



モデリングツール Elapiz BE オフショア開発 でのUML適用事例インタビュー報告

2009年12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 © 2009 特定非営利活動法人 UML モデリング推進協議会 All rights reserved

株式会社オージス総研

Copyright © 2009 OGIS-RI Co.,Ltd All rights reserved

目次

開発の概要	3
UML/ガイドラインの使用状況	5
UML を使って良かった点・困った点	9
今後もオフショア開発で UML を活用するか.....	11
オフショアに UML を導入する際に必要なもの、ガイドラインについての意見・要望	12
成功へのポイント.....	13

インタビューの内容

インタビュー日付：2009年6月23日

インタビュアー：長田 真理重（一部 中原俊政、竹政昭利、藤野博之）

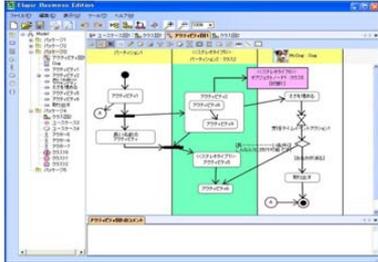
インタビュー：正田 墨

開発の概要

まずプロジェクトの概要を教えてください。

ご紹介するオフショア開発の概要

- 開発対象 自社製品: Elapiz Business Edition (UMLモデリングツール)
※ 自社で仕様コントロール可能、長期プロジェクトという点で
オフショアに出しやすいプロジェクト
- 委託先 上海 I 社 → 上海欧計斯軟件有限公司
(上海オージス:SOT)
- プロジェクト規模
 - 要員 日本側: 2~6名
要求管理、アーキテクチャ設計、外部設計、
(リファレンス実装)、受入テスト
 - 上海側: 3~25名
(外部設計)、詳細設計、実装、テスト、
技術サポート、ユーザマニュアル作成
- オフショア期間 2004年12月 ~ [継続中]
※ 本プロジェクト自体は、2004年8月より
- 開発言語 C#



Copyright (C) OGIS-RI Co. Ltd. All rights reserved.

対象は、Elapiz BE (イラピス ビジネスエディション) というツールです。

UMLとかビジネスモデリングをするためのツールですので、業務アプリケーションの開発とか組み込みの開発ではなくて、パッケージツールの開発が対象になります。

開発言語は C#で、2004年の8月に開発をスタートし現在も継続中です。オフショア先は、中国上海です。特殊事情として、当社の関連会社を2007年10月に設立したため途中で委託先の会社が変わっています。ただ、主要メンバーは移籍してもらいました。

それでは、開発期間について教えてください。

開発バージョン	開発期間	体制(人数)		上海への委託範囲					
		日本	上海	外部設計	詳細設計	実装・単体テスト	結合テスト	システムテスト	ユーザマニュアル作成
v1.0	2004.12~2005.12	4	3→11	-	△	○	△	△	-
v1.1	2005.12~2006.8	6	11	△	○	○	○	△	-
v1.2	2006.8~2006.11	3	12	○	○	○	○	△	-
v1.3	2006.11~2007.2	2	14	○	○	○	○	○	△
v1.4	2007.2~2007.6	2	16	○	○	○	○	○	△
v1.5	2007.6~2007.8	2	16	○	○	○	○	○	○
v1.6	2007.8~2007.12	2	25	○	○	○	○	○	○
v1.7	2007.12~2008.2	2	25	○	○	○	○	○	○
v1.8	2008.2~2008.5	2	14	○	○	○	○	○	○

バージョン 1.0 が 2004 年 8 月から 2005 年の年末ぐらいまでですね。8 月というのは日本での企画・計画の開始なのでオフショアは 12 月開始です。それから、翌年の夏 2006 年 8 月ぐらいまでがバージョン 1.1 です。その後はだいたい四半期に 1 回ぐらいのペースでマイナーバージョンアップしているという感じです。現在の最新バージョン 2.1 ですが、私が関わったのはバージョン 1.8 までです。

