的なやりとりをする専門組織を 96 年、インドへのアウトソーシングをスタートした時点で作りました。私と部長、その他に日本人スタッフ、インド人、中国人、フィリピン人、ベトナム人、シンガポール人という構成でつくった SE 専任部隊で仕事を受け入れました。

【細谷】 それは日本にそういう部隊をつくられた。

【渡邊】 はい、その組織で社内からのニーズを吸い上げて、そこで必要があれば翻訳も対応しながらやっていました。

【細谷】 じゃあ日本側の現場の方は、今までどおり日本語で仕事しながら。

【渡邊】 そうですね、基本的に日本語です。

【細谷】 それはブリッジという形でもやっている。

【渡邊】 ブリッジですね。従来は個別にブリッジの担当者をその都度設定してやっていたのを、専任部隊をつくったということです。外国人を組織してブリッジ支援をするなどやって、出来るのかという意見もあったのですが。その当時としては珍しい取り組みでしたからね。

【細谷】 それが、いつからですか。

【渡邊】 96年から立ち上げて2001年まで運営しました。試行的な組織ということでやって実行と検証を行い、その成果は別子会社のソフトウェア専門会社へ機能を移管しました。

## 独立後:ITコーディネーターとして、ベトナムを活用

【細谷】 渡邊さんはいつ独立されたのですか。

【渡邊】 4年前になります。

【細谷】 今は、どういう業務をやられているのでしょうか。

【渡邊】 今は、ITコーディネーターというタイトルで、企業さんのいろいろな案件に対応させてもらっています。

【細谷】 それはどういう形で入るのですか。アドバイザー的な形ですか。

【渡邊】 そうですね、技術アドバイザーとして入っているのが一つと、あとはITシステムの運營業務に関する支援業務担当として入っています。システム開発の受託案件があれば行っています。

【細谷】 受託されて、またどこかパートナーに渡すということですか。

【渡邊】 そうです。今回のお話はその開発受託があったときの話です。そのとき開発リソースとしてベトナムの会社を活用したということです。

【細谷】 それは以前からつき合われているベトナムのホーチミンの会社と。

【渡邊】 以前にお付き合いの合った会社から分離した会社に御願いしました。元の会社は100名を超える会社ですが、今回お付き合いしていただいた会社は小規模な会社です。

【細谷】 それは技術的にはC言語の組み込みのほうですか、それともJAVAであるとか。

【渡邊】 二つあります。一つは、PC (Windows) 上で動くホストアプリケーションを、新しい端末機種に対応するように変更していく作業です。二つ目はその組み込みの端末装置開発です。

【細谷】 ごめんなさい、ホストというのはいわゆるメインフレーム。

【渡邊】 いいえPC (Windows) サーバーです。

【細谷】 サーバーのほうですか、はい。

【渡邊】 PC サーバーで端末の管理をする機能を持ち、端末はサーバーと連動、非連動モードで動くような機能を持つシステムでした。

【細谷】 その案件に関しては、ベトナム側は何人ぐらい動いているわけでしょうか。

【渡邊】 そんなに多くはなかったです。最大で5人ぐらいです。

【細谷】 それは日本に来るといことはなくて。

【渡邊】 日本には来てもらいませんでした。全部現地で、オフショアで対応してもらいました。

【細谷】 こちらからは、どなたか行かれていますか。

【渡邊】 私が大体、月に1度か二月に1度をベースに出張対応しました。通常は、メールと併用してプロジェクトの進捗状況の確認を行い、別にコミュニケーションツールとして Redmine を使っていました。あと、ソースコードの制作状況も Redmine を経由して確認を行いました。

【細谷】 そうしますと、いわゆるウォーターフォールみたいなやり方でやられているのか、それとも、もっとアジャイルというかコード主体で進めるようなイメージでしょうか。

【渡邊】 基本はウォーターフォールですが、サイクルは短めにして動かししました。開発は仕様未定なところがありましたので、その辺は開発を3フェーズに分けて、各フェーズ毎に完結させるという格好です。

【細谷】 フェーズとおっしゃいますと、具体的にはどういうことですか。

【渡邊】 既存にあるデバイスとつないだ上で、PC サーバー側の機能拡張、改善をしていくというのがまず最初です。次に新しい端末を接続してシステム構成を確認して行く作業。最後に既存の端末と新しい端末両方をシステムに統合した形で運用ができるかという機能検証の実施という順番になりました。

【細谷】 そうすると、最初にかっちりした仕様書があって設計書を書いてというような感じではない。

【渡邊】 PC サーバーに関しては一応の仕様書はあったのですが、それほど詳しいものはなかったもので、現物とコードで確認を進めました。その後、要求仕様を反映する作業をおこないました。PC サーバー・端末共に MMI (Man Machine Interface) を持ちますので、操作画面仕様書を基本に、データベースの拡張仕様、サーバー/デバイス間通信に関するものに分けて作成しました。

【細谷】 それは、もともとあるものをちょっと変更、あるいは機能強化という形ですか。

【渡邊】 そうですね、はい。

【細谷】 もともとあるものはベトナムのほうで、そのときに初めて、こういうものというのができた。以前から、ずっとつくられていたというわけではない。

【渡邊】 元のシステムは日本で開発された物でした。ベトナム側には、先ず、既存に有る仕様書を説明しました。続いて、変更箇所について打ち合わせを行いました。打ち合わせを基に、要求仕様書を私の方でとりまとめ、発注先の承認を得た上でベトナム側に伝え

ました。その経過を踏まえてベトナム側で設計仕様書の作成をおこないました。

## オフショア開発での UML の活用

【細谷】 では業務知識とかはあまり必要としなくて、技術的な要件がわかれば。

【渡邊】 そうですね。類似なシステムが現地でも使われている物でしたので、追加で必要とされる業務知識は限定されました。業務については現地で説明を行いました。そのときに最初のユースケースをこちらで作って説明しました。同時に、既存の PC サーバー／端末を動作させて説明を行いました。

自然言語で書かれたドキュメントと現場で実機を使った確認に加えユースケースを使っての一連説明で理解してもらいました。

【細谷】 ユースケースも、いわゆるユースケース図というものをお書きになって。

【渡邊】 そうです、簡単なものですが。システムに関わる装置の管理に関する業務とか、システムを実際に利用するユーザーの行動、システムの動作に伴うデータの関係性、を簡潔に書いたものです。

【細谷】 何かツールはお使いですか。

【渡邊】 ツールはいろいろ探したのですが、エンタープライズアーキテクト (EA) というツールを使いました。

【細谷】 あれは高くはないですが、有料ですね。

【渡邊】 そうですね。比較的値段が安かったということと、使ってみて使い勝手がよかったということで採用しました。

【細谷】 ソース生成機能とか、そういったところは使われてないのですか。

【渡邊】 最初にソースコードの分析するのにリバース機能を少し使っています。

【細谷】 そうすると、リバースすると多分クラス図が出てくるという。

【渡邊】 そうですね。

【細谷】 あれはかなり役に立ちましたか。

【渡邊】 概略はどんな感じかなとイメージをつかむ程度で、やはり現実にそれを元に設計を進めるのは難しいかなという印象です。

【細谷】 その後 EA のツールを使って、別途こういうユースケース図を書かれたと。

【渡邊】 そうですね。

【細谷】 その後、ほかに何か UML の図を書くことはあるのですか。

【渡邊】 クラス図他は担当の設計者で作成してもらいました。

【細】 それはベトナム側ですか。

【渡邊】 そうです。

【細谷】 ベトナムにも EA を使わせて。

【渡邊】 そうです。

【細谷】 そうですか。彼らは UML をもともと知っていた。

【渡邊】 基本的には、大学レベルでそういうものを一通り学んでいますね。

【細谷】 では、あまり違和感なく、ずっと。

【渡邊】 そうですね。

【細谷】 何か書き方とか、UML の作法に関してちょっと違うとか、その辺のやりとりはありましたか。

【渡邊】 その辺は私も、UML 図を使いこなしていたわけではないので、作って行くところを見ながら、わからないところがあれば質問しながら進めるという形になりました。絶対にこうでなければいけないと言うものでは無いと考えていたのでお互いが理解出来る所を目指しました。

【細谷】 そうしますと図の種類としては、ユースケース図とクラス図。シーケンス図などはお使いですか。

【渡邊】 シーケンス図はつくりました。システムの動作の中で端末操作オペレーションにより起動されるシーケンスがありますが、併行して動作している PC サーバーでの操作オペレーションと、端末側でのオペレーションが関連しますので、その様なシーケンスをはっきりさせる目的でシーケンス図を作成してもらいました。

【細谷】 それはやはり日本側で、こういうシーケンスだよというのを書いて渡す。

【渡邊】 いいえ、担当者に作成してもらいました。基本になる PC サーバー／端末間通信仕様をこちらで作成し、それに基づいてベトナム側で操作シーケンスを検討して書いてもらいました。