開発規模はどうでしたでしょうか。

当初は日本側と上海側が同じぐらいずつの 3~4 人の規模でスタートしています。開発が進むにつれ、上海側の人数が徐々に増えて 10 人余りに。バージョン 1.2 で日本側が急に減っているのは、私が上海に半年間常駐して上海側に技術移転を進めた時期だからです。機能を大幅に作りこんだバージョン 1.6 と 1.7 では、一時的に上海側の人数が増え 25 人体制となっています。

正田さんが上海に常駐をされている間、上海側の方々に上流工程も出来るように教えられたのでしょうか。

はい。常駐する前から上流を任せていくという方向性がありました。もちろん最初は、日本側でかなり多くの部分を担当し、実装とか単体テストとかだけを上海に任せていました。一部設計を任せるとか、テストの範囲も広げていくとか、徐々に常駐する前から準備は進めていました。それでも最初のバージョンは、それなりに動くものができるまでに 1 年ぐらいかかっています

最初の 1 年間の状況をもう少し教えて頂けないでしょうか。

機能的に最初のもの作りなので、土台となる部分を作らなくてはいけないという意味で開発ボリュームももちろん大きいのですが、UML を使用してオフショアに出すオーバーヘッドもあったと思います。試行錯誤の時間とか、教育コストも含まれて、1 年間という時間がかかってしまいました。モデルは読めるけど、十分には書けないという状態なので教育しながら進めていったという感じですね。

今は、上海側が UML を読めて書けるようになっているのでしょうか。

読めて書けるメンバーがいる、ということですね。全員ではないですけど。最初は、ユースケース図とかなり詳細レベルの設計まで落としこんだクラス図と、そのクラス図を説明するための主要なシーケンス図も日本で書いていました。上海側には理解を確認の意味で、日本側のシーケンス図を参考にして上海側でもシーケンス図を作成してもらいました。なので、基本的には読めれば良いというところからスタートしています。その後、徐々に設計も上海側に任せましょうということで、設計レベルのクラス図やシーケンス図は上海側で作ってもらいました。最初の 1 年間はこれも完全に任せきりではなかったという状態ですね。

その後、画面や操作性などの外部設計やシステムテストも任せ、マニュアルとか日本語のヘルプも委託しています。日本側が最後まで握っているのは、要求定義です。つまり、どういう機能をどういう優先度で実現していくかは日本側で決定します。

現在は要求をするだけで、もの作りとテストを含めた残りすべてを上海側でできるようになっているということでしょうか。

はい、そうですね。自社のパッケージツールなので要求をコントロールできるという点と長期にわたって続けられるプロジェクトだったという点が、かなり広範囲にわたって上海に任せられた理由だと思います。お客様向けのシステムを3ヶ月間で仕上げて終わりといったプロジェクトでは、難しかったと思います。あとは、そうですね、会社は変わっていますが、主要メンバーが変わっていないというのも大事な点かなあ。

中国の社会全体で頻繁な転職が問題になっているようですが、主要なメンバーは変わらなかったのですね。

中国は離職率高いので、キーマンが一度にいなくなってしまうと、プロジェクトが長く続いても、上流から任せるのは無理だったと思います。どちらが先か分からないですけど、他のオフショア企業に比べると上流の責任ある仕事を任せてもらえるというところが、モチベーション向上につながったのかなと思えるところもあります。

UML/ガイドラインの使用状況

UML 適用ガイドラインの使用状況について教えてください。

ガイドライン v2.0 の適用ポイントごとにいくつかトピックスをお話します。

開発プロセス・マネジメントの「見える化」－ アクティビティ図による作業指示

4th Iteration 以降の開発作業の進め方(Draft)

1. 開発を上海に委譲する範囲

Activity 図で示す通り、設計 Model の一部の package を除く詳細設計、Sub-system level Testcase の作成を OGIS から 上海 に委譲する。

開発作業フローをアクティビティ図で示した例
Copyright (C) OGIS-RI Co. Ltd. All rights reserved.