【細谷】 ほかに何かございますか、ステートチャート図であるとか。

【渡邊】 ステートチャート図は上位層の簡単な物を作成しました。通信に関しては、UML 図ではなく状態遷移表を作成しました。

【細谷】 アクティビティ図とか。

【渡邊】 端末についてアクティビティ図の一部をつくりました。端末はそれ程複雑な物でなかったもので、理解し難いと思われる所を作成して止めました。

【細谷】 基本的に対応したのは、ユースケース図、クラス図、シーケンス図。

【渡邊】 そうです。

【細谷】 以前は UML を使ってそういった開発に役立てたというご経験はありましたか。

【渡邊】 以前から使いたいと思っていましたが機会がありませんでした。実際に自分の担当する業務で使ったのは今回が初めてです。

【細谷】 今回使ってみてどうですか、UML の使いどころとか、思ったのとちょっと違う部分とかはありますか。

【渡邊】 UML を使いたいと思ったのは、96 年にインドの会社と仕事を始めるときでした。インドへ仕事を出すにあたり、仕様書は日本語のドキュメントが基本で、多少の図は書いていましたが、ほとんど自然言語でのドキュメントでした。その様なドキュメントを

(翻訳して) 仕事を依頼するということに対して、やはり彼らとしては非常に理解しにくいという話をした事を覚えています。技術力は非常に高いのですが、我々の提供するドキュメントからその書かれた要求の背景まで理解するというのは難しいと話した記憶があります。

そういうことで、その当時 DFD の利用も話題になっていましたが、DFD で仕様を全て置きかえるのは難しいと考えていました。データを中心に構造を示すためにある程度役に立つのですが、組み込み機器での適用はもう一つだなという話をしていました。UML はその当時まだ仕様が統一されていなくて、各作図法がブーチ (Booch) とか提唱者の名前で呼ばれていましたね

DFD に代わって、そういったものが利用できないかという話はしていたのですが、結局その当時は UML の利用は見送りました。UML1.0 が出てから、実際に利用する可能性を気にかけていました。そうこうしている中で UML2.0 が発表され、それをサポートするツールが安価に提供されるようになってきたということもあり、適当なプロジェクトに巡り会ったので使ってみようと言う事になりました。

**【細谷】** ではもう大分以前からクラス図とか、オブジェクト指向の表記をウオッチされていきましたか。

**【渡邊】** そうですね。図的表現による仕様定義については注視していました。DFD もそうでしたが、コンピューター能力が上がり、ツールが非常によくなってきたので利用しやすい環境になって来たと思っていました。

**【細谷】** 今後こういうオフショアを扱うようなケースでは、どういったところが有効とされますか。

**【渡邊】** 基本的に UML は自然言語での限界を補完するものであるという事でしょうね。やはり図で見て認識することは全体を把握しやすいですね。その意味で、こういうものを積極的に使っていくべきだと思います。

**【細谷】** ということは言いかえますと、まずは全体を俯瞰するという役割と、あとは日本語でだらだら書くと伝わらないものを図で補うというようなイメージ。

**【渡邊】** そうです、どうしても言語で仕様書を記述する場合は、以前から指摘されていますが、“行間を読め” という様に、“その組織の常識” という言葉にあらわされる“暗黙知”を表すことが難しいと思います。その辺を仕様書を図で表現したら 100%書けるのかと言えばそうではないのですが、“暗黙知”がどこら辺にあるのかが分かりやすくなるという気がします。