ポイント1番の「作業範囲/作業分担の明確化」ということでは、途中で1回混乱があって私が上海常駐したタイミングで、常駐しているがゆえに、日本側の立場なのか、上海側の立場なのか、

お互いよく分からなくなっていて、責任分担があいまいになった時期がありました。ガイドラインを読み直したときに、「あれ、昔は明確化していたはずなのに、今はちょっとまずい状況になっているな」ということで、もう1回再確認したといった事象もありました。ちなみに役割分担は、アクティビティ図で定義していました。

ポイント2番の「利用するUML図の確定」については、開発の概要でも説明しましたように、最初はクラス図・シーケンス図を日本側で書いて上海は読むだけ、徐々に上海がUMLで書く範囲を広げていきました。その時々で日本側・上海側がどの図をどう使うかを明確にしていました。

オフショア開発でのUMLの利用

	ダイアグラム	利用状況		用途
		初期	中期 後期	
構造図	クラス図	◎	◎	
	オブジェクト図	-	-	コラボレーション図を利用
振る舞い図	ユースケース図	△	-	
	シーケンス図	◎	△	上海側に実装前にシーケンス図を描いてもらうことで仕様理解度を確認
	コラボレーション図	△	△	必要に応じて
	ステートチャート図	△	△	必要に応じて
	アクティビティ図	○	○	作業分担の指示に利用
実装図	コンポーネント図	△	△	必要に応じて
	配置図	-	-	

利用したダイアグラム(UML1.5)

Copyright (C) OGIS-RI Co. Ltd. All rights reserved.

ポイント3番「必ずUMLである必要はない」に関しては、適材適所でUMLを使用しています。要求定義以外は、すべて上海へ委託といたしましたけれども、時々、リファレンス実装というか、実装のサンプルを日本側で作って上海側に渡すということもしています。リファレンス実装を渡すことによって、理解のギャップを埋めることができます。ドキュメントの中にはお絵描きツールで描いた画面も入っていますし、テストではExcelの表を使っています。UMLではないですけど、「見える化」というか、視覚的に分かりやすいようなものをなるべくコミュニケーションツールとして使うように工夫しています。

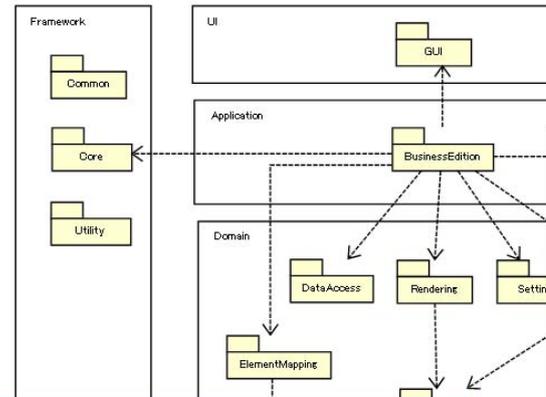
ポイント4番「上流工程への参画」は、既にお話したように要求定義以外のかなりの部分を任せられており、本プロジェクトの大きな特長の1つです。

ポイント11、12番の「アーキテクチャ・モデル」「アーキテクチャ説明成果物の作成」にはエピソードがあります。開発スタート時に日本側で作成したアーキテクチャモデルの説明会も実施したので当初は上海側も認識していたはずなのです。それがいつの間にかお蔵入りになっていました。上海側のファイルサーバーの中には存在していたのですが、後からプロジェクトに加わったメンバーは存在すら知らない、ということが途中で発覚しました。反省点としては、最初に見ただけではダメで、それがきちんと使われていることを確認することが大事なのです。

大事なこと、複雑なことをUMLで正確に伝える

アーキテクチャモデル

アーキテクチャモデルは当初日本側で作成し、実装・テストのみを上海へ委託。開発プロセスもウォーターフォールの品質を慎重に確認。



コンポーネントに基づく
アーキテクチャを使用すべし

複雑な状態遷移の表現
には自然言語で伝える
より効果的

2.1.2.4 図ごとの印刷設定 Dialog へ遷移

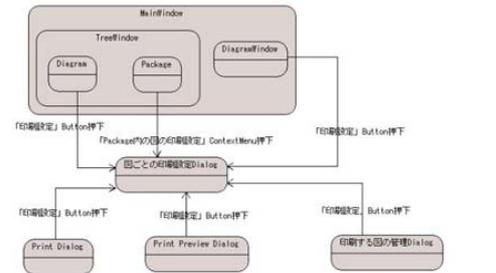
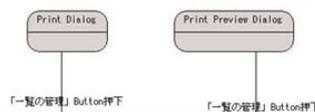


図 2-4 図ごとの印刷設定 Dialog への遷移

2.1.2.5 印刷する図の管理 Dialog へ遷移



Copyright (C) OGIS-RI Co. Ltd. All rights reserved.