**【細谷】** 情報の入り口のような形。まず全体を俯瞰して、例えばこのクラスはもうちょっと中を見ようと思ったら、そこからドキュメントとか、ソースコードとかに入るというような。

**【渡邊】** 全体のオブジェクトの連結を示したときに、仕様に書かれた振る舞いがなぜこのオブジェクトで実現されるのかという見方ができるので、文章を一行ずつ読んで頭の中



でそれを構築してみるよりも、UML 図で描かれたもので俯瞰すれば疑問となる所が多分早く出るように思います。

## Redmine で UML モデルを共有

**【細谷】** UML でかかれたものはいつもファイルをメールで送っているか、サーバーに、Redmine のほうに登録されている。

**【渡邊】** 基本的に Redmine の中にため込んで情報共有をしました。バージョンの変更が行われた場合にも、変更前の内容が確認出来る様にしました。

**【細谷】** そうするとベトナム側は、どこか変更があったら変更があった箇所もすぐ特定できるということになっているわけですか。

ツールによるスケルトン生成とかがありますが、実際、こういった機能もお使いになられていたのですか。

**【渡邊】** そうですね。クラス図をかいた後でスケルトンが生成されるので、その機能を利用しました。スケルトンの評価までは行っていません。

**【細谷】** あれはいかがですか、生成するまではいいのですが、生成した後にソースをいじったりするとなかなかリバースしても同期したりするのが大変なので。

**【渡邊】** その部分は難しいですね。今回はソースの変更に対応してリバースするという事はしませんでした。

**【細谷】** アンケートのご回答に、「設計レベルに応じたドキュメントの作成容易化」とありますが、設計レベルというのは具体的にはどういうことですか。

**【渡邊】** それは最初に申しましたが、上流設計工程ではユースケース図でシステム、ユーザーの振る舞いを俯瞰できたということ。構造設計レベルではクラス図を階層的に作成して行く事で途中から参加したエンジニアへのトランスファーが比較的容易に行えた事。今回、最初の段階で2名エンジニアが設計をすすめ、コーディングの段階からエンジニアが少しふえました。その人たちに説明するのに階層化されたクラス図を見せて説明を行う事で理解が早かったと聞いています。そういう意味で、UML 図を採用したのは良かったと思います。

**【細谷】** ここは説明としては、図があります。それをまた実際に会議か何かで1回説明されるというような感じですか。それとも、ここに書いてあるからまず見て、質問があったら聞いてくださいというような。

**【渡邊】** 基本的には、書いてあるものを見てもらい確認を行いました。あと、工程ごとにレビュー検証を随時行い結果を反映させました。規模もそれほど大きなものではなかったので、日程を決めて会議を開くというフォーマルな形は取りませんでした。

## 分散開発におけるコミュニケーションの問題

【細谷】 その規模は2名とおっしゃいました。

【渡邊】 上流工程2名でスタートしました。別に端末設計担当に1名を採用しました。また、PCサーバーのコーディング段階で1名追加がありました。

【細谷】 大体何人月ぐらいベトナム側が作業したのですか。

【渡邊】 全体で約30人月になります。

【細谷】 そうですか。それなりの規模にはなりますよね。この人数ですと、結構期間は長い。

【渡邊】 10カ月です。

【細谷】 結構長いですね。それは御社が案件を受託して、御社のほうでベトナムに発注する。

【渡邊】 東京のパートナー会社が受託したプロジェクトに開発の責任者という事で参加しました。

【細谷】 その東京側とか、もっとその上の発注者の方とのやりとりでは何かUMLを使ったということはありませんか。

【渡邊】 発注元及びパートナー会社とも Redmine を使って成果物は、共有出来る形にしていました。発注元に対しては、適宜進捗ミーティングにて説明させていただきました。

【細谷】 ベトナムとのコミュニケーションは、実際に行かない場合、例えば Skype でやるとかというようなことはあるのですか。テレビ会議のようなものとか、電話会議とか。

【渡邊】 基本的にメールで対応しました。その他には、電話、チャットを使いました。最初のプロジェクトの運営に関して出張頻度が多いと感じられたかもしれません。これまでの経験から新たなメンバーと仕事を始めるときはなるべく顔を合わせて人柄を確認して進める様にしてきました。こうした接触を省略すると、如何に優れたコミュニケーションツールを使ってもお互いの溝を埋めるのは至難と思います。

【細谷】 Skype のほうのチャット。

【渡邊】 そうです。Skype のチャット機能を使いました。

【細谷】 それはパソコンでチャットを開きっ放しにして、いつでも質問するような。それとも、ちゃんと「Skype を開いてください」と言って操作させる。

【渡邊】 事前にお互いのスケジュールがありますので、時間を予約しておきました。

【細谷】 ベトナムと日本の時差は。

【渡邊】 2時間です。

【細谷】 じゃあ、あまり時差というのは気にならない。

【渡邊】 そうですね。気になる程の時差では有りませんね。

【細谷】 これがもしもっと時差が大きかったら、やっぱりコミュニケーションはしづらくなる。

【渡邊】 そうですね、インドとだと4時間半の時差ですね。これくらいになるとつらい

場面がありました。ヨーロッパ、米国とでは更に大きな時差になりますね。

【細谷】 地球の裏側みたいなもので。

【渡邊】 昼、夜逆転になってしまいますので、時差の大きな国と仕事をするときは仕事の調整が大変ですね。

【細谷】 そういったところでは、例えばやっぱり、どちらかのほうが夜まで待って会議して。

【渡邊】 時差の大きな国と仕事をする時は、大体向こうに少し早く出てきてもらって、日本は時間を遅らせて対応するというパターンが多かったですね。

【細谷】 そういう環境でも、例えば Redmine とか UML とか、そういった道具立てで仕事するという可能性は考えられますか。

【渡邊】 そうですね。メーカー勤務時代ヨーロッパ、米国の開発拠点にワークステーションを配置し、ソフトウェアの資産共有を行おうとした事が有ります。国内ではワイドプロジェクト (WIDE PROJECT) が始まった早い時期から、ワークステーション、インターネットを利用する環境が利用できましたが、全世界を繋いだ現在の様なインターネット網の構築はまだままだでした。一応拠点間を繋いだネットワークが構築されメールを使って連絡をとっていました。しかしながら、情報共有の手段はメールと FTP を使ったファイル交換位で苦労しました。現在の様な優れた情報共有ツールが存在していれば時差のハンディを乗り越える助けになったと思います。

【細谷】 多分、今だと時差がなければ国をまたがっても、あまりコミュニケーションはそんなに苦労なくできるからという気もします。

【渡邊】 ええ。ベトナムは2000年以降急速に情報インフラが改善されてきました。その点は10年前とは雲泥の差を感じます。

## 中小企業と付き合い易いベトナム

【細谷】 そのベトナムの会社とは、同じ会社でずっと10年お付き合いされていますか。

【渡邊】 そうです。同じ会社ともつき合っていますし、そこからいろいろ分かれた会社ともずっとお付き合いさせてもらっています。

【細谷】 じゃあ今後も GKN さんとしては、オフショアといたらベトナムがメインというふうに。

【渡邊】 その辺はケース・バイ・ケースになると思います。アウトソーシング先として、まず第一に中国の会社という選択肢は当然あると思います。中国には日本語に非常に堪能であるエンジニアも多く、距離も近いというのは大きなメリットです。逆に、アウトソーシングの仕事規模に、そこそこの規模を条件とする会社が多いと言う点では、我々中小零細企業はちょっとお付き合いしにくい所が有ると思います。

次にはインドという選択肢は、まず日本から遠いという距離 (時差) の問題と、あとカ

ルチャーギャップの問題が有ります。日本人がインドとつき合う場合、このカルチャーギャップが結構しんどいんじゃないかと思います。それと、年成長率20%を目標にした規模を追うというスタイルが基本的にあると思います。この点は中国にも似た所がありますね。

ベトナムはドイモイ（経済開放政策）以来、経済成長を続け振興していますが、企業規模はまだまだ中国、インドとは比較にならない小さい会社が多いです。そういう意味で、お付き合いを始めやすいし、上手につき合いをしていけば、長くつき合っていく可能性があると思います。あと、文化的な面で我々に非常に近い文化を持っていると思います。

**【細谷】** 中国と比べても、やっぱり近いところがありますか。

**【渡邊】** そうですね。宗教的は大乗仏教で日本と同じ流れです。また米作中心の農耕民族であると言う点も似ています。都市の近郊へ出ると、日本の田舎の風景に非常に近い景観が広がります。文化的な面では人の肌ざわりに親近感を感じます。中国では、常に喧騒の中で生きていると言う感じでいつも落ち着かないと思うのとは違いますね。

**【細谷】** そうです、疲れてしまいますね。

**【渡邊】** ちょっとほっとするところがあります。その他の地域では、フィリピンも居心地の良さを感じますし、非常に優秀な技術者もいます。もう30年近くお付き合いしている友人もいますが、政情が不安定だったなどの影響で今はビジネス上のお付き合いがありません。

## 次の発展段階を迎えたベトナム

**【細谷】** そうですね。政情という面では、ベトナムはそれなりに独裁国家なので逆に安定しているというか。

**【渡邊】** そうです。まず共産圏であるということですね。共産圏のいいところは、基本的に教育の機会が均等に与えられていることです。昔は文盲率が高かったと聞いていますが、今はクオックグー(國語)と呼ばれる文字が使われ識字率は上がっています。ベトナム語の表記法としてフランスがローマ字の様に導入した物です。これによりほとんどの人が読み書きができるという事です。大学教育のレベルは、何度か視察したのですが思っていた以上に高いと感じました。ポートピープル等難民として海外に出られた方の中から祖国復興に貢献しようと戻ってこられた教育者等思いを持って帰国された人材の多い事が影響しているかも知れません。

**【細谷】** じゃ結構優秀な方が、そういう産業をつくられているという。

**【渡邊】** そうですね。そういう国を支える意識を持った方が現場で活躍される事で、つくられたところもあるなという感じもしました。

**【細谷】** ラオスとかミャンマーとかはいかがですか。

**【渡邊】** そういうところで事業展開をしないかと言う話もあったのですが、一つは絶対

人口が少ないというところが問題ですね。カンボジアには徐々に企業が進出して来てはいますけど、産業インフラの面とか政情面とかロジスティックスの面とか考えるとまだ出にくいかなと言う印象があります。情報産業が出ていくというのは、ちょっと先になるのではないのでしょうか。今は、労働集約型企業（繊維産業、製造業の一部とか）が更なる低賃金労働力をともめて進出を始めているという状況の様です。