ポイント 16 番 「仕様未決定部分は明確に」も失敗があつて、日本と上海で同じ部分を作っていたことがあります。日本側としては全体像を理解してもらうつもりで日本側担当部分のクラス図も提供したのですが、意思疎通がうまくいっていなかったようです。その後は、UML 図上のパッケージの色分けで担当を区別するようにしました。

ポイント 17 番 「オフショア側での成果物のレビュー実施」ですが、常駐して実際のレビュー状況を確認してみると、確かに形式的には時間も取って実施しているのですが、内容的にポイントについてレビューをしていないということが分かりました。そこでレビュー用のチェックリストの作成などの改善をしてもらいました。

ポイント 20 番 「モデルで詳細設計のレビュー」に関しては、モデルでレビューしているからこそ任せられる部分が増え、中身の実装の細かいところまで気にしなくてもよいというところがあります。レビューの観点の 1 つが、ポイント 21 番の「UML 図間の整合性」です。UML ツールを使っていればツールが整合性を保ってくれるので、UML の理解度が多少劣っていてもツールが補完してくれます。ツール利用のメリットですが、逆に上海側メンバーの実際の UML の理解度レベルに気付くのが遅れたというデメリットもありました。

[中原] このプロジェクトが 2004 年からスタートして、ガイドラインが出来たのが 2007 年です。ガイドラインを見て新たに追加したとか、追加してこういう効果があったとかはありますか。

正直目新しいことはないのですが、これだけきちんと網羅されていると、チェックリストとして有用だと思います。先ほど言ったように、スタート時点ではアーキテクチャモデルを作って説明したのにお蔵入りしているとか。作業範囲/作業分担の明確化も、忙しくなっていくとつい忘れてしまつて曖昧になっているとか。1 年に 1 回もしくは半年に 1 回ぐらいこのガイドライン

でプロジェクトの状況をチェックすると、プロジェクトのその時々の弱点が浮き彫りになると思
います。

[藤野] 最初に1年ぐらいかけて教育されていますが、最初からラボ的なことまで意識して、育成計画を立てて教育されたのですか？

最初からそんな綿密な計画があったわけではないのですが、以前にインドのオフショアで痛い目にあった経験があり、ある程度上流まで任せていきたいという思いはありました。またツール開発ということで、相当コストを安く作らないと、ビジネス的に継続ができません。日本側で設計もやるし、作ったものの受け入れ試験もやるし、とやっていると、あまり日本側が楽じゃないとか、日本側がボトルネックになるわけです。受け入れ試験を綿密にやっていると、次のバージョンのスペックが考えられないので、どうしても手抜きになっていきます。そうすると、中途半端なものが出来てきて、悪循環となってしまいます。中途半端に実装とテストだけ委託するというのは、日本側が苦しかったというのが正直な感想です。上海側にたくさん出さないと、オフショアを使っているメリットが感じられないという意識が現場にもありました。

[藤野] UMLの教育にはどれぐらいの時間をかけたのですか。

UMLの教育は、実は体系的にはほとんどやっていないです。たまたまUMLツールを作っているということで、副次的に理解している部分はあるかとは思いますが、たとえば1ヶ月間教育するとかいうようなことは、当初はしていない。すればよかったかなとは、今では思うのですが。

[竹政] UMLの図を段階的に理解してもらっているようですが、どこまで理解していれば任せられるとか目安がありますか。

実装、テストだけであれば読めればいいので、いわゆるL1とかいうレベルで、逆にUMLよりも実装能力の方が問われるのかなとは思いますが。しかし、一部と言えども設計を任せるあたりからは、1人でもいいので、オフショア先にモデルをきちんと書ける人がいないと厳しいと思います。

[竹政] 体系的な教育というよりも実際にモデルを書いてもらい指摘しながら育成してきたということですね。

そうですね。でも、それは結構特殊事情だと思います。ある程度書ける人が最初からいて欲しいとは思いますが。英語とそれこそUMLでコミュニケーションし、試行錯誤で苦労しました。