**【細谷】 たしか、最近ではメイド・イン・ミャンマーを見かけることがあります。**

**【渡邊】** 少しずつ目にするようになって来ていますね。今後、あの地域が発展するにはエネルギーの問題を含めた社会インフラ整備を解決する必要が有るでしょうね。ミャンマーも天然資源（石油、天然ガス）は出るけど、自分のところで加工できないところに問題があります。ベトナムも、つい最近までそうですよね。石油は出るけどガソリンは作れないで、シンガポールから輸入していたと聞いていました。

**【細谷】** そうしますと、割とインフラ面とか通信インフラも含めて、ベトナムは大分もう今はよくなってきている。

**【渡邊】** そうですね。今では IT ビルと呼ばれるビルが結構あちこちにできています。ホーチミンの空港のすぐそばにも E-Town と呼ばれるビルが建設されています。

**【細谷】** そうですか。私は 3 年前に行きましたが、気が付きませんでした。

**【渡邊】** 2000 年前まではサイエンスパークと呼ばれる、公園のような敷地内に平家の研究棟を配置した施設が主体だったのですが、2000 年を過ぎたあたりから E-Town の様に、無停電電源、ネットワークを完備し外観も先進的なデザインを採用した IT ビルが作られるようになってきました。現在もその様な施設は拡充されて来ています。

**【細谷】** そうですか。今でもちょくちょく行かれますか。

**【渡邊】** ここ 2 年ほど行っていませんが、それまではよく行ってました。多分 50 回以上訪問しています。

**【細谷】** 50 回、すごいですね。1 回何時間かかるのでしょうか。

**【渡邊】** 関西空港から片道、行きが 5 時間、帰りが 4 時間です。

**【細谷】** 意外とシンガポールに行くよりは近いですね。

**【渡邊】** 1 時間近いです。

**【細谷】** それはホーチミンですね。

**【渡邊】** そうです、ホーチミンです。ハノイにも何度か行った事は有ります。ハノイだともう 1 時間弱早くなります。

**【細谷】** そうですか。ベトナムというのは市場としていい立地というのか、それとも労働コストが安いということか。

**【渡邊】** 1998 年当時人口は 7,800 万人あり、市場としての魅力も多く企業は感じていたようです。また労働賃金も中国と比べ優位制が有ったという事で、どちらが主で企業が進出をしていたかというのはよく分かりません。私が担当したソフトウェアアウトソーシングでは安い労働コストというのが魅力でした。

市場として期待して進出した企業は購買の中心となる中間所得者の層がまだ薄く、やっと今になって期待した市場としての形が出来て来たようです。

**【細谷】** ベトナムですと、やはり中国よりもさらに一段安いという感じですかね。

**【渡邊】** そうですね。日本語での対応が出来ない等の部分は有りましたが、単純比較すると中国に比べコスト優位性がありました。課題はリソースの量です。大量の人的リソースを必要とする場合、ベトナムではその受け皿がありませんでした。

現状を比較しても中国とのコスト差では優位性があると思います。人的リソースに関しては政府も注力しているようですが、中国に優位性が有ると思います。

## 英語で仕事をすれば選択肢が増える

**【細谷】** コストという意味では、今つき合われているベトナムの会社だと、差し支えなければどれぐらいの単価というかコスト感覚で使えるのでしょうか。

**【渡邊】** 仕事の内容、規模により代わってくると思います。例えば、Windows で何らかのデータベースを使ったあまり複雑でないビジネスアプリケーションを委託する場合、かつ小規模なソフトハウスに委託する場合、US\$2,000 以下で発注する事が可能だと思います。

**【細谷】** それは英語で仕事を受けられるのですか。

**【渡邊】** そうです。英語で対応するというのが前提です。

**【細谷】** 英語で仕事ということでは、そういった会社は結構見つけやすいですか。

**【渡邊】** 2000 年頃までのベトナムのソフト産業はほとんど外需、いわゆる米国等の英語圏及びフランスからのアウトソーシングで有ったと聞いています。

現在でも多くの会社が WindowsAP, Mobile Solution, Main Frame(COBOL 等)の業務のアウトソーシングを引き受けています。

2000 年以降は内需も増加し、オープンソースの利用も活発化して来ている様です。

**【細谷】** たしかホーチミンというのはフランス留学の人がいて、英語はわからないけどフランス語ぺらぺらという人も結構いたような印象がありますけど。そうすると、わざわざ日本語をしゃべる会社を見つかり、あるいは育成したりするよりは、英語でやるといほうがやりやすいですか。