[藤野] 最初はUML適用ガイドラインもまだなかったわけですが、最初からガイドラインがあれば、立ち上がりは変わったでしょうか。

多少は、変わったかもしれないですね。

[中原] UMLを使った効果が出だしたのはどの頃だったのですか。

実感として、効果が出始めたのは1年半とか2年ぐらいしてからですかね。もうちょっと上海側に委託する範囲を広げてもいいかなと思って実行すると、問題が起きて、また日本側に作業を戻したり、試行錯誤をしているので、1、2年ぐらい経ってから回り始めたかな、という感じですかね。

[竹政] 設計から分析を任せるところのギャップというか、そこはどんな感じでしたか？

分析を任せる前に、外部仕様の一部を任せましたね。日本語ができるテスターのエンジニアに3ヶ月間東京に来てもらって、お客さん側の視点に立って外部仕様を一部作ってもらいました。そのテスターが上海に帰ってから分析を任せたので、ある程度上海側でどういうゴールになるかを理解したところで、分析をスタートしているのでスムーズにいきました。

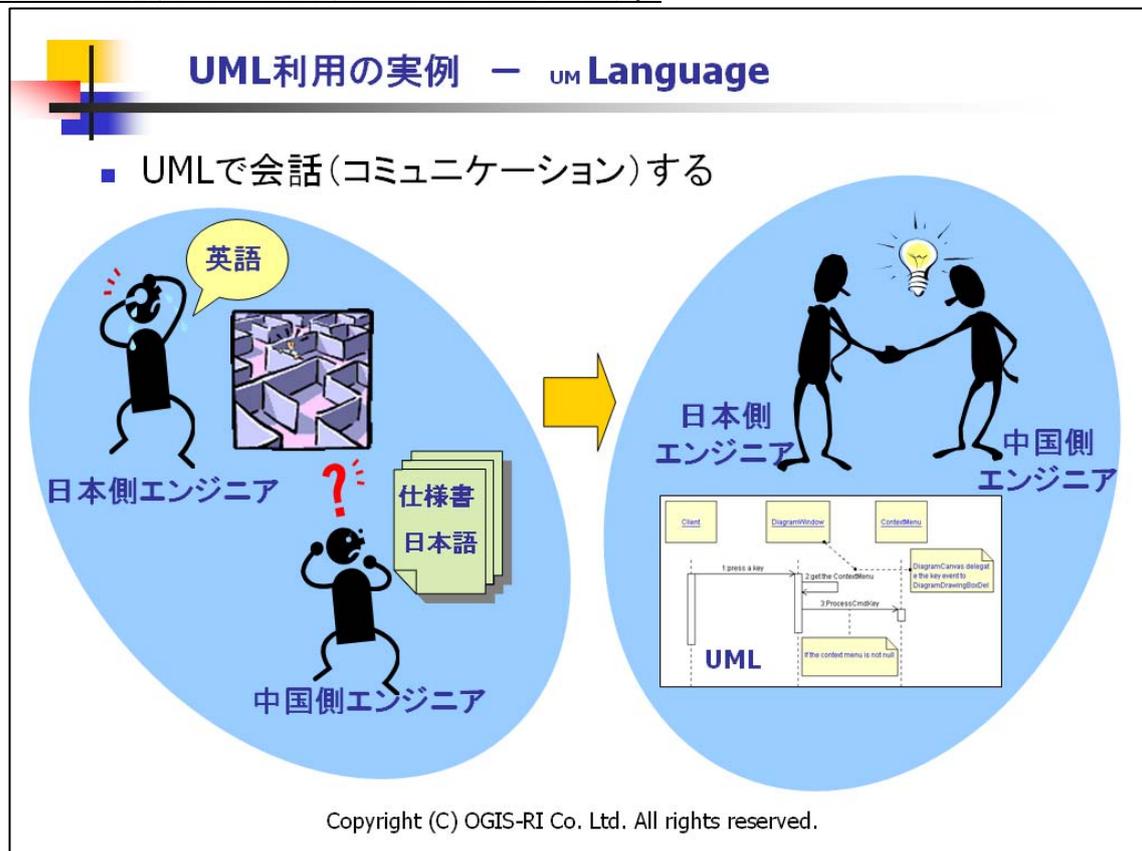
UML を使って良かった点・困った点

UML を使ってうまくいった点はどのような点だと感じていらっしゃるでしょうか。

逆に UML がないと開発はできなかつたのかなあとと思います。日本語ができるエンジニアが増えてきたとはいえ、ドキュメントベースで長い日本語を書いたりするとどうしても理解度は落ちるので、UML で仕様を書くということによって、何を作らなくては行けないかというものに対する理解度は格段に高まっているという風に感じます。それを UML なしで、日本語にしても英語にしても中国語にしても、説明するのは非常にしんどいのかなという風には思います。

UML を使うことによって相手が理解しやすいようになったと感じていらっしゃるのでしょうか。

そうですね。後は、上海側がどれぐらい理解しているかというのを日本側が測る意味でも、上海側が書いた UML の図をレビューするのは意味があることだと思います。上海側のスキルもある程度みえてきます。一方通行ではなくて、両方向で役に立っているのかなあとこの風に思います。たぶんそれは、相手が上海ではなくて日本人同士でも拠点が離れて開発していたとすると、やはり UML を使うメリットは出てくるのかなと思います。



物理的に距離がある場合、それは外国でなくても、UML を使う方がコミュニケーションを取りやすいと感じていらっしゃるのでしょうか。

そうですね。複雑なものを伝える場合とか、モデルでレビューすることによって品質を確保したい場合には効果が高いと思います。

モデルを使って品質を確保するというのはどういうことなのでしょう。

モデルの段階でレビューするので、C#で書いたソースコードレベルの細かいところを見なくても、おおよそ間違っただけでできていないということが確認できます。逆にソースコードを全部見なくては行けないとすると、ほとんど自分で作ったのと同じようにコストがかかってくるので

すが、モデルをレビューすることによって、それほどコストをかけずに品質が高いものを作ることができるわけです。

一方で、問題点は何かありますでしょうか。

そうですね。悪いところは、モデルでレビューしてよしとしているので、実装レベルでのネーミングルールとかまでは、正直、目が行き届いていませんでした。本来であればそこもきちっとコントロールすることで、メンテナンスしやすいソースコードになります。それには、人間の目でレビューするだけじゃなくて、もうちょっと環境とかツールとか、そういった仕組みを作らないといけません。上流はモデル、下流はツールでチェックするという組み合わせが効果的だと思います。

他にも何かありますか。

さっき程も話しましたが、UMLを使うのであれば、最初にドカンと集合教育をすれば結果的には近道だったのではないかと思います。ある程度知っているメンバーもいたので OJT 的にやってきたのですけれど、時々「あれ、本当は分かっていたのかな」と思う問題に直面したのも事実です。今にして思えば最初に集合教育はしておけば、結果的には開発時間を短縮できたかもしれません。

[藤野] 読むほうは大丈夫でしょうけど、書かせるときに問題が発生するということですね。

そうですね。設計は比較的似たパターンを横展開していただけなので大丈夫ですが、分析を任せる前に集合教育をしておけば良かったと思います。全員ではなく、キーマン 2、3 人の育成だけでも効果があったと思います。

[藤野] UML 利用の効果があったというのはどの点ですか。レビューの指摘件数が減ったとか、バグが減ったとか、工期が減ったとか、生産性が上がったとか。

一番顕著なのは日本側の要員を減らせているところです。そもそも最初のやり方では日本側だけが忙しく、日本側が頑張ると上海側は定時で帰るという状況で、何のためにオフショアに出しているのかよく分からないような状況でした。UML を使って上流から委託することで、日本側の工数が減り、それでもユーザーに使ってもらえる品質のものが出来上がってきているというところです。

[竹政] インドの時と比べて中国ではどのへんがよかったとかはありますか。

インドの時はとりあえず作っちゃって、バグのもぐらたたきゲームに突入したのですけれど、中国委託時にはその反省も含めて、徐々に徐々に任せていったってということもあり、当然バグはありますけれどもコントロール不能な状態には陥っていません。

[藤野] よく言われるのは、ソフト開発に関してインドは優秀で中国はインド程ではない。逆に中国の場合は日本のやり方でやってくださいと指導までするけれども、インドの場合にはインド側のやり方について口出しできない、そういう観点はどうですか。