**【渡邊】** 以前勤めていた会社でソフトウェアアウトソーシング専任組織を作ったときの一つのコンセプトは、基本は英語でやりましょうということなのです。

理由はリスク回避が容易にできる事。英語をしゃべる国は複数ありますよね。インドでもいい、フィリピンでもいい、ベトナムでもいい。もちろん、ほかの国でもいいですけど。

日本語をしゃべれるところと限定してしまうと、どうしてもそこに依存してしまう。前職では中国のパートナー企業と仕事もしました。パートナー企業という事で技術者への日本語教育に熱心に取り組んでくれ日本語(検定)の1級を持っている技術者も随分育成し、仕事は楽でした。ただ、何かあったとき(例えば新型インフルエンザの発生)の代替枠は

なかなか見つからない。

それから、場合によっては転注をしても良いのですね。英語で対応していれば、ある所でうまくいかなかったら、仕様を全部引き揚げて転注もできます。実際に、そういうことをやったケースもあります。

**【細谷】** それは何かトラブルがあって、ということですか。

**【渡邊】** ええ、そこはどうしてもうまくいかなくて。これはインドの会社との事ですが、あるものを任せたのですが、それがどうしても完成しないのです。その一つの原因は漢字です。一般には、私達がアラビア文字、デーヴァナーガリー文字（ヒンディー語）の文字が認識出来ないのと同じ様に、彼らは漢字を文字として認識していないので、フォントの一部が欠けていたり、（全角や半角、かな、カタカナ混じりによって）文字間隔が不揃いであったりしても気にならない。ということで、結局どうにもならなくなって、それは別のところに頼んで完成させました。だけど、仕様書はそのまま転注先に任せました。仕事の内容により向き、不向きみたいところがあることを学習しました。

逆にインドで非常に成功した例もあります。そのケースは、ある種のアルゴリズム開発でしたが、非常に大きな効果がありました。ですから向き、不向きが、ある程度はあるという前提で発注先を検討する事が必要ですね。非常に優れた技術、人材がそろっているからここが良いと思っていて、全てのプロジェクトが部成功するという事ではないかも知れないということです。場合によってはあるところまではここで作るが、そこから先はこっちで作ったほうがパフォーマンスがいいというもあり得ると言う考え方も必要です。そういう意味で、仕様書や関連するドキュメントを英語にすることによってポータビリティが出る。

ただ、そうすると別の問題として、発注者、受注者の間のそこをどう埋めるかと言う問題が出てきますが、そこは翻訳家をうまく間に入れるとか、あとは発注者側からプロジェクトの様子を見たいというときに少し工夫をしていくとかしてやってきました。

**【細谷】** そういうポータビリティという面を考えても、一つは英語と、例えば **UML** を使うと多少楽になることもあると考えてもいいでしょうか。

**【渡邊】** そうですね。ポータビリティを高める為に、**UML** を利用するのは良いアイデアだと思います。**UML** を使う事により、対象の全体を俯瞰する、各要素の関係性をつかめるなどは自然言語の文書で表記しきれない事に対応出来ると思います。図を使うとしても独自の物ではなく、標準化された記法に従った **UML** を使えば読み方を説明する必要も有りませんね。

アウトソーシングの事例として、おもしろい事例をもう一つ上げます。ある装置開発プロジェクトで、基本となる中核部分の **OS** とかファームウェアまわりは日本で、アプリケーションは中国で、中国へ装置を持っていけないので、シミュレーション検証環境関係はベトナムで作らせるということをやったことがあります。



【細谷】 分散開発ですね。

【渡邊】 適材適所で、コストを最適化させていくという事で実施しました。

【細谷】 それはうまくいきましたか。

【渡邊】 それが非常にうまくいったのです。

【細谷】 そうですか。それは何か特別工夫された点がありますか。

【渡邊】 先にお話した専任組織に所属していた中国人 SE にアプリケーション開発の担当を任せました。専任組織内で活動していたので、日本側の状況も十分に理解しており、プロジェクト実施検討の最初の段階から担当してくれたので仕事の落とし込みがスムーズに行きました。また、その他の部分も同じ専任組織内の担当者が動いたので情報を共有しつつ進められたという事が効果を上げたと思います。

## オブジェクト指向設計と実装言語は分けられる

【細谷】 先ほど、UML みたいなものは結構以前から使用しているということですが、渡邊さん自身は、オブジェクト指向技術みたいなものは全般的にはだいぶ前からやられていたのでしょうか。

【渡邊】 私自身は、興味は有ったので自分で勉強はしたのですが、本当の所よく理解できていないと思っています。以前の会社ではオブジェクト指向設計に組織的に取り組みを強化していました。社内研修として外部講師を招いて若手技術者のトレーニングを実施したり、(既存のプロジェクトにオブジェクト指向設計を取り入れるにはハードルが高いため) 特定のプロジェクトで試行的にオブジェクト指向設計を取り入れる等の様子を管理者として横から見っていました。

【細谷】 ただ、やはり組み込みの世界だとまだまだC言語で、あまりオブジェクト指向のほうには。上のアプリケーションのほうは JAVA など知らず知らずのうちにオブジェクト指向になっていますけど。もっと下の OS に近いところなどはCの世界かなとは認識しています。

【渡邊】 そうですね。基本的には、私の勝手な理解ですが、オブジェクト指向だからC++、C#でなければいけないという話ではないのではないかと思います。対象となるシステムをどういう機能単位にまとめられるかということが基本になって、その構成を記述するのに JAVA などのオブジェクト指向言語を使うのが有利だろうという理解をしています。

【細谷】 そのまま落とし込めますからね。

【渡邊】 しかし、対象システムをどのように構成するかというのが重要なので、C言語など設計から落とし込む場合、制約のある言語で作られた物でもまとまった形で再利用が出来るようであれば、それはそれで一つのオブジェクト指向の形ではないかと思っています。



【細谷】 一種のモジュール化みたいな形ですか。

【渡邊】 ええ。本当のオブジェクト指向とは言えないかもしれませんが、機能単位のインタフェースに抽象化された概念を持ってくることで、それに近いものが得られるのはいかと考えています。

例えば、“お金を支払う”という機能モジュールにメソッドとして“支払い”、インスタンスとして“指定金額”を指定すれば、モジュール内部で硬貨、紙幣を管理し適切な金種で支払う振る舞いを閉じ込める、という考え方を指定する事になる、というのでも意味があると思うのですが。

【細谷】 そういうお考えがあって、実装部分はオブジェクト指向ではないかもしれないが、設計レベルでは UML とかそういったことを。

【渡邊】 はい、そういう観点から UML を利用した仕様記述は有効だと思っています。

【細谷】 確かにそれに似たご意見をお持ちの方も結構いらっしゃって、オブジェクト指向というのは考え方だから、実装言語に依存するのはおかしいじゃないか、JAVA じゃないといけないとか、C#でなければいけないというのはおかしいとおっしゃる方も、時々やはりいらっしゃいます。

【渡邊】 そこまでは言わないです。理想的にはオブジェクト指向という方法論とそれを実装する C#とか JAVA とかいう世界で実現するのがいいのでしょうけど。きれいなオブジェクト指向の形にはならない制限はあるがそういったものの考え方を使うことによって、効果を得る事が出来るのではないかと思います。

## 上流工程でのモデリング技術の活用

【細谷】 そういう意味でいうと、いま結構 UML の世界もトータルに非常に C++ に寄った、実装寄りのデザインを絵であらわすというところから来ましたが、最近はビジネスプロセスをあらわしている。あとは、SysML という組み込みの仕様法をあらわすというほうにかなりシフトしてきています。

【渡邊】 そうですね。私見ですが、やはりソフトウェアとかシステムデザインというものが、“実装”からスタートはしてないという事ではないかと思います。システムが利用される社会の動きの中で、システムが必要とされる場面が理解されずに仕様定義してもあまり意味がないし、よくわからない物になってしまうでしょうね。上流工程での仕様ミス、判断ミスが下流工程で発生するミスの修復とは比較にならない大きな修復工数を発生する問題を防止するためにもその様な打ち手が必要とされているのでしょうね。

## UML 適用ガイドラインの活用

【細谷】 最後になりますが、UMTP のオフショア開発向け UML 適用ガイドラインをダ

ダウンロードされましたが、それは特にプロジェクトで使うためということではなくて、参考にされるためにということですか。

【渡邊】 最初にそれを読んでおけばまだよかったです。実はプロジェクトが終わってから、いろいろと後の整理とかという中で調査しているときに WEB で検索して見つけました。こういうガイドラインが公開されているのであれば先に読んでおけばよかったと思いました。

【細谷】 それは内容的には結構役に立ちそうですか。

【渡邊】 そうですね、基本的な考え方がよく整理されていて、皆さん随分努力していただいたと思いました。なので、まず導入される場合のテンプレートという形で使われたら非常に有効ではないかと思います。要らない苦勞をしないで済んだかもしれません。

【細谷】 次回はぜひご活用いただいて。またご意見がありましたら、ぜひお寄せいただければと思います。

今日はどうもありがとうございました。

【渡邊】 ありがとうございました。

#### 成功へのポイント

- 日本語に依存しない。英語で仕事をすることで、海外パートナーの選択肢が増え、仕事のポータビリティが上がる。
- UML は情報の俯瞰のために有効。モデルは Redmine のようなプロジェクト管理ツールを使って拠点間で共有することで、バージョンや進捗が可視化できる。