そういう観点もあると思いますね。ただ、インドにしても最初から日本側がコントロールしてなくてはいけなかったと思います。でも「我々は CMM レベル 5 持っています」とか言われると、「そうなの」ということで、事態を把握するまでに大分時間がかかったという反省があります。

[藤野] インドでも UML は使っていたのですか？

はい、使っています。

[藤野] では UML があるからといってトラブルが防げるわけではないということですか。

そうですね。インドから提出された UML の成果物をレビューしていました。確かに UML の知識は格段に上、というかアーキテクチャを任せられるぐらいの人がゴロゴロとは言わないけど確かにいます。そのエンジニアが真剣にやってくれたらすごいものができるだろうな、っていうのは中国ではないですね。ただ末端までいくと似たような状況なので、いくら優れた人がいても、

訳が分からない人がたくさん入ってきてしまうと、出来上がったものは訳が分からなくなってしまいます。

[藤野] UML で記述された仕様を実装するときにバグが発生したということですか。

アーキテクチャ、分析の途中ぐらいまでは UML ベースで非常にうまく進んでいたと思います。そこから設計、実装と進むにつれ、UML の知識もない人間がどっと投入されて、混乱してしまいました。

[竹政] モデルとソースコードにギャップができたということですね。そのコントロールがインド任せになってしまったので…。

分析ぐらいまでは素晴らしいのですが、設計あたりでは似たような機能を持ったクラスがそこら中に現れ始めるといった状況。逆に上流は素晴らしかったので、任せりゃ大丈夫だと(笑)。

[竹政] 中国の方は始めから怪しいなと思ってやっていたので、大きな問題になっていないということですね。

[藤野] 非常に貴重な話ですね(笑)。裏を返せば会社の規模が大きくても小さくてもちゃんと見るところまで見なきゃいかんということですね。

そりゃそうです。

[竹政] モデルをうまくいったり来たりするような形で細かくチェックしていけば何とかできるけれども、逆に UML を使用していてもポンと任せたりであまり見ていないとダメだよ、という話ですね。

UML を使っても、最初はやはり苦労しますね。相手のレベルが分からないから、どこまでチェックしたらいいかも分からない。立ち上がり時は、相当きちんと見なくては行けないのかなとは思っています。

今後もオフショア開発で UML を活用するか

「積極的に使用する」、「使用しない」、「条件により使用する」の中から今後の UML の活用について教えてください。

使用しないということはありません。個人的には積極的に使用したいですけれど、オフショアということであれば「条件により」ですね。先にも話したように、条件としては、たとえば長く続けられるプロジェクトであるとか、仕様が比較的コントロールしやすいとか。

オフショアで UML を使用する条件は、長く続けられるプロジェクトであるということですか。
パッケージ開発でなくても、システムの維持管理とかであれば、長く続けられますね。システムに対する理解が徐々に上がっていくことができます。結果的に一定期間後には、ある程度上流から任せることが可能になります。

[竹政] ユーザードメインの知識や UML 的な知識ということですね。

そういう意味ですね。

もう 1 つ、仕様をコントロールできるとはどういったことでしょうか。

ガイドラインの補足部分にも関係するのですが、日本におけるシステム開発の特徴として、基本的に仕様変更が前提となっています。エンドユーザーレベルで比較的短い間に仕様変更が入ってしまうと、それをオフショア先に伝えるというのは難しいわけです。だからある程度仕様の変化が少ないところを切り出して任せるとか、仕様変更が入ったとしても優先順位を調整できるようなことができないと、オフショアに出すオーバーヘッドを吸収できないですね。

[藤野] 長く続けられるプロジェクト、仕様をコントロールできるプロジェクトというのは、オフショアに出すときの条件であって、たまたま UML が使えればそれに越したことはないのだけ

れども、今の2つの条件というのは、UMLを使う／使わないの判断にはならないのではないですか。

そうですね。加えて言うなら、お互いにUMLを知っているという条件でしょうか。

[中原] 業務アプリの場合はどうすればいいのでしょうか。

先ほども言いましたが、システムの維持管理とかであれば可能性があると思います。でも3ヶ月単位のバラバラのプロジェクトでは難しい。

[中原] そこなんですよね。開発の大半がそうなんですけど、安定したお客様の仕事ばかりでないので、そういうところでオフショアを使うにはどうすればよいか。

システムの維持管理でも、それなりの覚悟が必要ですね。

[竹政] 業務は、1ヶ月や2ヶ月では分かりません。中国から日本に来てもらって何年か開発してもらうか、あるいは日本側が中国に行って何年か一緒に開発するかといったことも必要なのではないですか。

資本まで入れてという話になれば出来ませんが、全然違う会社だといつ仕事が切れるかもしれないので難しいですね。

[中原] 今後の課題ですね。

オフショアにUMLを導入する際に必要なもの、ガイドラインについての意見・要望

オフショアにUMLを導入する際に必要を感じた物があれば教えていただけますでしょうか。

集合教育をすればよかったという、さっきの話ですかね。これも話をしましたが、下流側のソースコードチェックツールもあれば良かったですね。

[中原] お勧めのツールなどはありますか？

コーディング規約のチェックツールなども大事なかもしれませんが、メトリクスツールとかでしょうか。モデルの段階ではいったん見ているのですが、実装に落ちたらグチャグチャになってしまったところを救うために、いちいち全部読まなくてもある程度統計的にソースコードレベルの品質を表してくれるものとか。

[中原] クラスの凝集度とかですか。

それもあります。あるいは、1クラスの中に大量のメソッドが詰まっているものを分解するとか。それはそれで使いこなす技術は必要かとは思いますが。上流でモデルレビューをしているのはいいのですが、ついつい下流がおろそかになってしまうので、テストツールも重要です。

[竹政] モデリングツールに求めたい機能は何かありますか。

赤ペンで添削するようなイメージで簡単にコメントが書けるといいですね。モデルの上に1枚透明なシートがかかっているようなイメージです。添削は透明なシート上に記載して、はがせばモデルには影響していないというのが便利です。

「オフショア開発向けUML適用ガイドライン」についての意見やご要望事項があれば何でも。

初めて見る人にとってみれば、やはりサンプルとかが欲しいですかね。サンプル、テンプレートですね。用語辞書にしても、命名ルールにしても、サンプル、もしくはここらへんにそういう情報がありますよ、とか。

[中原] 初めて見る人は言葉では分かって、じゃあどう使おうか?と思うので、あった方がいいですね。

[藤野] 読めるだけの活用レベルと、実際に書けるレベルまで持っていく活用レベルというのが
あるのでは。いろんな活用のレベルがあるということと、それに向けてどんな教育をしていか
なければいけないということが書いてあればよいのではないのでしょうか。

そうですね。

[藤野] 今日の正田さんの話だと、設計まではある程度読んで理解できるだけでもよい。書いた
ものが良いものかどうかの判断することが難しいということですね。

そうですね。逆にそれは、オフショア側だけじゃなくて、日本側もそうでしょうね。いいか悪い
かの判断ができないと。

[竹政] たぶん、読めれば比較はできるはずですね。こっちとこっちのどっちのモデルが優れて
いるかといえば判断できるレベルはあるのだけれども、モデルを1つだけを見たときにどこが
悪いっていうのが分からないと。だからそれを誘導してくれる誰かが「このへんはどう？」とい
う話をすれば、たぶんそこを見直すことはできる。

そこまでいかないと、UMLを使用する効果は薄いですね。でもそれはオフショアだからという
話ではなくて、なかなか日本でUMLが普及していかない根本的な問題がそこにあるような気は
します。

以上です。ありがとうございました。

成功へのポイント

- 定期的（1年もしくは半年ごと）にガイドラインの各ポイントの適用状況をチェックする
ことで、プロジェクトのその時々弱点が浮き彫りになり改善に生かすことができる。
- UMLをオフショア開発で利用するのであれば、委託範囲は実装・単体テストに限定せず
上流（分析・設計フェーズ）にも参画してもらう方が品質・費用の両面から効果が高い。
- オフショア委託先に上流から参画してもらう場合、分析フェーズを任せる前にキーとなる
エンジニアにUMLの集合教育を行なった方がよい。